ArabicNLP 2026

The Fourth Arabic Natural Language Processing Conference

Volume 1. Proceedings of the ArabicNLP 2026 Conference

October 28-29, 2026

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

Platinum

Google

Gold

مناظرات قطر
qatar debate
عضو في مؤسسة قطر
Member of Qatar Foundation

Silver

مختبر كامل
CAMeL Lab

Aram Lab

RESEMBLE.AI

أكاديمية طويق
Tuwaiq Academy

©2026 Association for Computational Linguistics

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 979-8-89176-473-6

Introduction

Welcome to The **Fourth Arabic Natural Language Processing Conference (ArabicNLP 2026)**, co-located with **EMNLP 2026** in Budapest, Hungary. This year's edition is held under the theme Towards Inclusive Arabic NLP,a call for language technologies that serve the full range of Arabic varieties and the full range of their speakers. ArabicNLP 2026 is the eleventh edition of the WANLP/ArabicNLP series, which has developed a growing reputation as a high quality venue for researchers and engineers working on Arabic NLP, where they share and discuss their ongoing work. The first in the WANLP series was held in Doha, Qatar (EMNLP 2014), followed by Beijing, China (ACL 2015), Valencia, Spain (EACL 2017), Florence, Italy (ACL 2019), online with COLING 2020, online with EACL 2021, a hybrid event in Abu Dhabi, UAE (EMNLP 2022), and then in-person events in Singapore with EMNLP 2023, Bangkok, Thailand with ACL 2024, and Suzhou, China with EMNLP 2025Ḟor this year's edition of ArabicNLP, we received a record 193 valid main conference submissions, more than double the 95 received last year. Of these, 164 received at least one review, the remainder having been desk rejected before review for anonymity, formatting or completeness violations, and 139 went through the full review process, in which each paper was read by at least three reviewers and then by one Area Chair, a role newly added to ArabicNLP this year. We accepted 37 papers, for an overall acceptance rate of 22.56% across all submissions that received at least one review. This is the most selective edition to date in the ArabicNLP series.

The accepted papers are presented over the two days of the conference in five oral sessions and two poster sessions: 20 papers as oral presentations and 17 as posters. The oral sessions are organized around the themes that emerged most strongly from the accepted set: Arabic speech, resources and benchmarks; dialects, machine translation and content moderation; Arabic for education and Islamic knowledge; Arabic resources, morphology and syntax; and LLM evaluation, safety and detoxification. Alongside the main conference papers, ArabicNLP 2026 hosts fifteen shared tasks with 212 papers (15 overview papers and 197 system description papers), which appear in Volume 2 of these proceedings.

ArabicNLP 2026 includes an invited talk by **Mona Diab** (Carnegie Mellon University), entitled "**From NLP for Arabic to NLP Through Arabic: What Must AI Understand Before We Trust It to Act?**", a fireside chat, and two panel discussions. We also recognize the strongest contributions through a Best Paper Award and a Best Resource Award, and the review effort behind them through Outstanding Reviewer and Outstanding Area Chair awards, all announced during the conference.

We were able to secure sponsorship funding from different institutions: Google Research, QatarDebate Center (Qatar), Aram Lab (MBZUAI), CAMeL Lab (NYU Abu Dhabi), Tuwaiq Academy (Saudi Arabia), and Resemble AI. We used the sponsorship funds to support student registrations. We thank all our sponsors for their generous support and their help in building up the Arabic NLP community. Finally, we extend our gratitude to everyone who submitted a paper to the conference, and to the Area Chairs and Program Committee members for their diligent efforts in providing reviews within a very tight time frame.

Muhammad Abdul-Mageed

General Chair, on behalf of the conference organizers.

Website of the conference: https://arabicnlp2026.sigarab.org

Organizing Committee

General Chair

Muhammad Abdul-Mageed, The University of British Columbia, Canada

Program Chairs

Tamer Elsayed, Qatar University, Qatar
Houda Bouamor, CMU-Q, Qatar
Samhaa El-Beltagy, Nawy, Egypt
Hend Al-Khalifa, King Saud University, Saudi Arabia

Publication Chairs

Ahmed Abdelali, HUMAIN, Saudi Arabia
AbdelRahim Elmadany, The University of British Columbia, Canada
Abdellah El Mekki, The University of British Columbia, Canada

Growth and Initiatives Chair

Kareem Darwish, QCRI, Qatar

Shared Tasks Chairs

Sakhar Alkhereyf, HUMAIN, Saudi Arabia
Hamzah Luqman, King Fahd University of Petroleum & Minerals, Saudi Arabia

Publicity and Social Chairs

Salam Khalifa, Stony Brook University, USA
Samar Magdy, The University of British Columbia, Canada

Scholarships and Awards Chairs

Zaid Alyafeai, King Abdullah University of Science and Technology, Saudi Arabia
Injy Hamed, MBZUAI, UAE

Ethics Chair

Wajdi Zaghouani, Northwestern University, Qatar

Sponsorship Chair

Imed Zitouni, Meta

Program Committee

Area Chairs

Ahmed Abdelali, Humain
Haithem Afli, Munster Technological University
Firoj Alam, Hamad Bin Khalifa University
Fethi Bougares, Elyadata
Karim Bouzoubaa, Ecole Mohammadia d'Ingénieurs
Violetta Cavalli-Sforza, Al Akhawayn University
Nadir Durrani, Hamad Bin Khalifa University
Mo El-Haj, VinUniversity
Shady Elbassuoni, American University of Beirut
Mustafa Jarrar, Hamad Bin Khalifa University
Mohammed Khalilia, Birzeit University
Mucahid Kutlu, Qatar University
Walid Magdy, University of Edinburgh
Hamdy Mubarak, Hamad Bin Khalifa University
El Moatez Billah Nagoudi, Prince Sultan University
Paul Rayson, Lancaster University
Nadi Tomeh, Université Sorbonne Paris Nord
Fadi Zaraket, Arab Center for Research and Policy Studies

Reviewers

Abdurahman Khalifa AAlAbdulsalam, Sultan Qaboos University
Mostafa Abdalla, New York University
Dana Abdulrahim, University of Bahrain
Chadi Abou Abou Chakra, Arab Center for Research and Policy Studies
Raia Abu Ahmad, German Research Center for AI
Mouath Abu Abu Daoud, New York University
Ibrahim Abu Farha, Alsun AI
Mustafa Ali Abuzaraida, Libyan Authority for Scientific Research
Mohamed Adel, New York University
Mohamed Nizzad Rameez Mohamed Ahamed Rameez, British College of Applied Studies
Hossam Ahmed, Leiden University
Rania Azad M. San Ahmed, Sulaimani Polytechnic University
Sharif Ahmed, University of Central Arkansas
Khadija Ait Ait ElFqih, Cadi Ayyad University
Youssef Al Hariri, University of Edinburgh
Khloud Al Jallad, HIAST
Osama Rakan Al Mraikhat, Arab Center for Research and Policy Studies
Sharefah Al-Ghamdi, King Saud University
Almoataz B. Al-Said, Cairo University
Maged S. Al-shaibani, SDAIA-KFUPM Joint Research Center for Artificial Intelligence
Jalal Al-Tamimi, Université Paris Cité
Abdulmohsen Al-Thubaity, HUMAIN
Ali Al-Zawqari, Vrije Universiteit Brussel
Salam Albatarni, Qatar University
Hanan Aldarmaki, Mohamed bin Zayed University of Artificial Intelligence

Noof Alfear, King Saud University
Bashar Alhafni, Mohamed bin Zayed University of Artificial Intelligence
Abdulaziz Alhamadani, Florida Polytechnic University
Abdulsalam Obaid Alharbi, University of New South Wales
Hassan Alhuzali, Umm Al-Qura University
Alaa Aljabari, Birzeit University
Rabab Alkhalifa, Imam Abdulrahman Bin Faisal University
Sakhar Alkhereyf, HUMAIN
Lulwah AlKulaib, Kuwait University
Ali Allaith, Copenhagen University
Abdulrahman Alosaimy, Al-Imam Mohamed Ibn Saud Islamic University
Abdulkareem Alsudais, Prince Sattam bin Abdulaziz University
Malik H. Altakrori, IBM TJ Watson Research Center
Shatha Altammami, King Saud University
Duygu Altinok, Independent Researcher
Nora Alturayeif, Humain
Zaid Alyafeai, King Abdullah University of Science and Technology
Ahmed Alzubaidi, TII
ZIANI Amel, Chadli Benjedid University
Wissam Antoun, INRIA
Mohammed Attia, Google
Joseph Attieh, University of Helsinki
May Bashendy, Qatar University
Slimane Bellaouar, Université de Ghardaia
Ismail Berrada, Mohammed VI Polytechnic University
Mabrouka Bessghaier, Northwestern University
Ann Bies, Linguistic Data Consortium, University of Pennsylvania
Md. Rafiul Biswas, Hamad Bin Khalifa University
Zahra Bokaei, University of Edinburgh
Claudia Borg, University of Malta
Abdesselam Bouchekif, Hamad Bin Khalifa University
Ibrahim Bounhas, Université de la Manouba
Basma El Amel Boussaha, Technology and Innovation Institute
Hadda Cherroun, Université Amar Telidji
Shammur Absar Chowdhury, Hamad Bin Khalifa University
Kareem Mohamed Darwish, Hamad Bin Khalifa University
Pavel Denisov, Fraunhofer IAIS
Mai Mohamed Eida, University of Illinois at Urbana-Champaign
Amina El Ganadi, University of Modena and Reggio Emilia
Yassine El Kheir, German Research Center for AI
Abdellah El Mekki, University of British Columbia
Mohammed Elkholy, Mohamed bin Zayed University of Artificial Intelligence
Khalid N. Elmadani, Mohamed bin Zayed University of Artificial Intelligence
AbdelRahim A. Elmadany, University of British Columbia
Ashraf Elnagar, Google
Hossam Elsafty, Fraunhofer Institute IAIS, Fraunhofer IAIS
Ahmed Elshabrawy, Mohamed bin Zayed University of Artificial Intelligence
Sohaila Eltanbouly, Qatar University
Fadhl Eryani, Eberhard-Karls-Universität Tübingen
Amany Fashwan, Bibliotheca Alexandrina
Abed Alhakim Freihat, Mohamed bin Zayed University of Artificial Intelligence

Kamel Gaanoun, Institut National de Statistiques et d'Economie Appliquées
Pagon Gatchalee, Chiang Mai University
Sahar Ghannay, Universuté paris saclay
Nada Ghneim, Arab International University
Omer Goldman, University of Cambridge
Nizar Habash, New York University Abu Dhabi
Nagham hamad, Birzeit University
Nancy Hamdan, Arab Center for Research and Policy Studies
Injy Hamed, Mohamed bin Zayed University of Artificial Intelligence
Zineb Ferhat Hamida, Université de Sétif
Hadi Hamoud, Arab Center for Research and Policy Studies
Fatima Haouari, University of Sheffield
Salima Harrat, Ecole Normale Supérieure Bouzaréah, Alger
Kedir Yassin Hussen, Bahir Dar University
Go Inoue, Mohamed bin Zayed University of Artificial Intelligence
Elsayed Issa, Purdue University
Sanaa Kaddoura, Zayed University
Ahmad Kashmar, Arab Center for Research and Policy Studies
Amr Keleg, Mohamed bin Zayed University of Artificial Intelligence
Ayman Khalafallah, Alexandria University
Salam Khalifa, New York University
Khalid Al Khatib, University of Groningen
Md Tawkat Islam Khondaker, University of British Columbia
Jordan Kodner, State University of New York, Stony Brook
David Langlois, Université de Lorraine
Mohamed Lichouri, Université des Sciences et de la Technologie Houari Boumediène
Hamzah Luqman, King Fahad University of Petroleum and Minerals
Rana Malhas, Qatar University
Watheq Mansour, The University of Queensland
Reham Marzouk, Mohamed bin Zayed University of Artificial Intelligence
Salima Mdhaffar, Université d'Avignon
Cheragui Mohamed Mohamed Amine, Ahmed Draia University
Ghassan Mourad, Lebanese University
Amel Muminovic, International Balkan University
Ahmad M Mustafa, Jordan University of Science and Technology
Sara Nabhani, University of Groningen
Omer Nacar, Tuwaiq Academy
Alexis Nasr, Aix Marseille University
Iben Iben Nasr Latifa, University of Sfax
Mayar Nassar, Ain Shams University
Shah Nawaz, Johannes Kepler Universität Linz
Usman Nawaz, University of Palermo, Italy
Attia Nehar, Ziane Achour University
Ahamed Rameez Mohamed Nizzad, Zayed University
Arwa Omayrah, HUMAIN
Ömer Tarik Özyilmaz, University Medical Center Groningen
David M. Palfreyman, United Arab Emirates University
Abdelrahim Qaddoumi, State University of New York at Stony Brook
Chatrine Qwaider, Mohamed bin Zayed University of Artificial Intelligence
Nour Rabih, Mohamed bin Zayed University of Artificial Intelligence
Owen Rambow, Stony Brook University

Mahmoud Reda, New York University
Caroline Sabty, German International University
Waseem SAFI, Damascus University
Hassan Sajjad, Dalhousie University
Marwan Sayed, Qatar University
Muhammad Shakeel, Honda Research Institution Japan Co., Ltd.
Nada Sharaf, The German International University
Serry Sibaee, Prince Sultan University
Peter Sullivan, University of British Columbia
Reem Suwaileh, Civil Service and Government Development Bureau, Qatar
Minh Ngoc Ta, Mohamed bin Zayed University of Artificial Intelligence
Marwan Torki, Alexandria University
Samia Touileb, University of Bergen
Ahmed Wasfy, KAND CA Corp.
Moustafa Wassel, Alexandria University
Suveyda Yeniterzi, GenAIus Technologies
Seid Muhie Yimam, Universität Hamburg
Wajdi Zaghouani, Northwestern University
Bielasan Zaina, Georgetown University
Hoda Zaiton, Pharos University in Alexandria
Petr Zemánek, Charles University Prague
Taha Zerrouki, Bouira University

Keynote Talk

From NLP for Arabic to NLP Through Arabic What Must AI Understand Before We Trust It to Act?

Dr. Mona Diab
Carnegie Mellon University

Abstract: What does it mean for an AI system to understand a request, recognize an intention, or know when it has misunderstood? These questions reach into the essence of intelligence—and acquire urgency when systems act as autonomous agents. An instruction can be followed literally while its purpose is missed. A fluent answer can conceal uncertainty. Understanding a harmful request perfectly can itself enable harm. This keynote argues that the science of language, meaning, understanding, and interaction should fundamentally reorient NLP's research agenda and inform how we assess AI's capabilities and risks. Arabic offers a particularly productive setting for developing that science. Rich morphology, unwritten vowels, dialect diversity, diglossia, script variation, and code-switching converge within Arabic communicative encounters shaped by social relationships and historical knowledge. I introduce the Arabic Conjunction Hypothesis to investigate how these dimensions jointly shape interpretation and communicative action. Arabic's connections with Persian, Urdu, and Turkic traditions extend the inquiry across languages, scripts, and communities. Arabic's scientific value lies in this productive conjunction, without requiring a claim of exceptionalism. The resulting questions connect directly to debates about alignment and control: whose intentions should a system follow, what evidence supports its interpretation, and when should it ask, defer, challenge, or revise? Understanding is necessary for many forms of trustworthy assistance, but it does not establish benevolent goals or legitimate authority. A science of meaning and interaction can help make those distinctions explicit and testable. Through this lens, annotation, evaluation, and dialogue become instruments for discovering how understanding works. The shift from NLP for Arabic to NLP through Arabic makes Arabic a source of insights that can reshape the field's questions, methods, and foundations. What would we build—and what would we measure—if understanding in interaction became NLP's organizing purpose? Pursuing that question also challenges how we define and evaluate intelligence in AI more broadly. If we fear what AI might do, we must examine what it understands, whose purposes it serves, and what warrants trusting it to act.

Bio: Mona T. Diab is Director of the Language Technologies Institute and a Full Professor in the School of Computer Science at Carnegie Mellon University. An ACL Fellow, she has devoted more than 13 years of her career to Arabic NLP and received the 2023 King Salman Global Academy for Arabic Language prize for advancing Arabic language technologies. She is a founding member of SIGARAB, a former President of SIGDAT and SIGLEX, and a cofounder of the *SEM conference and the CADIM Arabic Dialect Group. Her career spans academia and industry, including leadership roles at Meta and Amazon AWS. Her research connects multilingual NLP, cultural and social dimensions of meaning, and responsible AI. She advocates for a science of language, meaning, and interaction, with Arabic helping shape foundational questions about intelligence, understanding, and trustworthy AI.

Table of Contents

Same Legal Facts, Different Verdicts: Evaluating Cross-Lingual and Dialectal Robustness in LLM Legal Reasoning
Dana R Alsagheer, Mohammad Kamal, Abdulrahman Kamal, Yang Lu, Nour Diallo, Weidong Shi and Omar Kamal 1

Fine-tuning Small Language Models for Cross-Prompt Arabic Essay Scoring: A Benchmarking Study
Salam Albatarni, Fahim Dalvi, Sohaila Eltanbouly and Tamer Elsayed 11

Align-Extract-Score: Graded Relevance of Arabic Essays
Marwan Sayed, Salam Albatarni, Reem Suwaileh and Tamer Elsayed 22

AraDetox: A Multi-Dialect Arabic Detoxification Dataset
Mo El-Haj 33

Some Dialects are More Equal than Others: Non-Prestigious Arabic Dialectal Bias in LLMs
Mai Mohamed Eida, Ryan Dolan, Paul de Nijs and Jonathan Dunn 48

DAASI: Arabic Text Simplification Corpus for Israeli Government Documents
Natalia Vanetik, Maysa Mosa Abu Shareb and Hussein Abo Ismael 65

Beyond Gender, Number, and Rationality: Modeling Extended Arabic Morpho-Syntactic Agreement with Aggregation
Bielasan Zaina, Salam Khalifa and Nizar Habash 77

SHAMS: An Audio-Grounded Pronunciation Benchmark for Levantine Arabic
Ben Sapirstein, Roy Mattar, Guy Mor-Lan, Ahlam Mohamed, Letizia Cerqueglini and Morris Alper 88

BAREC-TB: A Dependency Treebank and Parsing Benchmark for Arabic Children's Texts
Khalid N. Elmadani, Mohamed Adel, Muhammed AbuOdeh, Reem Faraj, Jamila El-Gizuli, Hanada Taha and Nizar Habash 115

HalluTruthQA: A Fine-Grained Benchmark for Hallucination Detection, Localization, and Explanation in Arabic Question Answering
Abdessalam Bouchekif, Mohammed-En-Nadhir Zighem, Salah Eddine Bekhouche, Hichem Telli, Somaya Eltanbouly, Shahd Gaben, Heba Sbahi, Samer Rashwani, Mutaz al-Khatib, Emad Soliman Ali Mohamed, Mohammed Ghaly and Abdenour Hadid 130

MAQAM: A Richly Annotated Corpus for Quranic Arabic Morphological Analysis and Disambiguation
Mostafa Saeed, Mohammad Amjad, Mahmoud Kassem, Rawan Bondok, Mayar Nassar, Adel Mahmoud Wizani, Salman Elgamal, Mahmoud Reda and Nizar Habash 151

Using LALMs for Pronunciation Error Detection in Arabic
Osama E. Youssef, Ahmed Ali and Walid Magdy 164

Optimizing Inference Cost for Arabic–English Mathematical Reasoning in LLMs
Sahar Melhem and Ahmad M Mustafa 170

Shaer: Controlled Arabic Poetry Generation with Meter Subform and Semantic Conditioning
Ahmad Abbas, Tamara Fakih, Nour Fakih and Ammar Mohanna 179

Cultural Understanding in Large Language Models: A Comparative Study with Human Judgments
Elsayed Issa and Kamilia Rahmouni 200

Gahwa: A Computational Lexicon for Emirati Arabic
Saeed Alzaabi, Salam Khalifa, Mostafa Saeed, Shahd Dibas, Hanan Aldarmaki and Nizar Habash 219

Arabic Safety Alignment as Selective Refusal: An Empirical Study of SFT, DPO, and Guard Calibration
Mohammad Zbeeb and Ammar Mohanna 228

What Makes a Good Fiqh Retriever? Answer Retrieval for Arabic Islamic Jurisprudence
Somaya Eltanbouly, Heba Sbahi, Samer Rashwani, Abdessalam Bouchekif, Mutaz al-Khatib, Shahd Gaben and Mohammed Ghaly 247

From Subwords to Pixels: Evaluating Text Representations for Arabic Machine Translation
Abdelrahman Shaheen, Mohamed Abbas, Tereza Truckova, Yova Kementchedjhieva and Bashar Alhafni 263

SAFIR: A Sub-Dialectal Benchmark for Saudi Arabic–English Machine Translation
Nour Qandos, Omer Nacar, Aya Hamod, Samar Ahmad, Rahaf Saeed Al Hallay and Ahmad Al Rabghi 289

QuranicMMLU: A Cognitively-Aware Benchmark for Evaluating Generative AI Solutions on Quranic Linguistic Knowledge
Rawan El Ghali, Umm Kulsoom, Anas Madkoor, Roaa Abdelmagid, Abdullah Khanafer, Raghad Mousa, Roaa Ibrahim, Hamza Aljaji, Abdallah Alkanani and Ehsaneddin Asgari 302

Comparative Study of Classical NLP and Pretrained Language Models for Tunisian Dialect Processing
Latifa Iben Nasr, Ismail Badache and Lamia Belguith 317

AraMIP: Extending MIPVU Towards Metaphor Identification in Arabic
Mandar Marathe, Manar Ali, Sara Nabhani, Raia Abu Ahmad, Ibrahim Baroud and Omar Momen 325

AraScholar: Calibrating Academic Paraphrasing in Modern Standard Arabic
Omer Nacar, Asma Al Wazrah, Rawan Nasser Almatham, Afrah Altamimi and Abdulrahman Alosaimy 340

Arabic-Centric Preference Optimization for Efficient Dialect-Sensitive Machine Translation: A Seq2Seq Approach
Abdelrahman Hisham, Malak Fathallah and Ayman Khalafallah 356

What WER Hides: A Closer Look at Algerian Code-Switched ASR
Othman Istaiteh, Yassine Toughrai, Kamel Smaili, Salima Mdhaffar, David Langlois and Yannick Estève 373

AHA-Memes: A Fine-Grained Multimodal Benchmark for Hate Detection in Arabic Memes
Mohamed Bayan Kmainasi, Ali Ezzat Shahroor, Abul Hasnat, Md. Rafiul Biswas, Wajdi Zaghouani and Firoj Alam 384

Towards Speech ALDi: Predicting the Arabic Level of Dialectness Directly from Speech
Wageeh Khadawardi, Amr Keleg and Walid Magdy 409

Definition-Guided Learning and Root-Stratified Evaluation for Arabic Lexical Semantics
Samar Ahmad, Abdullah I. Alharbi, Omer Nacar, Shahad Altamimi and Yahya Assiri 418

Real-Time Quranic Recitation Assessment on Mobile Devices Using Streaming Inference and Model Compression
Abdelrahman Ayman Amin Mohamed and Abdullah Haytham Mohamed Salah Hedeya 433

KASHAF: A Hierarchical Resource and Pipeline for Arabic Interpretable Semantic Textual Similarity
Fahad AlGhamdi, Abdelati Hawwari, Abdulrahman Alosaimy, Rawan Nasser Almatham and Asma Al Wazrah 444

Quanta: A Fine-Grained Dataset and Benchmark for Arabic Machine-Translation Error Analysis
Afraa Alshammari, Ahmad Alahmari, Amal Alaboud, Jaber Ahmed Harthi, Karima Ali Shahwan Alzahrani, Ahnaf Adib, Patrick Brandt, Javier Osorio, Vito D'Orazio and Latifur Khan 462

Beyond Monolithic Libyan Arabic: A Sub-Regional Speech Corpus and Dialect Identification Benchmark
Nour A. Zohdy and Mansour Essgaer 485

SALM: Social Arabic Language Model
Saad Mankarious and Aya Zirikly 497

Knowledge Graphs and Distributional Readability of Arabic Documents
Mohamad Chhaytle, Hadi Hamoud, Sally Choker, Chadi Abou Chakra, Mohamad Ballout and Fadi Zaraket 508

Why LLM Aggregation Falls Short: Correlated Errors and Selection Gaps in Arabic Punctuation Restoration
Rana Alshaikh and Majedah Alrehiely 522

Child ASR Adaptation with Adult Retention: An Empirical Study
Houssam Eddine-Othman Lachemat and Shammur Absar Chowdhury 538

Program

Wednesday, October 28, 2026

08:50 - 09:30 *Opening, Welcome & Sponsor Acknowledgements*

09:30 - 10:30 *Keynote: From NLP for Arabic to NLP Through Arabic: What Must AI Understand Before We Trust It to Act? by Dr. Mona Diab*

10:30 - 11:00 *Coffee Break*

11:00 - 12:25 *Arabic Speech, Resources and Benchmarks*

Child ASR Adaptation with Adult Retention: An Empirical Study
Houssam Eddine-Othman Lachemat and Shammur Absar Chowdhury

Towards Speech ALDi: Predicting the Arabic Level of Dialectness Directly from Speech
Wageeh Khadawardi, Amr Keleg and Walid Magdy

MAQAM: A Richly Annotated Corpus for Quranic Arabic Morphological Analysis and Disambiguation
Mostafa Saeed, Mohammad Amjad, Mahmoud Kassem, Rawan Bondok, Mayar Nassar, Adel Mahmoud Wizani, Salman Elgamal, Mahmoud Reda and Nizar Habash

KASHAF: A Hierarchical Resource and Pipeline for Arabic Interpretable Semantic Textual Similarity
Fahad AlGhamdi, Abdelati Hawwari, Abdulrahman Alosaimy, Rawan Nasser Almatham and Asma Al Wazrah

12:30 - 14:00 *Lunch Break*

14:00 - 15:08 *Dialects, Machine Translation and Content Moderation*

Some Dialects are More Equal than Others: Non-Prestigious Arabic Dialectal Bias in LLMs
Mai Mohamed Eida, Ryan Dolan, Paul de Nijs and Jonathan Dunn

SAFIR: A Sub-Dialectal Benchmark for Saudi Arabic–English Machine Translation
Nour Qandos, Omer Nacar, Aya Hamod, Samar Ahmad, Rahaf Saeed Al Hallay and Ahmad Al Rabghi

From Subwords to Pixels: Evaluating Text Representations for Arabic Machine Translation
Abdelrahman Shaheen, Mohamed Abbas, Tereza Truckova, Yova Kementchedjhieva and Bashar Alhafni

Wednesday, October 28, 2026 (continued)

AHA-Memes: A Fine-Grained Multimodal Benchmark for Hate Detection in Arabic Memes
Mohamed Bayan Kmainasi, Ali Ezzat Shahroor, Abul Hasnat, Md. Rafiul Biswas, Wajdi Zaghouani and Firoj Alam

15:11 - 15:30 *Shared Task Highlights I: 7 shared tasks, 2 minutes each*

15:30 - 16:30 *Coffee Break & Poster Session 1 (main conference and shared task posters)*

16:30 - 17:20 *Panel 1: Inclusive Arabic NLP*

17:20 - 17:30 *Awards*

Thursday, October 29, 2026

08:50 - 09:00 *Opening*

09:00 - 09:30 *Fireside Chat*

09:30 - 10:21 *Arabic for Education and Islamic Knowledge*

Align-Extract-Score: Graded Relevance of Arabic Essays
Marwan Sayed, Salam Albatarni, Reem Suwaileh and Tamer Elsayed

Fine-tuning Small Language Models for Cross-Prompt Arabic Essay Scoring: A Benchmarking Study
Salam Albatarni, Fahim Dalvi, Sohaila Eltanbouly and Tamer Elsayed

What Makes a Good Fiqh Retriever? Answer Retrieval for Arabic Islamic Jurisprudence
Somaya Eltanbouly, Heba Sbahi, Samer Rashwani, Abdessalam Bouchekif, Mutaz al-Khatib, Shahd Gaben and Mohammed Ghaly

10:30 - 11:00 *Coffee Break*

11:00 - 12:25 *Arabic Resources, Morphology and Syntax*

Beyond Gender, Number, and Rationality: Modeling Extended Arabic Morpho-Syntactic Agreement with Aggregation
Bielasan Zaina, Salam Khalifa and Nizar Habash

BAREC-TB: A Dependency Treebank and Parsing Benchmark for Arabic Children's Texts
Khalid N. Elmadani, Mohamed Adel, Muhammed AbuOdeh, Reem Faraj, Jamila El-Gizuli, Hanada Taha and Nizar Habash

Why LLM Aggregation Falls Short: Correlated Errors and Selection Gaps in Arabic Punctuation Restoration
Rana Alshaikh and Majedah Alrehiely

SALM: Social Arabic Language Model
Saad Mankarious and Aya Zirikly

Beyond Monolithic Libyan Arabic: A Sub-Regional Speech Corpus and Dialect Identification Benchmark
Nour A. Zohdy and Mansour Essgaer

Thursday, October 29, 2026 (continued)

12:30 - 14:00 *Lunch Break*

14:00 - 15:08 *LLM Evaluation, Safety and Detoxification*

HalluTruthQA: A Fine-Grained Benchmark for Hallucination Detection, Localization, and Explanation in Arabic Question Answering
Abdessalam Bouchekif, Mohammed-En-Nadhir Zighem, Salah Eddine Bekhouche, Hichem Telli, Somaya Eltanbouly, Shahd Gaben, Heba Sbahi, Samer Rashwani, Mutaz al-Khatib, Emad Soliman Ali Mohamed, Mohammed Ghaly and Abdenour Hadid

Arabic Safety Alignment as Selective Refusal: An Empirical Study of SFT, DPO, and Guard Calibration
Mohammad Zbeeb and Ammar Mohanna

Definition-Guided Learning and Root-Stratified Evaluation for Arabic Lexical Semantics
Samar Ahmad, Abdullah I. Alharbi, Omer Nacar, Shahad Altamimi and Yahya Assiri

AraDetox: A Multi-Dialect Arabic Detoxification Dataset
Mo El-Haj

15:11 - 15:30 *Shared Task Highlights II: 8 shared tasks, 2 minutes each*

15:30 - 16:30 *Coffee Break & Poster Session 2 (main conference and shared task posters)*

16:30 - 17:20 *Panel 2: ArabicNLP Growth*

17:20 - 17:30 *Closing Remarks*

Same Legal Facts, Different Verdicts: Evaluating Cross-Lingual and Dialectal Robustness in LLM Legal Reasoning

Dana R. Alsagheer[1], Mohammad Kamal[2], Abdulrahman Kamal[1]
Omar Kamal[1], Yang Lu[1], Nour Diallo[1], Weidong Shi[1]
[1]University of Houston, Houston, Texas, USA
[2]Cornell University, Ithaca, New York, USA

Abstract

Large language models (LLMs) are increasingly evaluated on legal reasoning tasks, but most benchmarks remain English-centered and test each legal problem in only one linguistic form. We examine whether LLMs preserve final-answer accuracy when the same legal question is expressed in English, Modern Standard Arabic, Simple Arabic, and Egyptian Arabic. We construct a controlled benchmark of 200 unique MBE-style multiple-choice problems represented in four semantically aligned versions, yielding 800 linguistic instances. The legal facts, answer choices, legal issue, and correct answer are preserved across versions through human translation and legal and linguistic validation. We evaluate GPT-5, Gemini 2.5 Pro, Grok, and Qwen using standardized language-matched prompts and a shared, human-verified answer key. All models achieve their highest accuracy in English, but their Arabic performance varies substantially. Qwen scores 97.0% in English and 79.0% in each Arabic condition, while GPT-5 exhibits the greatest cross-lingual stability, scoring 90.0% in English and 80.0–82.0% across the Arabic conditions. Gemini 2.5 Pro declines from 80.5% in English to 17.5–21.0% in Arabic, while Grok declines from 45.0% to 16.5–26.5%. These results demonstrate substantial cross-lingual final-answer gaps within this translated MBE-style setting. They do not establish robustness across legal systems or equivalence in the models' underlying reasoning processes. Our findings show that English-language accuracy alone is insufficient to assess model reliability in Arabic legal question answering.

1 Introduction

Large language models (LLMs) are increasingly evaluated on legal reasoning tasks, including bar-exam-style questions, statutory interpretation, rule application, and legal judgment prediction. Prior studies have reported strong performance on English legal benchmarks, including LegalBench and simulated bar examinations (Guha et al., 2023; Katz et al., 2024; Alsagheer et al., 2025). However, most legal evaluations remain English-centered and present each legal problem in only one linguistic form. Consequently, high English accuracy does not establish that a model will preserve its performance when the same problem is expressed in another language, register, or dialect.

This limitation is especially important for Arabic-speaking users. Arabic is not a single uniform linguistic condition: users may interact with LLMs in Modern Standard Arabic (MSA), simplified registers, regional dialects, or mixed forms. Multilingual benchmarks have shown that strong English performance does not necessarily imply comparable cross-lingual generalization (Hu et al., 2020; Liang et al., 2020; Bandarkar et al., 2024). Arabic benchmarks such as LAraBench and ArabicMMLU further demonstrate the importance of language- and variety-sensitive evaluation (Abdelali et al., 2024; Koto et al., 2024). In the legal domain, ALARB evaluates Arabic legal argument reasoning (Abu Shairah et al., 2025), but existing work does not directly test whether LLMs preserve their answers when the same legal problem is presented in semantically aligned English and Arabic forms.

Legal multiple-choice questions provide a demanding setting for this evaluation because the correct answer may depend on precise interactions among facts, rules, exceptions, procedural posture, and closely related answer choices. Even a small semantic change can legitimately alter the answer. An uncontrolled comparison may therefore conflate model sensitivity with translation differences. To address this problem, we construct a paired benchmark in which every item preserves the legal facts, legal issue, answer choices, answer order, and gold answer across four linguistic conditions: English, Original Arabic/Modern Standard Arabic (OA/MSA), Simple Arabic (SA), and Egyptian

Proceedings of The Fourth Arabic Natural Language Processing Conference (ArabicNLP 2026), pages 1–10
October 28-29, 2026 ©2026 Association for Computational Linguistics

Arabic (EA).

The benchmark contains 200 unique MBE-style legal problems, each represented in four aligned versions, yielding 800 linguistic instances: 200 English and 600 Arabic instances. These are not 800 independent legal problems. Reusing the same problems enables controlled within-item comparisons in which the legal content remains fixed while the linguistic form changes.

The Arabic versions were translated by humans and underwent legal and linguistic validation. Two bilingual lawyers reviewed every version for the preservation of the facts, the applicable legal issue, the rule's application, the answer-choice meanings, and the correct answer. A Ph.D.-level Arabic-language reviewer separately assessed fluency, clarity, register appropriateness, and dialectal naturalness. Items identified as ambiguous or misaligned were revised and re-reviewed before evaluation.

Simple Arabic was included as a readability-oriented condition to examine whether reducing linguistic complexity improves model performance without changing the legal task. Egyptian Arabic was selected because it is widely understood and because the research team had the expertise needed to validate it rigorously. We do not assume that the observed patterns generalize to Gulf, Levantine, Maghrebi, or other Arabic varieties; evaluating additional dialects and mixed MSA–dialect forms remains important future work.

We frame the measured property as *cross-lingual robustness* because the legal facts, issue, answer choices, and gold answer are held constant while only the linguistic form changes. Under this controlled transformation, a robust model should preserve its final-answer performance. The resulting disparities also raise fairness and access concerns for Arabic-speaking users, but fairness itself is not directly measured in this study. Moreover, exact-match accuracy evaluates final-answer correctness rather than the faithfulness, legal validity, or cross-language equivalence of the models' underlying reasoning processes.

We investigate the following research questions:

RQ1: How accurately do LLMs answer the same legal questions in English, OA/MSA, Simple Arabic, and Egyptian Arabic?

RQ2: How large are the English–Arabic performance gaps when the legal content and answer key are held constant?

RQ3: Which evaluated models are most robust to changes in Arabic register and dialect?

RQ4: Does simplifying Arabic improve final-answer accuracy, or can it introduce additional instability?

To answer these questions, we evaluate GPT-5 (OpenAI, 2025), Gemini 2.5 Pro (Google DeepMind, 2025), Grok 3 (xAI, 2025), and Qwen 2 (Yang et al., 2024) using standardized, language-matched prompts and a shared, human-verified answer key. We report language-specific accuracy, average Arabic accuracy, English–Arabic performance gaps, maximum cross-language variation, and Wilson 95% confidence intervals. All four models achieve their highest accuracy in English, but their Arabic performance differs substantially. Qwen achieves the highest English accuracy at 97.0% but scores 79.0% in each Arabic condition. GPT-5 exhibits the greatest cross-lingual stability, scoring 90.0% in English and 80.0–82.0% across Arabic conditions. Gemini 2.5 Pro declines from 80.5% in English to 17.5–21.0% across Arabic conditions, while Grok declines from 45.0% to 16.5–26.5%.

This work makes three principal contributions:

1. We introduce a controlled paired evaluation design containing 200 unique MBE-style legal problems and 800 semantically aligned English and Arabic instances.

2. We evaluate four widely used LLM interfaces using language-specific accuracy and complementary measures of cross-lingual final-answer robustness.

3. We provide empirical evidence that English accuracy alone is insufficient for assessing model reliability on semantically aligned Arabic legal questions.

Our conclusions concern cross-lingual final-answer robustness on translated MBE-style multiple-choice questions. The study does not compare different legal systems, evaluate Arabic legal doctrine, assess open-ended legal advice or professional judgment, or determine whether models use equivalent reasoning processes across languages. The findings should therefore be interpreted within the evaluated task, source domain, linguistic conditions, prompts, and model interfaces.

2 Related Work

Legal reasoning benchmarks for LLMs. Recent work has developed benchmarks for evaluating the legal capabilities of large language models (LLMs). LegalBench provides a collaboratively constructed collection of legal reasoning tasks designed by legal professionals and evaluated across multiple LLMs (Guha et al., 2023). Other studies assess LLM performance on professional legal examinations, including the performance of GPT-4 on the Uniform Bar Examination (Katz et al., 2024). These studies demonstrate substantial progress on English-language legal tasks, but they primarily measure legal competence in English. They do not test whether a model preserves its answer when the same legal problem is presented across English and multiple Arabic language varieties.

Multilingual and Arabic LLM evaluation. Multilingual benchmarks such as XTREME, XGLUE, and Belebele demonstrate that strong English-language performance does not necessarily imply comparable cross-lingual generalization (Hu et al., 2020; Liang et al., 2020; Bandarkar et al., 2024). Arabic evaluation introduces additional challenges because Arabic is used in Modern Standard Arabic, regional dialects, code-switched forms, and registers that vary in linguistic complexity. Arabic benchmarks such as LAraBench and ArabicMMLU evaluate LLM performance across a broad range of Arabic-language tasks and domains (Abdelali et al., 2024; Koto et al., 2024). However, these benchmarks are general-purpose evaluations and are not designed as controlled, within-item tests in which the legal facts, answer choices, and correct answer are preserved across English and multiple Arabic versions.

Arabic-centric language models. In parallel with benchmark development, recent technical reports have introduced language models specifically designed to strengthen Arabic capabilities. ALLaM is a family of Arabic–English language models that uses bilingual pretraining, vocabulary expansion, parallel data, and preference alignment to improve Arabic performance while retaining English capabilities (Bari et al., 2024). Fanar is an Arabic-centric multimodal generative AI platform whose language-model components are trained on large-scale Arabic, English, and code corpora (Fanar Team, 2025). ALLaM and Fanar are models rather than evaluation benchmarks and therefore represent complementary work to Arabic benchmark development. Our experiments evaluate four general-purpose model interfaces rather than Arabic-centric models. Evaluating ALLaM, Fanar, and other Arabic-focused or fixed open-weight models under the same controlled legal protocol is an important direction for future work. Such comparisons could help determine whether Arabic-focused pretraining and alignment reduce the cross-lingual final-answer gaps observed in the present study.

Work	Leg.	Ar.	Align.	Var.
LegalBench (Guha et al., 2023)	Y	N	N	N
Bar Exam (Katz et al., 2024)	Y	N	N	N
LAraBench (Abdelali et al., 2024)	N	Y	N	Y
ArabicMMLU (Koto et al., 2024)	N	Y	N	N
MultiLegalPile (Niklaus et al., 2024)	Y	Part.	N	N
ALARB (Abu Shairah et al., 2025)	Y	Y	N	N
Ours	**Y**	**Y**	**Y**	**Y**

Table 1: Comparison of related legal and Arabic benchmarks. Leg.: legal focus; Ar.: Arabic coverage; Align.: aligned questions; Var.: Arabic varieties.

Multilingual legal resources. Research has also expanded the availability of multilingual legal NLP resources. MultiLegalPile provides a large multilingual legal corpus spanning multiple languages and jurisdictions (Niklaus et al., 2024). Within Arabic legal NLP, ALARB introduces an Arabic legal argument-reasoning benchmark derived from Saudi commercial court cases and includes tasks involving verdict prediction, legal-reasoning completion, and regulation identification (Abu Shairah et al., 2025). These resources advance multilingual and Arabic legal AI, but they address objectives different from ours. MultiLegalPile primarily supports multilingual legal language modeling, while ALARB evaluates legal reasoning within an Arabic legal setting. Neither is designed to isolate the effect of linguistic form by presenting the same legal problem across aligned English, Modern Standard Arabic, Simple Arabic, and Egyptian Arabic versions.

Table 1 shows that prior resources do not jointly evaluate legal reasoning, Arabic variation, and aligned versions of the same problem. Our paired design holds the legal content and gold answer constant to measure cross-lingual final-answer robustness, not robustness across legal systems or reasoning-process equivalence.

3 Methodology

This section describes the construction of our controlled multilingual legal reasoning benchmark and

the evaluation procedure used to test whether LLMs preserve legal reasoning performance across English, formal Arabic, simplified Arabic, and Egyptian Arabic. Our goal is to isolate the effect of language, register, and dialect while keeping the underlying legal task fixed. Figure 1 summarizes the full evaluation pipeline.

3.1 Dataset Source and Benchmark Design

The English source questions were selected from protected National Conference of Bar Examiners (NCBE) Multistate Bar Examination (MBE) preparation materials. MBE-style questions are suitable for this study because each item contains a legal fact pattern, four answer choices, and one designated correct answer. Prior work has similarly used bar examinations and legal benchmarks to evaluate LLM legal reasoning (Katz et al., 2024; Guha et al., 2023), although such evaluations remain largely English-centered.

We selected 200 questions from a larger pool. We retained items with a complete fact pattern, four well-formed answer choices, and a clearly specified correct answer. We excluded items with incomplete or malformed content, ambiguous answers, English-specific wording, or legal distinctions that could not be reliably preserved in Arabic. The selected questions were approximately balanced across the principal MBE subject areas, with greater representation of Civil Procedure.

Each question was represented in English, Modern Standard Arabic (MSA), Simple Arabic, and Egyptian Arabic, producing 800 aligned instances: 200 English and 600 Arabic. These instances represent 200 unique legal problems, not 800 independent problems. Across the four versions, the legal facts, legal issue, answer choices, answer order, and gold answer were held constant through the legal and linguistic validation process described below. This paired design isolates changes associated with language, register, and dialect while keeping the underlying legal task fixed.

Because the source questions and official explanations are protected NCBE materials, we cannot reproduce, publicly redistribute, or release the complete benchmark, including the original English questions, their Arabic translations, answer choices, and official explanations. We instead report aggregate results, evaluation procedures, prompt templates, scoring rules, and non-reconstructive synthetic examples. Further details about data availability and reproducibility are provided in Section 6.

3.2 Language Conditions

Each legal problem appears in four aligned language conditions:

- **English (EN):** the original source version.
- **Modern Standard Arabic (MSA/OA):** a formal Arabic version preserving the structure and legal meaning of the English item.
- **Simple Arabic (SA):** a readability-oriented version using simpler wording and sentence structure while preserving the same legal task.
- **Egyptian Arabic (EA):** a dialectal version written in natural Egyptian Arabic while preserving the same legal content.

These conditions test whether model performance changes when the linguistic form varies while the underlying legal problem and correct answer remain fixed.

3.3 Arabic Translation and Validation

All Arabic versions were human-translated. Two bilingual lawyers reviewed every MSA, Simple Arabic, and Egyptian Arabic version to verify preservation of the legal facts, issue, applicable rule, answer-choice meanings, answer order, and gold answer. A Ph.D.-level Arabic-language reviewer separately evaluated each version for fluency, clarity, register appropriateness, and dialectal naturalness.

When a reviewer identified ambiguity, unnatural wording, mistranslation, or a possible change in legal meaning, the translation and review team discussed and revised the item. Every revised item was then re-reviewed for both legal and linguistic equivalence before inclusion in the benchmark.

3.4 Evaluated Models and Prompting

We evaluated four general-purpose LLMs: GPT-5, Gemini 2.5 Pro, Grok 3, and Qwen 2. These models were selected because they represent widely used commercial or publicly accessible LLM families with multilingual capabilities. They were accessed via the official user-facing interfaces for ChatGPT, Gemini, Grok, and Qwen Studio, respectively. We recorded the displayed model identifier and access date when available; otherwise, we recorded the interface's default model.

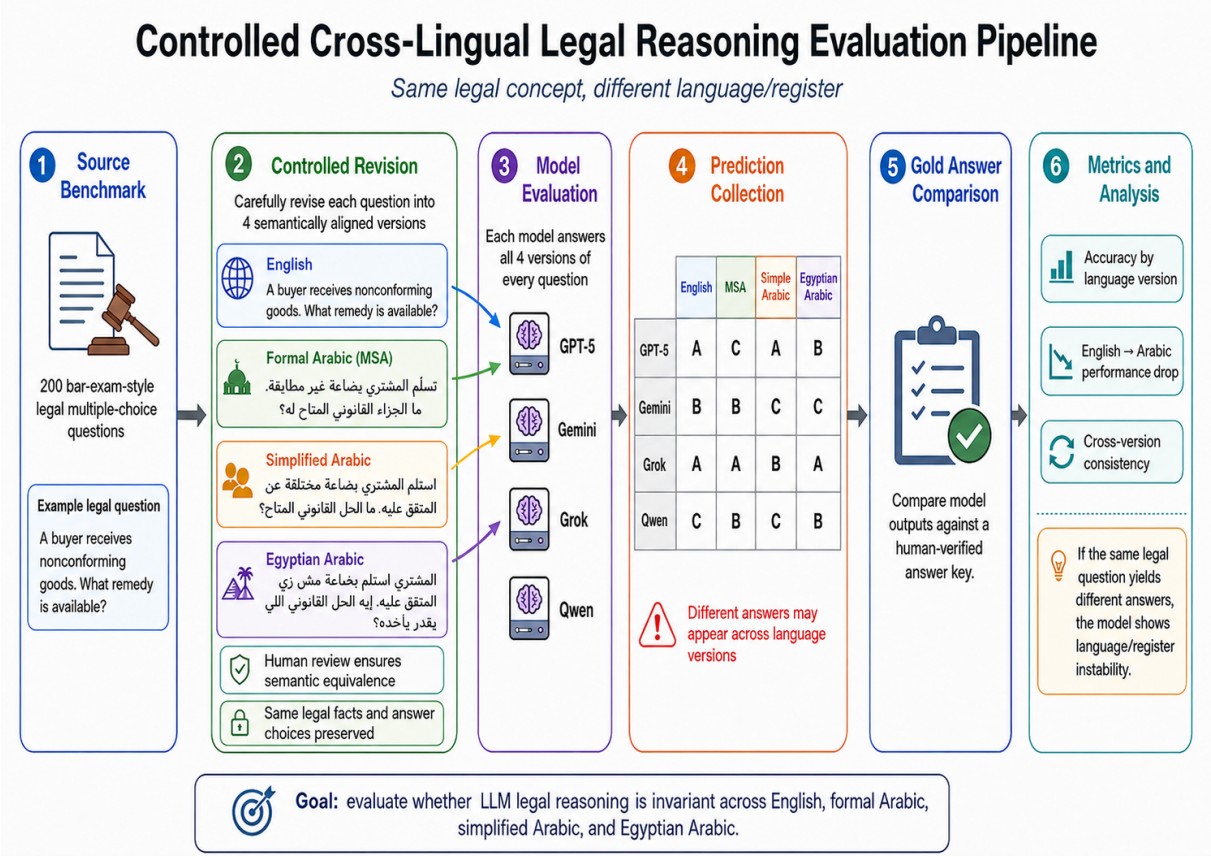


Figure 1: Controlled evaluation pipeline for legal reasoning across English and three Arabic language varieties.

Each model was evaluated on the same 800 aligned instances: 200 English, 200 Original Arabic/MSA, 200 Simple Arabic, and 200 Egyptian Arabic. We accessed all models in May 2026 and performed one run per model–item–language combination, yielding 800 responses per model and 3,200 responses overall. No repeated sampling or majority voting was used.

We used each interface's default settings. Web browsing, retrieval, external documents, file uploads, memory, and legal-assistance tools were not enabled. The results therefore represent interface-level behavior during the evaluation period rather than deterministic fixed-checkpoint performance.

Before the full evaluation, we piloted several instruction variants for output compliance and then fixed one concise template:

> Read the following legal multiple-choice question and select the single best answer. Return only the answer letter: A, B, C, or D.

For Arabic conditions, the instruction was translated into the corresponding Arabic variety. Each prompt contained one complete legal fact pattern and four choices labeled A–D. The question language and instruction language matched, while answer labels and choice order were preserved across conditions.

If a model provided an explanation, its final answer was extracted when clearly identifiable. Responses with a missing, invalid, or ambiguous A–D selection were marked incorrect. Because commercial interfaces and hidden system prompts may change, exact replication over time may be limited.

3.5 Scoring and Evaluation Metrics

Model outputs were scored by exact match against the human-verified answer key. A response was correct only when its selected answer letter matched the gold answer. When a response included an explanation, we extracted the final answer if it was clearly identifiable. Missing, ambiguous, conflicting, or non-A–D answers were marked incorrect.

For model m and language condition l, accuracy is

$$A_{m,l} = \frac{C_{m,l}}{N} \times 100, \tag{1}$$

where $C_{m,l}$ is the number of correct responses

and $N = 200$. We report Wilson 95% confidence intervals for each language-specific accuracy estimate (Wilson, 1927). Because every item has four answer choices, uniform random selection gives a nominal chance baseline of 25%. We also compute an empirical majority-answer-position baseline that always predicts the most frequent gold-answer position:

$$B_{\text{mode}} = \frac{\max(n_A, n_B, n_C, n_D)}{N} \times 100, \quad (2)$$

where $n_A, \ldots, n_D$ are the frequencies of the four gold-answer positions. This baseline accounts for possible answer-position imbalance.

We additionally report average Arabic accuracy,

$$A_m^{\text{Ar}} = \frac{A_{m,\text{OA}} + A_{m,\text{SA}} + A_{m,\text{EA}}}{3}, \quad (3)$$

the English–Arabic gap,

$$G_m = A_{m,\text{EN}} - A_m^{\text{Ar}}, \quad (4)$$

and maximum cross-language variation,

$$V_m = \max_{l \in L} A_{m,l} - \min_{l \in L} A_{m,l}, \quad (5)$$

where $L = \{\text{EN}, \text{OA}, \text{SA}, \text{EA}\}$. Lower values of G_m and V_m indicate greater cross-lingual final-answer stability. These measures complement language-specific accuracy but do not assess the validity or cross-language equivalence of the models' underlying reasoning processes.

3.6 Experimental Scope and Reproducibility

This study evaluates final-answer accuracy on translated MBE-style multiple-choice questions. It does not evaluate open-ended explanations, citation quality, legal advice, reasoning-chain faithfulness, different legal systems, or professional legal judgment. The findings are therefore limited to the evaluated questions, linguistic conditions, prompts, and model interfaces.

All models were accessed through their official user-facing interfaces in May 2026 using default settings and one run per model–item–language combination. Because commercial models, hidden system prompts, and interfaces may change, the results represent behavior during the recorded evaluation period rather than deterministic fixed-checkpoint performance. Repeated-run stability and temporal variation remain directions for future work.

4 Results

We evaluate model accuracy and robustness across English (EN), Original Arabic/MSA (OA), Simple Arabic (SA), and Egyptian Arabic (EA). Table 2 reports accuracy, average Arabic accuracy, English–Arabic gaps, and cross-language variation. Figure 2 compares English and Arabic performance across models and language conditions.

Because each condition contains 200 questions, we report Wilson 95% confidence intervals alongside the language-specific accuracies in Table 2. Gemini 2.5 Pro performs below the nominal 25% chance baseline in all three Arabic conditions, while Grok performs below chance in Simple and Egyptian Arabic and near chance in MSA. These comparisons describe observed accuracy relative to the nominal baseline and should not be interpreted as formal significance tests. As shown in Figure 2, all evaluated models perform better in English than their average performance across the three Arabic conditions. The English–Arabic gaps remain substantial, reaching 61.2 percentage points for Gemini 2.5 Pro and 23.0 percentage points for Grok.

4.1 English Accuracy Does Not Transfer Reliably to Arabic

All four models achieve their highest accuracy in English, even though the same 200 legal problems, answer choices, legal issues, and correct answers are preserved across all language conditions. This result directly supports the central claim of the paper: LLM legal reasoning is not fully language-invariant.

Qwen obtains the highest English accuracy at 97.0%, but its performance drops to 79.0% in all three Arabic conditions, producing an English–Arabic gap of 18.0 points. GPT-5 is more stable: it scores 90.0% in English and remains between 80.0% and 82.0% across Arabic variants, with the smallest English–Arabic gap of 9.0 points. In contrast, Gemini 2.5 Pro drops from 80.5% in English to 19.3% average Arabic accuracy, producing the largest gap at 61.2 points. Grok also shows a substantial gap, falling from 45.0% in English to 22.0% average Arabic accuracy.

These results show that English performance alone is not a reliable estimate of Arabic legal reasoning performance. A model may appear strong on English legal questions while becoming substantially less reliable when the same questions are

Model	EN	OA	SA	EA	Avg. Ar.	EN–Ar Gap	Max Var.
GPT-5	90.0 (85.1–93.4)	82.0 (76.1–86.7)	81.0 (75.0–85.8)	80.0 (73.9–85.0)	81.0	9.0	10.0
Gemini 2.5 Pro	80.5 (74.5–85.4)	19.5† (14.6–25.5)	21.0† (15.9–27.2)	17.5† (12.9–23.4)	19.3	61.2	63.0
Grok	45.0 (38.3–51.9)	26.5 (20.9–33.0)	16.5† (12.0–22.3)	23.0† (17.7–29.3)	22.0	23.0	28.5
Qwen	97.0 (93.6–98.6)	79.0 (72.8–84.1)	79.0 (72.8–84.1)	79.0 (72.8–84.1)	79.0	18.0	18.0

Table 2: Accuracy (%) and cross-language robustness. Parentheses show Wilson 95% confidence intervals for language-specific accuracies ($N = 200$). † Observed accuracy below the nominal 25% uniform-choice baseline.

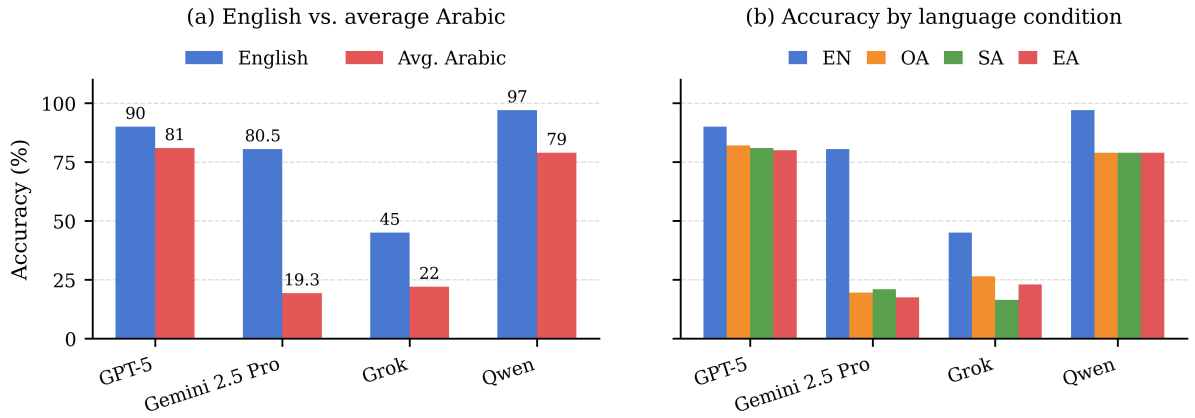


Figure 2: Cross-lingual accuracy across models. (a) English versus average Arabic accuracy. (b) Accuracy disaggregated by language condition: English (EN), Original Arabic (OA), Simple Arabic (SA), and Egyptian Arabic (EA).

expressed in Arabic.

4.2 Accuracy and Robustness Give Different Rankings

The results also show that accuracy and robustness are not the same. Qwen is the strongest model by peak English accuracy and average Arabic accuracy, but GPT-5 is the most robust model across language conditions. GPT-5 has the lowest maximum cross-language variation at 10.0 points, compared with 18.0 points for Qwen, 28.5 points for Grok, and 63.0 points for Gemini 2.5 Pro.

This distinction matters for multilingual legal AI. A model with the highest English score may still show a meaningful English–Arabic gap. Conversely, a model with slightly lower peak accuracy may be more dependable across language varieties. In this benchmark, Qwen is the strongest model by English accuracy, while GPT-5 is the strongest model by cross-language stability.

4.3 Arabic Variants Reveal Model-Specific Weaknesses

The three Arabic conditions reveal different model-specific robustness patterns. GPT-5 shows the most stable Arabic behavior, with only a two-point range across Original Arabic/MSA, Simple Arabic, and Egyptian Arabic. This suggests that GPT-5 is affected by the shift from English to Arabic, but is less sensitive to differences among Arabic registers.

Qwen produces identical accuracy across the three Arabic variants, with 79.0% in Original Arabic/MSA, Simple Arabic, and Egyptian Arabic. This pattern suggests a general English–Arabic transfer gap rather than a weakness tied to one Arabic register or dialect. In other words, Qwen remains strong in Arabic relative to the other models, but its English performance still overestimates its Arabic legal reliability.

Grok is most sensitive to Simple Arabic. Its accuracy drops from 26.5% in Original Arabic/MSA to 16.5% in Simple Arabic, before increasing to 23.0% in Egyptian Arabic. This pattern shows that simplification does not necessarily improve model

Model	Gap	Var.	Pattern
GPT-5	9.0	10.0	Mild sensitivity
Qwen	18.0	18.0	English–Arabic gap
Grok	23.0	28.5	Simple Arabic weakness
Gemini 2.5 Pro	61.2	63.0	Broad Arabic degradation

Table 3: Dominant robustness patterns by model. Gap = English–Arabic gap; Var. = maximum cross-language variation.

reasoning. Instead, simplified wording may change how the model interprets the relationship among facts, rules, and answer choices.

Gemini 2.5 Pro performs poorly across all Arabic conditions, with scores below 22.0%. This indicates broad Arabic degradation rather than a dialect-specific failure. These findings show that Arabic should not be treated as a single evaluation condition. Formal Arabic, simplified Arabic, and dialectal Arabic expose different weaknesses, and evaluating only one Arabic variety may hide important reliability problems.

4.4 Robustness Pattern Analysis

Table 3 summarizes the dominant robustness pattern for each model. The robustness patterns reinforce the main findings. GPT-5 shows only mild cross-lingual sensitivity, while Qwen shows a consistent English–Arabic transfer gap. Grok shows the clearest simplification-sensitive weakness because Simple Arabic is its lowest condition. Gemini 2.5 Pro shows broad Arabic degradation, with low accuracy across all Arabic variants. Overall, the results show that changing the language, register, or dialect can substantially change model performance even when the legal meaning remains fixed. Therefore, multilingual legal evaluation should report not only accuracy, but also English–Arabic gaps and cross-language variation.

4.5 Qualitative Error Inspection

To contextualize the aggregate results, we conducted a preliminary qualitative inspection of Arabic failure cases. This inspection was illustrative rather than exhaustive and focused on cases where the English version was answered correctly but at least one Arabic version was answered incorrectly.

The inspection showed that Arabic errors were not limited to output-format problems. In many cases, models selected a different legal conclusion in Arabic even when the facts, answer choices, and intended correct answer were preserved through legal and linguistic review.

Common failure patterns included confusion between legally similar answer choices, weaker handling of translated legal conditions or exceptions, and instability when the same legal relationship was expressed in simplified or dialectal Arabic. These observations support the quantitative finding that cross-language performance loss reflects robustness limitations rather than only formatting noise.

5 Discussion

5.1 English Accuracy Overestimates Arabic Legal Reliability

All models achieve their highest accuracy in English despite receiving semantically aligned questions across all language conditions. This indicates that performance depends not only on the legal task, but also on its language, register, and dialect. Similar cross-lingual gaps have been reported in multilingual evaluations (Hu et al., 2020; Liang et al., 2020; Bandarkar et al., 2024) and Arabic-focused benchmarks (Abdelali et al., 2024; Koto et al., 2024).

English-only testing may therefore overestimate reliability for Arabic-speaking users. This is especially concerning in legal contexts, where differences in interpreting facts, rules, exceptions, or answer choices can change the legal conclusion (Guha et al., 2023; Katz et al., 2024).

5.2 Accuracy and Robustness Are Different Evaluation Dimensions

Qwen achieves the highest English and average Arabic accuracy, whereas GPT-5 exhibits the smallest cross-language variation. Thus, the most accurate model is not necessarily the most stable across languages. Existing legal benchmarks emphasize task accuracy and legal knowledge, while multilingual benchmarks emphasize cross-language transfer (Guha et al., 2023; Niklaus et al., 2024; Joshi et al., 2024). Multilingual legal evaluations should combine both perspectives by reporting language-specific accuracy, English–Arabic gaps, and cross-language variation.

5.3 Arabic Variety Matters

Models behave differently across Original Arabic/MSA, Simple Arabic, and Egyptian Arabic. GPT-5 and Qwen remain relatively stable, Grok

performs worst in Simple Arabic, and Gemini 2.5 Pro performs poorly across all Arabic conditions.

Formal Arabic evaluation alone may therefore hide important weaknesses. Dialectal Arabic remains underrepresented in LLM evaluation (Mousi et al., 2025), with persistent challenges in dialect identification, generation, and translation (Mousi et al., 2025). Broader Arabic benchmarks also demonstrate the importance of evaluating diverse tasks and linguistic conditions (Abdelali et al., 2024; Koto et al., 2024).

Arabic legal benchmarks should consequently include multiple language varieties. However, our findings are limited to MSA, Simple Arabic, and Egyptian Arabic and should not be generalized to untested dialects.

5.4 Implications for Multilingual Legal AI

Multilingual legal evaluation should test whether models preserve performance across semantically equivalent language versions. Existing benchmarks evaluate legal knowledge, argument reasoning, and rule application (Guha et al., 2023; Abu Shairah et al., 2025), while multilingual legal resources extend evaluation across languages, domains, and jurisdictions (Niklaus et al., 2024; Joshi et al., 2024). However, limited work directly tests whether models preserve their answers across aligned language varieties. Our controlled design shows that model performance changes with language, register, and dialect even when the underlying legal content remains fixed. These results concern final-answer robustness and do not establish the validity or cross-language equivalence of the models' reasoning processes. Such instability may create unequal reliability for users asking equivalent questions in different language varieties. Prior work therefore emphasizes the need for linguistically inclusive evaluation (Hu et al., 2020; Bandarkar et al., 2024; Mousi et al., 2025). Legal AI systems should be evaluated for both accuracy and cross-lingual robustness before use in multilingual, high-stakes settings.

6 Limitations and Future Work

This study evaluates 200 MBE-style multiple-choice questions, four general-purpose models, four aligned language conditions, and one run per model–item–language combination. The findings are therefore specific to this benchmark and should not be generalized to other jurisdictions, Arabic legal systems, legal tasks, language varieties, or models. Because commercial models and their interfaces may change, the results represent an interface-level snapshot from May 2026. Moreover, final-answer accuracy does not demonstrate that models use faithful or equivalent legal reasoning across languages.

Because NCBE materials are protected, we cannot release the original questions, translations, answer choices, or official explanations. Our GitHub repository will instead provide aggregate and anonymized item-level results, model predictions, correctness labels, prompts, scoring and analysis code, and author-created synthetic examples: `https://github.com/hala00001/Rationality-of-Large-language-models-`. These resources support partial, but not exact, reproducibility.

Future work should expand the evaluation to additional legal domains, jurisdictions, task formats, models, and Arabic varieties, including Gulf, Levantine, Iraqi, Moroccan, and Sudanese Arabic. Repeated runs could measure response stability, answer-flip rates, and temporal variation. Future studies should also examine explanations, citations, legal-rule application, and cross-language reasoning consistency, as well as mitigation methods such as Arabic legal retrieval, dialect-aware prompting, self-consistency, and reliability calibration.

7 Conclusion

We evaluated four LLMs on 200 aligned legal questions across English, Original Arabic, Simple Arabic, and Egyptian Arabic. All models performed best in English, but their Arabic performance varied substantially. GPT-5 showed the greatest cross-lingual stability, while Qwen achieved the highest average accuracy. These results demonstrate that strong English performance does not guarantee reliable Arabic legal reasoning, highlighting the need for multilingual and dialect-sensitive evaluation in high-stakes legal AI.

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, Yousseif 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, pages 487–520.

Harethah Abu Shairah, Somayah AlHarbi, Abdulaziz AlHussein, Sameer Alsabea, Omar Shaqaqi, Hebah AlShamlan, Omar Knio, and George Turkiyyah. 2025. Alarb: An arabic legal argument reasoning benchmark. In *Proceedings of the Third Arabic Natural Language Processing Conference*.

Dana R Alsagheer, Abdulrahman Kamal, Mohammad Kamal, Cosmo Yang Wu, and Weidong Shi. 2025. The lawyer that never thinks: Consistency and fairness as keys to reliable AI. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9943–9954, Vienna, Austria. Association for Computational Linguistics.

Lucas Bandarkar, Davis Liang, Benjamin Muller, Mikel Artetxe, Sahaj Shukla, Dimi Husa, Naman Goyal, Abhinandan Krishnan, Luke Zettlemoyer, and Madian Khabsa. 2024. The belebele benchmark: a parallel reading comprehension dataset in 122 language variants. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*.

M. Saiful Bari, Yazeed Alnumay, Norah A. Alzahrani, and 1 others. 2024. ALLaM: Large language models for arabic and english. *Preprint*, arXiv:2407.15390.

Fanar Team. 2025. Fanar: An arabic-centric multimodal generative ai platform. *Preprint*, arXiv:2501.13944.

Google DeepMind. 2025. Gemini 2.5 pro model card. Accessed: 2026-05.

Neel Guha, Julian Nyarko, Daniel E. Ho, Christopher Ré, Adam Chilton, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel N. Rockmore, Diego Zambrano, Caleb Talisman, Enamul Hoque, Faiz Surani, Frank Fagan, Galit Sarfaty, Gregory M. Dickinson, Hadar Porat, Jacob Hegland, Jessica Wu, and 2 others. 2023. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization. In *Proceedings of the 37th International Conference on Machine Learning*.

Abhinav Joshi, Shounak Paul, Akshat Sharma, Pawan Goyal, Saptarshi Ghosh, and Ashutosh Modi. 2024. IL-TUR: Benchmark for Indian legal text understanding and reasoning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11460–11499. Association for Computational Linguistics.

Daniel Martin Katz, Michael James Bommarito, Shang Gao, and Pablo Arredondo. 2024. Gpt-4 passes the bar exam. *Philosophical Transactions of the Royal Society A*, 382(2270):20230254.

Fajri Koto, Haonan Li, Sara Shatnawi, 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*, pages 5622–5640.

Yaobo Liang, Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, and 1 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*, pages 6008–6018.

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.

Joel Niklaus, Veton Matoshi, Matthias Stürmer, Ilias Chalkidis, and Daniel E. Ho. 2024. MultiLegalPile: A 689gb multilingual legal corpus. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pages 15077–15094.

OpenAI. 2025. Introducing gpt-5. Accessed: 2026-05.

Edwin B. Wilson. 1927. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212.

xAI. 2025. Grok 3 Beta – The Age of Reasoning Agents. Official xAI announcement. Accessed 2026-04-30.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, and 1 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Fine-tuning Small Language Models for Cross-Prompt Arabic Essay Scoring: A Benchmarking Study

Salam Albatarni[1], **Fahim Dalvi**[2], **Sohaila Eltanbouly**[1], **Tamer Elsayed**[1]

[1]Computer Science and Engineering Department, Qatar University
[2]Qatar Computing Research Institute, Hamad Bin Khalifa University

{sa1800633, se1403101, telsayed}@qu.edu.qa,
faimaduddin@qf.org.qa

Abstract

Automated Essay Scoring (AES) has been studied since the early days of computational methods. While recent advances incorporating large language models (LLMs) have significantly improved performance, most research has focused on English. In this paper, we present the first benchmark of Arabic cross-prompt AES using Arabic-centric LLMs. We evaluate ALLAM 7B, JAIS 8B, and FANAR 9B under zero-shot, few-shot, and fine-tuning settings, and compare them against established AES systems and frontier LLMs. Fine-tuning substantially improves performance across all models, with FANAR achieving state-of-the-art results and outperforming the strongest baseline by 8 points on LAILA dataset. We further show that fine-tuned Arabic-centric LLMs generalize effectively in cross-dataset setting (LAILA $\rightarrow$ ZAEBUC). Finally, we analyze the best-performing model, examining error patterns, trait interactions, and the effects of multi-trait autoregressive scoring.

1 Introduction

Automated Essay Scoring (AES) is the task of automatically assigning a proficiency score to a student's essay written in response to a specific writing prompt, where the prompt defines the writing task and the topic the student is expected to address. Its importance lies in its ability to reduce the time, cost, and human effort required to grade essays, particularly in large-scale assessment settings. Beyond holistic scoring, which assigns a single overall score, trait scoring evaluates specific aspects of writing such as grammar, relevance, and vocabulary. Trait-level feedback is especially valuable in educational contexts, enabling students to identify strengths and weaknesses in their writing.

AES systems are typically developed under two paradigms. In prompt-specific AES, models are trained and evaluated on essays from the same prompt, enabling specialization in that prompt. However, they often fail to generalize to essays written for unseen prompts, limiting practicality in real educational settings. In contrast, the cross-prompt setup trains models on essays from a set of source prompts and evaluates them on unseen target prompts, with the goal of improving generalization across prompts. A wide range of approaches has been proposed for cross-prompt AES, starting with feature-based methods (Li and Ng, 2024), followed by neural encoder-based models (Chen et al., 2025), and more recently, large language model (LLM)-based approaches (Yoshida, 2025). Although LLMs have demonstrated strong zero-shot capabilities across NLP tasks (Noureldien et al., 2025), existing AES studies suggest that their effectiveness is often limited when used in isolation (Mansour et al., 2024; Lee et al., 2024).

While most research has focused on English AES, Arabic AES is gaining momentum through new shared tasks and datasets (Bashendy et al., 2025, 2026). However, the progress has largely been limited to feature-based and neural approaches (Sayed et al., 2025; Albatarni et al., 2026), with little attention paid to LLMs. This gap is particularly notable given the recent emergence of Arabic-centric LLMs (Team et al., 2025; Bari et al., 2025) and the availability of large-scale Arabic AES benchmarks (Bashendy et al., 2026).

In this work, we investigate the potential of LLMs for Arabic AES. Specifically, we fine-tune three Arabic-centric LLMs (hereafter LLM^{ft}) for cross-prompt AES and evaluate them on the largest Arabic AES dataset, with the goal of providing a benchmark for Arabic-centric LLM performance. We aim to answer the following research questions:

RQ1 Does LLM^{ft} improve performance compared to zero-shot and few-shot variants?

RQ2 How does LLM^{ft} perform compared to state-of-the-art (SOTA) models?

RQ3 How does LLM^{ft} perform in a cross-dataset setting compared to zero-shot and few-shot

Proceedings of The Fourth Arabic Natural Language Processing Conference (ArabicNLP 2026), pages 11–21
October 28-29, 2026 ©2026 Association for Computational Linguistics

prompting, and established AES baselines?

RQ4 What insights additional analyses provide into the reliability and scoring behavior of LLM^{ft}?

Our experiments show that LLM^{ft} significantly improves over its base variants and significantly outperforms SOTA baselines. Our contributions are three-fold:

(1) We evaluate the performance of fine-tuning small Arabic-centric LLMs for Arabic cross-prompt AES, providing a systematic benchmark for their performance.

(2) We provide a new SOTA for Arabic cross-prompt AES, offering a stronger point of comparison for future research.

(3) We conduct extensive analyses, including cross-dataset evaluation and error analysis, providing insights into the behavior, generalization, and limitations of LLM^{ft}.

2 Related work

This section reviews prior work on AES, focusing on Arabic AES, followed by studies applying LLMs to AES through prompting and fine-tuning.

AES for Arabic Early Arabic AES research explored rule-based approaches and handcrafted linguistic features (Alqahtani and Alsaif, 2020; Alsanie et al., 2022), fine-tuning AraBERT (Ghazawi and Simpson, 2024), and applying parameter-efficient fine-tuning (Mahmoud et al., 2024). Sayed et al. (2025) pioneered cross-prompt AES for Arabic by introducing a large set of 816 handcrafted features and using them with various cross-prompt models. To advance research in Arabic AES, Bashendy et al. (2025) organized the TAQEEM shared task targeting cross-prompt scoring for Arabic AES, where participating teams primarily used AraBERT-based models and LLM prompting approaches. Most recently, Albatarni et al. (2026) proposed MAPLE, a cross-prompt AES model based on a meta-learning framework that leverages prototypical networks to learn transferable representations across different writing prompts, achieving SOTA performance on Arabic AES.

LLMs Prompting for AES When it comes to the utilization of LLMs for AES, most studies rely on prompting rather than fine-tuning. For example, Stahl et al. (2024) evaluated zero-shot, few-shot, and Chain-of-Thought prompting for essay scoring and feedback generation, while Mansour et al. (2024) explored prompt-engineering strategies with ChatGPT and LLaMA. Similarly, Lee et al. (2024) proposed a zero-shot multi-trait specialization framework that decomposes the rubric into traits, for sequential evaluation. More advanced paradigms include reasoning-model comparisons (Yoshida, 2025) and a multi-agent evaluation framework (Jang et al., 2025). For Arabic AES, Mandhari et al. (2026) introduced a trait-specific framework that explored standard prompting, hybrid prompting with trait-specialist evaluators, and rubric-guided prompting under zero-shot and few-shot settings. Meanwhile, Qwaider et al. (2025) investigated prompt–essay relevance using conventional approaches such as threshold-based classifiers and SVMs, as well as prompting LLMs, including FalconH1, Qwen3, Gemma3, and Jais. Despite these innovations, prompting-based methods still fall short of SOTA AES systems in overall performance.

LLMs Finetuning for AES Few studies have explored fine-tuning LLMs for AES. Liu et al. (2025) fine-tuned GPT-3.5-turbo on TOEFL11 and observed an overrating tendency, though incorporating linguistic features improved the accuracy. Xiao et al. (2025) proposed a dual-process AES framework and showed that fine-tuning LLaMA 3 and GPT-3.5 can achieve performance comparable to conventional SOTA systems. In Arabic AES, Ghazawi and Simpson (2025) evaluated several fine-tuning strategies on ACEGPT and LLaMA, yet their models still underperformed compared to a smaller fine-tuned BERT-based model. Overall, while fine-tuning improves LLM performance, it often remains less competitive than smaller models. Our work addresses this gap by fine-tuning LLMs specifically for Arabic AES.

3 Study Design

In the cross-prompt AES setting, given a set of source prompts P_{src}, and a set of target prompts P_{trg}, the objective is to train a model on P_{src} such that it generalizes effectively to the *unseen* prompts P_{trg}. We aim to estimate the upper-bound performance of LLMs on cross-prompt AES by fine-tuning them for this task.

Multi-trait Scoring We fine-tune LLMs for trait-based essay scoring using a *Multi-Trait (MT)* setup, where a single model jointly predicts scores for multiple essay traits. This formulation allows the model to capture dependencies between traits while

enabling all traits to be predicted using a single model, improving both training and inference efficiency. Given a writing prompt and an essay, the model generates all trait scores in a structured JSON format. The fine-tuning prompt is shown in Figure 1.

```
سيتم إعطاؤك مقال كتبه الطالب ردًا على الموضوع المعطى. مهمتك هي تقييم جميع السمات الكتابية ضمن النطاق المحدد. أخرج النتيجة بصيغة JSON فقط، دون أي شرح أو نص إضافي. يجب أن تكون درجة السمة رقمًا صحيحًا واحدًا داخل النطاق المحدد.

---

الموضوع: نص الموضوع الذي طُلب من الطالب الكتابة عنه
المقال: نص المقال المكتوب من قبل الطالب ردًا على الموضوع
الدرجات: { " الصلة بالموضوع " : 0 - 2 ، " الهيكل العام " : 0 - 5 ، " المفردات " : 0 - 5 ، " االأسلوب والتماسك البنائي " : 0-5 ، " الأفكار والمضمون " : 0-5 ، " الإملاء والترقيم والشكل " : 0-5 ، " البناء والتراكيب " : 0-5 ، " التقييم الكلي " : 0-32 }

- - -

الموضوع: {prompt}
المقال: {essay}
الدرجات:
```

Figure 1: The Arabic prompt used for multi-trait prediction. English translation is provided in Figure 2.

```
You will be given an essay written by a student in response to a prompt. Your task is to evaluate all the writing traits within the specified range.
Output the result in JSON format only, without any additional explanation or text. Each trait's score should be an integer within the specified range.
—
Prompt: The prompt that the student was asked to write about
Essay: The essay written by the student in response to the prompt
Scores: {'Relevance': (0, 2), 'Organization': (0, 5), 'Style': (0, 5), 'Development': (0, 5), 'Mechanics': (0, 5), 'Grammar': (0, 5), 'holistic': (0, 32)}
—
Prompt: {prompt}
Essay: {essay}
Scores:
```

Figure 2: English translation of the Multi-trait prediction prompt. The Arabic version is shown in Figure 1.

Fine-Tuning Setting We fine-tune the LLMs using the Hugging Face `Transformers` library with the `PEFT`[1] framework for parameter-efficient adaptation. Specifically, we apply LoRA to the causal language modeling objective, keeping the base model frozen while training low-rank adapters (rank $r = 64$, $\alpha = 128$, dropout $= 0.1$). The adapters are applied to the attention projection matrices (Key, Query, Value, and Output) and MLP layers (Gate, Up, and Down projections). Supervised fine-tuning is conducted using the `TRL` library's `SFTTrainer`,[2] with AdamW optimizer, a learning rate of 2×10^{-4}, and a cosine scheduler. These hyperparameter values were determined based on preliminary experiments. Moreover, training uses a micro-batch size of 4 with gradient accumulation steps of 16 (global batch size 64), for 10 epochs with a maximum sequence length of 2048. Mixed-precision training with `bfloat16` is enabled for computational efficiency. For inference, we use greedy decoding for all experiments. The best checkpoint is selected based on performance on the development set. All experiments were run using one H200 GPU.

Dataset We used LAILA (Bashendy et al., 2026), a large-scale Arabic AES dataset with 7,859 essays written in response to eight prompts (P1-P8). Essays were annotated for seven traits: relevance (REL) on a scale of 0–2, and organization (ORG), vocabulary (VOC), style (STY), development (DEV), mechanics (MEC), and grammar (GRA) on a scale of 0–5, and a holistic (HOL) score as the sum of all trait scores. We adopt the cross-prompt splits provided by LAILA authors.[3]

Fine-tuned LLMs For fine-tuning, we select small open-source Arabic-centric LLMs. We focus on models in the 7B–9B parameter range, as they are practical to fine-tune while remaining computationally efficient. Moreover, to ensure architectural diversity, we choose models built on different base architectures. Specifically, we finetune FANAR 9B (Team et al., 2025),[4] ALLAM 7B (Bari et al., 2025),[5] and JAIS 8B (Anwar et al., 2025).[6]

Other Scoring Models We compare the fine-tuned LLMs against a range of prompting-based LLM baselines and SOTA engineered models:

1. **Zero-shot (LLM^0)**: We evaluate the 3 fine-tuned LLMs (FANAR 9B, ALLAM 7B, and JAIS 8B) in a zero-shot setting to assess their inherent scoring capabilities without task-specific adaptation. We additionally evaluate larger variants, namely FANAR 27B and JAIS

[1] https://huggingface.co/docs/peft/en/index

[2] https://huggingface.co/docs/trl/en/sft_trainer

[3] https://gitlab.com/bigirqu/laila

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

[5] https://huggingface.co/humain-ai/ALLaM-7B-Instruct-preview

[6] https://huggingface.co/inceptionai/Jais-2-8B-Chat

Model	Size	REL	ORG	VOC	STY	DEV	MEC	GRA	HOL	AVE
ALLAM^0	7B	0.230	0.388	0.416	0.463	0.393	0.415	0.434	0.377	0.390
JAIS^0	8B	0.112	0.126	0.169	0.176	0.153	0.215	0.211	0.234	0.175
JAIS^0	70B	0.112	0.334	0.431	0.367	0.413	0.433	0.443	0.461	0.374
FANAR^0	9B	0.457	0.536	0.421	0.550	0.487	0.371	0.448	0.521	0.474
FANAR^0	27B	0.040	0.459	0.262	0.084	0.261	0.175	0.095	0.427	0.225
GPT-5^0	-	0.620	0.709	0.540	0.586	0.652	0.410	0.547	0.716	0.598
ALLAM^{fs}	7B	0.096	0.253	0.313	0.316	0.273	0.350	0.339	0.347	0.286
JAIS^{fs}	8B	0.093	0.226	0.277	0.278	0.255	0.260	0.261	0.304	0.244
FANAR^{fs}	9B	0.425	0.591	0.552	0.607	0.563	0.582	0.594	0.651	0.570
MOOSE	0.6B	0.411	0.627	0.642	0.649	0.585	0.586	0.623	0.649	0.597
XGB	-	0.360	0.645	0.641	0.641	0.583	0.577	0.619	0.679	0.593
MAPLE	0.16B	0.290	0.647	0.686	0.702	0.607	0.634	0.664	0.723	0.619
TRATES	-	0.557	0.696	0.657	0.664	0.652	0.608	0.643	0.744	0.653
ALLAM^{ft}	7B	0.654$^{\star\bullet}$	0.771$^{\star\bullet}$	0.731*	0.741*	0.716	0.696$^{\star\bullet}$	0.708$^{\star\bullet}$	0.781$^{\star\bullet}$	0.725$^{\star\bullet}$
JAIS^{ft}	8B	0.552	0.743*	0.708*	0.714	0.681	0.659	0.678	0.746	0.685
FANAR^{ft}	9B	**0.656**$^{\star\bullet}$	**0.789**$^{\star\bullet}$	**0.743***	**0.752**$^{\star\bullet}$	**0.718**$^{\star\bullet}$	**0.714**$^{\star\bullet}$	**0.726**$^{\star\bullet}$	**0.796**$^{\star\bullet}$	**0.737**$^{\star\bullet}$

Table 1: Performance of LLM^{ft} against the other scoring models on LAILA test split. Statistical significance is done using a t-test ($p < 0.05$) over results across individual prompts. Improvements over TRATES are denoted by *. Improvements over JAIS^{ft} are denoted by $^{\bullet}$. FANAR^{ft} and ALLAM^{ft} are not statistically significant.

70B. We also include GPT-5 as a reference for large frontier models.

2. **Few-shot (LLM^{fs})**: Similarly, we evaluate the 3 LLMs under few-shot scoring using 5 in-context examples randomly sampled from different prompts in P_{src}, to assess the extent to which task demonstrations improve scoring performance. Since few-shot performance can be sensitive to the selected examples, we repeat each experiment three times with different random seeds and report the average performance.
3. **SOTA**: We select 4 engineered models that achieve the best performance on LAILA. MOOSE is an adaptation of an English cross-prompt AES model based on a mixture-of-experts architecture (Chen et al., 2025; Bashendy et al., 2026). XGB is a strong feature-based model reported in (Bashendy et al., 2026). MAPLE is a meta-learning framework for cross-prompt AES that incorporates prompt, rubric, and generic features into the essay representation (Albatarni et al., 2026). TRATES is a system that generates trait-specific features and combines them with linguistic features for scoring using a simple NN (Eltanbouly et al., 2025). We use its Arabic adaptation by Elbahnasawi et al. (2026), which is the current SOTA on LAILA.

The LLM experiments were run using vLLM (Kwon et al., 2023) and relevant API's for closed models. For smaller LLMs, we use *structured outputs*, since the smaller models are not able to adhere to the output format consistently.

Evaluation Metric We evaluate our models using Quadratic Weighted Kappa (QWK) (Cohen, 1968), a widely used metric in AES. QWK quantifies agreement between human and system scores while accounting for the severity of disagreements, making it suitable for ordinal score scales.

4 Experimental Evaluation

This section addresses the first three research questions: LLM^{ft} vs. Non-finetuned LLMs, LLM^{ft} vs. SOTA, and cross-dataset performance of LLM^{ft}.

4.1 LLM^{ft} vs. Non-tuned LLMs (RQ1)

Table 1 compares LLM^{ft} performance with the other scoring models on LAILA test set.[7]

Non-tuned LLMs For the prompting-based LLMs, FANAR^{fs} and JAIS^{fs} perform better than their LLM^0 variants, with improvements of 10 and 7 points, respectively. Notably, the larger model, JAIS^0 70B, outperforms JAIS^{fs} 8B by 13 points. In contrast, ALLAM shows the opposite trend, where few-shot prompting leads to a performance drop of

[7]Per-prompt results are in Table 4.

almost 10 points. Overall, the results of non-tuned LLMs are underwhelming, not surpassing existing SOTA models or the frontier model, GPT-5[0].

Fine-tuning Fine-tuning leads to substantial performance improvements, with gains of 17, 34, and 44 points for FANAR^{ft}, ALLAM^{ft}, and JAIS^{ft}, respectively, over their best non–tuned configurations. Interestingly, although the differences in performance between LLM^0 models and also between the LLM^{fs} models were large (e.g., FANAR^0 outperforming ALLAM^0 and JAIS^0 by 8 and 30 points, respectively), those gaps narrow significantly after fine-tuning, with FANAR^{ft} exceeding ALLAM^{ft} and JAIS^{ft} by only 1 and 5 points, respectively.

4.2 LLM^{ft} vs SOTA Models (RQ2)

Table 1 shows that LLM^{ft} outperforms all SOTA models by a substantial margin. In fact, the best-performing LLM, FANAR^{ft}, outperforms the best baseline model, TRATES, by 8 points, establishing a new SOTA performance in Arabic AES on LAILA dataset. Moreover, improvements of FANAR^{ft}, ALLAM^{ft}, and JAIS^{ft} over TRATES are statistically significant in 8, 7, and 2 traits, respectively, with both FANAR^{ft} and ALLAM^{ft} also showing statistically significant improvements on average. Notably, differences in performance between FANAR^{ft} and ALLAM^{ft} are not statistically significant across any individual trait, indicating comparable performance between the two models. In contrast, improvements of FANAR^{ft} over JAIS^{ft} are statistically significant in all traits except VOC.

4.3 LLM^{ft} in Cross-dataset Setting (RQ3)

We now turn to evaluating LLM^{ft} in a cross-dataset setting. This setting assesses the generalizability of AES models not only across unseen prompts but also across different essay types and demographics (Eltanbouly et al., 2026). It is particularly relevant in low-resource scenarios, where small datasets are insufficient for training billion-parameter LLMs.

Dataset For this experiment, we use the ZAEBUC dataset (Habash and Palfreyman, 2022). It contains 214 Arabic essays written by first-year university students across three topics. While ZAEBUC differs from LAILA in prompts, essay characteristics, and annotation scheme, the two populations are expected to have broadly comparable language proficiency, as LAILA targets high school students, including those in their final year, who are at a similar educational stage to first-year university students. The dataset provides CEFR language proficiency annotations, which we map to scores in the range [1–6], following Habash and Palfreyman (2022). For predictions, we map LAILA's holistic scores from the original [0–32] range to the [1–6] range used by ZAEBUC. Given the small size of ZAEBUC, we report QWK over the entire dataset rather than computing prompt-level scores and averaging them. Additionally, we include RMSE as a complementary metric, as QWK can exhibit instability on small datasets (Albatarni and Elsayed, 2026).

Model	Size	QWK	RMSE
ALLAM^0	7B	0.267	1.189
JAIS^0	8B	0.160	1.187
FANAR^0	9B	0.451	0.877
ALLAM^{fs}	7B	0.470	0.772
JAIS^{fs}	8B	0.333	1.363
FANAR^{fs}	9B	0.306	0.972
RF	-	0.477	0.639
XGB	-	0.427	0.838
MOOSE	0.6B	0.329	1.083
ALLAM^{ft}	7B	0.481	0.697
JAIS^{ft}	8B	0.444	0.710
FANAR^{ft}	9B	**0.533**	**0.628**

Table 2: Cross-dataset performance of LLM^{ft} and baseline systems (LAILA → ZAEBUC)

Scoring Models We evaluate the following models on ZAEBUC:

1. **LLM^0**: Since ZAEBUC contains only the overall score of the essays, we prompt the same 3 LLMs (FANAR, ALLAM, and JAIS) to predict the holistic score (prompt depicted in Figure 6), with a score range of 1–6.
2. **LLM^{fs}**: We use the same prompt as LLM^0, augmented with 5 random examples from LAILA. The examples include holistic scores only. Since the holistic score range in LAILA is 0–32, we prompt the LLM to generate scores in this range, then map the predictions to ZAEBUC's score range. We repeat the experiment three times with different random seeds and report the average performance.
3. **Qayyem's Models**: We consider the models available in Qayyem,[8], an online Arabic AES platform (Elbahnasawi et al., 2026) namely RF, XGB, and MOOSE.

[8]At the time of the experiments

نص الموضوع: باتَ إهْتمام وحماس المراهقين لِتعلُّمْ رِياضةٍ جديدة أو الإنتظام في حُضور التَدْريبات الرياضية الخاصة بِها يَتضَاءل يومًا بعد يوم؛ حتَى صَارَ يدُق نَاقُوس خَطرٍ مقلق ينْذر بوجُود جيلٍ ضَعيف البنيَة (الجسم). اكتب مقالًا مكوّنا من ثلاثمائة (٣٠٠) كلمة توضّح فيه أسباب انتشار هذه الظاهرة، مُراعيًا سمات المقال التفسيري، وسلامة اللغة، ومُوظفًا علامات الترقيم وأدوات الربط بشكلٍ صحيح.

Prompt (English translation): The interest and enthusiasm of teenagers to learn a new sport or regularly attend its training sessions is dwindling day by day, to the point that it is sounding an alarming warning of a generation with weak physiques. Write an essay of 300 words explaining the reasons for the spread of this phenomenon, observing the characteristics of an expository essay, maintaining correct grammar, and employing punctuation and conjunctions correctly.

الموضوع: أن المراهقين اليوم أصبحو يميلون الى الاشياء الخطرة عليهم مما أدى الى ترك الرياضة والاكتفاء بالراحة لم يدركو بان الراحة لا تكمل الى بالرياضة فراحة الجسد والبدن تكمل بتنوع الرياضة مثل التنس والجري وكرة القدم وغيرها من ذلك فاصبحو يروون بعدم وجود أهمية الرياضة كونها ليست أساسية في الحياة فيجب على كل مراهق بان يدرك خطورة التهاون في الجسد وسلامتة فسلامة الجسد تكمل باستخدام الرياضة كافة بانواعها والالتزام به يجب عليهم الذهاب البطولات حتى يتشجع من خلاله أو حضور مبارة أو دوريات عالية حتى يتجشع ويبادر من نفسه الصحة جسمه من الامراض مستقبلاً ويجب الالتزام بالتدريبات حتى يتمكن من الاستفادة من ذلك فلا يركز الفرد على شئ معين في الرياضة حتى لا يتملل من ذلك وترك الرغبة بالالتزام فيه يخصص يومين فالاسبواع على الاقل الذهاب حتى لو يوم فالرياضة في يومنا هذاء مهمه ومهمه جداً يشارك في البطولات والانديه المدرسية حتى يتمكن من الاستفادة بها وفيها أيضاً فترك الشيء يعني أنعدامه في حياة كل مراهق ومراهقه اليوم وغداً

الموضوع: تزداد السمنة يوم بعد يوم في قطر وذلك لاهمال الافراد لصحتهم وعدم ممارستهم للرياضة وقضاء اغلب اوقاتهم في المنزل وعدم استغلالهم للوقت وعمل انشطة رياضية. ارى ان هناك عدة اسباب لهذه المشكلة ومنها ان العائلة قد تهمل احيانا الرياضه ولا تمانعون في جلوس ابنائهم في البيت طوال الوقت وعدم الحركه ولكن يجب على الاهالي تخصيص وقت للرياضة او تسجيل ابنئهم في اندية رياضية فنحن في قطر تتوفر لدينا العديد من الاندية للانشطة الرياضية وهي نوادي الرياضة او الاندية الكروية او السباحة او حتى تعلم فنون القتال، كل هذه الاندية تمكن للاطفال الانضمام لها وتخصيص وقت يوميا او حتى تخصيص اربع ايام في الاسبوع لهذه الانشطة وتسجيل كل ابن في ما يحب للاستمتاع والفائدة من حيث عمله لحركات رياضية. ايضا اصبحت بعض المدارس تهمل الحصص الرياضية لانهم يعتقدون بأن المواد الاخرى اكثر اهمية ولكن هذا خطأ ويجب على المدارس ايضا تخصيص انشطة رياضية للطلاب وتمكن ذلك عن طريق عمل عدة انشطة في وقت الفسحه وجعل الطلاب يشاركون.

Figure 3: Two essays from Prompt 1 with a true relevance score of 1. The left essay (011026) is correctly scored as 1, while the right essay (010926) is incorrectly scored as 2.

4. **LLM^{ft}**: We train LLM^{ft} on all prompts in LAILA, using the most frequent number of steps across the individual folds for each LLM, and then evaluate the resulting model on ZAEBUC. LLM^{ft} predicts all LAILA traits; however, we consider only HOL prediction and map the scores to ZAEBUC's scale.

Evaluation Results The results are presented in Table 2. Overall, the performance of LLM^0 on ZAEBUC closely mirrors its behavior on LAILA dataset, where ALLAM and FANAR consistently outperform JAIS. Notably, although JAIS^0 shows lower performance in terms of QWK, RMSE indicates that its errors are comparable to ALLAM^0, with only a small difference. LLM^{fs} prompting improves the performance of ALLAM and JAIS by 20 and 17 points, respectively, whereas FANAR experiences a 15-point decline. This suggests that few-shot examples from another dataset can benefit some LLMs but may also introduce instability for others.

In contrast, fine-tuning on LAILA leads to substantial performance gains across all models. Specifically, JAIS^{ft} and FANAR^{ft} improve by 11 and 8 points, respectively, over their best non-tuned configurations, with consistent improvements also observed in RMSE. ALLAM^{ft}, however, improves by only 1 QWK point. This demonstrates that supervised fine-tuning on essay data enhances cross-dataset generalization, even when the training and test sets differ in demographics and topics. Importantly, direct fine-tuning on ZAEBUC would not be practical due to its extremely small size.

Finally, when compared against the engineered AES systems, FANAR^{ft} outperforms the strongest model (RF) by 6 QWK points while achieving comparable RMSE. Additionally, ALLAM^{ft} achieves performance comparable to RF on both QWK and RMSE. Unlike the substantial gains observed on LAILA, improvements in the cross-dataset setting are more modest, underscoring the continued challenge of generalizing across datasets.

5 Performance Analysis

In this section, we address the final research question: what do additional analyses reveal about the reliability and scoring behavior of LLM^{ft}? We conduct a detailed analysis on the best-performing model, FANAR^{ft}, and investigate the following sub-research questions:

1. How do errors in the REL trait affect the prediction of other traits? (§5.1)
2. What are the underlying causes of errors in the REL trait, as observed through qualitative case study analysis? (§5.2)
3. Does single-trait fine-tuning yield better performance than multi-trait fine-tuning? (§5.3)
4. To what extent is the multi-trait model influenced by autoregressive bias during generation? (§5.4)

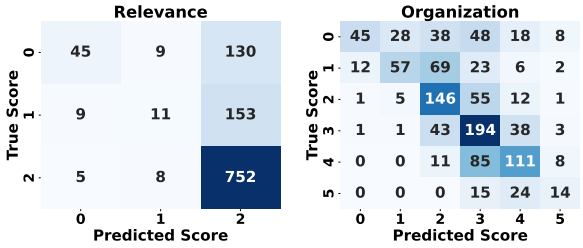


Figure 4: REL and ORG traits confusion matrix for P1.

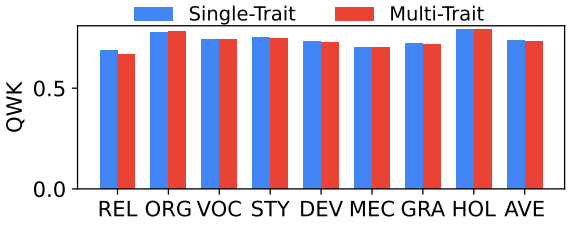


Figure 5: LLM^{ft} results with ST and MT models.

5.1 Error Analysis

We begin by examining writing prompt P1, which is the most challenging prompt, with an average QWK of 0.488, compared to the other prompts, with QWK ranging from 0.733 to 0.805 (see Appendix A). To understand the source of this performance gap, we examine the REL trait, which obtains the lowest QWK for P1 (0.303). We then analyze the ORG trait to investigate how errors in REL prediction propagate to the scoring of other traits. Figure 4 shows the confusion matrix for REL and ORG traits, explaining this weaker performance. In LAILA, essays with a REL score of zero automatically receive zeros for all other traits, and P1 contains a relatively high proportion of such essays (16.4%). Our analysis shows that although the model captures this dependency (assigning zero to other traits when REL is predicted as zero), it struggles to predict REL accurately. For example, 130 essays with a true REL of 0 were predicted as 2, causing excessive over-predictions in the other traits. Specifically, 112 essays of them received a score above 1 for ORG, contributing to the lower overall performance.

Another interesting observation highlights that across all prompts, only 8.1% of essays received the same scores as the gold labels across all traits. However, 41.6% of essays are predicted within a one-point difference for all traits, suggesting that while exact matches are relatively limited, the model's predictions are pretty close to the true scores, with an average RMSE of 1.123 across all traits.

5.2 Case Study

To further investigate the REL issue, we examine two essays: one correctly predicted and one misclassified. Figure 3 presents two essays from P1 with a true REL score of 1. The model correctly predicts the first essay (left), which exhibits a clear topical mismatch with the prompt, focusing on obesity and general health-related issues rather than teenagers' declining interest in sports participation. However, the second essay (right) is overpredicted as 2. While it contains several prompt-related keywords (e.g., teenagers, sports, training, and competitions), it provides limited discussion of the prompt's main question and lacks sufficient explanatory depth. This example suggests that the model relies heavily on lexical overlap when assessing REL. As a result, essays containing prompt-related keywords may receive inflated scores despite only partially addressing the prompt.

5.3 ST vs. MT

```
سيتم إعطاؤك مقال كتبه الطالب ردًا على الموضوع المعطى. مهمتك هي تقييم سمة التقييم الكلي
ضمن النطاق (٠-٣٢). أخرج النتيجة بصيغة جصو فقط، دون أي شرح أو نص إضافي. يجب
أن تكون درجة السمة رقمًا صحيحًا واحدًا داخل النطاق المحدد.
- - -
الموضوع: نص الموضوع الذي طُلب من الطالب الكتابة عنه
المقال: نص المقال المكتوب من قِبل الطالب ردًا على الموضوع
الدرجات: { ألتقييم الكلي: 0-32 }
- - -
الموضوع: {prompt}
المقال: {essay}
الدرجات:
```

Figure 6: The Arabic prompt used for the ST prediction. English translation is provided in Figure 7.

We next investigate whether fine-tuning a separate model for each trait, referred to as **Single-Trait (ST)**, yields better performance than MT. For this experiment, we use FANAR. Under the ST setting, a dedicated model is fine-tuned to predict a single trait score for a given essay, resulting in one model per trait (the prompt is depicted in Figure 6).

The results on the development set are shown in Figure 5. Interestingly, the average performance of ST and MT models is very close. REL is the only trait with a noticeable difference, where ST outperforms MT by about 2 points. This may be because REL is scored on a smaller range (0-2) than the other traits (0-5), so MT training may dilute its gradient signal. Despite this small difference, the MT

You will be given an essay written by a student in response to a prompt. Your task is to evaluate the holistic trait within the range (0, 32).
Output the result in JSON format only, without any additional explanation or text. The score should be an integer within the specified range.
—
Prompt: The prompt that the student was asked to write about
Essay: The essay written by the student in response to the prompt
Score: {'Holistic': (0, 32)}
—
Prompt: {prompt}
Essay: {essay}
Score:

Figure 7: English translation of the ST prediction prompt.

model is more efficient, predicting all eight scores with one model, while the ST approach requires eight separate models. Specifically, the inference time of the ST approach is 0.35 seconds per essay for all traits, compared to 0.07 seconds for the MT model. Training the ST models for all traits over 10 epochs requires approximately 678 hours, compared to about 102 hours for the MT model.

5.4 Autoregressive Copying

We also observed a tendency of the MT model to repeat scores across traits. On the test set, 45.7% of essays have identical scores for all traits (excluding REL and HOL), compared to 15.2% in the gold labels. This raises a concern: "is the MT model biased by its autoregressive generation process?" To investigate this concern, we compare MT and ST predictions on the development set. For each consecutive trait pair $(i, i+1)$, we focus on cases where the *gold scores differ* ($y_i \neq y_{i+1}$) and define two probabilities:

$$P_{\text{SEQ}} = p(\hat{y}^M_{i+1} = \hat{y}^M_i \text{ and } \hat{y}^S_{i+1} \neq \hat{y}^S_i \mid y_{i+1} \neq y_i)$$

$$P_{\text{IND}} = p(\hat{y}^M_{i+1} = \hat{y}^M_i \text{ and } \hat{y}^S_{i+1} = \hat{y}^S_i \mid y_{i+1} \neq y_i)$$

where $\hat{y}^S$, $\hat{y}^M$ are the predicted scores by ST and MT models, respectively. P_{SEQ} measures the probability that MT introduces copying between trait scores that ST distinguishes, suggesting *sequential* influence, while P_{IND} captures cases where the MT model's repetition aligns with the ST approach's *independent* assessment.

As shown in Table 3, P_{IND} is significantly higher than P_{SEQ} across all pairs (averaging 0.526 vs. 0.215). This indicates that repeated scores in MT are not primarily due to autoregressive propagation; rather, the MT model assigns these scores deliberately, reflecting a consistent evaluation process similar to the independent ST predictions.

trait$_i$	**trait**$_{i+1}$	P_{SEQ}	P_{IND}
ORG	VOC	0.152	0.377
VOC	STY	0.217	0.476
STY	DEV	0.243	0.598
DEV	MEC	0.249	0.432
MEC	GRA	0.214	0.750
AVE		0.215	0.526

Table 3: Consecutive trait copying.

6 Conclusion and Future Work

This study evaluated the effectiveness of fine-tuning relatively small Arabic-centric LLMs, namely ALLAM 7B, JAIS 8B, and FANAR 9B, for cross-prompt Arabic automated essay scoring. The results show that LLM^{ft} consistently outperforms strong baselines, including much larger proprietary models such as GPT-5. Specifically, the best-performing model, FANAR^{ft}, surpasses the strongest baseline on LAILA dataset by 8 points. Moreover, we evaluate LLM^{ft} in a cross-dataset setting (training on LAILA and testing on ZAEBUQ), where FANAR^{ft} achieves the strongest performance among all models. Further analysis of FANAR^{ft} reveals several performance patterns, highlighting both its strengths and weaknesses. In future work, we plan to explore alternative prompting strategies, incorporate rubric-based information, and investigate model distillation to improve efficiency while maintaining strong performance.

Acknowledgments

This work was made possible by NPRP grant NPRP14S-0402-210127 from the Qatar Research Development and Innovation (QRDI) Council. The work of Sohaila Eltanbouly was supported by GSRA grant# GSRA12-L-0413-250111, and the work of Tamer Elsayed was also supported by the awarded research fund QUEX-CENG-GOOGLE-2025-1 from Google. The statements made herein are solely the responsibility of the authors.

7 Limitations

While our results are substantial, this study has several limitations. First, we evaluated only two prompt formulations and did not explore a broader

range of prompting strategies that may further improve performance. Second, our benchmark focuses on fine-tuning relatively small Arabic-centric LLMs and does not investigate larger or multilingual LLMs. Future work could examine whether the observed gains transfer to these settings and scale with model size and language diversity. Finally, fine-tuning LLMs requires non-trivial computational resources, which may limit accessibility in some settings.

References

Salam Albatarni, May Bashendy, Sohaila Eltanbouly, and Tamer Elsayed. 2026. MAPLE: A meta-learning framework for cross-prompt essay scoring. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 19948–19961, San Diego, California, United States. Association for Computational Linguistics.

Salam Albatarni and Tamer Elsayed. 2026. Quadratic weighted kappa is not enough for evaluating automated essay scoring models. In *Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026)*, pages 4447–4456, Palma, Mallorca, Spain. European Language Resources Association (ELRA).

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.

Mohamed Anwar, Abdelhakim Freihat, George Ibrahim, Mostafa Awad, Abdelrahman Atef Mohamed Ali Sadallah, Gurpreet Gosal, Gokul Ramakrishnan, Sarath Chandran, Biswajit Mishra, Rituraj Joshi, Ahmed Frikha, Etienne Goffinet, Abhishek Maiti, Ali El Filali, Sarah Al Barri, Samujjwal Ghosh, Rahul Pal, Parvez Mullah, Awantika Shukla, and 41 others. 2025. Jais 2: A family of Arabic-centric open large language models. Technical report, IFM.

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*.

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: Shared Tasks*, pages 966–976, Suzhou, China. Association for Computational Linguistics.

May Bashendy, Walid Massoud, Sohaila Eltanbouly, Salam Albatarni, Marwan Sayed, Abrar Abir, Houda Bouamor, and Tamer Elsayed. 2026. LAILA: A large trait-based dataset for Arabic automated essay scoring. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3074–3091, Rabat, Morocco. Association for Computational Linguistics.

Po-Kai Chen, Bo-Wei Tsai, Shao Kuan Wei, Chien-Yao Wang, Jia-Ching Wang, and Yi-Ting Huang. 2025. Mixture of ordered scoring experts for cross-prompt essay trait scoring. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18071–18084, Vienna, Austria. Association for Computational Linguistics.

Jacob Cohen. 1968. Weighted kappa: nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, 70(4):213.

Hoor Tamer Elbahnasawi, Marwan Sayed, Sohaila Eltanbouly, Fatima Zahra Brahamia, and Tamer Elsayed. 2026. Qayyem: A real-time platform for scoring proficiency of Arabic essays. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 536–545, San Diego, California, United States. Association for Computational Linguistics.

Sohaila Eltanbouly, Salam Albatarni, and Tamer Elsayed. 2025. TRATES: Trait-specific rubric-assisted cross-prompt essay scoring. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20528–20543, Vienna, Austria. Association for Computational Linguistics.

Sohaila Eltanbouly, Marwan Sayed, and Tamer Elsayed. 2026. Is one dataset enough for evaluation? studying generalizability of automated essay scoring models. In *Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026)*, pages 431–440, Palma, Mallorca, Spain. European Language Resources Association (ELRA).

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? *Computers and Education: Artificial Intelligence*, page 100449.

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.

Jinhee Jang, Ayoung Moon, Minkyoung Jung, Young-Bin Kim, and Seung Jin Lee. 2025. LLM agents at the roundtable: A multi-perspective and dialectical reasoning framework for essay scoring. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 19674–19687, Suzhou, China. Association for Computational Linguistics.

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*.

Sanwoo Lee, Yida Cai, Desong Meng, Ziyang Wang, and Yunfang Wu. 2024. Unleashing large language models' proficiency in zero-shot essay scoring. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 181–198, Miami, Florida, USA. Association for Computational Linguistics.

Shengjie Li and Vincent Ng. 2024. Conundrums in cross-prompt automated essay scoring: Making sense of the state of the art. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7661–7681, Bangkok, Thailand. Association for Computational Linguistics.

Yingying Liu, Huilei Qi, and Xiaofei Lu. 2025. Enhancing gpt-based automated essay scoring: the impact of fine-tuning and linguistic complexity measures. *Computer Assisted Language Learning*, pages 1–20.

Somaia Mahmoud, Emad Nabil, and Marwan Torki. 2024. Automatic scoring of arabic essays: A parameter-efficient approach for grammatical assessment. *IEEE Access*.

Salim Al Mandhari, Hieu Pham Dinh, Mo El-Haj, and Paul Rayson. 2026. Structured prompting for arabic essay proficiency: A trait-centric evaluation approach. In *Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026)*, pages 10532–10544, Palma, Mallorca, Spain. European Language Resources Association (ELRA).

Watheq Ahmad Mansour, Salam Albatarni, Sohaila Eltanbouly, and Tamer Elsayed. 2024. Can large language models automatically score proficiency of written essays? In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2777–2786, Torino, Italia. ELRA and ICCL.

Yossra Noureldien, Abdelrazig Mohamed, and Farah Attallah. 2025. Zero-shot and fine-tuned evaluation of generative LLMs for Arabic word sense disambiguation. In *Proceedings of The Third Arabic Natural Language Processing Conference*, pages 298–305, Suzhou, China. Association for Computational Linguistics.

Chatrine Qwaider, Kirill Chirkunov, Bashar Alhafni, Nizar Habash, and Ted Briscoe. 2025. Evaluating prompt relevance in Arabic automatic essay scoring: Insights from synthetic and real-world data. In *Proceedings of The Third Arabic Natural Language Processing Conference*, pages 162–178, Suzhou, China. Association for Computational Linguistics.

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 Third Arabic Natural Language Processing Conference*, pages 231–245, Suzhou, China. Association for Computational Linguistics.

Maja Stahl, Leon Biermann, Andreas Nehring, and Henning Wachsmuth. 2024. Exploring LLM prompting strategies for joint essay scoring and feedback generation. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 283–298, Mexico City, Mexico. 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, and 1 others. 2025. Fanar: An arabic-centric multimodal generative ai platform. *arXiv preprint arXiv:2501.13944*.

Changrong Xiao, Wenxing Ma, Qingping Song, Sean Xin Xu, Kunpeng Zhang, Yufang Wang, and Qi Fu. 2025. Human-ai collaborative essay scoring: A dual-process framework with llms. In *Proceedings of the 15th International Learning Analytics and Knowledge Conference*, LAK '25, page 293–305, New York, NY, USA. Association for Computing Machinery.

Lui Yoshida. 2025. Are the reasoning models good at automated essay scoring? In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 8388–8394, Suzhou, China. Association for Computational Linguistics.

A Per-Prompt Results

We present the per-prompt results on LAILA dataset in Table 4. Prompt P1 shows the lowest performance, with an average QWK of 0.488, compared to the other prompts, whose performance ranges from 0.733 to 0.805.

LLM	Prompt	REL	ORG	VOC	STY	DEV	MEC	GRA	HOL	AVE
ALLAM^{ft}	P1	0.373	0.568	0.547	0.553	0.531	0.504	0.514	0.553	0.518
	P2	0.573	0.757	0.730	0.733	0.699	0.658	0.683	0.756	0.699
	P3	0.710	0.876	0.782	0.798	0.755	0.786	0.812	0.868	0.798
	P4	0.690	0.815	0.759	0.769	0.731	0.764	0.755	0.813	0.762
	P5	0.625	0.774	0.691	0.729	0.728	0.679	0.713	0.775	0.714
	P6	0.683	0.782	0.762	0.758	0.752	0.732	0.757	0.814	0.755
	P7	0.699	0.811	0.792	0.755	0.712	0.742	0.756	0.838	0.763
	P8	0.757	0.817	0.835	0.812	0.740	0.772	0.764	0.854	0.794
JAIS^{ft}	P1	0.235	0.524	0.502	0.525	0.458	0.471	0.473	0.500	0.461
	P2	0.334	0.731	0.709	0.745	0.670	0.657	0.675	0.741	0.658
	P3	0.654	0.830	0.791	0.751	0.709	0.748	0.787	0.826	0.762
	P4	0.599	0.814	0.739	0.771	0.722	0.751	0.747	0.800	0.743
	P5	0.614	0.761	0.683	0.729	0.732	0.677	0.714	0.771	0.710
	P6	0.625	0.761	0.756	0.742	0.722	0.706	0.741	0.799	0.732
	P7	0.685	0.802	0.808	0.739	0.742	0.711	0.731	0.829	0.756
	P8	0.521	0.740	0.737	0.716	0.648	0.647	0.662	0.738	0.676
FANAR^{ft}	P1	0.303	0.548	0.505	0.545	0.502	0.482	0.494	0.522	0.488
	P2	0.731	0.815	0.793	0.786	0.772	0.740	0.756	0.835	0.779
	P3	0.697	0.885	0.772	0.823	0.762	0.796	0.813	0.874	0.803
	P4	0.619	0.837	0.742	0.788	0.736	0.757	0.750	0.811	0.755
	P5	0.647	0.784	0.724	0.734	0.749	0.697	0.734	0.793	0.733
	P6	0.701	0.783	0.756	0.771	0.733	0.742	0.761	0.820	0.759
	P7	0.751	0.829	0.808	0.752	0.718	0.737	0.746	0.847	0.774
	P8	0.799	0.831	0.842	0.820	0.772	0.759	0.753	0.866	0.805

Table 4: Per prompt and per trait QWK performance on LAILA test set

Align-Extract-Score: Graded Relevance of Arabic Essays

Marwan Sayed[1], **Salam Albatarni**[1], **Reem Suwaileh**[2], **Tamer Elsayed**[1]

[1]Computer Science and Engineering Department, Qatar University
[2]Institution of Public Administration, Civil Service and Government Development Bureau, Qatar

{me2104862, sa1800633, telsayed}@qu.edu.qa,
rsuwaileh@cgb.gov.qa

Abstract

Relevance is a central component of Automated Essay Scoring (AES), as it measures how well a student essay addresses a given writing prompt. In this work, we propose AESER, a novel approach to address the problem of graded relevance scoring in Arabic cross-prompt AES. AESER adapts the given prompt more towards the writing space of essays and computes token-level prompt-essay similarity matrices from contextual embeddings. These matrices are processed by a convolutional neural network to predict the relevance score. Experiments on a large Arabic AES dataset show that AESER outperforms strong baselines, establishing a new state-of-the-art performance.

1 Introduction

Automated Essay Scoring (AES) assigns quality scores to essays and offers clear benefits for large-scale assessments where manual grading is costly. A key component of AES is trait-level scoring, which assesses specific aspects of writing, e.g., organization, style, or relevance, to provide more targeted feedback.

The relevance trait, commonly referred to as *prompt adherence* (Persing and Ng, 2014; Mathias and Bhattacharyya, 2018), measures how well an essay addresses the given writing prompt. The prompt defines both the writing task and the topic that the student is expected to address. Relevance is particularly important because it reflects the student's ability to correctly interpret the topic and maintain focus throughout the essay. It also determines the validity of other traits; an off-topic essay cannot be considered high quality, regardless of its structure or fluency. Considering the importance of this trait, in this work we focus on predicting *graded* (i.e., multi-level) *relevance* of essays given a writing prompt. We address this problem in the *cross-prompt* setting, where models are evaluated on writing prompts that were *unseen* during training, which reflects realistic educational scenarios where instructors regularly introduce new writing tasks for which no annotated essays are available.

Most of the existing trait-level AES literature focuses on simultaneous scoring across multiple traits, including relevance, but with no special attention given to the particular notion or nature of relevance (Ridley et al., 2021; Do et al., 2023; Chen and Li, 2023; Eltanbouly et al., 2025). Only a few studies directly target the relevance trait (Albatarni et al., 2024). However, the field remains largely English-centric, and *Arabic* relevance scoring is still scarce, typically limited to prompt-specific or binary formulations (Bashendy et al., 2024; Machhout and Zribi, 2024; Qwaider et al., 2025). We address this gap by targeting *graded*, *cross-prompt* relevance scoring for Arabic essays.

To this end, we propose AESER, an **A**lign-**E**xtract-**S**core approach for **E**ssay **R**elevance.[1] Our approach *aligns* the given writing prompt to the scope of a written essay by using a large language model (LLM) to "rewrite" it into one or more essays that reflect the prompt's topic at different relevance levels. An encoder then *extracts* a token-wise similarity matrix between the transformed prompt and the essay, explicitly capturing fine-grained prompt–essay interactions.

Finally, inspired by Pang et al. (2016), we *score* the essay by treating the similarity matrix as a grayscale image passed to a convolutional neural network (CNN), framing similarity patterns in a visual representation for graded relevance scoring. Experiments on LAILA, a recently released large Arabic AES dataset (Bashendy et al., 2026), show that AESER establishes a new state-of-the-art (SOTA) over strong baselines.

Overall, our contributions are three-fold:

1. We propose AESER, a novel interaction-

[1]Pronounced as "ACER" with a different spelling.

Proceedings of The Fourth Arabic Natural Language Processing Conference (ArabicNLP 2026), pages 22–32
October 28-29, 2026 ©2026 Association for Computational Linguistics

based relevance scoring framework that uses token-level similarity matrices to capture fine-grained prompt–essay relationships.
2. We show that AESER achieves a new SOTA on LAILA dataset, surpassing all baselines by substantial margins and remaining robust to the choice of LLM.
3. We release our code[2] for reproducibility and to support future work on AES relevance, and Arabic in particular.

2 Related Work

Our work brings together two lines of research, trait-level cross-prompt AES with the focus on relevance trait modeling, and Arabic AES.

Trait-level Cross-Prompt AES Early cross-prompt work focused on scoring essays *holistically* with shared neural architectures (Jin et al., 2018; Li et al., 2020; Ridley et al., 2020), after which Ridley et al. (2021) introduced *trait-level* cross-prompt scoring, combining shared low-level layers for generalization with prompt-specific high-level layers. Later models kept this hierarchical design and added prompt-aware attention (Do et al., 2023), contrastive prompt-mapping (Chen and Li, 2023) and meta-learning training (Chen and Li, 2024). Eltanbouly et al. (2025) combined hand-crafted features with trait-specific LLM-derived features. Throughout this line of work, relevance is scored jointly with other traits and is not modeled explicitly, despite its importance for essay quality.

Relevance was incorporated in earlier work to support holistic scoring through classic IR signals such as bag-of-words similarity (Rei and Cummins, 2016) and query expansion (Cummins et al., 2016). Albatarni et al. (2024) addressed graded relevance in English essays by embedding essays with an unsupervised dense encoder and scoring each by its nearest relevance centroid (1-NN).

Arabic AES The large majority of the work was English-centric, motivated by the availability of English datasets (Li and Ng, 2024), with little work on other languages including Arabic (Bashendy et al., 2024). Indeed, Arabic AES faces major challenges due to data scarcity. Nevertheless, recent efforts have explored Arabic trait-level scoring. Approaches range from hand-crafted feature-based models (Bashendy et al., 2024; Sayed et al., 2025), to multitask AraBERT fine-tuning with separate regression heads (Bashendy et al., 2025), few-shot LLM prompting with GPT4o (Almarwani et al., 2025) or 10-shot chain-of-thought prompting with GPT-4.1 (Alnajjar et al., 2025), and more recently, Albatarni et al. (2026) proposed MAPLE for Arabic trait scoring using a meta-learning approach. Though performance on relevance in particular remains limited compared to other traits. Studies specifically targeting *Arabic relevance trait scoring* are scarce: Machhout and Zribi (2024) fine-tuned AraBERT in a prompt-specific setting,[3] while Qwaider et al. (2025) compared multiple methods on real and synthetic data, finding 2-shot prompting with Falcon and Qwen most effective for *binary* relevance classification rather than graded scoring.

Despite the progress in Arabic AES, relevance scoring has received limited attention and has been studied primarily in prompt-specific or binary settings. We address these gaps by focusing on graded relevance scoring in the cross-prompt setting.

3 Methodology

Relevance scoring task in the context of automated essay scoring aims to assess the topical alignment of a student-written essay with a given writing prompt. Formally, for a prompt p and an essay e, a relevance scoring model maps (p, e) to a score $s \in \{0, 1, \ldots, K\}$.

In this section, we present our proposed model, AESER, illustrated in Figure 1. Existing AES relevance models typically use the prompt in its original form alongside the essay, without reformulating it to better align with student responses or modeling fine-grained token-level interactions. In contrast, our method adapts query rewriting from information retrieval to AES and leverages explicit interaction modeling for graded relevance scoring.

AESER consists of three main phases: *Prompt Representation* (subsection 3.1), where the writing prompt is transformed to better align with student essays; *Feature Extraction* (subsection 3.2), where a token-level similarity matrix captures prompt–essay interactions; and *Graded Relevance Scoring* (subsection 3.3), where these signals are aggregated to capture hierarchical patterns from token-level alignments to higher-level structures, producing a normalized relevance score.

[2] https://github.com/Maroibo/AESER

[3] Where models are trained and evaluated on essays written for the same prompt.

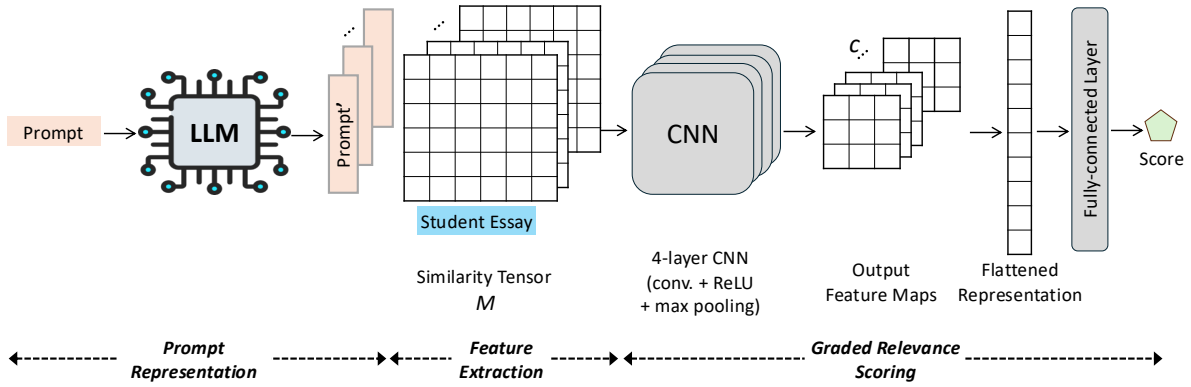


Figure 1: Overview of AESER including prompt representation, feature extraction, and graded relevance scoring.

3.1 Prompt Representation

The writing prompt typically includes *both* the *topic*, which defines the subject the student must address, and the *instructions*, which specify the essay type, word limit, and general structural requirements. [4]

While the topic part defines the scope and the key aspects that the essay should address, the instructions part can *distract* the model's ability to judge relevance by causing topical drift. Therefore, the topic part of the prompt potentially provides a more reliable basis for relevance assessment, motivating the use of an *instruction-free* prompt that preserves only the core subject.

Nevertheless, an instruction-free prompt remains structurally and length-wise different from the student essays being evaluated, as it represents a concise topic description rather than a complete essay. To address that, we consider prompt representations that resemble student writing. Rewriting the prompt as a *pseudo essay* reflects that relevance manifests in essay-level structure and content, enabling the model to compare texts of similar form. Extending this idea, *rubric guided pseudo essays* incorporate the grading rubric by generating one essay per relevance level, each capturing its expected characteristics and enabling comparison with graded reference representations.

To that end, we examine four different prompt representations:

1. **Original Prompt** ($\boldsymbol{p}_{ORIG}$): The prompt is used without any modification, as a baseline representation.

2. **Instruction-free Prompt** ($\boldsymbol{p}_{TOPIC}$): The prompt is transformed to retain the topic component, excluding instructional content.

3. **Pseudo Essay** ($\boldsymbol{p}_{ESSAY}$): The original prompt is transformed into an essay that addresses the given topic.

4. **Rubric-based Pseudo Essays** ($\boldsymbol{p}_{RUBRIC}$): The prompt is represented as multiple essays, each reflecting a distinct relevance level defined by the rubric.

Each representation strategy produces T text sequences per prompt. For $\boldsymbol{p}_{ORIG}$, $\boldsymbol{p}_{TOPIC}$, and $\boldsymbol{p}_{ESSAY}$, $T = 1$. For $\boldsymbol{p}_{RUBRIC}$, one sequence is generated per relevance level, resulting in $T = K$.

We note that in this work, we leveraged the powerful capabilities of LLMs to generate the transformed prompts. The exact instructions prompted to the LLM to generate $\boldsymbol{p}_{TOPIC}$, $\boldsymbol{p}_{ESSAY}$, and $\boldsymbol{p}_{RUBRIC}$ prompt representations are depicted in Figure 2.

3.2 Feature Extraction

As relevance depends on how well essay content adheres to the prompt, the feature extraction stage computes a *token-level* similarity tensor to explicitly model prompt–essay alignment. An off-the-shelf encoder E generates contextualized token embeddings for each generated text sequence $p'^{(t)}$ and the essay e:

$$\vec{u}_i^{(x)} = E\left(p'^{(t)}\right)_i, \qquad \vec{v}_j = E(e)_j, \qquad (1)$$

where $\vec{u}_i^{(t)}$ is the embedding of the i-th token of the t-th generated text sequence, and $\vec{v}_j$ is the embedding of the j-th essay token.

As illustrated in Figure 1, a similarity tensor $M \in \mathbb{R}^{T \times |p'| \times |e|}$ is then constructed as:

[4] An example of this distinction is shown in Figure 5, where the topic is marked in bold.

English translation. You are an expert in analyzing Arabic texts. Your task is to extract only the main writing topic from the following text, removing all writing instructions and directives. Rephrase the topic clearly and directly. Very important: do not add any comments, questions, or extra explanations. Only the topic.

Text:
(the writing prompt)

The topic only (without any extra comments):

(a) Prompt for the topic representation $\boldsymbol{p}_{TOPIC}$. The model is asked to drop the writing instructions and return only the essay topic.

English translation. You are an expert essay writer. Write a coherent, well-organized essay in Arabic that addresses the following writing topic:

(the writing prompt)

The essay:

(b) Prompt for the essay representation $\boldsymbol{p}_{ESSAY}$. The model writes a full essay on the topic, which we use as a hypothetical-document expansion of the prompt.

English translation. You are an expert essay writer. Write a coherent, well-organized essay in Arabic that addresses the following writing topic:

(the writing prompt)

Relevance evaluation criteria:
(the rubric describing relevance level ℓ)

The essay:

(c) Prompt for the rubric representation $\boldsymbol{p}_{RUBRIC}$. We run this prompt once per relevance level ℓ: the rubric placeholder is filled with the descriptor of level ℓ so the generated essay is written to match that exact level. With K levels this produces K essays per writing prompt, one targeted at each level.

Figure 2: English translated versions of the LLM prompts we passed to generate $\boldsymbol{p}_{TOPIC}$ (Figure 2a), $\boldsymbol{p}_{ESSAY}$ (Figure 2b), and $\boldsymbol{p}_{RUBRIC}$ (Figure 2c). The original prompt is depicted in Figure 3

$$M_{tij} = \phi\left(\vec{u}_i^{(t)}, \vec{v}_j\right), \tag{2}$$

where ϕ denotes a vector similarity function (e.g., cosine similarity or dot product).

3.3 Graded Relevance Scoring

The similarity tensor M can be viewed as a stack of 2D interaction signals capturing token-level correspondences between the prompt representation and the essay. Interpreting this tensor as structured spatial input enables a CNN to model local alignment patterns and progressively aggregate them into higher-level interaction features.

Motivated by this intuition, we design a 4-layer convolutional architecture in which kernel sizes increase across layers, and the number of feature maps expands progressively.

Let $L(=4)$ denote the total number of convolutional layers. We denote by $Z^{(\ell)}$ the output of convolutional layer ℓ, with $Z^{(0)}$ as the input to the first layer. Since $|p'|$ and $|e|$ vary across instances, adaptive max pooling $\mathcal{P}$ is applied to M to obtain a fixed spatial resolution of $S \times S$, resulting in $Z^{(0)} = \mathcal{P}(M) \in \mathbb{R}^{T \times S \times S}$.

The output of each layer, $Z^{(\ell)}$, is computed as:

$$Z^{(\ell)} = \mathcal{P}\left(\sigma\left(W^{(\ell)} * Z^{(\ell-1)} + b^{(\ell)}\right)\right) \tag{3}$$

where $*$ denotes the 2D convolution operator, $W^{(\ell)}$ and $b^{(\ell)}$ are the learnable convolution kernels and biases of layer ℓ, and $\sigma(\cdot)$ is the ReLU activation function. A progressive reduction scheme is employed so that earlier layers retain fine-grained local alignment patterns, whereas deeper layers operate over coarser representations, thereby capturing broader and more global matching structures.

After the last convolutional and pooling layers, the tensor $Z^{(L)}$ is flattened into a vector $\vec{z}$, as shown in the final stage of Figure 1, representing dense relevance features learned by the CNN. At this stage, hand-crafted features can be incorporated by concatenating them with $\vec{z}$. The resulting vector is then passed through a fully connected layer followed by a regression head to predict a continuous relevance score. The score is clipped to the interval $[0, 1]$. During training, the original discrete relevance labels are linearly rescaled to $[0, 1]$, and the model is optimized using mean squared error (MSE) between the predicted score and the scaled target. At inference, the predicted score is mapped back to the original discrete label range to obtain the final integer relevance score.

4 Experimental Setup

This section describes our experimental setup, covering the implementation and hyperparameter tuning of AESER, the dataset, the LLMs used to generate the prompt representations, the evaluation measure, and the baselines.

Prompt Representation Setup We consider three LLMs for generating prompt representations:

- Fanar-27B (hereafter Fanar) is a medium-size Arabic-centric LLM accessed via API.[5]
- Gemma 4 31B (hereafter Gemma) (Google DeepMind, 2026) is a medium-size multilin-

[5] https://api.fanar.qa/docs

أنت خبير في تحليل النصوص العربية. مهمتك هي استخراج موضوع الكتابة الأساسي فقط من النص التالي مع إزالة جميع التعليمات الكتابية والتوجيهات. أعد صياغة الموضوع بشكل واضح ومباشر. مهم جداً: لا تضع أي تعليقات أو أسئلة أو شروحات إضافية. فقط الموضوع.
النص:
(نص الموضوع)
الموضوع فقط (بدون أي تعليقات إضافية):

(a) LLM prompt for the topic representation $\boldsymbol{p}_{TOPIC}$.

أنت كاتب مقالات خبير. اكتب مقالاً متماسكاً ومنظماً باللغة العربية يتناول موضوع الكتابة التالي:
(نص الموضوع)
المقال:

(b) LLM prompt for the essay representation $\boldsymbol{p}_{ESSAY}$.

أنت كاتب مقالات خبير.
اكتب مقالاً متماسكاً ومنظماً باللغة العربية يتناول موضوع الكتابة التالي:
(نص الموضوع)
معايير التقييم للصلة بالموضوع:
(معيار التصحيح للمستوى المستهدف)
المقال:

(c) LLM prompt for the rubric representation $\boldsymbol{p}_{RUBRIC}$.

Figure 3: LLM prompts for generating different representation for $\boldsymbol{p}_{ORIG}$. English translation for each prompt is depicted in Figure 2.

gual open-weights LLM, served with vLLM[6] for faster inference.

- The closed-source GPT-4o (OpenAI, 2024) is the highest-ranked OpenAI model on the SILMA Arabic LLM Leaderboard at the time of writing.[7]

For all LLMs, the temperature was set to 0.7 and the maximum number of generated tokens (`max_tokens`) to 1000, with all other hyperparameters left at their default values. The LLM prompts used to generate $\boldsymbol{p}_{TOPIC}$, $\boldsymbol{p}_{ESSAY}$, and $\boldsymbol{p}_{RUBRIC}$ are shown in Figure 3a, Figure 3b, and Figure 3c, respectively. For *LLM-based* representations, we further generate N *instances* for each text sequence, so that for a prompt representation producing T text sequences, a total of $N \times T$ text sequences are generated. These generated sequences are then added as separate input channels to the CNN.

Implementation, Training, and Hyperparameter Tuning Experiments were conducted on an NVIDIA RTX A6000 GPU. Models were trained for up to 150 epochs using AdamW with a batch size of 32 and a learning rate of $\eta = 5 \times 10^{-4}$. Early stopping was applied using a patience of 15 epochs based on the development set performance. Token embeddings were obtained using Qwen3-Embedding-0.6B.[8] We selected this model because it ranks first among models under 1B parameters on the MTEB leaderboard.[9] For the hand-crafted features, we used the 816 features introduced by Sayed et al. (2025) for Arabic AES, covering surface, readability, lexical, semantic, and syntactic aspects.

The input similarity matrices were pooled to a fixed spatial resolution of $S = 256$. The CNN backbone consisted of four convolutional layers, with $c_{l_1} = 32$ denoting the number of output maps in the first layer. Subsequent layers followed a progressive doubling scheme: $2c_{l_1}$, $4c_{l_1}$, and $8c_{l_1}$. Let $r_{\ell 1} = 5$ denote the kernel size of the first layer. The kernel sizes of subsequent layers followed an incremental scheme: $r_{\ell 1}+2$, $r_{\ell 1}+4$, and $r_{\ell 1}+6$. Adaptive max pooling was applied after each convolutional layer, with output sizes following a progressive reduction scheme: $o_{l_1} = 256$, $o_{l_2} = o_{l_1}/4$, and $o_{l_{3-4}} = o_{l_1}/16$. The hidden dimension of the fully connected layer denoted by h was initially fixed to 128.

AESER consists of several components that can be tuned, including the prompt representation, the similarity function, the number of retrieved references (N), the inclusion of hand-crafted features, the loss function, and the choice of the generation LLM. To study these components in isolation, we first fix the generation LLM to Fanar and search for the best configuration based on the development set performance.

Specifically, we perform sequential tuning over the components in a fixed order: (1) prompt representation, (2) similarity function, (3) number of generated references (N), (4) inclusion of hand-crafted features, and (5) loss function. For each component, we evaluate all candidate choices while keeping previously selected components fixed, and retain the best-performing option before moving to the next stage. This procedure yields a single optimized configuration, which is then used across all generation LLMs in the subsequent experiments.

[6] https://github.com/vllm-project/vllm
[7] https://huggingface.co/spaces/silma-ai/Arabic-LLM-Leaderboard
[8] https://huggingface.co/Qwen/Qwen3-Embedding-0.6B
[9] https://huggingface.co/spaces/mteb/leaderboard

We then tune the hyperparameters $S, r_{\ell 1}, c_{l_1}, o_{l_1}$, h, and η of this configuration separately for each generation LLM on the development set. Specifically, $S \in \{128, 256, 512\}$, $r_{\ell 1} \in \{3, 5, 7, 9\}$, $c_{l_1} \in \{8, 16, 32, 64, 128\}$, $o_{l_1} \in \{128, 256, 512\}$, $h \in \{64, 128, 256\}$, and $\eta \in \{1e^{-5}, 1e^{-4}, 5e^{-4}\}$. Hyperparameters were optimized via Bayesian optimization using the Tree-structured Parzen Estimator (TPE) algorithm (Bergstra et al., 2011), implemented with Optuna's `TPESampler`.[10] We conducted 100 trials with 5 startup trials and a fixed random seed. We then evaluate each LLM's best-tuned model on the test set.

Because most essays in LAILA are fully relevant, we compare class-weighted MSE (WMSE) and focal MSE (FMSE) with standard MSE to test whether reweighting training errors improves relevance scoring. For WMSE, weights were computed using the training split by dividing the number of essays in the most frequent relevance class by the number of essays in each class. Each essay's squared error was multiplied by its class weight before averaging. FMSE places greater relative emphasis on larger prediction errors by averaging absolute prediction errors raised to $2+\gamma$, with focusing parameter $\gamma \in \{0.5, 1, 2\}$.

Dataset LAILA dataset is the first large-scale Arabic AES dataset with annotations for seven writing traits including relevance, which is scored on a 0–2 scale (Bashendy et al., 2026). It contains 7,859 essays in response to eight different prompts spanning two types: persuasive and explanatory. The relevance labels are heavily imbalanced toward fully relevant essays: 86% are scored 2 (fully relevant), 6% are scored 1 (partially relevant), and 8% are scored 0 (not relevant). We use the official cross-prompt data split shared by LAILA authors.[11]

Evaluation Measure We evaluate our approach using Quadratic Weighted Kappa (QWK) (Cohen, 1968), a standard metric in AES. QWK measures agreement between human and system scores while accounting for the severity of disagreements.

Baselines We adopt SOTA models for relevance cross-prompt evaluation: (1) **AraBERT4GR**: Originally proposed for English essay relevance grading (Albatarni et al., 2024). We adapted it to Arabic by replacing Contriever with AraBERT[12] and tuned all hyperparameters on our training data. (2) **MOOSE**: An adaptation of an English cross-prompt SOTA model based on mixture-of-experts architecture (Chen et al., 2025; Bashendy et al., 2026). (3) **Fanar**^{fs} (5-shots): Second-best model for relevance on LAILA (Bashendy et al., 2026). (4) **Qwen**^{fs} and (5) **Falcon**^{fs} (3-shots): Top-performing models on QAES and ZAEBUC by Qwaider et al. (2025). We reformulated their binary setup into a three-class task aligned with LAILA relevance levels.[13]

5 Experimental Evaluation

This section presents and discusses our results in two parts. We first report the main results that answer our research questions in subsection 5.1. Then, we analyze the model's performance more closely through an error analysis and a case study in subsection 5.2.

5.1 Main Results

We address three research questions: (**RQ**1) What is the optimal configuration for AESER? We examine different prompt representations, similarity functions, the number of generated instances (N), the inclusion of additional features, and the loss function, (**RQ**2) Which generation LLM performs best for AESER?, and (**RQ**3) How does AESER perform compared to SOTA baselines?

RQ1: AESER Configuration We first examine the different *prompt representations*, depicted in the first section of Table 1. Among them, $\boldsymbol{p}_{TOPIC}$ performed the worst, scoring nearly 7 points below $\boldsymbol{p}_{ORIG}$, indicating that the discarded information is essential for scoring. $\boldsymbol{p}_{ESSAY}$ yielded only marginal gains over $\boldsymbol{p}_{ORIG}$ despite producing longer prompts; suggesting that length alone is insufficient for improvement. In contrast, $\boldsymbol{p}_{RUBRIC}$ achieved the best performance, with 6 points gain, as it extends $\boldsymbol{p}_{ESSAY}$ with explicit score-level distinctions, offering more informative guidance. Accordingly, $\boldsymbol{p}_{RUBRIC}$ is selected for all subsequent experiments.

We then evaluate the effect of different *feature extraction* setups, including the similarity function

[10] https://optuna.readthedocs.io/en/stable/reference/samplers/generated/optuna.samplers.TPESampler.html

[11] https://gitlab.com/bigirqu/laila

[12] https://huggingface.co/aubmindlab/bert-base-arabert

[13] We keep their prompts and hyperparameters. Few shots sampled randomly; results averaged over 3 runs with different seeds.

Experiment	p	ϕ	N	Features	Loss	QWK$_{avg}$
Prompt Representation	p_{ORIG}	Cos	1	×	MSE	0.511
	p_{TOPIC}	Cos	1	×	MSE	0.444
	p_{ESSAY}	Cos	1	×	MSE	0.516
	p_{RUBRIC}	Cos	1	×	MSE	**0.573**
Feature Extraction Setup	p_{RUBRIC}	Dot	1	×	MSE	0.423
	p_{RUBRIC}	Cos	2	×	MSE	0.565
	p_{RUBRIC}	Cos	3	×	MSE	0.550
	p_{RUBRIC}	Cos	4	×	MSE	0.520
	p_{RUBRIC}	Cos	5	×	MSE	0.543
	p_{RUBRIC}	Cos	1	✓	MSE	0.542
Loss Function	p_{RUBRIC}	Cos	1	×	WMSE	0.438
	p_{RUBRIC}	Cos	1	×	FMSE$_{0.5}$	0.549
	p_{RUBRIC}	Cos	1	×	FMSE$_{1}$	0.507
	p_{RUBRIC}	Cos	1	×	FMSE$_{2}$	0.502

Table 1: AESER performance on LAILA dev set. ϕ is the similarity function, N is the number of generated instances, 'Features' indicates hand-crafted features inclusion, and FMSE$_\gamma$ denotes focal MSE with γ.

ϕ, the number of generated instances (N), and the inclusion of hand-crafted features. Results in the second section of Table 1 show that cosine similarity consistently outperformed dot product. Increasing the number of generated instances beyond $N = 1$ did not improve performance; instead, performance degraded as N increased. Incorporating hand-crafted features also did not yield improvements, consistent with prior findings (Bashendy et al., 2026; Albatarni et al., 2026).

We finally examine the loss function in the last section of Table 1. Neither FMSE nor WMSE improved performance over standard MSE. FMSE performance decreased as γ increased, while WMSE resulted in an even larger QWK drop. These results motivate the need for more effective approaches to address class imbalance, which we leave to future work. Overall, the best configuration uses cosine similarity, $N = 1$, and no hand-crafted features, with MSE as the loss function.

RQ2: Best Generation LLM The best configuration identified in RQ1 is applied to different generation LLMs, tuning the hyperparameters separately for each LLM. Table 2 reports AESER on the test set across different generation LLMs. Overall, no single generator consistently dominates across prompts: Fanar leads on P1, P4, P5, and P8, Gemma on P3 and P7, and GPT-4o on P2 and P6. Gemma achieves the best average performance, with Fanar close behind, while GPT-4o trails both by approximately 5 points. These results indicate that Arabic-centric and open multilingual LLMs can match and even outperform the proprietary GPT-4o, suggesting that a strong closed-source model is not necessary for generating prompt representations in AESER.

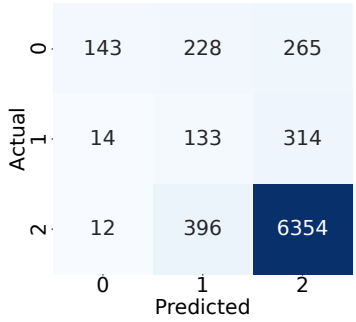


Figure 4: Confusion matrix of AESER$_{\text{Gemma}}$, our best configuration, on the full LAILA test set.

RQ3: AESER vs. SOTA Table 2 compares AESER with the baselines. The best, AESER$_{Gemma}$, reaches an average QWK of 0.517, improving over the strongest baseline, MOOSE, by 11 points (26% relative). AESER also obtains the best score on seven of the eight prompts, with P1 being the only exception. Notably, both AESER$_{\text{Fanar}}$ and AESER$_{\text{Gemma}}$ show substantial gains on P5 compared to other prompts, improving by 34 and 29 points, respectively, over MOOSE. This improvement can be attributed to the highly specific nature of the prompt topic, *"Qatar's development and culture."* Unlike more open-ended prompts such as *"staying up late,"* P5 is grounded in a narrow and concrete topical space, which reduces semantic ambiguity

Moreover, AESER$_{\text{Fanar}}$ performance tends to be stronger on persuasive prompts (P4, P6, and P8), with an average improvement of 14 points over MOOSE compared to explanatory prompts (P1 and P7). This likely reflects the more structured nature of persuasive writing (clear stance, arguments, and conclusion), which promotes greater lexical and discourse-level alignment between generated and student essays. In contrast, as noted in LAILA dataset (Bashendy et al., 2026), explanatory essays have no particular structure, allowing for wider variation in content and organization and making relevance distinctions more challenging.

5.2 Performance Analysis

To understand where AESER succeeds and fails, we examine its predictions in more detail. We first inspect its errors across the three relevance levels through a confusion matrix, and then study two contrasting essays to see how the model judges rel-

Model	P1	P2	P3	P4	P5	P6	P7	P8	Ave.
AraBERT4GR (Albatarni et al., 2024)	0.173	0.396	0.304	0.362	0.186	0.361	0.087	0.307	0.272
Qwen$^{fs(3)}$ (Qwaider et al., 2025)	**0.510**	0.133	0.281	0.229	0.446	0.285	0.163	0.255	0.288
Falcon$^{fs(3)}$ (Qwaider et al., 2025)	0.312	0.110	0.355	0.260	0.335	0.129	0.172	0.101	0.222
Fanar$^{fs(5)}$ (Bashendy et al., 2026)	0.291	0.476	0.497	0.361	0.478	0.378	0.363	0.424	0.409
MOOSE (Bashendy et al., 2026)	0.119	0.471	0.406	0.506	0.325	0.415	0.593	0.450	0.411
AESER$_{\text{Fanar}}$	0.298	0.472	0.468	**0.628**	**0.669**	0.555	0.429	**0.598**	0.514
AESER$_{\text{Gemma}}$	0.217	0.528	**0.533**	0.523	0.610	0.553	**0.663**	0.505	**0.517**
AESER$_{\text{GPT-4o}}$	0.199	**0.531**	0.414	0.595	0.393	**0.584**	0.658	0.362	0.467

Table 2: AESER results on the LAILA test set. AESER$_{\text{FANAR}}$, AESER$_{\text{Gemma}}$, and AESER$_{\text{GPT-4o}}$ use FANAR-27B, Gemma 4 31B, and GPT-4o, respectively, to generate the prompt representations. Best score per prompt in **bold**.

Writing prompt (P5, Explanatory).

تعمل دولة قطر على تحقيق توازن بين تراثها الثقافي والحداثة. اكتب مقالًا مكونًا من ثلاثمائة (300) كلمة توضّح فيه السبل والوسائل التي تستخدمها الدولة في تحقيق هذا التوازن، مُراعيًا سمات المقال التفسيري، وسلامة اللغة، ومُوظفًا علامات الترقيم وأدوات الربط بشكلٍ صحيح.

English translation: **The State of Qatar is working to achieve a balance between its cultural heritage and modernity.** Write a 300-word essay explaining the methods and means the country uses to achieve this balance, taking into account the features of an expository essay, ensuring linguistic correctness, and using punctuation marks and linking devices properly.

Essay 50310 (Relevance score: 1, AESER$_{\text{Gemma}}$: 1)

السبل و الوسائل التي تستخدمها الدولة في تحقيق هذا التوازن . اولا تحفيز المواطنين و المقيمين في الدولة . ثانيا الحفاظ على الاهمام بالبيئة و نضافتها و نشر اهميتها في المجتمع و الانتباه عليها . ثالثا التعاون فهو مهم في توحيد الدولة و هذا يزيد المحبة في البلد و نشر التعاون و اهمية في الدولة للحفاظ على الاستقرار و تحقيق التوازن في الدولة . رابعا الاهتمام في الرياضة تطور الرياضة يزيد العديد من السكان بسبب حبها للرياضة وهنالك الكثير من الرياضات مثل : كرة القدم , كرة السلة , كرة الطائرة ,الكرة الامريكية و المزيد من الرياضات . خامسا الامور الدينية يجب علينا نشر قيمنا و ديننا للعالم و ثقافتنا للكسب في الدنيا و الاخرة .

Essay 50124 (Relevance score: 1, AESER$_{\text{Gemma}}$: 2)

التراث الثقافي والحادثة هي ظواهر طبيعية تحدث مع تغير الاجيال والقرون بين الناس ،وكما انه في الزمن الحالي نرى انه تنتشر الحداثة والتطور في دولتنا قطر، كما ان دولة قطر تفتخر وتحافظ على التراث الثقافي والقطري ،وسأحدثكم في الاسطر التالية عن الوسائل التي تستخدمها دولة قطر في تحقيق هذا التوازن .

ومن وسائل تحقيق التوازن ان دولة قطر تحافظ على المباني التراثية و الاحتفاظ ب الادوات التراثية القطرية في المتاحف .

وكما ان الشعب القطري يحافظ على العادات والتقاليد مثل : لبس الثوب القطري و رقص العرضة في المناسبات ،وكما انه اعتاد على كرم الضيافة على سبيل المثال : كرم اهل قطر في كأس العالم 2022 ،وكما انهم نشروا الثقافة القطرية في ذلك الزمان.

في الختام، يجب علينا الافتخار ونشر ثقافتنا القطرية بين الزوار و الاحتفاظ بكرم دولة قطر.

Figure 5: Two essays from Prompt 5, both labeled as partially relevant (score 1) by the human annotator. The prompt uses bold to mark the topic. AESER$_{\text{Gemma}}$ correctly scores the left essay, but over-scores the right essay.

evance in practice.

Error Analysis Figure 4 shows the confusion matrix of AESER$_{\text{Gemma}}$, our best-performing configuration, over LAILA test set across all prompts. The model is highly accurate on fully relevant essays (score 2), correctly classifying 6,354 of 6,762. It is much weaker on the two minority classes, recovering only 143 of 636 not-relevant essays (score 0) and 133 of 461 partially relevant essays (score 1). These minority essays are mostly over-predicted toward the majority class, with 265 not-relevant and 314 partially relevant essays assigned score 2. This behavior is largely due to the severe class imbalance of the relevance labels (section 4), where the vast majority of essays are fully relevant. AESER's bias toward the majority class further motivates exploring class-balanced training or data augmentation to improve minority-class performance.

Case Study To complement the aggregate analysis, Figure 5 shows two essays written for the same explanatory prompt on Qatar's efforts to balance cultural heritage and modernity. Both are labeled as partially relevant (1), but they reflect different patterns in how relevance is judged. The first essay (left) is clearly weak in task fulfillment, relying on a fragmented list structure with minimal sentence development and little coherent explanation of how balance is achieved, making its low engagement with the prompt easy to identify. The second essay is more fluent and includes explicit references to culturally salient entities such as "traditional

dance" and "Qatari dress," which increases its apparent relevance; however, it remains largely descriptive, focusing on heritage display rather than explaining mechanisms of balance, and it omits the modernity aspect. These patterns are expected given the information available to AESER, namely token-level prompt–essay similarity matrices (subsection 3.2). As a result, AESER responds to surface topical cues and shared entities but does not model task completion by design. Consequently, strong topical overlap can raise the predicted relevance even when the essay does not fully address the prompt.

6 Conclusion and Future Work

In this paper, we presented AESER, a framework that formulates relevance assessment of essays as an interaction-based matching task. Combining LLM-based prompt reformulation with token-level similarity matrices and convolutional matching, AESER enables fine-grained modeling of prompt–essay alignment. Our experiments on a large Arabic dataset show that aligning prompt representations with the relevance rubric provides an effective basis for relevance scoring, with AESER outperforming strong baselines by 10 points. Future work will explore more effective approaches to address class imbalance and apply AESER to essays in other languages to evaluate its generalizability.

Acknowledgments

This work was made possible by NPRP grant NPRP14S-0402-210127 from the Qatar Research Development and Innovation (QRDI) Council. The work of Tamer Elsayed was also supported by the awarded research fund QUEX-CENG-GOOGLE-2025-1 from Google. The statements made herein are solely the responsibility of the authors.

7 Limitations

AESER relies on an LLM to generate prompt representations, which introduces an additional requirement compared to feature-based or fine-tuned approaches, and its computational cost increases as the rubric introduces more relevance levels. Despite this, our results indicate that relatively small Arabic-centric or multilingual LLMs are generally sufficient (section 5), which substantially reduces practical cost compared with larger proprietary LLMs. The evaluation is also relatively limited in scope, as it is conducted on a single Arabic dataset with three relevance levels, leaving the behavior of AESER on other datasets and more fine-grained scoring rubrics as an open direction for future work.

References

Salam Albatarni, May Bashendy, Sohaila Eltanbouly, and Tamer Elsayed. 2026. MAPLE: A meta-learning framework for cross-prompt essay scoring. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 19948–19961, San Diego, California, United States. Association for Computational Linguistics.

Salam Albatarni, Sohaila Eltanbouly, and Tamer Elsayed. 2024. Graded relevance scoring of written essays with dense retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 1329–1338, New York, NY, USA. Association for Computing Machinery.

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: Shared Tasks*, pages 989–997, Suzhou, China. Association for Computational Linguistics.

Mohamad Alnajjar, Ahmad Almoustafa, Tomohiro Nishiyama, Shoko Wakamiya, Eiji Aramaki, and Takuya Matsuzaki. 2025. ARx-HYOKA at TAQEEM2025: Comparative approaches to Arabic essay trait scoring. In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 977–982, Suzhou, China. Association for Computational Linguistics.

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: Shared Tasks*, pages 966–976, 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, Bangkok, Thailand. Association for Computational Linguistics.

May Bashendy, Walid Massoud, Sohaila Eltanbouly, Salam Albatarni, Marwan Sayed, Abrar Abir, Houda Bouamor, and Tamer Elsayed. 2026. LAILA: A large trait-based dataset for Arabic automated essay scoring. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3074–3091, Rabat, Morocco. Association for Computational Linguistics.

James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. 2011. Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc.

Po-Kai Chen, Bo-Wei Tsai, Shao Kuan Wei, Chien-Yao Wang, Jia-Ching Wang, and Yi-Ting Huang. 2025. Mixture of ordered scoring experts for cross-prompt essay trait scoring. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18071–18084, Vienna, Austria. Association for Computational Linguistics.

Yuan Chen and Xia Li. 2023. PMAES: Prompt-mapping contrastive learning for cross-prompt automated essay scoring. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1489–1503, Toronto, Canada. Association for Computational Linguistics.

Yuan Chen and Xia Li. 2024. PLAES: Prompt-generalized and level-aware learning framework for cross-prompt automated essay scoring. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 12775–12786, Torino, Italia. ELRA and ICCL.

Jacob Cohen. 1968. Weighted kappa: nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, 70(4):213.

Ronan Cummins, Helen Yannakoudakis, and Ted Briscoe. 2016. Unsupervised modeling of topical relevance in l2 learner text. In *Proceedings of the 11th workshop on innovative use of NLP for building educational applications*, pages 95–104.

Heejin Do, Yunsu Kim, and Gary Geunbae Lee. 2023. Prompt- and trait relation-aware cross-prompt essay trait scoring. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1538–1551, Toronto, Canada. Association for Computational Linguistics.

Sohaila Eltanbouly, Salam Albatarni, and Tamer Elsayed. 2025. TRATES: Trait-specific rubric-assisted cross-prompt essay scoring. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20528–20543, Vienna, Austria. Association for Computational Linguistics.

Google DeepMind. 2026. Gemma 4: Byte for byte, the most capable open models. Google Blog.

Cancan Jin, Ben He, Kai Hui, and Le Sun. 2018. TDNN: A two-stage deep neural network for prompt-independent automated essay scoring. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1088–1097, Melbourne, Australia. Association for Computational Linguistics.

Shengjie Li and Vincent Ng. 2024. Automated essay scoring: A reflection on the state of the art. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17876–17888, Miami, Florida, USA. Association for Computational Linguistics.

Xia Li, Minping Chen, and Jian-Yun Nie. 2020. Sednn: Shared and enhanced deep neural network model for cross-prompt automated essay scoring. *Knowledge-Based Systems*, 210:106491.

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.

Sandeep Mathias and Pushpak Bhattacharyya. 2018. ASAP++: Enriching the ASAP automated essay grading dataset with essay attribute scores. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

OpenAI. 2024. GPT-4o system card. *arXiv preprint arXiv:2410.21276*.

Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2016. Text matching as image recognition. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)*, pages 2793–2799.

Isaac Persing and Vincent Ng. 2014. Modeling prompt adherence in student essays. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1534–1543, Baltimore, Maryland. Association for Computational Linguistics.

Chatrine Qwaider, Kirill Chirkunov, Bashar Alhafni, Nizar Habash, and Ted Briscoe. 2025. Evaluating prompt relevance in Arabic automatic essay scoring: Insights from synthetic and real-world data. In *Proceedings of The Third Arabic Natural Language Processing Conference*, pages 162–178, Suzhou, China. Association for Computational Linguistics.

Marek Rei and Ronan Cummins. 2016. Sentence similarity measures for fine-grained estimation of topical relevance in learner essays. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 283–288, San Diego, CA. Association for Computational Linguistics.

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.

Robert Ridley, Liang He, Xinyu Dai, Shujian Huang, and Jiajun Chen. 2020. Prompt agnostic essay scorer: a domain generalization approach to cross-prompt automated essay scoring. *arXiv preprint arXiv:2008.01441*.

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 Third Arabic Natural Language Processing Conference*, pages 231–245, Suzhou, China. Association for Computational Linguistics.

AraDetox: A Multi-Dialect Arabic Detoxification Dataset

Mo El-Haj
VinUniversity, Vietnam
Lancaster University, UK
https://elhaj.uk

Abstract

Arabic harmful-language detection has received considerable attention, yet Arabic text detoxification remains underexplored. We introduce AraDetox, a multi-dialect Arabic detoxification dataset comprising 10,500 harmful social-media posts and 84,000 detoxified rewrites generated using GPT-5 and Gemini 2.5 Flash across Modern Standard Arabic, Gulf, Levantine, and Egyptian Arabic. The generated outputs were assessed through human evaluation and automatic analyses of lexical change, semantic preservation, sentiment, and dialectal style. Results show that detoxification is primarily a meaning-preserving rewriting task: substantial lexical and structural reformulation is accompanied by consistently high semantic similarity. Human evaluation confirms successful harmful-language removal while largely preserving the original meaning. Dialectal analyses further indicate that the generated variants exhibit measurable stylistic alignment with reference Arabic dialect corpora. Comparison with existing resources highlights two complementary approaches to detoxification: minimal-edit lexical substitution and meaning-preserving reformulation. Our findings demonstrate that large-scale Arabic detoxification resources can be constructed through LLM-assisted generation and human verification. The dataset is publicly available at https://github.com/ArabicNLP-UK/AraDetox to support future research on Arabic detoxification, safe text generation, and multi-dialect Arabic NLP.

1 Introduction

Research on harmful language in Arabic has largely focused on detection, resulting in numerous datasets and models for hate speech, offensive language, and toxicity classification (Al Mandhari et al., 2024). Comparatively little attention has been paid to *text detoxification*, where harmful content is rewritten into a safer form while preserving its meaning. Creating detoxification datasets is particularly challenging because it requires annotators to rewrite offensive, politically sensitive, and often dialectal content while preserving meaning, stance, target, and communicative intent. As a result, manual dataset creation is both costly and time-consuming (Logacheva et al., 2022; Dementieva et al., 2024). Recent advances in large language models (LLMs) provide a promising alternative, enabling large-scale generation of fluent rewrites that remove harmful language while retaining the underlying message. Unlike harmful-language detection, detoxification requires balancing two competing objectives: removing harmful expressions while preserving the original claim, stance, target, and communicative intent. This challenge is particularly pronounced in Arabic social-media discourse, where criticism, sarcasm, political disagreement, and identity-related language are often intertwined with offensive expressions. As a result, successful detoxification cannot be reduced to simple lexical substitution, but instead requires broader meaning-preserving reformulation.

We introduce AraDetox, a large-scale multi-dialect Arabic detoxification dataset containing 10,500 harmful social-media posts and 84,000 detoxified rewrites generated using GPT-5 and Gemini 2.5 Flash across four Arabic varieties: Modern Standard Arabic (MSA), Gulf Arabic, Levantine Arabic, and Egyptian Arabic. The dataset was created using an LLM-assisted synthesis pipeline followed by human quality control by native Arabic speakers. We evaluate AraDetox using lexical change, semantic similarity, sentiment analysis, dialectal style analysis, and human assessment. Our findings show that Arabic detoxification is best characterised as a meaning-preserving rewriting task. Across models and dialects, substantial lexical and structural reformulation is accompanied by consistently high semantic similarity. Human evaluation further confirms that harmful language can be removed while largely preserving

Proceedings of The Fourth Arabic Natural Language Processing Conference (ArabicNLP 2026), pages 33–47
October 28-29, 2026 ©2026 Association for Computational Linguistics

the original meaning. Beyond introducing a new resource, our work demonstrates that existing Arabic harmful-language datasets can be transformed into large-scale multi-dialect detoxification resources through LLM-assisted generation and human verification. The dataset is publicly available through the AraDetox repository[1]. The main contributions of this work are threefold. First, we introduce a large-scale Arabic detoxification resource containing eight rewrites for each source post across two LLM families and four Arabic varieties. Second, we provide a multi-perspective evaluation covering lexical reformulation, semantic preservation, sentiment, dialectal style, and independent human assessment. Third, we release the dataset to support reproducible research on Arabic detoxification and dialect-aware safe generation.

2 Related Work

Research on harmful language in Arabic has focused primarily on detection, resulting in datasets and models for hate speech, offensive language, and abusive-language classification (Mubarak et al., 2021, 2022; Alghamdi et al., 2024), as well as Arabic language models such as AraBERT (Antoun et al., 2020) and ARBERT/MARBERT (Abdul-Mageed et al., 2021). Comparatively little attention has been paid to *text detoxification*, which aims to rewrite harmful content into safer forms while preserving meaning. ParaDetox introduced a parallel-data setting based on toxic and non-toxic paraphrase pairs (Logacheva et al., 2022), while subsequent work extended detoxification to multilingual settings through MultiParaDetox (Dementieva et al., 2024, 2025). These resources established detoxification as a generation task distinct from harmful-language detection, but existing Arabic resources remain relatively limited in scale and provide only a single detoxified rewrite per source text. For Arabic, the original MultiParaDetox dataset was derived from approximately 2,100 candidate posts and resulted in 1,181 manually annotated source–detoxified pairs after filtering and quality control. For our empirical comparison, we use the publicly available Arabic evaluation split released through Hugging Face[2], which contains 400 source–detoxified pairs. Recent work has also shown that LLMs can generate high-quality synthetic datasets and annotations, making them increasingly useful for data creation and augmentation (Gilardi et al., 2023; Long et al., 2024; Nadăș et al., 2025; Yong et al., 2024). AraDetox builds on these developments by providing detoxified rewrites for Arabic social-media content across Modern Standard Arabic, Gulf, Levantine, and Egyptian Arabic using GPT-5 and Gemini. Unlike existing Arabic detoxification resources, AraDetox emphasises semantic preservation while retaining the original intent, stance, and target of the source text. AraDetox substantially expands the scale of Arabic detoxification resources and, to the best of our knowledge, is the first Arabic detoxification dataset to provide rewrites generated across multiple dialects and multiple LLMs within a single framework.

3 Methodology

3.1 Dataset Construction

AraDetox is derived from the Arabic Hate Speech Superset (Tonneau et al., 2024), which integrates ten Arabic hate-speech and abusive-language datasets: T-HSAB (Haddad et al., 2019), JHSC (Ahmad et al., 2024), MLMA (Ousidhoum et al., 2019), L-HSAB (Mulki et al., 2019), OSACT (Mubarak et al., 2020), Let-Mi (Mulki and Ghanem, 2021), Saudi Tweets (Alshaalan and Al-Khalifa, 2020), AraCOVID19-MFH (Ameur and Aliane, 2021), Brothers (Albadi et al., 2018), and Alsafari et al. (Alsafari et al., 2020). From the 16,000 harmful posts available in the superset, we selected 10,500 posts while retaining coverage across all ten constituent datasets. No class balancing was applied, and the source datasets were not equally represented because they varied substantially in size. The selection was intended to provide broad coverage of the available harmful-language data rather than a balanced classification sample. Table 1 summarises the resulting resource.

Using GPT-5 and Gemini 2.5 Flash, we generated detoxified rewrites in MSA, Gulf, Levantine, and Egyptian Arabic. Models were instructed to preserve the original meaning, target, stance, sentiment, and communicative intent while removing profanity, insults, slurs, dehumanising language, discriminatory expressions, and threats. For the dialectal variants, outputs were additionally required to adapt the source text into Gulf, Levantine, or Egyptian Arabic. These outputs therefore combine two transformations: harmful-language detoxification and adaptation to the requested Arabic variety.

[1] https://github.com/ArabicNLP-UK/AraDetox

[2] https://huggingface.co/datasets/textdetox/multilingual_paradetox

Statistic	Value
Source posts	10,500
Detoxified outputs	84,000
Arabic varieties	MSA, Gulf, Levantine, Egyptian
Generation models	GPT-5, Gemini 2.5 Flash
Outputs per source post	8
Outputs generated by GPT-5	42,000
Outputs generated by Gemini	42,000
Total words	1,318,257
Average words per output	15.69
Median words per output	13
Human-evaluation sample	300 source posts
Human-evaluated outputs	2,400
Annotators	3
Annotation records	7,200

Table 1: Summary statistics for AraDetox.

The MSA outputs primarily represent detoxification within the same language variety, whereas the Gulf, Levantine, and Egyptian outputs jointly involve detoxification and dialect adaptation. Consequently, lexical and structural differences between the source posts and the dialectal outputs cannot be attributed exclusively to detoxification. We therefore evaluate harmful-language removal, meaning preservation, and dialectal style as separate dimensions and interpret the dialectal results as evidence of joint detoxification and dialect adaptation. This produced eight detoxified variants per source post, yielding 84,000 detoxified instances and more than 1.3 million words. The full prompts are provided in Appendix A. Dataset construction incorporated automatic validation, regeneration of invalid outputs, and human quality control. Automatic checks identified missing fields, malformed JSON, null values, duplicated outputs, incomplete generations, and source–output alignment errors. Safety refusals, generic moderation responses, summaries, external-observer descriptions, and outputs that substantially altered or added to the source meaning were treated as invalid and regenerated. A native Arabic-speaking reviewer subsequently verified that the regenerated outputs performed the requested detoxification task, preserved the source meaning, target, stance, and communicative intent, and reflected the requested Arabic variety. Further details of the generation procedure and quality-control workflow are provided in Appendix B.

As an additional quality-assurance step, 300 source posts were randomly selected from the 10,500-post collection. The sample was stratified to preserve the distribution of the ten source datasets across the ten source datasets and available harm categories. For each selected post, all eight detoxified variants were evaluated by three native Arabic-speaking annotators, yielding 2,400 generated outputs and 7,200 annotation records. The annotators involved in this evaluation were distinct from the native Arabic speaker who performed quality control during dataset construction. Appendix E reports the distribution of the full dataset and the human-evaluation sample across the constituent source datasets.

3.2 Evaluation Design

We evaluate AraDetox from five complementary perspectives: lexical change, semantic preservation, sentiment shift, dialectal style, and human evaluation. We additionally compare AraDetox with MultiParaDetox-Ar (Dementieva et al., 2024, 2025) to contextualise its detoxification behaviour relative to an existing Arabic detoxification resource. A key challenge in detoxification is preserving meaning while removing harmful content. Offensive expressions often convey stance, criticism, and emotion rather than functioning as removable noise. For example, the insult لا انت الكلب يا خنزير ("No, you are the dog, you pig") can be rewritten as لا، بل إنك أنت من يتصرف بطريقة غير مقبولة ("No, rather, you are the one behaving in an unacceptable manner"), preserving the criticism while removing the offensive language. Appendix F presents additional qualitative examples from AraDetox. F1 score is not used as a generation-quality metric because detoxification does not have a single correct reference sequence from which token-level true positives, false positives, and false negatives can be meaningfully derived. Multiple lexically different rewrites may all be valid. We therefore combine surface-change measures, semantic similarity, sentiment analysis, dialectal-style analysis, and human judgements, with each evaluation component measuring a distinct requirement of successful detoxification.

3.2.1 Lexical Change

We measure lexical change between each source post and its detoxified counterpart using six surface-level metrics: lexical overlap, Jaccard similarity, character-level edit distance, percentage of changed characters, word-level edit distance, and output length. Lexical overlap and Jaccard similarity measure how much of the original vocabulary is retained, with higher values indicating more conservative rewriting. Character- and word-level edit distances quantify the amount of rewriting required to transform the source post into the detoxified out-

put, while the percentage of changed characters normalises this change by source length. Output length is included to capture whether detoxification tends to shorten, preserve, or expand the original text. These measures allow us to distinguish minimal-edit detoxification from broader sentence-level reformulation.

3.2.2 Semantic Preservation

To quantify semantic fidelity, we compute cosine similarity between each source post and its detoxified counterpart using `multilingual-e5-large` (Wang et al., 2024). This embedding-based measure assesses whether the detoxified text remains semantically close to the original despite lexical and structural changes. To complement the pairwise similarity analysis, we examine the global organisation of the embedding space using UMAP. By visualising source posts, detoxified outputs, and external neutral Arabic corpora in a shared semantic space, we assess whether detoxified texts remain close to their source content while exhibiting characteristics associated with naturally occurring neutral language.

3.2.3 Sentiment and Neutrality Analysis

Detoxification should remove harmful language without eliminating criticism, disagreement, or negative opinions. We therefore examine whether detoxification systematically shifts sentiment by increasing neutral or positive predictions at the expense of negative ones. This follows the broader use of sentiment analysis for examining evaluative language and affect across different domains and languages, including financial discourse, social media, and multilingual settings (El-Haj et al., 2016; Alwakid et al., 2022; Hunter et al., 2023; Huy et al., 2026). Sentiment distributions are computed using three CAMeLBERT sentiment classifiers: CAMeLBERT-DA, CAMeLBERT-MSA, and CAMeLBERT-Mix (Inoue et al., 2021). Changes in positive, neutral, and negative predictions are analysed before and after detoxification to assess whether rewriting reduces hostile affect while preserving the original stance.

3.2.4 Dialectal Adaptation

AraDetox includes detoxified variants in Gulf, Levantine, and Egyptian Arabic. To examine whether these rewrites exhibit stylistic alignment with their intended varieties, we compare them against reference dialect corpora (Zaidan and Callison-Burch, 2014) using TF–IDF cosine similarity over word and character n-grams. This analysis provides corpus-level evidence of lexical and orthographic alignment, but it is sensitive to topic, register, corpus composition, and vocabulary shared across Arabic varieties. It is therefore treated as an exploratory measure of dialectal style rather than definitive validation of dialect authenticity. For example, the MSA phrase أغلقوا الشارع ("they blocked the street") may be realised as سادين الشارع in Gulf Arabic, سكروا الشارع in Levantine Arabic, and قفلوا الشارع in Egyptian Arabic, illustrating meaning-preserving adaptation across Arabic varieties.

3.2.5 Human Evaluation Protocol

To assess the quality of the generated outputs, three native Arabic-speaking annotators independently reviewed a random sample of 300 source posts. For each post, all eight detoxified variants were evaluated, yielding 2,400 generated outputs and 7,200 annotation records. Each output was assessed for offence removal, meaning preservation, and introduction of new content. Offence removal was annotated using a three-way scheme: offence not removed, offence removed, or not applicable when the source text did not contain offensive content. Meaning preservation and new content were annotated using binary judgements. The full annotation guidelines are provided in Appendix D. Dialect authenticity was not included as a separate human-evaluation criterion. The dialectal findings therefore rely on corpus-level stylistic analysis and should not be interpreted as direct human validation of native-like dialect use.

3.2.6 Comparison with Existing Resources

To contextualise AraDetox, we compare it descriptively with MultiParaDetox-Ar using the same lexical, semantic, and sentiment-based measures. The resources differ in source-domain distribution, size, annotation procedure, and the initial sentiment and severity of their source texts. The comparison is therefore intended to characterise their rewriting behaviour rather than establish that one resource or construction method is superior. It also contrasts the relatively minimal-edit strategy represented by MultiParaDetox-Ar with the broader meaning-preserving reformulation and dialect adaptation represented by AraDetox.

4 Results

4.1 Lexical Change

Table 2 summarises the lexical transformations introduced by all eight detoxification systems. Overlap and Jaccard measure vocabulary retention between the original and detoxified texts, while CharEdit, Char%, WordEdit, and Word% capture character- and word-level modifications. Across all variants, detoxification results in substantial modification of the original text, indicating that the generated outputs frequently involve sentence-level reformulation rather than simple replacement of offensive expressions. Among all systems, GPT-MSA is the most conservative. It achieves the lowest character-level edit distance (42.19), the lowest proportion of modified characters (61.58%), the lowest word-level edit distance (11.77), the lowest word-level modification rate (94.15%), and produces the shortest outputs (11.28 words on average). GPT-MSA therefore preserves more of the original surface form than the other systems.

In contrast, Gemini-MSA performs more extensive rewriting than GPT-MSA, exhibiting larger character- and word-level modifications, higher proportions of changed characters, and longer outputs. The dialectal variants generally involve even greater reformulation. Across both models, output lengths increase from an average of 13.21 words in the original posts to 15.66–16.79 words, while character-level modification rates exceed 78%. Normalised word-level edit rates range from 115.79% to 122.94%. Values above 100% occur because word-level edit distance is normalised by the original text length and includes insertions, deletions, and substitutions, allowing the number of edit operations to exceed the number of words in the source text.

GPT-Gulf exhibits the highest lexical overlap (0.2942) and Jaccard similarity (0.1719), indicating greater retention of source vocabulary. In contrast, Gemini-MSA and Gemini-Egyptian show the lowest overlap scores, suggesting a greater degree of lexical reformulation. The lexical analyses suggest that Arabic detoxification is best characterised as a meaning-preserving rewriting task rather than simple lexical substitution. Whether these extensive modifications preserve the underlying message is examined through the semantic and human evaluations that follow.

4.2 Semantic Preservation

Table 3 reports semantic similarity between the original posts and the AraDetox variants using multilingual E5 embeddings. Across all variants, semantic similarity remains consistently high, with mean scores ranging from 0.9234 to 0.9308 and all examples exceeding the 0.70 similarity threshold. These results suggest that the generated rewrites preserve the core meaning of the source posts despite the substantial lexical modifications reported in Section 4.1.

The dialect-aware variants achieve slightly higher similarity scores than the MSA variants, with Levantine obtaining the highest mean similarity (0.9308), followed by Egyptian and Gulf Arabic. GPT-5 and Gemini also exhibit comparable levels of semantic preservation, with mean similarities of 0.9234 and 0.9249, respectively. The relatively small differences across variants suggest that semantic fidelity is largely unaffected by the choice of dialect or generation model.

The lexical and semantic results show that substantial reformulation does not necessarily come at the expense of semantic fidelity. The generated outputs often differ considerably from the source posts at the surface level, while remaining close in meaning. This supports treating detoxification as a meaning-preserving rewriting task rather than simple lexical substitution.

Figures 1 and 2 visualise the embedding space using UMAP. Across all variants, substantial overlap is observed between the original posts and their detoxified counterparts, with no clear separation between source texts and generated rewrites. Similar patterns are observed for the dialect-aware variants, whose outputs occupy largely the same regions of the embedding space as the original posts. These visualisations are consistent with the high semantic-similarity scores reported in Table 3 and provide additional evidence that detoxification and dialect adaptation preserve the semantic structure of the source content.

4.3 Sentiment Shift

We investigate whether detoxification influences negative affect using three Arabic sentiment models: CAMeLBERT-DA, CAMeLBERT-MSA, and CAMeLBERT-Mix. Sentiment is not treated as a direct measure of toxicity, since a post may remain critical, negative, or emotionally charged without being abusive, hateful, or discriminatory. Rather,

System	Overlap	Jaccard	CharEdit	Char%	WordEdit	Word%	Len
GPT-MSA	0.2062	0.1339	42.19	61.58	11.77	94.15	11.28
GPT-Gulf	0.2942	0.1719	45.52	81.24	13.08	115.79	15.99
GPT-Levantine	0.2737	0.1610	45.90	80.66	13.24	116.14	15.66
GPT-Egyptian	0.2537	0.1454	48.07	84.72	13.67	120.51	16.04
Gemini-MSA	0.2007	0.1201	52.74	83.61	14.59	119.38	16.76
Gemini-Gulf	0.2342	0.1353	49.90	83.90	14.39	122.94	16.79
Gemini-Levantine	0.2440	0.1471	47.69	78.88	13.94	117.08	16.27
Gemini-Egyptian	0.2188	0.1263	49.93	81.26	14.50	120.85	16.75

Table 2: Lexical change statistics.

Variant	Mean	Median	Min.	>0.90
Levantine	**0.9308**	**0.9349**	**0.7883**	**80.31%**
Egyptian	0.9274	0.9307	0.7789	77.93%
Gulf	0.9256	0.9293	0.7700	76.15%
Gemini	0.9249	0.9278	0.7691	73.30%
GPT-5	0.9234	0.9278	0.7504	71.05%

Table 3: Semantic similarity using multilingual E5 embeddings.

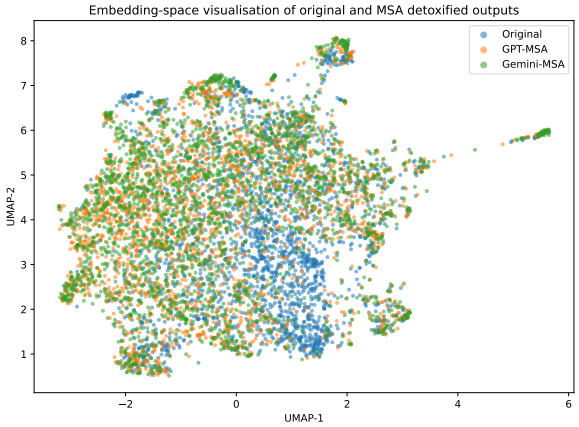


Figure 1: UMAP visualisation of original posts, GPT-MSA detoxifications, and Gemini-MSA detoxifications using multilingual E5 embeddings.

sentiment provides an additional perspective on how detoxification alters the tone of the text while preserving its meaning and intent.

Table 4 and Figure 3 report the percentage of posts classified as negative before and after detoxification. The original posts are classified as negative at high rates across all three sentiment models, ranging from 72.41% to 75.47%. After detoxification, the strongest reduction is observed for the MSA variants. GPT-MSA reduces negative sentiment to 49.20%–53.77%, while Gemini-MSA reduces it to 53.19%–60.26%. MSA detoxification substantially softens the affective tone of harmful posts.

The dialectal variants also reduce negative sentiment, but to a more limited extent. Across both GPT and Gemini, Gulf, Levantine, and Egyptian outputs remain closer to the original sentiment distribution,

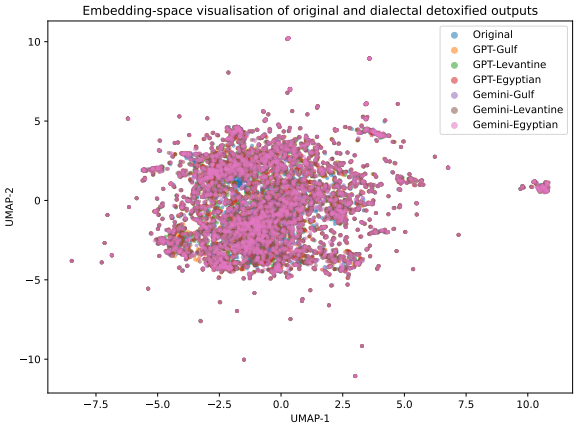


Figure 2: UMAP visualisation of original posts and GPT/Gemini dialectal detoxified variants using multilingual E5 embeddings.

Model	Orig.	GPT-MSA	GPT-Gulf	GPT-Lev.	GPT-Egy.
DA	74.29	53.77	70.32	71.14	69.38
MSA	75.47	53.48	71.23	73.67	70.44
Mix	72.41	**49.20**	68.62	69.56	66.36

Model	Orig.	Gem-MSA	Gem-Gulf	Gem-Lev.	Gem-Egy.
DA	74.29	55.79	68.07	70.13	70.07
MSA	75.47	60.26	69.69	72.24	70.21
Mix	72.41	53.19	65.45	67.42	65.18

Table 4: Negative sentiment before and after detoxification using CAMeLBERT-DA, MSA and Mix models.

with negative rates generally between 65% and 74%. Dialectal rewriting preserves more of the original emotional tone, even after harmful expressions have been removed. The pattern is also visible in Figure 4, where the MSA variants show the clearest drop in average negative sentiment, while the dialectal variants remain nearer to the original posts.

These results indicate that part of the negative affect detected by sentiment models is associated with abusive wording, insults, profanity, and other harmful linguistic constructions. Once such language is removed, the resulting texts are generally perceived as less negative, particularly in MSA. At the same time, the dialectal variants show that detoxification does not necessarily eliminate criticism, disagreement, or emotional stance. Rather, the generated

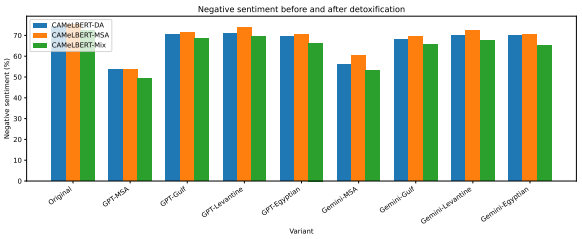


Figure 3: Negative sentiment across GPT and Gemini variants using three CAMeLBERT models.

outputs preserve much of the communicative intent while reducing harmful expression.

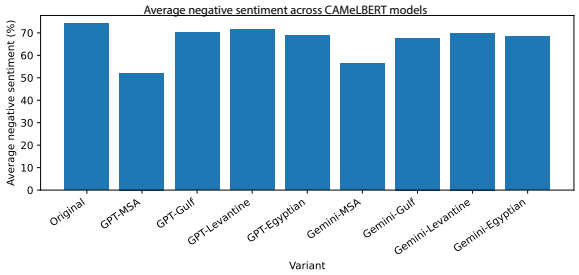


Figure 4: Average negative-sentiment rate across the three CAMeLBERT models.

4.4 Dialectal Style Analysis

AraDetox includes detoxified rewrites in MSA, Gulf, Levantine, and Egyptian Arabic. To examine corpus-level dialectal style, we compare the generated variants with reference texts from the Arabic-Dialects corpus using TF–IDF cosine similarity over word and character n-grams.

Figures 5 and 6 show the resulting similarity matrices. The word n-gram results provide the clearest evidence of dialectal alignment. Both MSA systems align most strongly with the MSA reference corpus, with Gemini-MSA achieving the highest MSA similarity (0.373), followed by GPT-MSA (0.350). The Egyptian variants likewise show strong alignment with the Egyptian reference corpus, with GPT-Egyptian obtaining the highest similarity (0.406), followed by Gemini-Egyptian (0.383).

The Gulf and Levantine results are less distinct. Gemini-Gulf aligns most strongly with the Gulf reference corpus (0.285), and Gemini-Levantine aligns most strongly with the Levantine reference corpus (0.297), suggesting clearer dialect-specific stylistic patterns. In contrast, GPT-Gulf and GPT-Levantine show their highest similarity with the Egyptian reference corpus, likely reflecting lexical overlap among informal Arabic varieties and the strong signal of the Egyptian reference set. The character n-gram analysis broadly confirms the

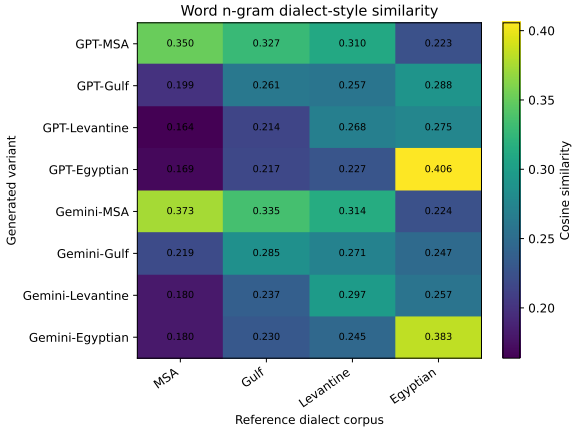


Figure 5: Word n-gram similarity to reference dialect corpora.

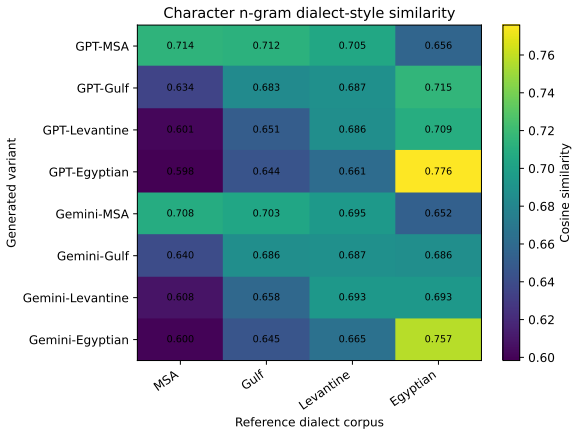


Figure 6: Character n-gram similarity to reference dialect corpora.

MSA and Egyptian findings but provides weaker separation between Gulf and Levantine. Character-level features capture orthographic and morphological properties shared across Arabic dialects, making them less discriminative than word n-grams.

The results provide measurable evidence of corpus-level dialectal style, particularly for MSA and Egyptian Arabic. Gemini also shows stronger intended-variety alignment for Gulf and Levantine than GPT-5. However, the reference corpora consist largely of general social-media discussions and forum-style interactions, whereas AraDetox is derived from harmful, highly negative, politically sensitive, and identity-related discourse. Topic, register, and corpus composition may therefore influence the reported similarities. The weaker distinction between Gulf and Levantine further reflects the substantial lexical and orthographic overlap between informal Arabic varieties. These results should consequently be interpreted as evidence of stylistic alignment rather than definitive evidence of native-like dialect authenticity.

4.5 Human Evaluation

Automatic metrics provide useful insights into lexical change, semantic similarity, and sentiment shifts, but they cannot directly determine whether harmful content has been removed while preserving the intended meaning of the original post. To address this limitation, we conducted a human evaluation following the protocol described in Section 3.2.5. Three native Arabic-speaking annotators independently evaluated all outputs in the evaluation sample. The annotators represented Saudi Gulf Arabic, Jordanian Levantine Arabic, and Egyptian Arabic and were regular users of Arabic social media. They were independent of the LLM-generation process and distinct from the native Arabic speaker who performed quality control during dataset construction.

The evaluation focused on three criteria central to detoxification: *offence removal*, *meaning preservation*, and *introduction of new content*. These dimensions assess whether generated rewrites successfully remove harmful language, retain the original communicative intent, and avoid introducing information not present in the source text. To minimise potential bias, annotators were not informed that the texts had been generated by LLMs.

Inter-annotator agreement was substantial across all evaluation dimensions. Fleiss' κ values ranged from 0.715 to 0.795, indicating consistent judgements among annotators and supporting the reliability of the evaluation results (Table 5).

Criterion	Agreement (%)	Fleiss' κ
Offence Removal	96.44	0.795
Meaning Preservation	95.30	0.715
New Content	92.00	0.716

Table 5: Inter-annotator agreement for human evaluation.

Variant	Offence	Meaning	New
Gemini MSA	**92.90**	**92.57**	**12.42**
Gemini Gulf	92.35	91.46	15.52
Gemini Levantine	92.24	91.69	13.75
Gemini Egyptian	91.91	89.36	17.52
GPT MSA	87.25	86.25	13.08
GPT Gulf	87.25	91.57	24.83
GPT Levantine	87.47	91.80	24.28
GPT Egyptian	87.58	91.69	24.17

Table 6: Human evaluation results (%). Higher is better except for New Content.

Table 6 shows strong performance across all three evaluation dimensions. Offence-removal rates range from 87.25% to 92.90%, showing that harmful language can generally be removed without substantially altering the underlying message. Gemini achieves the highest offence-removal scores across all language varieties.

Meaning-preservation rates are also high, ranging from 86.25% to 92.57%. The generated rewrites therefore largely retain the original claims, targets, stances, and communicative intent of the source posts despite modifications introduced during detoxification.

The clearest differences between the model families emerge in the introduction of new content. Gemini introduces additional information in 12.42%–17.52% of cases, whereas GPT's dialectal variants do so in 24.17%–24.83% of outputs. Although GPT tends to produce more expansive rewrites, its meaning-preservation scores remain comparable to those of Gemini, suggesting that the additional content generally supplements rather than alters the core meaning of the source text.

The human evaluation findings are consistent with the automatic analyses. Both model families are effective at removing harmful language while preserving the intended meaning of the original posts. The results also point to different rewriting strategies: Gemini tends to produce more conservative reformulations, whereas GPT more frequently elaborates on the source content while maintaining similar levels of meaning preservation.

4.6 Comparison with Existing Arabic Detoxification Resources

To better understand the characteristics of AraDetox, we compare it descriptively with MultiParaDetox-Ar (Dementieva et al., 2024, 2025), one of the few publicly available Arabic detoxification resources. The comparison covers lexical change, semantic similarity, and sentiment shift. Because the two resources differ substantially in source distribution, size, annotation procedure, and source-text negativity, the results should be interpreted as a comparison of dataset characteristics rather than as a controlled benchmark.

Table 7 reveals clear differences in rewriting behaviour. MultiParaDetox-Ar follows a relatively conservative editing strategy, exhibiting higher word overlap (0.559), lower character-level modification (32.96%), and lower word-level modification (46.72%). In contrast, the AraDetox variants exhibit substantially greater reformulation, with word-overlap scores ranging from 0.201 to 0.294 and con-

siderably higher character- and word-level modification rates than those observed in MultiParaDetox-Ar. Despite these larger lexical and structural changes, semantic similarity remains consistently high across all AraDetox variants. MultiParaDetox-Ar achieves the highest E5 similarity (0.957), consistent with its minimal-edit approach. The AraDetox variants obtain E5 similarities between 0.919 and 0.930, indicating that extensive rewriting can occur while remaining semantically close to the original content. Among the AraDetox variants, the dialect-aware rewrites achieve the strongest semantic preservation, with Gulf and Levantine Arabic producing the highest similarity scores. These findings suggest that the two resources operationalise detoxification through different rewriting strategies. MultiParaDetox-Ar primarily reflects *minimal-edit detoxification*, characterised by conservative lexical modifications and high surface-level similarity to the source text. In contrast, AraDetox emphasises *meaning-preserving reformulation*, retaining the original claim, target, and stance while permitting substantially broader lexical and structural rewriting. This distinction is particularly evident in the dialectal variants, where detoxification is combined with adaptation to dialect-specific linguistic conventions.

Dataset	Pairs	Overlap	Char. %	Word %	E5
MultiParaDetox-Ar	400	**0.559**	**32.96**	**46.72**	**0.957**
AraDetox GPT-MSA	10,500	0.206	61.58	94.15	0.924
AraDetox GPT-Gulf	10,500	0.294	81.24	115.79	0.930
AraDetox GPT-Lev.	10,500	0.274	80.66	116.14	0.930
AraDetox GPT-Egy.	10,500	0.254	84.72	120.51	0.927
AraDetox Gemini-MSA	10,500	0.201	83.61	119.38	0.919
AraDetox Gemini-Gulf	10,500	0.234	83.90	122.94	0.924
AraDetox Gemini-Lev.	10,500	0.244	78.88	117.08	0.928
AraDetox Gemini-Egy.	10,500	0.219	81.26	120.85	0.925

Table 7: Comparison of MultiParaDetox-Ar and AraDetox.

We further compare the datasets using negative-sentiment predictions from three CAMeLBERT sentiment models. Table 8 should be interpreted with caution because the source datasets differ substantially. Only around 10–11% of MultiParaDetox-Ar source texts are classified as negative, compared with 72–75% for AraDetox, reflecting the more strongly negative nature of the harmful-language corpora from which AraDetox was derived.

Even with these differences, distinct patterns emerge. MultiParaDetox-Ar shows relatively small reductions in negative sentiment (1.22–2.39 percentage points), whereas AraDetox MSA variants produce much larger reductions. GPT-MSA reduces negative predictions by 20.51–23.21 percentage points across the three sentiment models, while Gemini-MSA achieves reductions of 15.21–19.22 points. The dialectal variants show smaller reductions, typically between 1.80 and 7.23 points. These results reinforce the distinction between detoxification and sentiment modification. AraDetox is not designed to convert negative opinions into positive ones, but to express criticism, disagreement, or opposition without insults, slurs, threats, or abusive language. As a result, many detoxified outputs remain negative in sentiment while no longer containing harmful expressions.

Model	Dataset	Orig.	Detox	Red.
DA	MultiParaDetox-Ar	9.92	7.53	2.39
	GPT-MSA	74.29	53.77	20.51
	GPT-Gulf	74.29	70.32	3.96
	GPT-Lev.	74.29	71.14	3.14
	GPT-Egy.	74.29	69.38	4.90
	Gemini-MSA	74.29	55.79	18.50
	Gemini-Gulf	74.29	68.07	6.22
	Gemini-Lev.	74.29	70.13	4.15
	Gemini-Egy.	74.29	70.07	4.22
MSA	MultiParaDetox-Ar	11.00	9.78	1.22
	GPT-MSA	75.47	53.48	21.99
	GPT-Gulf	75.47	71.23	4.24
	GPT-Lev.	75.47	73.67	1.80
	GPT-Egy.	75.47	70.44	5.03
	Gemini-MSA	75.47	60.26	15.21
	Gemini-Gulf	75.47	69.69	5.78
	Gemini-Lev.	75.47	72.24	3.23
	Gemini-Egy.	75.47	70.21	5.26
Mix	MultiParaDetox-Ar	10.14	7.78	2.36
	GPT-MSA	72.41	49.20	23.21
	GPT-Gulf	72.41	68.62	3.79
	GPT-Lev.	72.41	69.56	2.85
	GPT-Egy.	72.41	66.36	6.05
	Gemini-MSA	72.41	53.19	19.22
	Gemini-Gulf	72.41	65.45	6.96
	Gemini-Lev.	72.41	67.42	4.99
	Gemini-Egy.	72.41	65.18	7.23

Table 8: Negative sentiment before and after detoxification using CAMeLBERT-DA, MSA and Mix.

5 Conclusion

We introduce *AraDetox*, a large-scale Arabic detoxification dataset containing 10,500 harmful social-media posts and 84,000 detoxified rewrites across MSA, Gulf, Levantine, and Egyptian Arabic. The resource combines LLM-assisted generation with automatic validation, native-speaker quality control, and independent human evaluation. The findings show that Arabic detoxification frequently requires substantial reformulation while preserving the source claim, target, stance, and communicative intent. Human evaluation indicates high rates of offence removal and meaning preservation, although some outputs introduce additional content. The dialectal variants also show measurable corpus-level alignment with their intended Arabic varieties. AraDetox provides a substantial resource for developing and evaluating Arabic detoxification systems, while highlighting the need for continued work on dialect validation, pragmatic meaning preservation, and downstream evaluation.

6 Limitations

This study has several limitations. First, AraDetox was constructed primarily through GPT-5 and Gemini 2.5 Flash rather than large-scale human rewriting. Although automatic validation and native-speaker quality control were applied, the outputs may retain model-specific stylistic patterns and should not be treated as naturally occurring human rewrites.

Second, detailed human evaluation covered 300 source posts, corresponding to 2,400 generated outputs, and therefore represents only a subset of the full 84,000-output resource. Rare source types or harm categories may be underrepresented.

Third, semantic-preservation analyses rely partly on multilingual embedding models. High cosine similarity does not guarantee preservation of pragmatic meaning, sarcasm, humour, presupposition, target, or subtle stance. Human judgements partly address this limitation but cannot eliminate it.

Fourth, dialectal evaluation relies on corpus-level word and character n-gram similarity, which may be influenced by topic, register, corpus composition, and shared vocabulary across Arabic varieties. As dialect authenticity was not separately evaluated by human annotators, these results should be interpreted as evidence of stylistic alignment rather than native-like dialect use.

Fifth, comparison with MultiParaDetox-Ar is descriptive because the resources differ in size, source distributions, and annotation procedures. We also do not yet assess whether models trained or fine-tuned on AraDetox outperform those trained on existing Arabic detoxification datasets. A controlled downstream benchmark remains future work.

Finally, the dataset was generated using two proprietary frontier model families. Evaluating smaller open-weight models would help determine sensitivity to model scale, architecture, and training data. AraDetox is also derived from Arabic social-media content and may not generalise to long-form, news, or formal political discourse.

The AraDetox dataset is publicly available at `https://github.com/ArabicNLP-UK/AraDetox` to support transparency and reproducibility.

7 Ethical Considerations

This work involves harmful Arabic social-media content, including offensive, abusive, political, sectarian, and identity-related language. The purpose of the study is to develop safer Arabic NLP resources and evaluation methods for neutralising harmful language. We minimise the reproduction of harmful examples in the paper and report results in aggregate form.

The study included a human evaluation component involving up to ten native Arabic-speaking adult annotators. Ethical approval was granted by an Institutional Ethical Review Board for Biomedical Research in Vietnam. Annotators provided informed consent and were given detailed annotation guidance prior to participation. No personal, sensitive, or health-related data were collected, and all procedures were conducted in accordance with institutional policy, Vietnamese regulations, and internationally recognised ethical standards.

The detoxified outputs should not be treated as authoritative or universally acceptable rewrites, since judgements of harmfulness and acceptable neutralisation are shaped by social, cultural, dialectal, and political context. We also recognise that automatic detoxification can introduce risks, including over-sanitising legitimate political expression, weakening the speaker's intended stance, or removing important evidence of abuse. Any practical deployment of Arabic detoxification systems should therefore include human oversight and clear moderation guidelines.

Outputs that produced refusals, malformed responses, summaries, external-observer descriptions, or substantial meaning changes were treated as invalid and regenerated during dataset construction. The released resource distinguishes the complete LLM-generated collection from the independently human-evaluated subset. Human-evaluation labels for offence removal, meaning preservation, and new-content introduction are provided for the evaluated outputs, allowing users to identify and filter unsuccessful or uncertain generations. The complete synthetic collection should not be treated as a gold-standard dataset of human-authored detoxifications.

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.

Ashraf Ahmad, Mohammad Azzeh, Eman Alnagi, Qasem Abu Al-Haija, Dana Halabi, Abdullah Aref, and Yousef AbuHour. 2024. Hate speech detection in the arabic language: corpus design, construction, and evaluation. *Frontiers in Artificial Intelligence*, 7:1345445.

Salim Al Mandhari, Mo El-Haj, and Paul Rayson. 2024. Is it offensive or abusive? an empirical study of hateful language detection of arabic social media texts. In *Proceedings of the First International Conference on Natural Language Processing and Artificial Intelligence for Cyber Security*, pages 137–146.

Nuha Albadi, Mohamed Kurdi, and Swapna Mishra. 2018. Are they our brothers? analysis and detection of religious hate speech in the arabic twittersphere. In *Proceedings of AICS*.

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.

Safa Alsafari, Samira Sadaoui, and Malek Mouhoub. 2020. Hate and offensive speech detection on arabic social media. *Online Social Networks and Media*, 19:100096.

Raghad Alshaalan and Hend Al-Khalifa. 2020. Hate speech detection in saudi twittersphere: A deep learning approach. In *Proceedings of the fifth Arabic natural language processing workshop*, pages 12–23.

Ghadah Alwakid, Taha Osman, Mahmoud El Haj, Saad Alanazi, Mamoona Humayun, and Najm Us Sama. 2022. Muldasa: Multifactor lexical sentiment analysis of social-media content in nonstandard arabic social media. *Applied Sciences*, 12(8):3806.

Mohamed Seghir Hadj Ameur and Hassina Aliane. 2021. Aracovid19-mfh: Arabic covid-19 multi-label fake news & hate speech detection dataset. *Procedia Computer Science*, 189:232–241.

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.

Daryna Dementieva, Nikolay Babakov, and Alexander Panchenko. 2024. MultiParaDetox: Extending text detoxification with parallel data to new languages. 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 124–140, Mexico City, Mexico. Association for Computational Linguistics.

Daryna Dementieva, Nikolay Babakov, Amit Ronen, Abinew Ali Ayele, Naquee Rizwan, Florian Schneider, Xintong Wang, Seid Muhie Yimam, Daniil Moskovskiy, Elisei Stakovskii, Eran Kaufman, Ashraf Elnagar, Animesh Mukherjee, and Alexander Panchenko. 2025. Multilingual and explainable text detoxification with parallel corpora. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 7998–8025, Abu Dhabi, UAE. Association for Computational Linguistics.

Mahmoud El-Haj, Paul Rayson, Steve Young, Andrew Moore, Martin Walker, Thomas Schleicher, and Vasiliki Athanasakou. 2016. Learning tone and attribution for financial text mining. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1820–1825.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. Chatgpt outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30):e2305016120.

Hatem Haddad, Hala Mulki, and Asma Oueslati. 2019. T-hsab: A tunisian hate speech and abusive dataset. In *International conference on Arabic language processing*, pages 251–263. Springer.

Myra S Hunter, Mahmoud El-Haj, Eleanor Thorne, Amanda Griffiths, and Claire Hardy. 2023. # menopause: Examining the frequency of communications about menopause on twitter between 2014 and 2022. *Maturitas*, 177:107806.

Hung Nguyen Huy, Mo El-Haj, Dawn Knight, and Paul Rayson. 2026. Freetxt-vi: A benchmarked vietnamese-english toolkit for segmentation, sentiment, and summarisation. *arXiv preprint arXiv:2603.05690*.

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.

Varvara Logacheva, Daryna Dementieva, Sergey Ustyantsev, Daniil Moskovskiy, David Dale, Irina Krotova, Nikita Semenov, and Alexander Panchenko. 2022. ParaDetox: Detoxification with parallel data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6804–6818, Dublin, Ireland. Association for Computational Linguistics.

Lin Long, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. 2024. On llms-driven synthetic data generation, curation, and evaluation: A survey. In *Findings of the Association*

for Computational Linguistics: ACL 2024, pages 11065–11082.

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.

Hamdy Mubarak, Ammar Rashed, Kareem Darwish, Younes Samih, and Ahmed Abdelali. 2021. Arabic offensive language on Twitter: Analysis and experiments. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 126–135, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Hala Mulki and Bilal Ghanem. 2021. Let-mi: An arabic levantine twitter dataset for misogynistic language. *arXiv preprint arXiv:2103.10195*.

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*, pages 111–118.

Mihai Nadăş, Laura Dioşan, and Andreea Tomescu. 2025. Synthetic data generation using large language models: Advances in text and code. *IEEE Access*.

Nedjma Ousidhoum, Zizheng Lin, Hongming Zhang, Yangqiu Song, and Dit-Yan Yeung. 2019. Multilingual and multi-aspect hate speech analysis. *arXiv preprint arXiv:1908.11049*.

Manuel Tonneau, Diyi Liu, Samuel Fraiberger, Ralph Schroeder, Scott A. Hale, and Paul Röttger. 2024. From languages to geographies: Towards evaluating cultural bias in hate speech datasets. In *Proceedings of the 8th Workshop on Online Abuse and Harms (WOAH 2024)*, pages 283–311, Mexico City, Mexico. 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*.

Zheng-Xin Yong, Cristina Menghini, and Stephen Bach. 2024. Lexc-gen: Generating data for extremely low-resource languages with large language models and bilingual lexicons. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 13990–14009.

Omar F. Zaidan and Chris Callison-Burch. 2014. Arabic dialect identification. *Computational Linguistics*, 40(1):171–202.

Appendix

A Detoxification Prompt

For each source post, GPT-5 and Gemini 2.5 Flash were instructed to generate detoxified rewrites in Modern Standard Arabic (MSA), Gulf Arabic, Levantine Arabic, and Egyptian Arabic. The prompt emphasised meaning preservation while removing harmful language.

> Rewrite the following Arabic social-media post into four civil and non-offensive Arabic versions:
>
> 1. Modern Standard Arabic (MSA)
> 2. Gulf Arabic
> 3. Levantine Arabic
> 4. Egyptian Arabic
>
> Preserve the original meaning, target, stance, criticism, sentiment, and communicative intent.
>
> Remove profanity, insults, slurs, dehumanising language, discriminatory expressions, threats, calls for violence, and unnecessarily inflammatory wording.
>
> Do not defend, refute, explain, summarise, fact-check, or respond to the statement. Rewrite directly from the perspective of the original author.
>
> Produce natural social-media language in the requested variety and return the output as JSON:

```
{
  "msa_detox": "...",
  "gulf_detox": "...",
  "levantine_detox": "...",
  "egyptian_detox": "..."
}
```

B Generation Procedure and Quality Control

Detoxified rewrites were generated using GPT-5 and Gemini 2.5 Flash through their respective APIs. Generation was performed in batches using structured JSON output to ensure consistent formatting across all generated variants. The final resource contains 42,000 GPT-generated rewrites and 42,000 Gemini-generated rewrites, corresponding to eight detoxified variants for each of the 10,500 source posts. Additional generations were performed during prompt development, quality control, and regeneration of invalid outputs. GPT-5 and Gemini 2.5 Flash were selected as representatives of two contemporary frontier LLM families with strong multilingual generation capabilities. Gemini 2.5 Flash provides a favourable trade-off between generation

Parameter	GPT-5	Gemini 2.5 Flash
API access	OpenAI API	Gemini API
Temperature	API default	API default
Top-p	API default	API default
Maximum output tokens	API default	API default
Structured JSON output	Yes	Yes
Prompt development	Iterative pilot testing	Iterative pilot testing
Automatic regeneration	Yes	Yes
Human quality control	Yes	Yes

Table 9: Generation settings used during dataset construction. Parameters not explicitly specified in the API requests used the provider's default values.

quality, latency, and cost for large-scale dataset construction, while GPT-5 was included to provide a complementary model family and enable cross-model comparisons. Our objective was not to compare LLM performance, but to investigate whether meaning-preserving detoxification remained consistent across different model families and Arabic language varieties.

Prompt development. The final prompt was developed through iterative pilot testing on 100 source posts. Earlier prompt versions were revised to reduce safety refusals, summaries, external-observer responses, malformed JSON, unsupported additions, and changes to the original target or stance. The final prompt was selected because it produced the most consistent structured outputs while preserving the source perspective and communicative intent. Table 9 summarises the generation settings used for both models.

The generation scripts implemented automatic validation checks to detect missing fields, malformed JSON, duplicated outputs, null values, and incomplete generations. Outputs that failed these checks were automatically flagged for regeneration. Retry mechanisms were also implemented to handle API failures, empty responses, malformed outputs, and safety-related refusals. Following generation, each batch underwent a quality-control stage performed by a native Arabic speaker. The objective of this stage was not to rewrite or annotate the outputs, but to verify that the generation process had been completed correctly and that the resulting texts were suitable for inclusion in the dataset.

Quality control focused on several aspects. The reviewer checked that each generated output corresponded to the correct source post and preserved its intended meaning, target, stance, and communicative intent. They also verified that the MSA, Gulf, Levantine, and Egyptian variants reflected the intended language variety and exhibited natural linguistic usage. The reviewer inspected the outputs for technical issues, including alignment errors between source posts and generated rewrites, inconsistencies across dialectal variants, and generation failures that were not detected automatically.

A particular challenge arose from the safety mechanisms of contemporary LLMs. When presented with highly offensive or sensitive content, models occasionally refused the task and produced safety-oriented responses rather than detoxified rewrites (e.g. *I cannot fulfil this request. I am programmed to be a helpful and harmless AI assistant*). In other cases, the models adopted an external observer perspective by describing the content of a post rather than rewriting it from the perspective of the original author (e.g. *The speaker appears unhappy with the government's stance*). Such outputs were considered invalid because they failed to perform the intended detoxification task.

When these issues occurred, the affected instances were regenerated and, where necessary, additional clarification was provided that the task formed part of a research experiment focused on meaning-preserving detoxification. The reviewer also checked for hallucinated content, omissions of important information, and cases where the generated text substantially altered the meaning of the original post.

This quality-control stage served as a human verification process rather than an annotation exercise. The reviewer did not manually rewrite the posts. Their role was to ensure that the LLMs correctly executed the detoxification task and that the generated outputs were complete, aligned with the source posts, linguistically appropriate, and suitable for subsequent analysis.

C Experimental Pipeline

The experimental pipeline used in this paper consists of the following stages:

1. Collect harmful Arabic social-media posts from the Arabic Hate Speech Superset.
2. Generate detoxified Modern Standard Arabic (MSA) rewrites using GPT-5 and Gemini 2.5 Flash.
3. Generate detoxified Gulf Arabic rewrites using GPT-5 and Gemini 2.5 Flash.
4. Generate detoxified Levantine Arabic rewrites using GPT-5 and Gemini 2.5 Flash.
5. Generate detoxified Egyptian Arabic rewrites using GPT-5 and Gemini 2.5 Flash.
6. Compute lexical-change metrics between orig-

inal and detoxified texts.
7. Compute semantic similarity using multilingual E5 embeddings.
8. Visualise embedding spaces using UMAP.
9. Measure sentiment shift using three CAMeL-BERT sentiment models.
10. Evaluate dialectal style similarity using authentic Arabic dialect corpora.
11. Conduct human evaluation using three native Arabic-speaking annotators.
12. Compute inter-rater agreement and aggregate evaluation statistics.

D Human Annotation Guidelines

Human annotators assessed each generated output according to the following criteria:

1. **Offence Removal**: whether harmful language present in the source text was successfully removed.
2. **Meaning Preservation**: whether the main meaning and communicative intent of the source text were preserved.
3. **New Content**: whether the generated output introduced information not present in the source text.

Offence removal was annotated using a three-way scheme:

- 0 = offence not removed,
- 1 = offence removed,
- 2 = not applicable (no offence present in the source text).

All remaining criteria were annotated using binary judgements, where 1 indicates that the criterion was satisfied and 0 indicates that it was not.

E Human-Evaluation Sample Distribution

Table 10 compares the distribution of the complete AraDetox source collection with that of the 300-post human-evaluation sample.

Source dataset	Evaluation sample	Evaluation sample (%)
JHSC	30	10.00
MLMA	30	10.00
T-HSAB	30	10.00
L-HSAB	30	10.00
AraCOVID19-MFH	30	10.00
Brothers	30	10.00
Saudi Tweets	30	10.00
Alsafari et al.	30	10.00
Let-Mi	30	10.00
OSACT	30	10.00
Total	**300**	**100.00**

Table 10: Distribution of the 300 source posts included in the human-evaluation sample, with 30 posts sampled from each dataset.

F Qualitative Examples

Tables 11 and 12 present representative examples from AraDetox. The examples illustrate a range of detoxification behaviours, including insult mitigation, meaning-preserving reformulation, stance preservation, and dialect-specific adaptation. The examples also highlight differences between GPT-5 and Gemini, with some outputs favouring direct lexical substitution and others employing broader paraphrastic rewriting.

Source	GPT MSA	GPT Gulf	GPT Levantine	GPT Egyptian
انت انسانه تافهه	أنتِ شخصية سطحية	تصرفك سطحي ومخيّب للآمال	تصرفك سطحي ومخيّب للآمال	تصرفك سطحي ومخيّب للآمال
الملاحده بقر ياخي	أرى أن بعض الملحدين يفتقرون إلى المنطق في طرحهم لهذه القضايا	الملاحدة تفكيرهم سطحي يا أخي، ونقاشهم دايم يلف ويدور	الملحدين تفكيرهم سطحي يا زلمة، ونقاشهم دايم بلف وبدور	الملحدين تفكيرهم سطحي يا جدع، وكلامهم كتير بيلف ويدور
حمير	أشخاص يفتقرون إلى الحد الأدنى من الوعي والفهم	تصرفاتكم غير مقبولة ومحبطة	تصرفاتكم فعلاً مستفزة ومخيبة	تصرفاتكم دي مستفزة ومخيبة
كلي خرا مع محبتي	مع كامل الاحترام، لا أوافقك الرأي	مع محبتي، تصرفك مستفز وما كان له داعي	مع محبتي، تصرفك مستفز وما الو داعي	مع محبتي، تصرفك مستفز وملوش أي لزمة

Table 11: Representative detoxification examples from AraDetox across GPT-5 variants. The examples illustrate meaning-preserving detoxification and dialect-specific adaptation across MSA, Gulf, Levantine, and Egyptian Arabic.

Source	Gemini MSA	Gemini Gulf	Gemini Levantine	Gemini Egyptian
انت انسانه تافهه	أرى أن مقاربتكِ للأمور تفتقر إلى الجديّة والعمق	أشوف إن طريقتكِ في التعامل مع الأمور ما فيها جدية أو عمق	أنا بشوف إنه طريقتكِ بالتعامل مع الأمور ما فيها جدية أو عمق	أنا شايفة إن طريقتكِ في التعامل مع الأمور بتفتقر للجدية والعمق
الملاحده بقر ياخي	إن الملحدين يتسمون بقصور في التفكير	الملاحده عندهم قصور في فهم بعض الأمور ياخي	الملاحده تفكيرهم سطحي يا أخي	يا أخي، الواحد بيستغرب أوي من طريقة تفكير بعض الملحدين
حمير	قصور واضح في الفهم والإدراك	للأسف، يبدو أن البعض يجد صعوبة في استيعاب الأمور الواضحة	يا جماعة، في تصرفات بتدل على قلة تفكير واضحة	مفيش أي منطق في اللي بيحصل ده
كلي خرا مع محبتي	لا تضيع وقتك في هذا الهراء، مع خالص تقديري	ما عليك أمر، بلا هالحكي الفاضي، مع خالص تقديري	خلصنا من هالحكي الفاضي، مع كل احترامي	بلا كلام فارغ، مع كل احترامي

Table 12: Representative detoxification examples from AraDetox across Gemini variants. The examples illustrate meaning-preserving detoxification and dialect-specific adaptation across MSA, Gulf, Levantine, and Egyptian Arabic.

Some Dialects Are More Equal Than Others: Non-Prestigious Arabic Dialectal Bias in LLMs

Mai Mohamed Eida[1], Ryan Dolan[1], Paul de Nijs[1], Jonathan Dunn[1]
[1]University of Illinois Urbana Champaign
{maimm2, ryanpd3, pdenijs2, jedunn}@illinois.edu

Abstract

Previous work on Egyptian Arabic in NLP has focused largely on the prestigious Cairene Egyptian Arabic (CEA) dialect, resulting in a lack of representation for the less prestigious Sa'idi Egyptian Arabic (SEA) dialect both in LLM and resource development. Does this lack of representation influence an LLM's view of the acceptability of SEA (upstream), and does an upstream bias against SEA lead to worse performance (downstream)? We investigate the upstream effect of SEA dialectal features on LLM preferences in a Targeted Syntactic Evaluation (TSE) task which reveals a significant bias against SEA across multiple LLMs. We then analyze the effect of these same features on downstream model performance on MMLU benchmarks and show that models experience a degradation in performance when presented with SEA. This work highlights the need for further exploration on how sub-dialectal variation impacts language technologies.

1 Introduction

Arabic dialectal variation remains a continuous challenge in Arabic Natural Language Processing (NLP). With over twenty-six dialects, and hundreds of sub-dialects (Behnstedt and Woidich, 2013; Versteegh et al., 2006–2009), a central goal of the field has been to model dialectal varieties of Arabic across language technologies, including Large Language Models (LLMs). However, accounting for such diverse dialectal variation is difficult and does not necessarily account for sub-dialectal variation within each country, and therefore, in this process, much of sub-dialectal variation is lost in practice.

Egyptian Arabic (EA) provides a clear case for supporting marginalized sub-dialectal varieties in Arabic language technologies. Egyptian Arabic is the most thoroughly modeled dialect in Arabic NLP, but it has tacitly become synonymous with "Educated Cairene Egyptian Arabic", the most prestigious dialect (Bassiouney, 2018) spoken by middle class educated Egyptians in Cairo and surrounding urban cities (Abdel-Malek, 1972; Badawi, 1973). This variety represents only one of the five groups of Egyptian Arabic sub-dialects, which vary across phonology, morphology, syntax, semantics, and lexicon (Behnstedt and Woidich, 1985; Badawi, 1973) and also vary culturally from Cairo. In this paper, we focus on the second most spoken dialect in Egypt, Sa'idi Egyptian Arabic (SEA) which is not linguistically represented within the dialectal landscape in current Arabic language technologies (Mohamed Eida et al., 2024).

The focus on Cairene Egyptian Arabic (CEA) is consistent with accessibility and availability of resources for prestigious dialects. Dialect attitudes are mediated by socioeconomic status, educational access, and language ideology (Labov, 2006). Speakers of prestigious dialects have the means to participate in digital spaces and contribute linguistic data which increases their representation in language technologies. On the other hand, speakers of marginalized dialects may avoid using their dialects online, even when access is available. This produces a self-reinforcing cycle in which language technologies prefer certain dialects while excluding others due to lack of linguistic resources.

Previous work has demonstrated that SEA is not well represented in online Egyptian Arabic corpora (Mohamed Eida et al., 2024). However, whether this lack of representation actually results in upstream or downstream bias against SEA in model behavior has not yet been demonstrated. It is possible that, despite the scarcity of SEA representation in online corpora on which LLMs are trained, the internal representations learned by LLMs are generalized enough that they are insensitive to dialect differences. If so, models may exhibit comparable downstream performance on SEA and CEA, suggesting that the scarcity of SEA resources does not necessarily translate into a poorer experience

Proceedings of The Fourth Arabic Natural Language Processing Conference (ArabicNLP 2026), pages 48–64
October 28-29, 2026 ©2026 Association for Computational Linguistics

for SEA speakers.

This paper addresses two research questions. **First, do LLMs replicate the same bias humans have towards prestigious and non-prestigious sub-dialects in their upstream representations?** If so, this supports previous work on the lack of SEA representation in pre-training data in Arabic NLP and that such lack of representation can result in an upstream bias. In experiment 1, our results confirm there is a bias in monolingual, bilingual, and multilingual LLMs against SEA, with internal representations favoring CEA or Modern Standard Arabic (MSA) over SEA in almost every single case. **Our second research question addresses the implications of upstream bias: does this upstream bias impact the performance of LLMs on downstream tasks?** Specifically, we examine performance on Arabic Massive Multitask Language Understanding (MMLU) benchmarks. If there is no impact on downstream tasks, this may shift the focus of future work to other necessary tasks such as cultural awareness of sub-dialectal variation rather than curating SEA-specific corpora. The results from our second experiment show that downstream bias varies more across models, but still shows consistent favor towards CEA overall[1].

2 Background

2.1 Language Attitudes

Language attitudes are structured by social prestige and power (Labov, 2006). Dialect choice is not neutral. It is shaped by socially embedded evaluations that influence both perception and production. Speakers converge toward or diverge from others' speech depending on social attitudes and group identity (Giles, 1970). Perceptual dialectology shows that listeners systematically associate dialects with social attributes, assigning traits such as intelligence, competence, and professionalism to prestigious varieties, while attributing negative characteristics such as lack of education or credibility to marginalized dialects (Giles, 1970; Lambert et al., 1960; Preston, 1993).

This extends to Arabic dialects, with language attitudes spanning sub-dialects of many Arabic dialects (Aldosaree, 2016; Eltouhamy, 2015; Hachimi, 2015). As the most prestigious dialect, Cairene Egyptian Arabic (CEA) is socially constructed as educated, modern, and authoritative, while Sa'idi Egyptian Arabic is stereotyped as rural, uneducated, and socially marginal (Bassiouney, 2017; Haeri, 1997). This bias affects how speakers are evaluated and treated in education, employment, and institutional settings, with speakers of prestigious dialects being perceived as more competent and credible than speakers of stigmatized varieties. However, do LLMs exhibit a similar bias toward prestigious dialects mirroring existing human bias?

2.2 Dialectal Bias in LLMs

Recent research has shown that LLMs frequently demonstrate bias against certain language varieties across many different languages. Lin et al. (2025) show that LLMs systematically display a high degree of unfairness (performance gaps) and brittleness (sensitivity to input perturbations) when responding to reasoning queries in African American English (AAE). Similar work by Pan et al. (2025) shows that such bias also exists for multiple-choice reasoning benchmarks and extends to other English varieties such as Chicano, Appalachian, Southern, Indian, and Singaporean English. These works crucially demonstrate that even in tasks such as code generation or queries for factual world knowledge which should not reasonably vary across dialects, LLMs show a systematic bias.

Outside of benchmark performance bias, Fleisig et al. (2024) use qualitative evaluations from native speakers of 10 different varieties of English to show that GPT-3.5 and GPT-4 often use demeaning and stereotypical language when responding to varieties other than Standard American and Standard British English. They show that the models often misuse or exaggerate dialectal features in a way that is unnatural and perceived as mockery. Hassan et al. (2025) additionally show that LLMs demonstrate a systematic bias against AAE when used for sentiment analysis tasks, with AAE language being much more likely to be identified as negative. Outside of English, Bui et al. (2025) show that systematic bias also exists for non-standard dialects of German such as Bavarian and Alemannic varieties, with non-standard varieties more likely to be associated with negative adjectives in an adjective-association task. There is also work on Bengali and dialectal bias in this low-resource language based on the minorities/power structures (Sadhu et al., 2025), showing LLM production exhibit significant gender and religious bias in Bangala. In all of this research, LLMs do exhibit bias against the less prestigious varieties, and this impacts downstream

[1]All data, code, and prompts are available on https://github.com/maimm2/EA_Dialectal_Bias.

performance. Is this also the case for Egyptian Arabic?

3 Experiment 1: Do LLMs have a preference for one dialect?

Since SEA is highly under-represented compared to CEA in existing Egyptian Arabic datasets and tools (Mohamed Eida et al., 2024), is there a preference for one dialect over the other due to the scarcity of data? If so, then we expect LLMs to prefer CEA to SEA by finding CEA sentences more probable than an equivalent SEA sentence in a minimal pair.

3.1 Methodology

Models We select five pretrained models to test upstream preferences: one monolingual Arabic model specifically trained on both MSA and Tweets in Dialectal Arabic (DA) (henceforth "monolingual", Jasmine-350M: Billah Nagoudi et al. 2023), two English-Arabic bilingual models trained on a variety of Arabic dialects (henceforth "bilingual", Jais-2-8B & Jais-1-13B: Anwar et al. 2025; Sengupta et al. 2023), one multilingual model trained on generic Arabic data (henceforth "multilingual", Llama 3.1-8B: AI@Meta 2024), and one multilingual model continuously-pretrained specifically on Egyptian Arabic (henceforth "EA", NileChat-4B: Shang et al. 2025)[2].

Task To test upstream preferences, we employ Targeted Syntactic Evaluation (TSE, Linzen et al., 2016). In this approach, the probabilities of two sentences in a minimal pair are compared to decide which of the minimal pair of dialects the model prefers. We create minimal pairs between CEA and SEA by sampling items from an SEA corpus (Mohamed Eida and Habash, 2025) and changing features specific to SEA to their CEA equivalents. These features include phonetic, morphological, orthographic, and lexical variations. For most minimal pairs, only one feature at a time is changed, although we also include a "mixed" condition in which multiple features are changed at once.

Minimal Pair Construction We additionally create an ungrammatical baseline (UG) in where we replace the valid Arabic features such as morphemes and lexical items with meaningless nonce forms

[2]NileChat-4B is only continuously-pretrained on Egyptian Arabic text, but is later instruct-tuned with both Egyptian Arabic and English.

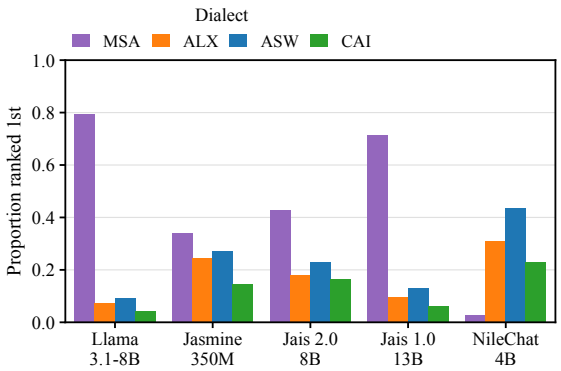


Figure 1: Proportion of dialect preference by model in MADAR, defined as being the most preferred variation in the TSE task. MSA is consistently most preferred, followed ASW (Aswan), ALX (Alexandria), then CAI (Cairo). With the exception of NileChat-4B, MSA is most preferred. NileChat-4B was precontinuously trained on Egyptian Arabic and therefore, preferring EA to MSA is expected.

that are less likely to correspond to an Arabic variety. For example, the CEA future marker هـ (h) and its SEA counterpart ع (ʕ) were replaced with د (d), which does not function as a future tense marker in Arabic dialects. We consistently replaced د (d) as future tense in every applicable case thereafter. This ensures that the modified forms preserved superficial characteristics such as length while removing their grammatical interpretation, and we control for as much variable change as possible. The UG forms generated matched the morphological, phonological, and orthographic features in length and script, as illustrated in Appendix B in Table 4. Lexical features were altered with +/-4 letters depending on how easy it was to create a nonce word. This accounts for the possibility of an LLM considering one or two letter changes in variants a typological error ("typo"). The UG baseline allows us to identify whether the model prefers CEA over SEA because it finds SEA to be purely ungrammatical, or because it considers SEA as a valid but dispreferred dialect. This yields a total of 134 minimal pairs, with samples illustrated in Table 3 in Appendix B.

Evaluation To compute sentence probabilities, we follow Leong and Linzen (2023) in our implementation of TSE by adding the (negative) log probabilities of each token in a given sentence (conditioned on previous tokens) to obtain the model's "judgment" for that sentence. We then normalize

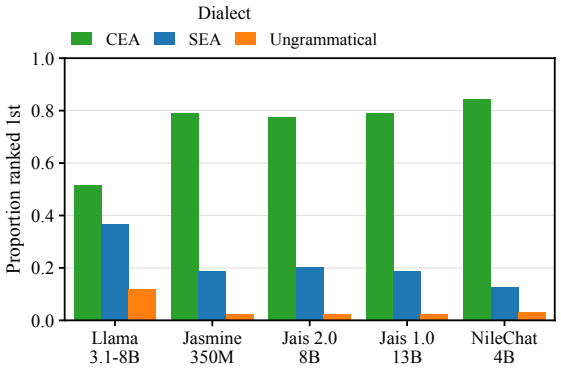


Figure 2: Proportion of dialect preference by model, defined as being the most preferred variation in the TSE task. CEA is consistently most preferred, followed by SEA and UG.

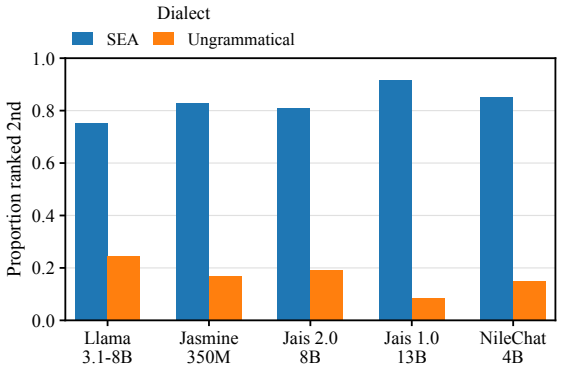


Figure 3: Second-choice model preferences when CEA is ranked most probable. All models strongly prefer SEA to the Ungrammatical baseline, indicating that SEA is viewed as legitimate, but dispreferred, variation.

this value by the length of the sequence to ensure that longer sentences are not artificially less probable.

As MSA remains the dominant Arabic variety in available NLP resources and is substantially better represented than dialectal Arabic in LLM training data (Mousi et al., 2025), we first check if MSA is the most preferred dialect over Egyptian Arabic (EA). To do so, we use the MADAR (Bouamor et al., 2018) parallel Arabic corpus and extract Arabic minimal pairs between MSA and the three Egyptian cities in the corpus: Cairo (CAI), Alexandria (ALX), and Aswan (ASW). We hypothesize that MSA will be the most preferred variety, given both its prestige and prevalence in available training data. Then, we examine which of our three CEA, SEA, and UG variants is most preferred by the model in each minimal pair. We then determine pairwise preferences between CEA and SEA and SEA and the ungrammatical baseline (UG). If the model exhibits a bias against SEA, we expect that the CEA variant will be consistently ranked the most probable sentence. If little or no bias exists, we expect that CEA will be ranked over SEA in roughly equal proportion to SEA being ranked over CEA. Furthermore, if CEA is consistently ranked higher than SEA, and SEA is consistently ranked higher than UG, this would indicate that the model recognizes SEA as grammatical, but still disprefers it to CEA. If there is no consistent preference for SEA over UG, this indicates that the model views SEA as fully ungrammatical.

3.2 Results

Is MSA more preferred than EA? As illustrated in Figure 1, MSA is the most preferred variety of Arabic when compared to Egyptian Arabic. Therefore, our hypothesis that MSA, as both the most prestigious and the most represented variety in training data, would be preferred holds across all models except NileChat-4B, where Egyptian Arabic emerges as the most preferred variety. Given that NileChat-4B is the only model continuously pre-trained specifically on Egyptian Arabic, its preference for Egyptian Arabic over MSA is expected.

Do LLMs prefer CEA over SEA? The ranking preferences of each model are given in Figure 2. In all models, a clear preference is shown for the prestigious CEA dialect over the less prestigious SEA dialect. The preference is very clear in the Arabic monolingual (Jasmine-350M), Arabic-English bilingual (Jais-1-13B and Jais-2-8B), and Egyptian Arabic (EA) model (NileChat-4B), with CEA being preferred around 80% of the time. This preference notably drops in the multilingual model (Llama-3.1-8B) to only 50%. A similar pattern is found for the ungrammatical baseline UG, which is found most preferable only around 2% of the time for all models except Llama-3.1-8B (11%). Therefore, LLMs prefer CEA over SEA and display an upstream bias against SEA.

Do LLMs consider SEA a legitimate variety? Having demonstrated that all models prefer CEA to SEA, we now ask whether this is because SEA is viewed as completely ungrammatical, or whether the model recognizes the SEA variants as being

legitimate, but still dispreferred. The results in Figure 2 suggest that SEA should be viewed as legitimate given that it is ranked above CEA in a non-negligible number of cases. Indeed, we find that in cases where CEA is ranked to be the most probable, SEA is preferred over the ungrammatical baseline in around 90% of cases (Figure 3). Thus, it is not the case that SEA is viewed as completely ungrammatical; rather, the models are simply biased towards CEA.

The absolute rankings of negative log probabilities clearly demonstrate a preference hierarchy of CEA > SEA > UG across all models, with only Llama-3.1-8B showing a somewhat weaker preference. Thus, in the majority of cases, CEA is indeed preferred over SEA, but SEA dialectal features are recognizably legitimate variations such that SEA is still systematically preferred over the UG ungrammatical baseline.

To what degree do LLMs prefer SEA over the ungrammatical baseline? Having established this preference hierarchy and the fact that SEA is not considered purely ungrammatical, we now turn to the question of whether SEA is considered closer to CEA overall than to the ungrammatical baseline. We do so by comparing the distribution of the *differences* in sequence probabilities between CEA and SEA ($P(Seq_{CEA}) - P(Seq_{SEA})$) and SEA and the ungrammatical baseline UG ($P(Seq_{SEA}) - P(Seq_{UG})$) for each minimal pair using a paired t-test. This allows us to determine the degree of preference; if, on average, $P(Seq_{CEA}) - P(Seq_{SEA}) < P(Seq_{SEA}) - P(Seq_{UG})$, this implies that the model considers SEA to be overall closer to CEA than the ungrammatical baseline. If the converse is true, this implies that the model considers SEA to be closer to the ungrammatical baseline than to CEA, even if it consistently ranks SEA higher.

The paired t-test is only significant for Jais-1-13B ($t = -2.59, p = 0.03^{*}$), which indicates that for all other models, SEA is not closer on average to CEA than to complete ungrammaticality, or vice versa. This can also be seen visually in appendix A. This shows that SEA interestingly occupies a sort of midpoint between CEA and UG in terms of grammaticality.

Does the number of characters changed matter? Finally, we assess whether the upstream bias exhibited by the models correlates with the scale of the grammatical feature change as measured by character edit distance. We correlate minimum edit distance and the difference in probabilities examined in the previous section. Edit distance was positively correlated and statistically significant with the probability difference CEA - SEA ($r = 0.18, p \approx 0.00$) and CEA - UG ($r = 0.17, p \approx 0.00$) for all models except for Llama-3.1-8B. However, SEA - UG did not show a significant correlation ($r = 0.009, p = 0.8$) with edit distance for any model. This shows that the magnitude of the change does affect the degree to which CEA is preferred over SEA, but that UG is heavily dispreferred regardless of the number of characters changed.

In summary, the consistent ranking of CEA over SEA shows a strong bias against SEA in upstream model preferences, answering our first research question. It is especially noteworthy that the minimal pairs were constructed using samples from an SEA corpus, changing only one overt SEA feature to its CEA counterpart. Despite this SEA source context, models still consistently prefer the CEA variant once that single feature is changed. In other words, the models prefer a CEA feature in an SEA context even over an SEA feature in an otherwise SEA context. Thus, the preference is strong enough that the model prefers an out-of-place prestigious dialectal feature to a well-suited non-prestigious dialectal feature. On the other hand, SEA is ranked higher than the ungrammatical baseline, demonstrating that SEA is a viable but dispreferred dialect occupying a midpoint between the prestigious CEA dialect and total ungrammaticality.

4 Experiment 2: Does Upstream Bias Impact Downstream Performance?

The results in Experiment 1 reveal a persistent upstream dialectal preference for CEA over SEA. However, does upstream bias actually impact downstream performance? Experiment 2 aims to answer this question. If so, this would indicate that marked sub-dialectal variation affects downstream performance, and emphasizes the need for sub-dialectal variation in resource development, model development, and language technologies overall.

4.1 Methodology

We adapt Pan et al. (2025)'s methodology, which evaluates the extent to which grammatical features associated with non-standard English varieties influence LLM performance on benchmarks such as Massive Multitask Language Understanding

(MMLU), Boolean Questions (BoolQ), and Science Questions (SciQ). We extend this methodology to three Arabic varieties: MSA, CEA, and SEA using the same set of features as in Experiment 1. We exclude lexical and orthographic variations because they are more difficult to consistently apply across all benchmarks. For each feature, we use a regular expression to first select all benchmark questions in which the feature occurs. We then create a variant of the question and multiple-choice answers with the target feature changed to its equivalent in MSA, CEA, and SEA. As in Experiment 1, we also produce an ungrammatical control variant (UG) as a baseline following the same rules of length, script and consistency. This transforms each original item into four versions corresponding to MSA (ArabicMMLU only), CEA, SEA, and UG conditions.

Benchmarks For this experiment, we focus on three MMLU benchmarks specifically developed for Arabic varieties. We select one MSA MMLU benchmark, ArabicMMLU (Koto et al., 2024), and two Dialectal Arabic (DA) benchmarks, AraDiCE-ArabicMMLU-EGY (Mousi et al., 2025) and DialectalArabicMMLU EGY (Altakrori et al., 2026). For the latter two benchmarks, we extract the Egyptian Arabic items only. This creates two conditions: one in which the original questions are fully in MSA, and one in which the original questions are in Egyptian Arabic. Having both types of conditions is important, because the model may behave very differently when faced with an SEA feature in MSA versus an SEA feature in Egyptian Arabic. Additionally, because MSA is a standard/neutral variety and the most preferred compared to EA as illustrated in Experiment 1, we can safely add CEA and SEA features in isolation from one another without the potential confound of an item already being written in CEA. By using MSA as the source condition in at least one benchmark, we can more directly isolate the effect of introducing CEA and SEA features while holding all other aspects of the item constant. The two Egyptian Arabic benchmarks are still included because both CEA and SEA are primarily spoken dialects, and thus provide a more natural context in which SEA features would actually occur. We list the benchmarks and the 24 extracted features in Table 4 in Appendix B.

Models We evaluate benchmark performance on three of the five models examined in Experiment 1: Llama 3.1-8B (AI@Meta, 2024), Jais-2-8B (Anwar et al., 2025), and NileChat-4B (Shang et al., 2025). Jasmine-350M (Billah Nagoudi et al., 2023) was excluded because it was not instruct-tuned, and Jais-1-13B (Sengupta et al., 2023) was excluded because the results of Experiment 1 were extremely similar to those of the newer Jais-2-8B.

Evaluation We initially evaluate each grammatical feature in isolation, modifying a single feature at a time and measuring benchmark accuracy across the MSA, CEA, SEA, and UG conditions. We then introduce a "mixed-features" condition, in which two or more grammatical features are changed simultaneously within the same benchmark item. Some feature perturbations involve only minimal changes, sometimes only a single character. It is possible that models treat these forms as spelling variations or noise rather than meaningful dialectal distinctions. This is corroborated by the positive correlation between edit distance and difference in upstream preference in Experiment 1 between CEA and SEA. Therefore, evaluating both isolated and combined feature substitutions we can determine whether minimal dialectal perturbations affect model accuracy to the same degree as larger dialectal perturbations.

5 Results

Single Feature Accuracy Overall The results for the single feature condition are given in Table 1. We first report the accuracy across all items on each benchmark for each model and dialect condition ("Overall Accuracy"). The results show that across almost all models and benchmarks, the model achieves higher accuracy when the questions and answers are in CEA than when they are in SEA. The magnitude of difference is not particularly consistent across models, with different models being impacted more across different benchmarks. Furthermore, the original dialect of the question does not appear to impact the degree of difference, with ArabicMMLU (MSA) having a similar margin of difference as compared to AraDiCE-ArabicMMLU-EGY and DialectalArabicMMLU (Egyptian Arabic).

Although higher accuracy is observed on CEA as compared to SEA, the margin is often small (and in fact there is one marginally positive case in favor of SEA: Jais-2-8B in the DialectalArabicMMLU benchmark). This is consistent with the results of Pan et al. (2025), who observe that non-prestigious English varieties degrade performance

		ARABIC MMLU MSA (2024) % (n=9689)			ARDICE MMLU EGY (2025) % (n=13400)			DIALECTALARABICMMLU (2026) % (n=6739)		
		JAIS 2.0 (8B)	LLAMA 3.0 (8B)	NileChat (4B)	JAIS 2.0 (8B)	LLAMA 3.0 (8B)	NileChat (4B)	JAIS 2.0 (8B)	LLAMA 3.0 (8B)	NileChat (4B)
Overall Accuracy	MSA	**65.86% (+0.46%)**	43.60% (-0.21%)	46.39% (+0.52%)	-	-	-	-	-	-
	CEA	65.39%	**43.80%**	**45.88%**	**57.45%**	**37.43%**	**43.54%**	42.25%	**34.59%**	**43.10%**
	SEA	64.28% (-1.11%)	43.03% (-0.77%)	45.29% (-0.59%)	56.05% (-1.40%)	36.77% (-0.66%)	43.05% (-0.49%)	**42.34% (+0.09%)**	34.13% (-0.46%)	41.39% (-1.71%)
	UNG	62.41% (-2.98%)	42.63% (-1.18%)	44.39% (-1.49%)	53.31% (-4.13%)	36.59% (-0.84%)	41.22% (-2.32%)	41.00% (-1.34%)	33.70% (-0.43%)	41.43% (-1.68%)
CEA 100% Accuracy	MSA	99.07% **(-0.93%)**	98.39% **(-1.61%)**	72.50% **(-27.50%)**	-	-	-	-	-	-
	CEA	100% (n=6336)	100% (n=4224)	100% (n=4495)	100% (n=7698)	100% (n=5015)	100% (n=5834)	100% (n=2847)	100% (n=2331)	100% (n=2238)
	SEA	94.27% (-5.73%)	98.39% **(-1.61%)**	70.43% (-29.57%)	92.61% **(-7.39%)**	91.74% **(-8.26%)**	67.95% **(-32.05%)**	93.64% **(-6.36%)**	94.85% **(-5.15%)**	65.19% **(-34.81%)**
	UNG	90.48% (-9.52%)	91.41% (-8.59%)	68.12% (-31.88%)	85.80% (-14.20%)	88.91% (-11.09%)	62.82% (-37.18%)	87.64% (-12.36%)	93.31% (-6.69%)	64.34% (-35.66%)
SEA 100% Accuracy	MSA	95.95% **(-4.05%)**	94.22% **(-5.78%)**	71.79% **(-28.21%)**	-	-	-	-	-	-
	CEA	95.91% (-4.09%)	94.58% (-5.42%)	72.15% (-27.85%)	94.91% **(-5.09%)**	93.38% **(-6.62%)**	68.71% **(-31.29%)**	93.45% **(-6.55%)**	96.13% **(-3.87%)**	67.89% **(-32.11%)**
	SEA	100% n=(6228)	100% n=(4169)	100% (n=4388)	100% (n=7511)	100% (n=4927)	100% (n=5769)	100% (n=2853)	100% (n=2300)	100% (n=2149)
	UNG	92.68% (-7.32%)	93.84% (-6.16%)	68.69% (-31.31%)	88.63% (-11.37%)	92.10% (-7.9%)	64.21% (-35.79%)	88.78% (-11.22%)	94.91% (-5.09%)	65.57% (-34.43%)

Table 1: **Top:** Overall accuracy on single grammatical feature minimal pairs. **Middle:** Accuracy on the corresponding MSA, SEA, and ungrammatical (UG) variants restricted to items answered correctly in the CEA condition (100% CEA subset). **Bottom:** Accuracy on the corresponding MSA, SEA, and ungrammatical (UG) variants restricted to items answered correctly in the SEA condition (100% SEA subset). Unperturbed items are excluded from the overall evaluation. Random-chance accuracy is 29% for ArabicMMLU and AraDiCE-ArabicMMLU-EGY, and 25% for DialectalArabicMMLU. Bold indicates the highest accuracy within each model and benchmark column. Values in parentheses indicate the absolute percentage-point difference relative to the corresponding CEA-correct or SEA-correct baseline performance.

at a similarly small margin (between -0.5% and -5%). However, it is important to note that finding downstream performance degradation overall (though marginal) in favor of CEA is still noteworthy given that we focus on a single grammatical perturbation at a time, and do not include any syntactic, lexical or orthographic changes in the perturbations. This could indicate that more SEA linguistic dialect markers could contribute in further performance degradation in downstream performance when the full dialect is used. Another possible reason for the marginal difference is that benchmark performance is generally quite low across most models, peaking at around 66% in the best case. Thus, it may be that many cases in which the model answers a question in SEA incorrectly can be attributed to general benchmark difficulty, and not difficulty specific to understanding the dialect. We therefore follow Pan et al. (2025) in also examining the difference in accuracy on the questions which the model got correct in the preferred dialect (in this case, CEA). In other words, if the model got the question correct in the preferred dialect, how likely is the model to get that same question wrong if SEA is used instead and vice versa?

Single Feature Accuracy - CEA/SEA Correct Only Examining only the questions which the model got correct in CEA, the difference between CEA and SEA becomes much more severe. NileChat-4B in particular exhibits an extremely strong effect, with performance dropping nearly 35% when the question is posed in SEA. Llama-3.1-8B by comparison shows the smallest differences almost across the board, which is consistent with its comparative lack of upstream bias observed in Experiment 1. However, these differences are certainly not negligible, with performance degrading up to 8.3% on questions that were answered correctly in CEA. Lastly, Jais-2-8B exists at somewhat of a midpoint, sometimes patterning closely to Llama-3.1-8B and other times showing a stronger bias, but still far less than NileChat-4B. This shows that having an extremely strong upstream preference does not necessarily lead to a proportional impact on downstream performance.

To ensure the robustness of our results, we extend Pan et al. (2025)'s approach looking at prestigious-correct only items to also examining items which the model got correct in the non-preferred dialect (SEA) as well. Examining both

		ARABIC MMLU MSA (2024) % (n=2194)			ARDICE MMLU EGY (2025) % (n=3200)			DIALECTALARABICMMLU (2026) % (n=1844)		
		JAIS 2.0 (8B)	LLAMA 3.0 (8B)	NileChat (4B)	JAIS 2.0 (8B)	LLAMA 3.0 (8B)	NileChat (4B)	JAIS 2.0 (8B)	LLAMA 3.0 (8B)	NileChat (4B)
Overall Accuracy	MSA	**66.00% (+1.55%)**	**45.49% (+6.84%)**	**45.76% (+2.55%)**	-	-	-	-	-	-
	CEA	64.45%	38.65%	43.21%	**57.19%**	**37.22%**	**43.63%**	**43.93%**	**36.44%**	**44.31%**
	SEA	61.17% (-3.28%)	36.65% (-2.01%)	41.29% (-1.91%)	53.25% (-3.94%)	36.44% (-0.78%)	40.03% (-3.59%)	42.46% (-1.46%)	35.03% (-1.41%)	41.59% (-2.71%)
	UNG	55.88% (-8.57%)	34.78% (-3.87%)	37.60% (-5.61%)	48.59% (-8.59%)	36.47% (-0.75%)	38.06% (-5.56%)	39.05% (-4.88%)	35.79% (-0.65%)	37.47% (-6.83%)
CEA 100% Accuracy	MSA	96.68% **(-3.32%)**	91.86% **(-8.14%)**	71.94% **(-28.06%)**	-	-	-	-	-	-
	CEA	100% n=(1647)	100% n=(1003)	100% (n=1100)	100% (n=2223)	100% (n=1450)	100% (n=1696)	100% (n=1259)	100% (n=1053)	100% (n=1270)
	SEA	88.47% (-11.53%)	85.26% (-14.74%)	64.14% (-35.86%)	86.17% **(-13.83%)**	87.49% **(-12.51%)**	64.04% **(-35.96%)**	85.31% **(-14.69%)**	87.80% **(-12.20%)**	64.87% **(-35.13%)**
	UNG	79.42% (-20.58%)	80.07% (-19.93%)	56.65% (-43.35%)	74.10% (-25.90%)	83.63% (-16.37%)	58.81% (-41.19%)	74.69% (-25.31%)	85.57% (-14.43%)	55.57% (-44.43%)
SEA 100% Accuracy	MSA	92.66% **(-7.34%)**	86.36% (-13.64%)	69.61% (-30.39%)	-	-	-	-	-	-
	CEA	92.65% (-7.35%)	89.93% **(-10.07%)**	66.67% **(-33.33%)**	92.36% **(-7.64%)**	89.07% **(-10.93%)**	68.93% **(-31.07%)**	88.21% **(-11.79%)**	91.13% **(-8.87%)**	69.30% **(-30.7%)**
	SEA	100% n=(1566)	100% n=(953)	100% (n=1053)	100% (n=2055)	100% (n=1427)	100% (n=1564)	100% (n=1204)	100% (n=1015)	100% (n=1176)
	UNG	84.22% (-15.78%)	84.78% (-15.22%)	58.31% (-41.69%)	80.92% (-19.08%)	87.74% (-12.26%)	58.89% (-41.11%)	78.49% (-21.51%)	89.06% (-10.94%)	57.99% (-42.01%)

Table 2: **Top:** Overall accuracy on 2+ mixed grammatical feature minimal pairs. **Middle:** Accuracy on the corresponding MSA, SEA, and ungrammatical (UG) variants restricted to items answered correctly in the CEA condition (100% CEA subset). **Bottom:** Accuracy on the corresponding MSA, SEA, and ungrammatical (UG) variants restricted to items answered correctly in the SEA condition (100% SEA subset). Unperturbed items are excluded from the overall evaluation. Random-chance accuracy is 29% for ArabicMMLU and AraDiCE-ArabicMMLU-EGY, and 25% for DialectalArabicMMLU. Bold indicates the highest accuracy within each model and benchmark column. Values in parentheses indicate the absolute percentage-point difference relative to the corresponding CEA-correct and SEA-correct performance.

CEA-correct and SEA-correct items ensures that our analysis does not presuppose that there are no items that the model got correct in CEA but not SEA. We observe in Table 1 that the absolute magnitudes of the number of questions got correct on each benchmark is quite similar between both CEA and SEA. It will therefore be useful to validate the systematicity of the downstream bias with statistical means, since it not clear from the counts/proportions alone that the bias is significant. We will return to this point after reviewing the mixed-feature results.

Mixed-Feature Accuracy Turning to the mixed feature conditions (Table 2), we observe that many of the same patterns hold, but are more pronounced. Both the overall and CEA-correct accuracies are lower in the mixed feature condition than in the single feature condition across all models and benchmarks. Therefore, as more features associated with SEA are introduced into the prompt, the more likely the model is to get the question wrong. This could be for a few different reasons: one reason is simply that the more dialectal features are introduced into the prompt, the more probable it is that the model will encounter a feature with which it is less familiar. Another possibility is that introducing more features makes it more likely that a feature which materially affects the question/answers is changed. For example, changing the plurality of a single noun in the question may not affect the model's ability to recognize the overall meaning of the prompt, whereas replacing a negation marker with one which is no longer comprehensible to the model may impact model performance significantly.

Which models have a significant bias against SEA in downstream performance? We have so far demonstrated that across nearly all models and benchmarks, questions posed to the model in CEA are more likely to be answered correctly than those in SEA. However, the margin of difference for many models and benchmarks is marginal, sometimes within one percent. Moreover, the CEA-correct patterns mirror that of SEA-correct patterns, and therefore, how do we know this is attributed to dialect? To determine which models have significant bias against SEA in downstream performance, we perform McNemar's test between each pair of dialects, treating each question as a sample and whether the question was answered correctly

by the model as the response variable (Pan et al., 2025). We perform the paired tests independently for each model, benchmark, and number of features changed (single vs. mixed). If the result of the test is significant for that pair of dialects, this substantiates that the apparent presence of bias found in the accuracy score differential is not due to chance.

The results show that across nearly all benchmarks, Jais-2-8B exhibits the most consistent bias, with the proportion of correct answers in CEA being significantly higher than those in SEA. The only exception is in the DialectalArabicMMLU benchmark, in which it only becomes significant after three or more SEA features are present. NileChat-4B exhibits a significant bias in favor of CEA on DialectalArabicMMLU, as well as a significant bias in the case where two or more SEA features are present, yet the effect is no longer significant when three or more features are present. In all other benchmarks, the presence of SEA features is not significantly correlated with lower performance on the benchmark. Llama-3.1-8B exhibits the least significant downstream bias, with SEA features resulting in lower benchmark accuracy only in the mixed feature case for ArabicMMLU and in the single feature case for AraDiCE-ArabicMMLU-EGY. Interestingly, in both cases the effect becomes no longer significant once more SEA features are introduced (three or more features for ArabicMMLU, two or more features for AraDiCE-ArabicMMLU-EGY). These results show that the effect of upstream bias on downstream performance varies substantially by model, although no model is completely unanimous across conditions in preferring CEA over SEA or exhibiting no preference.

6 Discussion

The experiments in this paper have shown that the lack of SEA representation in text corpora used to train LLMs leads these models to view sentences with SEA features as less probable than their prestigious CEA counterparts, and that this upstream bias can lead to further impact downstream. It is therefore not the case that we can purely rely on the model's ability to generalize from massive amounts of text in ensuring that the model is equitable towards all Egyptian Arabic speakers.

This demonstration of bias leads us to assert the importance of not viewing dialect too reductively. Although ongoing efforts in Arabic NLP have aimed to face the diglossic nature of the language head-on, a lack of attention to the underlying makeup of major regional dialects implicitly favors the prestigious, "default" dialect. As was mentioned, this leads to an unfortunate feedback loop: the less that existing language technologies support SEA or other non-prestigious sub-dialects, the more likely SEA speakers are to avoid using their native dialect in digital spaces, resulting in less data on which to train and improve the model for that population.

7 Conclusion and Future Work

Given the importance of accounting for sub-dialectal variation, the next step is identifying the existing varieties within each Arabic speaking country, the degree of variation, and whether this variation exists in the upstream representations and affects the downstream performance of LLMs for this dialect. This may require further efforts beyond available digital data, as some of these non-prestigious dialectal speakers could prefer not to position themselves as speakers of these marginalized dialects online, and accordingly we lack the necessary representative data. The present study has focused on dialectal features in an aggregate sense, without particular consideration for exactly which features drive upstream and downstream bias. Future work should aim to establish a causal relationship between individual features and model bias in order to develop a more comprehensive characterization of the problem.

Limitations

The selected grammatical features distinguishing SEA and CEA were drawn from dialect surveys conducted in the 1980s (Behnstedt and Woidich, 1985; Khalafallah, 1969). As these descriptions were collected several decades ago, it is possible that some reported SEA specific features have changed in frequency or fallen out of use over time. To mitigate this limitation, we cross-referenced all selected features against a recently collected spoken Sa'idi Arabic corpus (Mohamed Eida and Habash, 2025) and adjusted the feature set to reflect their observed frequency and usage. We then used regular expressions to automatically generate the MSA, CEA, SEA, and ungrammatical variants by replacing the selected grammatical features, followed by manual validation of the resulting items. Although every generated item was reviewed, some unintended substitutions or annotation errors may

have remained despite our best efforts.

Another limitation of our work is that the models used in both experiments are open source models varying between 350M-13B parameters which are low in size in comparison to larger proprietary models. We understand that the performance on these larger proprietary models perform much better in multiple choice items, however, we were restricted to open source low-moderate sized models due to budgetary and computational constraints.

References

Zaki N Abdel-Malek. 1972. *The closed-list classes of colloquial Egyptian Arabic*. Mouton & Co. N.V., Publishers, The Hauge.

AI@Meta. 2024. Llama 3 model card.

Osamh M Aldosaree. 2016. *Language attitudes toward Saudi dialects*. California State University, Long Beach.

Malik H. Altakrori, Nizar Habash, Teresa Lynn, Younes Samih, Abed Alhakim Freihat, Kirill Chirkunov, Muhammed AbuOdeh, Radu Florian, Preslav Nakov, and Alham Fikri Aji. 2026. Dialectalarabicmmlu: Benchmarking dialectal capabilities in arabic and multilingual language models. In *Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026)*, pages 3199–3219, Palma, Mallorca, Spain. European Language Resources Association (ELRA).

Mohamed Anwar, Abdelhakim Freihat, George Ibrahim, Mostafa Awad, Abdelrahman Atef Mohamed Ali Sadallah, Gurpreet Gosal, Gokul Ramakrishnan, Sarath Chandran, Biswajit Mishra, Rituraj Joshi, Ahmed Frikha, Etienne Goffinet, Abhishek Maiti, Ali El Filali, Sarah Al Barri, Samujjwal Ghosh, Rahul Pal, Parvez Mullah, Awantika Shukla, and 41 others. 2025. Jais 2: A family of Arabic-centric open large language models. Technical report, IFM.

El-Said Badawi. 1973. *Mustawayat al-Arabiyah al-muasirah fī Misr : bahth fi alaqat al-lughah bi-al-hadarah*. Dār al-Mārif, Cairo.

Reem Bassiouney. 2017. *Identity and dialect performance: A study of communities and dialects*. Routledge.

Reem Bassiouney. 2018. Constructing the stereotype: Indexes and performance of a stigmatised local dialect in Egypt. *Multilingua*, 37(3):225–253.

Peter Behnstedt and Manfred Woidich. 1985. Die ägyptisch-Arabischen dialekte. *Tübinger Atlas des Vorderen Orients/Beihefte/B*, 50.

Peter Behnstedt and Manfred Woidich. 2013. Arabic Dialectology. In *The Oxford Handbook of Arabic Linguistics*. Oxford University Press.

El Moatez Billah Nagoudi, Muhammad Abdul-Mageed, AbdelRahim Elmadany, Alcides Inciarte, and Md Tawkat Islam Khondaker. 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, Singapore. Association for Computational Linguistics.

Houda Bouamor, Nizar Habash, Mohammad Salameh, Wajdi Zaghouani, Owen Rambow, Dana Abdulrahim, Ossama Obeid, Salam Khalifa, Fadhl Eryani, Alexander Erdmann, and 1 others. 2018. The MADAR Arabic dialect corpus and lexicon. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Minh Duc Bui, Carolin Holtermann, Valentin Hofmann, Anne Lauscher, and Katharina von der Wense. 2025. Large language models discriminate against speakers of German dialects. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 8223–8251, Suzhou, China. Association for Computational Linguistics.

Ibrahim Eltouhamy. 2015. Language attitudes towards dialects of arabic in egypt. Master's thesis, American Univeristy in Cairo.

Eve Fleisig, Genevieve Smith, Madeline Bossi, Ishita Rustagi, Xavier Yin, and Dan Klein. 2024. Linguistic bias in ChatGPT: Language models reinforce dialect discrimination. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13541–13564, Miami, Florida, USA. Association for Computational Linguistics.

Howard Giles. 1970. Evaluative reactions to accents. *Educational review*, 22(3):211–227.

Atiqa Hachimi. 2015. "good arabic, bad arabic" mapping language ideologies in the arabic-speaking world. *Zeitschrift für arabische Linguistik*, (61):35–70.

Niloofar Haeri. 1997. *The Sociolinguistic Market of Cairo: Gender, Class, and Education*, 1st edition. Routledge, London.

Muhammad Furquan Hassan, Faiza Khan Khattak, and Laleh Seyyed-Kalantari. 2025. Dialectic preference bias in large language models. *Proceedings of the AAAI Symposium Series*, 5(1):365–369.

Abdelghany A Khalafallah. 1969. *A descriptive grammar of Saidi Egyptian colloquial Arabic*, volume 32. Walter de Gruyter GmbH & Co KG.

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.

William Labov. 2006. *The social stratification of English in New York city*. Cambridge University Press.

Wallace E Lambert, Richard C Hodgson, Robert C Gardner, and Samuel Fillenbaum. 1960. Evaluational reactions to spoken languages. *The journal of abnormal and social psychology*, 60(1):44.

Cara Su-Yi Leong and Tal Linzen. 2023. Language models can learn exceptions to syntactic rules. *Proceedings of SCiL*.

Fangru Lin, Shaoguang Mao, Emanuele La Malfa, Valentin Hofmann, Adrian de Wynter, Xun Wang, Si-Qing Chen, Michael J. Wooldridge, Janet B. Pierrehumbert, and Furu Wei. 2025. Assessing dialect fairness and robustness of large language models in reasoning tasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6317–6342, Vienna, Austria. Association for Computational Linguistics.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Mai Mohamed Eida and Nizar Habash. 2025. Beyond Cairo: Sa'idi Egyptian Arabic literary corpus construction and analysis. In *Proceedings of the 5th International Conference on Natural Language Processing for Digital Humanities*, pages 292–304, Albuquerque, USA. Association for Computational Linguistics.

Mai Mohamed Eida, Mayar Nassar, and Jonathan Dunn. 2024. How well do tweets represent sub-dialects of Egyptian Arabic? In *Proceedings of the Eleventh Workshop on NLP for Similar Languages, Varieties, and Dialects (VarDial 2024)*, pages 41–55, Mexico City, Mexico. Association for Computational Linguistics.

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.

Eileen Pan, Anna Seo Gyeong Choi, Maartje Ter Hoeve, Skyler Seto, and Allison Koenecke. 2025. Analyzing dialectical biases in LLMs for knowledge and reasoning benchmarks. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 20882–20893, Suzhou, China. Association for Computational Linguistics.

Dennis R Preston. 1993. Folk dialectology. *American dialect research*, pages 333–378.

Jayanta Sadhu, Maneesha Rani Saha, and Rifat Shahriyar. 2025. Social bias in large language models for Bangla: An empirical study on gender and religious bias. In *Proceedings of the First Workshop on Language Models for Low-Resource Languages*, pages 204–218, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Neha Sengupta, Sunil Kumar Sahu, Bokang Jia, Satheesh Katipomu, Haonan Li, Fajri Koto, Osama Mohammed Afzal, Samta Kamboj, Onkar Pandit, Rahul Pal, Lalit Pradhan, Zain Muhammad Mujahid, Massa Baali, Alham Fikri Aji, Zhengzhong Liu, Andy Hock, Andrew Feldman, Jonathan Lee, Andrew Jackson, and 3 others. 2023. Jais and jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models. *Preprint*, arXiv:2308.16149.

Guokan Shang, Hadi Abdine, Ahmad Chamma, Amr Mohamed, Mohamed Anwar, Abdelaziz Bounhar, Omar El Herraoui, Preslav Nakov, Michalis Vazirgiannis, and Eric P. Xing. 2025. Nile-chat: Egyptian language models for Arabic and Latin scripts. In *Proceedings of The Third Arabic Natural Language Processing Conference*, pages 306–322, Suzhou, China. Association for Computational Linguistics.

Kees Versteegh, Mushira Eid, Alaa Elgibali, Manfred Woidich, and Andrzej Zaborski, editors. 2006–2009. *Encyclopedia of Arabic Language and Linguistics*. Brill, Leiden.

A Probability Distributions

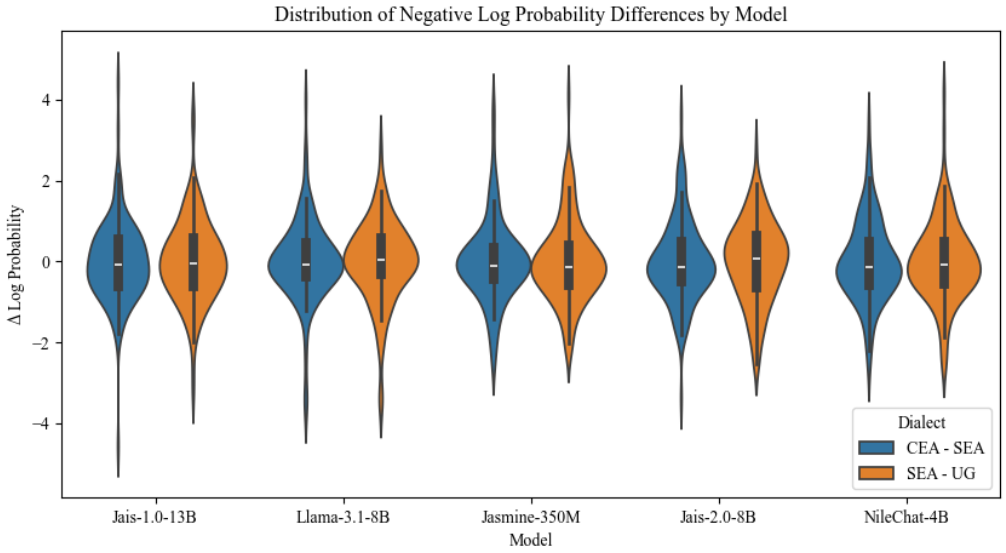


Figure 4: Distribution of differences in negative log probabilities between Cairene Egyptian Arabic (CEA) and Sa'idi Egyptian Arabic (SEA), and between SEA and the ungrammatical baseline (UG) across five language models (Llama-3.1-8B, Jasmine-350M, and Jais-1-13B, Jais-2-8B, and NileChat-4B). Positive values indicate greater preference for the first variety in each comparison (CEA over SEA and SEA over UG), while negative values indicate greater preference for the second.

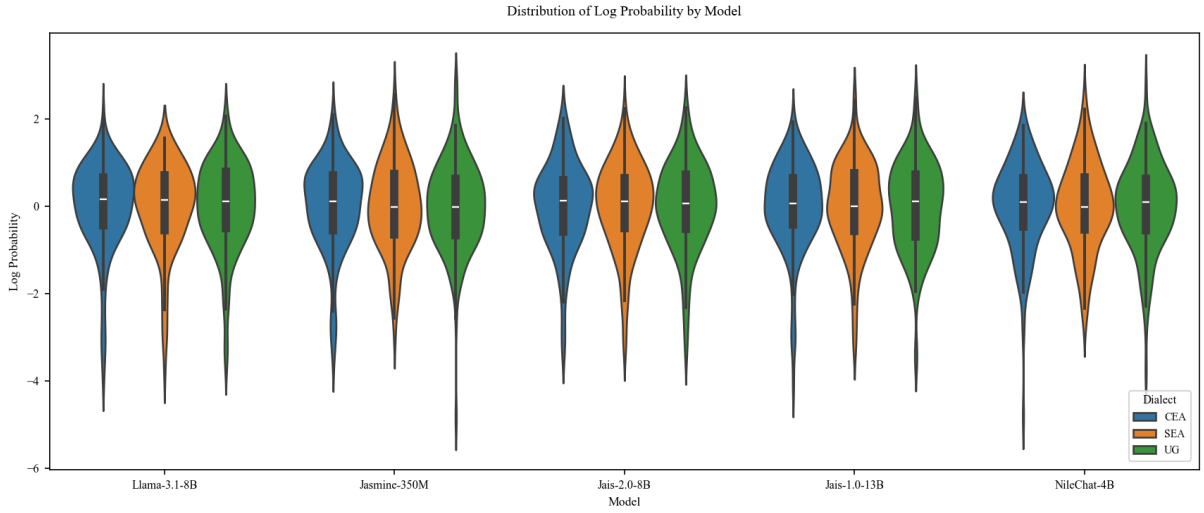


Figure 5: Distribution of negative log probabilities for Cairene Egyptian Arabic (CEA), Sa'idi Egyptian Arabic (SEA), and ungrammatical inputs across five language models (Llama-3.1-8B, Jasmine-350M, and Jais-1-13B, Jais-2-8B, and NileChat-4B). Higher (positive) values indicate greater model preference.

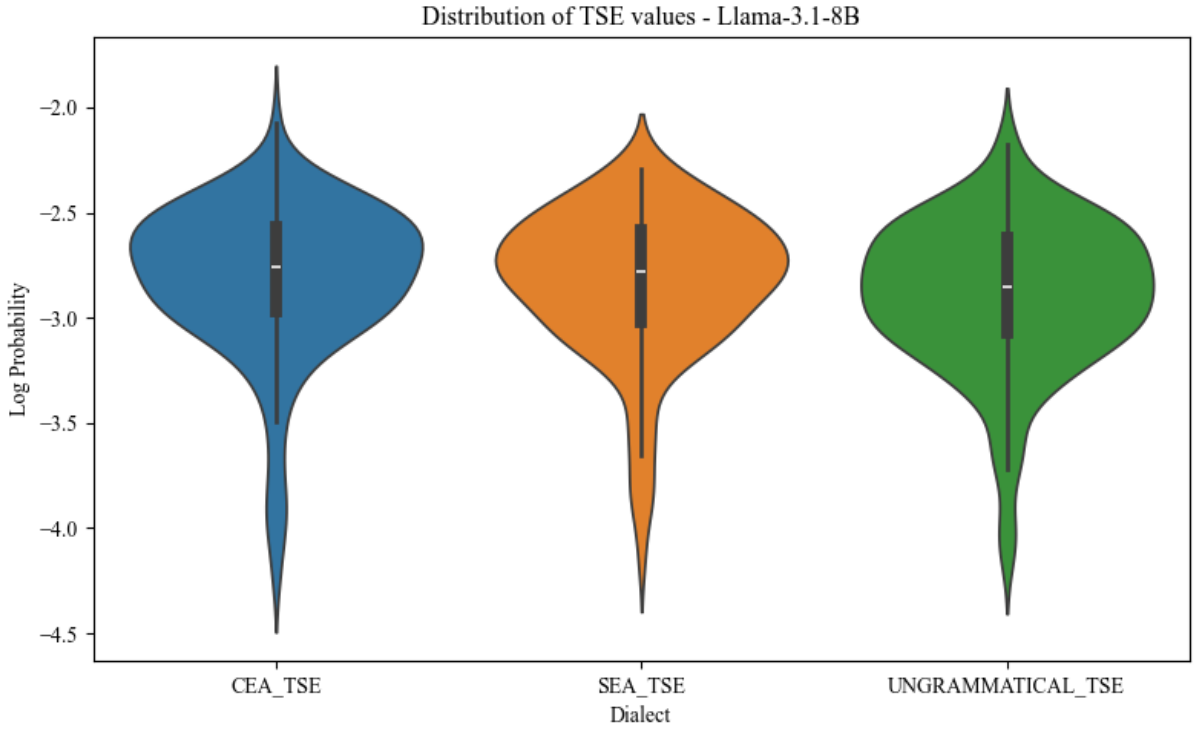


Figure 6: Distribution of token-sequence log probabilities (TSE) for Cairene Egyptian Arabic (CEA), Sa'idi Egyptian Arabic (SEA), and ungrammatical for Llama-3.2-8B.

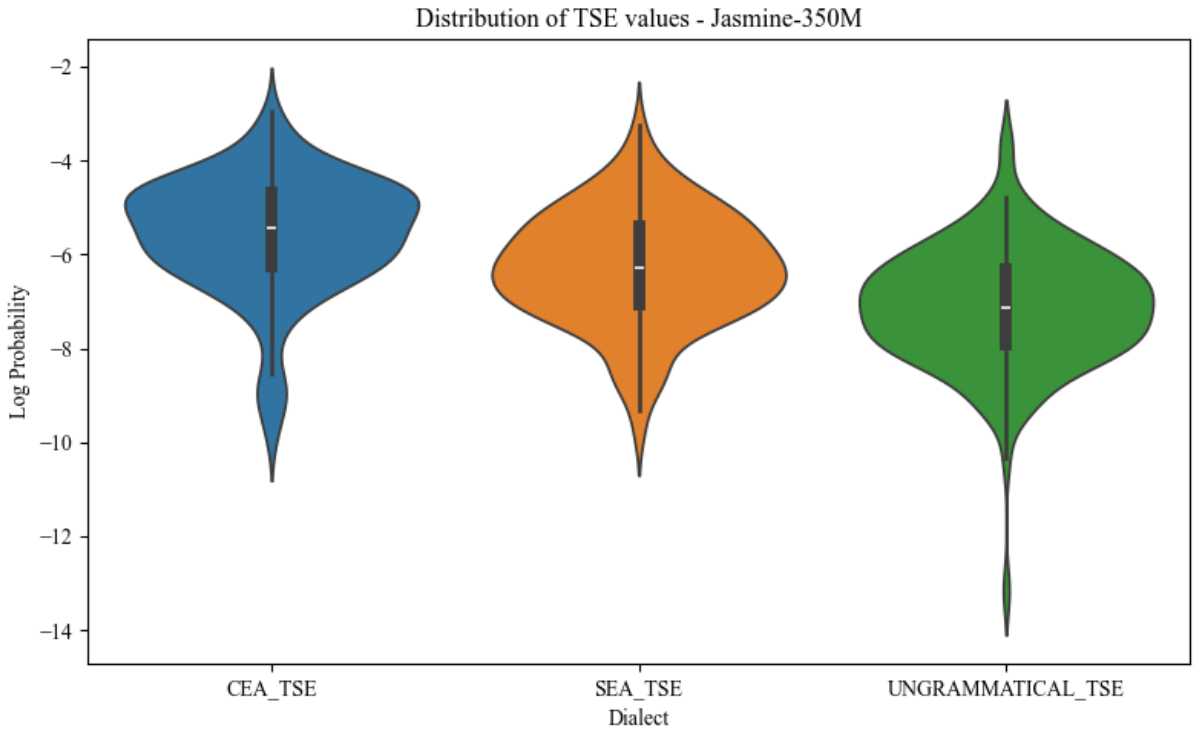


Figure 7: Distribution of token-sequence log probabilities (TSE) for Cairene Egyptian Arabic (CEA), Sa'idi Egyptian Arabic (SEA), and ungrammatical for Jasmine-350M.

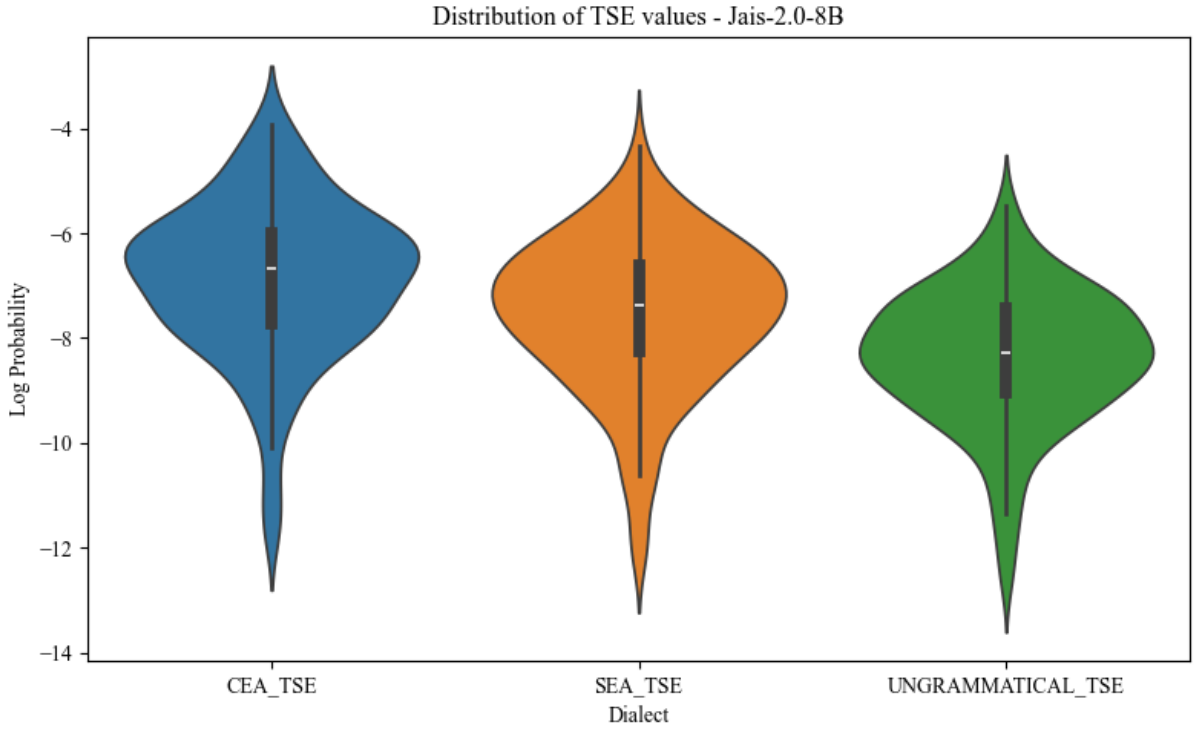


Figure 8: Distribution of token-sequence log probabilities (TSE) for Cairene Egyptian Arabic (CEA), Sa'idi Egyptian Arabic (SEA), and ungrammatical for JAIS-2-8B

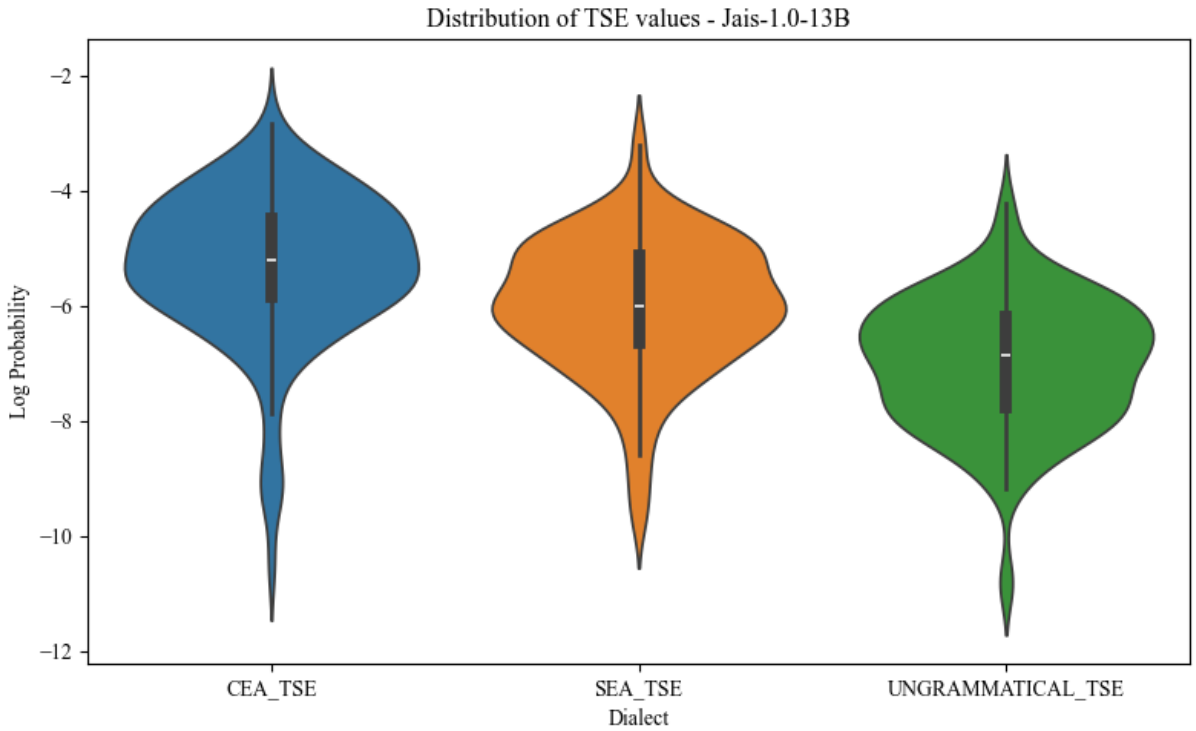


Figure 9: Distribution of token-sequence log probabilities (TSE) for Cairene Egyptian Arabic (CEA), Sa'idi Egyptian Arabic (SEA), and ungrammatical for JAIS-1-13B.

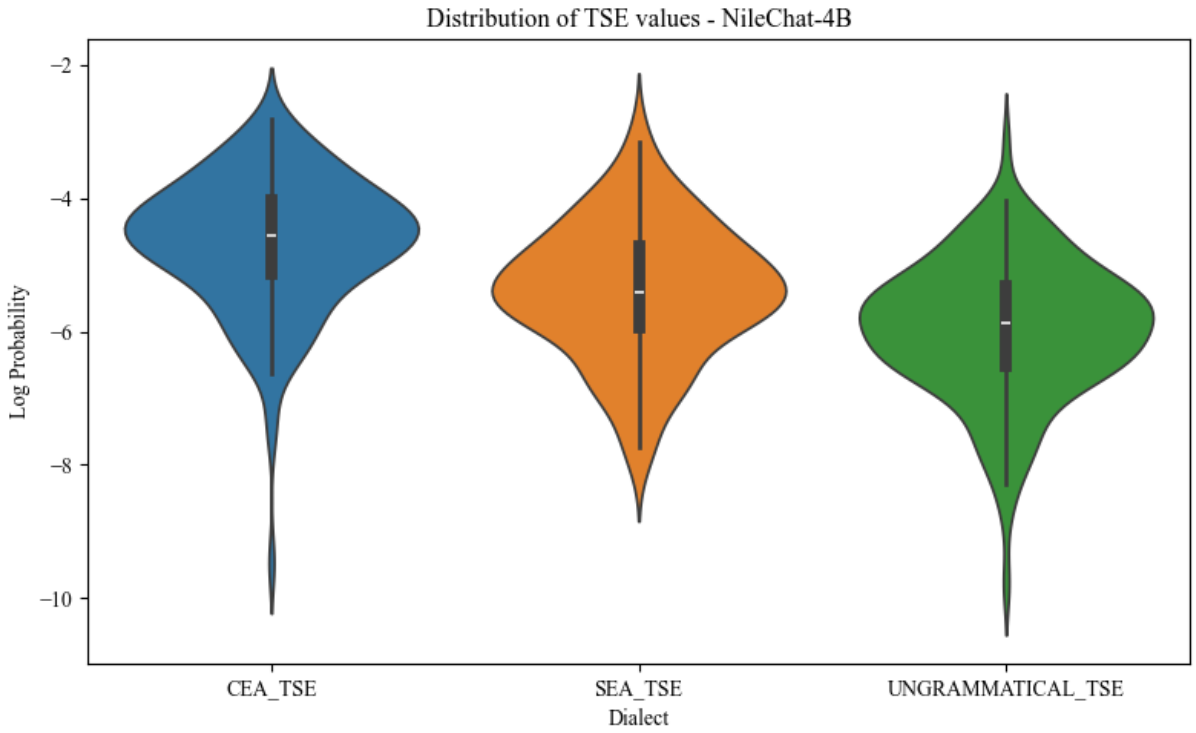


Figure 10: Distribution of token-sequence log probabilities (TSE) for Cairene Egyptian Arabic (CEA), Sa'idi Egyptian Arabic (SEA), and ungrammatical in NileChat-4B.

B Grammatical Features

	# Items	CEA Sample	SEA Sample	Ungrammatical (UG) Sample
Phonetics	41	ليه التأخير ده كله انا حاسس ان الأمور فيها حاجة غلط Why are you so late I feel like something is wrong	ليه التأخير ده كله اني حاسس ان الأمور فيها حاجة غلط Why are you so late I feel like something is wrong	ليه التأخير ده كله لديسلب حاسس ان الأمور فيها حاجة غلط Why are you so late I feel like something is wrong
Morphology	27	في الجوابات الجاية هقولك واحكي لك In my next letters I'll tell you and explain	في الجوابات الجاية عقولك واحكي لك In my next letters I'll tell you and explain	في الجوابات الجاية دقولك واحكي لك In my next letters I'll tell you and explain
Orthography	21	انت مش قلت لى مرة ان امك ميتة Didn't you tell me once your mother had died	انت مش جلت لى مرة ان امك ميتة Didn't you tell me once your mother had died	انت مش سلت لى مرة ان امك ميتة Didn't you tell me once your mother had died
Lexical items	20	وعزيزة امبارح سخنت لكن قايمة الصبح كويسة And Aziza yesterday had a fever but she woke up this morning feeling better	وعزيزة امبارح سخنت لكن قايمة الصبح مليحة And Aziza yesterday had a fever but she woke up this morning feeling better	وعزيزة امبارح سخنت لكن قايمة الصبح ازيلفغ And Aziza yesterday had a fever but she woke up this morning feeling better

Table 3: Samples of CEA–SEA minimal pairs and ungrammatical controls (UG) used to evaluate dialectal preference in LLMs in Experiment 1. This table reports the number of items in each feature category and illustrates the manipulated marked grammatical dialectal features highlighted in red.

Feature	SEA	CEA	MSA	UG	Gloss	ARABIC MMLU MSA	ARADICE MMLU EGY	DIALECTAL ARABIC MMLU
Ad1	دلوق، دلوقت	دلوقتي	الآن	دلوڤ	**Now**	12	11	31
Ad2	واصل	خالص	ابداً	يبلغ	**at all**	5	4	7
Ad3	برا	برة، بره	الخارج	ضصبس	**Outside**	47	20	20
Ad4	برضك، برض	برضه، بردو	أيضاً	وببغلعا	**Also**	5	0	13
Ad5	قوي، جوي	اوي	جداً	ابينتبل	**Very**	25	16	12
Ad6	اهنه، اهنيه	هنا	هنا	بنسليتعبت	**Here**	4	7	26
Ad7	اكده، اكديه	كده، كدا	هكذا	سبسيبيزس	**Like this**	0	48	78
Ad8	لازمن، لازماً	لازم	يجب	ييسظبن	**Have to**	47	393	153
Ad9	بجى	بقى	-	يتزب	**Already**	-	18	27
Dem1	دا	ده	هذا	صيصث	**This**	159	110	381
Introg2	وين	فين	أين	بدشلبس	**Where**	14	17	2
Introg3	ميتى، ميته	امتى	متى	بليظ	**When**	41	33	12
Introg4	كيف	ازاي	كيف	لديظيب	**How**	10	21	2
Prep1	ع	على	على	ز	**On**	2,371	2,120	1,102
Prep2	ف	في	في	ض	**In**	4,593	5,296	1,791
Prep3	كيف	زي	مثل	جش	**Similar to**	89	147	147
Pron1	أني	أنا	أنا	لديسلب	**I**	34	61	20
Pron2	نحنا	احنا	نحن	ضظلذر	**We**	0	13	8
RPron3	ال-	اللي	الذي	جشسي	**which**	1,508	3,934	1,547
Neg1	ما + ش	مش	ليس	يوذس	**Negation**	43	408	583
Neg3	ما + شي	ما + ش	لا ، لن	بيسيظبلدبي	**Negation**	511	276	135
Conjun1	لمن	لما	عندما	ضجطب	**when**	79	277	239
Conjun2	لاجل	عشان/علشان	من أجل	جشصس	**Because**	47	46	98
V-Fut1	ع -	ه -	سوف /س -	د	**Future**	45	124	305
					Total	**9,689**	**13,400**	**6,739**

Table 4: SEA, CEA, MSA and UG variants of selected marked grammatical features across each dialect, followed by frequency of benchmark items containing each feature in the ArabicMMLU, AraDiCE-ArabicMMLU-EGY, and DialectalArabicMMLU benchmarks.

DAASI: Arabic Text Simplification Corpus for Israeli Government Documents

Natalia Vanetik, Maysa Abu Shareb, Huseen Abo Ismael
Department of Software Engineering, Shamoon College of Engineering,
Beer Sheva, Israel
natalyav@sce.ac.il, maysaab@ac.sce.ac.il, huseen1141@outlook.com

Abstract

Arabic-speaking citizens in Israel routinely encounter government website documents written in Modern Standard Arabic (MSA), a variety that differs substantially from the Levantine dialect they use daily. This diglossia gap limits access to public services and creates informational inequity, especially for elderly citizens unfamiliar with modern language tools. We address this problem through three contributions. We target simplification within Modern Standard Arabic (MSA) itself, motivated by the diglossia gap between MSA and the spoken varieties readers use daily; we do not perform translation or adaptation into a spoken dialect. First, we present **DAASI** (**D**iglossia-**A**ware **A**rabic **S**implification for **I**nclusion): a new parallel corpus of 2,350 sentence pairs drawn from Israeli National Insurance Institute documents, constructed via GPT-4o draft simplification followed by expert human validation. Second, we conduct a sociolinguistic survey of 50 Arabic-speaking adults in Israel confirming that government documents are harder to comprehend than school-level texts. Third, we benchmark two Arabic transformer models and GPT-4o across multiple fine-tuning strategies and conduct human evaluations to assess simplified text. Our results confirm that existing Arabic NLP benchmarks do not generalize to the Israeli government domain and that DAASI fills a critical resource gap for inclusive Arabic NLP in a diglossic context.

1 Introduction

Over 473 million people speak Arabic worldwide, yet the Arabic-speaking population in Israel faces a specific and underexplored challenge: formal government communication is conducted in Modern Standard Arabic (MSA), a variety that differs sharply from the Levantine dialects spoken daily by most Arab citizens (Ferguson, 1959; Saiegh-Haddad, 2003). This diglossia gap means that legal forms, healthcare instructions, and administrative notices are, for many readers, closer to a foreign language than to their mother tongue.

Our survey of 50 Arabic-speaking adults in Israel confirms the severity of this gap empirically: participants rated government documents significantly harder to understand than school-level texts across all measured dimensions – vocabulary complexity, sentence clarity, and overall comprehension.

Despite growing interest in Arabic NLP, existing text simplification resources focus on literary and news text in MSA (Nassiri et al., 2023; Alhafni et al., 2024) and do not cover the government domain relevant to Arabic-speaking citizens of Israel, including several health- and disability-related administrative categories (e.g., medical committee decisions, disability benefits) that are part of the government forms we collect, though the corpus is not a clinical or medical-text corpus in the narrower sense. This motivates our central contribution: DAASI (Diglossia-Aware Arabic Simplification for Inclusion), a new parallel corpus of 2,350 sentence pairs built from real Israeli official documents, all sourced from the National Insurance Institute (Bituach Leumi) website (see Section 4).

The stakes of this problem extend well beyond linguistic inconvenience. Arabic is one of Israel's two official languages, and government agencies are legally required to publish documents and websites in Arabic alongside Hebrew (Wikipedia contributors, 2024). Accessibility provisions further mandate that official content be made available in formats suitable for people with disabilities. In practice, however, formal compliance with these requirements does not guarantee communicative accessibility: official Arabic texts are typically written in bureaucratic Modern Standard Arabic (MSA), a register characterised by long nominal constructions, complex subordination, and specialised vocabulary that differs markedly from the Levantine dialect spoken at home by most Arabic-

Proceedings of The Fourth Arabic Natural Language Processing Conference (ArabicNLP 2026), pages 65–76
October 28-29, 2026 ©2026 Association for Computational Linguistics

speaking citizens of Israel, especially the elderly ones (Ayari, 1996; Hudson, 2002). This gap is not unique to Arabic – plain-language movements in many countries have demonstrated that formally translated or legally accurate documents can still be practically inaccessible to their intended audiences (Ondov et al., 2022; Sikka and Mago, 2020). This is particularly evident in a diglossic setting where the written standard and the spoken language are structurally distant varieties (Ferguson, 1959; Saiegh-Haddad and Schiff, 2025). Automatic text simplification has emerged as a practical NLP response to this class of problem, with growing work on clinical and administrative domains in English and European languages (Espinosa-Zaragoza et al., 2023); yet Arabic remains severely under-resourced in this domain.

We make the following contributions:

- we introduce DAASI, a parallel corpus for simplification within Modern Standard Arabic (MSA), targeting government documents directed at Arabic-speaking citizens of Israel; the corpus is motivated by the diglossia gap but does not perform dialect adaptation (see Section 1 for this distinction);
- user survey quantifying perceived comprehension difficulty in formal Arabic texts, motivating the simplification task;
- benchmarks of transformer and generative models, quantifying the domain gap and measuring the gains from fine-tuning with automatic and human evaluation.
- we release `Tabsit.ai`, an open web platform for automatic Arabic text simplification using a hybrid pipeline, which additionally offers an optional experimental dialect-adaptation mode as a separate feature (Section B).

We emphasize that DAASI addresses *intralingual simplification of MSA* (lexical, syntactic, and structural simplification of formal MSA into more accessible MSA), not translation or adaptation into spoken Levantine Arabic. Diglossia motivates the task – it explains why MSA government text is difficult for readers whose everyday spoken variety differs substantially from MSA – but the corpus itself consists of MSA-to-simplified-MSA sentence pairs. We view dialect-level adaptation as a related but distinct direction, and discuss it as a limitation below.

2 Background and Related Work

Text simplification. Text simplification transforms a complex source text into a simpler form while preserving its meaning (Saggion and Hirst, 2017). It involves lexical simplification (replacing rare words with common synonyms), syntactic simplification (splitting long sentences), and structural reorganization (North et al., 2025; Carroll et al., 1998). Research in English has shown clear comprehension gains for learners, individuals with reading difficulties, and non-native speakers (Carroll et al., 1998; Espinosa-Zaragoza et al., 2023). Recent work has extended simplification to clinical and legal domains, where the gap between expert register and lay comprehension is particularly consequential (Ondov et al., 2022; Agrawal and Carpuat, 2024). Large language models have further broadened the toolkit: few-shot and instruction-tuned systems now achieve competitive simplification quality across multiple languages without domain-specific parallel data (Martin et al., 2022; Feng et al., 2023). Evaluation has likewise matured, moving beyond surface readability metrics toward human-centred assessments of meaning preservation and comprehension gain (Agrawal and Carpuat, 2024; Guidroz et al., 2025).

Arabic diglossia. Ferguson's (1959) foundational concept of diglossia describes communities in which a high (formal) variety and a low (spoken) variety coexist with distinct social functions. In the Arabic case, MSA serves formal writing while regional dialects serve everyday speech. In Israel, the dominant spoken variety is Levantine Arabic, and research has confirmed that even highly educated readers show reduced comprehension of MSA government texts compared to dialect-proximate texts (Saiegh-Haddad, 2003; Abu-Rabia, 2000). The phonological, morphological, and lexical distances between MSA and spoken Arabic are substantial enough that children effectively encounter a second language when they first enter formal literacy instruction (Saiegh-Haddad and Schiff, 2025; Abu-Rabia et al., 2022). This distance is not uniform across dialects: Levantine Arabic has been identified as structurally closer to MSA than many other spoken varieties, yet comprehension deficits in formal written registers persist even among proficient adult readers (Saiegh-Haddad and Schiff, 2025). The implications extend well beyond the classroom: bureaucratic and legal documents written in MSA nom-

inal constructions and complex subordination remain opaque to a significant share of the adult Arabic-speaking population, motivating register-aware NLP interventions (Hudson, 2002; Saiegh-Haddad, 2003).

Arabic text simplification. Nassiri et al. (2023) developed the first lexical simplification model for MSA based on word frequency analysis. Al-Thanyyan and Azmi (2023) proposed a hybrid machine-translation and transformer-based approach. The SAMER corpus (Alhafni et al., 2024) contains 28,507 complex-simple sentence pairs from news and educational texts and is currently the largest public Arabic simplification dataset, but it does not cover the government or medical domain. The BAREC shared task (Elmadani et al., 2025) has further advanced readability assessment for Arabic, providing fine-grained difficulty annotations across 19 levels that can inform simplification targets. BAREC is a readability-assessment resource (a classifier assigning a difficulty level to a given text) rather than a text-to-text simplification system, and is not directly comparable to the generative simplification models we benchmark in Section 5; we nonetheless see BAREC-style difficulty scoring as a promising direction for automatically screening or targeting simplification output in future work. Parallel efforts on Arabic readability, including lexicon-based tools (Jiang et al., 2020) and eye-tracking corpora (Baazeem et al., 2025), have supplied the field with resources for modeling difficulty, yet domain coverage remains skewed toward literary and journalistic text. No existing Arabic simplification resource addresses the governmental register.

Arabic NLP models. AraT5 (Nagoudi et al., 2021) is a sequence-to-sequence transformer pretrained on 70 GB of Arabic data and is the strongest open-source Arabic generative model for paraphrase and simplification. AraBERT (Antoun et al., 2020) is a discriminative encoder used for classification and lexical selection. mT5 is a massively multilingual text-to-text transformer that supports Arabic (Xue et al., 2021). GPT-4o serves as a strong few-shot commercial baseline. These models differ substantially in their exposure to dialectal Arabic: AraT5 and AraBERT are trained predominantly on MSA corpora, while mT5's pre-training includes some dialectal web text but without explicit dialect supervision (Xue et al., 2021). MARBERT and CAMeLBERT models (Abdul-Mageed et al., 2021; Inoue et al., 2021) represent more recent attempts to cover dialectal variation, though neither has been evaluated on simplification tasks.

3 Quantifying the Comprehension Gap

We recruited 50 adult Arabic-speaking participants residing in Israel through a purposive sample stratified across three age groups and genders (see Table 1), three education levels (primary, high school, and higher), and diverse occupational backgrounds. All participants volunteered for the task, and their identifying information was protected by providing them anonymous IDs. All participants are native speakers of Levantine Arabic and are proficient in MSA. As Table 1 shows, our convenience sample is heavily skewed toward the 18–25 age group (36 of 50 participants); older adults, who arguably face the largest practical barriers to accessing government services, are underrepresented (only 6 participants aged 41+). We flag this imbalance explicitly as a limitation on the generalizability of our comprehension-gap findings (see Limitations) and identify a larger, age-stratified follow-up study as important future work.

Category	Count	Category	Count
Age 15–18	4	Female	34
Age 18–25	36	Male	16
Age 26–40	4		
Age 41+	6		

Table 1: Participant demographics.

We selected 100 texts, 50 of which are authentic government document passages and 50 of which are school-level descriptive passages. The school-level texts were authentic educational materials collected from the Ministry of Education website, corresponding approximately to middle school level. They were selected as a lower-complexity comparison group representing texts accessible to school-age readers, rather than as simplification targets. Each participant read one text from each group in a random order. After each text, they completed a Likert questionnaire (Nemoto and Beglar, 2014) on four parameters: (1) overall difficulty, (2) vocabulary complexity, (3) sentence clarity, and (4) overall comprehension. In this type of questionnaire, response options were presented as ordered text labels (e.g., "very easy" to "very difficult") rather than bare numbers; the number of response

options varied slightly by item (3–5 options depending on the question), and we code each item on its own ordinal scale from 1 (easiest / clearest / most comprehensible) to N (hardest / least clear / least comprehensible), where N is that item's number of response options. Direction is applied consistently across all four parameters, so that for every dimension a *higher* score indicates a *harder-to-process, less clear, or less comprehensible* text, never the reverse. Additionally, we included two further items: a closed checkbox question listing candidate reasons for comprehension difficulties (participants selected all that applied), and a separate free-text field for any additional comments. We refer to these consistently as the *closed-item* responses and *open-text* responses, respectively, in what follows.

Table 2 summarizes the mean Likert scores for all survey participants. We see that the government documents consistently received higher difficulty ratings across all four metrics. We tested each of the four government-vs-school comparisons using a Wilcoxon signed-rank test (participants rated one text from each group, so ratings are paired); all four differences are statistically significant at $p < 0.01$ (three of the four at $p < 10^{-6}$) (full scores appear in the Appendix). Figure 1 shows distribution of answers to the reasons for comprehension difficulties question.

Dimension	Gov. doc.	School text
Overall difficulty	3.64 ± 1.01	2.20 ± 1.07
Vocabulary complexity	2.86 ± 0.73	1.66 ± 0.80
Sentence clarity	2.88 ± 0.90	2.38 ± 1.05
Overall comprehension	2.22 ± 0.58	1.38 ± 0.64

Table 2: Mean $\pm$ SD perceived difficulty of survey documents (higher = more difficult / less clear / less comprehensible for all four rows; each row uses that item's own response-option range, from 1 to N, where N is 5 for Overall difficulty, 4 for Vocabulary complexity, 4–5 for Sentence clarity, and 3 for Overall comprehension).

We adopt mean($\pm$SD) as our Likert-reporting convention for this survey. Table 9's (Section 5.5) separate human evaluation survey reports only aggregated %$\geq$4 proportions, since its underlying raw per-participant ratings were not retained; we note this difference explicitly where Table 9 is presented.

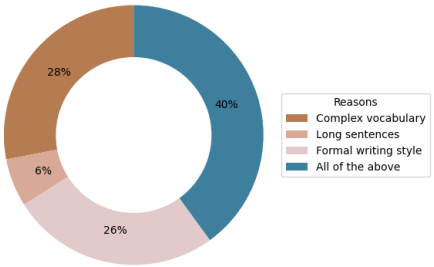


Figure 1: Distribution of factors leading to comprehension difficulty (according to survey participants).

Qualitative analysis of the open-text responses identified three recurring difficulty sources: (1) bureaucratic MSA vocabulary absent from everyday speech (e.g., استيفاء المتطلبات vs. the colloquial تحقيق الشروط), (2) long nominal chains typical of formal Arabic legal style, and (3) passive constructions with dropped agents that obscure responsibility. These findings directly motivate our text simplification strategy in Section B.

4 The DAASI Corpus

In this section, we describe the DAASI parallel simplification corpus which is the first such resource for Israeli government Arabic texts.

Data collection. We collected 50 official documents containing Arabic-language content directed at Arabic-speaking citizens of Israel from the official government website. We used the Arabic section of the National Insurance Institute (Bituach Leumi) as our primary source: https://www.btl.gov.il/Arabic%20HomePage/about_ar/forma_and_authorization_ar/Forms_ar/Pages/default.aspx.

Document categories
General; Unemployment; Domestic Violence; Maternity; Insurance Contributions; Work Injury Benefit; Accident Benefit; Income Support; Senior Citizen (Old Age); Children; Child Support (Alimony); Reserve Military Service; Rehabilitation; Burial Grant; Prisoners of Zion; Survivors (Widow/Widower and Orphans); Work Accident Victims; Hostile Action Victims; Sickness Benefit; Bankruptcy; Work Grant (Negative Income Tax / Employment Grant); Employees; Mobility Benefits; General Disability; Research Funding Assistance; Savings Program for Every Child; Special Services; Counseling Service for Senior Citizens and Their Families; Funds and Trusts; Medical Committees; Children with Cerebral Palsy; Volunteers; International Conventions; Disabled Child

Table 3: Document categories at the National Insurance Institute website.

This site contains more than 30 categories of forms and public-service documents (see Table 3). We selected documents from these categories and extracted the Arabic content available on the official website. The collected documents include forms, eligibility information, public service instructions, and citizen guidance materials intended for Arabic-speaking users. Documents were manually segmented into individual sentences, yielding 2,702 source sentences. Segmentation was performed by the two human annotators described below, following a written guideline instructing them to (i) split at sentence-final punctuation (periods, question marks) and at semicolons introducing an independent clause, (ii) keep list items introduced by a shared stem together with that stem as a single unit rather than splitting mid-list, and (iii) exclude non-sentential content such as form field labels, headers, and tables of numeric values, which were removed prior to segmentation.

Simplification and annotation. Corpus construction was performed in four stages:

1. **Automatic draft.** First, we automatically simplified each sentence using GPT-4o using a few-shot prompt instructing the model to (a) replace formal MSA vocabulary with everyday Arabic equivalents, (b) split sentences longer than 20 words, and (c) retain the original meaning (see Section 8).

2. **Human validation.** Two native Arabic-speaking annotators – both proficient in MSA and Levantine Arabic – reviewed every simplified sentence for accuracy, naturalness, and meaning preservation. Disagreements were resolved by a third independent annotator.

3. **Alignment and filtering.** Sentence pairs where the simplification was judged to change the meaning or omit key information were discarded, yielding 2,350 final pairs.

To make the extent of human correction to the GPT-4o draft transparent: of the 2,702 candidate simplifications, 352 (2,702 − 2,350) were discarded during alignment and filtering because the simplification was judged to change the meaning or omit key information (Section 4); the remaining 2,350 pairs were retained after human review, with corrections ranging from none (accepted as-is) to substantive revision. We did not retain a fine-grained accept/revise/discard breakdown at the level of individual edits, which we note as a limitation; future iterations of this corpus will log this breakdown explicitly. Disagreements between the two primary annotators were referred to a third independent annotator, whose decision was final.

Corpus analysis. To characterize the corpus, we computed descriptive statistics for both the original and simplified sentences. Average character and word length was calculated directly from the text, while average subword token count was measured using the tokenizer of AraBERT v0.2 (Antoun et al., 2020). We also report total word, unique word, and unique lemma counts computed using the Arabic pipeline of the Stanza NLP toolkit (Qi et al., 2020). The results are shown in Table 4.

Measure	Original	Simplified
avg char. length	41.14	41.34
avg word count	10.95	10.70
avg subword count	15.68	15.09
total unique words	3245	2890
total unique lemmas	1918	1622

Table 4: DAASI corpus statistics.

We evaluate lexical diversity in original and simplified texts of the DAASI corpus using four standard metrics: Type–Token Ratio (TTR) (Carter and McCarthy, 2014), which measures the proportion of unique word types; HD-D, a robust estimate of lexical diversity based on probabilistic sampling; MTLD (Measure of Textual Lexical Diversity) (McCarthy and Jarvis, 2010), which captures vocabulary variation independent of text length; and Yule's K, a measure of lexical repetition where lower values indicate higher diversity (Williams, 1946). The results in Table 5 show that the simplified corpus exhibits a slightly lower TTR alongside a lower (i.e., more diverse, since lower Yule's K indicates less repetition) Yule's K, and marginal increases in HD-D and MTLD. Taken together we interpret this as lexical diversity being essentially preserved after simplification: the length-independent metrics (HD-D, MTLD, Yule's K) all move in the direction of equal-or-greater diversity, while the small TTR decrease is consistent with TTR's known sensitivity to text length rather than an actual drop in diversity. Readability, measured using an Arabic-adapted Flesch-like metric (where higher scores indicate easier text), improves in the simplified texts, which obtain higher scores and are

Lexical Diversity & Readability			Lexical and Syntactic Complexity		
Metric	**Original**	**Simplified**	**Metric**	**Original**	**Simplified**
TTR	0.0881	0.0815	Noun Ratio	0.4254	0.3769
HD-D	0.6785	0.6830	Verb Ratio	0.0511	0.0795
MTLD	11.7198	11.9494	Adjective Ratio	0.0945	0.0746
Yule's K	373.5122	371.0642	Dependency Depth	5.1203	4.9305
Flesch Score	-209.8095	-190.3803	Content Word Ratio	0.5710	0.5310

Table 5: Lexical and syntactic complexity and readability of the DAASI corpus.

therefore easier to process. This metric adapts the classic Flesch Reading Ease formula to Arabic by substituting characters per word for syllables per word, following (El-Haj and Rayson, 2016).

To assess lexical and syntactic complexity, we compute the proportions of nouns, verbs, and adjectives, the content word ratio, and dependency depth using the Stanza NLP toolkit (Qi et al., 2020). As shown in Table 5, simplified texts exhibit reduced linguistic complexity, with lower noun and adjective ratios, higher verb ratio, lower content word ratio, and reduced dependency depth, indicating simpler and less syntactically embedded structures.

To evaluate semantic consistency within and across text versions, we compute embedding-based cosine similarity scores for original and simplified texts as well as between them. The results in Table 6 show high semantic consistency between the two versions in the DAASI dataset, with inter-group similarity comparable to intra-group scores, indicating that the simplification process largely preserves meaning. The slightly higher intra-group similarity in the simplified corpus suggests reduced linguistic variation after simplification.

Metric	**Avg cosine similarity**
original intra-group	0.4884
simplified intra-group	0.5068
original vs simplified	0.4917

Table 6: Semantic similarity analysis of the DAASI dataset.

5 Experimental Evaluation

Figure 2 illustrates the evaluation setup used in this study.

We split the DAASI dataset with 2,350 sentences as 80/10/10 train/validation/test split and fine-tune transformer models on the DAASI train set, optionally enhanced with the SAMER dataset (Alhafni et al., 2024). The SAMER dataset (Alhafni et al., 2024) is a large-scale Arabic text simplification corpus containing pairs of original and simplified texts, which we use both as an external source of simplification knowledge and to examine the impact of domain differences between SAMER and DAASI on model performance and generalization. The SAMER dataset is used in full or in a form of a random sample of the exact same size as the DAASI train set to ensure balanced representation of both types of data.

For the non-trainable LLM with prompting (GPT-4o), we use zero-shot and few-shot prompts. The zero-shot prompt is nearly identical to the base prompt (see Section 8) except for examples; for the few-shot setup, it is enhanced by examples from the DAASI training set.

5.1 Models

We evaluate four models:

1. AraT5 (Nagoudi et al., 2021) (`UBC-NLP/AraT5-base` on HuggingFace). A sequence-to-sequence transformer specifically pre-trained for Arabic text generation and understanding tasks, following the T5 text-to-text paradigm.

2. mT5 (Xue et al., 2021) (`google/mt5-base` on HuggingFace). A massively multilingual text-to-text transformer pre-trained on 101 languages and designed for a wide range of multilingual generation and understanding tasks.

3. GPT-4o (Hurst et al., 2024) in a zero-shot setting, denoted by GPT ZS. Prompted directly to simplify text without task-specific examples, relying solely on the task instructions.

4. GPT-4o (Hurst et al., 2024) in a few-shot setting, denoted by GPT FS. Prompted with input-output simplification examples from the DAASI dataset.

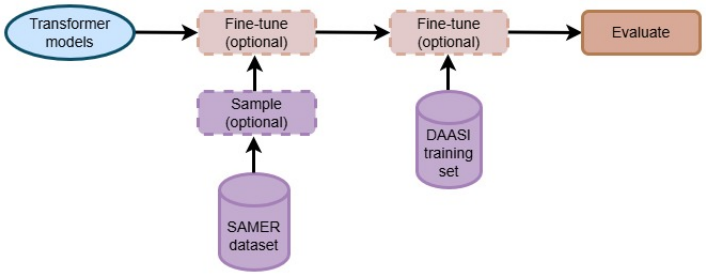


Figure 2: Evaluation pipeline.

Transformer models (AraT5 and mT5) are fine-tuned on training data for 10 epochs, batch size 16, with learning rate 5×10^{-5}.

5.2 Metrics

We report *BLEU* (Papineni et al., 2002), which measures n-gram overlap with human references, and *SARI* (Xu et al., 2016), which rewards appropriate additions, deletions, and paraphrases and is the standard evaluation metric for text simplification. We additionally report *BERTScore* (Zhang et al., 2020) (F1), computed with AraBERT v0.2 embeddings (Antoun et al., 2020) (9 layers, no baseline rescaling), as a meaning-preservation metric that is not tied to surface n-gram overlap and therefore complements BLEU/SARI, particularly for systems such as GPT-4o that rewrite aggressively. We also report *%unchanged*, the ratio of output sentences identical to the source, as a measure of simplification aggressiveness.

5.3 Automatic Evaluation Results

Table 7 presents the performance of AraT5 and mT5 models under different fine-tuning strategies, evaluated on the DAASI test set.

The results show that fine-tuning on the simplification-oriented DAASI dataset consistently yields better performance than training on the SAMER datasets alone. For both transformer models, the best results are achieved by first fine-tuning on the SAMER and DAASI datasets, with AraT5 obtaining the highest scores while also producing the lowest percentage of unchanged outputs (15.74%). In contrast, models trained only on SAMER exhibit substantially higher unchanged rates, indicating a tendency to preserve the original text rather than perform effective simplification, which highlights the domain gap between the two datasets and the importance of task-specific adaptation through subsequent fine-tuning on DAASI.

The 0% unchanged rate for GPT-4o means it rewrites every sentence, which reduces BLEU against a conservative gold reference.

5.4 Linguistic Analysis

We report the performance of the best-performing system (AraT5 fine-tuned on the combined SAMER and DAASI datasets), across lexical, readability, syntactic, and content-related metrics in Table 8. Overall, the results indicate that the model consistently preserves lexical diversity (i.e., diversity is not reduced relative to the gold standard) while producing texts that are easier to read and syntactically simpler than the original inputs. These trends align with effective simplification, showing reduced structural complexity and controlled changes in lexical usage without loss of semantic content.

Metric	Original	Gold Standard	Predicted
TTR	0.2079	0.1944	0.1918
HD-D	0.7363	0.7379	0.7251
MTLD	17.9541	18.1443	17.0296
Yule's K	277.5530	278.2512	305.2959
Flesch-like	-203.7940	-183.2127	-194.9636
noun ratio	0.4160	0.3707	0.3837
verb ratio	0.0520	0.0840	0.0709
adjective ratio	0.0983	0.0708	0.0900
dep. depth	5.2447	5.0106	4.9894

Table 8: Lexical diversity and complexity metric comparison on the DAASI test set.

5.5 Human Evaluation

The same 50 survey participants from Section 3 evaluated matched original and automatically simplified sentence pairs from the DAASI test set (we used the output of our best model). Participants rated each sentence on three 5-point Likert scales: comprehension, fluency, and vocabulary simplicity. They also indicated their preference between the original and simplified versions.

Model	FT on	BLEU	SARI	BERTScore	% unchanged
AraT5	DAASI	41.83	59.42	81.19	17.02%
AraT5	SAMER (sample)	30.20	45.03	76.45	56.81%
AraT5	SAMER	29.93	46.79	77.39	66.81%
AraT5	SAMER+DAASI	44.73	63.20	83.76	15.74%
AraT5	SAMER (sample)+DAASI	43.71	61.50	83.05	16.81%
mT5	DAASI	37.29	53.70	77.16	24.04%
mT5	SAMER (sample)	30.65	44.04	76.43	56.81%
mT5	SAMER	33.43	49.38	77.26	85.32%
mT5	SAMER+DAASI	39.97	56.55	79.97	31.49%
mT5	SAMER (sample)+DAASI	38.15	53.78	79.36	29.57%
GPT-4o ZS	–	25.64	52.25	78.26	0.00%
GPT-4o FS	–	29.00	54.70	80.00	0.02%

Table 7: Evaluation results (best scores are marked in gray).

As can be seen from Table 9, the automatically simplified sentences received substantially higher ratings across all three dimensions. We indicate there %≥4 – percentage of ratings in the top two points on the 5-point scale. As a result, 84% of ratings for simplified sentences fell in the 4-5 range, compared to 35% for originals on the comprehension dimension. The strongest effect was for vocabulary simplicity: ratings in the 4-5 range rose from 31% (original) to 88% (simplified), directly confirming the advantages of our best model.

Metric	Original (%≥4)	Simplified (%≥4)
Comprehension	35	84
Fluency	42	81
Vocabulary simplicity	31	88
Preference: simplified	84%	

Table 9: Human evaluation results (1–5 scale; higher = better for all three rows here, the opposite convention from Table 2, since this table measures positive quality rather than difficulty – noted explicitly to avoid the ambiguity raised in review)

.

These are the exact pooled %≥4 values reported in our final human-evaluation summary. We report only this statistic for Table 9 rather than a mean(+SD), as we did for Table 2: the underlying raw per-participant ratings for this particular evaluation were not retained, only the aggregated %≥4 figures, which we note as a limitation on the granularity of this specific analysis.

6 Discussion and Conclusion

Our results show that the main challenge in Arabic government text simplification is the domain mismatch between existing resources and the target register. Models trained only on SAMER perform poorly on DAASI, with AraT5 achieving BLEU 29.93, SARI 46.79, and a high unchanged rate (66.81%), indicating limited simplification behavior. In contrast, fine-tuning on DAASI improves performance substantially (BLEU 44.73, SARI 63.20, 15.74% unchanged), demonstrating the importance of domain-specific data. The gap reflects the formal MSA register, bureaucratic vocabulary, and institution-specific terminology present in government documents.

GPT-4o provides competitive zero-shot and few-shot simplification performance (SARI 52.25 and 54.70, respectively), but the best AraT5 configuration outperforms it while offering a locally deployable and auditable alternative. Human evaluation further confirms that DAASI-based simplification improves comprehension and vocabulary accessibility.

In this work, we introduced DAASI, a government-domain Arabic simplification corpus motivated by the comprehension gap between formal MSA documents and the needs of Arabic-speaking citizens in Israel. Our experiments demonstrate that targeted training data is critical for effective simplification, and we release DAASI and the Tabsit.ai platform to support future research on inclusive Arabic NLP.

7 Limitations

DAASI covers only 50 documents from Israeli government authority, which is modest compared to SAMER's 28,507 pairs, and is therefore limited in domain breadth. The current size reflects the cost of expert human validation, which we treat as necessary for a resource intended for public-facing accessibility use; scaling further will require either additional annotator time or a lighter-weight quality-filtering model trained on the existing validated pairs to pre-screen future GPT-4o drafts before human review, which we leave to future work. Human validation was performed by two annotators; Inter-annotator agreement was not recorded during corpus construction, so we cannot retrospectively report an agreement statistic for the Accept/Revise/Reject decisions. The platform currently serves Levantine Arabic speakers in Israel and has not been evaluated for other dialects. Our comprehension survey (Section 3) also relies on a convenience sample skewed toward younger adults (Table 1); a larger, age- and education-stratified sample would strengthen the generalizability of the reported comprehension gap.

The Tabsit.ai system was evaluated only with Levantine speakers; generalization to Egyptian, Gulf, or Moroccan dialect speakers has not been tested.

Finally, only aggregated summary statistics were retained for the original-vs-simplified human evaluation (Table 9); the raw per-participant ratings were not preserved, so we cannot report means, standard deviations, or a paired significance test for that particular survey, unlike the government-vs-school comprehension survey (Table 2) for which raw ratings were available.

8 Data and Model Availability

The data is publicly available at https://github.com/NataliaVanetik/DAASI-simplification-corpus, and the trained model and application are available at https://github.com/Huseen1141/tabsit. This GitHub project also contains full annotation guidelines and the base simplification prompt used in this work.

9 Ethical Considerations

Survey participants provided written informed consent and were made aware of the research purpose. No personally identifiable information was collected or retained. Participants volunteered for the task. Corpus documents were collected from publicly available Israeli government websites and used solely for academic research. Annotation was performed by native Arabic speakers who volunteered for the task. The Tabsit.ai application is designed to assist Arabic-speaking citizens in accessing public information; it does not alter legal documents or produce authoritative translations.

We use "Arabic-speaking citizens of Israel" to describe the population whose comprehension of government communication we study; this is a citizenship-based description and is not intended to imply a single linguistic, religious, or political identity – it may include, for example, Mizrahi and Sephardi Jewish citizens who also speak Arabic dialects, alongside Palestinian citizens of Israel. We use "Levantine Arabic" to refer to the spoken variety our participants speak, while noting that this label spans a broader region than Israel alone and that "Palestinian Arabic" is the more precise term for many of our participants specifically. We avoid the term "Arab Israeli" in the body of this paper, since it is rejected by many Palestinian citizens of Israel as eliding Palestinian national identity; we retain the term only where it appears verbatim in cited source material. We acknowledge that these three dimensions – citizenship, linguistic variety, and identity – are analytically distinct and should not be treated as interchangeable.

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.

Salim Abu-Rabia. 2000. Effects of exposure to literary arabic on reading comprehension in a diglossic situation. *Reading and Writing*, 13:147–157.

Salim Abu-Rabia, Eman Sarsour Kedan, Haneen Wattad, S Abu-Rabia, ES Kedan, and H Wattad. 2022. The influence of diglossia on syntactic proficiency in modern standard arabic among regular and struggling readers. *Creative Education*, 13(1):252–282.

Sweta Agrawal and Marine Carpuat. 2024. Do text simplification systems preserve meaning? a human evaluation via reading comprehension. *Transactions of the Association for Computational Linguistics*, 12:432–448.

Suha S. Al-Thanyyan and Aqil M. Azmi. 2023. Simplification of arabic text: A hybrid approach integrating machine translation and transformer-based lexical model. *Journal of King Saud University – Computer and Information Sciences*, 35(8):101662.

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.

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.

Salah Ayari. 1996. Diglossia and illiteracy in the arab world. *Language, culture and curriculum*, 9(3):243–253.

Ibtehal Baazeem, Hend Al-Khalifa, and Abdulmalik Al-Salman. 2025. Araeyebility: Eye-tracking data for arabic text readability. *Computation*, 13(5):108.

John Carroll, Guido Minnen, Yvonne Canning, Siobhan Devlin, and John Tait. 1998. Practical simplification of english newspaper text to assist aphasic readers. In *Proceedings of the AAAI-98 workshop on integrating artificial intelligence and assistive technology*, pages 7–10. Madison, WI.

Ronald Carter and Michael McCarthy. 2014. *Vocabulary and language teaching*. Routledge.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1:4171–4186.

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 2016)*, pages 5234–5241. European Language Resources Association.

Khalid N Elmadani, Bashar Alhafni, Hanada Taha, and Nizar Habash. 2025. Barec shared task 2025 on arabic readability assessment. In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 239–252.

Isabel Espinosa-Zaragoza, José Abreu-Salas, Elena Lloret, Paloma Moreda Pozo, and Manuel Palomar. 2023. A review of research-based automatic text simplification tools. In *Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing*, pages 321–330.

Yutao Feng, Jipeng Qiang, Yun Li, Yunhao Yuan, and Yi Zhu. 2023. Sentence simplification via large language models. *arXiv preprint arXiv:2302.11957*.

Charles A. Ferguson. 1959. Diglossia. *Word*, 15(2):325–340.

Theo Guidroz, Diego Ardila, Jimmy Li, Adam Mansour, Paul Jhun, Nina Gonzalez, Xiang Ji, Mike Sanchez, Sujay Kakarmath, Mathias MJ Bellaiche, and 1 others. 2025. Llm-based text simplification and its effect on user comprehension and cognitive load. *arXiv preprint arXiv:2505.01980*.

Alan Hudson. 2002. Outline of a theory of diglossia. *International journal of the sociology of language*, 2002(157).

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*.

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.

Zhengyang Jiang, Nizar Habash, and Muhamed Al-Khalil. 2020. An online readability leveled arabic thesaurus. In *Proceedings of the 28th International Conference on Computational Linguistics: System Demonstrations*, pages 59–63.

Louis Martin, Angela Fan, Éric Villemonte De La Clergerie, Antoine Bordes, and Benoît Sagot. 2022. MUSS: Multilingual unsupervised sentence simplification by mining paraphrases. In *Proceedings of the thirteenth language resources and evaluation conference*, pages 1651–1664.

Philip M McCarthy and Scott Jarvis. 2010. Mtld, vocd-d, and hd-d: A validation study of sophisticated approaches to lexical diversity assessment. *Behavior research methods*, 42(2):381–392.

El Moatez Billah Nagoudi, AbdelRahim Elmadany, and Muhammad Abdul-Mageed. 2021. Arat5: Text-to-text transformers for arabic language generation. *arXiv preprint arXiv:2109.12068*.

Naoual Nassiri, Violetta Cavalli-Sforza, and Abdelhak Lakhouaja. 2023. Approaches, methods, and resources for assessing the readability of arabic texts. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22(4):1–30.

Tomoko Nemoto and David Beglar. 2014. Likert-scale questionnaires. In *JALT 2013 conference proceedings*, volume 108 (1), pages 1–6.

Kai North, Tharindu Ranasinghe, Matthew Shardlow, and Marcos Zampieri. 2025. Deep learning approaches to lexical simplification: A survey. *Journal of Intelligent Information Systems*, 63(1):111–134.

Brian Ondov, Kush Attal, and Dina Demner-Fushman. 2022. A survey of automated methods for biomedical text simplification. *Journal of the American Medical Informatics Association*, 29(11):1976–1988.

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.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th annual meeting of the association for computational linguistics: system demonstrations*, pages 101–108.

Horacio Saggion and Graeme Hirst. 2017. *Automatic text simplification*, volume 32. Springer.

Elinor Saiegh-Haddad. 2003. Linguistic distance and initial reading acquisition: The case of Arabic diglossia. *Applied Psycholinguistics*, 24(3):431–451.

Elinor Saiegh-Haddad and Rachel Schiff. 2025. Diglossic and orthographic features of reading comprehension in standard arabic: The primacy of the spoken language. *Reading Research Quarterly*, 60(1):e598.

Pooja Sikka and Vijay Mago. 2020. A survey on text simplification. *arXiv preprint arXiv:2008.08612*.

Wikipedia contributors. 2024. Arabic language in israel. Wikipedia.

Carrington Bonsor Williams. 1946. Yules characteristic and the index of diversity. *Nature*, 157:482–482.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mt5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pages 483–498.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Appendix

A Likert Score Standard Deviations and Significance Tests

Item coding. Each survey item's response options were coded on its own ordinal scale from 1 (easiest/clearest/most comprehensible) to N (hardest/least clear/least comprehensible), where N is that item's number of distinct response options: Overall difficulty ($N = 5$), Vocabulary complexity ($N = 4$), Sentence clarity ($N = 4$ for the government-text item, $N = 5$ for the school-text item, which offered an additional "very clear" option), and Overall comprehension ($N = 3$).

Table 2 means with standard deviations:

Dimension	Gov. (mean±SD)	School (mean±SD)
Overall difficulty	3.64 ± 1.01	2.20 ± 1.07
Vocabulary complexity	2.86 ± 0.73	1.66 ± 0.80
Sentence clarity	2.88 ± 0.90	2.38 ± 1.05
Overall comprehension	2.22 ± 0.58	1.38 ± 0.64

Wilcoxon signed-rank tests (government vs. school, paired by participant, $n = 50$):

Dimension	Statistic	p-value
Overall difficulty	$W = 53.5$	$p = 4.5 \times 10^{-7}$
Vocabulary complexity	$W = 28.5$	$p = 1.3 \times 10^{-7}$
Sentence clarity	$W = 159.5$	$p = 9.1 \times 10^{-3}$
Overall comprehension	$W = 12.5$	$p = 3.3 \times 10^{-7}$

All four comparisons are statistically significant ($p < 0.01$); three of the four are significant well beyond conventional thresholds ($p < 10^{-6}$). The sentence-clarity comparison, while still significant, shows a smaller effect than the other three dimensions.

Table 9's (Section 5.5) original-vs-simplified human evaluation is a separate survey instrument from the government-vs-school comprehension survey analyzed above. Only the aggregated %≥4 values for that evaluation were retained; the raw per-participant ratings needed to compute means, standard deviations, and a paired significance test for that survey do not exist, which we note as a limitation.

B Tabsit.ai application

Tabsit.ai is a full-stack web application for Arabic text simplification. It follows a three-tier architecture consisting of a React/Node.js frontend, a Python FastAPI backend, and a simplification engine combining AraT5 (Nagoudi et al., 2021) and GPT-4o. Users can paste or upload Arabic text and receive side-by-side original and simplified versions. Additional features include an educational mode that highlights replaced terms with explanations and a dialect mode that adapts output toward Levantine Arabic. The simplification engine employs a two-stage hybrid pipeline. First, syntactic simplification uses Stanza's Arabic parser to identify sentence boundaries and part-of-speech tags, splitting overly long sentences at coordinating conjunctions (و, ف, لكن) and embedded clauses. Second, lexical simplification replaces difficult vocabulary using a lexicon of over 2,000 formal-colloquial word pairs derived from survey responses and administrative glossaries, with substitutions validated through mBERT pseudo-log-likelihood scoring (Devlin et al., 2019). For passages classified as highly complex by an AraT5-based difficulty classifier, GPT-4o few-shot generation serves as a fallback to simplify constructions not covered by the rule-based pipeline.

C Segmentation and Annotation Guidelines

Sentence segmentation. Documents were segmented by the two annotators using the following rules: (i) split at sentence-final punctuation (periods and question marks) and at semicolons introducing an independent clause; (ii) retain list items introduced by a shared stem with that stem as a single unit when the items are not independently comprehensible; (iii) exclude non-sentential content, such as form labels, headers, and tables of numeric values; and (iv) do not treat abbreviations or numbered references (e.g., article or paragraph numbers) as sentence boundaries.

Accept / revise / reject guideline for GPT-4o drafts. For each candidate simplification from the automatic-draft stage (see the prompt document), annotators classified it as **Accept** if it preserved all factual content (names, dates, numbers, and deadlines), was grammatically correct and natural, and was easier for a non-specialist reader to understand. A draft was marked **Revise** if it preserved the meaning and was directionally simpler but required only a minor correction, such as fixing an unnatural word choice, restoring an omitted number, or replacing an overly literal bureaucratic expression. A draft was **Reject** if it changed the meaning, omitted required factual content, or failed to simplify the sentence; such pairs were either regenerated using a revised prompt or discarded. Disagreements were resolved by a third independent annotator, whose decision was final.

Beyond Gender, Number, and Rationality: Modeling Extended Arabic Morpho-Syntactic Agreement with Aggregation

Bailasan Zaina[1,2] **Salam Khalifa**[1,3] **Nizar Habash**[1]
[1]Computational Approaches to Modeling Language Lab, New York University Abu Dhabi
[2]Georgetown University [3]Stony Brook University
btz3@georgetown.edu, {salam.khalifa,nizar.habash}@nyu.edu

Abstract

Arabic gender–number morpho-syntactic agreement exhibits systematic variation beyond functional gender, number, and rationality (GNR) (Alkuhlani and Habash, 2011). We identify and classify agreement patterns involving plural-to-unit and unit-to-members shifts, and introduce *aggregation* (AGG), a lexeme-level feature and corpus-derived lexical-semantic resource that captures these extended agreement affordances. We extend the GNR agreement framework with aggregation and evaluate it on over one million agreement pairs from ATB1/2/3 and BAREC. AGG improves agreement prediction over a strong baseline by 1.4% absolute on average (38% error reduction). Remaining errors largely reflect analyzer, parser, and treebank limitations. Interpretable agreement assessment of this kind can support the development of pedagogical applications and benchmarking the grammaticality of generative model outputs.[1]

1 Introduction

Arabic is a morphologically rich language whose words enter into agreement and assignment relations over several features, including gender, number, case, and definiteness (Habash, 2010). This paper focuses on gender and number agreement.

Basic Arabic agreement largely follows feature *matching*: the target (e.g. an adjective or verb) matches the noun that controls it in gender and number, with the well-known exception that irrational (non-human) plurals default to feminine-singular agreement. As Alkuhlani and Habash (2011) showed, what matters is the controller's *functional* (grammatical) gender and number rather than its surface form, since broken plurals and similar forms diverge from the features they control. Together, functional gender, number, and rationality (GNR) predict the default agreement target in most cases. Some controllers, however, license agreement options that fall outside this default. Table 1 previews the problem with five definite, masculine controllers: most of their agreement options follow from GNR (✓), but a few are extended options (+) that GNR cannot predict, such as a plural construed as a single unit or a unit accessed through its members. These extended options are the focus of this paper, and we model them with a new lexeme-level feature, *aggregation* (AGG).

	C	D	A	A	G	**Aggregation**	
	I	I	I	R	R	**Rationality**	
	S	S	P	P	P	**Number**	
	M	M	M	M	M	**Gender**	
	الشجر	الفريق	الكتب	الوزراء	العرب	← ***controller***	
	trees	team	books	ministers	Arabs	↓ ***target***	
said	✓	✓				MS	__ قال
			✓		+	FS	__ قالت
		+		✓	✓	MP	__ قالوا
important	✓	✓				MS	__ مهم
			✓		+	FS	__ مهمة
		+		✓	✓	MP	__ مهمون

Table 1: Agreement options for five controllers and two targets. A single ✓ marks default gender-number-rationality (GNR) agreement; + marks an extended agreement option not predicted by default GNR. Abbreviations: M masculine, F feminine, S singular, P plural, R rational, I irrational; A atomic, C collective, G groupable, D divisible.

Modeling interpretable agreement is useful for pedagogical applications (Saddiki et al., 2018) and for benchmarking the grammatical output of generative models (Alrajhi et al., 2022; Zbib et al., 2026).

Our contributions are threefold. **First**, we introduce aggregation (AGG) as a new lexical agreement feature for Arabic computational linguistics, together with a corpus-derived AGG lexi-

[1]Data and Code are publicly available: https://github.com/CAMeL-Lab/Arabic_GNRA_Lexicon

Proceedings of The Fourth Arabic Natural Language Processing Conference (ArabicNLP 2026), pages 77–87
October 28-29, 2026 ©2026 Association for Computational Linguistics

con for noun entries; and we implement an extended agreement rule system that captures both default and extended agreement options. **Second**, we broaden the empirical scope of Arabic agreement validation by expanding the construction inventory beyond the three phenomena targeted in prior work (verb-subject agreement, noun-adjective agreement, and numeral-noun agreement) to include subject-verb agreement, demonstratives, relative pronouns, subject-predicate constructions, *ḥāl* (circumstantials), ordinals, and fine-grained numeral–noun subtypes. We also refine the treatment of numeral–noun agreement by distinguishing fine-grained numeral subtypes. **Finally**, we conduct a large-scale evaluation of the above mentioned constructions over more than one million agreement pairs across numerous genres.

2 Linguistic Background

2.1 Agreement, Controllers, and Targets

Arabic distinguishes feature *assignment* (e.g. a verb assigning case to its subject) from feature *agreement* (a target reflecting a controller's features); we model agreement only. The basic relevant controller features are *gender* (masculine/feminine), *number* (singular/dual/plural), and *rationality* (rational/human vs. irrational/nonhuman). Each noun carries *form* features, which describe the surface morphology of the observed wordform, and *functional* features, which represent the agreement-relevant gender and number values used by agreement targets. The two diverge for broken plurals and similar forms, so agreement is best graded against functional features.

2.2 Simple Match and GNR Agreement

We organize agreement prediction into three families, realized as five configurations (§5). **Simple match** copies the controller's gender and number to the target, using either form or functional features. **GNR** agreement applies construction-specific rules over gender, number, and rationality, again over form or functional features; rationality is what derives the deflected feminine-singular agreement of irrational plurals (e.g. كلاب *kilāb* 'dogs' controls feminine singular), and functional-feature GNR is the strongest configuration in prior work. The third family, which we call **GNRA**, extends GNR with aggregation; we introduce it in §4 after motivating the gap it fills.

2.3 Agreement Constructions

Alkuhlani and Habash (2011) modeled three constructions (noun–adjective, verb–subject, and numeral–noun). We replicate their constructions and extend with six additional constructions that span adjectival, verbal, pronominal, and numeral targets and differ in which features they copy. All nine constructions are presented in Table 2

2.4 The problem of Extended Agreement

The agreement families above describe both default agreement and the additional (+) options shown in Table 1. What remains missing is a way to characterize which lexemes license these additional options and what semantic construals they make available. In such cases, a rational plural may be construed as one non-individuated unit, while a unit-like group noun may be accessed through its individual members. Traditional Arabic grammar already recognizes related noun classes, especially اسم الجمع *ism al-jam‘* 'collective/group noun' and اسم الجنس الجمعي *ism al-jins al-jam‘ī* 'collective generic noun'; the latter is often discussed together with اسم الوحدة *ism al-waḥda* 'singulative', which names a single unit/member of the class denoted by a collective generic noun. These traditional categories show that Arabic grammar distinguishes group, collective, and individual-unit readings, but they do not by themselves predict the agreement alternations targeted here. The additional options are not predictable from gender, number, and rationality alone; they depend on the lexical-semantic properties of the noun, which we formalize as *aggregation* in §4.

3 Related Work

3.1 Arabic Agreement Modeling

Arabic's templatic and concatenative morphology motivates a distinction between form-based and functional morphology (Habash, 2010; Alkuhlani and Habash, 2011). Alkuhlani and Habash (2011) enriched the Penn Arabic Treebank (Part 3) with functional gender, number, and rationality and modeled three constructions, showing that agreement cannot be reduced to surface feature copying; Marton et al. (2011) use such functional features to improve dependency parsing. Alkuhlani and Habash (2011)'s line of work is the closest to our work: we replicate its form/function and GNR rule sets, but extend it along three axes (more constructions,

Construction	Example	Controller	Target	Agreement Type
Noun–Adj	البنت^{A} الجميلة^{B} *al-bint*^{A} *al-jamīla*^{B} ‘the pretty^{B} girl^{A}’	**A**: Noun	**B**: Adj (Mod)	gender, number
Verb–Subj	قرأ^{A} الأولاد^{B} *qara’a*^{A} *al-awlād*^{B} ‘the boys^{B} read^{A}’	**B**: Noun (Subj)	**A**: Verb	gender; number=sg
Subject–Verb	الأولاد^{A} قرأوا^{B} *al-awlād*^{A} *qara’ū*^{B} ‘the boys^{A} read^{B}’	**A**: Noun (Subj)	**B**: Verb	gender, number
Subject–Pred	البيت^{A} كبير^{B} *al-bayt*^{A} *kabīr*^{B} ‘the house^{A} is big^{B}’	**A**: Noun (Subj)	**B**: Pred	gender, number
Numeral–Noun	خمسة^{A} أعضاء^{B} *khamsat*^{A} *a‘ḍā’*^{B} ‘five^{A} members^{B}’	**B**: Noun (counted)	**A**: Numeral	gender polarity
Demonstrative	هذا^{A} الكتاب^{B} *hādhā*^{A} *al-kitāb*^{B} ‘this^{A} book^{B}’	**B**: Noun	**A**: Dem	gender, number
Relative	الرجل^{A} الذي^{B} *al-rajul*^{A} *alladhī*^{B} ‘the man^{A} who^{B}’	**A**: Noun	**B**: Rel. pron	gender, number
Ḥāl	جاء^{A} الرجل^{B} ضاحكا^{C} *jā’a*^{A} *al-rajul*^{B} *ḍāḥikan*^{C} ‘the man^{B} came^{A} laughing^{C}’	**B**: Noun (Subj)	**C**: Ḥāl	gender, number
Ordinal	الفصل^{A} الثاني^{B} *al-faṣl*^{A} *al-thānī*^{B} ‘the second^{B} chapter^{A}’	**A**: Noun	**B**: Ordinal	gender, number

Table 2: The nine agreement constructions evaluated. Each example tags its words with superscripts (^{A}, ^{B}, ^{C}) that link to the Controller and Target columns. The subject–predicate family also covers كان *kāna*- and إنّ *’inna*-type predications.

	KH’2011	This paper
Inflectional Features	Form Gen Num Func Gen Num	From Gen Num Func Gen Num
Lexical Features	Rationality -	Rationality Aggregation
Constructions	Three	Nine
Datasets	PATB Part 3 -	PATB Parts 1/2/3 BAREC
Agreement Pairs	39,561	1,021,686

Table 3: Comparing our work with Alkuhlani and Habash (2011) (**KH’2011**) across different dimensions.

larger and cross-genre corpora, and aggregation) and address the extended-agreement cases (the + cells of Table 1) that it does not model. See Table 3 for full comparison.

3.2 Semantic Grounding of Aggregation

AGG is not a new semantic theory but a computationally usable operationalization of distinctions long discussed for nominal denotation: atomicity, collectivity, individuation, and group- versus member-level access (Chierchia, 1998; Pearson, 2011; D’Anna, 2020). Pearson (2011) shows that group nouns display both unit-level and member-level behavior and that “committee” nouns differ from “collection” nouns, so collectivity alone does not license member access. For Arabic, D’Anna (2020) treats agreement with collectives as reflecting holistic, distributive, or non-individuated construals of the controller. We are inspired by this literature in defining aggregation (§4). However, our contribution is a computational, lexeme-level agreement feature rather than a semantic theory.

4 Aggregation

GNR specifies the *default* agreement pattern of a lexeme, but, as §2.4 showed, some nouns license additional options. We capture this with **Aggregation (AGG)**, a lexeme-level feature on nouns that is independent of gender, number, and rationality and characterizes how a lexeme’s denotation is internally organized. GNR gives the default agreement; AGG specifies the affordances for *extended* agreement. AGG has four values for nouns, schematized in Figure 1. The figure separates nouns that remain

AGG	Default Construal	Extra Construal	Extra
A	individuated entity	–	–
C	non-individuated aggregate	–	–
G	rational plural individuals	collective / non-individuated unit	FS
D	unit-like group	rational/human members	MP

Table 4: Directionality of the two reactive AGG values. Groupable construes a rational plural as one unit (extra FS); Divisible accesses a unit's members (extra MP). Atomic and Collective license no extra agreement.

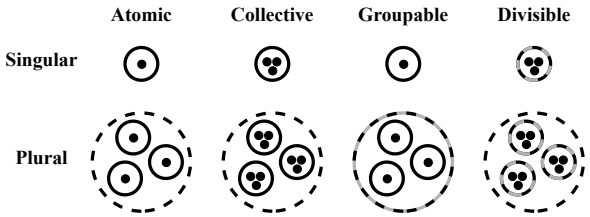


Figure 1: The four AGG values as denotational schemata, crossed with singular and plural. A grey ring is a *coercible* boundary licensing extended agreement: Atomic and Collective have none; Groupable adds a grey envelope over a rational plural (→ FS); Divisible opens a unit to its members (→ MP).

within default GNR agreement from nouns that license an additional directional construal. **Atomic** (A) and **Collective** (C) nouns license no agreement beyond GNR: an Atomic noun is an ordinary individuated entity, while a Collective noun is a non-individuated aggregate whose members are not independently accessible for agreement; member reference instead uses a singulative. The two *extended-agreement* values carry a directionality (Table 4). **Groupable** (G) lexemes have a rational-plural default but allow that plural to be construed as a single non-individuated unit, additionally licensing feminine-singular agreement (MP → FS). **Divisible** (D) lexemes have a unit-like default but allow access to their constituent members, additionally licensing masculine plural agreement when those members are rational/human (MS/FS → MP). Because AGG is lexeme-level, a value characterizes the whole lexeme even when the affordance surfaces only on particular number forms. Table 5 summarizes the full AGG inventory, including *Not Applicable* (N) for non-nouns. The agreement-reactive nouns are exactly the Groupable and Divisible cases.

The example in Figure 2 shows a Divisible controller in context: the noun المجموعة *al-majmūʿa* 'the group' takes default feminine-singular agreement on the adjective الأخيرة *al-'axīra* 'the last' (from GNR), but member-access masculine-plural agreement on the verb فازوا *fāzū* 'won', because the group is accessed through its individual members.

		Controller					Target Agreement	
Lemma	Gloss	Base	G	N	R	A	GNR	GNRA
موظف	employee	موظفون	M	P	R	A	MP	–
وزير	minister	وزراء	M	P	R	A	MP	–
كلب	dog	كلاب	M	P	I	A	FS	–
شجر	trees	أشجار	M	P	I	C	FS	–
رجل[1]	man	رجال	M	P	R	G	MP	FS
رجل[2]	leading figure	رجالات	M	P	R	G	MP	FS
عربي	Arab	عرب	M	P	R	G	MP	FS
ناس	people	ناس	F	S	I	D	FS	MP
قوم	nation	قوم	M	S	I	D	MS	MP
فريق	team	فريق	M	S	I	D	MS	MP

Table 5: Aggregation values for example controllers, grouped by aggregation value (A, C, G, D). A and C are *non-AGG-active* (default-only); G and D are *AGG-active*. "GNR" is the default agreement for the cited baseword, and "GNRA" is the extra agreement that Groupable and Divisible additionally license ("–" = none).

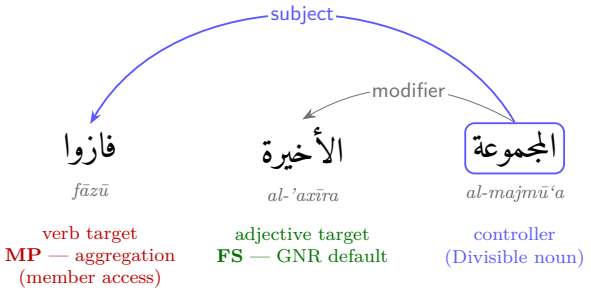


Figure 2: Example with mixed agreement patterns.

5 Methodology

5.1 Rule Configurations

We evaluate five configurations, the first four replicating Alkuhlani and Habash (2011). **Simple match** comprises (1) *Form Match* and (2) *Func Match*, where the target copies the controller's form or functional gender and number. **GNR** comprises (3) *Form Rules* and (4) *Func Rules*, construction-specific agreement rules over gender, number, and rationality applied to form or functional features;

Func Rules is the strongest replicated baseline. The fifth, **GNRA** or (5) *Extended*, adds aggregation. All five are evaluated on ATB1/2/3 and BAREC; the main comparison is Func Rules vs. Extended. The full extended rule inventory is in Table 6.

5.2 The Extended System

The extended rule set preserves the same mapping architecture as the baselines: rules take controller-side features as input and output the agreement features licensed on the target. The baseline rules use the controller's form features $F(N)$ or functional GNR features $G(N)$ to derive the default; the extended system keeps the functional GNR mappings as the default layer and adds further controller-side inputs, including, but not limited to, aggregation, to license additional target mappings (Table 6). Thus AGG is used as an input condition, parallel to gender, number, and rationality, not as an after-the-fact interpretation of errors. AGG is the central new feature of the layer but not its entirety. A controller–target pair is scored as correct when the gender–number features observed on the target are among the agreement options licensed by the corresponding rule configuration for that controller; otherwise it is scored as incorrect. A configuration may license more than one target option, and the observed target need only belong to the licensed set.

Separately, a small class of 24 lemmas (e.g. سوق *sūq* 'market', طريق *ṭarīq* 'road') admits two valid genders. We treat this as a lexical-resource matter: such nouns receive parallel masculine and feminine entries, so either agreement pattern counts as correct. This is not an aggregation feature, but an independent lexical alternation that feeds the GNRA agreement rules.

5.3 The AGG Lexicon

We developed a targeted, corpus-derived lexical-semantic AGG lexicon. Candidate nouns were extracted automatically from the evaluated ATB and BAREC agreement pairs and flagged when their features or observed agreement behavior suggested an alternation not derivable from gender, number, and rationality alone. We manually reviewed the candidates and distinguished AGG-reactive nouns, namely Groupable and Divisible lexemes, from default (non-reactive) nouns, including Atomic and Collective cases as well as other inanimate or abstract nouns whose agreement remains derivable from GNR. Concretely, a lemma is Groupable if its rational plural can control feminine-singular unit agreement, and Divisible if its singular unit form can control masculine-plural member agreement. Annotation is lexeme-level, and cases with unstable judgments were flagged and kept out of the central evidence.

The primary annotation was carried out by one author, a native Arabic-speaking PhD researcher in Arabic linguistics with relevant training and prior annotation experience. Following a review of the literature, we piloted the annotation on approximately 3,000 automatically extracted noun candidates in their agreement contexts; the pilot led us to refine the initial form-based labels into the final lemma-sensitive Atomic/Collective/Groupable/Divisible representation. Candidates were reviewed in context and classified using corpus evidence where available and semantic-class criteria otherwise. Difficult cases and the final category definitions were reviewed collaboratively by all three authors. The process took approximately 60–70 hours over several weeks.

The AGG-reactive portion of the resource contains 864 lemma entries. Because lemma types recur across many surface forms and agreement pairs in the evaluated data, the AGG-reactive subset is small in type coverage but high in token impact, accounting for 20.3% of noun tokens in the agreement pairs. The annotated lemmas fall into recognizable semantic clusters: Groupable lemmas are dominated by human plural classes (ethnonyms such as العرب *al-'arab* 'the Arabs'; religious and social groups such as اليهود *al-yahūd* 'the Jews'; professional and status groups such as العلماء *al-'ulamā'* 'scholars', القضاة *al-quḍāt* 'judges'; and kinship plurals such as الإخوة *al-'ixwa* 'brothers'), while Divisible lemmas are dominated by unit-denoting collectives accessed through human members (teams and groups such as فريق *farīq* 'team'; committees and bodies such as لجنة *lajna* 'committee', مجلس *majlis* 'council'; and organizations such as حزب *ḥizb* 'party', وفد *wafd* 'delegation').

5.4 Corpus Synchronization and Data

We use two corpora for evaluation. **ATB** (ATB1/2/3) is the replication corpus, with $58{,}501$ evaluated controller–target agreement pairs; Alkuhlani and Habash (2011) used only PATB Part 3. The

Controller Condition (Input)							Target Result	
Form Gen $F(N)$	**Form Num** $F(N)$	**Form Rat** $F(N)$	**Func Gen** $G(N)$	**Func Num** $G(N)$	**Func Rat** $G(N)$	**AGG value**	**Adj. Agr**	**Member Access**
*	*	*	M	S	*	*	MS	—
*	*	*	M	D	*	*	MD	—
*	*	*	M	P	R	*	MP	—
*	*	*	M	P	I	*	FS	—
*	*	*	F	S	*	*	FS	—
*	*	*	F	D	*	*	FD	—
*	*	*	F	P	R	*	FP	—
*	*	*	F	P	*	*	FS	—
*	S	*	*	P	R	G	FS	BLOCKED
*	*	*	*	*	I	D	MP	LICENSED

Table 6: **Noun–adjective extended rule inventory.** The left block gives the rule input conditions: form features ($F(N)$), functional features ($G(N)$), and AGG value; the result block gives the licensed adjective agreement. The final column is a semantic readout, not an input condition: feminine singular agreement with a Groupable (G) noun corresponds to blocked member access, and masculine plural agreement with a Divisible (D) noun to licensed member access. "*" marks an unconstrained feature value; "—" marks cases with no AGG-induced reading. Target abbreviations combine M/F (masculine/feminine) with S/D/P (singular/dual/plural); e.g., MD = masculine dual.

ATB data derives from the Penn Arabic Treebank (PATB; Maamouri et al., 2004). Its dependency view follows CATiB (Habash and Roth, 2009), converted from gold treebank annotation, and its functional morphology comes from CALIMAStar and the CAMeL resources (Taji et al., 2018; Obeid et al., 2020; Khairallah et al., 2024).

BAREC is the extension corpus, based on the BAREC-10M resource (Elmadani et al., 2026). From this corpus, our extraction pipeline yields 963,185 evaluated controller–target agreement pairs. Unlike ATB, whose syntactic layer is converted from gold treebank annotation, BAREC is automatically parsed with CAMeL Parser 2.0 (Elshabrawy et al., 2023).

For each corpus, extraction aligns a syntactic layer with a morphological layer: the syntax determines which controller–target pairs exist, and the morphology supplies the form and functional features. For ATB, we synchronize the CATiB view with CALIMA analyses. For BAREC, we apply the same logic to the automatically parsed CATiB view and CAMeL Morph analyses, with an added document-level alignment stage. Synchronizing BAREC's two views aligns 514,677 of 514,879 sentences in 20,535 matched documents (99.96%), and a further 37,745 tokens are dropped for surface-form disagreement. We do not model definiteness, which is out of scope. To assess extraction quality, we manually validated a stratified random sample of 500 retained BAREC controller–target pairs spanning all nine constructions; 4 pairs were incorrectly extracted, an estimated extraction error rate of 0.8%. This estimate concerns extraction among retained pairs and is separate from downstream analyzer and parser noise (see §7).

6 Results

6.1 Metrics

We report accuracy for each rule configuration and error reduction from the strongest functional baseline (Func Rules) to the Extended system. Since the evaluated constructions differ greatly in size, we use a macro-average across construction–corpus rows as the headline summary, giving each row equal weight. The error reduction in the average row is computed from the macro-average accuracies, rather than by averaging the row-level error-reduction values.

6.2 Overall Results

Table 7 reports accuracy for the five configurations per construction and corpus. The contrast of Alkuhlani and Habash (2011) reproduces: simple match is weakest, GNR rules capture most cases, and the Extended system is the maximum in every row. On average, the Extended system improves over the strongest functional baseline by 1.4 points absolute (96.2 → 97.7), corresponding to a 37.6% average reduction in remaining errors; on individual constructions the gain reaches 4.4 accuracy points (BAREC noun–adjective, 92.6 → 97.0). The gains capture a well-defined class of agreement alternations, namely collective-unit and member-access readings, that GNR alone cannot express.

Construction	Corpus	N	Simple Form Match	Simple Func Match	GNR Form Rules	GNR Func Rules	GNRA Extended	Error Redux (%)
Noun–Adjective	ATB	35,153	69.1	70.9	82.3	98.1	**99.1**	51.8
	BAREC	542,561	66.8	66.5	77.1	92.6	**97.0**	59.8
Verb–Subject	ATB	11,350	76.8	75.5	86.9	95.2	**95.5**	4.8
	BAREC	212,667	77.8	75.1	85.2	94.3	**96.4**	36.4
Subject–Verb	ATB	1,520	77.2	82.6	84.5	95.8	**97.0**	27.8
	BAREC	27,948	78.0	82.8	83.4	98.5	**99.8**	88.4
Subject–Predicate	ATB	386	71.8	74.6	84.7	97.2	**97.9**	27.4
	BAREC	6,763	77.2	80.2	81.6	94.1	**95.4**	22.5
Numeral–Noun	ATB	767	16.0	4.0	84.6	96.6	**96.7**	3.8
	BAREC	9,513	24.2	9.3	77.5	92.6	**92.9**	4.3
Demonstrative	ATB	2,830	74.5	75.4	85.8	98.5	**99.8**	86.3
	BAREC	61,729	74.5	74.0	83.0	96.2	**98.3**	54.6
Relative	ATB	4,413	63.5	67.4	77.8	95.8	**96.7**	23.1
	BAREC	67,141	61.1	60.3	75.2	96.2	**97.0**	20.6
Ḥāl	ATB	686	92.4	94.3	92.9	96.4	**99.1**	75.8
	BAREC	8,146	75.8	83.8	76.4	99.6	**99.7**	12.8
Ordinal	ATB	1,396	95.6	97.2	95.6	97.2	**99.9**	94.7
	BAREC	26,717	97.1	97.7	97.1	97.7	**99.7**	88.0
Average			70.5	70.6	84.0	96.2	**97.7**	37.6

Table 7: Per-construction accuracy (%) for the five configurations on ATB and BAREC. **Form/Func Match** = target copies the controller's form/functional gender–number; **Form/GNR Rules** = construction-specific agreement rules over form features, or over functional gender, number, and rationality; **Extended** = our system, conditioning on form, functional/GNR, and aggregation features.

BAREC Text Category	N	GNR	GNRA	Red.
Educational Materials	273,377	14,301	5,705	60.1
Literature	272,625	14,638	5,722	60.9
Media & Culture	235,528	11,271	4,722	58.1
Academic	160,586	15,702	11,060	29.6
Encyclopedic	18,406	1,107	474	57.2
Religion & Philosophy	2,098	130	56	56.9
Art & Music	565	81	54	33.3
Total	**963,185**	**57,230**	**27,793**	**51.4**

Table 8: BAREC Final-stage errors by Text Category, aggregated across all constructions. **GNR** and **GNRA** are the error counts of the GNR Func Rules baseline and the Extended system; **Red. (%)** is error reduction.

6.3 Where the Extension Helps

The Extended system improves most where a few systematic, aggregation-driven triggers dominate, as in noun–adjective, subject–verb, ordinal, and demonstrative; it improves least where the residual errors are *heterogeneous*, spread across many unrelated, lexeme- and analysis-specific causes rather than a few systematic triggers a single rule can capture (§7). Table 8 stratifies BAREC by Text Category; residual errors concentrate in *Academic* text, which is dominated by specialized vocabulary and heavier automatic-analysis noise rather than genuine agreement failures.

6.4 Specificity of the Extended System

A natural concern is whether the Extended system gains accuracy merely by licensing more alternatives. To address this, we compare, for each controller word and feature configuration, the number of distinct target gender–number options the system licenses (*permissiveness*) with the number of distinct options actually attested in the corpus for that configuration (*diversity*). A configuration is *More Specific* when the system licenses fewer options than are attested, *Exact* when the two counts coincide, and *Less Specific* when the system licenses more options than are attested. This measure compares the sizes of the licensed and attested option sets, rather than whether the individual licensed options match the attested ones; prediction accuracy is evaluated separately: it asks how tightly the licensed options track the attested alternatives, weighted by pair occurrences.

Table 9 reports occurrence-weighted results for all nine constructions on both corpora. Two patterns emerge. First, relative to the GNR baseline, the Extended system shifts part of the More Specific mass to Exact: on BAREC noun–adjective, More Specific decreases from 59.7% to 56.0% of pair occurrences while Exact rises from 40.3% to 42.4%, and the system becomes Less Specific

		GNR			Extended		
Construction	**Corpus**	**More**	**Ex.**	**Less**	**More**	**Ex.**	**Less**
Noun–Adj.	ATB	7.9	92.1	0.0	6.3	67.7	25.9
	BAREC	59.7	40.3	0.0	56.0	42.4	1.6
Verb–Subj.	ATB	9.2	90.8	0.0	6.4	78.7	15.0
	BAREC	46.9	53.1	0.0	40.6	55.8	3.6
Subj.–Verb	ATB	2.7	97.3	0.0	0.9	78.4	20.7
	BAREC	2.6	97.4	0.0	1.8	88.8	9.3
Dem.	ATB	0.8	99.2	0.0	0.8	84.4	14.8
	BAREC	20.2	79.8	0.0	18.2	75.8	6.0
Rel.	ATB	2.4	97.6	0.0	2.3	97.6	0.1
	BAREC	18.6	81.4	0.0	18.4	81.2	0.4
Subj.–Pred.	ATB	0.0	100	0.0	0.0	84.6	15.4
	BAREC	9.8	90.2	0.0	8.1	84.0	7.9
Num.–Noun	ATB	11.9	88.1	0.0	11.9	88.1	0.0
	BAREC	57.4	42.6	0.0	56.7	43.1	0.2
Ord.	ATB	1.4	98.6	0.0	1.4	95.1	3.4
	BAREC	4.1	95.9	0.0	3.5	93.1	3.4
Ḥāl	ATB	0.1	99.9	0.0	0.1	91.3	8.6
	BAREC	51.4	48.6	0.0	51.4	45.5	3.1

Table 9: Specificity of the GNR baseline and the Extended system, as percentages of controller–target pair occurrences (More/Exact/Less Specific – See §6.4).

in only 1.6% of pair occurrences. The BAREC verb–subject and numeral constructions show the same profile, whereas in constructions with little attested variation (preverbal subjects, demonstratives, predicates) the additional licensed options largely go unattested, so Exact decreases in favor of small Less Specific shares ($\leq 9.3\%$ on BAREC). Second, the two corpora distort specificity in opposite directions. ATB is roughly fifteen times smaller than BAREC, so most controllers occur with only one attested option even where the alternation is grammatical; this both suppresses ATB's More Specific shares and counts every licensed-but-unattested option against it, yielding the systematically larger Less Specific shares (up to 25.9% for noun–adjective). Conversely, BAREC's scale attests genuine alternation but also accumulates rare analyzer errors, and a single stray target form on a frequent controller counts its entire occurrence mass as More Specific: the ḥāl share, for instance, stems almost entirely from one high-frequency configuration attested with two stray plural tokens, whereas the noun–adjective, numeral, and verb–subject shares are dominated by controllers attested with multiple options at substantial rates.

Lexicon Source	**Test Corpus**	**Accuracy**
ATB	ATB	98.0
BAREC	ATB	98.0
ATB+BAREC	ATB	98.0
Func Rules	ATB	96.8
BAREC	BAREC	97.4
ATB	BAREC	97.3
ATB+BAREC	BAREC	97.4
Func Rules	BAREC	95.8

Table 10: Cross-corpus lexicon evaluation: macro-average accuracy (%) across the nine constructions when the AGG lexicon is restricted to vocabulary from one corpus and evaluated on each test corpus. Func Rules is the lexicon-free functional baseline.

6.5 Numeral Constructions

The numeral–noun rows behave differently from the others: Arabic numerals 3–10 take *gender polarity*, the gender opposite to the counted noun's singular, so the numeral's surface gender is the reverse of what plain matching expects. This is why the Simple Match columns are unusually low for numeral–noun (e.g. Func Match $4.0/9.3$): copying the controller's gender is systematically wrong under polarity, whereas the GNR and AGG configurations encode it.

6.6 Cross-Corpus Lexicon Evaluation

To assess whether the gains from AGG depend on constructing the lexicon from the same corpus used for evaluation, we restrict the AGG lexicon to vocabulary extracted from one corpus and evaluate on the other. Table 10 reports the macro-average accuracy across the nine constructions for each lexicon-source and test-corpus combination.

On BAREC, the ATB-only lexicon achieves 97.3% accuracy, compared with 97.4% for both the BAREC-derived and combined ATB+BAREC lexicons, while the Func Rules baseline reaches 95.8%. Thus, 1.5 of the 1.6 percentage-point improvement over the functional baseline is retained when BAREC does not contribute its vocabulary to lexicon construction. On ATB, all three AGG lexicon conditions yield approximately 98.0%, compared with 96.8% for Func Rules. These results indicate that the gains are not primarily dependent on deriving the AGG vocabulary from the same corpus on which it is evaluated, substantially reducing the circularity concern while not eliminating all possible corpus-derived bias.

7 Discussion and Residual Error Analysis

The Extended system improves accuracy across constructions, and the remaining errors are heterogeneous. They fall into three broad types: a small set of model-relevant hard cases, construction-specific issues, and upstream resource/analyzer errors. The last dominate: most residual disagreements are not agreement-rule failures but artifacts of the input resources, echoing Alkuhlani and Habash (2011), whose own residual analysis attributed roughly half of remaining disagreements to annotation errors and a further fifth to treebank errors.

7.1 Construction-specific Residuals

A transparent residual case is numeral polarity (above). The relevant rule is categorical: numerals 3–10 take the gender opposite to the singular counted noun. Thus, with feminine سنة *sana* 'year' the expected form is ست سنوات *sitt sanawāt* 'six years', not ستة سنوات *sittat sanawāt*. The latter nonetheless occurs in the data and is counted as an error, although the deviation originates with the source text rather than with the model (a gold/source error).

7.2 Upstream Resource and Analyzer Errors

A large share of the residual is upstream noise. The largest source is missing functional gender on named entities and place names, which the analyzer defaults to masculine: the toponym إسرائيل *isrā'īl* 'Israel' (likewise واشنطن *wāshinṭun* 'Washington', إيران *īrān* 'Iran') is tagged masculine singular yet controls feminine-singular targets (أظهرت إسرائيل *'aẓharat isrā'īl* 'Israel showed'), so the system records a mismatch. A second family is feminine forms tagged masculine, including *ta-marbuta* nouns such as جامعة *jāmi'a* 'university', حكومة *ḥukūma* 'government', and منطقة *manṭiqa* 'region'. A third is broken plurals assigned their lemma's singular features, e.g. الأحياء *al-'aḥyā'* 'the neighborhoods', tagged masculine singular like its lemma حي *ḥayy* 'neighborhood' instead of masculine plural. These call for resource-level fixes, and the rule system doubles as a diagnostic that surfaces them; for named entities we report the gaps rather than patching them with ad-hoc genders.

7.3 Out-of-scope Cases

Some constructions are excluded by reproducible filters because they require phrase- or subtree-level computation rather than word-level features. One case is quantificational NPs, where agreement depends on the quantifier and the quantified noun jointly. For example, كل طالب حضر *kull ṭalib ḥaḍara* 'every student attended' involves singular agreement with the quantified noun, whereas كل الطلاب حضروا *kull al-ṭullāb ḥaḍarū* 'all the students attended' involves a plural quantified noun and plural agreement. Another case is conjunction agreement, where the target may agree with one conjunct or with the whole coordinated phrase: in جاء محمد وفاطمة *ja'a Muḥammad wa-Faṭima* 'Muhammad and Fatima came', the verb may show closest-conjunct agreement with محمد *Muḥammad*, or, in SV order, agree with the coordinated phrase as a dual.

8 Conclusion and Future Work

We recast Arabic agreement so that GNR gives the default and aggregation, a lexeme-level noun feature, licenses extended agreement, with Groupable and Divisible modeling the opposite directions of plural-to-unit and unit-to-members. We contribute the AGG feature, an aggregation-aware rule system, a corpus-derived AGG lexicon, and a large-scale evaluation over more than one million agreement pairs from ATB1/2/3 and BAREC, on which AGG improves agreement prediction by up to 4.4 accuracy points over GNR features.

Future work includes phrase-level phenomena such as coordination and quantification, integrating AGG into Arabic NLP resources such as CAMeL Tools, expanding lexical coverage, and applying the framework to dialectal and learner Arabic, grammatical error correction and explanation (Elhady et al., 2026), and the evaluation of MT- and LLM-generated Arabic. Future work includes phrase-level phenomena such as coordination and quantification, integrating AGG into Arabic NLP resources such as CAMeL Tools, expanding lexical coverage, applying the framework to dialectal and learner Arabic, grammatical error correction and explanation (Elhady et al., 2026), the evaluation of MT- and LLM-generated Arabic, and automatic induction of AGG values from distributional or contextual representations, reducing the need for manual lexicon expansion.

Limitations

Our evaluation depends on upstream resources, including the morphological analyzer, treebank annotations, and automatically derived structure for BAREC. As a result, some apparent agreement errors reflect resource or extraction artifacts rather than failures of the agreement model itself. Direct comparison with Alkuhlani and Habash (2011) is limited by differences in corpora and construction coverage. Finally, the AGG lexicon is targeted and corpus-derived, leaving unattested nouns unannotated and some classifications judgment-sensitive; some agreement phenomena, such as coordination and quantification, require phrase-level computation beyond the current word-level scope.

Ethics Statement

The primary annotation of the AGG lexicon was conducted by one author; difficult cases, agreement rules, and the final category definitions were reviewed collaboratively by all authors to promote consistency and reduce individual annotator bias. The resulting lexical annotations, rule definitions, and evaluation scripts will be made publicly available to support transparency, reproducibility, and further research in Arabic NLP, subject to the licensing restrictions of the underlying corpora. We used AI writing assistance only for language-level editing of the paper, as permitted by the ACL Policy on Publication Ethics; all analyses, annotations, experiments, and results were produced and verified by the authors.

Acknowledgments

We thank the anonymous reviewers for their careful reading and constructive feedback, which helped improve the clarity and presentation of this work. We also thank Khalid N. Elmadani for helpful discussions and support related to the BAREC resource. We are grateful to the members of CAMeL Lab for their feedback and support throughout this project.

References

Sarah Alkuhlani and Nizar Habash. 2011. A corpus for modeling morpho-syntactic agreement in Arabic: Gender, number and rationality. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 357–362, Portland, Oregon, USA. Association for Computational Linguistics.

Wafa Abdullah Alrajhi, Hend Al-Khalifa, and Abdulmalik AlSalman. 2022. Assessing the linguistic knowledge in Arabic pre-trained language models using minimal pairs. In *Proceedings of the Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 185–193, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Gennaro Chierchia. 1998. Plurality of mass nouns and the notion of "semantic parameter". In Susan Rothstein, editor, *Events and Grammar*, pages 53–103. Kluwer Academic Publishers, Dordrecht.

Luca D'Anna. 2020. Collectives in the Qur'ān revisited: Another possibility of semantic agreement. *Journal of Semitic Studies*, 65(1):147–169.

Khaled Elhady, Omar Kallas, Nizar Habash, and Bashar Alhafni. 2026. ArabiGEE: A hierarchical taxonomy for Arabic grammatical error explanation. *Preprint*, arXiv:2606.10765.

Khalid N. Elmadani, Adel Mahmoud Wizani, Hanada Taha Thomure, and Nizar Habash. 2026. A large and balanced multi-domain arabic corpus annotated for morphology, syntax, and readability. In *Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026)*, pages 11761–11775, Palma, Mallorca, Spain. European Language Resources Association (ELRA).

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 and Ryan Roth. 2009. CATiB: The Columbia Arabic Treebank. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 221–224, Suntec, Singapore. Association for Computational Linguistics.

Nizar Y. Habash. 2010. *Introduction to Arabic Natural Language Processing*, volume 10 of *Synthesis Lectures on Human Language Technologies*. Morgan & Claypool Publishers.

Christian Khairallah, Salam Khalifa, Reham Marzouk, Mayar Nassar, and Nizar Habash. 2024. Camel morph MSA: A large-scale open-source morphological analyzer for Modern Standard Arabic. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2683–2691, Torino, Italia. ELRA and ICCL.

Mohamed Maamouri, Ann Bies, Tim Buckwalter, and Wigdan Mekki. 2004. The Penn Arabic Treebank: Building a large-scale annotated Arabic corpus. In *Proceedings of the NEMLAR Conference on Arabic Language Resources and Tools*, pages 102–109, Cairo, Egypt.

Yuval Marton, Nizar Habash, and Owen Rambow. 2011. Improving Arabic dependency parsing with form-based and functional morphological features. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1586–1596, Portland, Ore-

gon, 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.

Hazel Pearson. 2011. A new semantics for group nouns. In *Proceedings of the 28th West Coast Conference on Formal Linguistics (WCCFL 28)*, pages 160–168, Somerville, MA. Cascadilla Proceedings Project.

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.

Dima Taji, Salam Khalifa, Ossama Obeid, Fadhl Eryani, and Nizar Habash. 2018. An Arabic morphological analyzer and generator with copious features. In *Proceedings of the Fifteenth Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 140–150, Brussels, Belgium. Association for Computational Linguistics.

Mohamad Bilal Zbib, Hasan Abed Al Kader Hammoud, Ammar Mohanna, Nadine Rizk, Fatima Karnib, Sina Moukaled, and Bernard Ghanem. 2026. AraLingBench: A human-annotated benchmark for evaluating Arabic linguistic capabilities of large language models. In *Proceedings of the 2nd Workshop on NLP for Languages Using Arabic Script*, pages 385–393, Rabat, Morocco. Association for Computational Linguistics.

A License

We list below the licenses of the data and tools used in this work, all of which are employed in accordance with their intended use.

- PATB / ATB (Maamouri et al., 2004): Linguistic Data Consortium (LDC) License
- BAREC-10M Corpus (Elmadani et al., 2026): Creative Commons Attribution Share Alike 4.0
- CAMeL Tools (Obeid et al., 2020): MIT License
- CAMeL Morph (Khairallah et al., 2024): MIT License
- CamelParser2.0 (Elshabrawy et al., 2023): MIT License

SHAMS
An Audio-Grounded Pronunciation Benchmark for Levantine Arabic

Ben Sapirstein
Reichman University

Roy Mattar
Independent Researcher

Guy Mor-Lan
The Hebrew University of Jerusalem

Ahlam Mohamed
Tel Aviv University

Letizia Cerqueglini
Tel Aviv University

Morris Alper
University of Miami

Abstract

Levantine Arabic (LA) is spoken by tens of millions of people, creating a pressing need for shared benchmarks to evaluate LA speech-language technologies. Evaluating such technology is particularly challenging given LA's internal diversity and its opaque and non-standardized orthography. We present **SHAMS** (*SHami Annotated Multi-dialect Speech*), a benchmark comprising 1,300 utterances drawn from open audio corpora, balanced across five LA varieties (Urban and Rural Palestinian, and Urban Jordanian, Lebanese, and Syrian). Each utterance is represented across four aligned tiers: audio, unvocalized orthography, diacritized text, and phonetic transcription. This structure supports evaluation of various downstream tasks such as diacritization, grapheme-to-phoneme conversion, automatic speech recognition, and audio-to-phoneme, grounded in audio and stratified by variety. We benchmark open and proprietary models across these tasks to demonstrate the utility of this benchmark for measuring progress across LA. We release SHAMS at https://shams-nlp.github.io.

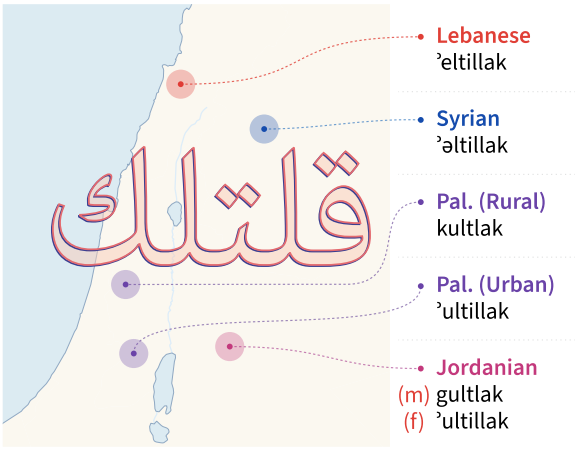


Figure 1: **SHAMS** includes audio and text for the Levantine Arabic varieties shown in the map above. Above, we illustrate different pronunciations of the word قلتلك (colloquial spelling, meaning *I told you*) to show the importance of our phonetic annotations; these pronunciation differences are not indicated in Arabic orthography, even with conventional diacritics.

1 Introduction

As speech technology improves for applications such as assistive technology and conversational AI, progress depends on shared evaluation benchmarks. However, these remain unevenly developed across the world's languages. Levantine Arabic (LA), spoken by tens of millions of people across Palestine, Israel, Jordan, Lebanon, and Syria (Brustad and Zuniga, 2019), poses particular challenges: it shows significant internal diversity across regions and speakers, and its written form is non-standardized and leaves key phonetic features unspecified. As a result, there is a pressing need for shared evaluation benchmarks for LA speech-language tasks.

Most Arabic-speaking communities exist in a state of *diglossia*, where regional varieties such as LA coexist with Modern Standard Arabic (MSA), a codified register learned through formal schooling and used in writing, news, and oratory, but not natively spoken in everyday conversation (Ferguson, 1959). Existing work largely focuses on MSA, with dialectal varieties significantly underrepresented in both corpora and models (Djanibekov et al., 2025; Lau et al., 2025). Recent work finds that even LLMs that can understand dialectal Arabic tend to default to MSA in generation (Robinson et al., 2025).

Proceedings of The Fourth Arabic Natural Language Processing Conference (ArabicNLP 2026), pages 88–114
October 28-29, 2026 ©2026 Association for Computational Linguistics

This is consistent with the well-known Arabic sociolinguistic trend of dialect leveling—the reduction of linguistic differences between varieties through contact and standardization (Al-Wer, 2007; Al-Wer et al., 2022).

Arabic is written in an *abjad*, a script type that represents consonants but omits short vowels and other phonetic features such as gemination (Daniels, 1990). Consider the word بقدر (*bqdr*). Without diacritics, it is ambiguous between three readings: بِقَدْر / بِقَدَر (*bi-qadr / bi-qadar*; 'according to'); بَقْدِر (*baqdir*; 'I can'); and بِقَدِّر (*bi-qaddir*; 'I estimate'). Even with diacritics added, بَقْدِر alone still underspecifies the regional pronunciation—across Palestinian varieties, ق (/q/) surfaces as a glottal stop [ʔ] in urban speech (*ba'dir*), as a voiceless velar [k] in rural speech (*bakdir*), and as a voiced velar [g] in Bedouin speech (*bagdir*). Figure 1 illustrates this diversity in pronunciation.

Arabic dialectology directly studies such regional pronunciations, using phonetic transcriptions of recorded speech (Henkin, 2010; Cerqueglini, 2022, 2024). When transcripts are only available as unvocalized Arabic text, these pronunciation differences are obscured and cannot be studied. However, producing phonetic transcriptions by hand is slow, and has been described as a major bottleneck for data collection in dialectology (Kuparinen, 2025). Automating this transcription could relieve the bottleneck, but for LA there has been no way to measure how well current systems perform the task.

Bridging the gap between written form and pronunciation involves several key tasks, including diacritization (restoring vowel and gemination marks), grapheme-to-phoneme conversion (G2P; mapping orthography to phonetic transcription), automatic speech recognition (ASR; transcribing audio to text), and audio-to-phoneme conversion (A2P; transcribing audio to phonetic transcription). Diacritization has been studied extensively for MSA (Mohamed and Mubarak, 2025), with more limited work on G2P and Arabic dialects (Masmoudi et al., 2019; Lodagala et al., 2025). Dialectal Arabic ASR has been advanced by recent audio corpora (Talafha et al., 2024; Sullivan et al., 2026). These efforts address individual tasks in isolation, and neither source corpus provides diacritics, a phonetic tier, or variety-level stratification; our novel contribution is an evaluation that ties diacritization, G2P, ASR, and A2P together over the same utterances, grounded in audio and stratified by LA variety.

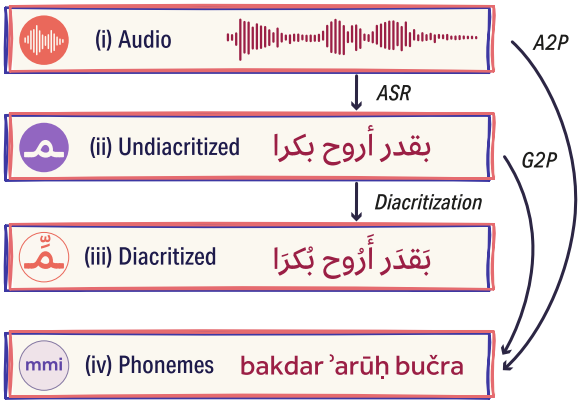


Figure 2: **SHAMS contents and benchmark tasks.** Each utterance in SHAMS comprises four aligned tiers (§2.2): (i) audio, (ii) undiacritized text, (iii) diacritized text and (iv) phonetic transcription. Arrows denote evaluation tasks; e.g., G2P predicts the phonetic transcription from unvocalized text.

To address this, we present **SHAMS** (*SHami Annotated Multi-dialect Speech*), an open benchmark for spoken LA. SHAMS comprises 1,300 utterances drawn from open audio corpora, balanced across varieties by word count, each represented across four aligned tiers (Figure 2). SHAMS is designed as an evaluation benchmark to test the robustness of LA systems beyond their training distribution. Recent work on other Semitic languages has shown that small expert-annotated evaluation benchmarks for tasks like G2P and ASR can drive substantial progress, including enabling training on large unlabeled corpora (Kolani et al., 2025; Melichov et al., 2026). We release SHAMS at the URLs below.[1]

To summarize, our key contributions are:

- **A balanced spoken corpus** of 1,300 utterances drawn from open audio sources, covering five Levantine varieties
- **Multi-tier annotations** aligned to each utterance consisting of diacritized Arabic and phonetic transcription, thus enabling joint evaluation of diacritization, G2P, ASR, and A2P.
- **Baseline results** across open and proprietary models, stratified by variety, demonstrating the benchmark's utility for distinguishing model performance across tasks and varieties.

[1]Data: https://huggingface.co/datasets/shams-nlp/shams. Evaluation code and model outputs: https://github.com/bensapirstein/shams.

2 SHAMS

2.1 Sources and Contents

SHAMS comprises 1,300 utterances totaling roughly 15,500 words and just over two hours of speech, spread across five Levantine varieties: Palestinian Urban, Palestinian Rural, Jordanian, Lebanese, and Syrian. Besides Palestinian Rural, all varieties in the benchmark are urban. The benchmark is roughly balanced by word counts, with about 3,100 words per variety, and contains a mix of speaker genders. Detailed statistics are provided in Table 1.

The benchmark is built on the Levantine portions of two open multi-dialectal Arabic speech corpora. From *Casablanca* (Talafha et al., 2024) we take scripted dialectal television dialogue; from the *LangMedia* archive (Five College Center for World Languages, n.d.), documentary interviews and spontaneous conversation on topics such as food, family, and daily life. The data features speakers of different generations and expressive styles who routinely mix registers and incorporate loanwords from contact languages.

We retain *Casablanca*'s existing utterance segmentation and identifiers, and segment the *LangMedia* recordings into short utterances following Talafha et al. (2024), excluding segments with fewer than three words or heavy background noise. We exclude designated training-split data, as SHAMS is designed solely for evaluation. Where original transcripts do not match the accompanying audio, we correct them manually.

As part of preprocessing we normalize Arabic Unicode across all text tiers, folding presentation forms, removing invisible controls and *tatweel*, and applying canonical composition so that diacritic sequences compare consistently. We apply no orthographic normalization at this stage: for example, *hamza* and *alif* variants, *tāʾ marbūṭa* and *alif maqṣūra* are left as originally transcribed.

2.2 Benchmark Structure

As shown in Figure 2, each utterance in SHAMS includes four aligned tiers: (i) the **audio** segment; (ii) **unvocalized Arabic orthography**, the consonantal text as conventionally written, corrected when needed to match the recording; (iii) **fully diacritized Arabic**, the same text with conventional diacritics matching the pronunciation in the audio; and (iv) a **phonetic transcription** in a Semitic transcription system.

Variety	Source	Utt.	Words	M/F %	Min.
Pal./Urban	Casablanca	22	267	49.8/50.2	1.9
	LangMedia	208	2,908	48.5/51.5	23.1
	both	230	3,175	48.6/51.4	25.0
Pal./Rural	Casablanca	234	3,153	69.0/31.0	21.8
Jordanian	Casablanca	137	1,451	60.0/40.0	10.5
	LangMedia	168	1,722	35.0/65.0	15.0
	both	305	3,173	46.5/53.5	25.5
Lebanese	LangMedia	316	3,008	49.9/50.1	29.1
Syrian	LangMedia	215	2,997	50.0/50.0	26.7
Total		**1,300**	**15,506**	**52.8/47.2**	**128.2**

Table 1: SHAMS composition by variety and source: utterances, words, the male/female share of words in per cent, and duration in minutes. Palestinian Urban and Jordanian each draw on both source corpora; Palestinian Rural comes from Casablanca only, and Lebanese and Syrian from LangMedia only.

2.3 Annotation

Diacritics. Diacritization, also called vocalization, is the process of adding conventional pronunciation markings to consonantal Arabic text. We annotate five categories. *Fatḥa*, *kasra*, and *damma* mark the three short vowels /a/, /i/, and /u/; *shadda* marks gemination of the consonant it sits on; and *tanwīn* marks nunation, a final /n/ added to a short vowel. We do not annotate *sukūn*, the explicit mark of a vowelless consonant.

Phonetic transcription. Even fully diacritized orthography leaves dialectal pronunciation underspecified (§1): for example, the realization of ق is not recoverable from the script. We therefore add a phonetic transcription tier, following the Semitic transcription system adopted by the *Encyclopedia of Arabic Language and Linguistics* (Versteegh, 2006–2009) and used in Arabic dialectological scholarship (Cerqueglini, 2022, 2024; Henkin, 2010; Shahin, 2011). While this notation follows Arabic dialectological conventions, in principle it maps to the International Phonetic Alphabet (IPA) by a deterministic character correspondence. Table 2 illustrates phenomena that surface in the phonetic transcription but not in the orthographic tier.

Annotation principles and procedure. The data exhibits substantial phonetic variability across and within dialects (Appendix A). Our guiding principle is to document this diversity as-is, rather than normalizing to canonical forms. The diacritized and phonetic tiers record the realization heard in

Example 1: Code-switching and *imāla*
selle d'agneau كمان عندك مثل
kamēn *ᶜandek* ***mitl sel danyō***

Example 2: Register mixing
من خبرتك الطويلة بالسوق، قديش تأثر الاقتصاد السوري بالأزمة هي؟
min ḫibritak iṭ-ṭawīle ***bis-sūʾ, ʾaddēš taʾassor il-iqtiṣād*** *is-sūri bi-l-ʾazme hiyye?*

Table 2: **Necessity of phonetic transcription.** Two *LangMedia* utterances (Lebanese, top; Syrian, bottom) show phenomena in SHAMS that are visible only in the phonetic tier. In Example 1, the French loan is written as pronounced (*sel danyō*), مثل shifts the fricative ث to plosive [t] and كمان carries *imāla* (*kamēn*). In Example 2, the same speaker realizes ق as [q] in the MSA-register noun *il-iqtiṣād* and as [ʔ] in the everyday *bis-sūʾ* and *ʾaddēš*, and shifts the interdental ث to a sibilant in *taʾassor*.

each recording, so the same orthographic word may carry different diacritics and phonetic transcriptions across utterances, even within the same dialect.

Both pronunciation tiers were produced semi-automatically. For 520 initial utterances, phonetic transcriptions were produced by hand and aligned to the Arabic orthography to place diacritics. For the remaining 780 utterances, Levanti (Mor-Lan, 2024) produced an automatic first pass over both tiers, which annotators then corrected. Consonantal spelling was also corrected against the audio during annotation. Annotators worked against the audio throughout, and every utterance was reviewed by an author. Because Levanti is also evaluated as a baseline (§3.1), we mark it as potentially in-distribution in result tables and we note that 161 utterances (12.4%) carry diacritics identical to its output after human review. Full annotation conventions are provided in Appendix A.

Annotators. Diacritics were added by three native speakers of Palestinian Arabic, led by a PhD student specializing in Arabic dialectology. Phonetic transcriptions were produced by a co-author with a PhD in Semitic linguistics and publications in Arabic dialectology.

2.4 Annotation Reliability

A second native speaker of Palestinian Arabic independently re-annotated a stratified subset of 150 utterances, 30 per variety, under the same audio-guided protocol. We measure agreement over the

	Utt.	Slots	Agree	κ [95% CI]
Overall	150	7,290	93.2%	0.898 [0.886, 0.909]
Pal./Urban	30	2,237	93.1%	0.898 [0.881, 0.917]
Pal./Rural	30	1,484	95.4%	0.930 [0.913, 0.944]
Jordanian	30	1,008	91.3%	0.867 [0.823, 0.901]
Lebanese	30	1,078	93.2%	0.896 [0.863, 0.927]
Syrian	30	1,483	92.4%	0.886 [0.861, 0.912]
Palestinian	60	3,721	94.0%	0.911 [0.898, 0.924]
Non-Palestinian	90	3,569	92.4%	0.883 [0.865, 0.901]

Table 3: Inter-annotator agreement on the diacritization subset. The unit is the diacritic slot: one per Arabic letter, labelled with the marks it carries. Words whose consonantal skeletons differ between the two annotators are an orthographic divergence rather than a diacritic disagreement and are excluded (142 of 1,881 words). Intervals are cluster bootstraps resampling utterances.

diacritic slot, one per Arabic letter, giving 7,290 paired observations. Agreement is 93.2%, with Cohen's $\kappa = 0.898$ (Table 3). Intervals are 95% confidence intervals from a cluster bootstrap resampling full utterances, in order to account for the non-independence of slots within an utterance.

Protocol. The second annotator was shown the first annotator's consonantal skeleton, so the comparison measures diacritic reliability only. The subset and both annotators' labels are released with the benchmark.

Cross-variety agreement. Agreement is high ($\kappa \geq 0.86$) across all varieties. All diacritization annotators are native Palestinian Arabic speakers; agreement is slightly higher on Palestinian material ($\kappa = 0.911$) than elsewhere (0.883), a difference of 0.027 (95% CI $[0.006, 0.050]$), consistent with the audio-grounded protocol reducing reliance on annotators' native-dialect intuitions.

Disagreements. Most disagreement is about whether a vowel is marked at all (عَلَى / عَلى), then whether to geminate (بَسّ / بَس), and then which vowel to indicate (جِدًّا / جِدًّا). A separate set of words differs in spelling rather than diacritics, mostly *hamza* seating (إنو / انو), and sometimes a dialectal form (متل / مثل).

2.5 Release and Licensing

SHAMS annotations are released under terms consistent with each source corpus's license. Utterances sourced from LangMedia and accompanying annotations are released in compliance with LangMedia's CC BY-SA license. Annotations linked

to Casablanca utterances are released with identifiers without redistributing audio; written correspondence with its authors confirmed this format is consistent with their CC BY-NC-ND license. Full license terms are provided with the released dataset.

3 Experiments and Results

We evaluate models on four tasks enabled by the SHAMS tier structure: diacritization, grapheme-to-phoneme (G2P) conversion, automatic speech recognition (ASR), and audio-to-phoneme (A2P) conversion. We test both dedicated systems and general-purpose LLMs. We report Diacritic Error Rate (DER) for diacritization, computed as in Catt (Alasmary et al., 2024), which aligns the diacritic sequence of each letter and word boundary, Character Error Rate (CER) for remaining tasks, as well as Word Error Rate (WER) for ASR. Results are reported in aggregate and stratified by variety; per-source results for the two varieties drawn from both corpora are in §B.2.

Scoring. Every score is computed after an evaluation-time normalization, applied symmetrically to references and hypotheses, at two levels. The *strict* tier scores the output as written; the *neutral* tier additionally folds spelling and notation that carry no phonemic contrast in Levantine, such as the *alif* and *tāʾ marbūṭa* variants in Arabic script and the definite-article vowel in the phonetic tier. ASR, G2P and A2P report the neutral tier, and the small grey figure beside each score gives the difference to the strict tier: the share of the error that is spelling or notation convention rather than recognition. Diacritization has its own fold (§3.1). The full rules are in Appendix B.1.

All dedicated systems are run zero-shot, while LLMs use a few-shot prompt derived from our annotation guidelines (with examples not from our benchmark). Full experimental details, including models (§B.4), configuration settings (§B.5), and prompt texts (§D), are provided in the Appendix.

3.1 Diacritization

Task description. Given an unvocalized Arabic utterance, predict the fully diacritized form, restoring the five diacritic categories mentioned in §2.3. The MSA case endings *-un*/*-in* are all but absent from the reference and, together with *sukun*, are normalized out of both references and predictions before scoring. The short vowels, *shadda*, and the accusative *tanwīn* are always scored.

	Gemini	Opus 4.8	GPT-5.5	Shakkala	Mishkal	CATT	Levanti*
DER (%)							
Overall	**11.6**	18.2	15.0	25.6	27.4	24.4	7.8
Pal./Urban	**11.3**	18.4	14.6	24.9	26.4	23.9	7.4
Pal./Rural	**10.9**	17.3	14.4	26.0	28.1	24.8	6.5
Jordanian	**11.3**	17.8	14.8	25.1	26.8	24.0	7.1
Lebanese	**12.2**	18.4	15.7	26.9	28.6	25.4	8.8
Syrian	**12.1**	18.8	15.6	25.1	27.2	24.2	9.3

Table 4: Diacritization results per variety (DER %, ↓) on all 1300 utterances. **Bold** = best per row, excluding Levanti (see asterisk). * Levanti produced silver diacritization labels used to annotate a portion of the dataset, so its score on those is not an independent measurement.

Models. We evaluate four dedicated systems; **Shakkala** (Barqawi, 2017), **Mishkal** (Zerrouki, 2014), **Catt** (Alasmary et al., 2024) (all designed for MSA), and **Levanti** (Mor-Lan, 2024) (designed for Levantine); alongside three LLMs: **Gemini 3.5 Flash**, **GPT 5.5**, and **Claude Opus 4.8**.

Results. Table 4 reports DER overall and by variety. Levanti reaches the lowest DER (7.8%), ahead of prompted Gemini 3.5 Flash (11.6%). GPT 5.5 (15.0%) and Claude Opus 4.8 (18.2%) form a middle tier, and the MSA-oriented Catt, Shakkala and Mishkal trail between 24.4% and 27.4%. Levanti's score is marked in the table, since it produced the silver diacritization used in annotating a portion of the utterances in the dataset (§2.3).

Analysis. Levantine adaptation matters more than system type or scale. The MSA-trained dedicated systems trail by a wide margin, while the dedicated Levantine system and the Gemini lead, with the larger LLMs forming a middle tier. A compact general LLM prompted with our Levantine-aware guidelines thus matches a purpose-built Levantine diacritizer, and outperforms both larger LLMs.

MSA systems predominantly over-mark word-final case endings. Our primary DER already excludes the MSA case-*tanwīn*, which occurs once in the reference; even so, these systems retain large residual error on the word-final vowels (*iʿrāb*) that Levantine speech does not realize. Counting the case-*tanwīn* as well raises Catt, Shakkala and Mishkal to 26.8%, 27.9% and 28.8% without changing the ranking. Table 5 illustrates the pattern: MSA systems impose word-final case vowels and the definite article *al-* (vs. dialectal *il-*), whereas Levanti and Gemini track the dialectal reference.

Reference	Model output
و كِيف بِدُّو يِعرِف اِنُّو خَبَّرنَاهُم Jordanian	Catt: وَ **كَيْفَ** بَدُوٌّ **يُعْرَفُ** انْو خَبَرْنَاهُمْ Gemini: و كِيف بِدّو يِعرِف اِنّو خَبَّرنَاهُم
نِمشِي بَكِّير لَأَنُّه فِي هَالمَسَافِة كِلّهَا Lebanese	Mishkal: نَمِشِيٌّ بِكِيرٍ لأنّهُ فِي هالمسافة **كُلّهَا** Gemini: نِمشِي بَكِّير لَأَنُّه فِي هَالمَسَافِة كِلّهَا
نِحنَا هَلَّق بمَوسَم، Lebanese	Mishkal: **نُحْنَا** هلق **بِمَوْسِمٍ** Levanti: نِحْنَا هَلَّق بْمَوْسِم،

Table 5: Qualitative diacritization outputs: an MSA-oriented dedicated system (Catt, Mishkal) against a Levantine-aware system (Gemini, Levanti) on the same utterance. **Bold** marks words that still differ from the reference after normalization; the caption cites them undiacritized. Row 1 (Jordanian): Catt adds MSA case endings, a final vowel on كيف and يعرف that Levantine speech drops; Gemini matches the reference. Row 2 (Lebanese): Mishkal opens كلها with damma where the reference has kasra. Row 3 (Lebanese): Mishkal opens نحنا the same way, and vocalizes the prefix of بموسم, which Levantine leaves unvowelled; Levanti keeps the dialectal reading.

	Gemini	Opus 4.8	GPT-5.5	JAIS-8B	Levanti*	Charsiu
CER (%)						
Overall	**11.9** -0.4	12.0 -0.7	12.2 -0.4	40.8 -0.8	18.3 -4.1	34.8 -2.4
Pal./Urban	12.1 -0.3	**11.0** -0.8	12.2 -0.3	41.8 -0.8	16.2 -4.0	32.3 -2.5
Pal./Rural	**11.8** -1.2	11.9 -0.9	12.6 -1.2	43.0 -1.1	17.6 -3.7	36.1 -2.5
Jordanian	11.6 -0.2	**11.3** -0.6	11.9 -0.3	36.6 -0.8	17.2 -3.7	34.0 -2.3
Lebanese	**13.0** -0.1	13.6 -0.6	13.1 -0.1	39.1 -0.5	20.4 -4.0	36.5 -2.2
Syrian	**11.2** -0.1	12.3 -0.6	11.3 -0.1	43.8 -0.8	20.2 -5.1	35.4 -2.6

Table 6: G2P results per variety (CER %, ↓) on all 1300 utterances, under *neutral* scoring. Small grey figures give the difference from the *strict* tier. **Bold** = best per row, excluding Levanti (see asterisk). * Levanti produced the expansion utterances' first-pass transcription (§2.3).

3.2 G2P

Task description. Given an undiacritized Arabic text, predict its phonetic transcription. The target uses the Semitic phonetic transcription system (§2.2). Because the systems emit different alphabets, outputs and targets are mapped onto the reference alphabet before scoring; the mapping is not exhaustive.

Models. We evaluate **CharsiuG2P** (Zhu et al., 2022) and **Levanti** (Mor-Lan, 2024) as dedicated G2P systems. For CharsiuG2P, we use the multilingual byT5 small model. We further test the LLMs Gemini 3.5 Flash, GPT-5.5, Claude Opus 4.8, and Jais v2 8B.

Results. Table 6 reports CER overall and by variety. The three frontier LLMs achieve similar CER, 11.9–12.2%, clearly outperforming the dedicated systems: Levanti (18.3%) and CharsiuG2P (34.8%).

Analysis. The gap between strict and neutral scoring reveals what drives each system's errors: Levanti's CER drops 4.1 points under neutral normalization, indicating much of its disadvantage is transcription convention (e.g., *āl-* article, *-eh* endings) rather than phonology; CharsiuG2P's gap is 2.4 points, confirming its errors are largely phonological—it restores MSA vocalization, gemination, and *q* for qaf. The LLMs show minimal strict–neutral differences (≤ 0.7), suggesting they already match the reference conventions. All systems struggle with rural Palestinian qaf, uniformly replacing the reference [k] with MSA [q] or urban [ʔ]. Table 7 illustrates these patterns. JAIS returned no transliteration for 131 of the 1,300 utterances, answering in Arabic script or replying to the utterance instead; those are scored as empty.

3.3 ASR

Task description. Given an audio recording, the model predicts the unvocalized Arabic text of the utterance, scored by WER and CER.

Models. We evaluate nine systems. Seven are dedicated speech recognizers: **SeamlessM4T-v2 large** (Barrault et al., 2023) in two configurations targeting MSA and Egyptian Arabic (**Seamless-MSA**, **Seamless-EGY**); **Whisper large v3** (Radford et al., 2023) and its two fine-tunes **Whisper-Levantine** and **BYZÖ-Whisper** (the latter covering eight Arabic dialects and serving as the third-ranking system at NADI 2025 (Talafha et al., 2025; M. Abdullah et al., 2025); it is fine-tuned on the *Casablanca* training split, so it is in-domain for the part of SHAMS drawn from that corpus, with no train/test overlap since we exclude the training splits); **MMS-1b** (Pratap et al., 2024); and **Cohere Transcribe** (Mack et al., 2026). Two are audio language models prompted to transcribe, **Gemini 3.5 Flash** and **GPT-Audio**. All are evaluated zero-shot with respect to SHAMS.

Source	Output
Notational — raw ↪ normalized	
البصل مفروم Lebanese	**Ref:** el-baṣal mafrūm ↪ ilbaṣal mafrūm Charsiu: aːlbasˤal mafruːm Levanti: ālbaṣal mafrūm Claude: il-baṣal mafrūm ↪ *all normalize to* ilbaṣal mafrūm
Phonological — persists after normalization	
أنا من لبنان Lebanese	**Ref:** ʾana min libnēn Charsiu: **ʾannā minna lubnān** Levanti: ʾana min **lubnān** Claude: ʾana min **lubnān**
إلحق يا سيدي Pal./Rural	**Ref:** ilḥak ya sīdi Charsiu: **ilḥaqqa yā sēadī** Levanti: **ʾilḥaq** ya sīdi Claude: **ilḥaʾ** ya sīdi

Table 7: G2P examples. *Top:* different raw notations converge after neutral normalization (↪). *Bottom* (normalized): phonological errors persist; **bold** marks words that differ from the normalized reference. CharsiuG2P adds MSA gemination/epenthesis; no system recovers the rural Palestinian qaf (*ḥak*, [k]), which is rendered as MSA [q] or urban [ʾ].

Results. Table 8 reports WER and CER overall and by variety, on all 1,300 utterances. The two prompted audio LLMs and the dialect-targeted systems lead, with Gemini 3.5 Flash strongest overall by a clear margin (27.6 WER, 9.6 CER). Among the open-weight systems the separating factor is dialectal adaptation rather than architecture or scale: Seamless-EGY improves on its MSA-targeted counterpart on every variety, and BYZÖ-Whisper reaches the second-best CER overall.

Analysis. Models designed for MSA (Seamless-MSA, Whisper v3, MMS-1b) commonly output MSA equivalents of reference dialectal forms (Table 9), likely reflecting training on dialectal audio with MSA captions. Dialect-adapted systems are more faithful, including Seamless-EGY, likely because Egyptian and Levantine share lexical and grammatical features that diverge from MSA. The two scoring levels are most consequential for Whisper-Levantine, whose 15.0-point strict–neutral gap is by far the largest in the table: the model emits no أ, إ, آ, ة or ى, accounting for its large strict–neutral gap.

3.4 A2P

Task description. Given an audio recording, predict its phonetic transcription directly.

Models. We evaluate two audio-LLMs prompted with audio and textual instructions requesting a phonetic transcription: **GPT-Audio** and **Gemini 3.5 Flash**.

Results. Table 10 reports CER overall and by variety. Gemini 3.5 Flash attains the lowest error (20.5%), with GPT-Audio behind at 24.9%.

Analysis. Table 11 shows the two systems are usually faithful, differing mainly in notation and vowel quality; the dominant residual error is occasional reversion to Arabic script on hard utterances, which is scored as fully wrong.

We additionally evaluate both audio LLMs on ASR, which permits a within-model comparison of orthographic and phonetic transcription. Gemini 3.5 Flash reaches 9.6% CER on ASR against 20.5% on direct A2P, and GPT-Audio 14.4% against 24.9%, both tasks under the neutral tier. The two references, Arabic orthography and Semitic phonetic transcription, are not on a common scale, so the comparison is qualitative: both models transcribe orthography more reliably than phonetics.

To locate where the error arises, we compare direct A2P against an ASR → G2P cascade over the same utterances, holding the cascade's G2P stage fixed at Gemini so that the contrast is cascade versus direct rather than one cascade against another. The cascade reaches 18.9% CER for Gemini and 23.3% for GPT-Audio, against 20.5% and 24.9% for direct A2P, while a gold-transcript → G2P oracle reaches 13.0%. Direct phonetic transcription therefore offers no advantage over transcribing and then converting. Roughly 13 CER points are intrinsic to the phonetic conversion itself, with ASR error propagation adding a further six. Because the G2P stage is Gemini's in both cascades, the GPT-Audio cascade is a hybrid, and its margin over GPT-Audio's direct A2P is an upper bound. Per-variety figures for all three paths are in Table 14.

4 Related Work

Diacritization and G2P. Most Arabic diacritization work targets MSA or Classical Arabic (Zerrouki and Balla, 2017; Fadel et al., 2019; Alasmary et al., 2024; Mohamed and Mubarak, 2025), with recent character-level and pretrained approaches achieving strong results (Elmadany et al., 2022; Elgamal et al., 2024). Extension to dialects remains limited: Harrat et al. (2013) is an early effort on Algerian, Masmoudi et al. (2019) addresses Tunisian,

	Sml-MSA	Sml-EGY	Whisper v3	Whisper-Lev.	MMS-1b	BYZÖ-W.	Cohere	Gemini	GPT Audio
WER (%)									
Overall	40.4 -4.6	36.3 -4.1	39.3 -5.3	38.9 -15.0	72.5 -1.7	37.8 -6.0	35.5 -5.7	**27.6** -5.4	34.9 -5.5
Pal./Urban	31.9 -4.9	29.1 -4.5	29.5 -6.5	35.3 -16.0	67.5 -2.1	39.2 -9.1	35.8 -5.7	**26.5** -5.5	35.9 -5.5
Pal./Rural	47.6 -3.8	42.9 -2.7	45.4 -3.2	34.9 -14.2	77.9 -0.7	**20.3** -3.1	33.2 -4.4	32.0 -2.9	41.2 -3.1
Jordanian	34.7 -5.8	31.8 -4.3	39.5 -5.5	30.3 -16.6	68.2 -1.9	34.4 -8.1	27.1 -6.5	**21.0** -5.5	28.7 -5.5
Lebanese	44.3 -4.3	38.8 -4.5	45.7 -5.5	44.7 -14.1	76.3 -1.9	50.2 -4.9	42.5 -5.6	**27.2** -6.1	34.0 -6.3
Syrian	43.9 -4.3	39.3 -4.5	36.9 -5.9	50.4 -14.4	72.9 -2.0	45.8 -4.7	39.6 -6.2	**31.5** -7.2	34.5 -7.3
CER (%)									
Overall	17.8 -1.4	14.1 -1.2	20.6 -1.5	13.7 -3.9	27.0 -1.1	11.9 -1.6	16.8 -1.6	**9.6** -1.5	14.4 -1.6
Pal./Urban	13.3 -1.5	10.8 -1.3	10.9 -1.8	11.4 -4.5	23.8 -1.2	11.4 -2.3	19.2 -1.6	**9.7** -1.5	16.2 -1.7
Pal./Rural	19.1 -1.2	16.3 -0.8	32.0 -1.0	10.5 -3.8	29.3 -0.7	**5.9** -0.8	11.7 -1.3	10.3 -0.9	16.6 -1.3
Jordanian	16.6 -1.6	12.8 -1.2	18.2 -1.5	10.1 -4.2	26.1 -1.1	10.3 -2.1	12.0 -1.8	**7.1** -1.5	11.2 -1.6
Lebanese	19.3 -1.3	14.9 -1.2	24.0 -1.5	17.5 -3.3	29.0 -1.0	16.3 -1.5	22.8 -1.4	**10.1** -1.4	14.7 -1.6
Syrian	21.0 -1.5	16.2 -1.4	18.9 -1.8	18.9 -3.9	27.2 -1.4	15.5 -1.3	18.2 -1.8	**10.9** -2.1	13.2 -2.1

Table 8: ASR results per variety (WER/CER %, ↓) on all 1300 utterances, under *neutral* scoring. Small grey figures give the difference from the *strict* tier. **Bold** = best per row.

Reference	MSA model	Dialect model
بس هيدي كانت مدرسة كتير منيحة Lebanese	كانت **لكن هذه** مدرسة **جيدة جداً** Seamless-MSA	بس **هادي** كانت مدرسة كتير منيحة Seamless-EGY
لا مش بس أبو الفوز Pal./Rural	**هذا ليس فقط** الفوز Seamless-MSA	لا مش بس ابو الفوز Whisper-Lev
بيطلع النوعين... دراق بكفيا ودراق نبع الصفا Lebanese	**هناك نوعان...من الدراجات: الدراجات الكافية و الدراجات الصالحة للسفر** Seamless-MSA	
بدك إياني أقطع رزقي بإيدي؟ Jordanian	**هل تريد أن أضع** رزقي **في يديك؟** Whisper-v3	

Table 9: MSA substitution in ASR. **Bold** marks words that differ from the reference. Rows 1–2 show all three: reference, MSA-biased output, and dialect-adapted output; the dialect model preserves the spoken forms. Rows 3–4 show further MSA errors (hallucinated vocabulary, full-sentence replacement).

	Gemini	GPT-Audio
CER (%)		
Overall	**20.5** -0.7	24.9 -0.5
Pal./Urban	**21.1** -0.9	27.2 -0.6
Pal./Rural	**21.0** -0.6	28.6 -0.5
Jordanian	**18.0** -0.6	23.4 -0.4
Lebanese	**20.8** -0.7	23.2 -0.5
Syrian	**22.4** -0.7	23.4 -0.5

Table 10: A2P results per variety (CER %, ↓) on all 1300 utterances, under *neutral* scoring. Small grey figures give the difference from the *strict* tier. **Bold** = best per row.

while Shatnawi et al. (2024) and Ghannam et al. (2025) explore speech-informed and multimodal diacritic restoration. Arabic G2P has traditionally been decomposed into diacritization followed by rule-based phoneme mapping (Biadsy et al., 2009; Ali et al., 2014; Lodagala et al., 2025), while recently, Hawasly et al. (2025) introduced end-to-end G2P for Egyptian Arabic. A parallel gap for Modern Hebrew is addressed by Kolani et al. (2025) and Melichov et al. (2026), who introduce audio-grounded G2P benchmarks, enhanced diacritization, and speech-supervised G2P methods. Recent work has shown that off-the-shelf LLMs can outperform specialized Arabic diacritization models, though evaluations have focused exclusively on MSA (Toyin et al., 2025). To the best of our knowledge, no dedicated diacritization or G2P benchmark exists for LA.

Dialectal Arabic Resources and ASR. Dialectal corpora have grown from single-region datasets (Jarrar et al., 2017; Kwaik et al., 2018; Nayouf et al., 2023) to multi-dialectal collections (Bouamor et al., 2018; Talafha et al., 2024). For spoken Arabic, large resources such as Common Voice (Ardila et al., 2020) and MASC (Al-Fetyani et al., 2023) provide broad but coarse-grained coverage, often collapsing dialect distinc-

Reference	GPT-Audio	Gemini 3.5 F.
šwayyit zēt Lebanese	šwayyit zēt	šwayyit zēt
ʾinti lāzim tistaġil-li hīk mōqif mitl ᶜādatek Jordanian	**inti** lāzim **tistaghillī hēk mawqif mitil ᶜādtik.**	**inti** lāzim **tistagallī hēk mawqif** mitl **ᶜādtik**
kamān il-ḥubb yaᶜni aḥyānan Syrian	kamān il-ḥubb yaᶜni aḥyānan…	kamān il-ḥubb **yaᶜnī ʾaḥyānan**
masalan, f-lnaftariḍ ʾanno hīyye ᶜayynat ᶜala tlāt bnāt. … Syrian	**بس أنا بفكر إنو هي عيلة على كلمة**	**matalan falan-naftariḍ innu hiyyi ᶜanīdi** ᶜala **talat banāt,** …

Table 11: Qualitative A2P outputs. Words diverging from the reference are shown in **bold**. Row 1 (Lebanese) is transcribed perfectly by both systems. Row 2 (Jordanian) illustrates the typical case: minor vowel-quality drift (*hīk* → *hēk*), notation variants (*ġ* → *gh*), and reduced vowels. Row 3 (Syrian) is again near-perfect, with only long-vowel marking differences in Gemini. Row 4 exposes the dominant failure mode on a harder Syrian clip: GPT-Audio reverts entirely to Arabic script (scored as fully wrong), while Gemini stays in phonetic transcription but introduces errors throughout.

tions into a single "Arabic" label (Lau et al., 2025). Dialect-specific ASR remains data-limited (Waheed et al., 2023; Djanibekov et al., 2025). The recent Arab Voices framework of Sullivan et al. (2026) spans dozens of datasets and dialects, but treats Levantine as a single variety and targets ASR alone. The NADI 2025 shared task (Talafha et al., 2025) recently introduced diacritic restoration for spoken dialects, but covers eight country-level dialects in a time-limited evaluation setting. Its Subtask 2 covered multidialectal ASR including Levantine, with six participating teams, notably Munsit (Salhab et al., 2025) and ELYADATA & LIA (Elleuch et al., 2025), along with related fine-tuning work such as M. Abdullah et al. (2025). AL-QASIDA (Robinson et al., 2025) evaluates LLM proficiency across eight dialectal varieties but focuses on text modeling rather than speech tasks. Cross-dialectal evaluation frameworks have been developed for other languages, notably Multi-VALUE for English (Ziems et al., 2023); no comparable resource exists for LA.

Phone Recognition and Dialectal Transcription. Transcribing speech directly to phones or phonemes has been studied under several names: multilingual phone recognition (Li et al., 2020; Zhu et al., 2025; Li et al., 2025), speech-to-IPA transcription for language documentation (Taguchi et al., 2023), audio-to-IPA models used to train and evaluate downstream speech systems (Kolani et al., 2025; Melichov et al., 2026), and automatic dialectal transcription for Finnish and Norwegian (Kuparinen, 2025).

5 Conclusion

We have presented SHAMS, a benchmark for Levantine Arabic speech-language technology comprising 1,300 utterances with four aligned annotation tiers. Our experiments on diacritization, G2P, ASR and A2P show that the benchmark effectively differentiates model performance across tasks and varieties, revealing both the progress of dialect-adapted models and LLMs and the substantial gaps that remain. The fine-grained annotations in SHAMS provide particular value in surfacing performance variation across different Levantine varieties, underscoring the importance of variety-aware evaluation.

Limitations

The benchmark's sample size and coverage of the Levantine dialect space are both limited; for example, lacking Bedouin varieties, rural Jordanian, Lebanese, and Syrian varieties, and diaspora speech. Most varieties are drawn from a single source; for these, differences in content, speaker, and recording quality are confounded with dialect. Only Jordanian draws substantially on both and supports the within-variety source comparison (§B.2); there, the two sources also differ in speaker gender, so source and gender are entangled. Speaker gender is imbalanced in some varieties, which future work could mitigate. The Casablanca television data features actors who may not be native speakers of the portrayed variety, which could affect the representativeness of those segments. Inter-annotator agreement is reported for the diacritization tier over a 150-utterance subset (§2.4); the phonetic and orthographic tiers were each produced by a single expert. The set of mod-

els and tasks tested could also be expanded, for example with audio-guided diacritization.

Ethical Considerations

Our data is drawn from existing open audio corpora; licensing and release details are in Section 2.5. Annotations were produced by co-authors and collaborators with linguistics training, including native speakers of Palestinian Arabic (§2.3). This work is motivated by concerns about the under-representation of Arabic dialectal varieties in speech technology (Traberg et al., 2026; Fleisig et al., 2024; Cerqueglini, 2026); our benchmark provides an instrument for measuring how well current systems serve LA speakers across varieties, and supporting work on increasing this representation.

Acknowledgements

We thank Bessan Adi and Ivan Khuri for their contributions to the annotation effort, and Shai Gordin for providing infrastructure enabling this project.

References

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.

Enam Al-Wer. 2007. The formation of the dialect of amman: From chaos to order. In *Arabic in the City*, pages 69–90. Routledge.

Enam Al-Wer, Uri Horesh, Bruno Herin, and Rudolf De Jong. 2022. *Arabic sociolinguistics*. Cambridge University Press.

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.

Ahmed Ali, Yifan Zhang, Patrick Cardinal, Najim Dahak, Stephan Vogel, and James Glass. 2014. A complete kaldi recipe for building arabic speech recognition systems. In *2014 IEEE spoken language technology workshop (SLT)*, pages 525–529. IEEE.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. 2020. Common voice: A massively-multilingual speech corpus. In *Proceedings of the twelfth language resources and evaluation conference*, pages 4218–4222.

Ahmad Barqawi. 2017. Shakkala: Arabic text vocalization using deep learning. `https://github.com/Barqawiz/Shakkala`.

Loïc Barrault, Yu-An Chung, Mariano Cora Meglioli, David Dale, Ning Dong, Paul-Ambroise Duquenne, Hady Elsahar, Hongyu Gong, Kevin Heffernan, John Hoffman, and 1 others. 2023. Seamlessm4t: Massively multilingual & multimodal machine translation. *arXiv preprint arXiv:2308.11596*.

Fadi Biadsy, Nizar Habash, and Julia Hirschberg. 2009. Improving the arabic pronunciation dictionary for phone and word recognition with linguistically-based pronunciation rules. In *Proceedings of human language technologies: The 2009 annual conference of the North American chapter of the association for computational linguistics*, pages 397–405.

Houda Bouamor, Nizar Habash, Mohammad Salameh, Wajdi Zaghouani, Owen Rambow, Dana Abdulrahim, Ossama Obeid, Salam Khalifa, Fadhl Eryani, Alexander Erdmann, and 1 others. 2018. The madar arabic dialect corpus and lexicon. In *Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018)*.

Kristen Brustad and Emilie Zuniga. 2019. Levantine arabic. In *The Semitic Languages*, pages 403–432. Routledge.

L. Cerqueglini. 2026. Humic (human–machine interaction), university of surrey, guildford, june 16th 2026. In *Lēš byiḥčīš? ('Why Don't You Speak?'): Conversational AI at the Margins – The Case of Palestinian Arabic*.

Letizia Cerqueglini. 2022. Negev bedouin daḥīyah songs: Celebrating brotherhood above war and love. *Folia Orientalia*, 59:95–126.

Letizia Cerqueglini. 2024. Ya ṛaḅḅí ! tanṭīni biné:t!: Ǧbēne in the traditional muṯallaṯ arabic of bāḳa l-ġarbiyye. *Folia Orientalia*, 60:227–275.

Peter T. Daniels. 1990. Fundamentals of grammatology. *Journal of the American Oriental Society*, 110(4):727–731.

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.

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.

Haroun Elleuch, Youssef Saidi, Salima Mdhaffar, Yannick Estève, and Fethi Bougares. 2025. ELYADATA

& LIA at NADI 2025: ASR and ADI subtasks. In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 762–766, Suzhou, China. 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.

Ali Fadel, Ibraheem Tuffaha, Mahmoud Al-Ayyoub, and 1 others. 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.

Charles A Ferguson. 1959. Diglossia. *word*, 15(2):325–340.

Five College Center for World Languages. n.d. LangMedia: Arabic. https://www.langmedia.fivecolleges.edu/languages/arabic. Accessed 2026-06-15.

Eve Fleisig, Genevieve Smith, Madeline Bossi, Ishita Rustagi, Xavier Yin, and Dan Klein. 2024. Linguistic bias in ChatGPT: Language models reinforce dialect discrimination. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13541–13564, Miami, Florida, USA. Association for Computational Linguistics.

Ahmad Ghannam, Naif Alharthi, Faris Alasmary, Kholood Al Tabash, Shouq Sadah, and Lahouari Ghouti. 2025. Abjad ai at nadi 2025: Catt-whisper: Multimodal diacritic restoration using text and speech representations. In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 757–761.

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

Salima Harrat, Mourad Abbas, Karima Meftouh, and Kamel Smaili. 2013. Diacritics restoration for arabic dialects. In *INTERSPEECH 2013-14th Annual Conference of the International Speech Communication Association*.

Majd Hawasly, Hamdy Mubarak, Ahmed Abdelali, and Ahmed Ali. 2025. Dialg2p: Dialectal grapheme-to-phoneme. arabic as a case study. In *Proceedings of The Third Arabic Natural Language Processing Conference*, pages 466–471.

Roni Henkin. 2010. *Negev Arabic: Dialectal, sociolinguistic, and stylistic variation*. Harrassowitz Verlag.

Mustafa Jarrar, Nizar Habash, Faeq Alrimawi, Diyam Akra, and Nasser Zalmout. 2017. Curras: an annotated corpus for the palestinian arabic dialect. *Language Resources and Evaluation*, 51(3):745–775.

Yakov Kolani, Maxim Melichov, Cobi Calev, and Morris Alper. 2025. Phonikud: Hebrew grapheme-to-phoneme conversion for real-time text-to-speech. *arXiv preprint arXiv:2506.12311*.

Olli Kuparinen. 2025. Automatic dialectal transcription: An evaluation on finnish and norwegian. In *Proc. Interspeech 2025*, pages 2390–2394.

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)*.

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.

Chin-Jou Li, Kalvin Chang, Shikhar Bharadwaj, Eunjung Yeo, Kwanghee Choi, Jian Zhu, David Mortensen, and Shinji Watanabe. 2025. Powsm: A phonetic open whisper-style speech foundation model. *arXiv preprint arXiv:2510.24992*.

Xinjian Li, Siddharth Dalmia, Juncheng Li, Matthew Lee, Patrick Littell, Jiali Yao, Antonios Anastasopoulos, David R Mortensen, Graham Neubig, Alan W Black, and 1 others. 2020. Universal phone recognition with a multilingual allophone system. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8249–8253. IEEE.

Vasista Sai Lodagala, Lamya Alkanhal, Daniel Izham, Shivam Mehta, Shammur Absar Chowdhury, Aqeelah Makki, Hamdy S Hussein, Gustav Eje Henter, and Ahmed Ali. 2025. Sawtarabi: A benchmark corpus for arabic tts. standard, dialectal and code-switching. In *Proc. Interspeech 2025*, pages 4793–4797.

Badr M. Abdullah, Yusser Al Ghussin, Zena Al-Khalili, Ömer Tarik 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 *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 745–751, Suzhou, China. Association for Computational Linguistics.

Julian Mack, Ekagra Ranjan, Walter Beller-Morales, Bharat Venkitesh, and Pierre Richemond. 2026. cohere-transcribe-03-2026 (revision d96e814).

Abir Masmoudi, Salima Mdhaffar, Rahma Sellami, and Lamia Hadrich Belguith. 2019. Automatic diacritics restoration for tunisian dialect. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 18(3):1–18.

Maxim Melichov, Yakov Kolani, and Morris Alper. 2026. Renikud: Audio-supervised hebrew grapheme-to-phoneme conversion. *arXiv preprint arXiv:2606.20179*.

Abubakr Mohamed and Hamdy Mubarak. 2025. Advancing arabic diacritization: Improved datasets, benchmarking, and state-of-the-art models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 16707–16719.

Guy Mor-Lan. 2024. Levanti: A levantine arabic dataset. https://huggingface.co/datasets/guymorlan/levanti.

Amal Nayouf, Tymaa Hammouda, Mustafa Jarrar, Fadi Zaraket, and Mohamad-Bassam Kurdy. 2023. Nâbra: Syrian arabic dialects with morphological annotations. In *Proceedings of ArabicNLP 2023*, pages 12–23.

Vineel Pratap, Andros Tjandra, Bowen Shi, Paden Tomasello, Arun Babu, Sayani Kundu, Ali Elkahky, Zhaoheng Ni, Apoorv Vyas, Maryam Fazel-Zarandi, and 1 others. 2024. Scaling speech technology to 1,000+ languages. *Journal of Machine Learning Research*, 25(97):1–52.

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.

Nathaniel Romney Robinson, Shahd Abdelmoneim, Kelly Marchisio, and Sebastian Ruder. 2025. Alqasida: Analyzing llm quality and accuracy systematically in dialectal arabic. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 22048–22065.

Mahmoud Salhab, Shameed Sait, Mohammad Abusheikh, and Hasan Abusheikh. 2025. Munsit at NADI 2025 shared task 2: Pushing the boundaries of multidialectal Arabic ASR with weakly supervised pretraining and continual supervised fine-tuning. In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 734–739, Suzhou, China. Association for Computational Linguistics.

Kimary N. Shahin. 2011. Palestinian arabic. Brill, Leiden, The Netherlands.

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.

Peter Sullivan, AbdelRahim Elmadany, Alcides Alcoba Inciarte, and Muhammad Abdul-Mageed. 2026. Arab voices: Mapping standard and dialectal arabic speech technology. *arXiv preprint arXiv:2601.13319*.

Chihiro Taguchi, Yusuke Sakai, Parisa Haghani, and David Chiang. 2023. Universal automatic phonetic transcription into the international phonetic alphabet. *arXiv preprint arXiv:2308.03917*.

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, and 1 others. 2024. Casablanca: Data and models for multidialectal arabic speech recognition. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21745–21758.

Bashar Talafha, Hawau Olamide Toyin, Peter Sullivan, AbdelRahim A Elmadany, Abdurrahman Juma, Amirbek Djanibekov, Chiyu Zhang, Hamad Alshehhi, Hanan Aldarmaki, Mustafa Jarrar, and 1 others. 2025. Nadi 2025: The first multidialectal arabic speech processing shared task. In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 720–733.

Hawau Olamide Toyin, Samar M Magdy, and Hanan Aldarmaki. 2025. Are llms good text diacritizers? an arabic and yor\ub\'a case study. *arXiv preprint arXiv:2506.11602*.

Cecilie Steenbuch Traberg, Jon Roozenbeek, and Sander Van Der Linden. 2026. Ai is turning research into a scientific monoculture. *Communications Psychology*, 4(1):37.

Kees Versteegh, editor. 2006–2009. *Encyclopedia of Arabic Language and Linguistics*, volume 1–5. Brill, Leiden.

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.

Taha Zerrouki. 2014. Mishkal: Arabic text vocalization software. https://github.com/linuxscout/mishkal.

Taha Zerrouki and Amar Balla. 2017. Tashkeela: Novel corpus of arabic vocalized texts, data for auto-diacritization systems. *Data in brief*, 11:147.

Jian Zhu, Farhan Samir, Eleanor Chodroff, and David R Mortensen. 2025. Zipa: A family of efficient models for multilingual phone recognition. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 19568–19585.

Jian Zhu, Cong Zhang, and David Jurgens. 2022. Byt5 model for massively multilingual grapheme-to-phoneme conversion.

Caleb Ziems, William Held, Jingfeng Yang, Jwala Dhamala, Rahul Gupta, and Diyi Yang. 2023. Multi-value: A framework for cross-dialectal english nlp.

In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 744–768.

A Further Annotation Details

This appendix provides detailed annotation conventions used in SHAMS.

Diacritization principles. Our annotation scheme balances phonetic fidelity with preservation of the original orthographic form, guided by readability, consistency, and recoverability of the underlying word. Diacritics capture dialectal pronunciation contrasts not reflected in the consonantal skeleton—for example, distinguishing عَلِي from عَلَيّ, or صُوت from صَوت—without introducing alternative spellings.

Original consonantal forms are generally preserved. The letter ق is retained regardless of whether it is realized as /q/, a glottal stop, /g/, or another variant; *imāla* is represented through vowel diacritics rather than changes to the underlying letters (e.g., سِاعَة rather than سيعة). Consonantal substitutions are reflected only when they do not create ambiguity and improve representation of dialectal speech.

Gemination is consistently marked with *shadda*, as it may affect lexical meaning and morphological structure. Assimilated or reduced forms are segmented into their underlying components (e.g., مِن هُن rather than مِنُّن) to preserve morphological transparency. Dialect-specific lexical and morphological features such as هيدي and verbal patterns such as مِنروح versus بِنروح are retained rather than normalized, as they provide valuable cues for dialect identification.

Transcription conventions. Word-initial *alif* ranges from a full glottal stop to complete elision depending on prosodic context. For example, in 46392_1: *haḍōl dār Abu Fijli. yā salām! ʾawwal ʾiši ...—ʾawwal* and *ʾiši* have glottal onset, whereas *Abu* does not.

Integrated loanwords are transcribed as pronounced rather than according to their source-language spelling (e.g., 46857_1: French *colonel* and *général* are realized as *il-kulunēl il-ǧinirāl*).

Phonetic variation. Because the tier records the realization in the recording, one orthographic word can carry several phonetic forms across the corpus. The reflex of ق varies: قاعد (‘sitting’) appears as *qāᶜid*, *gāᶜid*, *kāᶜid* and *ʾāᶜed*, and فوق (‘above’) as *fōq*, *fōʾ* and *fōk*. Affrication of ك appears in the Rural Palestinian material, where كل (‘all’) is transcribed *čill* beside *kull* and *kill*, and كفر, in the place name *Kafr il-Lōz*, as both *Kafr* and *Čafr*. The interdentals are kept, stopped or turned into sibilants: مثلا (‘for example’) gives *maṯalan*, *matalan* and *masalan*, and اذا (‘if’) gives *ʾiḏa* and *ʾiza*.

The vowels vary along several axes. *Imāla* raises the long *ā*: كمان (‘also’) appears as *kamān* and, in the Lebanese material, as *kamēn*, and لازم (‘must’) as *lāzim* and *lēzim*. Diphthongs either hold or collapse, so موسم (‘season’) appears as *mawsam* and *mōsam*, and صيف (‘summer’) as *ṣayf* and *ṣēf*. The long mid vowels raise, giving هون (‘here’) as *hōn* and *hūn*, and بيت (‘house’) as *bēt* and *bīt*. Short vowels are inserted or dropped, giving مثل (‘like’) as *mitl* and *mitil*.

Variation of this kind is not only between varieties. الفوز, part of the name *Abu l-Fōz*, splits between *il-Fawz* and *l-Fōz* within the Rural Palestinian material, and هيدا (‘this’) between *hayda* and *hēda* within the Lebanese.

B Experimental Details

B.1 Preprocessing and Normalization

The release already carries Unicode canonicalization at dataset construction (§2.1): the released tiers are canonically composed, with presentation forms folded, invisible controls and *tatweel* removed. We repeat the same step at scoring time, since system output is not released text. Everything below is an evaluation-time normalization, applied symmetrically to references and hypotheses. The fold is the reductive orthographic normalization of Habash (2010): the letter normalizations most Arabic NLP work applies by default. We do not fold the non-*alif* forms of *hamza*.

Both Arabic-script tiers, orthographic and diacritized, share a standard **orthographic normalization**: *alif* variants (أ, إ, آ, ٱ → ا), *alif maqṣūra* (ى → ي) and *tāʾ marbūṭa* (ة → ه). These distinctions carry no phonemic contrast in Levantine and systems spell them inconsistently. Normalizing them prevents spelling differences from counting as errors.

Diacritization. We apply this orthographic normalization, but keep all other diacritics since they are the evaluation target. Three steps are added. We strip *sukun*, since gold labels are sukun-free by construction. We canonicalize accusative *tanwin* onto the consonant preceding the seat alif, so that

كتاباً and كتابًا score as identical. And we drop the MSA case-*tanwin*, *dammatan* and *kasratan*, from both sides while keeping *fathatan*, the realized *-an*: spoken Levantine does not realize MSA case endings, so a system that supplies them is displaying MSA register rather than making a diacritization error. These endings are almost absent from the reference, 1 of 15,506 tokens. §3.1 reports the *tanwin*-included DER as an ablation.

ASR. Both tiers strip punctuation and collapse whitespace. The *neutral* tier also applies the orthographic normalization above and strips residual diacritics; the *strict* tier omits both and scores the orthography as written. Reporting the two separates recognition error from spelling convention: the difference is large only for systems trained on normalized text, and is given beside each neutral figure in Table 8.

G2P and A2P. The phonetic tier is Latin-script, so the Arabic-script normalization does not apply. The *strict* tier maps every output onto the reference alphabet (§2.2) by character rules (e.g., /ɣ/→ġ, /ʃ/→š, /aː/→ā, *ey*→ē) and strips segmentation hyphens, punctuation and casing. The *neutral* tier, the primary tier for both tasks, additionally folds notation choices that not every system supports: the definite-article vowel and sun-letter assimilation (*āl-*/*il-*), the *tāʾ marbūṭa* ending (*-eh*/*-e*, *-ah*/*-a*), the pairs *ǧ*/*j* and *č*/*k*, and the reduced vowel *ə*. Phonological contrasts (vowel length and quality, emphatics, gemination, qaf realization) are kept under both tiers. Table 7 shows raw and normalized outputs. The mapping is not exhaustive: notation the rules do not cover is scored as error under both tiers. For A2P, each per-utterance CER is capped at 100% before averaging, to bound looping or runaway audio-LLM output.

B.2 Per-Source Results

Two varieties draw on both corpora: Jordanian, 137 *Casablanca* utterances against 168 from *LangMedia* (Table 12), and Palestinian Urban, 22 against 208 (Table 13). For Palestinian Rural, Lebanese and Syrian the per-variety figures in Table 8 are the per-source figures, since each is drawn from one corpus. On Jordanian the direction of the source effect depends on the system. *LangMedia* is harder for six of the nine systems (from $+0.4$ to $+10.0$ WER); Whisper v3 and MMS-1b are better on it (-26.3 and -14.5), and GPT-Audio is level (-1.3). BYZÖ-Whisper, fine-tuned on the *Casablanca* training split, is better on *Casablanca* material in both varieties ($+7.5$ Jordanian, $+11.2$ Palestinian Urban) and overall (23.7 against 44.2 WER). Source sensitivity here tracks training exposure. The two Jordanian sources also differ in speaker gender (§5).

	Casablanca $n = 137$, 10.5 min	LangMedia $n = 168$, 15.0 min	Δ
WER (%)			
Sml-MSA	34.5	34.9	+0.4
Sml-EGY	29.0	34.1	+5.1
Whisper v3	53.8	27.4	-26.3
Whisper-Lev.	24.8	34.8	+10.0
MMS-1b	76.1	61.6	-14.5
BYZÖ-W.	30.3	37.8	+7.5
Cohere	22.9	30.5	+7.6
Gemini	19.4	22.2	+2.8
GPT Audio	29.4	28.1	-1.3
CER (%)			
Sml-MSA	15.1	17.9	+2.8
Sml-EGY	8.8	16.0	+7.2
Whisper v3	27.7	10.7	-17.0
Whisper-Lev.	7.4	12.1	+4.7
MMS-1b	29.0	23.9	-5.1
BYZÖ-W.	8.3	11.9	+3.5
Cohere	7.9	15.3	+7.4
Gemini	5.9	8.0	+2.1
GPT Audio	11.5	11.0	-0.5

Table 12: Within-variety source contrast for Jordanian, scored as in Table 8, the within-variety comparison the expanded dataset was selected to support. Δ is LangMedia − Casablanca, and is a difference of rates rather than an estimate with an interval; read it against the two utterance counts in the header. Source and speaker gender remain partly entangled (§5), so this bounds rather than isolates the source effect.

Each task's primary score is the one shown in its main table. Where a small grey number appears next to a score, it shows the difference from the alternative tier.

B.3 Per-Variety Audio Results

Per-variety WER and CER for all nine ASR systems are given in the main text (Table 8). Table 14 gives the same breakdown for the A2P comparison in §3.4: direct A2P, the ASR → G2P cascade, and the gold-transcript → G2P oracle.

The per-variety numbers add two points to the overall ones. The oracle ranges over 3.5 points across varieties (11.4 to 14.9), against 4.5 and 5.4 for the two direct A2P systems, so most of the per-variety spread comes from the audio stage and not from G2P. And the cascade does not beat direct A2P everywhere: on Palestinian/Rural Gemini's cascade is worse than its direct A2P (22.4 against 21.0), and on Lebanese GPT-Audio's cascade margin is nil

	Casablanca $n = 22$, 1.9 min	LangMedia $n = 208$, 23.1 min	Δ
WER (%)			
Sml-MSA	39.8	31.2	-8.6
Sml-EGY	36.5	28.4	-8.0
Whisper v3	29.3	29.6	+0.2
Whisper-Lev.	32.0	35.6	+3.7
MMS-1b	64.7	67.8	+3.2
BYZÖ-W.	28.9	40.1	+11.2
Cohere	31.6	36.2	+4.6
Gemini	29.3	26.3	-3.1
GPT Audio	39.5	35.6	-3.9
CER (%)			
Sml-MSA	10.8	13.5	+2.7
Sml-EGY	10.6	10.8	+0.1
Whisper v3	8.2	11.1	+2.9
Whisper-Lev.	8.6	11.6	+3.0
MMS-1b	23.6	23.8	+0.3
BYZÖ-W.	6.3	11.8	+5.5
Cohere	10.3	19.9	+9.6
Gemini	8.2	9.8	+1.6
GPT Audio	13.8	16.3	+2.5

Table 13: Within-variety source contrast for Pal./Urban, scored as in Table 8, small on the Casablanca side and therefore indicative rather than conclusive. Δ is LangMedia − Casablanca, and is a difference of rates rather than an estimate with an interval; read it against the two utterance counts in the header. Source and speaker gender remain partly entangled (§5), so this bounds rather than isolates the source effect.

(23.1 against 23.2). Palestinian/Rural is also the oracle's hardest variety, so it is where the shared G2P stage is weakest.

B.4 Models

Table 15 lists the diacritization models evaluated in our experiments with links and configuration details, Table 16 the G2P models, and Table 17 the ASR, and Table 18 the A2P models.

B.5 LLM Generation Settings

For reproducibility, Table 19 reports the decoding settings used for every prompted LLM. All prompted LLMs are accessed through their official APIs at the identifiers listed in Table 15 and Table 18, except for JAIS which is served locally with 4-bit quantization.

For the text-input tasks (diacritization and G2P) we request a JSON output and parse it with fallbacks. For A2P we request a single line of transliteration directly from the waveform (`gpt-audio` and `gemini-3.5-flash` accept audio natively). We cap the maximum number of output tokens to bound the rare runaway/looping generation.

B.6 Example Outputs

Table 20, Table 21, Table 22, and Table 23 show full-utterance diacritization, G2P, ASR and A2P outputs for one example per variety, alongside the gold reference.

C Generative AI Disclosure

Generative AI tools were used for polishing manuscript wording and coding assistance.

Audio	Path	All	Pal./Urban	Pal./Rural	Jordanian	Lebanese	Syrian
Gemini 3.5 Flash	direct A2P	20.5	21.1	**21.0**	18.0	20.8	22.4
	ASR → G2P cascade	**18.9**	**18.7**	22.4	**16.3**	**18.6**	**19.3**
GPT-Audio	direct A2P	24.9	27.2	28.6	23.4	23.2	23.4
	ASR → G2P cascade	23.3	24.6	27.5	21.0	23.1	20.7
Gold transcript*	G2P oracle	13.0	11.4	14.9	12.1	13.9	12.6

Table 14: Per-variety CER (%, ↓) for direct A2P against an ASR → G2P cascade and a gold-transcript → G2P oracle, on all 1300 utterances. The G2P stage is Gemini 3.5 Flash in every cascade row, so the GPT-Audio cascade is a hybrid and its margin over GPT-Audio's direct A2P is an upper bound. Scored with the G2P *neutral* normalization and the 100% per-utterance cap (§B.1); figures are macro means over utterances. **Bold** = best per column among the four audio systems. * Cascade: gold text is not an audio system. It passes the gold orthographic transcription through the same G2P module, and is the floor the two cascades are measured against rather than a system competing with them.

Model	Config	Source
Shakkala	char-BiLSTM, zero-shot	Barqawi (2017)
Mishkal	rule-based, zero-shot	Zerrouki (2014)
Catt	Transformer, zero-shot	Alasmary et al. (2024)
Levanti*	Levantine, zero-shot	Mor-Lan (2024)
Gemini 3.5 Flash	few-shot prompt	`gemini-3.5-flash`
GPT 5.5	few-shot prompt	`gpt-5.5`
Claude Opus 4.8	few-shot prompt	`claude-opus-4-8`

Table 15: Diacritization models evaluated. Dedicated systems are run zero-shot; the three prompted LLMs use the few-shot prompt in Appendix D.1. Shakkala, Mishkal, and Catt were developed for Modern Standard Arabic, whereas Levanti is trained on Levantine data. * Levanti produced the silver diacritization labels that the 780 expansion utterances were annotated from, so its score on those is not an independent measurement.

System	Configuration
Gemini 3.5 Flash	prompted
Claude Opus 4.8	prompted
GPT-5.5	prompted
JAIS v2 8B	prompted
Levanti	—
CharsiuG2P	multilingual byT5 small

Table 16: G2P systems evaluated, with configuration and model URLs.

System	Configuration
Seamless-MSA	`tgt_lang=arb`
Seamless-Egyptian	`tgt_lang=arz`
Whisper v3	—
Whisper-Levantine	—
MMS-1B	`ara` adapter
BYZÖ-Whisper	—
Cohere Transcribe	`language=ar`
Gemini 3.5 Flash	—
GPT-Audio	—

Table 17: ASR systems evaluated, with configuration and model URLs.

Model	Config	Source
GPT-Audio	direct, prompted, audio in	`gpt-audio`
Gemini 3.5 Flash	direct, prompted, audio in	`gemini-3.5-flash`
GPT-Audio ASR → G2P	cascade, Gemini 3.5 Flash G2P	`gpt-audio`, `gemini-3.5-flash`
Gemini ASR → G2P	cascade, Gemini 3.5 Flash G2P	`gemini-3.5-flash`
Gold text → G2P*	oracle, Gemini 3.5 Flash G2P	`gemini-3.5-flash`

Table 18: Audio-to-phoneme (A2P) systems. The two direct systems are audio language models prompted to output a single-line phonetic transcription from the waveform, with the identical prompt. The cascades pass an ASR transcript through the Gemini 3.5 Flash G2P module. * The oracle passes the gold orthographic transcription through the same module and is not an audio system. Outputs are scored after the shared G2P normalization (§B.1).

Task	Model	Serving	Precision	Temp.	Max tok.	Thinking
Diac.	Gemini 3.5 Flash	API	—	0	4096	minimal
Diac.	GPT 5.5	API	—	default	4096	none
Diac.	Claude Opus 4.8	API	—	default	4096	default
G2P	Gemini 3.5 Flash	API	—	default	default	default
G2P	GPT 5.5	API	—	default	default	default
G2P	Claude Opus 4.8	API	—	default	2048	default
G2P	JAIS 2 8B Chat	local	4-bit NF4 (DQ)	0	512	—
A2P	GPT-Audio	API	—	default	1024	—
A2P	Gemini 3.5 Flash	API	—	default	1024	disabled

Table 19: Decoding settings for the prompted LLMs.

Utterance	Diacritization: reference vs. model output
6539_1 Jordanian	Ref: وَلّ عَلَي وعَلَى بَختِي مَا فِي نَاس تِهتَمّ فِيي وَلَا بِمُستَقبَلِي عَشَان أَنَا يَتِيم أَنَا عَارِف Gemini 3.5 Flash: وِل عَلَيّ وِعَلَى بَختِي مَا فِي نَاس تِهتَمّ فِيّي وِلَا بِمُستَقبَلِي عَشَان أَنَا يَتِيم أَنَا عَارِف Claude Opus 4.8: وِل عَلَيّ وَعَلى بَختِي ما في ناس تِهتَمّ فِيّ وَلا بِمُستَقبَلِي عَشان أَنا يَتِيم أَنا عارِف GPT-5.5: وَل عَلَيّ وَعَلَى بَختي ما فِي ناس تِهتَمّ فِيّي وَلَا بِمُستَقبَلي عَشان أَنا يَتِيم أَنا عارِف Shakkala: وَلْ عَلَيَّ وَعَلَى بُخْتِي مَا فِي نَاسِ تُهْتُّمُ فِيِي وَلَا بِمُسْتَقْبِلِي عِشَانِ أَنَا يَتِيمُ أَنَا عَارِفٌ Catt: وَلِ عَلِيٍّ وَعَلَى بُخْتِي مَا فِي نَاسِ تَهْتَمُّ فِيي وَلَا بِمُسْتَقْبِلِي عَشَّانِ أَنَا يَتِيمٌ أَنَا عَارِفٌ Levanti: وَلّ عَلَيّ وْعَلَى بَخْتِي مَا فِي نَاس تِهْتَمّ فِيِّي وَلَا بِمُسْتَقْبَلِي عَشَان أَنَا يَتِيم أَنَا عَارِف
LEB_0 Lebanese	Ref: نِحنَا هَلَّق بمَوسَم، بمَوسَم شِتَا Gemini 3.5 Flash: نِحنَا هَلّق بِمَوسِم، بِمَوسِم شِتَا Claude Opus 4.8: نِحنا هَلَق بمَوسِم، بمَوسِم شِتا GPT-5.5: نِحنا هَلّق بِمَوسِم، بِمَوسِم شِتا Shakkala: نَحْنَا هَلَقَ بِمُوسِمٍ، بِمُوسِمٍ شَتَّا Catt: نَحْنَا هَلْقَ بِمَوْسِمٍ بِمَوْسِمِ شَتَا Levanti: نِحْنَا هَلَّق بْمَوْسِم، بِمَوْسِم شِتَا
46385_1 Pal./Rural	Ref: هَاظَا بِتسَلّمُو لعَمَّك أَبُو جَبِر تَسلِيم اِليَدّ، و بِتقُلُّو وَزَارة اِلحُكم اِلمَحَلِّي قَرَّرَت تَحوِيل اِلمَشَارِيع مِن كَفر اللُّوز لِإِمّ اِلحِيطَان Gemini 3.5 Flash: هَـاظَا بِتْسَلَّمُو لَعَمَّك أَبُو جَبْر تَسلِيم اِليَد، وِ بِتْقِلّو وِزَارَة اِلحُكُم اِلمَحَلِّي قَرَّرَت تَحوِيل اِلمَشَارِيع مِن كُفُر اِللُّوز لَإِمّ اِلحِيطَان Claude Opus 4.8: هاظا بِتسَلّمو لعَمّك أَبو جَبر تَسليم اِليَد، و بِتقُلّو وِزارِة اِلحُكم اِلمَحَلّي قَرّرَت تَحويل اِلمَشارِيع مِن كَفر اِللّوز لإِم اِلحيطان GPT-5.5: هَاظَا بِتسَلِّمُو لَعَمَّك أَبُو جَبِر تَسلِيم اِلإِيد، و بِتِقلُّو وِزَارِة اِلحُكم اِلمَحَلِّي قَرَّرَت تَحوِيل اِلمَشَارِيع مِن كَفِر اِلُّوز لِإِمّ اِلحِيطَان Shakkala: هَاظَا بِتَسْلُّمُو لِعَمِّكِ أَبُو جَبْرٍ تَسْلِيمُ الْيَدِ، وَ بِتَقْلُّو وَزَارَةِ الْحُكْمِ الْمَحَلِّيِّ قُرَّرَتْ تَحْوِيلُ الْمَشَارِيعِ مِنْ كُفْرِ اللَّوْزِ لِإِمّْ الْحِيطَانِ Catt: هَاظَا بِتُسَلَّمُو لِعَمِّكَ أَبُو جَبْرٍ تَسْلِيمُ الْيَدِ وَ بِتَقْلُو وِزَارَةُ الْحُكْمِ الْمَحَلِّيِّ قَرَّرْتَ تَحْوِيلَ الْمَشَارِيعِ مِنْ كُفْرِ اللَّوْزِ لِإِمِّ الْحِيطَانِ Levanti: هَاظَا بِتْسَلَّمُو لَعَمَّك أَبُو جَبِر تَسْلِيم اليَد، و بِتْقلُّو وِزَارَة الحِكم المَحَلِّي قَرَّرِت تَحْوِيل المَشَارِيع مِن كُفِر اللَّوْز لَإِمّ الحِيطَان
46465_1 Pal./Urban	Ref: يَعنِي مَيبِي سَبعَة مَاكسِيمُوم ثمَانية عَبِين مَا النَّاس تِتجَمَّع Gemini 3.5 Flash: يَعنِي مِيبِي سَبعَة مَاكسِيمُوم ثَمَانِيَة عَبَين مَا اِلنّاس تِتجَمَّع Claude Opus 4.8: يَعني مَيبي سَبعَة ماكسيمُوم ثَمانيَة عَبين ما اِلنّاس تِتجَمَّع GPT-5.5: يَعنِي مَيبي سَبعَة ماكسِيموم تَمانِيَة عَبَين ما اِلنّاس تِتجَمَّع Shakkala: يَعْنِي مِيْبِي سَبْعَةٍ مَاكِسِيمُومُ ثَمَانِيَةَ عَبَيْنِ مَا النَّاسُ تَتَجَمَّعُ Catt: يَعْنِي مِيبِي سَبْعَةٌ مَاكْسِيمُوم ثَمَانِيَةٌ عُبَينٍ مَا النَّاسُ تَتَجَمَّعُ Levanti: يَعْنِي مِيبِي سَبْعَة مَاكْسِيمُوم ثَمَانْيِة عْبَيْن مَا النَّاس تِتْجَمَّع
SYR_0 Syrian	Ref: اِلعِيلِة السُّورِيِّة مِثل اِلعِيلَات الثَّانِية، أَو مِثل اِلعِيلِة بِأَيّ بَلَد ثَانِية، اِلأب وِالإمّ وِالأوْلَاد. بَسّ بِالحَقِيقَة الحَيَاة الإجتِمَاعِيِّة Gemini 3.5 Flash: اِلعِيلِة اِلسُّورِيَّة مِتل اِلعِيلَات اِلثَّانِية، أَو مِتل اِلعِيلِة بْأَيّ بَلَد ثَانْية، اِلأب وِالإمّ وِالأولَاد. بَسّ بِالحَقِيقَة اِلحَيَاة الإجتِمَاعِيَّة Claude Opus 4.8: اِلعِيلِة اِلسّورِيّة مِتل اِلعِيلات اِلثّانِية، أو مِتل اِلعِيلِة بْأَي بَلَد تانْية، اِلأب واِلإم واِلأولاد. بَس بِالحَقِيقة اِلحَيَاة اِلإجتِماعِية GPT-5.5: اِلعِيلِة اِلسُّورِيِّة مِتل اِلعِيلَات اِلثّانِية، أَو مِتل اِلعِيلِة بِأَيّ بَلَد ثَانِية، اِلأبّ وِالإمّ وِالأولَاد. بَسّ بِالحَقِيقَة اِلحَيَاة اِلاجتِمَاعِيِّة Shakkala: اِلْعِيْلَةُ السُّورِيَّةُ مِثْلُ اِلْعَيلَاتِ الثَّانِيَةِ، أَوْ مِثْلُ اِلْعِيلَةِ بِأَيِّ بَلَدٍ ثَانِيَةٍ، الْأَبِ وَالْإِمْ وَالْأَوْلَادُ. بِس بِالْحَقِيقَةِ الْحَيَاةُ الْإِجْتَمَاعِيَّةِ Catt: اِلْعَيْلَةُ السُّورِيَّةُ مِثْلَ اِلْعَيْلَاتِ الثَّانِيَةِ أَوْ مِثْلَ اِلْعَيْلَةِ بِأَيِّ بَلَدٍ ثَانِيَةُ الْأَبُ وَالْإِمِّ وَالْأَوْلَادُ بس بِالْحَقِيقَةِ الْحَيَاةُ الْإِجْتِمَاعِيَّةُ Levanti: العِيلِة السُّورِيَّة مِثِل العِيلَات الثَّانْية، أَو مِثِل العِيلِة بْأَيّ بَلَد ثَانْية، الأب وَالإمّ وَالأوْلَاد. بَسّ بِالحَقِيقَة الحَيَاة الإجْتِمَاعِيَّة

Table 20: Full-utterance diacritization outputs, one example per variety (raw model output vs. the gold reference).

Utterance	G2P: reference vs. model output
6539_1 Jordanian	Ref: wallā ᶜalēy w ᶜala baḫṭi, ma fi nās tihtimm fī walā b-mustaqbali, ᶜashān ana yatīm, ana ᶜārif Gemini 3.5 Flash: wal ᶜalayy w-ᶜala baḫti, mā fī nās tihtamm fiyyi wala b-mustaqbali, ᶜašān ana yatīm ana ᶜārif. Levanti: wall alayy w ala baxti ma fi nās tihtamm fiyyi wala bimustaqbali ašān ana yatīm ana āref
LEB_0 Lebanese	Ref: neḥna hallaᵓ b-mawsam, b-mawsam šiṭa Gemini 3.5 Flash: niḥna hallaᵓ bi-mōsem, bi-mōsem šita Levanti: niħna hallaq bmōsem, bimōsem šita
46385_1 Pal./Rural	Ref: haẓa bitsallmu l-ᶜammak Abu Ǧabr taslīm il-yadd, w bitkullu wazārt il-Ḥukm il-Maḥallī qarrarat taḥwīl il-mašarīᶜ min Kafr il-Lōz li-ᵓimm il-Ḥīṭān. Gemini 3.5 Flash: hāẓa btisallmo la-ᶜammak Abu Ǧabr taslīm il-īd, w btigul-lo wizāret il-ḥukum il-maḥalli garrarat taḥwīl il-mašārīᶜ min Kafr il-Lōz la-Imm il-Ḥēṭān Levanti: hāðа bitsallamu la ammak abu žaber taslīm ālyad, w bitqllu wizārah ālḥikm ālmaḥalli qarraret taḥwīl ālmašārī min kufer āllōz la imm ālḥīṭān
46465_1 Pal./Urban	Ref: yaᶜni meybi sebᶜa, maksimum ṯmāniya, ᶜabīn ma n-nās titǧammaᶜ Gemini 3.5 Flash: yaᶜni mēbi sabᶜa maksimūm tmānye ᶜabēn ma in-nās titjammaᶜ Levanti: ya ni mībi sab ah māksīmūm θamānyeh beyn ma ālnnās titžamma
SYR_0 Syrian	Ref: il-ᶜayla s-sūriyye mitl il-ᶜaylāt it-tāniye, ᵓaw mitl il-ᶜayle bi-ᵓay balad tāni, il-ᵓab w il-ᵓamm w il-ulād. bass bi-l-ḥaᵓīᵓe il-ḥayāt il-ijtimāᶜīyye Gemini 3.5 Flash: il-ᶜēle s-sūriyye mitl il-ᶜēlāt it-tānye, aw mitl il-ᶜēle b-ayy balad tānye, il-ᵓab w il-imm w il-awlād. bass bi-l-ḥaᵓīᵓa l-ḥayā l-ijtimāᶜiyye Levanti: āl īleh ālssūriyyeh miθel āl īlāt ālθθānyeh, o miθel āl īleh b ayy balad θānyeh, āl ab wāl imm wāl ōlād. bass biālḥaqīqah ālhēāh āl iždimā iyyeh

Table 21: Full-utterance G2P (phonetic transcription) outputs for the same five utterances (reference vs. Gemini 3.5 Flash and Levanti).

Utterance	ASR: reference vs. model output
6539_1 Jordanian	Ref: ول علي وعلى بختي ما في ناس تهتم فيي ولا بمستقبلي عشان أنا يتيم أنا عارف Whisper-Levantine: والله علي وعلا بقتي ما فيه ناس مهتمه فيه ولا في مستقبلي عشان انا يتيم انا عارف. Seamless-EGY: ولا علي وعلى أختي ما في ناس بيهتم بيه ولا بمستقبلي عشان أنا يتيم أنا عارف Seamless-MSA: ولا عليّ ولا على أختي، ما في ناس يهتم بي ولا بمستقبلي، عشان أنا يتيم، أنا عارف
LEB_0 Lebanese	Ref: نحنا هلق بموسم، بموسم شتا Whisper-Levantine: نحن هلق بموسم موسم شتاء. Seamless-EGY: إحنا عندنا موسم، موسم شتاء Seamless-MSA: نحن نأخذ موسم الشتاء
46385_1 Pal./Rural	Ref: هاظا بتسلمو لعمك أبو جبر تسليم اليد، و بتقلو وزارة الحكم المحلي قررت تحويل المشاريع من كفر اللوز لإم الحيطان Whisper-Levantine: هذا بتسلمه لعمك ابو جبر تسليم اليد وبتقوللله وزاره الحكم المحلي قررت تحويل المشاريع من كفر اللوز لام الحيطان. Seamless-EGY: ده بتسلمو لعميك أبو جبر تسليم اليد وبتقوله وزارة الحكم المحلي قررت تحويل المشاريع من كفر اللوز لأم الحيطة Seamless-MSA: هذا بتسلمه لعمك أبو جبر تسليم اليد وبتقوله وزارة الحكم المحلي قررت تحويل المشاريع من كفر اللوز لأم الحيطة
46465_1 Pal./Urban	Ref: يعني ميبي سبعة ماكسيموم ثمانية عبين ما الناس تتجمع Whisper-Levantine: يعني ميبيو سبعه ماكسيموم ثمانيه عبين ما الناس تتج-. Seamless-EGY: يعني مايبي 7 ماكسيموم 8 قبل ما ننسى الجيم Seamless-MSA: يعني مايبي 7 ماكسيموم 8 قبل ما ننسى الجيم
SYR_0 Syrian	Ref: العيلة السورية مثل العيلات الثانية، أو مثل العيلة بأي بلد ثانية، الأب والإم والأولاد. بس بالحقيقة الحياة الإجتماعية Whisper-Levantine: العيله السوريه مثل العيلات الثانيه ام مثل العيله باي بلد ثاني الاب والام والاولاد بس بالحقيقه الحياه الاجتماعيه. Seamless-EGY: العيلة السورية مثل العائلات التانية أو مثل العيلة بأي بلد تاني الأب والأم والولاد بس بالحقيقة الحياة الاجتماعية Seamless-MSA: العيلة السورية مثل العائلات التانية أو مثل العيلة بأي بلد تاني الأب والأم والولاد بس بالحقيقة الحياة الاجتماعية

Table 22: Full-utterance ASR outputs, one example per variety (raw model output vs. the reference transcription).

Utterance	A2P (audio-to-phoneme): reference vs. model output
6539_1 Jordanian	Ref: wallā ᶜalēy w ᶜala baḫṭi, ma fi nās tihtimm fī walā b-mustaqbali, ᶜashān ana yatīm, ana ᶜārif GPT-Audio: w-alla yᶜīn ᶜarab aḫṭī, mā fī nās mihtamm fīyya w-lā fī mustaqbalī, ʾašān anā yatīm, anā ᶜārif Gemini 3.5 Flash: w-allāhi ᶜalā-bāḫṭi mā fī nās bi-yehtam fī-yi wala b-mustaqbal-i ᶜašān ana yatīm ana ᶜārif
LEB_0 Lebanese	Ref: neḥna hallaʾ b-mawsam, b-mawsam šiṭa GPT-Audio: niḥna hallaʾ b-mawsam šitaʾ Gemini 3.5 Flash: niḥna hallaʾ mawṣim mawṣim šita
46385_1 Pal./Rural	Ref: haẓa bitsallmu l-ᶜammak Abu Ǧabr taslīm il-yadd, w bitkullu wazārt il-Ḥukm il-Maḥallī qarrarat taḥwīl il-mašarīᶜ min Kafr il-Lōz li-ʾimm il-Ḥīṭān. GPT-Audio: haḏa bitsallim-u l-ᶜamm-ak Abu Jabr taslīm il-yadd, u bitkull-u wizarit il-ḥukm il-maḥalli qarrarat taḥwīl il-mašāriᶜ min kafr il-lawz li-imm il-ḥuṣṣ Gemini 3.5 Flash: hāḏa b-tsallim-o l-ᶜamm-ak abu ǧabr taslīm il-yad, wa-b-t-qull-o, wizārat il-ḥukm il-maḥallī qarrarat taḥwīl il-mašārīᶜ min kafri l-lōz la-ʾumm il-ḥīṭān
46465_1 Pal./Urban	Ref: yaᶜni meybi sebᶜa, maksimum ṯmāniya, ᶜabīn ma n-nās titǧammaᶜ GPT-Audio: yaᶜni mā bi-sabᶜa, māksimum ṯamānya, ᶜabēnhum naṣṣ an-nās Gemini 3.5 Flash: yaᶜnī max- b-sabᶜa, māksīmum tman-ya, ᶜabal-ma n-nās tis-
SYR_0 Syrian	Ref: il-ᶜayla s-sūriyye mitl il-ᶜaylāt it-tāniye, ʾaw mitl il-ᶜayle bi-ʾay balad tāni, il-ʾab w il-ʾamm w il-ulād. bass bi-l-ḥaʾīʾe il-ḥayāt il-ijtimāᶜīyye GPT-Audio: laᶜ-il-sūriyye mitl il-ᶜaylāt it-tānye aw mitl il-ᶜayle b-ʾayy balad tāni: il-ʾab w-il-ʾimm w-il-wlād. bas bil-ḥaʾīʾa il-ḥayāt il-iǧtimāᶜiyye Gemini 3.5 Flash: l-ᶜāʾile s-sūriyye miṯl il-ᶜāʾilāt t-tānye, aw miṯl il-ᶜāʾile bi-ʾaybalad tāni, l-ʾab w-il-ʾumm w-il-ʾwlād, bas bi-l-ḥaqīqa l-ḥayāt l-iǧtimāᶜiyye

Table 23: Full-utterance A2P (audio-to-phoneme) outputs for the same five utterances (reference vs. GPT-Audio and Gemini 3.5 Flash).

D Prompts

This appendix section provides the complete prompts supplied to language models in our experiments.

D.1 Diacritization Prompt

```
You are an expert in Levantine / spoken Arabic phonetics and Arabic
diacritization.

Your task is to add Arabic diacritics (tashkeel) to the provided Arabic
transcript by inferring the most likely Levantine / spoken Arabic pronunciation.

IMPORTANT:
You will NOT receive audio.
You must infer the most natural spoken Levantine pronunciation based only on the
written Arabic transcript and its context.

The goal is to test how well an LLM can predict likely spoken Levantine
pronunciation from text, while preserving the original written Arabic letters.

INPUT:
Original transcript:
{transcript}

OUTPUT:
Return only the following JSON object:

{
  "vocalized_transcript": "...",
  "notes": "..."
}

CORE RULES:

1. Preserve the original Arabic letters
- Do not add, remove, replace, or reorder Arabic letters.
- Only add diacritics to the existing letters.
- If the written transcript is ambiguous, choose the most likely Levantine
pronunciation.
- If the text seems impossible to vocalize naturally, keep the letters unchanged
and mention the issue briefly in "notes".

2. Infer spoken Levantine pronunciation
- Diacritize according to likely spoken Levantine / dialectal Arabic
pronunciation.
- Do not diacritize according to formal Fuṣḥā unless the word or phrase would
naturally be pronounced that way.
- Do not add Fuṣḥā case endings, iʿrāb, or formal grammatical endings unless they
are likely part of the spoken phrase.
- Prefer natural Palestinian / Levantine-style pronunciation when ambiguous.

3. Use the following diacritics when appropriate
Use:
- Fatḥa: ◌َ
- Kasra: ◌ِ
- Ḍamma: ◌ُ
- Shadda: ◌ّ
- Tanwīn: ◌ً ◌ٍ ◌ٌ only if it would naturally be pronounced in the spoken form.
- Madda: ◌ٓ only where appropriate and compatible with the written form

Notes:
- Tanwīn must be placed on the consonant preceding the alif, not on the alif
itself.
Example: ارکش → اًرکُش


4. Do NOT use sukūn
- Never use sukūn: ◌ْ
```

- Even when a consonant has no vowel, leave it without sukūn.
- Do not use sukūn as a placeholder for “no vowel”.

Examples:
Wrong: اوُلَغْشِي
Right: اوُلَغشِي

Wrong: تْوَص
Right: توَص

5. Do not force a diacritic onto every consonant
- A consonant should receive a vowel mark only if a vowel is likely pronounced on it.
- If no vowel is likely pronounced, leave the consonant without a mark.
- Do not add artificial vowels just so every consonant has a diacritic.
- The absence of a diacritic is acceptable and expected in some positions.

6. Shadda
- Use shadda when a consonant is likely geminated / doubled in spoken pronunciation.
- If the doubled consonant also has a vowel, write both shadda and the vowel.
- Do not overuse shadda on letters that are already doubled in writing unless the standard vocalized form requires it.
- Avoid unnatural over-marking, especially in words like الزول / النوميل where the written ل may reflect both the article and the word/root.

Example:
اندب → انّدِب
امه → اّمُه

7. Long vowels
Represent long vowels naturally using the existing Arabic letters:
- /ā/ is usually represented by fatḥa before ا
- /ī/ is usually represented by kasra before ي
- /ē/ is usually represented by kasra before ي
- /ū/ is usually represented by ḍamma before و
- /ō/ is usually represented by ḍamma before و

Examples:
فيه → فِيه
بيت → بِيت or بَيت, depending on the most likely dialectal pronunciation
نول → نوُل
نوكي → نوُكِي

8. Definite article in spoken Levantine Arabic
- Diacritize the definite article according to likely spoken pronunciation.
- In Levantine speech, the article is often pronounced اِ / il-, not اَ / al-.
- Since sukūn is forbidden, do not mark the ل with sukūn.
- Use kasra on the initial ا when the article is likely pronounced اِ.

Examples:
البيت → اِلبَيت
الحارة → اِلحاَرَة
المقفرعة → اِلمَرعَفِة
العدو → اِلعَدُو

9. Solar-letter assimilation
For words beginning with ال followed by a solar letter:

ت ث د ذ ر ز س ش ص ض ط ظ ل ن

- Infer the likely assimilated spoken pronunciation.
- If the ل of ال is not pronounced, do not place any diacritic on ل.
- Mark the solar consonant with shadda when it is likely pronounced doubled.
- Do not use sukūn on the ل.
- Do not force the Fuṣḥā-style الْ spelling.

Examples:

الشمس → اَلشَّمس
الناس → اَلنَّاس / اَلنَّاسِ, depending on the chosen annotation style
الرجل → اَلرَّجُل
الثلاثة → اَلثَّلاثَة or اَلثَّلاثَةِ, depending on the most likely spoken form

10. Moon letters
For words beginning with ال followed by a moon letter:
- The ل is likely pronounced.
- Mark only vowels that would naturally be pronounced.
- Do not use sukūn.

Examples:
البيت → اَلبَيت
القرعمة → اَلقَرعَمَة
الحرارة → اَلحَرارَة

11. Common spoken Levantine forms
Prefer common dialectal vocalizations where relevant.

Examples:
فيه → فِيه
بدن → بَدَن
همّ / همّه → هَمّ / هَمّه
بعال → بَعَال
خاي → خَايَ
وال / والّ → وَالّ or وَ الّ, depending on spacing in the original text

12. Ambiguity
- If multiple pronunciations are plausible, choose the most natural spoken Levantine one.
- Use context to decide where possible.
- Do not mention every ambiguity in "notes".
- Use "notes" only for important uncertainty, impossible forms, or cases where the transcript itself prevents a confident vocalization.

13. Dialectal consonant variation
- Some dialects may pronounce ك as “tch/ch” in certain contexts.
- Since the Arabic letters must not be changed, keep the original ك.
- Diacritize the word according to the likely vowel pattern.
- Do not reject or alter the word solely because of this pronunciation variation.

Example:
منكم → مِنكُم

14. Output cleanliness
- Return only valid JSON.
- Do not include explanations outside the JSON.
- Do not explain every word.
- If there are no special issues, set "notes" to an empty string.

EXAMPLES:

Input:
فيه

Output:
```
{
  "vocalized_transcript": "فِيه",
  "notes": ""
}
```

Input:
بدن

Output:
```
{
  "vocalized_transcript": "بِدّان",
  "notes": ""
```

```
}

Input:
امه

Output:
{
  "vocalized_transcript": "اُمّه",
  "notes": ""
}

Input:
اولغشي

Output:
{
  "vocalized_transcript": "ياوُلَغشِ",
  "notes": ""
}

Input:
السانا

Output:
{
  "vocalized_transcript": "اِنّاس",
  "notes": ""
}

Input:
ةجالثلا

Output:
{
  "vocalized_transcript": "اِلثّالَّاجِة",
  "notes": ""
}

Input:
لابع

Output:
{
  "vocalized_transcript": "عَبالَ",
  "notes": ""
}

Input:
دخاي

Output:
{
  "vocalized_transcript": "يَآخُد",
  "notes": ""
}

Now diacritize the following transcript:

{transcript}
```

D.2 Grapheme-to-Phoneme Prompt

```
You are an expert in Levantine Arabic phonology and transliteration.

Your task is to produce a Latin-script phonemic transliteration of an
unvocalized Arabic sentence, as a Levantine speaker would naturally pronounce
it.
```

Transliteration system

Use the transliteration system described below. It builds on DIN 31635 for consonants and extends it with additional symbols for Levantine phonology (extra vowel qualities, the affricate č, and the fricative j).

Consonants

Core consonants (DIN 31635):
ب=b ت=t ث=ṯ ج=ǧ ح=ḥ خ=ḫ د=d ذ=ḏ ر=r ز=z س=s ش=š ص=ṣ ض=ḍ ط=ṭ ظ=ẓ
ع=ʿ غ=ġ ف=f ق=q ك=k ل=l م=m ن=n ه=h و=w ي=y
ء/أ/إ/ؤ/ئ = ʾ (all hamzah forms)

Levantine consonant extensions:
- **č**: Palestinian rural affrication of /k/ → [tʃ]. Use č when ك is realized as an affricate (e.g. ريك → čbīr in rural Palestinian).
- **j**: voiced postalveolar fricative [ʒ], the urban Levantine realization of ج. Use j for the fricative realization (e.g. زوج → jōz) and ǧ for the affricate [dʒ] realization (e.g. ربج → Ǧabr).

Dialectal realization notes — write the symbol that matches the actual pronunciation:
- ث: may be realized as /ṯ/, /s/, or /t/ depending on subdialect
- ذ: may be realized as /ḏ/, /d/, or /z/ depending on subdialect
- ق: may be realized as /q/, /ʾ/, /g/, or /k/ depending on subdialect

Vowels

Short vowels (5 qualities + schwa):
- a (open, from fatḥa)
- i (close front, from kasra)
- u (close back, from ḍamma)
- e (mid front — between a and i in quality)
- o (mid back — between a and u in quality)
- ə (schwa — epenthetic vowels in consonant clusters and reduced unstressed vowels, e.g. əl-balad, yikuəḥ, əmnīḥ)

Long vowels:
- ā (fatḥa + alif)
- ī (kasra + yāʾ — when NOT derived from Classical *ay*)
- ū (ḍamma + wāw — when NOT derived from Classical *aw*)
- ē (Levantine monophthongization of Classical diphthong *ay*, e.g. بيت → bēt, عين → ʿēn, فيك → kēf)
- ō (Levantine monophthongization of Classical diphthong *aw*, e.g. موي → yōm, لون → lōn)

Critical vowel distinction: In Arabic script, ē and ī both appear as kasra+ي; ō and ū both appear as ḍamma+و. You must use lexical knowledge to disambiguate: use ē/ō when the word derives from a Classical Arabic form with the diphthong *ay*/*aw*; use ī/ū when it derives from original *iy*/*uw*.

Definite article

- Base form in Levantine: **il-** (sometimes **al-**)
- **Moon letters** (non-assimilating): il- + consonant → il-bēt, il-waṭan, il-ġassāle
- **Sun letters** (assimilating): double the consonant → iš-šuġul, id-dirāse, in-nās, iṭ-ṭallāje, iṣ-ṣiḥḥa
- **After a preceding vowel**, the article often reduces to **l-** → fi l-bēt, ya ǧamāʿa l-čbīr

Gemination

Write the consonant twice: در → radd, ابدن → biddna, الشمس → iš-šams

Clitics

Attach clitics with a hyphen:

- Conjunctions: والبيت → w il-bēt
- Prepositions: بالبيت → bi-l-bēt, للدراسة → li-d-dirāse, عالأرض → ʿa-l-arḍ
- Reduced prepositions: في البلد → f-il-balad or fi l-balad
- Pronominal suffixes on verbs/prepositions: قلتلي → ult-li, مهدب → biddhum, عليها → ʿalēha

Hamzah and elision

- Write **ʾ** when hamzah is clearly pronounced (typically at the start of an utterance or after a pause): ʾana, ʾawwal, ʾiḥna
- Omit ʾ when hamzah is elided in connected speech: ana, inno, inta (not ʾana, ʾinno, ʾinta)
- Both conventions may coexist naturally depending on expected speech tempo and position

Tāʾ marbūṭa

In connected speech, tāʾ marbūṭa is realized as a short vowel without h:
- Usually **-e**: ǧamāʿe, ṯallāje, ġassāle, ḥafle
- Sometimes **-a**: ǧamāʿa (varies by word and subdialect)
- Sometimes **-i**: Fiǧli, daḫiliyyi (in nisba endings)

Do NOT write -eh or -ah for tāʾ marbūṭa.

Capitalization

Capitalize proper nouns (names of people, places, and titles): Abu Ǧabr, Kafr il-Lōz, Widād, Masʿūd

Output format

Return only a JSON object:
```json
{
  "translit": "<full sentence transliteration>",
  "notes": "<brief note on any genuinely ambiguous choices, or empty string>"
}
```

Do not explain every word. Only add notes when there is real phonological ambiguity (e.g. a consonant realized differently across subdialects, or an ī/ē distinction you are uncertain about).

Examples

{"input": "عندنا معدات كلايتها جوا، كلايتها بنسويها عنا. ثم الهمبرغر كلايتو", "translit": "ʿanna muʿaddēt killayātha juwwa, killayātha binsawwīha ʿanna. masalan il-hamburġer killayāto"}

{"input": "طيب هلا انتو لتلي بتدخلوا الموا باللغة الانكليزية بتكلم باللغة الانكليزية كمان؟", "translit": "ṭayyib, halla intu ultu-lli btaḥudu l-mawād bi-l-luġa l-ingliziyye, btiḥku bi-ṣ-ṣaff bi-l-luġa l-inglīziy kamān?"}

{"input": "هنا من سوق الحوانين للمأكولات البحرية في مدينة وهران الصينية", "translit": "hunā min Sūq Huānān li-l-maʾkulāt il-baḥriyye fī madīnet Wūhān aṣ-Ṣīniyye"}

{"input": "كيف حالك ابن عمي؟ بخير الحمدلله، كل شي تمام", "translit": "čēf ḥālak ibn ʿammi? b-ḫēr il-ḥamdu li-llāh, kull əši tamām", "notes": "č used for Palestinian rural realization of ك; ə for epenthetic vowel in əši"}

{"input": "يعني جرا انا الجديد بشتغل بالجامعة والا بالمدرسة؟", "translit": "yaʿni jārna ij-jdīd bištuġul bi-l-jāmʿa willa bi-l-madrasi?", "notes": "j for urban Levantine [ʒ] realization of ج"}

D.3 Audio-to-Phoneme (A2P) Prompt

```
You are an expert phonetician of spoken Levantine Arabic and the DIN 31635
romanization system.

An audio clip of spoken Levantine Arabic IS ATTACHED to this message. Transcribe
the SPEECH IN THE AUDIO into a Latin-script phonetic transliteration of what is
actually said — the dialectal pronunciation you hear, not a Modern Standard
Arabic
normalization.

Transliteration standard — DIN 31635 with Levantine conventions (one symbol per
phoneme): b t ṯ ǧ ḥ ḫ d ḏ r z s š ṣ ḍ ṭ ẓ ʿ ġ f q k l m n h w y; glottal stop ʾ;
use č for the [tʃ] affrication and g for [g] where heard. Short vowels a i u (e,
o
where the dialect realizes them). Long vowels ā ī ū and the Levantine
monophthongs
ē (‹ay›) and ō (‹aw›). Definite article il- (assimilated to sun letters); attach
clitics with a hyphen (w-, l-, bi-). Geminate by doubling. Keep code-switched
English/French words in their normal Latin spelling.

OUTPUT RULES (critical):
- Output ONLY the transliteration of the attached audio, on a single line.
- No JSON, no quotation marks, no labels, no commentary, no step-by-step
analysis.
- Never ask for the audio — it is already attached.
```

D.4 Automatic Speech Recognition (ASR) Prompt

```
An audio clip of spoken Levantine Arabic IS ATTACHED to this message. Transcribe
the
SPEECH IN THE AUDIO into Arabic script, writing exactly what is said in the
dialect
(the actual spoken words, not a Modern Standard Arabic paraphrase).
Conventions:
- Write dialectal Arabic orthography WITHOUT diacritics (no tashkeel).
- Keep code-switched foreign words in their normal Latin spelling.
OUTPUT RULES:
- Output ONLY the Arabic transcription, on a single line.
- No diacritics, no phonetic/Latin transliteration, no JSON, no labels, no
commentary.
- Never ask for the audio — it is already attached.
```

BAREC-TB: A Dependency Treebank and Parsing Benchmark for Arabic Children's Texts

Khalid N. Elmadani[1,2] **Mohamed Adel**[1] **Muhammed AbuOdeh**[2] **Reem Faraj**[3]
Jamila El Gizuli[1] **Hanada Taha-Thomure**[4] **Nizar Habash**[1]
[1]Computational Approaches to Modeling Language Lab, New York University Abu Dhabi
[2]Mohamed bin Zayed University of Artificial Intelligence [3]Columbia University
[4]Zai Arabic Language Research Centre, Zayed University
{khalid.nabigh,nizar.habash}@nyu.edu

Abstract

We present BAREC-TB, the first dependency treebank of Arabic children's literature and educational texts. The 156K-word corpus includes texts drawn from eight sources, including curricula, reading materials, children's stories, poems, and songs. The treebank is annotated according to the Columbia Arabic Treebank (CATiB) dependency representation. We compare BAREC-TB with existing Arabic treebanks, including the Penn Arabic Treebank and Camel Treebank, using corpus statistics, dependency-based measures of syntactic complexity, readability and vocabulary-related features. We also establish parsing benchmarks using both a trainable parser and large language models. The corpus will be made publicly available to support Arabic NLP, educational NLP, and language pedagogy research.[1]

1 Introduction

Arabic children's literature and educational texts are an important yet underrepresented domain in Arabic NLP. These texts play a central role in classroom instruction, reading practice, and language learning. They also differ from adult-oriented Arabic in ways that are directly relevant to NLP: they often contain controlled vocabulary, shorter sentences, dialogues, questions, poems, songs, and pedagogically motivated discourse structures (Taha-Thomure, 2017; Habash et al., 2025). Syntactically annotated resources for this domain can support educational NLP, readability assessment, language pedagogy, and research on literacy development by providing evidence about the linguistic constructions presented to young readers at different stages.

This gap is especially visible in Arabic treebanking and dependency parsing. While Arabic has several major treebanking efforts, existing resources largely target adult-oriented text and do not provide dedicated coverage of children's literature or educational materials (MacWhinney, 2000; Maamouri et al., 2004; Habash and Roth, 2009; Dukes and Buckwalter, 2010; Al-Ghamdi et al., 2021; Habash et al., 2022; Elshabrawy et al., 2023; Khalifa et al., 2026). As a result, parsers trained on existing Arabic treebanks may not generalize well to children's materials because of domain differences in genre, sentence style, and syntactic distribution. This also makes child-directed Arabic a useful test case for large language models (LLMs). Prior work shows that their structured linguistic analyses are sensitive to prompting, tokenization, and morphosyntactic complexity (Adel et al., 2026). A dedicated dependency treebank of Arabic children's text therefore provides both a needed resource for studying child-directed Arabic syntax and a benchmark for evaluating trainable parsers and LLM-based parsing.

In this paper, we present **BAREC-TB**, the first dependency treebank dedicated to Arabic children's literature and educational texts. BAREC-TB contains approximately 156K words drawn from eight source types, including curricula, reading materials, children's stories, poems, and songs, all selected from the Foundational subset of The Balanced Arabic Readability Evaluation Corpus (BAREC) (Elmadani et al., 2025b). The treebank follows the Columbia Arabic Treebank (CATiB) dependency representation (Habash and Roth, 2009), extending an established Arabic treebanking framework to a new child-directed and educational domain. An example of a syntactic tree is shown in Figure 1.

Our contributions are: (a) We **introduce BAREC-TB**, the first dependency treebank dedicated to Arabic children's texts. (b) We **compare BAREC-TB with existing Arabic treebanks** using corpus statistics, syntactic complexity measures, readability and vocabulary-related features. (c) We **establish parsing benchmarks** using both trainable parsers and LLMs.

[1]https://barec.camel-lab.com/

Proceedings of The Fourth Arabic Natural Language Processing Conference (ArabicNLP 2026), pages 115–129
October 28-29, 2026 ©2026 Association for Computational Linguistics

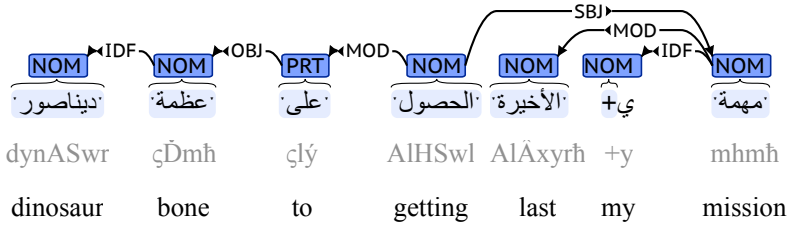


Figure 1: Example CATiB dependency tree for "*My last mission is to get a dinosaur bone*" (Majed Magazine, 2011). Words are shown with transliteration and gloss.

2 Background and Related Work

2.1 Arabic Treebanks

Arabic has several syntactically annotated resources that vary in size, domain, and annotation formalism. The Penn Arabic Treebank (PATB) is one of the most widely used Arabic treebanks (Maamouri et al., 2004). It was constructed primarily from news and web sources, and annotated using phrase structure. It has since been converted to dependency representations, including CATiB (Habash and Roth, 2009) and Universal Dependencies (UD) (Taji et al., 2017). The Prague Arabic Dependency Treebank (PADT) was the earliest Arabic dependency treebank (Smrž et al., 2002) and was later extended to UD. The Columbia Arabic Treebank (CATiB) was annotated directly in a dependency formalism inspired by traditional Arabic grammar (Habash et al., 2009), and Camel Treebank extends CATiB-style annotation to a multi-genre collection of Classical Arabic and Modern Standard Arabic texts (Habash et al., 2022). Other resources target specialized domains, such as the Quranic Arabic Corpus (Dukes and Buckwalter, 2010; Dukes et al., 2010), the *i3rab* treebank (Halabi et al., 2021), and the Arabic Poetry Treebank (ArPoT) (Al-Ghamdi et al., 2021). BAREC-10M includes children's texts among its 10 million words of synthetically generated syntactic trees (Elmadani et al., 2026). BAREC-TB addresses this limitation with manually annotated dependency trees for Arabic children's texts.

Despite this progress, existing Arabic treebanks mostly cover adult-directed material, including news, web text, religious scripture, poetry, and general-domain data. Children's texts remain largely absent from manually annotated Arabic dependency resources. BAREC-TB addresses this gap by providing the first dependency treebank dedicated to Arabic children's texts.

2.2 Children's Language Resources

Children's language resources support educational NLP, readability assessment, text simplification, and language pedagogy. Large multilingual resources such as CHILDES (MacWhinney, 2000) provide extensive child-directed speech corpora, with substantial work on syntactic parsing for English child language (Sagae et al., 2007; SAGAE et al., 2010; Liu and Prud'hommeaux, 2023; Szubert et al., 2025; Yang et al., 2025). In English, graded reading systems such as Fountas and Pinnell's 27-level scale are widely used to match texts to children's reading abilities (Fountas and Pinnell, 2006). For Arabic, Taha-Thomure (2017) proposed a 19-level leveling framework for children's books, using criteria such as genre, vocabulary, sentence structure, diacritization, illustrations, and word count.

Several Arabic resources target readability and educational materials. Early Arabic readability models relied on surface features such as word and sentence length (Al-Dawsari, 2004; Al-Khalifa and Al-Ajlan, 2010), while later work incorporated POS, morphology, orthographic features, and pretrained models (Forsyth, 2014; Saddiki et al., 2018; El-Haj and Rayson, 2016; Liberato et al., 2024). Resource-building efforts include SAMER (Al Khalil et al., 2020; Alhafni et al., 2024), CEFR-oriented Arabic datasets and vocabulary resources (Habash and Palfreyman, 2022; Naous et al., 2024; Soliman and Familiar, 2024), DARES (El-Haj et al., 2024; Almujaiwel et al., 2025), and BAREC, a fine-grained sentence-level readability corpus across 19 levels (Elmadani et al., 2025b).

2.3 Dependency Parsing for Arabic

Arabic dependency parsing has progressed from feature-rich and transition-based systems to neural models that use pretrained language representations (Green and Manning, 2010; Marton

et al., 2013; Shahrour et al., 2016; Straka, 2018; Kankanampati et al., 2020; Al-Ghamdi et al., 2023). A recent example is CAMeLParser2.0 (Elshabrawy et al., 2023), a strong BERT-based supervised parser for Arabic dependency parsing. Its compatibility with the CATiB representation makes it a suitable baseline for evaluating parsing performance on BAREC-TB.

Recent work has begun evaluating large language models (LLMs) on structured Arabic linguistic analysis. In particular, Adel et al. (2026) evaluate LLMs on Arabic morphosyntactic tagging and dependency parsing. We build on this line of work by evaluating LLMs under their strongest reported settings, while also comparing them with trainable supervised parsers.

2.4 BAREC Project

BAREC-TB is part of the broader BAREC project, which develops resources for Arabic readability across the Arab world. The project began with the 19-level BAREC Annotation Framework (Habash et al., 2025), followed by the BAREC Corpus (Elmadani et al., 2025b), a large-scale manually annotated sentence-level readability dataset. BAREC Demo (Altarbouch et al., 2025) made this work accessible through a web-based Arabic readability assessment system, while the BAREC Shared Task 2025 (Elmadani et al., 2025a) supported community benchmarking. BAREC-10M (Elmadani et al., 2026) further expanded the project with a large automatically labeled corpus. BAREC-TB extends this line of work by adding dependency parsing annotations for the child-directed portion of BAREC.

3 BAREC-TB Corpus

To ensure that the treebank focuses on children-oriented texts, we select documents from the Foundational level of the BAREC corpus (Elmadani et al., 2025b). This subset contains materials intended for early and developing readers (up to 4th grade or age 10), making it well suited for constructing a dependency treebank of Arabic children's texts. In total, BAREC-TB covers 50% of the words in the Foundational portion of BAREC. Our choice of BAREC is further motivated by its sentence-level readability annotations, which make it possible to study the relationship between syntactic structure and readability. We preserve the original BAREC train, development, and test splits for all selected documents. Table 1 shows the corpus splits in terms of documents, sentences, and words.

Split	#Docs		#Sents		#Words	
Train	193	60%	9,580	62%	96,054	62%
Dev	62	19%	3,071	20%	29,510	19%
Test	65	20%	2,732	18%	30,466	20%
BAREC-TB	320	100%	15,383	100%	156,030	100%

Table 1: Statistics for the BAREC-TB splits.

3.1 Data Sources

We select the corpus texts from eight resources (presented in order of sub-corpus word size):

Curriculum A portion of the UAE curriculum textbooks for the grades from 1 to 4 in three subjects: Arabic language, social studies, and Islamic studies (Khalil et al., 2018).

Majed A subset of 10 manually typed editions of Majed magazine for children from 1983 to 2019.[2]

Green Library 30 manually typed books from the Green Library.[3]

Hindawi A subset of 30 children stories books extracted from the Hindawi Foundation website.[4]

ArabicMMLU 878 question and answer pairs from the primary level of the ArabicMMLU benchmark dataset (Koto et al., 2024).

Collection of Children's Poems (Other) Example of the included poems: My Language Sings (لغتي تغني), and Poetry and News (أشعار وأخبار) (Al-Safadi, 2005; Taha-Thomure, 2007).

Spacetoon The opening songs of 53 animated children series from Spacetoon channel.

ChatGPT 443 sentences generated by ChatGPT and limited to children's vocabulary. All sentences were manually validated for correctness.[5]

3.2 Corpus Statistics

Table 2 presents corpus statistics for BAREC-TB and its source categories, together with PATB (Maamouri et al., 2004) and CamelTB (Habash et al., 2022).

[2] https://archive.org/details/majid_magazine
[3] https://archive.org/details/201409_201409
[4] https://www.hindawi.org/books/categories/
[5] https://chatgpt.com/

Source	Docs	Sents	Words	Tokens	Found.	Adv.	Spec.	Genre	W/S	T/S
Curriculum	38	5,361	39,113	46,166	100%	0%	0%	Education	7.3	8.6
Majed	98	3,954	38,386	46,152	100%	0%	0%	Comics	9.7	11.7
Green Library	30	1,410	23,248	29,412	100%	0%	0%	Stories	16.5	20.9
Hindawi	30	1,582	22,716	28,116	100%	0%	0%	Stories	14.4	17.8
ArabicMMLU	49	878	22,334	23,558	100%	0%	0%	Education	25.4	26.8
Other	19	886	3,890	4,664	100%	0%	0%	Poetry	4.4	5.3
Spacetoon	53	869	3,841	4,606	100%	0%	0%	Poetry	4.4	5.3
ChatGPT	3	443	2,502	2,878	100%	0%	0%	Education	5.6	6.5
BAREC-TB	320	15,383	156,030	185,552	100%	0%	0%	Children	10.1	12.1
PATB	1,834	19,738	628,656	738,889	0%	100%	0%	News	31.9	37.4
CamelTB	535	13,337	187,736	241,910	0%	56%	44%	Multi	14.1	18.1

Table 2: Corpus statistics for BAREC-TB sources, CamelTB and PATB. Found., Adv., and Spec. indicate the percentage of words in the Foundational, Advanced, and Specialized levels, respectively. W/S and T/S indicate average words and tokens per sentence, respectively.

The three treebanks differ clearly in readership level distribution: Foundational targets early readers building basic literacy, Advanced targets average adult reading, and Specialized targets complex domain-specific reading. All BAREC-TB texts belong to the Foundational level, reflecting its focus on children's text. In contrast, PATB is entirely Advanced, while Camel Treebank contains both Advanced and Specialized texts. This shows that BAREC-TB fills the child-directed portion of the Arabic treebanking landscape that is not covered by the other resources.

Table 2 also shows clear differences in sentence length. BAREC-TB has the lowest average sentence length, with 10.1 words and 12.1 tokens per sentence, compared with 31.9 words and 37.4 tokens in PATB and 14.1 words and 18.1 tokens in Camel Treebank. This is expected, since children's texts tend to use shorter sentences than adult-oriented text.

Within BAREC-TB, ArabicMMLU has the highest average sentence length, with 25.4 words and 26.8 tokens per sentence, because each sentence represents a question together with its answer options; excluding ArabicMMLU, the BAREC-TB average sentence length drops to 9.2 words and 11.2 tokens per sentence. The story sources follow, with Green Library averaging 16.5 words and 20.9 tokens per sentence and Hindawi averaging 14.4 words and 17.8 tokens per sentence. Poetry and song sources have the shortest sentences, which explains the low averages for Other Minor Sources and Spacetoon Songs, both at 4.4 words and 5.3 tokens per sentence.

4 Annotation Framework

4.1 CATiB Dependency Representation

We annotate BAREC-TB using the Columbia Arabic Treebank (CATiB) dependency representation (Habash and Roth, 2009). CATiB is well suited for Arabic treebanking because its design is influenced by traditional Arabic grammar, which makes the representation easier for annotators familiar with Arabic linguistic terminology.

Tokenization CATiB adopts the tokenization conventions used in earlier Arabic treebanking efforts, PATB and PADT (Maamouri et al., 2004). Under this scheme, all clitics are separated into independent tokens, except for the definite article +ال *Al+* 'the'.

POS Tags CATiB has six POS tags: **NOM** (non-proper nominals including nouns, pronouns, adjectives and adverbs), **PROP** (proper nouns), **VRB** (active-voice verbs), **VRB-PASS** (passive voice verbs), **PRT** (particles such as prepositions or conjunctions) and **PNX** (punctuation). Following CamelTB (Habash et al., 2022), we also use **FOREIGN** for tokens written in non-Arabic script.

Relations CATiB's dependency links are labeled with one of eight relation labels: **SBJ** (subject of verb or topic of simple nominal sentence), **OBJ** (object of verb, preposition, or deverbal noun), **TPC** (topic in complex nominal sentences containing an explicit pronominal referent), **PRD** (predicate marking the complement of the extended copular constructions), **IDF** (relation between the possessor [dependent] to the possessed [head] in the

idafa/possesive nominal construction), **TMZ** (relation of the specifier [dependent] to the specified [head] in the tamyiz/specification nominal constructions), **MOD** (general modifier of verbs or nouns), and **—** (marking flatness inside constructions such as first-last proper name sequences).

4.2 Annotation Workflow

Our annotation workflow follows the pipeline used for Camel Treebank (Habash et al., 2022): sentence segmentation, then automatic parsing, followed by manual correction of tokenization, POS tags, and dependency relations. Since BAREC-TB is selected from BAREC corpus, which is already sentence-segmented, we do not perform an additional sentence segmentation step.

Automatic Annotation We use CamelParser2.0 (Elshabrawy et al., 2023) for the initial automatic parsing. Within the parser component, we use the morphological disambiguation component of *CAMeL Tools* (Obeid et al., 2020; Inoue et al., 2022)[6] with SAMA database (Graff et al., 2009) for tokenization and POS tagging.

Manual Annotation The annotation team consists of two Arabic native speakers with extensive treebanking experience. Annotation is carried out using Palmyra3.0 (AbuOdeh et al., 2024),[7] a configurable, platform-independent tool for visualizing and editing dependency trees. It allows annotators to correct tokenization, POS tags, and dependency relations in a single interface.

Annotators' Observations Our annotation team observed that most annotation decisions in BAREC-TB were comparable to those encountered in newswire, literary, and other adult-oriented texts they previously worked on despite differences in genre and audience. In fact, syntactic annotation was sometimes simpler because children's texts tend to employ shorter sentences, more canonical word order, greater repetition, and fewer structurally ambiguous constructions, especially in sources like the Green Library and Hindawi. The main challenges arose from educational and interactive formats that weaken the correspondence between surface presentation and syntactic structure. Elements such as fill-in-the-blank exercises, matching tasks, tables, lists, multiple-choice questions, and comic-style layouts often introduce omitted constituents, unconventional sentence segmentation, and punctuation serving instructional or graphical rather than syntactic functions. This is reflected in the substantially higher frequency of punctuation in BAREC-TB and its frequent departure from the discourse- and syntax-oriented usage typical of conventional prose. Consequently, annotators could not consistently rely on punctuation, line breaks, or layout cues, and instead prioritized underlying grammatical relations when determining syntactic structure.

[6] *CAMeL Tools v1.5.5:* `BERT-Disambig`.

[7] `https://palmyra.camel-lab.com/`

Source	Docs	Sents	Tokens	UAS	LS	LAS
Curriculum	4	422	3,843	97.1	97.8	95.9
Majed	4	236	2,708	94.0	96.1	92.5
Green Library	5	269	5,195	95.4	97.3	94.2
Hindawi	4	153	2,720	97.2	98.9	96.5
ArabicMMLU	4	80	2,166	96.9	97.0	95.5
Other	1	39	557	98.4	97.3	96.6
Spacetoon	2	45	301	94.3	96.1	92.6
ChatGPT	1	152	827	99.9	99.8	99.6
Overall	**25**	**1,396**	**18,317**	**96.3**	**97.5**	**95.1**

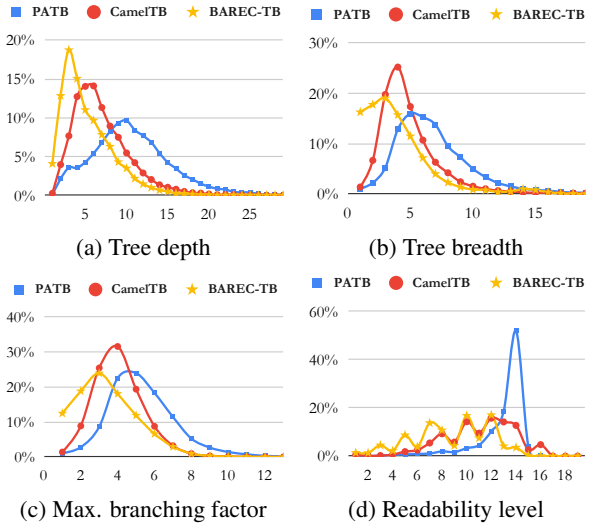
Table 3: Per-source inter-annotator agreement (IAA). The overall scores are **token-weighted averages** across all sources.

4.3 Inter-Annotator Agreement

To assess annotation consistency, we conducted two rounds of inter-annotator agreement (IAA). In the first round, two annotators independently annotated a subset of six BAREC-TB documents, then met to discuss disagreements and subsequently reviewed the annotations separately. Agreement improved from 95.6 UAS, 97.3 LS, and 94.3 LAS to 98.5 UAS, 99.2 LS, and 98.2 LAS,[8] allowing us to refine the annotation guidelines before continuing with the remaining corpus. Then, we conducted a second and final IAA evaluation on a larger set of 19 documents. The annotators achieved 96.4 UAS, 97.4 LS, and 95.2 LAS, corresponding to error reduction rates of 17.2%, 5.9%, and 16.5% over the first-round agreement for UAS, LS, and LAS, respectively.

Table 3 reports per-source inter-annotator agreement on the full IAA set. Agreement is consistently high across all sources, with LAS ranging from 92.5 to 99.6. The lowest agreement is observed for Majed and Spacetoon, while Green Library and Hindawi achieve higher agreement. These results are consistent with the annotation challenges dis-

[8] These dependency parsing metrics are explained in §6.1.



(a) Tree depth (b) Tree breadth

(c) Max. branching factor (d) Readability level

Figure 2: Distributions of tree depth, tree breadth, maximum branching factor, and readability level in PATB, CamelTB, and BAREC-TB.

cussed in § 4.2: sources with more interactive or artistic formats exhibit lower agreement than regular texts.

5 Analysis of BAREC-TB

5.1 Tree Structure and Readability

Figure 2 compares BAREC-TB sentences with PATB and CamelTB in terms of tree depth, tree breadth, maximum branching factor, and readability level. For BAREC-TB, we obtain the readability levels from BAREC Corpus, while for CamelTB and PATB we predict them using Elmadani et al.'s (2025b) best readability assessment system, the `AraBERTv02+D3Tok+Reg` model.[9]

Across the syntactic measures, BAREC-TB is concentrated at lower values, indicating shallower and less structurally dense trees. PATB shows the longest tail, especially for tree depth, while CamelTB generally falls between PATB and BAREC-TB. The readability distribution shows a similar contrast. BAREC-TB is concentrated in lower and middle readability levels, reflecting its focus on children's texts. PATB is concentrated at higher levels, consistent with its adult-oriented news domain, while CamelTB spans a broader range. Overall, these patterns show that BAREC-TB differs from existing Arabic treebanks both syntactically and in readability profile.

[9] https://huggingface.co/CAMeL-Lab/readability-arabertv2-d3tok-reg

BAREC-TB Source	CamelTB	PATB
Hindawi	98.7% (1)	98.7% (1)
Majed	97.4% (3)	98.1% (2)
Green Library	97.8% (2)	97.1% (4)
Curriculum	96.4% (4)	97.8% (3)
ChatGPT	93.8% (6)	92.9% (5)
Spacetoon	95.0% (5)	92.2% (6)
Other	92.3% (7)	91.4% (7)
ArabicMMLU	77.7% (8)	83.7% (8)
All	96.4%	97.6%

Table 4: Correlation of POS and dependency-relation distributions between each BAREC-TB source and the corresponding distributions in CamelTB and PATB. Numbers in parentheses indicate the source rank by correlation.

Subtree Structure Overlap We further compare the treebanks using local subtree structures. We define a subtree structure as a parent node with its immediate children, matching only when the parent POS, ordered child POS tags, and dependency relations are identical. Across PATB, CamelTB, and BAREC-TB, we find 21,084 unique subtree structures, but only 1,770 (8%) appear in all three treebanks. Although this shared set is small in terms of unique types, it accounts for most observed subtree tokens in PATB and CamelTB (93%) and slightly less in BAREC-TB (90%), suggesting a common core of frequent Arabic syntactic patterns alongside more domain-specific structures in children's texts.

Variation Across BAREC-TB Sources Table 4 groups BAREC-TB sources by their correlation with CamelTB and PATB **POS–relation distributions**. The highest-correlating sources are naturally structured children's texts, including stories, comics, and curricula, suggesting that their basic syntactic profiles remain close to existing Arabic treebanks. The middle group reflects more source-specific structures: ChatGPT contains short educational sentences, while Spacetoon and Other minor sources mainly contain songs and poems. ArabicMMLU is the most distinct source, since its sentences consist of questions and answer options rather than continuous text. Overall, the table shows that BAREC-TB combines sources that are broadly similar to the existing Arabic treebanks with others that introduce genre-specific structures characteristic of children's texts.

SAMER Level	BAREC-TB	CamelTB	PATB
1	24%	16%	12%
2	16%	12%	10%
3	21%	18%	15%
4	20%	22%	22%
5	10%	18%	22%
OOV	8%	14%	19%

Table 5: Distribution of lemma types across SAMER readability levels and OOV rate.

5.2 Lexical Analysis

We compare the lemma-level vocabularies of BAREC-TB, PATB, and CamelTB. All words are lemmatized in context using the best lemmatization system of Saeed and Habash (2025) with the SAMA analyzer (Graff et al., 2009), producing a shared inventory of SAMA and Out Of Vocabulary (OOV) lemmas for comparing overlap, coverage, and readability.

BAREC-TB contains 9,973 lemma types: 6,736 (68%) also occur in CamelTB, 7,214 (72%) in PATB, and 5,790 (58%) in all three treebanks. However, 1,813 BAREC-TB lemmas (18%) occur in neither CamelTB nor PATB, indicating both a substantial shared core and vocabulary specific to children's texts.

To control for corpus size, we randomly sample CamelTB and PATB to match BAREC-TB. OOV rates are low across all three treebanks: 1.0% for BAREC-TB, 1.5% for CamelTB, and 0.9% for PATB. CamelTB and PATB nevertheless cover 33% and 21% more lemma types than BAREC-TB, respectively. Lexical diversity, measured as lemma types divided by total lemmas, is lowest in BAREC-TB (6.4%), compared with CamelTB (8.5%) and PATB (7.7%), reflecting the repetition expected in children's texts.

Finally, we examine lemma readability using the SAMER readability lexicon (Al Khalil et al., 2020), which assigns SAMA lemmas to one of five readability levels. As shown in Table 5, BAREC-TB has a larger share of easier lemmas, with 40% of its lemma types in levels 1–2, compared with 28% in CamelTB and 22% in PATB. It also has fewer difficult and OOV lemmas, with 10% at level 5 and 8% OOV, compared with 18% and 14% in CamelTB, and 22% and 19% in PATB. These patterns confirm that BAREC-TB is lexically aligned with child-directed materials.

6 Parsing Benchmark

6.1 Experimental Setup

We evaluate Arabic dependency parsing on BAREC-TB using the CATiB dependency representation (Habash and Roth, 2009). In CATiB-style parsing, each token is assigned a syntactic head and a dependency relation label. Following prior works on Arabic dependency parsing, we evaluate systems under both gold-tokenized and raw-text input conditions.

Data We use BAREC-TB as the target treebank and evaluate all systems on its `Dev` and `Test` splits. Our experiments also involve using CamelTB and PATB for training. We consider three data configurations: BAREC-TB only, PATB+CamelTB, and PATB+CamelTB+BAREC-TB. These settings allow us to compare in-domain training, cross-domain transfer from existing Arabic treebanks, and mixed-domain training.

Supervised Parser For supervised dependency parsing, we follow CamelParser2.1 (Adel et al., 2026), which extends CamelParser2.0 (Elshabrawy et al., 2023) with projective multiroot decoding. Whereas CamelParser2.0 enforces exactly one root per input, CamelParser2.1 permits multiple roots while requiring at least one root and prohibiting cycles and self-loops.

The parser follows a pipeline design in which raw input is first preprocessed and morphologically disambiguated using CAMeL Tools (Obeid et al., 2020), while already tokenized input passes directly to the parser. The dependency parsing component is implemented using SuPar (Zhang et al., 2020), following the biaffine dependency parsing framework of Dozat and Manning (2017), and uses contextualized BERT representations instead of static word embeddings. Following the CamelParser2.0 setup, token-level representations are derived from WordPiece embeddings, and the parser predicts dependency arcs and labels over the tokenized sentence. In our experiments, we use CamelBERT-MSA (Inoue et al., 2021) as the pretrained encoder for all supervised parser configurations. We train CamelParser2.1 under the three data configurations described above: BAREC-TB, PATB+CamelTB, and PATB+CamelTB+BAREC-TB.

Large Language Models We also evaluate a set of strong LLMs spanning different model families

and access regimes. Our evaluation includes closed-source multilingual models, open-weight multilingual models, and Arabic-centric models: Gemini 3.5 Flash (Team et al., 2025), GPT-5.4 (Singh et al., 2026), Llama-4-Scout (Meta, 2025), and Fanar-2 (TEAM et al., 2026), henceforth **Gemini**, **GPT**, **Llama**, and **Fanar**, respectively. Each model is evaluated in a zero-shot setting and in a retrieval-based 10-shot in-context learning setting.

LLM Prompting and Retrieval For LLM-based parsing, we adopt the dependency parsing prompt formulation of Adel et al. (2026). The prompt explicitly defines the CATiB dependency labels, gives concise attachment and sentence-structure guidelines, and instructs the model to output only the required structured parse without explanations or extra fields. To ensure consistent machine-readable outputs, we use JSON-structured output interfaces for all LLM backends: the structured output modules of the OpenAI and Gemini APIs for proprietary models, and vLLM's structured decoding support for open-source models (Kwon et al., 2023). In the gold-tokenized setting, the model is instructed to preserve the provided token sequence. In the raw-text setting, the model is instructed to first tokenize the input according to CATiB conventions and then produce the dependency tree.

For the 10-shot setting, we follow the retrieval-based in-context learning approach of Adel et al. (2026). For each evaluation sentence, we retrieve examples from the corresponding training set using only the input side of the training examples, ensuring that no development or test labels are used during retrieval. We select the ten examples with the highest chrF++ similarity to the input sentence. We evaluate three retrieval pools: BAREC-TB, PATB+CamelTB, and PATB+CamelTB+BAREC-TB. Since chrF++ is based on character n-gram overlap, this retrieval strategy favors examples that are close in surface form and can capture shared orthographic, lexical, and morphological patterns, which are especially relevant for Arabic. Model licenses and details are in Appendices A and B.

Evaluation Metrics We report Labeled Attachment Score (LAS), the percentage of tokens with the correct head and dependency label, as our primary metric. We also report Unlabeled Attachment Score (UAS), the percentage of tokens assigned the correct head, and LS, the percentage of tokens assigned the correct dependency label. For raw-text experiments, we report tokenization F1 (Tok).

6.2 Results

Table 6 presents the parsing results on the development and test sets. Overall, CamelParser2.1 remains the strongest system. Its best configuration, trained on PATB+CamelTB+BAREC-TB, achieves 95.5 LAS on the test set with gold tokenization and 95.0 LAS from raw text. This setting consistently outperforms both BAREC-TB-only training and PATB+CamelTB training, showing that existing Arabic treebanks provide strong syntactic supervision while BAREC-TB adds useful in-domain adaptation.

Among LLMs, Gemini is the strongest model across both splits and input conditions. With 10 examples retrieved from BAREC-TB, it reaches 92.9 LAS on the gold-tokenized test set and 92.2 LAS in the raw-text setting, narrowing the gap with the best supervised parser to 2.6 and 2.8 LAS, respectively. GPT is the second-best LLM, while Llama and Fanar remain substantially weaker.

The comparison between zero-shot and 10-shot settings shows that retrieved demonstrations are crucial for LLM parsing. On the test set with gold tokenization, Gemini improves from 76.0 to 92.9 LAS, GPT from 57.6 to 88.9, Llama from 28.1 to 77.2, and Fanar from 15.9 to 72.0. These large gains indicate that LLM performance is highly conditioned by the availability of similar annotated examples, and should therefore be interpreted as retrieval-augmented in-context parsing.

The retrieval source also matters. For all LLMs, BAREC-TB-only retrieval gives the best or tied-best 10-shot results across development and test sets. This contrasts with the supervised parser, where all the combined training data performs best. Thus, supervised training benefits from broader data, while LLM prompting benefits most from in-domain demonstrations.

Finally, raw-text evaluation reveals differences in tokenization robustness. CamelParser shows only a small drop from gold-tokenized to raw-text parsing, and Gemini is similarly stable. In contrast, GPT, Llama, and Fanar suffer larger drops, partly reflecting weaker tokenization F1. These results highlight the importance of evaluating Arabic dependency parsers in both gold-tokenized and raw-text settings.

Variation Across Sources Table 7 shows substantial source variation under gold-tokenized evaluation: ChatGPT is the easiest source for every system, while Spacetoon is the hardest. Spacetoon

Model	Shots	Data	Dev							Test						
			Gold Tokens			Raw Text				Gold Tokens			Raw Text			
			UAS	LS	LAS	Tok	UAS	LS	LAS	UAS	LS	LAS	Tok	UAS	LS	LAS
CamelParser2.1	–	B	94.2	96.0	92.4	**99.3**	93.7	95.6	91.9	95.4	97.1	93.9	**99.3**	94.8	96.6	93.3
CamelParser2.1	–	P+C	90.7	93.9	89.4	**99.3**	90.4	93.5	89.0	91.9	94.8	90.5	**99.3**	91.4	94.4	90.0
CamelParser2.1	–	P+C+B	**95.5**	**96.9**	**94.2**	**99.3**	**95.1**	**96.6**	**93.7**	**96.8**	**97.7**	**95.5**	**99.3**	**96.3**	**97.3**	**95.0**
Gemini 3.5 Flash	0	–	81.4	83.3	73.8	92.3	72.4	76.1	66.8	82.4	85.6	76.0	95.4	74.5	78.5	69.7
Gemini 3.5 Flash	10	B	93.6	96.0	91.7	98.5	93.0	95.1	91.2	94.4	96.9	92.9	98.8	93.6	96.0	92.2
Gemini 3.5 Flash	10	P+C	87.7	92.6	86.0	98.0	87.1	91.7	85.4	88.8	94.0	87.2	97.5	87.3	92.2	85.8
Gemini 3.5 Flash	10	P+C+B	93.3	95.9	91.5	98.5	93.0	95.2	91.2	94.0	96.8	92.5	98.5	92.9	95.7	91.5
GPT-5.4	0	–	70.7	72.1	56.0	93.0	64.8	74.2	55.5	71.3	73.8	57.6	91.6	66.4	74.3	57.1
GPT-5.4	10	B	91.0	93.5	87.9	98.4	88.8	92.9	86.1	91.8	94.3	88.9	97.9	89.1	92.7	86.3
GPT-5.4	10	P+C	84.1	89.5	80.6	97.2	82.2	87.9	79.0	84.5	90.8	81.2	95.9	81.9	87.4	78.5
GPT-5.4	10	P+C+B	90.7	93.5	87.7	98.3	88.2	92.7	85.7	91.6	94.3	88.7	97.8	88.2	92.4	85.5
Llama-4-Scout	0	–	44.5	51.5	27.0	50.5	36.2	41.0	23.0	45.6	52.1	28.1	51.4	36.1	40.3	23.3
Llama-4-Scout	10	B	82.3	87.9	76.9	90.5	78.5	83.3	74.1	82.6	88.3	77.2	89.4	77.4	81.5	72.6
Llama-4-Scout	10	P+C	73.8	82.7	68.0	78.8	68.3	73.7	62.6	72.8	83.0	66.4	78.6	66.2	71.2	60.4
Llama-4-Scout	10	P+C+B	82.1	87.9	76.9	88.2	77.5	82.7	73.2	82.0	88.3	76.5	88.4	76.4	81.0	71.7
Fanar-2	0	–	34.9	34.1	14.9	43.8	26.3	31.1	14.2	35.8	34.6	15.9	46.8	26.1	30.5	14.4
Fanar-2	10	B	78.4	84.7	71.8	89.4	74.8	80.2	68.8	79.2	85.2	72.0	88.8	75.4	80.0	69.3
Fanar-2	10	P+C	67.6	78.6	60.6	78.9	62.6	71.1	56.1	68.7	79.0	60.7	79.0	62.1	69.8	55.1
Fanar-2	10	P+C+B	78.0	84.8	71.4	88.8	74.3	80.5	68.8	78.5	84.7	71.3	87.6	74.0	79.4	68.0

Table 6: **Development and test parsing results.** B, C, and P denote BAREC-TB, CamelTB, and PATB, respectively. For CamelParser2.1, Data indicates the training data. For LLMs, Data indicates the retrieval pool for 10-shot prompting. **Bold** indicates the best result in each metric column; dotted underline indicates the best result within each LLM block. Ties are marked at the displayed precision.

Source	n	CP2.1	Gemini	GPT	Llama	Fanar
Curriculum	804	**95.5**	92.9	90.1	83.6	76.7
Majed	976	**95.1**	91.4	85.9	71.7	65.8
Green Library	196	**95.9**	92.6	88.8	74.2	71.8
Hindawi	297	**94.7**	93.2	87.7	69.8	65.6
ArabicMMLU	171	**96.9**	96.2	94.4	88.7	84.5
Other	39	**93.4**	90.7	86.5	69.3	61.0
Spacetoon	97	**86.1**	85.3	76.2	62.3	50.5
ChatGPT	152	**99.6**	99.2	98.2	94.1	94.7

Table 7: **Gold-tokenized test LAS by source.** CP2.1 denotes CamelParser2.1, trained on PATB, CamelTB, and BAREC-TB; the LLM results use ten in-context examples retrieved from BAREC-TB. n denotes the number of test sentences.

consists of very short song-lyric fragments (4.3 tokens on average), providing little context for resolving attachments; 54% of its lines contain no verb, removing a central cue for clause structure; and 28% of its surface tokens are unseen in the BAREC-TB training data, compared with 5–16% for the other sources. Together, these properties create substantial syntactic and lexical difficulty for the parsers. Appendix C reports the full gold-tokenized and raw-text source-wise metrics.

7 Conclusion and Future Work

We presented BAREC-TB, the first dependency treebank for Arabic children's texts. The corpus contains approximately 156K words from curricula, reading materials, stories, comics, poems, and songs, and is annotated following the CATiB dependency representation. Our analyses show that BAREC-TB complements existing Arabic treebanks by covering lower-readability, child-directed text with shorter sentences, simpler tree structures, and a more child-oriented lexical profile. We also established parsing benchmarks with a trainable parser and LLMs, where the best performance is achieved by CamelParser trained on PATB, CamelTB, and BAREC-TB.

In future work, we plan to expand BAREC-TB by annotating more children's and educational texts from a broader range of genres and reading levels. We also aim to use the treebank for syntax-aware readability assessment and other educational NLP applications, including text simplification and automatic text leveling.

Acknowledgments

The BAREC project is supported by the Abu Dhabi Arabic Language Centre (ALC) / Department of Culture and Tourism, UAE. This research is in part with support from Google.org and the Google Cloud Research Credits program for the Gemini Academic Program. Finally, we acknowledge the support of the High Performance Computing Center at New York University Abu Dhabi.

Limitations

Although BAREC-TB provides a new resource for Arabic children's dependency parsing, it remains relatively small compared with larger Arabic treebanks. Its source coverage is also limited, and therefore may not fully represent the diversity of Arabic children's texts across age groups, curricula, and writing styles. In addition, BAREC-TB does not currently include dialectal data, although dialectal language may appear in some children's texts, especially in dialogue, songs, comics, and regionally oriented materials. We acknowledge that the parsers benchmarked in this work may produce errors, which can propagate to downstream educational NLP applications when parsing is used as a preprocessing step, including syntax-based readability assessment or essay scoring systems.

Ethics Statement

All annotators are paid fair wages for their contribution. The data will be openly available to promote transparency, reproducibility, and collaboration in Arabic language research. We used AI writing assistance within the scope of "Assistance purely with the language of the paper" described in the ACL Policy on Publication Ethics.

References

Muhammed AbuOdeh, Long Phan, Ahmed Farouk Zakaria Elshabrawy, and Nizar Habash. 2024. Palmyra 3.0: A user-friendly cloud-based platform for morphology and dependency syntax annotation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 12585–12591, Torino, Italia. ELRA and ICCL.

Mohamed Adel, Bashar Alhafni, and Nizar Habash. 2026. Arabic morphosyntactic tagging and dependency parsing with large language models. *Preprint*, arXiv:2603.16718.

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*.

Sharefah Al-Ghamdi, Hend Al-Khalifa, and Abdulmalik Al-Salman. 2021. A dependency treebank for classical Arabic poetry. In *Proceedings of the Sixth International Conference on Dependency Linguistics (Depling, SyntaxFest 2021)*, pages 1–9, Sofia, Bulgaria.

Sharefah Al-Ghamdi, Hend Al-Khalifa, and Abdulmalik Al-Salman. 2023. Fine-tuning bert-based pre-trained models for arabic dependency parsing. *Applied Sciences*, 13(7).

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.

Bayan Al-Safadi. 2005. *Al-Kashkoul: selection of poetry and prose for children* (الكشكول: مختارات من الشعر والنثر للأطفال). Al-Sa'ih Library (مكتبة السائح).

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.

Sultan Almujaiwel, Damith Premasiri, Tharindu Ranasinghe, Mo El-Haj, and Ruslan Mitkov. 2025. Complex concept-based readability estimation from arabic curriculum. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 24(11).

Kinda Altarbouch, Khalid N. Elmadani, Ossama Obeid, Hanada Taha, and Nizar Habash. 2025. BAREC demo: Resources and tools for sentence-level Arabic readability assessment. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 950–959, Suzhou, China. Association for Computational Linguistics.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. *Preprint*, arXiv:1611.01734.

Kais Dukes, Eric Atwell, and Abdul-Baquee M. Sharaf. 2010. Syntactic annotation guidelines for the

Quranic Arabic dependency treebank. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Kais Dukes and Tim Buckwalter. 2010. A Dependency Treebank of the Quran using Traditional Arabic Grammar. In *Proceedings of the Conference on Informatics and Systems (INFOS)*, Cairo, Egypt.

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: Shared Tasks*, pages 239–252, 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.

Khalid N. Elmadani, Adel Mahmoud Wizani, Hanada Taha Thomure, and Nizar Habash. 2026. A large and balanced multi-domain Arabic corpus annotated for morphology, syntax, and readability. In *Proceedings of the Fifteenth Language Resources and Evaluation Conference*, pages 11761–11775, Palma de Mallorca, Spain. ELRA Language Resource Association.

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.

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.

David Graff, Mohamed Maamouri, Basma Bouziri, Sondos Krouna, Seth Kulick, and Tim Buckwalter. 2009. Standard Arabic Morphological Analyzer (SAMA) Version 3.1. Linguistic Data Consortium LDC2009E73.

Spence Green and Christopher D. Manning. 2010. Better Arabic parsing: Baselines, evaluations, and analysis. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 394–402, Beijing, China. Coling 2010 Organizing Committee.

Nizar Habash, Muhammed AbuOdeh, Dima Taji, Reem Faraj, Jamila El Gizuli, and Omar Kallas. 2022. Camel treebank: An open multi-genre Arabic dependency treebank. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2672–2681, Marseille, France.

Nizar Habash, Reem Faraj, and Ryan Roth. 2009. Syntactic Annotation in the Columbia Arabic Treebank. In *Proceedings of the International Conference on Arabic Language Resources and Tools*, Cairo, Egypt.

Nizar Habash and 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 and Ryan Roth. 2009. CATiB: The Columbia Arabic Treebank. In *Proceedings of the Joint Conference of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 221–224, Suntec, Singapore.

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.

Dana Halabi, Ebaa Fayyoumi, and Arafat Awajan. 2021. I3rab: A new Arabic dependency treebank based on Arabic grammatical theory. *Transactions on Asian and Low-Resource Language Information Processing*, 21(2):1–32.

Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2021. The interplay of variant, size, and task type in Arabic pre-trained language models. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 92–104, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Go Inoue, Salam Khalifa, and Nizar Habash. 2022. Morphosyntactic tagging with pre-trained language models for Arabic and its dialects. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1708–1719, Dublin, Ireland. Association for Computational Linguistics.

Yash Kankanampati, Joseph Le Roux, Nadi Tomeh, Dima Taji, and Nizar Habash. 2020. Multitask easy-first dependency parsing: Exploiting complementarities of different dependency representations. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2497–2508, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Salam Khalifa, Abed Qaddoumi, Nizar Habash, and Owen Rambow. 2026. Computational benchmarks for Egyptian Arabic child directed speech. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2296–2307, Rabat, Morocco. Association for Computational Linguistics.

Muhamed Al Khalil, Hind Saddiki, Nizar Habash, and Latifa Alfalasi. 2018. A Leveled Reading Corpus of Modern Standard Arabic. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, Miyazaki, Japan.

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.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pages 611–626.

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.

Zoey Liu and Emily Prud'hommeaux. 2023. Data-driven parsing evaluation for child-parent interactions. *Transactions of the Association for Computational Linguistics*, 11:1734–1753.

Mohamed Maamouri, Ann Bies, Tim Buckwalter, and Wigdan Mekki. 2004. The Penn Arabic Treebank: Building a Large-Scale Annotated Arabic Corpus. In *Proceedings of the International Conference on Arabic Language Resources and Tools*, pages 102–109, Cairo, Egypt.

Brian MacWhinney. 2000. The childes project: Tools for analyzing talk: Volume i: Transcription format and programs, volume ii: The database. *Computational Linguistics*, 26(4):657–657.

Majed Magazine. 2011. Majed Magazine, Issue 1686. Internet Archive.

Yuval Marton, Nizar Habash, and Owen Rambow. 2013. Dependency parsing of Modern Standard Arabic with lexical and inflectional features. *Computational Linguistics*, 39(1):161–194.

Meta. 2025. The llama 4 herd: The beginning of a new era of natively multimodal ai innovation. Accessed: 2025-04-29.

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.

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 and Nizar Habash. 2025. Lemmatization as a classification task: Results from Arabic across multiple genres. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 30014–30029, Suzhou, China. Association for Computational Linguistics.

Kenji Sagae, Eric Davis, Alon Lavie, Brian MacWhinney, and Shuly Wintner. 2007. High-accuracy annotation and parsing of CHILDES transcripts. In *Proceedings of the Workshop on Cognitive Aspects of Computational Language Acquisition*, pages 25–32, Prague, Czech Republic. Association for Computational Linguistics.

KENJI SAGAE, ERIC DAVIS, ALON LAVIE, BRIAN MACWHINNEY, and SHULY WINTNER. 2010. Morphosyntactic annotation of childes transcripts. *Journal of Child Language*, 37(3):705–729.

Anas Shahrour, Salam Khalifa, Dima Taji, and Nizar Habash. 2016. CamelParser: A system for Arabic syntactic analysis and morphological disambiguation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, pages 228–232, Osaka, Japan. The COLING 2016 Organizing Committee.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, Akshay Nathan, Alan Luo, Alec Helyar, Aleksander Madry, Aleksandr Efremov, Aleksandra Spyra, Alex Baker-Whitcomb, Alex Beutel, Alex Karpenko, Alex Makelov, Alex Neitz, Alex Wei, Alexandra Barr, Alexandre Kirchmeyer, Alexey Ivanov, Alexi Christakis, Alistair Gillespie, Allison Tam, Ally Bennett, Alvin Wan, Alyssa Huang, Amy McDonald Sandjideh, Amy Yang, Ananya Kumar, Andre Saraiva, Andrea Vallone, Andrei Gheorghe, Andres Garcia Garcia, Andrew Braunstein, Andrew Liu, Andrew Schmidt, Andrey Mereskin, Andrey Mishchenko, Andy Applebaum, Andy Rogerson, Ann Rajan, Annie Wei, Anoop Kotha, Anubha Srivastava, Anushree Agrawal, Arun Vijayvergiya, Ashley Tyra, Ashvin Nair, Avi Nayak, Ben Eggers, Bessie Ji, Beth Hoover, Bill Chen, Blair Chen, Boaz Barak, Borys Minaiev, Botao Hao, Bowen Baker, Brad Lightcap, Brandon McKinzie, Brandon Wang, Brendan Quinn, Brian Fioca, Brian Hsu, Brian Yang, Brian Yu, Brian Zhang, Brittany Brenner, Callie Riggins Zetino, Cameron Raymond, Camillo Lugaresi, Carolina Paz, Cary Hudson, Cedric Whitney, Chak Li, Charles Chen, Charlotte Cole, Chelsea Voss, Chen Ding, Chen Shen, Chengdu Huang, Chris Colby, Chris Hallacy, Chris Koch, Chris Lu, Christina Kaplan, Christina Kim, CJ Minott-Henriques, Cliff Frey, Cody Yu, Coley Czarnecki, Colin Reid, Colin Wei, Cory Decareaux, Cristina Scheau, et al. 2026. Openai gpt-5 system card. *Preprint*, arXiv:2601.03267.

Otakar Smrž, Jan Šnaidauf, and Petr Zemánek. 2002. Prague Dependency Treebank for Arabic: Multi-Level Annotation of Arabic Corpus. In *Proceedings of the International Symposium on Processing of Arabic*, pages 147–155, Manouba, Tunisia.

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.

Milan Straka. 2018. UDPipe 2.0 prototype at CoNLL 2018 UD shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 197–207, Brussels, Belgium. Association for Computational Linguistics.

Ida Szubert, Omri Abend, Nathan Schneider, Samuel Gibbon, Louis Mahon, Sharon Goldwater, and Mark Steedman. 2025. Cross-linguistically consistent semantic and syntactic annotation of child-directed speech. *Language resources and evaluation*, 59(2):727—776.

Hanada Taha-Thomure. 2007. *Poems and News* (أشعار وأخبار). Educational Book House (دار الكتاب التربوي للنشر والتوزيع).

Hanada Taha-Thomure. 2017. *Arabic Language Text Leveling* (معايير هنادا طه لتصنيف مستويات النصوص العربية). Educational Book House (دار الكتاب التربوي للنشر والتوزيع).

Dima Taji, Nizar Habash, and Daniel Zeman. 2017. Universal dependencies for Arabic. In *Proceedings of the Workshop for Arabic Natural Language Processing (WANLP)*, Valencia, Spain.

FANAR TEAM, Ummar Abbas, Mohammad Shahmeer Ahmad, Minhaj Ahmad, Abdulaziz Al-Homaid, Anas Al-Nuaimi, Enes Altinisik, Ehsaneddin Asgari, Sanjay Chawla, Shammur Chowdhury, Fahim Dalvi, Kareem Darwish, Nadir Durrani, Mohamed Elfeky, Ahmed Elmagarmid, Mohamed Eltabakh, Asim Ersoy, Masoomali Fatehkia, Mohammed Qusay Hashim, Majd Hawasly, Mohamed Hefeeda, Mus'ab Husaini, Keivin Isufaj, Soon-Gyo Jung, Houssam Lachemat, Ji Kim Lucas, Abubakr Mohamed, Tasnim Mohiuddin, Basel Mousi, Hamdy Mubarak, Ahmad Musleh, Mourad Ouzzani, Amin Sadeghi, Husrev Taha Sencar, Mohammed Shinoy, Omar Sinan, and Yifan Zhang. 2026. Fanar 2.0: Arabic generative ai stack. *Preprint*, arXiv:2603.16397.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, et al. 2025. Gemini: A family of highly capable multimodal models. *Preprint*, arXiv:2312.11805.

Xiulin Yang, Zhuoxuan Ju, Lanni Bu, Zoey Liu, and Nathan Schneider. 2025. UD-English-CHILDES: A collected resource of gold and silver Universal Dependencies trees for child language interactions. In *Proceedings of the Eighth Workshop on Universal Dependencies (UDW, SyntaxFest 2025)*, pages 52–58, Ljubljana, Slovenia. Association for Computational Linguistics.

Yu Zhang, Zhenghua Li, and Min Zhang. 2020. Efficient second-order TreeCRF for neural dependency parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3295–3305, Online. Association for Computational Linguistics.

A Licenses

We list below the licenses of the data and tools used in this work, all of which are employed in accordance with their intended use.

- Penn Arabic Treebank (Maamouri et al., 2004) Part 1 v 4.1 (LDC2010T13), Part 2 v 3.1 (LDC2011T09), and Part 3 v 3.2 (LDC2010T08): LDC User Agreement for Non-Members.

- Camel Treebank (Habash et al., 2022): Non Commercial Use - NYUAD License.[10]
- SAMER Lexicon (Al Khalil et al., 2020): Non Commercial Use - NYUAD License.[11]
- BAREC Corpus (Elmadani et al., 2025b): Creative Commons Attribution Share Alike 4.0.
- BAREC-10M Corpus (Elmadani et al., 2026): Creative Commons Attribution Share Alike 4.0.
- CAMeL Tools (Obeid et al., 2020) and CAMeLBERT (Inoue et al., 2021): MIT License.
- SAMA3.1 (Graff et al., 2009): LDC Standard Arabic Morphological Analyzer (SAMA) Version 3.1 Agreement.[12]
- *AraBERTv2+D3Tok+Reg* (Elmadani et al., 2025b): MIT License.
- *CamelParser2.1* (Adel et al., 2026): MIT License.
- *Fanar-2-27B-Instruct* (TEAM et al., 2026): Apache License 2.0.[13]
- *Llama-4-Scout-17B-16E-Instruct*: Llama 4 Community License Agreement.[14]

B Model Sizes, Hyperparameters, and Computational Budget

This appendix reports model sizes, access methods, training hyperparameters, and computational budget for the parsing experiments. Supervised-parser training and decoding used one NVIDIA A100 GPU per run, while Llama-4-Scout and Fanar-2 inference used four NVIDIA A100 GPUs with tensor parallelism.. The supervised CamelParser model consists of a CamelBERT-MSA encoder and a SuPar biaffine dependency parser head, totaling approximately 110.3M parameters. Llama-4-Scout is a mixture-of-experts model with 109B total parameters and 17B active parameters, and was served using vLLM. Fanar-2 has 27B parameters and was also served using vLLM. GPT-5.4 was accessed through the OpenAI Batch API, and Gemini 3.5 Flash was accessed through the Gemini API. Since GPT-5.4 and Gemini 3.5 Flash are closed API models, their parameter counts are not publicly disclosed. We report API costs for proprietary models and runtime only for locally served models; we do not estimate institutional GPU electricity or amortized hardware cost.

Parameter	Value	Parameter	Value
bert_pooling	mean	mlp_dropout	0.33
clip	5.0	mu	0.9
encoder_dropout	0.1	n_arc_mlp	500
epochs	10	n_bert_layers	4
eps	1e-08	n_rel_mlp	100
fix_len	20	nu	0.999
lr	5e-5	update_steps	4
lr_rate	10	warmup	0.1
min_freq	2	weight_decay	0
mix_dropout	0.0		

Table 8: Hyperparameters used to train all supervised parsing models.

Model	Data	Train Time min.
CamelParser2.1	B	2.7
CamelParser2.1	P+C	10.7
CamelParser2.1	P+C+B	12.4
Total supervised training		**25.9**

Table 9: **Training time for supervised parsing models.** B, C, and P denote BAREC-TB, CamelTB, and PATB, respectively. Training time is reported in minutes.

Table 8 lists the hyperparameters used for all supervised CamelParser training runs. The same configuration is used across the three training-data settings.

Table 9 reports supervised parser training time. Training the three CamelParser variants required 25.9 minutes in total, corresponding to approximately 0.4 GPU hours on a single NVIDIA A100.

Table 10 reports evaluation runtime and API cost for all benchmark runs, broken down by split and input condition. Across all evaluation runs, the benchmark required 8,178.2 minutes, or approximately 136.3 hours, with a total API cost of $1,085.03. The monetary cost comes from proprietary API-based models; locally served models are reported with zero API cost.

C Full Source-wise Parsing Results

Table 11 provides the full metrics underlying the gold-tokenized LAS comparison in Table 7; Table 12 reports the corresponding raw-text results.

[10] http://treebank.camel-lab.com/

[11] https://camel.abudhabi.nyu.edu/samer-readability-lexicon/

[12] https://catalog.ldc.upenn.edu/LDC2010L01

[13] https://huggingface.co/QCRI/Fanar-2-27B-Instruct

[14] https://huggingface.co/meta-llama/Llama-4-Scout-17B-16E-Instruct/blob/main/LICENSE

Model	Shots	Data	Dev				Test			
			Gold Tokens		Raw Text		Gold Tokens		Raw Text	
			Time min.	Cost USD	Time min.	Cost USD	Time min.	Cost USD	Time min.	Cost USD
CamelParser2.1	–	B	0.1	0.00	2.5	0.00	0.1	0.00	2.6	0.00
CamelParser2.1	–	P+C	0.1	0.00	2.5	0.00	0.1	0.00	2.6	0.00
CamelParser2.1	–	P+C+B	0.1	0.00	2.5	0.00	0.1	0.00	2.6	0.00
Gemini 3.5 Flash	0	–	16.8	31.86	51.2	46.11	26.8	31.65	52.6	45.87
Gemini 3.5 Flash	10	B	25.0	40.00	50.8	48.74	30.1	39.73	30.5	48.33
Gemini 3.5 Flash	10	P+C	27.9	44.95	29.8	54.48	35.1	44.33	19.8	53.47
Gemini 3.5 Flash	10	P+C+B	50.3	41.08	45.2	48.94	50.1	40.21	18.1	48.73
GPT-5.4	0	–	600.9	7.67	207.8	8.88	602.9	7.69	207.7	8.73
GPT-5.4	10	B	578.6	25.81	201.8	27.17	210.0	26.44	522.1	27.50
GPT-5.4	10	P+C	589.2	31.54	573.6	32.17	605.2	31.44	204.9	31.83
GPT-5.4	10	P+C+B	202.7	26.55	219.2	27.73	586.4	26.99	213.1	28.39
Llama-4-Scout	0	–	29.1	0.00	33.1	0.00	29.4	0.00	31.8	0.00
Llama-4-Scout	10	B	41.7	0.00	45.4	0.00	45.5	0.00	49.1	0.00
Llama-4-Scout	10	P+C	54.2	0.00	57.1	0.00	58.9	0.00	60.7	0.00
Llama-4-Scout	10	P+C+B	44.9	0.00	48.8	0.00	50.7	0.00	53.4	0.00
Fanar-2	0	–	19.4	0.00	23.4	0.00	20.1	0.00	22.6	0.00
Fanar-2	10	B	29.4	0.00	35.7	0.00	32.1	0.00	41.4	0.00
Fanar-2	10	P+C	37.1	0.00	43.4	0.00	40.2	0.00	48.4	0.00
Fanar-2	10	P+C+B	31.3	0.00	39.5	0.00	35.0	0.00	43.0	0.00
All runs (sum)			2,378.8	249.47	1,713.2	294.21	2,459.0	248.50	1,627.2	292.85

Table 10: **Evaluation runtime and cost.** B, C, and P denote BAREC-TB, CamelTB, and PATB, respectively. Time is reported in minutes and cost in USD for each split and input condition. API-model costs reflect the charges at the time of the experiments; locally run models report 0.00 USD because no API charge was incurred. CamelParser2.1 used one NVIDIA A100 GPU, while Llama and Fanar used four. The final row sums all runs in each column.

Source	CamelParser2.1			Gemini			GPT			Llama			Fanar		
	UAS	LS	LAS	UAS	LS	LAS	UAS	LS	LAS	UAS	LS	LAS	UAS	LS	LAS
Curriculum	**96.4**	**97.7**	**95.5**	93.9	97.1	92.9	92.5	95.1	90.1	87.6	91.6	83.6	82.5	87.8	76.7
Majed	**96.4**	**97.6**	**95.1**	93.2	96.0	91.4	89.7	92.6	85.9	78.6	85.1	71.7	74.6	81.5	65.8
Green Library	**96.9**	**98.5**	**95.9**	93.9	97.5	92.6	91.1	95.6	88.8	78.9	89.8	74.2	78.3	87.5	71.8
Hindawi	**96.0**	97.4	**94.7**	94.7	**97.5**	93.2	90.9	94.2	87.7	76.1	84.9	69.8	73.4	82.8	65.6
ArabicMMLU	**98.2**	**98.2**	**96.9**	97.5	97.8	96.2	97.0	96.4	94.4	92.9	93.3	88.7	90.5	90.7	84.5
Other	**98.6**	**94.1**	**93.4**	95.9	93.5	90.7	90.3	92.8	86.5	76.8	84.9	69.3	69.5	81.5	61.0
Spacetoon	**92.3**	88.2	**86.1**	87.3	**90.6**	85.3	80.3	82.2	76.2	68.8	74.0	62.3	58.9	63.9	50.5
ChatGPT	**100.0**	**99.6**	**99.6**	99.8	99.4	99.2	98.4	99.0	98.2	94.4	95.4	94.1	95.8	96.5	94.7

Table 11: **Full gold-tokenized test results by source.** CamelParser2.1 is trained on PATB, CamelTB, and BAREC-TB; the LLM results use ten in-context examples retrieved from BAREC-TB. Bold indicates the best result in each row for each metric.

Source	CamelParser2.1				Gemini				GPT				Llama				Fanar			
	Tok	UAS	LS	LAS	Tok	UAS	LS	LAS	Tok	UAS	LS	LAS	Tok	UAS	LS	LAS	Tok	UAS	LS	LAS
Curriculum	99.2	**95.8**	**97.2**	**94.8**	**99.5**	94.1	97.0	93.1	99.1	91.7	95.3	89.9	92.9	85.0	89.3	81.9	91.8	82.5	86.4	78.1
Majed	**99.4**	**96.1**	**97.3**	**94.9**	99.1	92.7	95.5	90.9	98.1	87.3	91.3	83.6	88.6	74.1	78.1	67.8	87.7	70.0	74.8	61.9
Green Library	**99.2**	**96.4**	**98.1**	**95.4**	99.1	93.4	97.0	92.1	98.5	89.1	94.4	86.7	85.4	71.9	79.2	67.3	86.0	70.6	80.3	65.3
Hindawi	**98.9**	**95.1**	**96.6**	**93.7**	95.6	90.9	93.9	89.9	94.5	84.0	89.7	81.2	84.5	68.7	76.0	63.7	83.8	67.7	75.0	61.2
ArabicMMLU	**99.7**	**97.8**	**97.9**	**96.6**	99.3	96.8	97.2	95.5	98.2	93.5	93.9	90.9	91.7	84.4	84.1	80.5	92.5	87.0	86.5	82.2
Other	**100.0**	**98.4**	**94.1**	**93.2**	99.8	95.5	93.2	90.1	98.9	89.4	91.9	84.9	90.0	73.1	79.7	65.7	83.2	71.6	76.8	63.4
Spacetoon	97.5	**91.3**	86.5	**84.6**	**99.3**	85.3	**88.2**	83.7	97.0	75.7	80.0	71.4	93.0	73.6	76.0	67.5	88.1	66.3	68.3	58.2
ChatGPT	**100.0**	**100.0**	99.6	99.6	**100.0**	99.9	**99.9**	**99.8**	**100.0**	99.0	99.3	98.7	98.7	93.6	94.3	92.6	97.4	92.3	93.2	90.8

Table 12: **Full raw-text test results by source.** CamelParser2.1 is trained on PATB, CamelTB, and BAREC-TB; the LLM results use ten in-context examples retrieved from BAREC-TB. Bold indicates the best result in each row for each metric.

HalluTruthQA: A Fine-Grained Benchmark for Hallucination Detection, Localization, and Explanation in Arabic QA

Abdessalam Bouchekif [1], Mohammed-En-Nadhir Zighem [1], Salah Eddine Bekhouche [3]
Hichem Telli [2] Somaya Eltanbouly [1] Shahd Gaben [1] Heba Sbahi [1] Samer Rashwani [1]
Mutaz Al-Khatib [1] Emad Mohamed [4] Mohammed Ghaly [1] Abdenour Hadid [5]

[1] Hamad Bin Khalifa University, Qatar [2] University of Biskra, Algeria
[3] University of the Basque Country, Spain [4] Nazarbayev University, Kazakhstan
[5] Universiti Malaysia Kelantan, Malaysia

Abstract

Large language models (LLMs) can generate fluent Arabic answers, yet factual errors remain difficult to detect, localize, explain, and verify. Existing hallucination benchmarks often provide only response-level labels, offering limited support for identifying the exact erroneous content, explaining why it is incorrect, or recovering the correct factual answer. We introduce *HalluTruthQA*, a fine-grained benchmark for hallucination evaluation in Arabic question answering. HalluTruthQA contains 2,400 expert-curated examples across four knowledge-intensive domains: Islamic knowledge, history, science, and geography. Each example includes an Arabic question, a model-generated answer, a verified reference answer, a binary hallucination label, and six candidate answers for factual verification. For hallucinated answers, the benchmark additionally provides character-level erroneous spans, human-written explanations, and macro- and micro-level hallucination types.

We evaluate four open-source LLMs—ALLaM-7B, Falcon-H1R-7B, Qwen3-32B, and SILMA—in a zero-shot setting across four tasks: hallucination detection, span-level localization, factual verification, and explanation evaluation. The results show that these tasks measure complementary capabilities, with no single model achieving the best performance across all of them. The best observed scores are 0.880 Macro-F1 for hallucination detection, 0.516 F1-Sp for span localization, 0.852 LO-Score for factual verification, and 0.644 for explanation evaluation. Our findings highlight the importance of moving beyond binary hallucination detection toward fine-grained evaluation that also localizes, verifies, categorizes, and explains factual errors. The code, dataset, prompts, and evaluation scripts are available at https://gitlab.com/nlpresearcher/HalluTruthQA.

1 Introduction

Large language models have achieved strong performance in open-domain and knowledge-intensive question answering (QA). While they generate fluent and persuasive answers, their factual reliability remains a challenge. LLMs are prone to hallucinations: statements that are unsupported, fabricated, or factually incorrect despite appearing plausible (Ji et al., 2023; Huang et al., 2025b). These errors may involve wrong entities, dates, numerical values, locations, quotations, source attributions, or explanatory claims. In QA, even a short erroneous span can make an otherwise fluent answer unreliable. Most hallucination evaluation resources rely on response-level labels indicating whether an answer is correct or hallucinated. While useful, such labels provide limited insight into the nature of the error. A response may contain the correct short answer but still include a fabricated source, an incorrect quotation, a wrong date, or an unsupported explanation. In these cases, binary evaluation fails to identify where the error occurs, why it is wrong, or whether the model can select the correct answer. This motivates a fine-grained approach to hallucination evaluation that integrates answer-level detection, span-level localization, explanation, and verification.

This problem remains underexplored in Arabic. While existing Arabic resources have made progress, they do not provide comprehensive support for knowledge-intensive QA, exact character-level span annotations, human-written explanations, and candidate-based factual verification within a single benchmark. This gap is critical, as Arabic QA often involves domains where factual accuracy depends on fine-grained details and source-sensitive verification.

Islamic knowledge, in particular, represents a sensitive setting. An answer may be superficially correct while grounding its claim in an incorrect

Proceedings of The Fourth Arabic Natural Language Processing Conference (ArabicNLP 2026), pages 130–150
October 28-29, 2026 ©2026 Association for Computational Linguistics

Qur'anic reference, a fabricated hadith attribution, an invented quotation, or an unsupported legal or doctrinal claim. In this context, factual reliability depends not only on the final answer but also on the faithfulness of the supporting evidence. Similar challenges arise in history, science, and geography, where errors involve temporal reasoning, numerical values, spatial relations, and distinctions between closely related entities.

We introduce HALLUTRUTHQA, a fine-grained benchmark for hallucination evaluation in Arabic QA. The benchmark contains 2,400 expert-curated QA examples across four knowledge-intensive domains: Islamic knowledge, history, science, and geography. Each example pairs an Arabic question and a model-generated answer with a verified reference answer, a binary hallucination label, six candidate answers for factual verification, and, for hallucinated answers, character-level erroneous spans accompanied by human-written explanations. We evaluate four open-source LLMs in a zero-shot setting across the four tasks. Our results show that these tasks capture distinct challenges: models that perform well at detecting hallucinations do not necessarily localize erroneous spans, select the correct answer, or provide faithful explanations.

This work makes three contributions:

- We introduce HALLUTRUTHQA, a fine-grained Arabic QA hallucination benchmark with 2,400 expert-curated QA examples across four knowledge-intensive domains.
- We provide a multi-layer annotation scheme that includes response-level hallucination labels, character-level erroneous spans, human-written explanations, and six candidate answers for factual verification.
- We evaluate four open-source LLMs in a zero-shot setting across hallucination detection, span-level localization, factual verification, and explanation evaluation.

2 Related Work

Hallucination evaluation has become a central topic in large language model research. Hallucinations are commonly divided into factuality hallucinations, where generated claims contradict verifiable knowledge or introduce unsupported information, and faithfulness hallucinations, where the output is inconsistent with the input, context, instructions, or its own reasoning (Huang et al., 2025b). Existing detection approaches include fact verification, uncertainty estimation, self-consistency, textual entailment, QA-based consistency checking, and LLM-based evaluation. They differ in how precisely they identify hallucinations, from assigning a label to the whole response to detecting specific erroneous parts of the generated text.

Several benchmarks study hallucination across QA, summarization, retrieval-augmented generation, machine translation, and multilingual generation. TruthfulQA (Lin et al., 2022) uses adversarial questions designed to elicit common misconceptions, while HaluEval (Li et al., 2023) combines automatically generated and human-annotated examples. FELM (Chen et al., 2023) provides fine-grained factuality annotations across multiple domains, and RAGTruth (Niu et al., 2024) focuses on retrieval-augmented generation. Mu-SHROOM (Vazquez et al., 2025) formulates multilingual hallucination detection as span labeling, whereas HalluVerse25 (Abdaljalil et al., 2025) creates multilingual examples by injecting controlled errors into factual biographical sentences. HalOmi (Dale et al., 2023) studies hallucination and omission detection in multilingual machine translation, including Arabic, while Poly-FEVER (Zhang et al., 2025) addresses multilingual factual verification and hallucination detection across multiple languages, including Arabic. These resources differ in language coverage, task setting, grounding information, and annotation granularity, but few jointly support detection, exact localization, explanation, and factual verification. Arabic hallucination resources remain comparatively limited.

HALWASA (Mubarak et al., 2024) focuses on isolated LLM-generated sentences conditioned on predefined keywords rather than natural QA. Multilingual benchmarks provide partial Arabic coverage: HalluVerse25 includes Arabic in a controlled-error setting, while Mu-SHROOM provides span-level annotations for Modern Standard Arabic. Follow-up work has explored semantic-role decomposition and textual entailment for Mu-SHROOM (Elchafei and Abu-Elkheir, 2025); however, these multilingual settings are not specifically designed for knowledge-intensive Arabic QA. Other resources address Arabic generative QA more directly.

ISLAMICEVAL (Mubarak et al., 2025) evaluates hallucination detection and mitigation in Qur'anic and Hadith content, while AFTINA (Mohammed

et al., 2025) studies retrieval-based mitigation for Islamic fatwa generation. Their domain-specific scope, however, limits their coverage of broader Arabic knowledge. AraHalluEval (Alansari and Luqman, 2025) evaluates Arabic QA and summarization using 12 factuality and faithfulness indicators, including entity and numerical errors, contradictions, source conflicts, and fabrications. HalluScore (Alansari and Luqman, 2026a) provides a multi-domain Arabic QA benchmark covering adversarial formulations, reasoning, historical knowledge, and Arabic cultural contexts, together with verified evidence and reviewed answer explanations. Nevertheless, neither resource defines exact character-level localization as a general task for all hallucinated answers. In HalluScore, erroneous spans are recorded only for partially hallucinated responses.

Fine-grained localization is important because response-level labels cannot identify the precise nature or extent of an error. A response may contain the correct main answer while adding an incorrect entity, date, numerical value, quotation, attribution, or supporting claim. Similarly, two responses with the same hallucination label may differ substantially, with one being entirely incorrect and the other containing only a short unsupported span. This limitation is especially important in knowledge-intensive and source-sensitive domains. In legal and religious QA, for example, an answer may appear persuasive while misattributing a ruling, misquoting an authoritative text, or fabricating supporting evidence. Prior work shows that LLMs can misquote Qur'anic verses or generate unsupported evidence in Islamic answers (Bouchekif et al., 2025b). Evaluation should therefore assess not only overall correctness, but also source attribution, evidential grounding, and the exact location of erroneous content.

HalluTruthQA complements these resources by combining four evaluation dimensions for knowledge-intensive Arabic question answering: response-level hallucination detection, exact character-level span localization, human-written explanations, and candidate-based factual verification. Each model-generated answer is checked against a verified reference answer, allowing the benchmark to evaluate whether an answer is hallucinated, where the error occurs, why it is incorrect, and whether the correct factual answer can be selected. A discussion of hallucination causes and a feature-based comparison with related Arabic and multilingual resources are provided in Appendix A and Table 3.

Domain	Hallucinated	Non-hallucinated	Hallu. Spans
Islamic	341 (56.8%)	259 (43.2%)	393
History	193 (32.2%)	407 (67.8%)	199
Science	255 (42.5%)	345 (57.5%)	255
Geography	224 (37.3%)	376 (62.7%)	273
Total	**1013 (42.2%)**	**1387 (57.8%)**	**1120**

Table 1: Distribution of hallucinated and non-hallucinated responses across domains. "Hallucinated Spans" denotes the total number of distinct text segments annotated as factually incorrect or unsupported within the generated responses. A single response may contain multiple hallucinated spans.

3 Data Description

HalluTruthQA contains 2,400 Arabic QA examples across four knowledge-intensive domains: Islamic knowledge, history, science, and geography (Table 1). Each example includes an Arabic question, a model-generated answer, a verified reference answer, and a binary hallucination label. For hallucinated answers, we provide character-level erroneous spans and human-written explanations. These spans are further categorized using a two-level hallucination taxonomy. Macro-types capture failure categories, such as factual contradiction, context inconsistency, and factual fabrication, whereas micro-types identify the specific error affecting an annotated span, such as an incorrect entity, date, numerical value, citation, or source attribution. Each question is paired with six manually constructed candidate answers: one correct answer and five plausible distractors written in a style close to the generated output, so factual verification cannot rely on superficial stylistic cues. The benchmark supports hallucination detection, span-level localization, explanation evaluation, and multiple-choice factual verification.

We use Fanar-1-9B-Instruct[1] (Fanar Team et al., 2025) as the single response generator for all questions. This provides a controlled and reproducible setting, where all answers are produced by the same Arabic-centric instruction-tuned model. It also avoids additional variation that could come from using several generators with different styles, strengths, and error patterns. Since the evaluated answers are generated by FANAR-9B, we additionally report FANAR-9B as a self-detection refer-

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

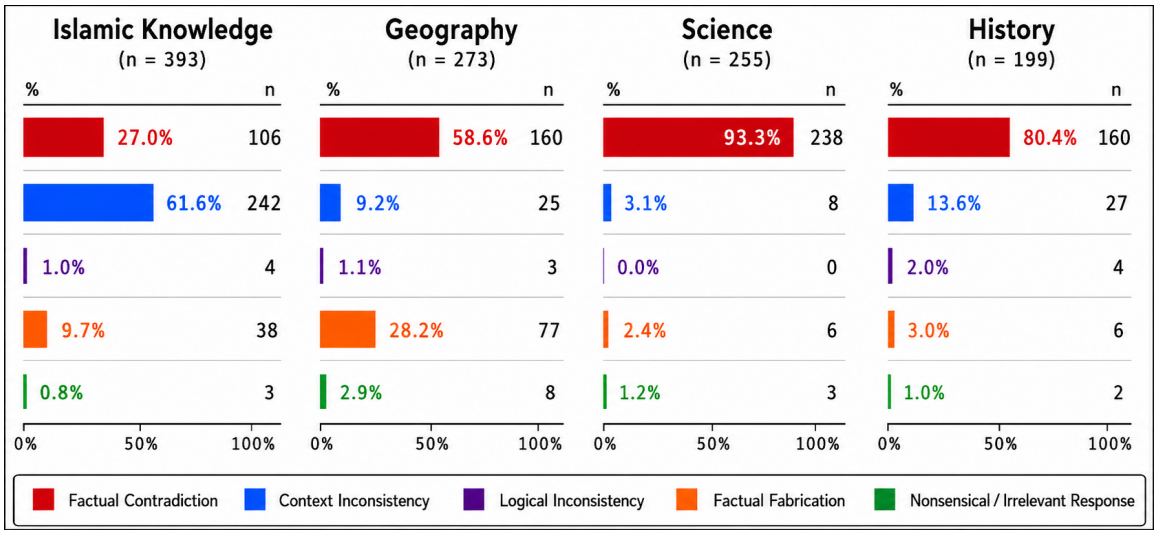


Figure 1: Distribution of hallucination macro-types across domains.

ence setting to study whether a model can detect, localize, verify, and explain hallucinations in its own outputs. These results are reported separately, rather than as evidence that FANAR-9B is superior to the other models. Answer generation used a maximum of 1024 new tokens, temperature 0.0, top-p = 1.0, batch size 4, and `bfloat16` precision on a 48 GB GPU.

3.1 Annotation Process

Annotation followed two passes: expert annotation followed by verification by trained research assistants. The four domains were handled by four experts with relevant academic backgrounds, with one expert responsible for each domain. This ensured that each subset was annotated by an expert familiar with the corresponding knowledge area and its factual verification needs. Annotators checked errors involving entities, dates, numerical values, unsupported claims, fabricated references, and incorrect attributions. For Islamic knowledge, they also verified Qur'anic references and wording, hadith attributions, and unsupported religious claims. For hallucinated responses, annotators marked character-level erroneous spans and wrote explanations. A response may contain multiple erroneous spans. Each span is annotated independently as the minimal text segment needed to isolate a single factual error. Each span is assigned one macro-type and one micro-type; multi-label assignment is therefore not used at the span level. When a response contains several distinct errors, they are annotated as separate spans.

Annotators manually created six candidate answers per question: one verified correct answer and five plausible distractors. The distractors were written in a style close to the generated response but were factually incorrect, so factual verification could not rely on superficial stylistic cues.

In the second pass, two research assistants reviewed all 2,400 examples. This pass flagged 84 examples (3.5%), which were returned to the original domain experts for adjudication. Of these, 34 were confirmed without changes, 11 questions were revised for ambiguity, and 39 examples were revised, mainly to span boundaries. Before final adjudication, expert–reviewer agreement was high across annotation dimensions, with Cohen's $\kappa = 0.93$ for binary labels, $\kappa = 0.87$ for macro-types, $\kappa = 0.81$ for micro-types, 97.9% agreement on the multiple-choice answer key, and span-level agreement of 0.89 character-F1 and 0.84 IoU. More annotation details are provided in Appendix C.

3.2 Hallucination Type Distribution

Figure 1 shows the distribution of hallucination macro-types across domains. Across all hallucinated spans, *Factual Contradiction* is the dominant category (59.3%), typically involving wrong entities, dates, locations, events, numbers, roles, or concepts. *Context Inconsistency* follows (27.0%), covering source, attribution, or evidential mismatches. *Factual Fabrication* accounts for 11.3%, while *Nonsensical / Irrelevant Response* and *Logical Inconsistency* are rare (1.4% and 1.0%). At the domain level, *Factual Contradiction* dominates in History, Science, and Geography, accounting for 80.4%, 93.3%, and 58.6% of hallucinated spans,

respectively. Islamic Knowledge differs: *Context Inconsistency* accounts for 61.6%, reflecting the need for faithful grounding in the appropriate verse, hadith, legal ruling, or source context. Geography shows the highest rate of Factual Fabrication (28.2%). Many geography questions require precise long-tail knowledge about locations and spatial relations, which may lead models to generate plausible but unsupported locations or overly specific geographic explanations. Overall, hallucinations are usually fluent and relevant, but factually incorrect, unsupported or fabricated.

These patterns highlight the need to evaluate attribution faithfulness and span-level localization beyond response-level detection. The complete hallucination taxonomy, including macro-type definitions, observed micro-types, is presented in Appendix D, with examples provided in Appendix E.

4 Experimental Setup

4.1 Evaluation Tasks

We assess model performance across four tasks:

- **Binary Hallucination Detection:** Given a question and a generated answer, the model predicts whether the response is HALLUCINATED or NON-HALLUCINATED. This task evaluates the model's ability to detect factually incorrect, fabricated, or unsupported content.
- **Span-level Localization:** For hallucinated instances, the model localizes erroneous spans at the character level. Performance is based on overlap between predicted and gold erroneous spans.
- **Explanation Evaluation:** The model generates an explanation identifying the erroneous content and justifying why it is factually incorrect. We use an LLM-as-a-judge protocol to compare generated explanations against human-written reference explanations. The judge assigns a score from 0 to 2 (0: incorrect/irrelevant; 1: partially correct; 2: complete/correct), normalized to $[0, 1]$.
- **Multiple-Choice Factual Verification:** The model selects the correct option from six candidate answers, including one correct answer and five distractors. This evaluates whether the model can identify the correct factual answer among plausible alternatives.

4.2 Models and Inference Protocol

Models. We evaluate four open-source LLMs: ALLAM-7B, FALCON-H1R-7B, QWEN3-32B, and SILMA. Since all evaluated answers are generated by FANAR-9B, we additionally report FANAR-9B as a self-detection reference setting to assess whether it can detect hallucinations in its own generated answers.

Inference Setup. All models are evaluated in a zero-shot, closed-book setting without access to external knowledge bases or web search. To ensure a fair comparison, we use identical generation parameters: temperature 0.0, top-p $= 1.0$, and a maximum of 1024 new tokens. We use a two-stage prompting strategy: the first prompt asks the model to perform binary hallucination detection; for cases predicted as HALLUCINATION, the second prompt asks for span localization, explanation, and factual verification.

4.3 Evaluation Metrics

We evaluate models on hallucination detection, span-level localization, factual verification, and explanation quality.

- **Hallucination Detection:** We report accuracy and F1-score for both hallucinated and non-hallucinated classes. We use Macro-F1 as the primary detection metric, as it equally weights both classes and is robust to label imbalance.
- **Span-level Localization:** We report partial-credit span-level F1 (*F1-Sp*) following (Sky et al., 2024). Although hallucinated spans are stored as character offsets, we compute *F1-Sp* after mapping predicted and gold spans to token sets. Recall is the average coverage of each gold span by the predicted spans, while precision is the average coverage of each predicted span by the gold spans. *F1-Sp* is the harmonic mean of these two values. Detection-label mismatches receive a span score of zero, while true-negative examples are excluded. This metric rewards partial span overlap without requiring exact boundary matches.
- **Multiple-Choice Factual Verification:** We report *Correct*, the number of correctly selected options among true-positive hallucination predictions, and *LO-Score* (Label–Option Score), which evaluates hallucination detection and option selection. For each example, LO-Score gives 1.0 for a correct NO-HALLUCINATION pre-

Domain	Model	Hallucination Detection				Span Localization			MCQ Selection		Explanation		
		Acc.	Hall. F1	No-Hall. F1	Macro-F1	Span P.	Span R.	F1-Sp	Correct	LO-Score	Err. Ident.	Fact. Corr.	Final Score
Islamic	QWEN3-32B	0.868	0.888	0.840	**0.864**	0.850	0.344	**0.489**	306/313	0.863	0.661	0.433	0.547
	FALCON-H1R-7B	0.863	0.888	0.826	0.857	0.769	0.312	0.444	312/324	0.853	0.659	0.494	0.576
	ALLAM-7B	0.862	0.876	0.844	0.860	0.839	0.248	0.383	247/293	0.823	0.668	0.530	**0.599**
	SILMA	0.737	0.713	0.757	0.735	0.802	0.102	0.181	81/196	0.641	0.243	0.119	0.181
	FANAR-9B	0.908	0.924	0.885	0.904	0.823	0.319	0.460	303/333	0.883	0.627	0.354	0.490
Geography	QWEN3-32B	0.895	0.870	0.912	**0.891**	0.635	0.409	**0.498**	207/211	0.892	0.594	0.584	**0.589**
	FALCON-H1R-7B	0.853	0.825	0.874	0.849	0.463	0.410	0.435	202/208	0.848	0.593	0.554	0.574
	ALLAM-7B	0.877	0.832	0.903	0.867	0.500	0.254	0.337	112/183	0.818	0.522	0.520	0.521
	SILMA	0.820	0.697	0.872	0.784	0.524	0.166	0.252	76/124	0.780	0.341	0.321	0.331
	FANAR-9B	0.877	0.835	0.902	0.868	0.510	0.254	0.340	175/187	0.867	0.517	0.499	0.508
Science	ALLAM-7B	0.915	0.908	0.921	0.915	0.766	0.344	**0.475**	243/253	0.907	0.761	0.723	**0.742**
	QWEN3-32B	0.778	0.793	0.762	0.777	0.570	0.388	0.462	254/254	0.778	0.688	0.485	0.587
	FALCON-H1R-7B	0.885	0.880	0.889	0.885	0.663	0.344	0.453	254/254	0.885	0.675	0.510	0.592
	SILMA	0.955	0.947	0.961	**0.954**	0.878	0.207	0.335	150/241	0.879	0.503	0.390	0.446
	FANAR-9B	0.965	0.960	0.969	0.965	0.849	0.408	0.551	253/255	0.963	0.685	0.466	0.575
History	QWEN3-32B	0.855	0.812	0.881	0.847	0.545	0.660	**0.597**	175/188	0.844	0.689	0.611	0.650
	ALLAM-7B	0.873	0.827	0.899	0.863	0.557	0.594	0.575	120/182	0.821	0.711	0.694	**0.702**
	FALCON-H1R-7B	0.836	0.794	0.864	0.829	0.513	0.634	0.567	172/189	0.822	0.687	0.623	0.655
	SILMA	0.881	0.819	0.912	**0.865**	0.484	0.362	0.415	97/161	0.828	0.263	0.246	0.255
	FANAR-9B	0.902	0.863	0.923	0.893	0.634	0.696	0.664	166/186	0.885	0.683	0.521	0.602
Global	QWEN3-32B	0.849	0.842	0.855	0.849	0.656	0.426	**0.516**	942/966	0.844	0.659	0.514	0.587
	FALCON-H1R-7B	0.859	0.853	0.866	0.859	0.618	0.400	0.486	940/975	0.852	0.654	0.536	0.595
	ALLAM-7B	0.882	0.865	0.894	**0.880**	0.688	0.333	0.449	722/911	0.842	0.673	0.614	**0.644**
	SILMA	0.848	0.799	0.878	0.838	0.699	0.188	0.296	404/722	0.782	0.351	0.272	0.312
	FANAR-9B	0.913	0.902	0.922	0.912	0.727	0.390	0.508	897/961	0.900	0.632	0.444	0.538

Table 2: Bold indicates the best performance for each metric within each domain and overall among the independent evaluation models; FANAR-9B is reported separately as a self-detection reference.

diction, 0.5 for a correct HALLUCINATION prediction with an incorrect option, 1.0 for a correct HALLUCINATION prediction with the correct option, and 0.0 for incorrect label prediction.

- **Explanation Evaluation.** We evaluate explanation quality using GPT-5.5 as an LLM judge. The judge is given the question, the model answer, the gold erroneous span, the human-written reference explanation, and the model explanation. It assigns a score reflecting how well the model justifies why the response is hallucinated. Explanation quality is computed only when both the gold label and the model prediction are hallucinated. The prompt is provided in Appendix F. To check the reliability of the scores, two annotators reviewed a random sample of 25 explanations per domain. This review did not reveal systematic disagreement between the judge scores and the human-written explanations.

Because span localization, factual verification, and explanation generation are performed only after a model predicts HALLUCINATION, these scores should be interpreted as pipeline metrics. Errors in binary hallucination detection can therefore propagate to the downstream tasks.

4.4 Results Analysis

Table 2 compares model performance across hallucination detection, span localization, multiple-choice factual verification, and explanation quality. Since all evaluated answers were generated by FANAR-9B, its self-detection result may partly reflect cues from its own writing style and the way it expresses uncertainty in generated answers. We therefore report FANAR-9B separately as a reference setting, rather than as direct evidence of general superiority over the other models.

Among the independent models, ALLAM-7B shows the strongest detection performance, achieving the highest global Macro-F1 score (0.880) with balanced performance across the two classes. FALCON-H1R-7B and QWEN3-32B obtain slightly lower Macro-F1 scores, with 0.859 and 0.849, respectively. SILMA shows a different behavior: its lower hallucination F1 compared to its non-hallucination F1 indicates that it is more conservative and misses more hallucinated responses.

Span-level localization shows that detecting a hallucinated answer is not sufficient: models must also identify the erroneous portion of the response. Under the partial-credit F1-Sp metric, QWEN3-32B achieves the best global localization performance, with an F1-Sp of 0.516, a precision of 0.656, and the highest recall of 0.426. FANAR-

9B follows closely with an F1-Sp of 0.508 and obtains the highest span precision (0.727), although its lower recall (0.390) indicates that it localizes fewer gold erroneous spans. FALCON-H1R-7B ranks third with an F1-Sp of 0.486, followed by ALLAM-7B with 0.449. These results reveal a precision–recall trade-off: although FANAR-9B and ALLAM-7B produce precise spans, they leave a substantial proportion of the annotated erroneous content uncovered. SILMA remains the weakest localization model, with an F1-Sp of 0.296 and particularly low recall (0.188), consistent with its conservative hallucination detection behavior.

The low F1-Sp scores show that span localization is substantially harder than binary hallucination detection. The difficulty is not only deciding whether an answer is hallucinated, but identifying the exact part of the Arabic response that makes it wrong. Many errors involve short factual units, such as dates, numbers, entities, verse references, or source attributions, where small boundary differences can reduce overlap. In other cases, the correct short answer is followed by hallucinated support, so models may select only the answer entity or miss the false supporting claim. Arabic also introduces boundary variation because clitics, attached particles, and surface forms can make character-level spans harder to align exactly. As a result, models may detect the hallucination correctly but produce spans that are too short, too broad, or shifted from the erroneous span.

For MCQs, FALCON-H1R-7B performs best among the independent models, with a global LO-Score of 0.852, closely followed by QWEN3-32B (0.844) and ALLAM-7B (0.842). FALCON-H1R-7B is effective at selecting the correct candidate answer once an error is identified, selecting the wrong option in only 3.6% of true-positive cases.

In terms of explanation quality, ALLAM-7B outperforms all evaluated systems, including FANAR-9B. It achieves the highest Error Identification (0.673) and Factual Correction (0.614) scores, leading to the best Final Score of 0.644. This highlights a difference between detection and explanation: while FANAR-9B excels at hallucination detection and candidate-option selection and remains competitive in span localization, ALLAM-7B provides stronger natural-language explanations with clearer factual corrections.

Domain-Specific Insights. The domain-level results show that hallucination evaluation presents different challenges across knowledge areas. In the Islamic Knowledge domain, factual correction and evidence grounding are difficult for all models, as errors often involve scriptural wording, hadith attribution, transmission-related information, or source-specific religious context. Although FANAR-9B and QWEN3-32B achieve strong detection performance, their factual-correction scores remain lower than in other domains. This suggests that detecting a hallucination does not necessarily mean that a model can recover or justify the correct religious evidence. For span localization in Islamic Knowledge, QWEN3-32B obtains the highest partial-credit F1-Sp score (0.489), followed by FANAR-9B (0.460), FALCON-H1R-7B (0.444), and ALLAM-7B (0.383). The relatively high span precision of QWEN3-32B (0.850), ALLAM-7B (0.839), and FANAR-9B (0.823), together with much lower recall, indicates that models often identify accurate erroneous spans when they make a prediction, but miss a large portion of the full annotated error. Overall, models can localize some explicit fabricated references or incorrect religious claims, but they struggle to cover all erroneous content and to produce factual corrections.

In Geography, QWEN3-32B achieves the strongest overall performance, leading in Macro-F1, F1-Sp, LO-Score, and explanation quality. Its F1-Sp score reaches 0.498, outperforming FALCON-H1R-7B (0.435), FANAR-9B (0.340), and ALLAM-7B (0.337). This suggests that QWEN3-32B is more effective at localizing errors involving geographic locations and spatial claims.

In Science, detection performance is generally high across models. Including the self-detection reference setting, FANAR-9B achieves the highest Macro-F1 score (0.965) and the highest F1-Sp score (0.551). Among the independent evaluation models, SILMA achieves the strongest detection performance with a Macro-F1 of 0.954, while ALLAM-7B obtains the best span localization score with an F1-Sp of 0.475 and the best explanation Final Score of 0.742. These results suggest that scientific hallucinations often involve erroneous values, entities, formulas, or causal claims, which can be localized more easily than implicit or context-dependent errors. However, substantial portions of the annotated erroneous spans remain uncovered.

In History, FANAR-9B achieves the best localization performance in the self-detection reference

setting, with an F1-Sp score of 0.664. QWEN3-32B obtains the highest F1-Sp score (0.597), followed by ALLAM-7B (0.575) and FALCON-H1R-7B (0.567). The higher recall of FANAR-9B (0.696) and QWEN3-32B (0.660) indicates that these models capture a larger share of historically erroneous content. At the same time, ALLAM-7B provides the strongest explanation quality, with the highest Final Score of 0.702.

Error Analysis. To better understand the types of errors, we analyze the 790 unique instances where at least one model makes a wrong prediction. Since the same instance can be misclassified by more than one model, these instances produce 1,556 model-level errors in total. We group the most frequent errors into four main families:

1. *Temporal, Numeric, and Unit Normalization (39.0%):* These errors occur when models fail to recognize that two different formats express the same fact. For example, a model may mark an answer as hallucinated because it says "40 AH", while the reference answer says "661 CE". Although the formats are different, they refer to the same historical period. This shows that models still struggle with converting dates, numbers, units, and other equivalent factual expressions (see Example Non-Hallucinated, Example 2, Example 3, and Example 17).
2. *Source Attribution and Evidence Verification (22.2%):* These errors are especially common in the Islamic domain. They occur when the model gives the correct main answer but supports it with an incorrect source, such as a fabricated verse reference, a wrong citation, or an inaccurate attribution. In these cases, models often focus on the fact that the short answer is correct, but they fail to verify whether the supporting evidence is also correct (see Example 11 and Example 15).
3. *Correct Short Answer with Hallucinated Support (17.1%):* These errors occur when the model gives the correct short answer but adds a false or unsupported explanation. For example, the named entity or final answer may match the reference answer, but the justification contains an incorrect claim. Detection models often miss these cases because they rely too much on the overlap with the correct answer and ignore the hallucinated information in the explanation (see Example 9 and Example 10).
4. *Fully Incorrect Answer (11.3%):* These errors occur when the model's main answer is factually incorrect. In some cases, the generated answer may be entirely different from, or directly contradict, the correct answer. The error affects the core answer rather than only the explanation or supporting evidence (see Example 7, Example 8, and Example 11).

Together, these four families cover 89.6% of the model-level errors. The remaining 10.4% correspond to mixed or less frequent cases that combine several patterns, such as partial factual mismatches, weak evidence checking, over-specific additions, or imprecise span localization. We do not treat these cases as a separate family because they do not form a single recurring error pattern.

Common Failure Cases. A total of 39 examples triggered an identical failure across all five evaluated systems, including FANAR-9B in the self-detection reference setting. The vast majority of these shared errors (79.5%) are false positives, where a valid response was uniformly misclassified as hallucinated. Only 20.5% correspond to missed true hallucinations. These shared failures are concentrated in History (18 cases), followed by Geography (10), Islamic (8), and Science (3).

These findings emphasize that resolving hallucinations in advanced Arabic models requires moving beyond factual recall; models must recognize semantic paraphrases, validate fine-grained source attributions, handle temporal and numerical conversions, isolate correct short answers from faulty supporting explanations, and localize the exact erroneous spans. Additional annotation and evaluation details are provided in Appendix C.

5 Conclusion

We introduced HALLUTRUTHQA, a fine-grained, expert-curated benchmark for evaluating hallucinations in Arabic QA. Spanning Islamic knowledge, history, science, and geography, HALLUTRUTHQA provides response-level hallucination labels, character-level erroneous spans, human-written explanations, and multiple-choice factual verification candidates. This multidimensional design shifts hallucination evaluation from binary correctness toward a granular understanding of where and why models fail, and whether they can recover the correct factual information. Our evaluation of four open-source instruction-tuned LLMs, together with the FANAR-9B self-detection reference setting, shows that hallucina-

tion detection, span localization, factual verification, and explanation quality remain distinct, non-trivial challenges. A key finding is that strong detection performance does not necessarily imply high-quality explanations or accurate span localization: FANAR-9B performs strongly in the self-detection reference setting, QWEN3-32B achieves the best global span localization, and ALLAM-7B provides the most reliable factual explanations. Moreover, our results reveal domain-specific patterns: Islamic Knowledge is affected by evidence-grounding and source-attribution errors, whereas history, science, and geography are mainly affected by factual contradictions and fabricated details. These findings underscore the need for Arabic hallucination benchmarks that move beyond binary detection to assess attribution faithfulness, temporal consistency, and reasoning errors. HALLUTRUTHQA provides a step toward more reliable Arabic QA systems. Future work will expand the benchmark to additional domains, broader model coverage, and training settings aimed at improving factual reliability in Arabic language models.

Limitations

This work has several limitations.

- First, HALLUTRUTHQA focuses on four knowledge-intensive domains: Islamic knowledge, history, science, and geography. Although these domains cover diverse factual phenomena, the benchmark does not capture all types of Arabic QA, such as legal, medical, financial, or conversational settings.
- Second, the model-generated answers in the dataset are produced using a single LLM, FANAR-9B. This design enables controlled annotation and analysis, but it may limit the diversity of hallucination patterns compared to datasets built from multiple generation models. Future work can extend the benchmark with answers generated by a range of Arabic and multilingual LLMs.
- Third, span-level hallucination annotation is challenging. In some cases, hallucinated content may involve implicit reasoning errors, unsupported claims, or source attribution errors that are difficult to isolate with exact character boundaries. To reduce this issue, we rely on expert validation and use overlap-based span evaluation, but boundary variation remains possible.
- Fourth, explanation quality is evaluated using an LLM-as-a-judge protocol, with additional human validation on a sample of judged explanations. While this helps make the evaluation scalable, automatic judging may still introduce bias or inconsistency. More extensive human evaluation would further strengthen the reliability of explanation assessment.
- Finally, we deliberately focus on open models and exclude proprietary API-based systems such as Gemini or Claude from the main evaluation. This choice is motivated by fairness and reproducibility. Our benchmark is designed as a closed-book evaluation, where all models are tested under the same inference conditions and without access to external search. In contrast, API-based systems may rely on changing backend versions, retrieval components, web search, or system-level grounding that are not always fully controllable by researchers. Including such systems could make it harder to ensure that the evaluation measures the model itself rather than external information access. This design also supports the analysis and improvement of open Arabic language models, which can be inspected, reproduced, and adapted by the research community.

References

Samir Abdaljalil, Hasan Kurban, and Erchin Serpedin. 2025. Halluverse25: Fine-grained multilingual benchmark dataset for llm hallucinations. *arXiv preprint arXiv:2503.07833*.

Samir Abdaljalil, Parichit Sharma, Erchin Serpedin, and Hasan Kurban. 2026. Halluverse-m^ 3: A multitask multilingual benchmark for hallucination in llms. *arXiv preprint arXiv:2602.06920*.

Aisha Alansari and Hamzah Luqman. 2025. AraHalluEval: A fine-grained hallucination evaluation framework for Arabic LLMs. In *Proceedings of The Third Arabic Natural Language Processing Conference*, pages 148–161, Suzhou, China. Association for Computational Linguistics.

Aisha Alansari and Hamzah Luqman. 2026a. Halluscore: Large language model hallucination question answering benchmark. *arXiv preprint arXiv:2605.17007*.

Aisha Alansari and Hamzah Luqman. 2026b.

Large language models hallucination: A comprehensive survey. *arXiv preprint arXiv:2510.06265*.

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: Shared Tasks*, pages 774–789.

Abdessalam Bouchekif, Somaya Eltanbouly, Samer Rashwani, Shahd Gaben, Mutaz Al-Khatib, Heba Sbahi, Emad Mohamed, and Mohammed Ghaly. 2026a. Qias 2026: Overview of the shared task on islamic inheritance reasoning. *arXiv preprint arXiv:2606.13756*.

Abdessalam Bouchekif, Shahd Gaben, Samer Rashwani, Somaya Eltanbouly, Mutaz Al-Khatib, Heba Sbahi, Mohammed Ghaly, and Emad Mohamed. 2026b. Mawarith: A dataset and benchmark for legal inheritance reasoning with llms. *arXiv preprint arXiv:2603.07539*.

Abdessalam Bouchekif, Samer Rashwani, Emad Soliman Ali Mohamed, Mutaz Alkhatib, Heba Sbahi, Shahd Gaben, Wajdi Zaghouani, Aiman Erbad, and Mohammed Ghaly. 2025a. QIAS 2025: Overview of the shared task on islamic inheritance reasoning and knowledge assessment. In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 851–860, Suzhou, China. Association for Computational Linguistics.

Abdessalam 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*, pages 246–257, Suzhou, China. Association for Computational Linguistics.

Shiqi Chen, Yiran Zhao, Jinghan Zhang, I-Chun Chern, Siyang Gao, Pengfei Liu, and Junxian He. 2023. FELM: Benchmarking factuality evaluation of large language models. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

David Dale, Elena Voita, Janice Lam, Prangthip Hansanti, Christophe Ropers, Elahe Kalbassi, Cynthia Gao, Loïc Barrault, and Marta R. Costa-jussà. 2023. HalOmi: A manually annotated benchmark for multilingual hallucination and omission detection in machine translation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 638–653, Singapore. Association for Computational Linguistics.

Passant Elchafei and Mervat Abu-Elkheir. 2025. Hallucination detectives at SemEval-2025 task 3: Span-level hallucination detection for LLM-generated answers. In *Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025)*, pages 601–606, Vienna, Austria. Association for Computational Linguistics.

Somaya Eltanbouly, Heba Sbahi, Samer Rashwani, Abdessalam Bouchekif, Mutaz Al-Khatib, Shahd Gaben, and Mohammed Ghaly. 2026. What makes a good fiqh retriever? answer retrieval for arabic islamic jurisprudence. *arXiv preprint arXiv:2608.20246*.

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.

Shahd Gaben, Heba Sbahi, Samer Rashwani, Abdessalam Bouchekif, Mutaz Al-Khatib, Emad Mohamed, Somaya Eltanbouly, and Mohammed Ghaly. 2026. Islamicturathbench: A multi-task, multi-discipline benchmark for evaluating large language models on the islamic scholarly tradition (turath). *arXiv preprint arXiv:2608.04703*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing

Qin, and Ting Liu. 2025a. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2).

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and 1 others. 2025b. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.

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.

Junyi Li, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023. HaluEval: A large-scale hallucination evaluation benchmark for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6449–6464, Singapore. Association for Computational Linguistics.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. 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*, 37(25):20957–20982.

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, Torino, Italia. ELRA and ICCL.

Hamdy Mubarak, Rana Malhas, Watheq Mansour, Abubakr Mohamed, Mahmoud Fawzi, Majd Hawasly, Tamer Elsayed, Kareem Mohamed Darwish, and Walid Magdy. 2025. Islamiceval 2025: The first shared task of capturing llms hallucination in islamic content. In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 480–493.

Cheng Niu, Yuanhao Wu, Juno Zhu, Siliang Xu, KaShun Shum, Randy Zhong, Juntong Song, and Tong Zhang. 2024. RAGTruth: A hallucination corpus for developing trustworthy retrieval-augmented language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10862–10878, Bangkok, Thailand. Association for Computational Linguistics.

CH-Wang Sky, Benjamin Van Durme, Jason Eisner, and Chris Kedzie. 2024. Do androids know they're only dreaming of electric sheep? In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 4401–4420.

Raul Vazquez, Timothee Mickus, Elaine Zosa, Teemu Vahtola, J"org Tiedemann, Aman Sinha, Vincent Segonne, Fernando Sanchez-Vega, Alessandro Raganato, Jindřich Libovick'y, Jussi Karlgren, Shaoxiong Ji, Jindřich Helcl, Liane Guillou, Ona De Gibert, Jaione Bengoetxea, Joseph Apidianaki, and Marianna Apidianaki. 2025. SemEval-2025 task 3: Mu-SHROOM, the multilingual shared-task on hallucinations and related observable overgeneration mistakes. In *Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025)*, pages 2472–2497, Vienna, Austria. Association for Computational Linguistics.

Hanzhi Zhang, Sumera Anjum, Heng Fan, Weijian Zheng, Yan Huang, and Yunhe Feng. 2025. Poly-FEVER: A multilingual fact verification benchmark for hallucination detection in large language models. *arXiv preprint arXiv:2503.16541*.

A Hallucination in Large Language Models and Related Resources

Definition. LLMs can generate fluent and convincing answers that are not necessarily factually

correct. Hallucination generally refers to generated content that is inaccurate, fabricated, misleading, unsupported by reliable evidence, or inconsistent with the provided input or context (Ji et al., 2023; Huang et al., 2025a). Hallucinations are commonly divided into two broad categories. A *factual hallucination* occurs when a generated statement contradicts verifiable real-world knowledge, invents an entity or event, or incorrectly attributes a fact to a person, place, time, or source. A *faithfulness hallucination* occurs when the generated content is not supported by the input or the provided context, even when the statement may be factually correct in isolation.

Causes of hallucination. Hallucination should not be attributed to a single cause. It can emerge at different stages of the LLM development and generation pipeline (Huang et al., 2025a; Alansari and Luqman, 2026b). One important cause is incomplete knowledge coverage: the information required to answer a question may be absent from, rare in, or insufficiently represented in the model's training corpus. As a result, the model may rely on related language patterns and generate a plausible but unsupported answer. Hallucinations may also arise from incorrect, outdated, noisy, or conflicting information in the training data, which can lead the model to learn unreliable associations.

This issue is particularly relevant for Arabic, where high-quality, diverse, and domain-specific data remain limited, especially in specialized areas such as Islamic knowledge and legal reasoning. Several Arabic-language resources have been introduced to evaluate LLMs on Islamic knowledge, legal reasoning, and general culture (Alwajih et al., 2025; Bouchekif et al., 2025a, 2026b,a; Gaben et al., 2026; Eltanbouly et al., 2026). However, many of these resources rely mainly on multiple-choice questions, where models are evaluated by selecting the correct option. This evaluation setting does not capture how models formulate complete answers. Without evaluating full generated responses, it is difficult to assess model behavior in depth, particularly whether the model introduces hallucinated claims, unsupported explanations, fabricated evidence, or incorrect source attributions.

The next-token prediction objective may further contribute to hallucination because it encourages the model to generate linguistically likely continuations rather than explicitly verify the factual correctness of each statement. Fine-tuning biases, alignment data, prompt ambiguity, insufficient contextual grounding, and decoding decisions may also increase the likelihood of hallucination. Therefore, a model may hallucinate not only because it has not encountered the required information, but also because it fails to retrieve, distinguish, or correctly use knowledge that was present during training.

Related Arabic and multilingual resources. Several recent studies have investigated hallucination in Arabic and multilingual LLMs from complementary perspectives. AraHalluEval introduces a multidimensional manual evaluation framework for Arabic generative QA and summarization, using fine-grained factuality and faithfulness indicators (Alansari and Luqman, 2025). HalluVerse-M^3 provides multilingual and multitask examples constructed through controlled editing and human validation (Abdaljalil et al., 2026). HalluScore focuses on hallucination-prone Arabic questions across multiple domains, cultural settings, and reasoning requirements, and provides fine-grained labels and explanations (Alansari and Luqman, 2026a). In the Islamic domain, Aftina studies hallucination mitigation in fatwa generation using retrieval-augmented generation and re-ranking (Mohammed et al., 2025), while IslamicEval evaluates the identification, validation, and correction of Qur'anic and Hadith quotations (Mubarak et al., 2025).
These resources differ in their languages, domains, annotation procedures, and levels of supervision. Importantly, AraHalluEval and HalluScore already go beyond binary response-level labels by providing fine-grained annotations, and HalluScore additionally provides explanations. However, neither resource provides exact character-level localization of the hallucinated content within generated responses. Span annotations are available in IslamicEval, but they identify complete intended Qur'anic and Hadith quotation spans rather than the specific erroneous segments within the overall response. In contrast, HalluTruthQA directly annotates the exact character-level spans responsible for factual errors and associates them with fine-grained hallucination types, human-written explanations, and verified factual answers. Table 3 summarizes these differences.

Overall, the existing resources address comple-

Table 3: Feature-based comparison of HALLUTRUTHQA with recent Arabic and multilingual hallucination resources. "Partial" indicates that the feature is available only for a subset of the data or is used only for system-level evaluation.

Resource	Language(s)	Domain / Task	Outputs	Manual annotation	Span annotation	Fine-grained labels	Explanations	Verified answer / Correction
ARAHALLUEVAL (Alansari and Luqman, 2025)	Arabic	Generative QA and abstractive summarization	LLM outputs	Yes; two annotators and expert adjudication	No	12 factuality and faithfulness indicators	No error-specific explanations	Reference answer for QA or source document for summarization
HALLUVERSE-M^3 (Abdaljalil et al., 2026)	English, Arabic, Hindi, and Turkish	QA and dialogue summarization	Controlled LLM-edited outputs	Human validation	No	Entity-, relation-, and sentence-level hallucination types	No	Original ground-truth output
HALLUSCORE (Alansari and Luqman, 2026a)	Arabic	Multi-domain generative QA	LLM outputs	Yes; human annotation of model responses	Partial; hallucinated text is recorded only for partial-hallucination cases, not as a general character-offset localization task	Factual, faithfulness, and partial hallucination labels, together with multi-label question metadata	Ground-truth answer explanations generated by an LLM and manually reviewed	Verified ground-truth answer and supporting source link
AFTINA (Mohammed et al., 2025)	Arabic	Islamic fatwa QA	LLM outputs + RAG	Partial; expert evaluation of 210 generated outputs	No	No corpus-level hallucination taxonomy	No	Dar Al-Ifta QA corpus and retrieved supporting content
ISLAMICEVAL (Mubarak et al., 2025)	Arabic	Qur'an and Hadith	LLM outputs	Yes; Islamic-studies experts	Yes; intended Qur'anic and Hadith quotation spans	Correct Qur'an, Incorrect Qur'an, Correct Hadith, and Incorrect Hadith	No	Canonical correction for incorrect quotations, or an indication that no valid correction exists
HALLUTRUTHQA Arabic	**Islamic knowledge, history, science, and geography**	**LLM outputs**	**Yes; domain-expert annotation followed by independent verification**	**Yes; exact character-level erroneous spans**	**Macro- and micro-level hallucination types**	**Human-written explanations of the localized errors**	**Verified reference answer and six candidate answers**	

mentary aspects of hallucination evaluation. ARAHALLUEVAL provides broad multidimensional evaluation across two Arabic generation tasks, while HALLUVERSE-M^3 supports controlled comparisons across languages and tasks. HALLUSCORE emphasizes hallucination-prone Arabic questions and provides verified evidence, answer explanations, and human response-level annotations. AFTINA primarily evaluates hallucination mitigation through retrieval and re-ranking, whereas ISLAMICEVAL specializes in the accurate identification and correction of Qur'anic and Hadith quotations. In contrast, HALLUTRUTHQA jointly supports response-level detection, exact erroneous-span localization, macro- and micro-level error classification, human-written explanation, and factual-answer verification in knowledge-intensive Arabic QA.

A.1 Question Construction and Verification

All questions were manually prepared by domain experts based on their knowledge and experience. For each question, the corresponding expert verified the reference answer using reliable sources. Experts were instructed to formulate clear, knowledge-intensive questions while maintaining diversity in topics and difficulty levels. Table 4 provides representative examples of the sources consulted for each domain; these examples are not intended to constitute an exhaustive list. For Islamic Knowledge, experts additionally checked Qur'anic wording and references, hadith attributions, and the factual support of religious claims. During verification, questions and reference answers were reviewed for accuracy, clarity, and ambiguity before inclusion in the benchmark.

After the initial annotation, two trained research assistants reviewed the questions, reference answers, hallucination labels, localized spans, explanations, and answer options. Unclear or disputed cases were returned to the original domain expert for final verification. The annotation process was based on meaning rather than exact string matching. Therefore, semantically equivalent answers were accepted when they expressed the same fact. This also applied to different date formats. For example, corresponding Hijri and Gregorian dates were accepted when the experts judged them to refer to the same historical event. In the example provided in Appendix A, "40 AH" and "661 CE" were treated as equivalent. Since both the questions and hallucination annotations were manually reviewed, differences between Hijri and Gregorian date formats were not automatically labeled as hallucinations.

B Data Format

Each instance is stored in a structured JSON format. The format includes the question, generated response, reference answer, response-level hallucination label, span-level annotations, explanations, multiple-choice candidate answers, and the correct option.

Domain	Representative Sources
Islamic Knowledge	IslamWeb; IslamQA; سؤال وجواب في القرآن الكريم 1213 (*1,213 Questions and Answers on the Holy Qur'an*) by Ahmad bin Ali Abu Islam; 200 سؤال وجواب في العقيدة الإسلامية (*200 Questions and Answers on Islamic Creed*) by Hafiz bin Ahmad Al-Hakami; السيرة النبوية في سؤال وجواب (*The Prophetic Biography in Questions and Answers*) by Abu Abdullah Muhammad Ali Samak; أكثر من 1500 سؤال وجواب في القرآن الكريم (*More Than 1,500 Questions and Answers on the Holy Qur'an*) by Muhsin Hussein Al-Ghamdi; and الموسوعة القرآنية (*The Qur'anic Encyclopedia*).
History	أطلس تاريخ الإسلام (*Atlas of Islamic History*); المفصل في تاريخ العرب قبل الإسلام (*The Detailed History of the Arabs Before Islam*).
Geography	أطلس الوطن العربي (*Atlas of the Arab World*) by Yahya Nabhan; جغرافية العالم الإسلامي (*Geography of the Islamic World*) by Mahmoud Taha Abu Al-Ala; and الموسوعة العربية العالمية (*The Global Arabic Encyclopedia*).
Science	موسوعة العلوم والتقانات (*Encyclopedia of Science and Technology*); الموسوعة العلمية الميسرة (*The Concise Scientific Encyclopedia*), edited and reviewed by Ahmad Shafiq Al-Khatib; and الموسوعة العربية العالمية (*The Global Arabic Encyclopedia*).

Table 4: Representative sources consulted by domain experts during question construction and reference-answer verification.

Hallucinated Answer


```
{
  "id": "islamicHallucination_8",
  "question":
                                    "ما البلد الذي ذُكر اسمه في القرآن الكريم؟",
  "generated_answer":
"بلد واحد فقط ذُكر اسمه في القرآن الكريم وهو مصر، وذلك في سورة يوسف الآية 47:
وَقَالَ الْمَلِكُ ائْتُونِي بِهِ فَلَمَّا جَاءَهُ الرَّسُولُ قَالَ ارْجِعْ إِلَى رَبِّكَ فَسْأَلْهُ مَا لِذِي النِّسْوَةِ إِنَّ
قَدْ ظَلَمْتُ إِنَّ",
  "gold_answer":
"البلد الذي ذُكر اسمه صراحة في القرآن الكريم هو مصر، ومن ذلك قوله تعالى: «وَقَالَ
الَّذِي اشْتَرَاهُ مِن مِّصْرَ لِامْرَأَتِهِ أَكْرِمِي مَثْوَاهُ»، وقوله تعالى: «ادْخُلُوا مِصْرَ إِن شَاءَ اللَّهُ
آمِنِينَ",
  "generator_model": "QCRI/Fanar-1-9B-Instruct",
  "label": "hallucination",
  "hallucinations": [
    {
      "span_start": 67,
      "span_end": 75,
      "hallucinated_span":
                                                "الآية 47",
      "explanation":
        "الإحالة إلى الآية غير صحيحة؛ اسم مصر لم يرد في يوسف: 47؛ وإنما ورد في
                                مواضع أخرى من السورة، منها يوسف: 21 ويوسف: 99."
    },
    {
      "span_start": 77,
      "span_end": 226,
      "hallucinated_span":
        "وَقَالَ الْمَلِكُ ائْتُونِي بِهِ فَلَمَّا جَاءَهُ الرَّسُولُ قَالَ ارْجِعْ إِلَى رَبِّكَ فَسْأَلْهُ مَا لِذِي
        النِّسْوَةِ إِنَّ قَدْ ظَلَمْتُ إِنَّ",
      "explanation":
        "هذا نص مخترع منسوب إلى القرآن، وفيه ألفاظ غير قرآنية مثل النسوة؛ والآية
        الصحيحة في يوسف: 47 تتحدث عن الزراعة والسنين: قال تزرعون سبع سنين
                                                دأبا."
    }
  ],
  "options": {
    "A":
      "- البلد الذي ذُكر اسمه صراحة في القرآن الكريم هو مصر، وقد ورد اسمها في سياق
      قصة يوسف عليه السلام.",
    "B":
      "- البلد الذي ذُكر اسمه صراحة في القرآن الكريم هو فلسطين، وقد ورد ذكرها في
      سياق الأرض المقدسة.",
    "C":
      "- البلد الذي ذُكر اسمه صراحة في القرآن الكريم هو العراق، وقد ارتبط بعدد من
      قصص الأنبياء والأمم السابقة.",
    "D":
      "- البلد الذي ذُكر اسمه صراحة في القرآن الكريم هو الأردن، وقد ارتبط بأحداث
      تاريخية وقعت في بلاد الشام.",
    "E":
      "- البلد الذي ذُكر اسمه صراحة في القرآن الكريم هو اليمن، وقد ارتبط بقصة سبأ
      وملكة سبأ.",
    "F":
      "- البلد الذي ذُكر اسمه صراحة في القرآن الكريم هو السعودية، وقد ارتبط بذكر مكة
      المكرمة والبيت الحرام."
  },
  "answer": "A"
}
```


For non-hallucinated responses, the span list is empty. This unified format allows the same dataset to be used for response-level hallucination detection, explainable hallucination analysis, span-level hallucination localization, multiple-choice factual verification, and training models to select or generate factually reliable answers.

Non-Hallucinated Answer


```
{
  "id": "historicHallucination_474",
  "question":
                                    "في أي عام انتهى عصر الخلفاء الراشدين؟",
  "generated_answer":
                                "انتهى عصر الخلفاء الراشدين في العام 40 هـ.",
  "gold_answer":
                                                "661 م",
  "generator_model": "QCRI/Fanar-1-9B-Instruct",
  "label": "no_hallucination",

}
```

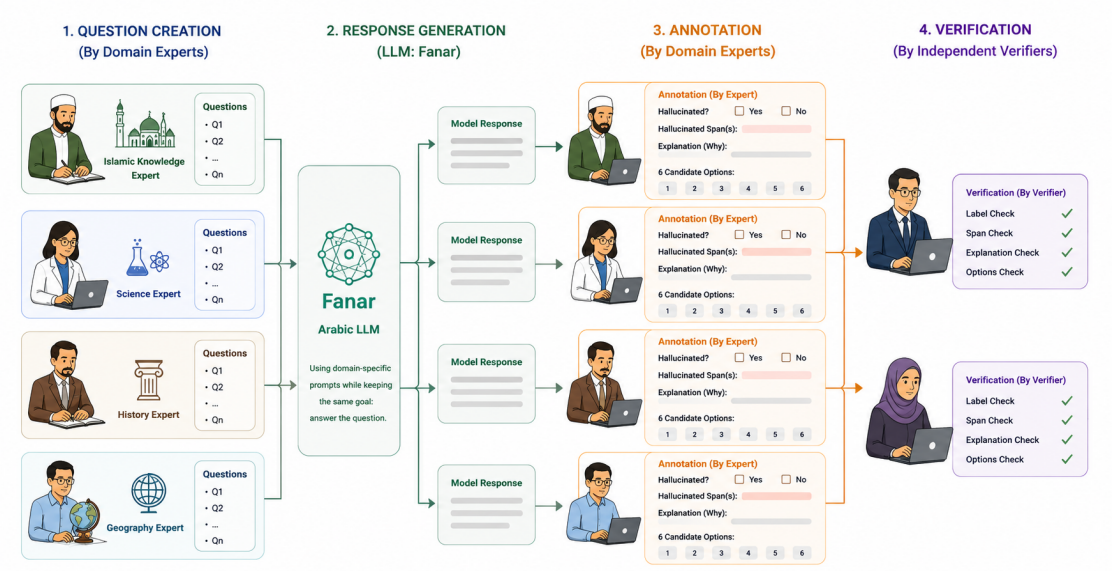


Figure 2: Overview of the HALLUTRUTHQA construction pipeline.

C Annotation Process and Reliability

The annotation process involved two stages. First, four domain experts annotated the dataset, with each expert responsible for one domain: Islamic knowledge, history, science, or geography. Each domain expert had a relevant academic background in the corresponding knowledge area.

The experts manually annotated the binary hallucination label, hallucination macro-type, hallucination micro-type, erroneous spans, explanations, and reference answers. They also manually constructed and validated the multiple-choice candidate options and the corresponding answer key. Annotators were instructed to ensure that the candidate options followed the same style, wording, and level of specificity as the model-generated answer, so that the correct option could not be identified from stylistic cues.

Second, two trained research assistants independently verified the annotations produced by the four domain experts. The verification pass checked the binary hallucination label, hallucination categories, span boundaries, explanations, and the correctness of the multiple-choice answer key. Since the four domain experts annotated disjoint domain-specific subsets, agreement was not computed among the four experts on shared items. Instead, we report expert–reviewer agreement between the initial domain-expert annotations and the independent verification pass.

Disagreements or flagged cases were resolved through adjudication by the original domain expert after the second verification pass. Agreement was computed before final expert adjudication. We used Cohen's κ for categorical annotations and character-level F1 and intersection-over-union (IoU) for span-level annotations. Disagreements and flagged cases were subsequently resolved through adjudication by the original domain expert after the second verification pass.

Annotation dimension	Metric	Score
Binary hallucination label	Cohen's κ	0.93
Macro hallucination type	Cohen's κ	0.87
Micro hallucination type	Cohen's κ	0.81
Multiple-choice answer key	Agreement	0.979
Span localization	Character-F1	0.89
Span overlap	IoU	0.84

Table 5: Expert–reviewer agreement for HalluTruthQA before final adjudication. Cohen's κ is reported for categorical annotations, agreement is reported for the multiple-choice answer key, and character-level F1 and IoU are used for span-level annotations.

The agreement results indicate that the annotation process was reliable across all dimensions. Binary hallucination labeling achieved the highest agreement ($\kappa = 0.93$), showing that the domain-expert annotations and the independent verification pass were highly consistent in deciding whether a generated answer contained hallucinated content. Agreement remained strong for macro-level hallucination categories ($\kappa = 0.87$), suggesting that the main hallucination types were clearly defined and consistently applied.

Agreement was slightly lower for micro-level categories ($\kappa = 0.81$), which is expected given the difficulty of distinguishing closely related fine-grained error types, such as unsupported claims, wrong attribution, citation mismatch, and over-specific additions. Nevertheless, this score still indicates strong agreement for fine-grained hallucination classification.

The multiple-choice answer key achieved 97.9% agreement, indicating that the correct option was generally unambiguous. For span-level annotation, the character-level F1 of 0.89 and IoU of 0.84 show that the reviewers largely identified the same erroneous content as the original expert annotations. Remaining differences were mainly due to span-boundary variation, such as whether only the minimal erroneous entity, date, or reference was marked, or whether surrounding explanatory context was also included.

Overall, these results confirm that the annotations are robust. They also show that response-level hallucination detection is the most stable annotation task, whereas micro-type classification and exact span localization are more challenging but still highly reliable.

D Hallucination Taxonomy

Figure 3 presents the hierarchical taxonomy of hallucination types used in HALLUTRUTHQA. The taxonomy was designed to describe the erroneous spans observed during annotation. It contains five macro-types: *Factual Contradiction*, *Context Inconsistency*, *Logical Inconsistency*, *Factual Fabrication*, and *Nonsensical / Irrelevant Response*. Each macro-type is further divided into fine-grained micro-types that describe the specific form of the error.

Purpose of the taxonomy The goal of the taxonomy is to move beyond a binary distinction between hallucinated and non-hallucinated answers. In Arabic QA, an answer can be fluent and relevant but still contain a small factual error, a wrong source attribution, an unsupported explanation, or a fabricated reference. For this reason, each hallucinated answer in HALLUTRUTHQA is annotated not only with a response-level label, but also with character-level erroneous spans, explanations, and hallucination types.

Macro and micro types Table 6 defines the macro-types and their corresponding micro-types. The macro-types capture the main ways in which a model answer can fail, while the micro-types provide a more precise description of the erroneous span. For example, a factual contradiction may involve an entity, date, numeric, or definition error, whereas a context inconsistency may involve a citation mismatch or a wrong source attribution.

E Hallucination Examples

To make the taxonomy easier to interpret, we provide examples for the hallucination micro-types observed in HALLUTRUTHQA. Each example is labeled with its hallucination type and includes the question, the model prediction, the verified answer, and a short explanation of the error.

These examples illustrate how different hallucination types appear in Arabic QA. Some cases involve direct factual contradictions, such as wrong entities, dates, numbers, or locations. Other cases involve context inconsistency, where the answer remains related to the question but the supporting evidence, citation, or source attribution is incorrect. We also include examples of logical inconsistency, factual fabrication, and nonsensical or irrelevant responses.

1 Factual Contradiction – Entity Error

Question

من هو الصحابي الذي دعا له النبي صلى الله عليه وسلم بقوله: اللهم فقهه في الدين؟

Who was the Companion for whom the Prophet, peace be upon him, supplicated: "O Allah, grant him understanding of the religion"?

Prediction

معاذ بن جبل رضي الله عنه

Muadh ibn Jabal, may Allah be pleased with him.

Gold answer

Abdullah Ibn Abbas الاسم هو عبد الله بن عباس

Comment

The model predicts the wrong Companion instead of Abdullah Ibn Abbas.

2 Factual Contradiction – Date / Temporal Error

Question

في أي عام تم توحيد ألمانيا؟

In which year was Germany unified?

Prediction

تم توحيد ألمانيا في العام 1871.

Germany was unified in the year 1871.

Gold answer

1990

Comment

Although the German Empire was founded in 1871, the term "German unification" in contemporary contexts refers to the reunification of East and West Germany in 1990.

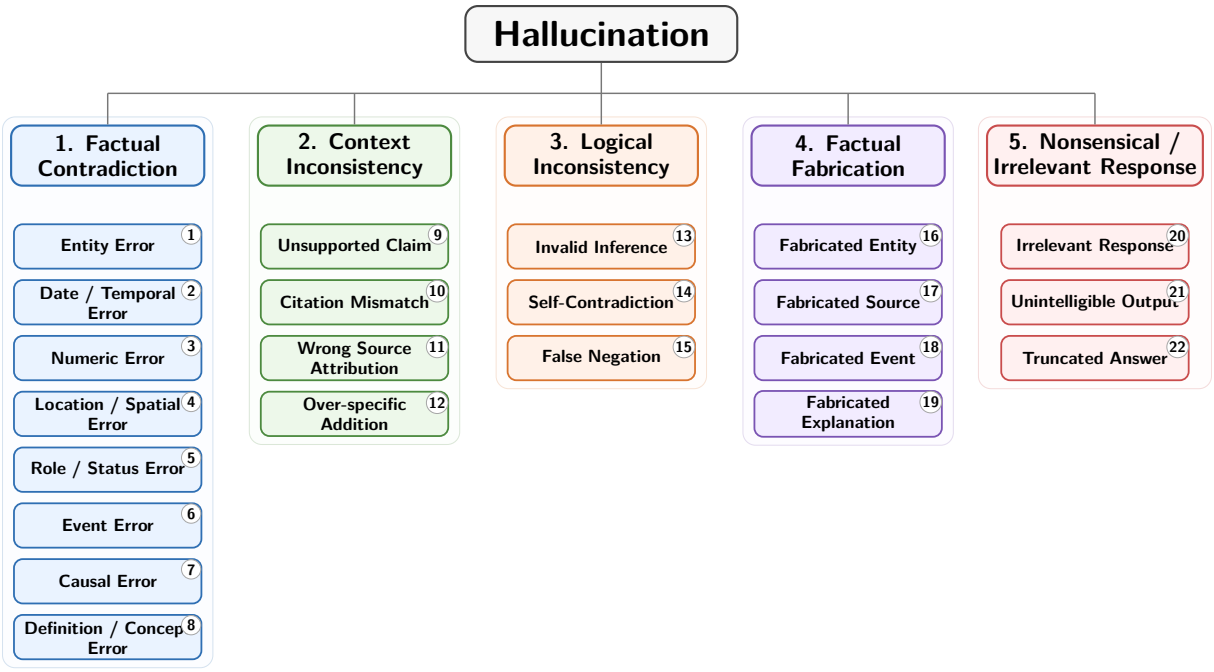


Figure 3: Hierarchical taxonomy of hallucination macro-types and observed micro-types.

3 Factual Contradiction – Numeric Error

Question

ما الطول التقريبي لنهر سانت لورانس، ومن أي بحيرة ينبع؟

What is the approximate length of the Saint Lawrence River, and from which lake does it originate?

Prediction

نهر سانت لورانس يمتد تقريبًا لـ 3,058 كيلومتر (1,900 ميل). وهو ينبع من بحيرة أونتاريو، التي هي واحدة من البحيرات الكبرى في أمريكا الشمالية.

The Saint Lawrence River extends for approximately 3,058 kilometers (1,900 miles). It originates from Lake Ontario, which is one of the Great Lakes of North America.

Gold answer

نحو 1,197 كم، وينبع من بحيرة أونتاريو.

Approximately 1,197 km, and it originates from Lake Ontario.

Explanation

الخطأ في الإجابة هو ذكر طول غير صحيح لنهر سانت لورانس. فطول النهر من بحيرة أونتاريو حتى المصب في خليج سانت لورانس نحو 1,197 كم، وليس 3,058 كم. وقد يشير الرقم 3,058 كم إلى منظومة البحيرات العظمى-سانت لورانس بأكملها، لا إلى النهر نفسه.

The error is the incorrect length given for the Saint Lawrence River. The river's length from Lake Ontario to its mouth in the Gulf of Saint Lawrence is approximately 1,197 km, not 3,058 km. The figure 3,058 km may refer to the entire Great Lakes–Saint Lawrence system, not to the river itself.

4 Factual Contradiction – Location / Spatial Error

Question

ما المدينة التي تسمى 'مدينة الموتى' في مصر؟

Which place is called the "City of the Dead" in Egypt?

Prediction

المدينة التي تُعرف باسم ``مدينة الموتى" في مصر هي مدينة الأقصر. ولكن، من الجدير بالذكر أن هذا الاسم ليس رسمياً وقد يكون مستخدماً بشكل غير رسمي بسبب الكثافة العالية للمواقع الأثرية والمدافن القديمة فيها.

The place known as the "City of the Dead" in Egypt is Luxor. However, it should be noted that this name is not official and may be used informally because of the high density of archaeological sites and ancient tombs there.

Gold answer

القرافة هي مدينة الموتى في قلب القاهرة الآن.

Al-Qarafa is the City of the Dead, located today in the heart of Cairo.

Explanation

The error is identifying Luxor as the "City of the Dead" in Egypt. This name refers to Al-Qarafa, the historic cemetery district in Cairo, not to the city of Luxor.

Macro-type	Micro-type	Definition
Factual Contradiction	Entity Error	The response gives an incorrect person, country, organization, object, or named entity instead of the verified answer.
	Date / Temporal Error	The response gives an incorrect date, year, period, duration, chronological order, or temporal relation.
	Numeric Error	The response gives an incorrect number, quantity, percentage, distance, length, count, or measurement.
	Location / Spatial Error	The response gives an incorrect place, region, country, spatial relation, or geographical location.
	Role / Status Error	The response assigns an incorrect role, title, status, function, affiliation, or relation to an entity.
	Event Error	The response describes a real or expected event incorrectly, assigns it to the wrong actor, time, place, or outcome, or confuses it with another event.
	Causal Error	The response gives an incorrect cause, effect, motivation, or explanatory relation between facts.
	Definition / Concept Error	The response confuses related concepts or gives a definition that contradicts the verified answer.
Context Inconsistency	Unsupported Claim	The response adds a claim that remains related to the question but is not supported by the verified evidence.
	Citation Mismatch	The response links a claim to the wrong citation, verse, hadith, book, source, or evidential reference.
	Wrong Source Attribution	The response attributes a statement to a source, scholar, text, or authority that does not support that statement.
	Over-specific Addition	The response adds details that are more specific than what the verified evidence allows, without necessarily inventing a new source or event.
Logical Inconsistency	Invalid Inference	The response uses related information but draws a conclusion that does not logically follow from it.
	Self-Contradiction	The response contains two or more claims that are mutually incompatible within the same answer.
	False Negation	The response incorrectly denies a true relation or states the opposite of the verified answer.
Factual	Fabricated Entity	The response introduces a non-attested entity and presents it as factual.
	Fabricated Source	The response invents a source, citation, reference, verse, hadith, book, or authority that does not exist or is not verified.
	Fabricated Event	The response describes an event that is not supported by the verified sources and appears to be invented.
	Fabricated Explanation	The response gives an explanation that sounds specific or authoritative but is not grounded in the verified evidence.
Nonsensical /	Irrelevant Response	The response is off-topic or does not address the question in a meaningful way.
	Unintelligible Output	The response is too unclear, incoherent, or broken to be interpreted as a valid factual answer.
	Truncated Answer	The response is incomplete or stops before providing a verifiable answer.

Table 6: Macro-types, micro-types, and definitions in the HALLUTRUTHQA hallucination taxonomy. The macro-level cues clarify the main distinction between wrong factual values, wrong grounding, wrong reasoning, invented factual content, and unusable responses.

5 Factual Contradiction – Role / Status Error

Question

من هو القائد الذي حرر الجزائر من الاستعمار الفرنسي سياسيًا؟

Who was the leader who politically liberated Algeria from French colonialism?

Prediction

القائد الذي حرر الجزائر من الاستعمار الفرنسي سياسيًا هو أحمد بن بلة. تولى رئاسة الحكومة المؤقتة للجمهورية الجزائرية في عام 1962 بعد مفاوضات إيفيان التي أدت إلى استقلال البلاد عن فرنسا.

The leader who politically liberated Algeria from French colonialism was Ahmed Ben Bella. He headed the Provisional Government of the Algerian Republic in 1962 after the Evian negotiations that led to Algeria's independence from France.

Gold answer

Benyoucef Benkhedda. يوسف بن خدة

Comment The model incorrectly attributes the presidency of the Provisional Government of the Algerian Republic during the Evian period to Ahmed Ben Bella. Ben Bella was imprisoned during the Evian negotiations, while Benyoucef Benkhedda headed the GPRA at that stage.

6 Factual Contradiction – Event Error

Question

ما المعركة التي حدثت بين علي بن أبي طالب وبين الخوارج؟

Which battle took place between Ali ibn Abi Talib and the Kharijites?

Prediction

معركة الجمل هي المعركة التي حدثت بين علي بن أبي طالب والخوارج. ولكن يجب التنويه إلى أن الخوارج لم يكونوا خصوم علي في هذه المعركة تحديداً، بل كانت معاوية بن أبي سفيان هو الخصم الرئيسي في معركة الجمل، بينما كانت معركة النهروان هي المعركة التي دارت بين علي بن أبي طالب والخوارج. لكن إذا كنت تقصد أي معركة بين علي والخوارج، فإن معركة النهروان هي الأكثر شهرة.

The Battle of the Camel was the battle that took place between Ali ibn Abi Talib and the Kharijites. However, it should be noted that the Kharijites were not Ali's opponents in this specific battle; rather, Muawiya ibn Abi Sufyan was the main opponent in the Battle of the Camel, whereas the Battle of Nahrawan was the battle that took place between Ali ibn Abi Talib and the Kharijites. But if you mean any battle between Ali and the Kharijites, then the Battle of Nahrawan is the most famous.

Gold answer

المعركة التي حدثت بين علي بن أبي طالب والخوارج هي معركة النهروان عام 38 هـ.

The battle that took place between Ali ibn Abi Talib and the Kharijites was the Battle of Nahrawan in 38 AH.

Explanation

The first error is identifying the Battle of the Camel as the battle between Ali ibn Abi Talib and the Kharijites, whereas the correct battle is the Battle of Nahrawan. The statement that Muawiya ibn Abi Sufyan was the main opponent in the Battle of the Camel is also incorrect; Muawiya is mainly associated with the Battle of Siffin, not the Battle of the Camel.

7 Factual Contradiction – Causal Error

Question

لماذا ينصح المرأة المرضع بعدم تناول الكحول؟

Why is a breastfeeding woman advised not to consume alcohol?

Prediction

يُسمح بالكحول أثناء الرضاعة بكميات معتدلة لأنه يُسرّع امتصاص الحديد في الحليب ولا يصل للرضيع.

Alcohol is allowed during breastfeeding in moderate amounts because it accelerates iron absorption in breast milk and does not reach the infant.

Gold answer

ينصح المرأة المرضع بعدم تناول الكحول لأن الكحول يمكن أن ينتقل إلى حليب الأم ويصل إلى الرضيع، وقد يؤثر في صحته ونموه ونمط نومه.

A breastfeeding woman is advised not to consume alcohol because alcohol can pass into breast milk and reach the infant, potentially affecting the infant's health, development, and sleep patterns.

Explanation

The error lies in giving a false causal justification for alcohol consumption during breastfeeding. Alcohol is not recommended because it does not accelerate iron absorption in breast milk, and the claim that it does not reach the infant is incorrect; alcohol can pass into breast milk and expose the infant.

8 Factual Contradiction – Definition / Concept Error

Question

ما هي الميتة الحلال والدم الحلال؟

What are the lawful dead animals and the lawful blood in Islamic law?

Prediction

الميتة الحرام هي ما مات بغير ذكاة شرعية، أما الميتة الحلال فهي ما ماتت ذكاة شرعية. أما الدم الحرام فهو الدم الذي لا يجوز أكله، مثل دم الحيوانات الميتة أو التي لم تذبح بشكل صحيح، بينما الدم الحلال هو دم الحيوانات التي ذبحت بطريقة شرعية. ولكن يجب التنويه إلى أن الإسلام يحرم تناول الدم بشكل عام، سواء كان من حيوان مذبوح بشكل صحيح أم لا.

Forbidden carrion is what dies without lawful slaughter, whereas lawful carrion is what dies through lawful slaughter. Forbidden blood is blood that may not be consumed, such as the blood of dead animals or animals not slaughtered properly, whereas lawful blood is the blood of animals slaughtered according to Islamic law. However, it should be noted that Islam generally forbids consuming blood, whether from a properly slaughtered animal or not.

Gold answer

الميتة الحلال هي السمك والجراد، والدم الحلال هو الكبد والطحال.

The lawful dead animals are fish and locusts, and the lawful blood refers to the liver and spleen.

Explanation

The error lies in giving an incorrect definition of lawful carrion and lawful blood. Lawful carrion does not mean an animal that died through lawful slaughter; rather, it refers to the recognized exceptions: fish and locusts. Similarly, lawful blood does not mean the blood of a properly slaughtered animal, but refers to the liver and spleen.

9 Context Inconsistency – Unsupported Claim

Question

ما هي أكبر جزيرة في العالم؟

What is the largest island in the world?

Prediction

جرينلاند هي الأكبر بمساحة تبلغ حوالي 2,166,086 كيلومتر مربع. ومع ذلك، يعتمد هذا على كيفية تعريف ``الجزر'' -- فجرينلاند تعتبر أحياناً شبه جزيرة بسبب اتصالها القاري مع كندا عبر مضيق دنمارك الصغير.

Greenland is the largest island, with an area of about 2,166,086 square kilometers. However, this depends on how "islands" are defined – Greenland is sometimes considered a peninsula because of its continental connection with Canada through the narrow Denmark Strait.

Gold answer

غرينلاند

Greenland.

Explanation

The model gives the correct main answer, but then adds an unsupported claim about Greenland being sometimes considered a peninsula because of a supposed continental connection with Canada. This extra claim is not supported by the question or the reference answer and introduces unnecessary, ungrounded information.

10 Factual Contradiction – Location / Spatial Error

Question

ما اسم البحيرة الواقعة على حدود زيمبابوي وزامبيا والمعروفة بأنها من أكبر البحيرات الاصطناعية؟

What is the name of the lake located on the border between Zimbabwe and Zambia, known as one of the largest artificial lakes?

Prediction

البحيرة التي تشير إلى هي بحيرة كاريبا، وهي تقع على الحدود بين زيمبابوي وزامبيا، وتشتهر بوجود شلالات فيكتوريا الشهيرة.

The lake referred to is Lake Kariba, located on the border between Zimbabwe and Zambia, and it is famous for the presence of Victoria Falls.

Gold answer

الاسم هو بحيرة كاريبا (Lake Kariba)

Error. The model identifies the correct lake, but adds a false geographical association with Victoria Falls.

11 Factual Contradiction – Citation Mismatch

Question

هل سنّ النبي صلى الله عليه وسلم الإسراع بتجهيز الميت أم الإبطاء به؟

Did the Prophet, peace be upon him, prescribe hastening the preparation of the deceased or delaying it?

Prediction

لم يسن النبي صلى الله عليه وسلم الإسراع أو الإبطاء بشكل خاص بتجهيز الميت، ولكن يُفضل الإسراع في تجهيزه دون تأخير طويل، وذلك لما روي عن عائشة رضي الله عنها أنها قالت: كان رسول الله صلى الله عليه وسلم إذا حضرته الوفاة جعل عنده شيئاً من التراب فيمسحه بوجهه ويقول: بِسْمِ اللَّهِ وَبِعِزَّةِ اللَّهِ، وَأَعُوذُ بِاللَّهِ .

The Prophet, peace be upon him, did not specifically prescribe either hastening or delaying the preparation of the deceased; however, it is preferable to prepare the deceased quickly without long delay, based on a report attributed to Aisha, may Allah be pleased with her, stating that when the Prophet was near death, he would place some dust near him, wipe his face with it, and say: In the name of Allah and by the might of Allah, and I seek refuge in Allah from....

Gold answer

سنّ النبي صلى الله عليه وسلم الإسراع بتجهيز الميت ودفنه، وعدم تأخيره بلا حاجة.

The Prophet, peace be upon him, prescribed hastening the preparation and burial of the deceased, and not delaying it without need.

Explanation

الخطأ الأول هو نفي وجود سنة خاصة بالإسراع في تجهيز الميت، بينما السنة هي الإسراع بتجهيزه ودفنه وعدم تأخيره بلا حاجة. أما الخطأ الثاني فهو الاستدلال برواية غير مناسبة لا تدعم الحكم المتعلق بتجهيز الميت.

The first error is denying that there is a specific prophetic practice of hastening the preparation of the deceased, whereas the Sunnah is to prepare and bury the deceased promptly without unnecessary delay. The second error is the use of an inappropriate and incomplete narration that does not support the ruling about preparing the deceased.

12 Context Inconsistency – Wrong Source Attribution

Question

أشار القرآن الكريم إلى عدة مهن كانت تزاول في الأمم السابقة، منها مهنة بناء السفن، فما الآيات التي أشارت إلى هذه المهنة؟ ومن هو النبي الذي زاول هذه المهنة؟

Prediction

الآية 14 من سورة هود؛ موسى عليه السلام كان يزاول مهنة بناء السفن؛ الآية 41 من سورة الأنبياء؛ الآية 40 من سورة هود؛ النبي الذي زاول مهنة بناء السفن هو موسى عليه السلام

Gold answer

هود: 37، 38؛ نوح عليه السلام

Error. The answer provides wrong Qur'anic references and assigns the action to the wrong prophet.

13 Logical Inconsistency – Invalid Inference

Question

ما السرية التي أرسل فيها سبعون من الصحابة فقُتل أكثرهم عند بئر معونة؟

Prediction

سرية الخرار؛ بقيادة عبد الله بن حرام؛ أكثر من سبعين

Gold answer

سرية بئر معونة، وتسمى أيضاً سرية القراء.

Error. The answer gives unsupported details and an incorrect inference.

14 Factual Fabrication – Fabricated Entity

Question

ما اسم الحوض الذي يضم أعمق نقطة في بحر بافين شمال كندا، وما عمقه التقريبي؟

Prediction

حوض نيلسون، وعمقه التقريبي حوالي 1,450 متر

Gold answer

حوض بافين بعمق نحو 2,400 م

Error. The model introduces an incorrect or fabricated basin name.

15 Factual Fabrication – Fabricated Source / Quote

Question

ما البلد الذي ذُكر اسمه في القرآن الكريم؟

Which country is mentioned by name in the Qur'an?

Prediction

بلد واحد فقط ذُكر اسمه في القرآن الكريم وهو مصر، وذلك في سورة يوسف الآية 47: وَقَالَ الْمَلِكُ ائْتُونِي بِهِ فَلَمَّا جَاءَهُ الرَّسُولُ قَالَ ارْجِعْ إِلَىٰ رَبِّكَ فَسْأَلْهُ مَا بَالُ النِّسْوَةِ اللَّاتِي قَطَّعْنَ أَيْدِيَهُنَّ إِنَّ رَبِّي بِكَيْدِهِنَّ عَلِيمٌ

Only one country is mentioned by name in the Qur'an, namely Egypt, in Surah Yusuf, verse 47: "And the king said: Bring him to me. Then, when the messenger came to him, he said: Return to your lord and ask him about what belongs to the two breasts..."

Gold answer

مصر

Egypt.

Error. The model gives the correct short answer, but fabricates a Qur'anic-style quotation and falsely attributes it to Surah Yusuf, verse 47.

16 Nonsensical Response – Irrelevant / Broken Output

Question

من هو الملك الهاشمي الذي يعرف بلقب الملك المؤسس؟

Prediction

العرابيين

Gold answer

الملك عبدالله الأول

Error. The output is malformed and does not answer the question.

17 Factual Fabrication – Date / Temporal Error

Question

ما المدينة التي تسمى ``عاصمة الثقافة العربية 2024"؟

Prediction

مدينة سطيف الجزائرية

Gold answer

المدينة التي حملت لقب `عاصمة الثقافة العربية لعام 2024" هي مدينة طرابلس اللبنانية. كانت مقررة أصلاً لعام 2023، لكنها تأجّلت بسبب الأزمة الاقتصادية اللبنانية. أما مدينة سطيف الجزائرية فقد حملت لقب `عاصمة الثقافة العربية" في عام 2007.

Error. The model fabricates the selected city by answering Setif, Algeria, whereas the correct city for 2024 is Tripoli, Lebanon. Setif was previously designated as the Arab Capital of Culture in 2007.

F Explanation Evaluation Prompt

Prompt for Explanation Evaluation

We evaluate the quality of the model explanation for hallucinated answers. This prompt is used only when both the gold label and the predicted label are *hallucination*.

The evaluation is based on two dimensions:

1. Error Identification (0–1)
   The explanation should identify the hallucinated span or claim and clearly explain why it is incorrect.

2. Factual Correction (0–1)
   The explanation should provide the correct factual information or a valid correction of the hallucinated content.

Each dimension is scored independently on a continuous scale from 0 to 1, where 0 means completely incorrect or missing, and **1** means fully correct and complete. The scoring should be consistent across examples: explanations with similar quality should receive similar scores.

The evaluator must return JSON only:

```
{
  "error_identification": float,
  "factual_correction": float,
  "reason": "short justification"
}
```

MAQAM: A Richly Annotated Corpus for Quranic Arabic Morphological Analysis and Disambiguation

Mostafa Saeed[1,2] **Mohammad Amjad**[1] **Mahmoud Kassem**[1]
Rawan Bondok[1] **Mayar Nassar**[1,3] **Adel Mahmoud Wizani**[1]
Salman Elgamal[1] **Mahmoud Reda**[1] **Nizar Habash**[1]
[1]Computational Approaches to Modeling Language Lab, New York University Abu Dhabi
[2]Mohamed bin Zayed University of Artificial Intelligence [3]Ain Shams University
{mostafa.saeed,nizar.habash}@nyu.edu

Abstract

Quranic Arabic poses important challenges for Arabic NLP due to its distinctive morphology, syntax, and orthography. We present **MAQAM**, a richly annotated **M**orphosyntactic **A**rabic **Q**uranic **A**nalysis and **M**odeling resource covering the entire Quran, providing 22 lexical, inflectional, and clitic features for every word. The resource is constructed through a semi-automatic pipeline combining feature enrichment, morphological synchronization, and expert manual annotation and validation. We further evaluate Quranic morphosyntactic disambiguation under a realistic setting where the input may be represented in either the Uthmani or Imla'i script, and investigate cross-domain transfer between Quranic Arabic and MSA. Our results show that MSA and Quranic resources are complementary for Quranic disambiguation, with the best performance obtained by combining MSA training data, Quran-specific analyzers, and both script representations, whereas for MSA, only integrating Quranic resources provide modest gains. We make **MAQAM** publicly available.[1]

1 Introduction

The Quran is central to both Islamic studies and Arabic linguistics and serves as an important benchmark for Arabic natural language processing (NLP). However, Quranic Arabic differs substantially from Modern Standard Arabic (MSA) and contemporary dialects in its morphology, syntax, and orthography, limiting the effectiveness of NLP tools developed primarily for MSA.

While Arabic NLP has primarily focused on MSA and, more recently, Arabic dialects, Classical Arabic has received comparatively less attention. Moreover, research on Quranic Arabic and MSA has largely progressed independently, with limited investigation of how resources and models transfer between the two varieties. A better understanding of their linguistic and computational relationship can facilitate the transfer of linguistic resources and NLP methodologies across both domains. This work investigates that relationship through morphological and morphosyntactic analysis, assessing the transferability of resources and tools between Quranic Arabic and MSA.

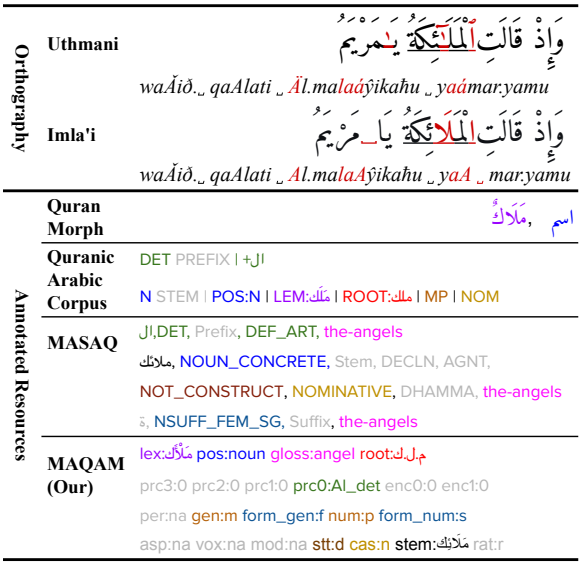


Figure 1: Annotation comparison for الملائكة *AlmlAyk*ħ 'when the angels said, O Mary! (3:42)'.

Over the years, numerous studies have explored Quranic NLP applications, including question answering (Saadaoui et al., 2024), topic modeling (Zafar et al., 2024), Lemmatization (Saeed and Habash, 2025) and diacritization (Cherradi and El Mahajer, 2025). Complementing these downstream applications, several Quranic NLP shared tasks have emerged to address a variety of linguistic and semantic challenges (Bouchekif et al., 2025; Mubarak et al., 2025; El Kheir et al., 2025; Basem et al., 2024; Malhas et al., 2023, 2022). These efforts underscore the need for comprehensive and richly annotated Quranic linguistic resources.

[1] github.com/CAMeL-Lab/MAQAM-Quranic-Corpus.

Proceedings of The Fourth Arabic Natural Language Processing Conference (ArabicNLP 2026), pages 151–163
October 28-29, 2026 ©2026 Association for Computational Linguistics

Figure 1 compares the annotation of a Quranic word across existing resources and the Uthmani and Imla'i scripts. The Quranic Arabic Corpus (QAC) (Dukes and Habash, 2010) and QuranMorph (Akra et al., 2025) use Uthmani, whereas MASAQ (Sawalha et al., 2025) uses Imla'i. Red highlights mark orthographic differences, while other colors indicate shared features and gray marks resource-specific features. The example reveals substantial variation in segmentation, annotation granularity, and feature coverage, often requiring information from multiple morphemes or fields to obtain a complete representation, with some annotations unavailable throughout the Quran.

In contrast, our proposed corpus consolidates all lexical, inflectional, and morphosyntactic information for each word into a single, consistently annotated record, eliminating the need to reconstruct annotations from multiple morphemes while providing richer and more comprehensive feature coverage. Furthermore, while modern Arabic NLP disambiguation tools, such as the CAMeL Tools disambiguator (Obeid et al., 2020; Inoue et al., 2022), perform well on MSA, they are not specifically designed for Quranic Arabic. We therefore use the CAMeL Tools disambiguator as a strong baseline and develop enhancements tailored to the linguistic characteristics of the Quran.

Our main contributions are as follows:

- **A New Quranic Morphological Resource.** We present **MAQAM** (**M**orphosyntactic **A**rabic **Q**uranic **A**nalysis and **M**odeling Resource) a richly annotated morphosyntactic resource for the entire Quran. We make this resource publicly available to support Arabic NLP research.[1,2]
- **Quranic Morphosyntactic Disambiguation.** To the best of our knowledge, we present the first benchmark for Quranic morphosyntactic disambiguation, under a realistic setting supporting both Uthmani and Imla'i scripts.
- **Cross-Domain Evaluation.** To the best of our knowledge, we present the first cross-domain evaluation of morphosyntactic disambiguation between Quranic Arabic and MSA.

The remainder of this paper presents the linguistic background and related work, describes the proposed resource and methodology, and reports experimental results and error analysis.

[2]The used Quran recitation version is Hafs 'an 'Asim.

2 Linguistic Background & Challenges

Arabic comprises several varieties, including Classical Arabic, MSA, and regional dialects. Differences between Classical Arabic and MSA in vocabulary, morphology, syntax, and orthography limit the transferability of MSA-based NLP tools to Quranic Arabic. Moreover, the Quran is represented in both *Uthmani* and *Imla'i* scripts, [3] with only 64.6% token overlap in their diacritized forms and 85.2% after removing diacritics, indicating that approximately 15% of the differences extend beyond diacritic variation.

From a morphosyntactic perspective, the differences between Quranic Arabic and MSA can be broadly categorized into orthographic, lexical, and inflectional variation. Orthographic variation concerns differences in surface word forms and spelling conventions. For example, the Uthmani form ٱمْرَأَتُ عِمْرَٰنَ *Äm.raÂatu ς im.raána*[4] 'the wife of Imran (3:35)' corresponds to the Imla'i form ٱمْرَأَتُ عِمْرَانَ *Äm.raÂatu ς im.raAna*, while MSA typically uses ٱِمْرَأَةُ عِمْرَانَ *Äm.raÂaħu ς im.raAna*. Such variation complicates lexicon lookup, morphological analysis, and resource alignment across Arabic varieties. Lexical variation is also evident in the Quranic lemma inventory. As will be illustrated later, manual word-by-word lemma annotation by a linguistic expert revealed 343 Quranic lemmas that are not covered by the MSA morphological database used in this study, as discussed later in the paper. These out-of-vocabulary lemmas span several categories, including proper nouns such as هَارُوتَ *haAruwta* 'Harut (2:102)', Quranic disjoint letters, abbreviations such as طسم *Tsm* (26:1), and rare lexical items and verbal nouns such as تَتْبِيبٍ *tat.biybĩ* 'ruin (11:101)'.

Inflectional variation further distinguishes Quranic Arabic from MSA. Quranic Arabic frequently employs emphatic constructions that are less common in contemporary MSA usage, such as the co-occurrence of *inna* and *emphatic lam*, as in إِنَّ فِي ذَٰلِكَ لَذِكْرَىٰ *Ăin a fiý ðaálika laðik.raýá* 'Surely in this is a reminder (39:21)'. Similarly, the emphatic nun (energetic mood) occurs in forms such as لَيَكُونَنَّ *layakuwnan∼a* 'surely, they would be (35:42)' , while MSA typically

[3]Data source: https://tanzil.net.

[4]Arabic in HSB Romanization (Habash et al., 2007).

avoids its use. Additional variation appears in the realization of clitics. For example, the final pronominal clitic is omitted in the word لِيَعْبُدُونِ *liyaς.buduwni* 'that they worship Me' in the verse وَمَا خَلَقْتُ الْجِنَّ وَالْإِنسَ إِلَّا لِيَعْبُدُونِ *wamaA xalaq.tu Äl.jin∼a waÄl.Ăinsa Ăil aA liyaς.buduwni* 'I did not create jinn and humans except to worship Me. (51:56)', whereas the corresponding MSA form would normally be لِيَعْبُدُونِي *liyaς.buduwniy*.

These orthographic, lexical, and inflectional differences collectively distinguish Quranic Arabic from MSA and contribute to the limited transferability of existing MSA-oriented NLP resources and models. As a result, morphological analyzers, disambiguation systems, and language models trained primarily on MSA often struggle to accurately process Quranic text, highlighting the need for dedicated Quranic Arabic resources and adaptation techniques.

3 Related Work

As previously mentioned, several efforts have annotated Quranic data at different linguistic levels, including morphology, syntax, and semantics (Dukes and Habash, 2010; Sawalha et al., 2025; Akra et al., 2025; Faqihi et al., 2026). However, these resources differ in their annotation schemes, objectives, and coverage, and some exhibit inconsistencies in annotation granularity across the Quran. These differences are discussed below.

The **Quranic Arabic Corpus (QAC)** is one of the earliest and most influential morphologically annotated resources for Classical Arabic. It was developed through a combination of automatic analysis using an adapted Buckwalter Arabic Morphological Analyzer (BAMA) and extensive manual refinement (Buckwalter, 2004) and the features are annotated at the morpheme level rather than the token level. While the corpus introduces rich morphological features that make it a foundational resource for Quranic Arabic research, it suffers from inconsistencies in feature assignment, where some words are annotated with more or fewer features than others.

The **QuranMorph Corpus** (Akra et al., 2025) is an annotated Quranic dataset providing lemma and part-of-speech annotations for each token by three linguists using the Tawseem Web Annotation tool. Lemmas are sourced from the Qabas lexicographic database (Jarrar and Hammouda, 2024),

	MAQAM	QAC	QM	MASAQ	QAMAR
Orthographic Style					
Uthmani	✓	✓	✓		✓
Imla'i	✓			✓	✓
Lexical Features					
Gloss	✓			✓	
Lex	✓	✓	✓		✓
Rationality	✓				
Root	✓	✓			✓
Stem	✓	✓		✓	✓
Morphological Features					
Aspect	✓	✓			
Case	✓			✓	
Gender	✓	✓			
Number	✓	✓			
Mood	✓			✓	
Person	✓				
POS	✓	✓	✓	✓	✓
State	✓			✓	
Voice	✓	✓			
Clitics	✓	✓		✓	

Table 1: Comparison of available scripts and morphological features across Quranic Arabic resources. QM (QuranMorph), QAC (Quranic Arabic Corpus).

which integrates 110 lexicons and corpora totaling over two million tokens, and POS tags follow the 40-category SAMA/Qabas tagset (Maamouri et al., 2010b; Jarrar and Hammouda, 2024). However, the corpus does not include morphological, syntactic, or semantic annotations.

The **MASAQ Dataset** (Sawalha et al., 2025) is a morphosyntactically annotated corpus covering the entire Quran. Built on the verified Tanzil text, it provides detailed morphological and syntactic annotations for each word on the morpheme level not the token level, produced and validated by expert linguists. The resulting resource offers rich grammatical information of Quranic Arabic but still missing other morphosyntactic features.

The **Quranic Arabic Morphological Analysis Resource (QAMAR)** corpus (Faqihi et al., 2026) is a manually verified morphological resource covering the entire Quran in both Uthmani and Imla'i forms. Developed using SAFAR-generated analyses followed by expert validation, it provides stems, lemmas, roots, and POS annotations while preserving Quranic orthography and providing normalized MSA representations.

As shown in Table 1, current Quranic resources

Orthography	Imla'i	الصَّالِحَاتِ	وَعَمِلُوا	آمَنُوا	وَالَّذِينَ
		AlS~aAliHaAti	waçamiluwA	ĀmanuwA	waAl~aðiyna
	Uthmani	ٱلصَّٰلِحَٰتِ	وَعَمِلُواْ	ءَامَنُواْ	وَالَّذِينَ
		ĀlS~aáliHaáti	waçamiluwA`	'aAmanuwA`	waAl~aðiyna
Lexical Features	gloss	good deed	work	believe	whom
	lex	صَالِحَة	عَمِل	آمَن	الَّذِي
		SaAliHah	çamil	Āman	Ala~ðiy
	rat	i	na	na	na
	root	ص.ل.ح	ع.م.ل	أ.م.ن	NTWS
		S.l.H	ç.m.l	Â.m.n	NTWS
	stem	صَالِح	عَمِل	آمَن	الَّذَيْنِ
		SaAliH	çamil	Āman	Ala~ðay.ni
Inflectional Features	asp	na	p	p	na
	cas	a	na	na	n
	form_gen	f	m	m	m
	form_num	p	p	p	s
	gen	f	m	m	m
	num	p	p	p	p
	mod	na	i	i	na
	per	na	3	3	na
	pos	noun	verb	verb	pron_rel
	stt	d	na	na	d
	vox	na	a	a	na
Clitic Features	prc3	0	0	0	0
	prc2	0	wa_conj	0	wa_conj
	prc1	0	0	0	0
	prc0	Al_det	0	0	0
	enc0	0	0	0	0
	enc1	0	0	0	0

Figure 2: Sample entries from MAQAM corpus with their associated lexical, inflectional, and clitic features: 'And those who believe and do good deeds (2:82).'

do not provide complete morphosyntactic annotation of every word in the Quran, with several linguistic features either missing, inconsistently represented, or unavailable. To fill this gap, we introduce **MAQAM**, a comprehensive Quranic corpus providing a unified set of morphosyntactic annotations for the whole Quran, supporting both fine-grained linguistic analysis and Quranic Arabic NLP.

4 The MAQAM Corpus Design

4.1 Word Entries

Each entry in MAQAM corresponds to a single Quranic token and includes both *Uthmani* and *Imla'i* representations together with unified lexical, inflectional, and clitic morphosyntactic features. Figure 2 summarizes the annotation schema.

Dataset	# Surahs	# Verses	# Words
Q-All	114	6,236	77,429
Q-Train	101	5,605	71,801
Q-Test	13	631	5,628

Table 2: Corpus splits for MAQAM.

4.2 Corpus Split

The corpus is organized into three splits: **Q-All**, **Q-Train**, and **Q-Test**. Table 2 summarizes their statistics in terms of surahs, verses, and words. Q-All covers the entire Quran and is partitioned into Q-Train and Q-Test. Q-Test comprises the 13 surahs located in the middle of the Quran, spanning Surahs 45–57, while Q-Train contains the remaining 101 surahs. All words in Q-All are manually annotated for lemma, part of speech, and gloss. Q-Train inherits these lexical annotations, while Q-Test is additionally fully manually validated for all morphosyntactic features for each word in it. We use the subscripted versions Q-Train$_i$ and Q-Train$_u$ to refer to the Imla'i vs Uthmani sub-corpus, respectively. When no subscript is used, both versions are included. The annotation and validation procedure is described in Section 5.3.

5 Corpus Construction Pipeline

Our corpus construction approach relies on four resources that are integrated through a synchronization process that integrates their complementary content. We present each of these resources next followed by the synchronization step.

5.1 Imla'i-Uthmani Spelling Alignment

We obtain the *Imla'i* and *Uthmani* Quranic scripts from the Tanzil website. Although both represent the same Quranic text, in some cases they differ in tokenization where a single token in one script corresponds to multiple tokens in the other. We therefore perform a token-level alignment between the two representations using the Character Edit Distance (CED) word alignment tool (Khalifa et al.), which is later used to map the morphosyntactic annotations from the QAC to the corresponding tokens. The final alignment achieves **100% accuracy**, ensuring an exact mapping between the tokens and their corresponding analyses for the whole Quran.

5.2 QAC Feature Extraction

After obtaining the alignment between the *Imla'i* and *Uthmani* representations, we extract the corresponding annotations from the QAC corpus for each aligned word. Since QAC follows a different annotation scheme, we convert its analyses into unified feature-value pairs and then enrich them using an extended version of the Arabic Extended Morphological Analysis and Disambiguation Tagset system (EMAD) (Kallas et al., 2024), a bridge tagset for unifying Arabic morphological annotation schemes. Our extension to EMAD adds support for enclitics, orthographic gender and number, emphatic mood, and enhanced feature mappings. The pipeline preprocesses QAC annotations, converts them into the EMAD representation, and applies consistency checks to remove invalid analyses. Due to morphological ambiguity, a word may produce multiple valid feature-value analyses, which are resolved during synchronization.

5.3 Manual Annotation

As mentioned earlier, the QAC has several limitations for our corpus construction: lemmas are represented in *Uthmani* orthography, some words lack lemma annotations, glosses are not provided, and POS tags are often underspecified (e.g., `N` and `ADJ` correspond to multiple fine-grained categories). To address these limitations, every Quranic word was manually annotated with its lemma, fine-grained POS, English gloss, root, and rationality (Alkuhlani and Habash, 2012; Zaina et al., 2026). The rest of the morphosyntactic features were automatically extended (see Section 5.5). However, every morphosyntactic feature in the Q-Test split was further manually validated,[5] producing a high-quality benchmark later used to evaluate downstream morphosyntactic disambiguation.

Our annotation team consisted of two expert native Arabic speakers with extensive experience in Arabic computational linguistics annotation: an Egyptian woman with a BA in English Language and Literature and working on a MS in Linguistics, and a Lebanese man with a BA and an MS in Philosophy. The former completed the lemma, POS, and gloss annotations for Q-All, while both annotators contributed to the rationality and root annotations, and the full Q-Test split validation. For inter-annotator agreement (IAA) on the 14 morphosyntactic features,[6] we used two Q-Test surahs comprising 702 tokens and 9,828 annotation cells (702×14). The IAA was **99.1%** for Surah 51 and **98.7%** for Surah 54, with an average IAA of **98.9%**. These manual feature annotations were then used to guide the automatic synchronization of the remaining morphosyntactic features.

The annotation process was time-intensive and required considerable care and research, including consultation of linguistic references and Quranic resources. Quranic Arabic has distinctive lexical, morphological, and syntactic properties, and its interpretation requires specialized, context-sensitive analysis informed by knowledge of Classical Arabic vocabulary and Quran-specific references. The two annotators and the lead authors met regularly to discuss and resolve manual annotation decisions. Nevertheless, we recognize that some decisions may admit alternative interpretations, including interpretations that we did not anticipate. We acknowledge this as an inherent limitation of annotating Quranic Arabic and welcome future extensions and corrections from the broader research community.

Level	Script	R13	S31	CM
Tokens	**Uthmani**	5.8	5.4	**3.2**
	Imla'i	2.1	1.8	**0.8**
Types	**Uthmani**	10.8	9.9	**5.9**
	Imla'i	6.6	5.7	**2.7**

Table 3: MAQAM OOV rates (%) under different orthographic representations and morphological databases.

5.4 Arabic Morphological Analyzer

The upcoming synchronization process relies on a morphological analyzer to generate candidate analyses for each Quranic word. We compare three MSA morphological databases: Calima-MSA-R13 (R13) (Buckwalter, 2002), Calima-MSA-S31 (S31) (Maamouri et al., 2010a; Taji et al., 2018), and CAMeL Morph (CM) (Khairallah et al., 2024).

Table 3 reports the out-of-vocabulary (OOV) rates of the three databases under both the *Uthmani* and *Imla'i* script representations. Among them, CAMeL Morph consistently achieves the lowest OOV rates at both the token and type levels. We therefore adopt CAMeL Morph with the *Imla'i* representation in the synchronization process.

[5] Validated Features: *lex*, *pos*, *prc3*, *prc2*, *prc1*, *prc0*, *per*, *asp*, *vox*, *mod*, *gen*, *num*, *stt*, *cas*, *enc0*, and *enc1*.

[6] We exclude lemma and POS from IAA because the annotators jointly validated them; and we exclude the form_gen and form_num because they are deterministic.

5.5 Synchronization

Finally, the synchronization process integrates the resources described above to produce a unified morphosyntactic annotation for every Quranic word. On one side, we use the enriched QAC annotations together with the manually annotated lemma, POS, and gloss. On the other side, we generate candidate analyses using CAMeL Morph using the CAMeL Tools disambiguation framework.

Synchronization is formulated as selecting the CAMeL Morph analysis that best matches one of the enriched Quranic annotations. Each candidate analysis is compared against the enriched annotation using the shared features *diac*, *lex*, *POS*, *per*, *asp*, *vox*, *mod*, *gen*, *num*, *form_gen*, *form_num*, *stt*, *cas*, *prc0–prc3* and *enc0–enc1*. A matching score is computed based on feature agreement, and the highest-scoring analysis is selected as the synchronized annotation. When multiple analyses obtain the same highest matching score, ties are resolved using the CAMeL Tools disambiguator, which reranks the analyses generated by CAMeL Morph according to how well they match the predicted features.

The synchronization procedure was iteratively refined using the manually validated Q-Test benchmark, allowing synchronization errors to be identified and corrected. The full process achieves an average accuracy of 98.3% on Q-Test. A detailed breakdown by morphosyntactic feature and vocabulary status (all, INV, and OOV words) is provided in Appendix B. This evaluation covers the features produced through automatic synchronization while *lex*, *POS*, *gloss*, *rat*, and *root* are excluded because they were manually annotated across the entire corpus and not automatically generated.

6 Morphosyntactic Disambiguation

Following the construction of the synchronized data, we investigate Quranic morphosyntactic disambiguation with the CAMeL Tools unfactored MSA disambiguator as a baseline. We examine the relationship between MSA and Quranic Arabic by varying the training data and morphological analyzer and evaluating bidirectional transfer. For MSA, we use the Penn Arabic Treebank (PATB) (Maamouri et al., 2004), on which the baseline is trained, while for Quranic Arabic, we use MAQAM. This assesses whether Quran-specific resources improve Quranic disambiguation while preserving MSA performance.

The unfactored disambiguator formulates morphosyntactic analysis as a multiclass classification task, where each class represents a complete morphosyntactic analysis set defined by 14 features: *POS, prc0-prc3, per, asp, vox, mod, form_gen, form_num, stt, cas,* and *enc0*. To illustrate the diversity of morphosyntactic analyses covered by each training configuration, we report the number of unique class labels in each dataset. Q-Train contains 1,939 classes, Q-All 1,982, Q-Test 695, and PATB 2,016. Combining Quranic and PATB data increases the unique label count to 3,213 classes for PATB+Q-Train and 3,269 classes for PATB+Q-All.

During inference, the model predicts the most likely class for each word in context, and candidate analyses from a morphological analyzer are ranked according to their agreement with the predicted class. The highest-ranked analysis in the top ranked set is selected, with ties resolved using *POS_lex_logprob* followed by *lex_logprob*.

Accordingly, the proposed morphological disambiguation framework relies on three key elements:

Disambiguator Training Data We fine-tune the disambiguator using the PATB training set, representing Modern Standard Arabic (MSA); the Quranic training split (Q-Train); the Quranic dataset (Q-All); and the datasets PATB + Q-Train and PATB + Q-All. For configurations containing Quranic data, we train using the Imla'i script, the Uthmani script, or both. These configurations allow us to examine the contributions of MSA and Quranic Arabic data and the effect of script representation on disambiguation performance.

Morphological Analyzer Analyses come from three sources: (1) MSA databases (R13, S31, and CAMeL Morph), (2) Quran-specific analyzers derived from Q-Train and Q-All, and (3) The combination of MSA and Quranic analyzers. Each Quran-specific analyzer is a lookup table mapping an undiacritized word to corpus-observed analyses, constructed from the *Imla'i* representation, the *Uthmani* representation, or both. When combined, in-domain analyses are prioritized: Quran-specific analyses for Quranic evaluation and MSA analyses for MSA evaluation.

Evaluation Corpus Models are evaluated on Q-Test using both Imla'i and Uthmani scripts as input and the PATB test set to measure in-domain performance and cross-domain transfer between Quranic Arabic and MSA.

	Training Data	Analyzer Database	Lexical Exact	Inflectional Macro Avg	Inflectional Exact	Clitics Macro Avg	Clitics Exact	All Macro Avg	All Exact
(a)	PATB	R-13	74.0	89.7	51.5	98.7	94.0	91.8	44.7
	PATB	S31	78.1	92.3	61.0	99.0	95.4	93.7	54.0
	PATB	CM	82.7	93.6	68.5	98.8	93.8	94.7	61.3
(b)	Q-Train	CM	85.7	95.6	75.4	99.2	95.7	96.2	67.1
	Q-Train	Q-Train	82.8	90.6	67.0	98.9	93.9	92.9	64.7
	Q-Train	CM+Q-Train	92.3	95.9	76.8	99.2	96.0	96.8	72.9
(c)	PATB+Q-Train	CM	86.0	96.1	76.6	99.3	95.9	96.6	68.1
	PATB+Q-Train	Q-Train	82.9	91.0	67.4	98.9	94.0	93.2	64.6
	PATB+Q-Train	CM+Q-Train	**92.7**	**96.5**	**78.0**	**99.3**	**96.2**	**97.2**	**73.9**
(d)	PATB+Q-All	CM+Q-All	**98.7**	**97.9**	**84.4**	**99.3**	**96.4**	**98.4**	**81.0**
(e)	PATB+Q-Train$_i$	CM+Q-Train$_i$	92.4	96.4	77.7	**99.3**	**96.2**	**97.2**	73.6

Table 4: Morphosyntactic tagging performance on the **Imla'i Q-Test**$_i$ set across different fine-tuning and morphological analyzer configurations.

	Training Data	Analyzer Database	Lexical Exact	Inflectional Macro Avg	Inflectional Exact	Clitics Macro Avg	Clitics Exact	All Macro Avg	All Exact
(a)	PATB	R-13	67.0	85.0	46.8	98.2	91.6	88.4	40.3
	PATB	S31	71.3	88.6	55.6	98.4	92.9	90.9	49.0
	PATB	CM	75.5	90.1	62.6	98.1	91.1	91.9	55.8
(b)	Q-Train	CM	78.5	92.3	69.2	98.6	93.7	93.7	60.8
	Q-Train	Q-Train	82.6	90.5	66.8	98.8	93.7	92.8	64.4
	Q-Train	CM+Q-Train	91.0	95.4	75.8	99.1	95.5	96.4	71.6
(c)	PATB+Q-Train	CM	78.7	92.8	70.2	98.7	93.7	94.0	61.7
	PATB+Q-Train	Q-Train	82.7	90.9	67.2	98.9	93.8	93.1	64.3
	PATB+Q-Train	CM+Q-Train	**91.3**	**95.9**	**76.7**	**99.2**	**95.7**	**96.8**	**72.5**
(d)	PATB+Q-All	CM+Q-All	**98.5**	**97.8**	**84.3**	**99.3**	**96.4**	**98.4**	**80.8**
(e)	PATB+Q-Train$_u$	CM+Q-Train$_u$	91.1	95.8	76.6	**99.2**	**95.7**	96.7	72.3

Table 5: Morphosyntactic tagging performance on the **Uthmani Q-Test**$_u$ set across different fine-tuning and morphological analyzer configurations.

7 Experiments and Results

7.1 Experimental Setup

Fine-tuning Setup Following the fine-tuning setup of (Inoue et al., 2022), we fine-tuned `CAMeL-Lab/bert-base-arabic-camelbert-msa` as a token-classification model on a single NVIDIA A100 GPU. Each word was assigned a composite label representing 14 morphosyntactic features, with supervision applied only to its first subword. Models were trained for 10 epochs using a batch size of 32, a learning rate of 5×10^{-5}, a maximum sequence length of 512 tokens, and a fixed seed of 12,345. The best checkpoint was selected based on validation accuracy evaluated every 500 training steps.

Evaluation Metrics Performance was evaluated using macro-average per-feature accuracy and exact-match (EM) accuracy across four feature groups: **Lexical** (*lex*), **Inflectional** (*asp, cas, form_gen, num, gen, mod, form_num, per, POS, stt, vox*), **Clitic** (*prc0–prc3, enc0, enc1*), and **Overall** (all 18 features). EM considers a prediction correct only when all features within the corresponding group match the gold analysis.

Disambiguation Configuration Predicted composite labels were used to rank the analyses generated by the morphological analyzer. All systems used the **NOAN_PROP** backoff strategy, which assigns a proper-noun analysis to words that are OOV with respect to the analyzer.

7.2 Results

Tables 4 and 5 present morphosyntactic tagging performance on the Imla'i and Uthmani Q-Test sets under different combinations of disambiguator training data and morphological analysis resources. All Quranic training data and Quran-specific analyzers

	Training Data	Analyzer Database	Lexical	Inflectional		Clitics		All	
			Exact	Macro Avg.	Exact	Macro Avg.	Exact	Macro Avg.	Exact
(a)	PATB	CM	93.5	**98.5**	**90.4**	**99.8**	**99.2**	**98.7**	**86.3**
	PATB	Q-All	40.9	85.3	37.8	99.6	97.5	87.6	32.6
	PATB	CM+Q-All	**93.6**	**98.5**	**90.4**	**99.8**	**99.2**	**98.7**	**86.3**
(b)	PATB+Q-All	CM	93.5	**98.5**	90.3	**99.8**	**99.2**	**98.7**	**86.3**
	PATB+Q-All	Q-All	40.8	85.3	37.8	99.6	97.5	87.6	32.6
	PATB+Q-All	CM+Q-All	**93.6**	**98.5**	**90.4**	**99.8**	**99.2**	**98.7**	**86.3**

Table 6: Morphosyntactic tagging performance on the PATB test set (63,172 words). Results are reported as percentages (%).

in Groups (a)–(d) are constructed from the combined Imla'i and Uthmani representations. Group (e) instead uses only the script corresponding to the evaluation set, allowing us to assess the benefit of dual-script training.

Groups (b) and (c) investigate the contribution of Quranic resources. In group (b), the disambiguator is trained only on Quranic data, with candidate analyses from the CM analyzer, the Q-Train analyzer, or both. CM outperforms Q-Train on all feature groups except lexical prediction, while combining both consistently yields the best performance, demonstrating the complementary strengths of MSA and Quran-specific resources.

Group (c) further combines MSA and Quranic data during disambiguator training while retaining the combined analyzer. This configuration achieves the best non-oracle performance across all feature groups, improving overall macro accuracy and exact match accuracy compared with training only on Quranic data. These results show that MSA and Quranic Arabic are complementary: MSA training data improve contextual disambiguation, while Quran-specific lexical resources improve coverage of Quranic vocabulary.

Finally, groups (d) and (e) provide two reference settings for interpreting the results. Group (d) represents an oracle experiment in which both the disambiguator and the Quran-specific analyzer are constructed using Q-All. Because Q-All includes Q-Test, this setting is not intended as a realistic deployment scenario; rather, it provides an approximate upper bound on the performance achievable when the test data are covered during training. Group (e), in contrast, uses the best-performing non-oracle configuration but is trained only on the Imla'i representation. Compared with the dual-script configuration in group (c), this model performs slightly worse on some feature groups, although the differences are generally small but still, this difference demonstrates the benefit of jointly supporting both the Imla'i and Uthmani input representations during training. In practice, we therefore recommend the non-oracle dual-script configuration in group (c), since it avoids test-set leakage while allowing newly annotated Quranic data to be incorporated into future versions of the training set.

Table 5 presents the results for the Uthmani representation. The same trends observed for Imla'i hold, although performance is lower. This is likely because Uthmani orthography differs considerably from MSA conventions, limiting the transferability of MSA-trained resources and highlighting the need for Quran-specific data when processing this script.

Table 6 reports the evaluation on the PATB test set. Unlike the Quranic evaluation, adding Quranic training data results in nearly identical MSA performance, with a marginal improvement of 0.1% for lexical feature EM accuracy. All other results remain unchanged, indicating that incorporating Quranic data neither improves nor degrades MSA disambiguation performance, consistent with the nearly identical results reported in the evaluation table. This may be because PATB represents the newswire genre, whereas Quranic resources primarily benefit Classical Arabic in the religious genre.

We assess significance using McNemar's test (McNemar, 1947) on paired word-level predictions under the overall exact-match metric. We compare the CAMeL Tools morphological-disambiguation baseline with the best-performing non-oracle system in Group (c). The improvements on the Imla'i and Uthmani Q-Test sets are statistically significant ($p < 0.05$). In contrast, the difference between the baseline and the best-performing system on the PATB test set is not statistically significant ($p \geq 0.05$), consistent with the nearly identical performance reported in the evaluation table.

Word	Analysis	Imla'i	Uthmani
OOV	**OOV**	2.7	3.7
INV	**OOV**	46.4	40.5
INV	**INV**	50.9	55.8

Table 7: Distribution of analyzer and disambiguator errors (%) comparing in-vocabulary and out-of-vocabulary patterns.

8 Error Analysis

To better understand the remaining errors, we analyze incorrectly predicted instances under the All Tags exact-match (EM) metric using the best-performing non-oracle system.

Table 7 summarizes this breakdown. For the Imla'i representation, only 2.7% of errors are due to out-of-vocabulary (OOV) words not covered by the analyzer, while 46.4% occur because the correct analysis is absent from the analyzer's candidate set despite the word being in vocabulary. The remaining 50.9% are true disambiguation errors, where the correct analysis is generated by the analyzer but the disambiguator selects a different candidate. Similar trends are observed for Uthmani, with 3.7%, 40.5%, and 55.8% of errors falling into these three categories, respectively.

We also compare the predictions produced by the Imla'i and Uthmani representations under the All Tags EM metric using the best performing system. The two systems produce the same prediction for the vast majority of words, correctly analyzing 72.1% of the test set and jointly failing on 25.9%. However, the Imla'i representation correctly predicts an additional 1.8% of words that Uthmani misses, whereas Uthmani uniquely predicts only 0.3%. This further demonstrates the consistent advantage of the Imla'i representation for morphosyntactic disambiguation.

The remaining errors are concentrated in a small number of morphosyntactic features rather than being distributed uniformly across the complete feature set. Among the inflectional features, *case* is the largest source of difficulty, accounting for more than a quarter of all errors (27.3%). This is followed by *gender* (12.9%), *state* (12.8%), *form_gen* (11.2%), and *POS* (11.1%), whereas every other inflectional feature individually contributes less than 6%. This distribution suggests that future improvements should primarily target features whose correct values depend strongly on syntactic context or distinctions between closely related analyses. Clitic errors exhibit an even stronger concentration: *prc2* (41.5%) and *prc1* (33.9%) jointly account for more than three quarters of all clitic errors. By comparison, enclitic errors are less frequent, representing 17.8% for *enc0* and less than 1% for *enc1*. Overall, these findings identify case assignment and proclitic disambiguation as the most important targets for reducing the remaining exact-match errors.

9 Conclusion & Future Work

In this paper, we introduced MAQAM, a comprehensive morphosyntactic resource providing unified token-level annotations for every Quranic word in both the Imla'i and Uthmani scripts. We also presented a synchronization framework and established a benchmark for Quranic morphosyntactic disambiguation. Our experiments demonstrate that MSA and Quranic resources are complementary: MSA data improve contextual disambiguation, while Quran-specific analyzers increase the coverage of valid Quranic analyses. The best non-oracle system significantly outperforms the CAMeL Tools baseline on both script representations, with Imla'i consistently performing better than Uthmani, while the differences on the PATB test set are not statistically significant which may be due to the domain difference between the news-oriented PATB and the newly introduced Quranic corpus.

We will publicly release the MAQAM corpus, synchronization pipeline, trained models, and evaluation code to support further research on Quranic and Classical Arabic NLP. Future work will focus on improving candidate generation and contextual disambiguation, particularly when the correct analysis is missing from the analyzer's candidate set. We also plan to investigate productive Quranic morphological analyzers that can generalize beyond observed corpus forms, as well as factored and multi-task models that predict individual morphosyntactic features and reduce label sparsity. Additional directions include improving the handling of Uthmani orthography, integrating recent Quranic lemmatization models (Saeed and Habash, 2025) into the pipeline to improve lexical prediction and overall morphosyntactic analysis, extending the evaluation to other Classical Arabic and religious-text genres, and assessing MAQAM's usefulness for downstream tasks such as diacritization.

Limitations

A key limitation of the current system is its reliance on analyzer-generated candidate analyses. When the correct analysis is absent from the candidate set, the disambiguator cannot recover it, regardless of the quality of the tagging model. Addressing this limitation will require more effective backoff or candidate generation strategies that can produce plausible analyses for unseen or missing forms. In addition, the Quran-specific analyzers used in this work are implemented as lookup tables over observed corpus forms, limiting their ability to generalize to unseen Classical Arabic words or generate productive morphological analyses beyond the Quran. Finally, the unfactored disambiguator models each complete morphosyntactic analysis as a single class label, resulting in a large and sparse label space that may hinder generalization to rare or unseen feature combinations.

Ethics Statement

All manual annotations used to construct the resource were performed by qualified annotators who were compensated at fair market rates for their work.

Because the Quran is the sacred scripture of over 2 billion Muslims, we approached its annotation and processing with care and respect for its religious significance. Throughout the development of this resource, we sought to faithfully preserve the original text while employing a rigorous and transparent scientific methodology. Although the resource underwent extensive automatic validation and expert manual review, some errors may nevertheless remain. Any such errors are solely our responsibility, and we welcome feedback and corrections from the research community to further improve the resource.

The annotations represent linguistic decisions made for research purposes and should not be treated as authoritative interpretations of the Quran. In particular, they should not be used as the basis for *tafsir* or for making religious or interpretive claims about the Quran. The resource is intended to support Quranic NLP research, facilitate downstream tasks, and improve the availability of Quranic resources with detailed morphosyntactic annotations.

As with many natural language processing tools, the morphosyntactic analyses and models presented in this work could potentially be misused. We encourage researchers and practitioners to use these resources only for ethical purposes and in accordance with applicable laws and ethical guidelines.

Large language models and AI-assisted writing tools were used solely to improve the clarity, grammar, and readability of the manuscript. All technical content, experimental design, analyses, and conclusions were developed and verified by the authors.

Acknowledgments

This research was conducted using the High Performance Computing resources at New York University Abu Dhabi (NYUAD). We gratefully acknowledge the Center of Interdisciplinary Data Science and AI (CIDSAI) at NYUAD for its generous support.

References

Diyam Akra, Tymaa Hammouda, and Mustafa Jarrar. 2025. Quranmorph: Morphologically annotated quranic corpus. *arXiv preprint arXiv:2506.18148*.

Sarah Alkuhlani and Nizar Habash. 2012. Identifying broken plurals, irregular gender, and rationality in Arabic text. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 675–685, Avignon, France. Association for Computational Linguistics.

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. *arXiv e-prints*, arXiv:2412.11431.

Abdessalam Bouchekif, Samer Rashwani, Emad Soliman Ali Mohamed, Mutaz Alkhatib, Heba Sbahi, Shahd Gaben, Wajdi Zaghouani, Aiman Erbad, and Mohammed Ghaly. 2025. QIAS 2025: Overview of the shared task on islamic inheritance reasoning and knowledge assessment. In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 851–860, Suzhou, China. Association for Computational Linguistics.

Tim Buckwalter. 2002. Buckwalter Arabic morphological analyzer version 1.0. Linguistic Data Consortium (LDC) catalog number LDC2002L49, ISBN 1-58563-257-0.

Tim Buckwalter. 2004. Buckwalter arabic morphological analyzer version 2.0. *(No Title)*.

Mohamed Cherradi and Hajar El Mahajer. 2025. Arabic text diacritization using deep neural networks and transformer-based architectures. *Knowledge and Decision Systems with Applications*, 1:257–269.

Kais Dukes and Nizar Habash. 2010. Morphological Annotation of Quranic Arabic. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, Valetta, Malta.

Yassine El Kheir, Amit Meghanani, Hawau Olamide Toyin, Nada Almarwani, Omnia Ibrahim, Yousseif Ahmed 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: Shared Tasks*, pages 443–452, Suzhou, China. Association for Computational Linguistics.

Sara Faqihi, Karim Bouzoubaa, Rachida Tajmout, and Driss Namly. 2026. QAMAR: A new fully verified and accurate Quranic Arabic morphological analysis resource. In *Proceedings of the 2nd Workshop on NLP for Languages Using Arabic Script*, pages 301–312, Rabat, Morocco. Association for Computational Linguistics.

Nizar Habash, Abdelhadi Soudi, and Tim Buckwalter. 2007. On Arabic Transliteration. In A. van den Bosch and A. Soudi, editors, *Arabic Computational Morphology: Knowledge-based and Empirical Methods*, pages 15–22. Springer, Netherlands.

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.

Go Inoue, Salam Khalifa, and Nizar Habash. 2022. Morphosyntactic tagging with pre-trained language models for Arabic and its dialects. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1708–1719, Dublin, Ireland. Association for Computational Linguistics.

Mustafa Jarrar and Tymaa Hasanain Hammouda. 2024. Qabas: An open-source Arabic lexicographic database. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 13363–13370, Torino, Italia. ELRA and ICCL.

Omar Kallas, Go Inoue, and Nizar Habash. 2024. EMAD: A bridge tagset for unifying Arabic POS annotations. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 5637–5643, Torino, Italia. ELRA and ICCL.

Christian Khairallah, Salam Khalifa, Reham Marzouk, Mayar Nassar, and Nizar Habash. 2024. Camel morph MSA: A large-scale open-source morphological analyzer for Modern Standard Arabic. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2683–2691, Torino, Italia. ELRA and ICCL.

Salam Khalifa, Ossama Obeid, and Nizar Habash. Character Edit Distance Based Word Alignment.

Mohamed Maamouri, Ann Bies, Tim Buckwalter, and Wigdan Mekki. 2004. The Penn Arabic Treebank: Building a Large-Scale Annotated Arabic Corpus. In *Proceedings of the International Conference on Arabic Language Resources and Tools*, pages 102–109, Cairo, Egypt.

Mohamed Maamouri, Dave Graff, Basma Bouziri, Sondos Krouna, Ann Bies, and Seth Kulick. 2010a. Ldc standard Arabic morphological analyzer (sama) version 3.1.

Mohamed Maamouri, David Graff, Basma Bouziri, Sondos Krouna, Ann Bies, and Seth Kulick. 2010b. Ldc standard arabic morphological analyzer (sama) version 3.1. *(No Title)*.

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.

Quinn McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157.

Hamdy Mubarak, Rana Malhas, Watheq Mansour, Abubakr Mohamed, Mahmoud Fawzi, Majd Hawasly, Tamer Elsayed, Kareem Mohamed Darwish, and Walid Magdy. 2025. IslamicEval 2025: The first shared task of capturing LLMs hallucination in islamic content. In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 480–493, Suzhou, China. Association for Computational Linguistics.

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.

Zakia Saadaoui, Ghassen Tlig, and Fethi Jarray. 2024. Llms based approach for quranic question answering. In *WEBIST 2024*.

Mostafa Saeed and Nizar Habash. 2025. Lemmatization as a classification task: Results from Arabic across multiple genres. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 30014–30029, Suzhou, China. Association for Computational Linguistics.

Majdi Sawalha, Faisal Al-Shargi, Sane Yagi, Abdallah T. AlShdaifat, Bassam Hammo, Mariam Belajeed, and Lubna R. Al-Ogaili. 2025. Morphologically-analyzed and syntactically-annotated quran dataset. *Data in Brief*, 58:111211.

Dima Taji, Salam Khalifa, Ossama Obeid, Fadhl Eryani, and Nizar Habash. 2018. An Arabic morphological

analyzer and generator with copious features. In *Proceedings of the Fifteenth Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 140–150, Brussels, Belgium. Association for Computational Linguistics.

Tanzil Project. 2021. Tanzil quran text: Text license. Accessed: 2026-09-16.

Amna Zafar, Muhammad Wasim, Shaista Zulfiqar, Talha Waheed, and AbuBakar Siddique. 2024. Transformer-based topic modeling for urdu translations of the holy quran. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 23(10).

Bailasan Zaina, Salam Khalifa, and Nizar Habash. 2026. Beyond gender, number, and rationality: Modeling extended arabic morpho-syntactic agreement with aggregation. In *Proceedings of the Fourth Arabic Natural Language Processing Conference (Arabic-NLP 2026)*, Budapest, Hungary.

A License

The original annotations created by the MAQAM authors are released under the Creative Commons Attribution 4.0 International License (CC BY 4.0), and the accompanying code is released under the MIT License. In Table 8, we list the license of the data and tools used in this work. All of them are used under their intended use.

Data/tool	License
Arabic Treebank: Part 1 v 4.1 (LDC2010T13)	LDC User Agreement for Non-Members
Arabic Treebank: Part 2 v 3.1 (LDC2011T09)	LDC User Agreement for Non-Members
Arabic Treebank: Part 3 v 3.2 (LDC2010T08)	LDC User Agreement for Non-Members
CAMeL Tools (Obeid et al., 2020)	MIT License
CAMeLBERT (Inoue et al., 2021)	MIT License
Character Edit Distance (CED) word alignment tool (Khalifa et al.)	MIT License.
CamelMorph (Khairallah et al., 2024)	MIT License
Tanzil Quran Text (Tanzil Project, 2021)	Creative Commons Attribution 3.0 (CC BY 3.0).
MASAQ Data (Sawalha et al., 2025)	Creative Commons Attribution 4.0 license
Quranic Arabic Corpus (Dukes and Habash, 2010)	GNU General Public License
QuranMorph (Akra et al., 2025)	Creative Commons Attribution 4.0 license
MAQAM original annotations (ours)	Creative Commons Attribution 4.0 License
MAQAM source code (ours)	MIT License

Table 8: License of the data and tools.

B Per-Feature Synchronization Evaluation

Feature	**All Words**	**INV Words**	**OOV Words**
prc3	99.8	99.8	100.0
prc2	98.3	98.3	96.9
prc1	98.6	98.7	96.9
prc0	96.9	96.8	100.0
per	99.6	99.6	100.0
asp	99.6	99.6	98.4
vox	99.6	99.6	98.4
mod	99.1	99.2	93.8
form_gen	95.8	95.8	95.3
gen	95.4	95.4	96.9
form_num	98.8	98.9	95.3
num	98.3	98.4	95.3
stt	96.9	96.9	96.9
cas	96.0	96.0	98.4
enc0	99.4	99.5	93.8
enc1	100.0	100.0	96.9
Average	**98.3**	**98.3**	**97.1**

Table 9: Q-Test synchronization accuracies (%) for all, in-vocabulary (INV), and out-of-vocabulary (OOV) words.

Using LALMs for Pronunciation Error Detection in Arabic

Osama Yousef[1], Hossam Hassan[1], Amin Ahmad[2], Walid Magdy[3], Ahmed Ali[4]
[1]Kalimat, UK
[2]PAIR System, Palo Alto, California, United States
[3]School of Informatics, The University of Edinburgh, UK
[4]HUMAIN, Saudi Arabia

info@kalimat.dev

Abstract

This paper presents Qwen-MDD-AR, our system for the Iqra'Eval.2 Interspeech 2026 challenge on Modern Standard Arabic pronunciation assessment. Built on Qwen3-ASR-1.7B, it frames Arabic mispronunciation detection and diagnosis as autoregressive speech-to-phoneme generation. We add the 68-phoneme MSA inventory as special tokens, freeze the audio encoder, and adapt the decoder with LoRA. Using only provided challenge data, our submission ranked 6th of 19 teams on the hidden QuranMB.v2 test set, with 0.6702 F1 and 0.0445 PER. In a controlled comparison on our internal mixed held-out set, a Wav2Vec2-BERT CTC baseline trained on the same data achieved better PER, precision, and F1, while Qwen-MDD-AR obtained the same recall and a small Correct Diagnosis advantage (0.9496 vs. 0.9328). Results suggest LoRA-adapted LALMs are a practical alternative for Arabic MDD, while alignment accuracy and inference cost remain open.

1 Introduction

Automatic pronunciation assessment is central to Computer-Assisted Pronunciation Training (CAPT), where learners need scalable and localized feedback on spoken errors. Mispronunciation Detection and Diagnosis (MDD) remains difficult for morphologically rich and phonetically complex languages such as Modern Standard Arabic (MSA), where small phonetic confusions can change the pedagogical feedback required (Kheir et al., 2023).

Recent Arabic MDD work has benefited from standardized benchmarks and phoneme inventories, including QuranMB.v1 and the Iqra'Eval shared tasks (Kheir et al., 2025; El Kheir et al., 2025). Iqra'Eval 2 adds Iqra_Extra_IS26, a roughly two-hour set of real human mispronunciations, making it a useful setting for testing whether models trained mostly on clean or synthetic data can diagnose real errors.

Most high-performing Arabic MDD systems use CTC models built on self-supervised speech representations such as Wav2Vec2 (Graves et al., 2006; Baevski et al., 2020). Strong Iqra'Eval systems such as AraS2P and Hafs2Vec used multi-stage adaptation and Tajweed-aware phonemizers, often with external acoustic data (Mattar et al., 2025; Ibrahim, 2025). In parallel, autoregressive speech models and Large Audio-Language Models (LALMs) have become plausible alternatives for speech-to-phoneme generation, although they do not remove inference-cost concerns and are not uniformly better than CTC systems.

The motivation for using a LALM is therefore not to replace CTC systems outright, but to test a different inductive bias under the same challenge data constraints. CTC models are naturally suited to monotonic acoustic-to-token alignment, while autoregressive decoders can condition each generated phoneme on a broader output history and the model's pretrained speech-language representations. For a small real-error dataset, this broader context may help diagnosis, but it can also introduce risks: the decoder may rely on linguistic priors, smooth over acoustically present errors, or generate extra tokens. A careful comparison must therefore report both the strengths and weaknesses of the generative formulation.

We present Qwen-MDD-AR, a LoRA-adapted Qwen3-ASR-1.7B system for autoregressive speech-to-phoneme generation with 68 atomic MSA phoneme tokens (Shi et al., 2026; Hu et al., 2021). Unlike prior Iqra'Eval systems centered on acoustic-model adaptation and specialized phonemization (Mattar et al., 2025; Ibrahim, 2025), our contribution is an empirical investigation of this recent LALM family for Arabic MDD, not a new architecture or fine-tuning algorithm. We evaluate it under challenge-data constraints against a matched-data Wav2Vec2-BERT CTC baseline (Communication et al., 2023) and analyze recognition errors and

Proceedings of The Fourth Arabic Natural Language Processing Conference (ArabicNLP 2026), pages 164–169
October 28-29, 2026 ©2026 Association for Computational Linguistics

diagnosis outcomes; these results do not establish architectural or diagnostic superiority.

2 Related Work

Self-supervised acoustic models with CTC remain strong for Arabic MDD. Prior Iqra'Eval systems adapted Wav2Vec2 and HuBERT variants, while AraS2P and Hafs2Vec showed the value of domain-specific pre-training and phonemization (El Kheir et al., 2025; Baevski et al., 2020; Hsu et al., 2021; Mattar et al., 2025; Ibrahim, 2025). Transformer-based Arabic MDD systems have also been used to detect insertion, deletion, and substitution errors (Alrashoudi et al., 2025). Our work keeps this CTC line as a baseline and studies a different adaptation path.

Autoregressive speech models provide another route by generating phoneme sequences directly. Arabic evidence includes MUBEEN, which adapted a Whisper encoder for pronunciation assessment and showed that encoder-decoder models can be competitive in a shared-task setting (Radford et al., 2022; El Kheir et al., 2025). Related English work reports gains from instruction-tuned ALMs over non-fine-tuned or cascaded baselines for mispronunciation detection and feedback generation (Liu et al., 2026; Wu et al., 2024). We treat these studies as motivation rather than direct evidence that LALMs dominate MDD, since their datasets, languages, and baselines differ from the present Arabic challenge.

Parameter-efficient fine-tuning is especially relevant when adapting billion-parameter speech models to small task-specific datasets. LoRA reduces the number of trainable parameters by injecting low-rank updates into selected projection matrices (Hu et al., 2021), and has been used in speech settings such as multi-accent ASR (Bagat et al., 2025). Ahn and Nam (Ahn and Nam, 2025) show that a LoRA-tuned multimodal LLM can support a simpler APA/MDD methodology, while also indicating that specialized systems can retain advantages on core PER metrics. We adapt this PEFT direction to Arabic MDD and compare it with a Wav2Vec2-BERT CTC baseline trained on identical data.

3 System Description

3.1 Task and Data

Qwen-MDD-AR frames MDD as speech-to-phoneme generation. The Iqra'Eval output is a whitespace-separated phoneme sequence, e.g., `l a < a k a l uu m i n f a w q I h i m`. We added the 68 MSA phonemes as atomic special tokens so the model can generate the target phoneme inventory directly. The inventory follows the MSA phonetizer of Halabi and Wald (Halabi and Wald, 2016), which represents gemination by doubling the consonant symbol (e.g., b → bb); it comprises 40 base symbols (12 vowels and 28 consonants) plus 28 geminates, and the exact 68-symbol mapping is given in the official challenge inventory (IqraEval, 2026). This avoids a separate explicit CTC or forced-alignment decoding stage, though the model still learns implicit acoustic-text alignment through attention.

We used only the challenge data for adaptation: Iqra_train (∼79 hours of error-free speech), Iqra_TTS (∼80 hours with artificial mispronunciations), and Iqra_Extra_IS26 (∼2 hours of real mispronunciations). We excluded 133 utterances (10%) of Iqra_Extra_IS26 from training. The remaining portion received a sampling weight of 20 and was combined and shuffled with Iqra_train and Iqra_TTS at the start of training to reduce data imbalance.

Configurations were selected using the official Dev split (2,588 Iqra_train utterances). The 133 excluded utterances were not used for checkpoint selection, hyperparameter tuning, early stopping, or other training decisions. After configurations were fixed, we evaluated an internal evaluation set of 433 utterances: these 133 real-error examples plus 300 randomly selected error-free Dev utterances. Thus, the real-error subset is independent of model selection, but the clean component overlaps Dev. This mixed set supports a matched-example comparison, not an independent test estimate, and its F1 is not directly comparable to hidden QuranMB.v2.

3.2 Qwen-MDD-AR

The system builds on Qwen3-ASR-1.7B (Shi et al., 2026). We froze the audio encoder and adapted the autoregressive decoder with LoRA (Hu et al., 2021), targeting `q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, and `down_proj` with rank $r = 64$, $\alpha = 128$, and dropout 0.05. We also unfroze `embed_tokens` and `lm_head` so the model could emit the new phoneme tokens. This left 700,551,168 trainable parameters, about 25.6% of the 2.04B logical parameters that include the audio encoder; the PEFT wrapper reported 2.738B total because it retained separate trainable copies of the token embeddings and language-model head.

The encoder was frozen to retain its pretrained acoustic representation and reduce adaptation memory; partial encoder fine-tuning was not evaluated. LoRA rank 64 and $\alpha = 128$ were practical capacity choices under the compute budget, not experimentally established optima. Encoder freezing, LoRA, phoneme-token extension, and 20-fold oversampling were not independently ablated.

Training used an effective batch size of 32, learning rate 5×10^{-5}, cosine scheduling with 0.05 warmup, maximum gradient norm 1.0, and maximum audio duration 100 seconds. The curriculum first fine-tuned without augmentation to 13,500 steps, then continued with SpecAugment (Park et al., 2019) to 19,000 steps, and finally used the stable 13,500-step checkpoint with SpecAugment plus 0.1 label smoothing to produce the 25,000-step submission. Training ran on a single NVIDIA L4 (24 GB); total time was approximately 40 h 24 m, with the final 19,000-to-25,000-step stage measured at 9 h 41 m 50 s. Peak GPU memory was not recorded.

3.3 CTC Baseline

For a matched-data comparison, we trained a Wav2Vec2-BERT CTC baseline (Communication et al., 2023; Graves et al., 2006) on the same oversampled challenge data. It was optimized for 26,000 steps with effective batch size 32, learning rate 1×10^{-5}, linear scheduling, 500 warmup steps, weight decay 0.01, and maximum gradient norm 1.0.

4 Results

4.1 Evaluation Metrics

Iqra'Eval ranks systems primarily by phoneme-level F1. For internal evaluation, we use the same evaluation implementation for both models: `align_data.py` aligns canonical, human-annotated, and predicted sequences using `metric.py`; `ins_del_cor_sub_analysis.py` computes detection and diagnosis metrics. True Accept (TA) and False Reject (FR) count correctly pronounced events accepted and rejected, respectively; True Reject (TR) and False Accept (FA) count actual pronunciation errors detected and missed. With $T = \mathrm{TA} + \mathrm{FR} + \mathrm{TR} + \mathrm{FA}$,

$$P = \frac{\mathrm{TR}}{\mathrm{TR} + \mathrm{FR}}, \qquad R = \frac{\mathrm{TR}}{\mathrm{TR} + \mathrm{FA}},$$
$$F_1 = \frac{2PR}{P + R}, \qquad \mathrm{DA} = \frac{\mathrm{TA} + \mathrm{TR}}{T}.$$

Detection Accuracy (DA) measures detection decisions; Correct Diagnosis (CD) is the fraction of TR events whose predicted realization matches the human annotation. TA/FR rates divide by $\mathrm{TA} + \mathrm{FR}$; TR/FA rates divide by $\mathrm{TR} + \mathrm{FA}$.

Separately, against the human sequence, let C, S, I, D denote matches, substitutions, insertions, and deletions, and $N = C+S+D$. Phoneme Error Rate (PER) is $(S + I + D)/N$, Accuracy is $1-\mathrm{PER}$, and Correct Rate is C/N. Only Accuracy and PER penalize insertions; PER is not generally $1 - \mathrm{DA}$. The alignment treats supplied symbols as tokens and uses `<eps>` for gaps. We retain the leaderboard's reported values separately from internal sequence metrics. The internal set has only 122 canonical–human error events, making precision and F1 sensitive to false rejections; its F1 cannot be directly compared with hidden QuranMB.v2.

4.2 Baseline Comparison: CTC vs. Autoregressive LALM

To compare a CTC baseline and our frozen-encoder generative approach under matched data conditions, we evaluated both the Wav2Vec2-BERT baseline and Qwen-MDD-AR on the exact same subsets. The results of this controlled comparison are detailed in Table 2.

The Wav2Vec2-BERT baseline achieved lower PER on both Dev (0.0515 vs. 0.0649) and the internal evaluation set (0.0472 vs. 0.0541). It also achieved higher Accuracy (0.9528), precision (0.1608), and F1 (0.2761) on the internal set, making it stronger on aggregate alignment and detection metrics.

Qwen-MDD-AR matched Recall (0.9754) and obtained higher CD (0.9496 vs. 0.9328), corresponding to 6 vs. 8 diagnosis errors among 119 True Reject events. Because this is a two-error difference without a significance test, we treat it as preliminary. The identical FA rate (0.0246) is not a positive differentiator.

The comparison suggests a trade-off rather than a clean winner. The CTC model is better on alignment-sensitive aggregate metrics, as expected for a monotonic alignment objective. Qwen-MDD-AR is competitive despite a frozen encoder and generative decoder, but the internal evidence does not establish a diagnostic advantage.

4.3 Error Analysis

Against the human pronunciation on the 433 utterances, Qwen-MDD-AR makes 440 substitutions,

Table 1: Qwen-MDD-AR results on the hidden QuranMB.v2 Iqra'Eval 2026 leaderboard.

Team	F1	Precision	Recall	Correct Rate	TA	FR	FA	CD	PER
Organizer's baseline	0.4414	0.3039	0.7707	0.8385	0.8763	0.1237	0.2293	0.612	0.1308
Qwen-MDD-AR (25k steps)	**0.6702**	**0.6666**	**0.6738**	**0.9379**	**0.9758**	**0.0242**	**0.3262**	**0.8508**	**0.0445**
Qwen-MDD-AR (19k steps)	0.6670	0.6678	0.6663	0.9378	0.9762	0.0238	0.3337	0.8462	0.0446
Qwen-MDD-AR (13.5k steps)	0.6288	0.5855	0.6791	0.9263	0.9654	0.0346	0.3209	0.8326	0.0538

Table 2: Matched-data comparison on Dev and internal evaluation sets. Internal PER is recomputed from the stored human–prediction alignments.

Metric	Wav2Vec2-BERT	Qwen-MDD-AR
Dev Set (2,588 samples)		
PER	**0.0515**	0.0649
Internal Evaluation Set (433 samples)		
PER	**0.0472**	0.0541
Detection Accuracy	**0.9600**	0.9573
Accuracy	**0.9528**	0.9459
Correct Rate	**0.9596**	0.9569
MD&D Standard Metrics		
F1-Score	**0.2761**	0.2633
Precision	**0.1608**	0.1522
Recall	0.9754	0.9754
MD&D Detection Rates		
True Accept (TA)	**0.9599**	0.9572
False Reject (FR)	**0.0401**	0.0428
True Reject (TR)	0.9754	0.9754
False Accept (FA)	0.0246	0.0246
Correct Diagnosis (CD)	0.9328	**0.9496**

171 insertions, and 232 deletions, versus 342, 106, and 288 for the CTC baseline (15,591 aligned phones; the internal PER values in Table 2 come from these alignments). The CTC model trades 98 fewer substitutions and 65 fewer insertions for 56 more deletions. Qwen-MDD-AR is exact on 213/433 utterances versus 196/433, but its errors are heavier when they occur: 90th-percentile utterance PER is 17.25% versus 15.31%, with 41 versus 28 utterances containing runs of $\geq$3 consecutive errors. A paired-utterance bootstrap puts the PER gap at 0.686 points (95% CI 0.239–1.149, $p \approx 0.002$).

This pattern is consistent with the decoder formulations. The autoregressive decoder conditions each token on its output history, so a wrong local hypothesis can cascade into coherent substitution/insertion bursts, while CTC's frame-level blank/repeat collapse keeps errors acoustically local but omits weakly evidenced phones. Qwen's output length is nearly unbiased (hypothesis/reference ratio 0.996 vs. 0.988), and errors are flat across utterance thirds, contradicting end-of-sequence drift. Vowel-quality confusions dominate for both models (58.6% of Qwen substitutions vs. 68.1% of CTC substitutions involve vowel-like symbols on both sides); Qwen additionally substitutes rather than deletes on emphatics and pharyngeals. Two utterances illustrate the complementarity: on `audio_73881` Qwen-MDD-AR makes 2 substitutions, 2 insertions, and 2 deletions while the CTC baseline is exact, whereas on `audio_5069_57` Qwen-MDD-AR is exact while the CTC baseline makes 3 substitutions and 1 deletion. With only 122 real error events, these observations are descriptive and exploratory; architecture is a plausible mechanism, not an isolated cause.

4.4 Official Challenge Results

On hidden QuranMB.v2, the 25,000-step checkpoint ranked 6th of 19 teams with 0.6702 F1 and 0.8508 CD (Table 1). The 13,500- and 19,000-step submissions achieved 0.6288 and 0.6670 F1, respectively. Leaderboard scores were available after each submission, but they were not used for tuning; configuration decisions were based on the official Dev split. Each stage was trained to convergence, evaluated, and then continued from the stable checkpoint with additional regularization. This is a sequential development history, not an ablation: training duration, SpecAugment, and label smoothing vary jointly. The final combined configuration obtained the best hidden-set F1 among our submissions, but these results do not isolate any regularizer's effect. We used no additional task-specific external data or custom acoustic-pretraining stage; Qwen3-ASR itself is extensively pretrained.

5 Conclusion and Future Work

We introduced Qwen-MDD-AR, an autoregressive PEFT-based pipeline for Arabic MDD, and its first empirical investigation with a matched-data CTC baseline under challenge constraints. Our submission achieved 6th of 19 teams with an F1-score of 0.6702 using only provided Iqra'Eval 2026 data.

Our comparative analysis revealed a trade-off: the Wav2Vec2-BERT CTC baseline maintained an advantage in Phoneme Error Rate (PER), precision, F1-score, and overall accuracy on our internal evaluation set, while Qwen-MDD-AR matched its Recall and showed a small Correct Diagnosis difference (0.9496 vs. 0.9328). We treat this diagnosis result as preliminary because it corresponds to only

two fewer errors and has not been tested for statistical significance. Overall, Qwen-MDD-AR is best viewed as a promising LALM-based alternative for Arabic MDD rather than a replacement for strong CTC systems.

The main practical lesson is that LALM adaptation can be made simple: the system does not require a custom acoustic pre-training stage, and most of the backbone remains frozen. At the same time, the controlled baseline shows why simplicity alone is not enough. Any future LALM-based MDD system should be evaluated against strong CTC models on both alignment-sensitive metrics and diagnosis-specific metrics, and should report inference cost in addition to trainable-parameter count.

Future work should evaluate multi-stage domain adaptation and reference-guided prompting, while testing whether reference context improves diagnosis or masks real pronunciation errors. A reference-guided prompt is especially attractive for Quranic recitation, where the intended text is known in advance; however, this same prior can encourage the model to normalize away acoustic evidence. Error-type analysis over substitutions, insertions, and deletions would help determine whether the reference improves genuine diagnosis or merely improves fluent phoneme generation.

We plan to release the training and inference code, tokenizer modifications for the 68 phoneme tokens, configuration files, evaluation scripts, and the LoRA adapter, subject to the dataset license.

6 Limitations

The CD difference in the internal comparison is small (6 vs. 8 diagnosis errors among 119 True Reject events), and we do not report a significance test for it; only the recognition PER gap is bootstrap-tested. The internal evaluation set and hidden QuranMB.v2 have different error distributions, so their F1-scores should not be compared directly. We also lack detailed speaker demographics, phoneme-level error distributions, and linguistic-complexity statistics for the hidden test set, limiting deeper error analysis.

Our efficiency discussion is limited to training and adaptation: LoRA reduces trainable parameters relative to full SFT, but autoregressive inference with a 1.7B decoder is likely heavier than a CTC forward pass, and we do not report latency, FLOPs, or throughput. Finally, this challenge-system paper does not compare against GOP variants, RNN-T MDD systems, or non-Arabic benchmarks such as L2-ARCTIC, and it does not quantify the risk that autoregressive decoding may use linguistic priors to mask or "correct" pronunciation errors.

References

Taekyung Ahn and Hosung Nam. 2025. English Pronunciation Evaluation without Complex Joint Training: LoRA Fine-tuned Speech Multimodal LLM. *arXiv preprint*. ArXiv:2509.02915 [cs].

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.

Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. *arXiv preprint*. ArXiv:2006.11477 [cs].

Raphaël Bagat, Irina Illina, and Emmanuel Vincent. 2025. Mixture of LoRA Experts for Low-Resourced Multi-Accent Automatic Speech Recognition. In *Interspeech 2025*, pages 1143–1147. ArXiv:2505.20006 [cs].

Seamless Communication, Loïc Barrault, Yu-An Chung, Mariano Coria Meglioli, David Dale, Ning Dong, Mark Duppenthaler, Paul-Ambroise Duquenne, Brian Ellis, Hady Elsahar, Justin Haaheim, John Hoffman, Min-Jae Hwang, Hirofumi Inaguma, Christopher Klaiber, Ilia Kulikov, Pengwei Li, Daniel Licht, Jean Maillard, and 46 others. 2023. Seamless: Multilingual Expressive and Streaming Speech Translation. *arXiv preprint*. ArXiv:2312.05187 [cs].

Yassine El Kheir, Amit Meghanani, Hawau Olamide Toyin, Nada Almarwani, Omnia Ibrahim, Yousseif Ahmed 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: Shared Tasks*, pages 443–452. 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 - ICML '06*, pages 369–376, Pittsburgh, Pennsylvania. ACM Press.

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. *arXiv preprint*. ArXiv:2106.07447 [cs].

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 [cs].

Ahmed Ibrahim. 2025. Hafs2vec: A system for the IqraEval arabic and qur'anic phoneme-level pronunciation assessment. In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 453–456. Association for Computational Linguistics.

IqraEval. 2026. Modern Standard Arabic phoneme inventory. `https://huggingface.co/spaces/IqraEval/ArabicPhoneme`. Hugging Face Space; accessed 2026.

Yassine El Kheir, Ahmed Ali, and Shammur Absar Chowdhury. 2023. Automatic pronunciation assessment – a review. *Preprint*, arxiv:2310.13974 [cs].

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 [cs].

Hongfu Liu, Zhouying Cui, Xiangming Gu, and Ye Wang. 2026. Unlocking Large Audio-Language Models for Interactive Language Learning. *arXiv preprint*. ArXiv:2601.14744 [cs].

Bassam Mattar, Mohamed Fayed, and Ayman Khalafallah. 2025. AraS2p: Arabic speech-to-phonemes system. In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 469–474. Association for Computational Linguistics.

Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. 2019. SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition. In *Interspeech 2019*, pages 2613–2617. ArXiv:1904.08779 [eess].

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust Speech Recognition via Large-Scale Weak Supervision. *arXiv preprint*. ArXiv:2212.04356 [eess].

Xian Shi, Xiong Wang, Zhifang Guo, Yongqi Wang, Pei Zhang, Xinyu Zhang, Zishan Guo, Hongkun Hao, Yu Xi, Baosong Yang, Jin Xu, Jingren Zhou, and Junyang Lin. 2026. Qwen3-ASR technical report. *Preprint*, arxiv:2601.21337 [cs].

Minglin Wu, Jing Xu, Xixin Wu, and Helen Meng. 2024. Prompting Large Language Models with Mispronunciation Detection and Diagnosis Abilities. In *Interspeech 2024*, pages 2990–2994. ISCA.

Optimizing Inference Cost for Arabic–English Mathematical Reasoning in LLMs

Sahar Melhem **Ahmad Mustafa***
Computer Information Systems Department
Jordan University of Science and Technology, Irbid, Jordan
semelhem24@cit.just.edu.jo ammustafa@just.edu.jo

Abstract

Large language models (LLMs) can solve mathematical problems through prompting, yet not all reasoning is cost-justified when using paid APIs. We study prompt-only methods for overthinking control in the black-box setting across four datasets: Arabic GSM8K, English GSM8K, English SVAMP, and ArMATH, each evaluated on 100-question subsets divided into Easy, Medium, and Hard categories. We test Direct Answer, Chain-of-Thought (CoT), First Answer Anchoring (FAA-lite), Confidence-Triggered Escalation (CTE), Self-Consistency, Verification, Arabic prompt variations, and Language-Optimal Prompt Routing (LOPR). CTE achieves 90% accuracy on Arabic GSM8K at 1.14 calls/query, 96% on English GSM8K at 1.06 calls/query, and 98% on SVAMP at 1.08 calls/query. Self-Consistency leads on ArMATH at 90% but costs 2.00 calls/query. Verification consistently increases token cost without accuracy gains. Format-Controlled CoT is the best single-call Arabic method; Bilingual-Glossary CoT is consistently harmful. CTE calibration reveals that explicit low-confidence outputs are rare ($<1\%$), so CTE succeeds mainly because high-confidence predictions are accurate, not through active uncertainty detection.

1 Introduction

Mathematical word problems involve multiple reasoning steps and are a natural challenge for LLMs, since text completion alone cannot reliably solve them. Chain-of-Thought (CoT) prompting encourages step-by-step reasoning before reaching a conclusion and substantially improves accuracy on multi-step arithmetic problems (Wei et al., 2022). Self-Consistency (SC) extends CoT by sampling multiple reasoning paths at nonzero temperature and selecting the most common final answer (Wang et al., 2022).

In black-box API use, every additional call costs money, users pay per token and per query without access to model weights, logits, or hidden states. Our question is not whether more reasoning helps, but whether it is *worth the extra cost*. Efficient reasoning has been described as additional computation that reliably improves quality (Sui et al., 2025). Overthinking occurs when extra calls or tokens fail to produce consistent accuracy gains (Sui et al., 2025; Wei et al., 2025; Qiu and Srivastava, 2025). Most existing efficient-reasoning methods require model access or training (Wei et al., 2025; Wen et al., 2025); our methods use only text outputs and work in any black-box API.

Initial trials on GSM8K-like tasks showed that single-query CoT already yields high accuracy. Multi-sample techniques sometimes outperformed single-query CoT, but in other cases provided no improvement, echoing overthinking findings where longer or repeated reasoning yields diminishing or even detrimental accuracy gains (Sui et al., 2025; Wei et al., 2025). We operationalize overthinking as: if a technique requires additional calls or tokens without consistent accuracy improvement, it overthinks.

We evaluate prompt-only controls on four datasets spanning Arabic and English. Arabic has fewer mathematical reasoning benchmarks; ArMATH provides 6,000 Modern Standard Arabic word-problem examples (Alghamdi et al., 2022). Multilingual CoT studies show that the language of the reasoning trace affects model behavior (Shi et al., 2022), motivating separate Arabic and English analyses.

The methods we propose are intentionally minimal. FAA-lite asks the model to commit to an initial numerical answer before a brief verification step. CTE attaches a strict confidence label to the first CoT call and escalates to additional samples only when confidence is low or absent. Neither requires training, special decoding, or internal probabilities (Hu et al., 2025; Zhao et al., 2025; Ehab and Hamdi, 2026).

Proceedings of The Fourth Arabic Natural Language Processing Conference (ArabicNLP 2026), pages 170–178
October 28-29, 2026 ©2026 Association for Computational Linguistics

2 Contributions and Scope

We make six contributions, presented in order.

Black-box cost-accuracy evaluation. A black-box cost-accuracy evaluation across four datasets. Prior work usually reports only accuracy (Wei et al., 2022; Wang et al., 2022; Cobbe et al., 2021; Patel et al., 2021). We additionally report tokens per question, calls per question, confidence path counts, and per-difficulty accuracy, making the cost-accuracy tradeoff explicit. A method that improves accuracy by one point while tripling calls may be unacceptable in an API setting.

CTE calibration analysis. An empirical analysis of CTE confidence calibration separating responses by path (trusted-high, escalated-low, missing-confidence). The low-confidence path rarely fires, which has broad implications for confidence-based prompt routing. To our knowledge this is the first study reporting this calibration pattern across Arabic and English math reasoning under matched conditions.

CTE as adaptive inference rule. CTE uses one deterministic CoT-style call with a required confidence line. Trusted-high answers return immediately; low or missing confidence triggers two additional stochastic CoT samples with majority voting. CTE resembles cost-aware routing methods such as PromptWise (Hu et al., 2025) and difficulty-aware routing (Zhao et al., 2025; Ehab and Hamdi, 2026), but requires only text outputs.

FAA-lite analysis. The explicit study of FAA-lite diagnoses whether anchoring the initial answer prevents detrimental self-revisions, as suggested by past literature (Sui et al., 2025; Wei et al., 2025). FAA-lite performs better than CoT in Arabic GSM8K, but not in the other three datasets.

Arabic-specific prompt variants. We test Format-Controlled CoT, Numeral-Normalized CoT, and Bilingual-Glossary CoT only on Arabic datasets, focused on final answer extraction, mixed digit systems, and lexicon ambiguity, respectively. Only Format-Controlled CoT improves over CoT. Bilingual-Glossary CoT consistently hurts, reported as a negative result with practical implications for Arabic prompt design.

Preliminary LOPR-style analysis. LOPR-style prompting keeps the Arabic problem as input but asks the model to reason in English (Shi et al., 2022). Results are reported as a language-routing diagnostic, not a main method. We fix $K{=}2$ for all final experiments and avoid dataset-specific tuning, distinguishing $K{=}1$ from true Self-Consistency since no voting occurs at one sample (Wang et al., 2022).

3 Related Work

CoT prompting converts direct-answer tasks into step-by-step reasoning tasks and improves accuracy on arithmetic problems in capable models (Wei et al., 2022). Self-Consistency extends CoT by sampling several reasoning paths and voting on the final answer (Wang et al., 2022). Our experiments include both, but our focus is whether extra cost is justified.

GSM8K is the standard grade-school math benchmark (Cobbe et al., 2021). SVAMP tests robustness against surface-level dataset cues (Patel et al., 2021). ArMATH is a Modern Standard Arabic word-problem dataset with 6,000 examples (Alghamdi et al., 2022). Multilingual CoT research shows that the reasoning language impacts model performance and output formatting (Shi et al., 2022), motivating our separate Arabic and English reporting.

Efficient-reasoning work includes early-exit methods (Wei et al., 2025), Budget Thinker which uses control tokens and training to impose a token budget (Wen et al., 2025), and batch prompting that amortises cost across related queries (Qiu and Srivastava, 2025), all requiring model access or training. Routing approaches such as PromptWise (Hu et al., 2025), difficulty-aware routing (Zhao et al., 2025; Ehab and Hamdi, 2026), and RADAR (Fernandez et al., 2025; Qu et al., 2026) are related but require separate routing models. CTE resembles these but uses text only.

Verbalized confidence can proxy logit-based estimates (Kadavath et al., 2022; Tian et al., 2023; Lin et al., 2022); our calibration analysis extends this to Arabic and English math tasks without any fine-tuning. Prompt engineering reproducibility (White et al., 2023) informed our logging and versioning protocol.

4 Methods

All methods share the same model, output format (`#### <number>`), and strict parser (Cobbe et al., 2021). Methods fall into three cost tiers: one-call (Direct, CoT, FAA-lite, Arabic variants,

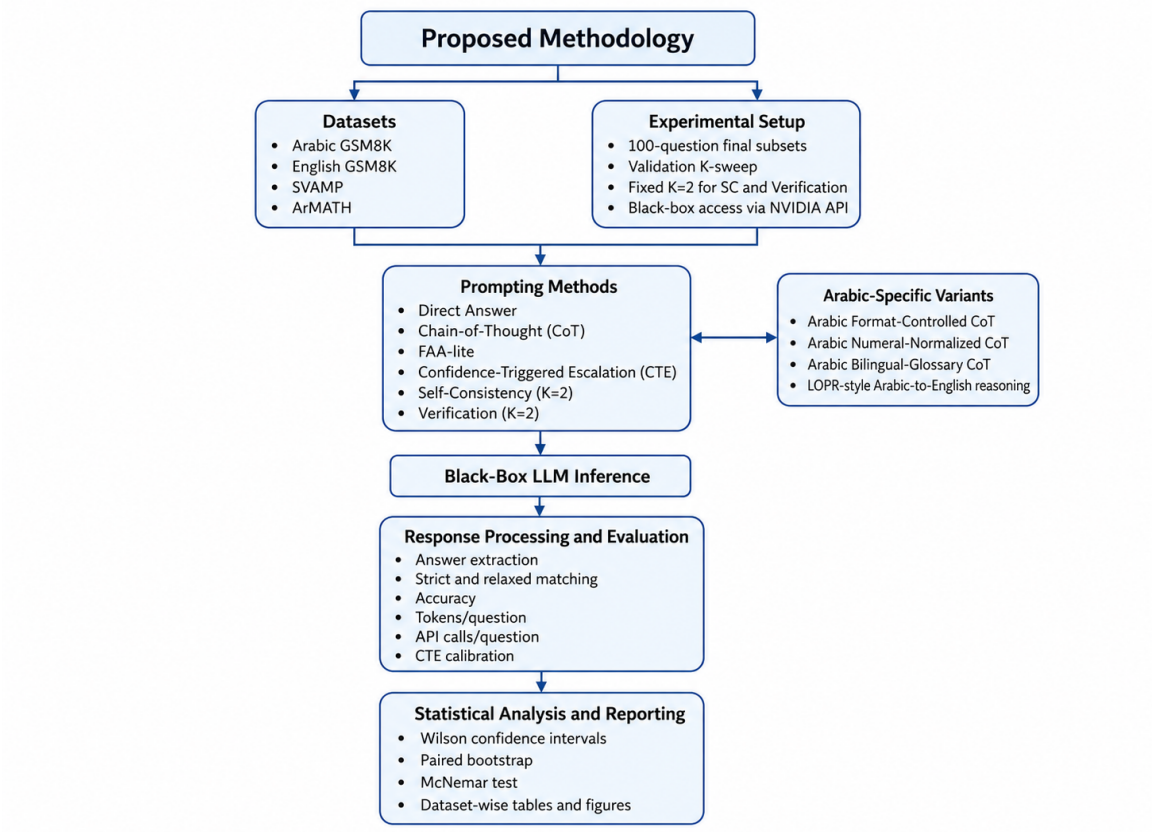


Figure 1: Overview of the experimental pipeline, from dataset selection and prompting methods through black-box LLM inference to statistical evaluation.

LOPR), adaptive up to 3 calls (CTE), and fixed multi-call (SC K=2, Verification). Figure 1 provides a high-level overview of the full experimental pipeline, from dataset selection through statistical evaluation.

Direct Answer. Final number only; underthinking baseline (Sui et al., 2025).

Chain-of-Thought (CoT). Step-by-step reasoning; primary single-call baseline (Wei et al., 2022). All proposed methods are compared against CoT.

Self-Consistency K=2. Two independent CoT samples at T=0.7, majority vote (Wang et al., 2022). $K = 2$ was chosen via a held-out validation sweep (Section 5).

Verification K=2. Two candidates generated, then a separate verifier selects the final answer. Total cost ≈3 calls per question.

FAA-lite. Single call: model commits to an initial numerical answer, then briefly verifies and produces the final answer line. Tests whether anchoring reduces harmful self-revision.

CTE. Confidence-Triggered Escalation. A CoT call requiring an explicit `CONFIDENCE: High` or `CONFIDENCE: Low` line. Trusted-high answers return immediately; low or missing confidence triggers two additional stochastic CoT samples with majority voting. Maximum 3 calls; average approximately 1.06–1.16 calls depending on the dataset. Confidence detection is strict: Only an exact `CONFIDENCE:` line is accepted.

Arabic variants (Arabic datasets only). *Format-Controlled CoT*: enforces Western-Arabic numerals and a single final answer line. *Numeral-Normalized CoT*: converts Eastern Arabic numerals. *Bilingual-Glossary CoT*: prepends Arabic mathematical term definitions.

LOPR-style. Language-Optimal Prompt Routing. The Arabic problem is kept as input, but the model is instructed to reason in English and return the answer in the required format. Unlike full routing systems that select the reasoning language per-problem based on a difficulty signal, this condition always routes to English and serves as a diagnostic of cross-lingual reasoning potential rather than a

deployable routing strategy. Reported as a preliminary analysis from an earlier experimental run, not as a method in the main evaluation tables.

The eight core methods organize into three cost tiers. Direct Answer, CoT, FAA-lite, Format-Controlled CoT, Numeral-Normalized CoT, Glossary CoT, and LOPR-style are one-call methods. CTE is adaptive with a maximum of three calls. Self-Consistency and Verification are fixed multi-call methods at two and three calls, respectively.

Both English and Arabic prompts follow the same structure. Full prompt templates for all remaining methods (Direct Answer, FAA-lite, Verification, Numeral-Normalized CoT, and Bilingual-Glossary CoT), along with the complete experimental code, are publicly available at URL

5 Experimental Setup

Model & API. All experiments use `openai/gpt-oss-20b` accessed via the NVIDIA API. Deterministic first-pass calls (Direct, CoT, FAA-lite, the CTE first pass, and the verifier) use T=0. Stochastic samples (Self-Consistency, Verification, and CTE escalation) use T=0.7, top-p=1.0. Rate limit: 30 requests per minute. These parameters are reported because inference cost is part of the object of study.

Datasets. Arabic GSM8K (Omartificial-Intelligence-Space, 2025), English GSM8K (Cobbe et al., 2021), SVAMP (Patel et al., 2021), and ArMATH (Alghamdi et al., 2022). Each dataset uses a 100-question final test set stratified into Easy (34 questions), Medium (33), and Hard (33). Difficulty is derived from reasoning-step count for GSM8K-style datasets, equation structure for SVAMP, and equation-template tertiles for ArMATH (Alghamdi et al., 2022). Difficulty labels are used only for stratified sampling and per-difficulty reporting.

K selection. A preliminary K-sweep was conducted once on a held-out 50-question Arabic validation subset with $K \in \{1, 2, 4, 8, 16\}$. K=4 gained 4 accuracy points over K=2 (92% vs. 88%) but nearly doubled token cost; K=8 and K=16 returned to 90% with no stable improvement. We fix K=2 as the cost-efficiency elbow to avoid dataset-specific tuning. The K=2 choice is a deployment-oriented compromise: K=4 might give 2–4 additional accuracy points at twice the cost, but picking the elbow of the validation curve preserves comparability across datasets. A dedicated K=4 sensitivity check is left for future work. Figure 2 plots accuracy against token cost.

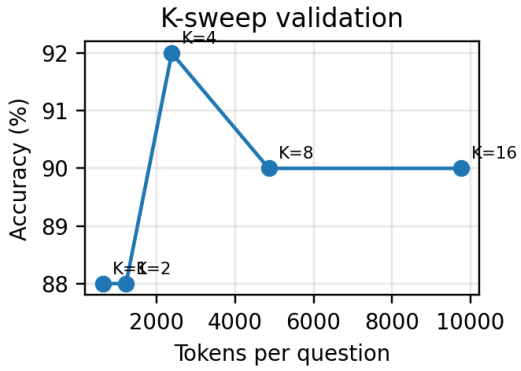


Figure 2: Plots accuracy against token cost K-sweep on different values of (K=1,2,4,8,16)

Table 1: K-sweep summary (n=50 Arabic GSM8K validation set).

K	Acc. %	Tok./Q	Calls/Q	Note
1	88.0	634	1.00	single stochastic ref.
2	88.0	1223	2.00	min. true multi-sample
4	92.0	2387	4.00	higher acc., high cost
8	90.0	4857	8.00	no stable gain
16	90.0	9762	16.00	large cost, no gain

LOPR-style condition handling. The LOPR-style condition uses Arabic problems and is part of the Arabic analysis, not the English benchmarks. LOPR-style numbers come from a test run with a separate experimental pipeline and are reported in Section 7.1 rather than merged into the main Arabic tables.

Logging and reproducibility. Raw model responses, fallback-extraction cases, API errors, prompt versions, cache keys, and per-problem records were saved for every run. Prompt-only experiments are difficult to reproduce when prompts or parsers change (White et al., 2023).

Metrics and Statistics. We report accuracy, tokens per question, calls per question, and fallback-extraction rate. Where two methods produce similar accuracy, we additionally report paired bootstrap differences and McNemar exact p-values from per-problem records. A two-point gap on 100 problems is not strong evidence without a paired test.

We report accuracy, tokens per question, calls per question, and fallback-extraction rate. Where two methods produce similar accuracy, we addi-

Table 2: Arabic GSM8K (n=100). **Bold** = best accuracy.

Method	Acc. %	Tok./Q	Calls/Q
Direct Answer	65.0	344	1.00
CoT	84.0	701	1.00
FAA-lite	87.0	707	1.00
CTE	**90.0**	893	1.14
Self-Consistency K=2	88.0	1368	2.00
Verification K=2	86.0	1768	2.98
Format-Controlled CoT	87.0	533	1.00
Numeral-Normalized CoT	86.0	765	1.00
Bilingual-Glossary CoT	72.0	865	1.00

Table 3: English GSM8K (n=100).

Method	Acc. %	Tok./Q	Calls/Q
Direct Answer	85.0	302	1.00
CoT	95.0	488	1.00
FAA-lite	93.0	550	1.00
CTE	96.0	583	1.06
SC K=2	**97.0**	1007	2.00
Verification K=2	94.0	1392	2.98

tionally report paired bootstrap differences and McNemar exact p-values from per-problem records. A two-point gap on 100 problems is not strong evidence without a paired test. Unless otherwise noted, bracketed values in the results denote 95% Wilson confidence intervals

6 Results

6.1 Arabic GSM8K

CTE achieves the highest accuracy at 90% $[0.83, 0.94]$ with only 1.14 calls/question (Table 2). CoT reaches 84% $[0.76, 0.90]$ at one call. Format-Controlled CoT reaches 87% at 533 tokens, below CoT's 701,making it the most token-efficient single-call method. Bilingual-Glossary CoT is harmful at 72% $[0.63, 0.80]$ (McNemar p=0.008).

6.2 English GSM8K

CoT is already strong at 95% [0.89, 0.98] (Table 3). SC reaches 97% $[0.92, 0.99]$ but requires 2.00 calls; CTE matches 96% $[0.90, 0.98]$ in 1.06 calls. No method significantly beats CoT (Direct Answer is significantly worse p=0.006). CTE's value here is cost reduction.

6.3 SVAMP

CTE and SC both reach 98% [0.93, 0.99] (Table 4); CTE uses 1.08 calls vs. SC's 2.00 and 488 tokens vs. SC's 833. Verification reaches 95% at 3.00 calls, the clearest demonstration that extra calls do not guarantee accuracy gains.

Table 4: English SVAMP (n=100).

Method	Acc. %	Tok./Q	Calls/Q
Direct Answer	91.0	233	1.00
CoT	96.0	433	1.00
FAA-lite	94.0	425	1.00
CTE	**98.0**	488	1.08
SC K=2	**98.0**	833	2.00
Verification K=2	95.0	1117	3.00

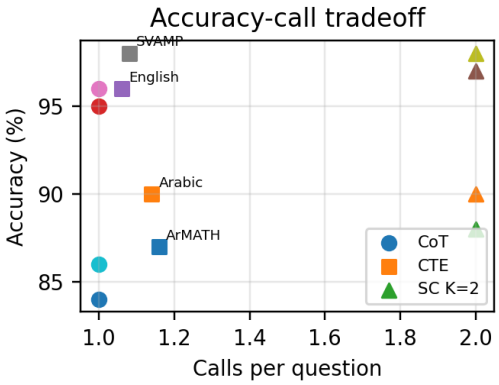


Figure 3: Accuracy vs. calls/question for CoT, CTE, and SC $K = 2$ across all four datasets. CTE matches SC at near single-call cost on three of four datasets.

6.4 ArMATH

SC achieves the best accuracy at 90% [0.83, 0.94] (Table 5); the bootstrap CI vs. CoT excludes zero ($[+0.01, +0.08]$), the only significant gain on this dataset. Format-Controlled CoT is the best single-call method at 88% $[0.80, 0.93]$ with 477 tokens. Bilingual-Glossary CoT is again harmful (bootstrap -0.06, CI $[-0.12, -0.01]$). CoT drops from 94.1% on Easy to 72.7% on Hard (Table 6).

Figure 4 compares Arabic prompt variants across both Arabic datasets; Figure 3 shows accuracy vs. calls across all four datasets.

7 Calibration Analysis

Table 7 breaks CTE responses into three paths. Across 400 total runs, only 2 responses produced `CONFIDENCE: Low`. Escalation fires almost entirely on *missing-confidence* cases where the model failed to follow the format, and those cases have very low accuracy (12.5–66.7%). Trusted-high cases are accurate 93.5–99.0% of the time.

Three observations follow:

First, trusted-high cases are highly accurate: 93.55% on Arabic GSM8K, 97.94% on English GSM8K, 98.96% on SVAMP, and 93.48% on ArMATH. When the model claims high confidence on its own initiative, it is usually right.

Table 5: ArMATH (n=100).

Method	Acc. %	Tok./Q	Calls/Q
Direct Answer	82.0	262	1.00
CoT	86.0	559	1.00
FAA-lite	84.0	509	1.00
CTE	87.0	735	1.16
SC K=2	**90.0**	1194	2.00
Verification K=2	87.0	1503	2.92
Format-Controlled CoT	88.0	477	1.00
Numeral-Normalized CoT	86.0	660	1.00
Bilingual-Glossary CoT	80.0	732	1.00

Table 6: ArMATH accuracy by difficulty level.

Method	Easy	Med.	Hard
CoT	94.1	90.9	72.7
Format-Ctrl CoT	97.1	90.9	75.8
CTE	97.1	87.9	75.8
SC K=2	97.1	93.9	78.8

Second, missing-confidence cases have very low accuracy: 33.33% on Arabic GSM8K, 33.33% on English GSM8K, 66.67% on SVAMP, and 12.50% on ArMATH. Escalation does not recover these problems. The two extra stochastic samples do not reliably converge to the right answer because the model is already confused.

Third, the explicit low-confidence path is essentially empty. Across all four datasets, we observed exactly two `CONFIDENCE: Low` cases out of 400 total CTE runs. Both were answered correctly after escalation, but n=2 is not enough to draw any conclusion.These three confidence paths are visualized in Figure 5.

The practical implication is that CTE works for a reason different from its design. It was intended to detect uncertain responses and spend extra compute on them. In practice, the model rarely produces uncertain responses on its own, and when it fails to follow the output format the extra compute does not help. CTE delivers cost savings because the trusted-high accuracy is high, not because the gating mechanism actively recovers wrong answers (Kadavath et al., 2022; Tian et al., 2023; Lin et al., 2022). Without explicit calibration training, the model defaults to an overconfident high-confidence stance on grade-school math, with low-confidence emission too rare to be useful as a gating signal.

7.1 LOPR-style Preliminary Analysis

We conducted a prior LOPR-style experiment on Arabic GSM8K. The accuracy was 0.84 $[0.76, 0.90]$ at 661 tokens per question and one

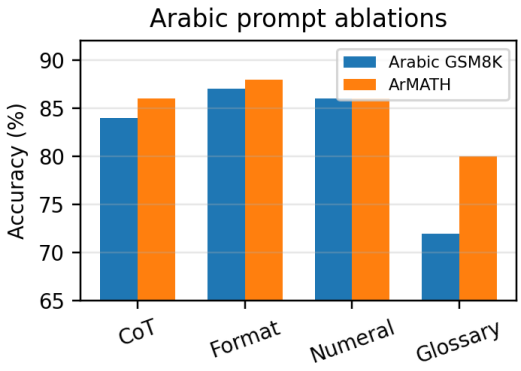


Figure 4: Accuracy of Arabic-specific prompt variants on Arabic GSM8K and ArMATH. Format-Controlled CoT achieves higher accuracy than standard CoT. Bilingual-Glossary CoT is the lowest-performing variant on both datasets.

Table 7: CTE confidence-path calibration.

Dataset	Path	N	Acc %	Calls
Ar.GSM8K	trusted_high	93	93.6	1.00
Ar.GSM8K	escalated_low	1	100.0	3.00
Ar.GSM8K	missing_conf.	6	33.3	3.00
En.GSM8K	trusted_high	97	97.9	1.00
En.GSM8K	missing_conf.	3	33.3	3.00
SVAMP	trusted_high	96	99.0	1.00
SVAMP	escalated_low	1	100.0	3.00
SVAMP	missing_conf.	3	66.7	3.00
ArMATH	trusted_high	92	93.5	1.00
ArMATH	missing_conf.	8	12.5	3.00

call. Accuracy by difficulty: 0.91 on Easy, 0.82 on Medium, and 0.79 on Hard. The Easy and Medium scores match those from the Arabic CoT (Easy $= 0.91$, Medium $= 0.85$).

We report LOPR-style as a diagnostic only. The qualitative takeaway is that routing the reasoning language to English does not preserve accuracy on the hardest Arabic problems. A rigorous evaluation requires paired Arabic-English problems run through the identical pipeline, which is left for future work.

8 Discussion

CTE is useful when CoT has headroom and high-confidence answers are reliable. Arabic GSM8K satisfies both conditions: CoT sits at 0.84, and trusted-high accuracy is 0.94. CTE produces a $+0.06$ gain at 1.14 calls. On SVAMP, the headroom is small, but the high-confidence accuracy is even higher (0.99), so CTE matches Self-Consistency at near single-call cost. On English GSM8K, the headroom is minimal (CoT 0.95), and CTE delivers cost reduction rather than accuracy gain. On ArMATH, the trusted-high accuracy is 0.93, but

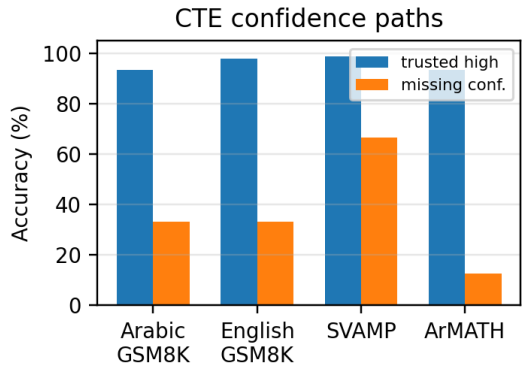


Figure 5: CTE confidence-path calibration. Trusted high-confidence answers are much more reliable than missing-confidence cases across all four datasets.

the model is still wrong often enough that Self-Consistency's exploration of alternative reasoning paths pays off.

Verification is consistently not cost-effective. It costs about three calls per question and does not produce the optimal accuracy on any of the four datasets. It performs worse than CoT on English GSM8K. On ArMATH, it matches CTE at 0.87 but requires 2.5 times the number of calls needed by CTE.

FAA-lite is a targeted intervention, not a general improvement. It improves performance on Arabic GSM8K (+0.03 relative to CoT, bootstrap CI $[0.01, 0.08]$) but not on the other three. This implies that errors on Arabic GSM8K may be due to answer flipping in the case of Arabic GSM8K. The errors in answering on other datasets are not in the verifying and revising phase.

Arabic prompt design matters more than expected. Format-Controlled CoT outperforms other methods on both Arabic datasets in terms of precision as well as the number of tokens, without any additional API calls. Bilingual-Glossary CoT performs considerably worse on both datasets. The gloss introduces additional instruction weight, which causes distraction from mathematics.

K=2 versus K=4. Table 1 shows that K=4 attains a validation accuracy of 92%, compared to 88% for K=2. K=4 requires twice the token cost for an increase of four points on a 50-question validation set, within the bounds of the bootstrapping noise band.

Toolkit framing. Our results do not support a single best prompt-only method. They support a toolkit:

- If only one call is allowed on Arabic, use Format-Controlled CoT.
- If the model follows confidence formatting and the application tolerates occasional escalation, use CTE.
- If maximum accuracy is more important than cost and the dataset is hard (such as ArMATH), Self-Consistency K=2 is justified.
- For Arabic systems with cross-lingual potential, LOPR-style prompting can be used as a diagnostic but not yet as a deployment method.

9 Error Analysis

Failure logs reveal four recurring categories.

Arithmetic errors. Correct reasoning structure but wrong computation; present in both languages and not addressable through formatting.

Operation-selection errors. Wrong arithmetic relationship between variables; a core failure mode in word problems (Patel et al., 2021).

Extraction errors (Arabic). Eastern Arabic numerals, spelled-out numbers, or extra numerals after the #### marker. Format-Controlled CoT eliminates most of these, explaining why it saves tokens while improving accuracy.

Confidence-format failures in CTE. Missing `CONFIDENCE:` lines route to escalation, but escalated accuracy remains low (Section 7). These are not low-confidence outputs, the model generated a correct reasoning chain but failed to include the required format line. The strict parser correctly refuses to trust these responses and sends them to escalation, but the two extra stochastic samples do not reliably converge to the correct answer when the model is already confused.

Verification errors. The model generates a plausible candidate but the separate verification stage selects the wrong final number. These cases are expensive because the user pays for both stages while the final answer remains wrong. They partly motivated the FAA and CTE designs, although neither fully eliminates them.

10 Limitations

One model. All experiments use `gpt-oss-20b`. Results reflect behavior on

that model under the NVIDIA API and are not a universal property of all LLMs. Different model families differ in confidence calibration and multilingual reasoning (Shi et al., 2022; Sui et al., 2025). Replicating the protocol on additional models is important future work.

100-question subsets. Final test sets contain 100 examples per dataset. This is enough for cost-accuracy patterns but small for fine-grained accuracy comparisons. We interpret close differences conservatively and rely on paired bootstrap and McNemar tests for significance claims.

Unpaired Arabic and English. The Arabic and English problem sets are not aligned item by item, which is why we report them separately. A paired design would require verified Arabic-English problem pairs. Machine-translating SVAMP into Arabic was rejected because translation quality would become a confound. Using ArMATH avoids that problem at the cost of cross-dataset comparison.

LOPR-style not on the same pipeline. The LOPR-style condition is reported from a test run, not the main 4-dataset evaluation. A reviewer-grade LOPR-style result requires the same pipeline with paired Arabic-English problems.

Binary confidence labels. CTE uses a two-level confidence label. The calibration analysis shows that low-confidence outputs are very rare. A graded numeric scale or a separate lightweight uncertainty prompt might fire the escalation path more usefully. Such extensions remain in the black-box setting if they use only text outputs.

FAA-lite is a one-call approximation. A full FAA design would use a separate commitment call followed by an anchored verification call. We used the one-call version to keep the call cost equal to CoT, which limits the comparison but matches the efficiency theme of the paper.

Benchmark contamination. Contamination cannot be ruled out for public benchmarks such as GSM8K and SVAMP (Cobbe et al., 2021; Patel et al., 2021). We therefore focus on relative method behavior under matched conditions rather than absolute model capability.

11 Conclusion

Prompt-only overthinking control is effective but dataset-dependent. CTE matches or exceeds Self-Consistency at substantially lower cost on Arabic GSM8K, English GSM8K, and SVAMP. Self-Consistency $K=2$ remains the strongest method on ArMATH. Arabic Format-Controlled CoT is the best single-call method on both Arabic datasets; Bilingual-Glossary CoT consistently hurts and should be avoided. Calibration analysis reveals that CTE succeeds mainly because the model's unprompted high-confidence claims are accurate, not because the gating mechanism actively detects uncertainty.

The practical recommendation is a toolkit, not a single prompt: Format-Controlled CoT for single-call Arabic deployment; CTE when the model follows confidence formatting and occasional escalation is acceptable; Self-Consistency $K=2$ when maximum accuracy on difficult Arabic problems justifies the additional cost. This finding warrants future research into more sophisticated confidence-scoring schemes, paired Arabic-English datasets, and replication across other model families to address the calibration limitations identified here.

References

Reem Alghamdi, Zhenwen Liang, and Xiangliang Zhang. 2022. ArMATH: A dataset for solving arabic math word problems. In *Proceedings of LREC*, pages 351–362, Marseille, France.

Karl Cobbe, Vineet Kosaraju, Mo Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *ArXiv*, abs/2110.14168.

Mohamed Ehab and Ali Hamdi. 2026. Adaptive multi-expert reasoning via difficulty-aware routing and uncertainty-guided aggregation. *arXiv preprint arXiv:2604.10335*.

Nicolas Fernandez, Alejandro Maldonado, Changho Kim, and Tatsunori B Hashimoto. 2025. RADAR: Reasoning-ability and difficulty-aware routing for reasoning language models. *arXiv preprint arXiv:2509.25426*.

Xinyi Hu, Lior Pick, Ho-fung Leung, and Farzan Farnia. 2025. PromptWise: Online learning for cost-aware prompt assignment in generative models. *arXiv preprint arXiv:2505.18901*.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, and

17 others. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Teaching models to express their uncertainty in words. *arXiv preprint arXiv:2205.14334*.

Omartificial-Intelligence-Space. 2025. Arabic gsm8k: Arabic grade school math dataset. https://huggingface.co/datasets/Omartificial-Intelligence-Space/Arabic-gsm8k.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP models really able to solve simple math word problems? In *Proceedings of NAACL-HLT*, pages 2080–2094.

Wanyi Qiu and Siddharth Srivastava. 2025. Batch prompting suppresses overthinking in reasoning models. *arXiv preprint arXiv:2511.04108*.

Yun Qu, Qi Wang, Yixiu Mao, Vincent Tao Hu, Björn Ommer, and Xiangyang Ji. 2026. *Can Prompt Difficulty be Online Predicted for Accelerating RL Fine-tuning of Reasoning Models?*, page 1240–1250. Association for Computing Machinery, New York, NY, USA.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, Dipanjan Das, and Jason Wei. 2022. Language models are multilingual chain-of-thought reasoners. *ArXiv*, abs/2210.03057.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Hanjie Chen, and Xia Hu. 2025. Stop overthinking: A survey on efficient reasoning for large language models. *Trans. Mach. Learn. Res.*, 2025.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D Manning. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. *arXiv preprint arXiv:2305.14975*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

Zheng Wei, Zhenghao Wang, Zhiyuan Liu, and Maosong Sun. 2025. Stop spinning wheels: Mitigating LLM overthinking via mining patterns for early reasoning exit. *arXiv preprint arXiv:2508.17627*.

Hao Wen, Xin Wu, Yue Sun, Feng Zhang, Liang Chen, Ji Wang, Yucheng Liu, Ya-Qin Zhang, and Ying Li. 2025. BudgetThinker: Empowering budget-aware LLM reasoning with control tokens. *arXiv preprint arXiv:2508.17196*.

Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C Schmidt. 2023. A prompt pattern catalog to enhance prompt engineering with ChatGPT. *arXiv preprint arXiv:2302.11382*.

Bowen Zhao, Berkay Kapusuzoglu, Karthikeyan Balasubramaniam, Sourav Sahu, Supriyo Chakraborty, and Genta Indra Winata. 2025. Optimizing reasoning efficiency through prompt difficulty prediction. *arXiv preprint arXiv:2511.03808*.

Shaer: Controlled Arabic Poetry Generation with Meter Subform and Semantic Conditioning

Ahmad Abbas **Tamara Fakih** **Nour Fakih** **Ammar Mohanna**
Department of Electrical and Computer Engineering
American University of Beirut, Beirut, Lebanon
{asa108,tmf14,nhf16}@mail.aub.edu am288@aub.edu.lb

Abstract

Classical Arabic poetry generation requires simultaneously satisfying semantic, linguistic, and fine-grained prosodic constraints. Existing systems typically control broad poetic attributes but do not jointly model semantic intent, meter subform, and poem length. We present SHAER, a controllable Classical Arabic poetry generation framework jointly conditioned on natural-language descriptions, meter subforms, and target hemistich counts. To support this task, we construct an enriched corpus of 116,032 classical Arabic poems derived from Ashaar, containing normalized meter-subform labels and automatically generated, validated semantic descriptions. We then adapt Yehia-7B using QLoRA-based supervised fine-tuning with a completion-only objective. Our evaluation combines automatic assessment of base-meter conformity, requested-subform adherence, and length control with three LLM judges, blinded human evaluation, and memorization analysis. SHAER achieves 95.17% base-meter accuracy, 91.75% poem-level meter-subform accuracy, and 83.40% exact count accuracy. Relative to its untuned foundation model, these results represent gains of 68.68, 57.77, and 38.93 percentage points, respectively; SHAER also attains the highest base-meter accuracy among all evaluated systems. Multi-LLM evaluation and a blinded human assessment of top-ranked outputs further indicate competitive semantic and literary quality. Finally, analysis of all 3,481 test generations finds no exact copies from the training corpus or paired source poems. Code, models, and datasets are publicly available.[1]

1 Introduction

Classical Arabic poetry is a central component of Arabic literary heritage, making controlled generation both a technically challenging problem and a culturally grounded Arabic NLP task. Unlike conventional language generation tasks, successful poetry generation must simultaneously satisfy semantic, stylistic, and prosodic constraints. Beyond producing fluent and coherent text, a generated poem must adhere to a prescribed meter while faithfully expressing the intended meaning. These requirements make Arabic poetry generation a particularly demanding controllable generation problem. A brief overview of Classical Arabic poetic structure, meter, and meter subforms is provided in Appendix A.

Despite recent progress in Arabic language generation and computational poetry, fine-grained control over generated Arabic poetry remains challenging. Existing generation settings largely operate over broad structural or categorical attributes, leaving important dimensions of poetic control insufficiently addressed. In particular, jointly controlling natural-language semantic intent, fine-grained meter subforms, and explicit poem length remains underexplored. This limits the ability to generate poems that simultaneously realize a desired meaning and satisfy precise prosodic and structural requirements.

To address these gaps, we introduce SHAER, a controllable Arabic poetry generation framework that jointly models semantic intent and poetic structure. By conditioning generation on natural-language descriptions, meter subforms, and target hemistich counts, SHAER enables fine-grained control over meaning, meter, and poem structure. Figure 1 presents representative poems generated by SHAER from prompts drawn from the test split of the released dataset.

Our contributions are as follows: (1) We introduce a controllable Arabic poetry generation task conditioned jointly on semantic descriptions, meter subforms, and requested poem length, enabling fine-grained control over semantic and structural properties. (2) We construct an enriched Arabic poetry corpus derived from Ashaar, comprising

[1]Code: https://github.com/AhmaddAbbass/Shaer. Models and datasets: https://huggingface.co/Shaer-AI.

Proceedings of The Fourth Arabic Natural Language Processing Conference (ArabicNLP 2026), pages 179–199
October 28-29, 2026 ©2026 Association for Computational Linguistics

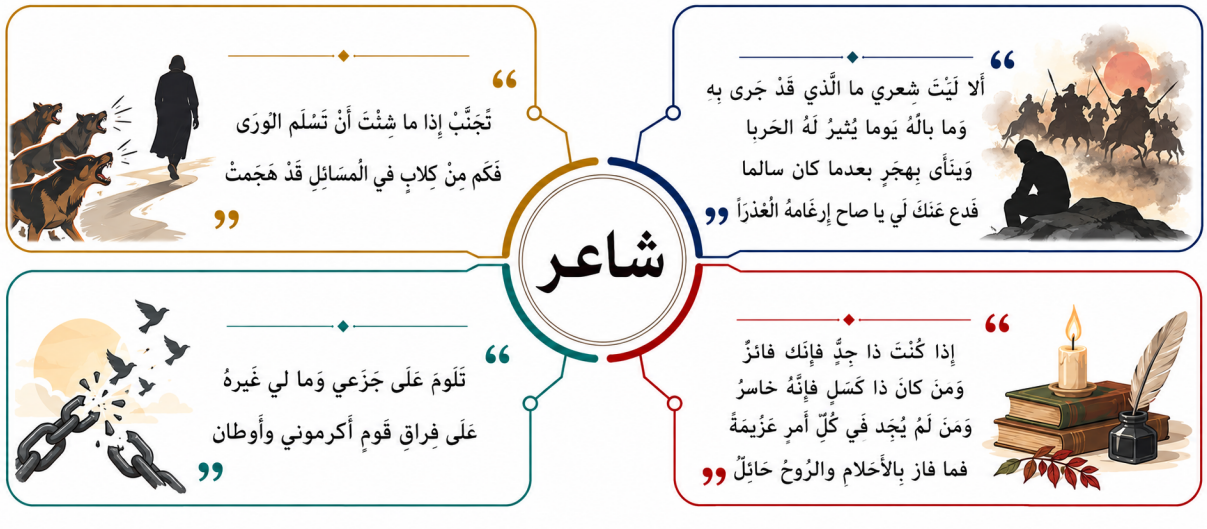


Figure 1: Example poems generated by Shaer. Each poem is generated from a semantic description, meter specification, and length constraint derived from a held-out test example.

116,032 classical poems annotated with normalized meter subforms and semantic descriptions, together with the complete data processing pipeline. (3) We develop SHAER, a controllable Arabic poetry generation model built by adapting Yehia-7B, a strong Arabic-focused language model, through parameter-efficient fine-tuning and explicit conditioning on semantic and prosodic constraints. (4) We introduce a multi-perspective evaluation framework combining automatic assessment of base-meter conformity, meter-subform adherence, and length control with three LLM judges, blinded human evaluation, and memorization and copying analysis.

Experimental results show that SHAER attains the highest base-meter accuracy among all evaluated systems and achieves high requested-subform adherence. Relative to its untuned foundation model, Yehia-7B, SHAER provides markedly stronger base-meter, meter-subform, and length control. Evaluation with three LLM judges and a blinded human assessment of top-ranked outputs further indicates that these gains in controllability are achieved while retaining competitive semantic and literary quality.

2 Related Work

2.1 Arabic Poetry Generation

Early neural approaches to Arabic poetry generation relied on task-specific architectures. Talafha and Rekabdar (2019) combined keyword-based semantic guidance with Bi-GRU generation and hierarchical attention over previously generated verses, while their later work incorporated phonetic CNN subword embeddings and keyword expansion to improve metrical conformity and semantic consistency (Talafha and Rekabdar, 2021).

Pretrained language models subsequently enabled transfer-based approaches. Beheitt and Ben Haj Hmida (2022) pretrained GPT-2 on Arabic text and fine-tuned it for poetry generation, while Abboushi and Azzeh (2023) fine-tuned AraGPT2 on a large poetry corpus with explicit meter and rhyme conditioning.

Recent work has expanded both resources and controllability. Ashaar (Alyafeai et al., 2023) introduced datasets and pretrained models for Arabic poetry analysis and conditional generation. Fanar-Diwan (FANAR Team et al., 2026) supports controllable classical Arabic poetry generation conditioned on meter, topic, era, poet, and rhyme. From an instruction-following perspective, Sadallah et al. (2026) introduced a large-scale resource spanning Modern Standard Arabic and multiple dialects and supporting poetry generation, continuation, revision, and analysis under user-specified constraints.

Building on this progression, SHAER offers a complementary approach to controllable classical Arabic poetry generation through joint conditioning on natural-language semantic descriptions, meter subforms, and explicit poem length.

2.2 Arabic Poetry Analysis and Evaluation

Arabic poetry analysis has extensively addressed meter identification. Al-Shaibani et al. (2020) proposed a character-based bidirectional recurrent model for classifying poetry into 14 meters without

traditional Arudi representation, while Abandah et al. (2022) developed recurrent models covering the 16 classical meters and prose, alongside automatic diacritization. We reproduce the latter meter-classification architecture to assess base-meter conformity.

For finer-grained prosody, Qarah (2024) introduced AraPoemBERT, a poetry-specific language model evaluated on several analysis tasks, including meter-subform classification. We reproduce this setting to assess subform adherence in controllable generation.

At the semantic level, Al Ghallabi et al. (2025) introduced Fann or Flop, a benchmark for LLM understanding of Arabic poetry across semantic, figurative, prosodic, and cultural dimensions. Our evaluation similarly adopts a multidimensional perspective, combining automatic assessment of meter, subform, and length control with LLM-based and human judgments of semantic and literary quality.

3 Dataset

3.1 Source Corpus

Our work builds on Ashaar (Alyafeai et al., 2023), a large-scale Arabic poetry resource containing 254,630 poems from multiple online repositories. Ashaar supports several poetry-related tasks and provides poem-level metadata including meter, theme, poet identity, era, and descriptions. However, its metadata organization and description coverage are not directly suited to controllable generation based on fine-grained poetic structure and semantic intent. Appendix B provides a detailed analysis of the original corpus.

3.2 Corpus Construction

We construct an enriched corpus for controllable Arabic poetry generation through multi-stage filtering and normalization. We retain only classical Arabic poetry in the canonical Khalilian meters, removing colloquial or non-classical forms, malformed poems, and entries with invalid verse structure. To align with the 1,024-token maximum sequence length used for supervised fine-tuning, we exclude poems exceeding twenty verses. A post-tokenization audit using the final instruction format found that only 4 of 109,070 training sequences (0.0037%) exceeded this limit, by 5–25 tokens, with none in the validation or test sets. Finally, we remove insufficiently represented meter categories. Appendix C details the cleaning, normalization, and filtering procedures.

3.3 Meter Subform Annotation and Validation

All 116,032 retained poems inherit their base-meter labels from Ashaar. Explicit meter-subform metadata was available for 5,106 poems (4.4%); we normalized and retained these labels without independent re-annotation. The remaining 110,926 poems (95.6%) were assigned the default تام (Complete) label, corresponding to the canonical realization of their base meter in classical Arabic prosody.

We assessed this default assignment in two ways. First, an author formally trained in Arabic prosody manually inspected 1,000 default-labeled poems, confirming the Complete form in 97.2% of cases. Second, we reproduced the AraPoemBERT meter-subform classification setting (Qarah, 2024), achieving 97.38% accuracy compared with the originally reported 97.79%. Applied to all 110,926 default-labeled poems, the classifier predicted Complete for 103,748 (93.53%). Agreement exceeded 98.7% for nine of thirteen base meters, with disagreements concentrated in four meters affected by classifier limitations in class coverage or reported per-class performance. We use these predictions only to assess the default-assignment convention, not to replace corpus labels; complete per-meter results and discussion are provided in Appendix D.1.

After normalization and default assignment, all poems have meter-subform annotations, yielding 21 distinct meter–subform combinations whose distribution is reported in Appendix D.2.

3.4 Semantic Description Generation

Ashaar descriptions are missing or insufficiently informative for much of the corpus, limiting their use as semantic conditioning signals (Appendix B). We therefore generate a semantic description for every poem.

We compare three generation configurations differing in model and prompting strategy. For each, three native Arabic-speaking annotators independently evaluate a stratified sample of 600 descriptions using a predefined Pass/Fail rubric. The best-performing configuration, based on Qwen3.5-35B-A3B (Qwen Team, 2026a), achieves a 92.8% pass rate and is selected for corpus-wide generation. Configuration and human-evaluation details are provided in Appendix E.1.

The selected configuration uses deterministic decoding to generate concise descriptions capturing

Meter	Sub-form	Requested Hemistichs	Description	Poem Hemistichs
الوافر	تام	4	القصيدة تتحدث عن ليلٍ طويلٍ كليل يوم القيامة، جعل السهر مستحيلاً، لدرجة أن الشاعر كاد يظن في سهادته أن الله لم يخلق نهارًا.	وَلَيل مثل يَوم الحَشر طولاً مَنَعَت بِهِ مِن السَهر القَرارا وَكدت أَظُنُّ فيهِ مِن سُهادي بِأَن اللَه لَم يَخلق نَهارا

Table 1: Example Shaer dataset instance with its conditioning signals: base meter, meter subform, target hemistich count, and semantic description.

each poem's meaning, imagery, and tone while excluding references to meter, rhyme, or length. The resulting descriptions undergo automatic structural validation and corpus-wide quality checks. Appendices E.2 and E.3 provide full prompting, decoding, and verification details.

3.5 Final Dataset Statistics

The final corpus pairs each poem with its base meter, meter subform, target hemistich count, and generated semantic description, jointly supporting control over poetic form and semantic content. It comprises 116,032 poems spanning 13 classical meters and 21 meter–subform combinations, providing substantially finer-grained metrical supervision than the coarse meter categories commonly used in prior Arabic poetry generation systems. Appendix F summarizes the dataset characteristics, while Table 1 illustrates the enriched metadata and controlled generation format.

3.6 Data Splits

We partition the corpus into train (94%), validation (3%), and test (3%) splits, stratified by a composite key of base meter, meter subform, and length bucket to preserve poetic-structure distributions across partitions. Length buckets are defined by poem length as 1–3, 4–6, 7–10, and 11–20 verses.

4 Methodology

4.1 Task Formulation

We formulate Arabic poetry generation as a controllable generation task. Given a conditioning tuple

$$x = (m, s, n, d),$$

where m is the base meter, s the meter subform, n the requested hemistich count, and d a natural-language description of the desired content, the model generates

$$y = (y_1, \ldots, y_T),$$

where y_t is the t-th generated token and T the total number of generated tokens. The objective is to model $P(y \mid x)$ such that the generated poem satisfies the specified semantic, prosodic, and length constraints.

4.2 Base Model Selection

Model selection is informed by AraLingBench (Zbib et al., 2026), a human-annotated benchmark of Arabic linguistic competence across grammar, morphology, spelling, syntax, and reading comprehension, capabilities closely aligned with classical Arabic poetry generation.

Yehia-7B (Navid-AI, 2025), a 7B-parameter Arabic-focused decoder-only language model, achieved the strongest overall performance among the evaluated models. We therefore select Navid-AI/Yehia-7B-preview[2] as the foundation model for SHAER. Its Apache-2.0 license also facilitates public release of adapters and derived resources.

4.3 Supervised Fine-Tuning

We adapt Yehia-7B through supervised fine-tuning on the enriched corpus training split using a completion-only causal language modeling objective. Given prompt x and target completion y, the loss is computed only over poem tokens:

$$\mathcal{L}(\theta) = -\sum_{t \in \mathcal{C}} \log p_\theta(y_t \mid x, y_{<t}), \qquad (1)$$

where $\mathcal{C}$ denotes completion-token positions; prompt tokens are masked from the optimization objective.

[2]Hugging Face model repository handle.

To mitigate dataset imbalance, we use weighted sampling over groups defined by (base meter, meter subform, length bucket):

$$w(g) \propto \text{count}(g)^{-0.6}.$$

The exponent 0.6 provides moderate rebalancing between unweighted (0) and full inverse-frequency (1) sampling, increasing representation of rare groups without excessive oversampling.

Fine-tuning uses QLoRA (Dettmers et al., 2023), with the base model loaded in 4-bit NF4 precision, double quantization, and bfloat16 computation. LoRA adapters are applied to all linear layers with rank $r = 64$, $\alpha = 128$, dropout 0.05, and RS-LoRA enabled.

Appendix G provides complete training hyperparameters, runtime settings, and instruction templates.

5 Experimental Setup

5.1 Baselines

We evaluate SHAER against three Arabic poetry generation baselines spanning foundation and poetry-specific models:

- **Yehia-7B** (Navid-AI, 2025), the unfine-tuned foundation model underlying SHAER.
- **Ashaar** (Alyafeai et al., 2023), a poetry generation model trained on the Ashaar corpus.
- **Fanar-Diwan** (FANAR Team et al., 2026), a classical Arabic poetry model optimized for prosodic and stylistic constraints.

Yehia-7B provides the cleanest controlled comparison, sharing SHAER's architecture, tokenizer, and instruction format but lacking task-specific fine-tuning. Ashaar and Fanar-Diwan serve as system-level comparisons under their native conditioning interfaces, which do not support SHAER's full set of semantic, subform, and length controls. Earlier systems in Section 2 are not directly evaluated due to differing generation and evaluation settings and, where applicable, unavailable public checkpoints or inference implementations.

All models are evaluated on the 3,481-instance held-out test split. Yehia-7B uses the same instruction-style prompt as SHAER, while Ashaar and Fanar-Diwan use their native conditioning formats. Table 2 summarizes conditioning capabilities; Appendices H and I detail input representations and decoding settings, respectively.

5.2 Evaluation Dimensions

Because Arabic poetry quality cannot be captured by a single score, we evaluate generated poems using complementary structural and semantic-literary metrics. Before structural evaluation, outputs are deterministically normalized and segmented into hemistichs and bayts by pairing consecutive hemistichs; any incomplete final hemistich is excluded from bayt-level evaluation. Appendix J provides the full normalization and parsing rules.

Automatic Meter Evaluation. We evaluate meter using an external Arabic meter classifier based on the deep bidirectional recurrent architecture of Abandah et al. (2022). Operating at the verse level, it produces a meter prediction and probability distribution over the supported classes. Appendix K provides complete implementation and evaluation details.

We report two complementary metrics. *Meter Accuracy* measures exact adherence to the requested meter by majority-voting verse-level predictions into a poem-level prediction and computing the percentage matching the target meter.

Meter Conformity provides a softer confidence-based measure of metrical adherence. Let m^* denote the requested meter and $p_i(m^*)$ its predicted probability for verse i. Because meter is a poem-level property, we aggregate verse-level probabilities using the geometric mean:

$$C_{\text{poem}} = \left(\prod_{i=1}^{n} p_i(m^*) \right)^{1/n}, \qquad (2)$$

where n is the number of verses. The geometric mean penalizes metrically inconsistent verses, providing a more faithful estimate of poem-level conformity than an arithmetic average. Final Meter Conformity is the arithmetic mean across all generated poems.

Automatic Meter Subform Evaluation. We evaluate fine-grained prosodic adherence using our reproduced AraPoemBERT 25-class meter-subform classifier (Qarah, 2024), which operates at the bayt level. We report *Bayt-Level Subform Accuracy* and *Poem-Level Subform Accuracy*; the latter averages class-probability vectors across valid bayts and selects the highest-scoring class. The evaluator covers 3,379 of 3,481 test targets (97.07%), excluding targets outside its class inventory. Appendix L details classifier reconstruction, input rep-

Model	Meter	Subform	Rhyme	Hemistich Count	Description
Yehia-7B	✓†	✓†	×	✓†	✓†
Ashaar	✓	×	✓	×	×
Fanar-Diwan	✓	×	✓	×	×
SHAER	✓	✓	×	✓	✓

Table 2: Conditioning capabilities of the evaluated models. † denotes variables supplied through prompting rather than learned as native conditioning fields.

resentation, sequence-length configuration, and unsupported meter-subform classes.

For SHAER and Yehia-7B, these metrics measure adherence to the requested subform. Because Ashaar and Fanar-Diwan do not support subform conditioning, their outputs are evaluated against the corresponding source poem's subform, measuring reference-subform agreement rather than controllability.

Hemistich Count Evaluation. The second structural dimension evaluates adherence to the requested hemistich count using two complementary metrics.

Count Accuracy measures exact count matching as the percentage of generated poems whose hemistich count equals the requested count.

Count Adherence (CA) provides a softer measure of length conformity. Let H_g and $H_r > 0$ denote the generated and requested hemistich counts, respectively:

$$\mathrm{CA} = \max\left(0,\, 1 - \frac{|H_g - H_r|}{H_r}\right). \qquad (3)$$

The score equals 1 for an exact match and decreases with relative count discrepancy, reaching 0 when the discrepancy is at least as large as the requested count. Normalization by H_r makes the metric more sensitive to absolute deviations for shorter requested poems. Final Count Adherence is the arithmetic mean across all generated poems.

Both metrics are reported only for models supporting hemistich-count conditioning, namely SHAER and Yehia-7B.

LLM-Based Evaluation. Semantic and literary quality are evaluated using three LLM judges: Qwen3-235B-A22B (Qwen Team, 2025), Qwen 3.7 Max (Qwen Team, 2026b), and GPT-5.6 Terra (OpenAI, 2026). Qwen3-235B-A22B was applied to all 3,481 held-out outputs from each system, while Qwen 3.7 Max and GPT-5.6 Terra evaluate a shared subset of 2,500 test instances across the four systems. Following recent work on LLM-based evaluation of open-ended generation, we assess five dimensions: *Description Adherence*, *Meaning*, *Fluency*, *Coherence*, and *Poeticness*. Table 3 defines the evaluation dimensions.

Metric	Definition
Description Adherence	Faithfulness to the conditioning description.
Meaning	Clarity and semantic depth of the poem.
Fluency	Linguistic correctness and naturalness of Arabic.
Coherence	Continuity and unity across the poem.
Poeticness	Literary quality beyond meter, including imagery and expression.

Table 3: LLM-based evaluation dimensions.

All three judges use identical dimension-specific rubrics. Each dimension is assessed independently on a strict 1–5 integer scale, where higher scores indicate better performance. Description Adherence is evaluated only for description-conditioned models, namely SHAER and Yehia-7B. Scores are first averaged over the available valid responses separately for each judge and system. The equal-weight mean of the three judge-level means is reported in the main results. To reduce evaluation variance, all judgments use deterministic decoding (temperature=0). Full evaluation rubrics, decoding settings, coverage details, and judge-specific results are provided in Appendix M.

Human Evaluation. We complement the LLM-based evaluation with a blinded human assessment of top-ranked outputs from all four systems. We evaluate Meaning, Fluency, Coherence, and Poeticness using the same 1–5 rubrics as the LLM judges, excluding Description Adherence because it is not applicable to Ashaar and Fanar-Diwan. For each system, we select the five highest-ranked outputs within the shared instance subset based on aggregate multi-LLM scores, yielding 20 poems. Three authors independently rate the selected poems; all are native Arabic speakers, and one additionally has

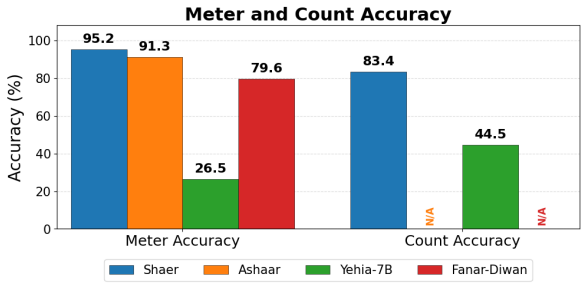


Figure 2: Meter accuracy and count accuracy.

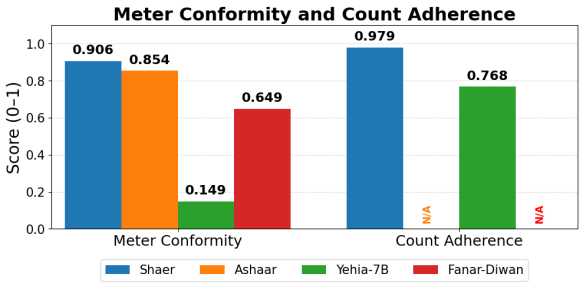


Figure 3: Meter conformity and count adherence scores.

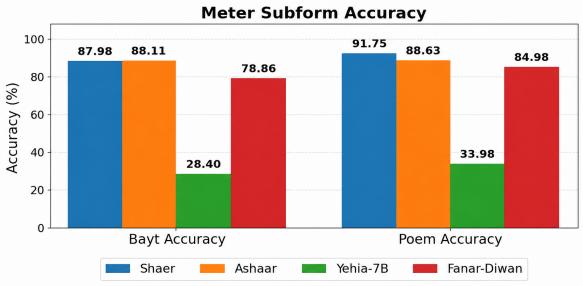


Figure 4: Bayt-level and poem-level meter-subform accuracy.

formal training in Arabic prosody and an academic minor in Arabic. Each reported system–dimension score averages 15 ratings across the five poems and three annotators. Further details on selection, blinding, evaluation procedure, and inter-annotator agreement are provided in Appendix N.

Memorization and Copying Evaluation. We additionally assess whether SHAER reproduces content from its training data or from the held-out poems underlying the test instances. Each of the 3,481 generated poems is compared against all 109,070 poems in the SFT training set and against its corresponding held-out source poem after deterministic text normalization.

We measure three complementary indicators of textual overlap: *Exact Match*, which identifies identical normalized poems; *Longest Shared Span*, defined as the longest contiguous sequence of words shared between a generation and a comparison poem; and *5-Gram Jaccard Similarity*, which measures overlap between their sets of word 5-grams. Together, these metrics capture exact reproduction, localized phrase-level reuse, and broader lexical overlap, respectively.

6 Results and Discussion

Appendix O presents representative poems generated by SHAER, together with their conditioning signals and corresponding meter conformity scores. The remainder of this section focuses on quantitative evaluation.

6.1 Formal Control Evaluation

Figures 2 and 3 summarize meter and length-control performance.

SHAER achieves the highest base-meter accuracy (95.2%) and meter-conformity score (0.906), outperforming Ashaar (91.3%, 0.854), Fanar-Diwan (79.6%, 0.649), and Yehia-7B (26.5%, 0.149). It also substantially exceeds Yehia-7B in exact count accuracy (83.4% vs. 44.5%) and count adherence (0.979 vs. 0.768). Because Yehia-7B shares the same foundation model, these differences demonstrate the stronger meter and length control achieved by the jointly conditioned and fine-tuned SHAER system.

Appendix P reports results by base meter and requested-length bucket. SHAER achieves the highest accuracy for 12 of the 13 meters and maintains 93.0–97.0% base-meter accuracy across length buckets. Its exact count accuracy decreases from 98.4% for poems of 1–3 verses to 29.0% for poems of 11–20 verses, but remains substantially above Yehia-7B in every bucket, indicating that longer-form length control remains challenging.

Figure 4 evaluates fine-grained meter-subform control. Among models explicitly conditioned on a target subform, SHAER achieves a bayt-level accuracy of 87.98% and a poem-level accuracy of 91.75%, substantially outperforming Yehia-7B, which obtains bayt-level and poem-level accuracies of 28.40% and 33.98%, respectively. Since both systems share the same foundation model and receive the same target subform, this comparison demonstrates the stronger fine-grained prosodic control achieved by the jointly conditioned and fine-tuned SHAER system.

Since Ashaar and Fanar-Diwan do not support subform conditioning, their scores instead measure

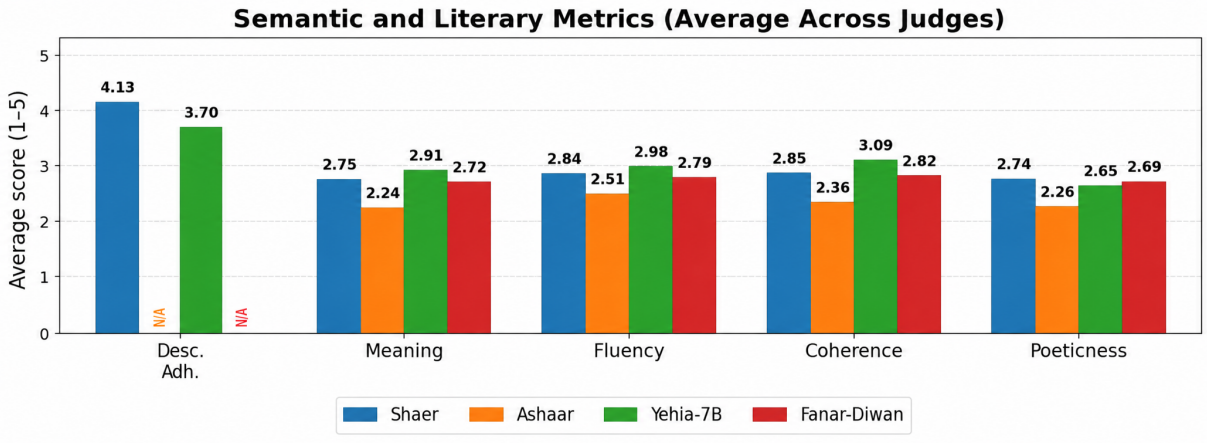


Figure 5: Equal-weight mean semantic and literary scores across three LLM judges. Description adherence applies only to description-conditioned systems.

agreement with the corresponding source poem's subform. Ashaar achieves 88.11% bayt-level and 88.63% poem-level agreement, while Fanar-Diwan achieves 78.86% and 84.98%, respectively. SHAER attains the highest poem-level accuracy overall, while its bayt-level accuracy is comparable to Ashaar's reference-subform agreement. These results should be interpreted according to the distinct conditioning settings. Additionally, rare subforms are more frequently classified as the dominant Complete (تام) form; accordingly, evaluation is restricted to target classes represented in the evaluator's 25-class inventory.

6.2 Semantic and Literary Quality

Figure 5 summarizes semantic and literary quality across the three LLM judges, with judge-specific results reported in Appendix M. Across the four dimensions shared by all systems, SHAER consistently exceeds the two poetry-specific baselines, Ashaar and Fanar-Diwan. The differences relative to Fanar-Diwan are small but consistent across Meaning (2.75 vs. 2.72), Fluency (2.84 vs. 2.79), Coherence (2.85 vs. 2.82), and Poeticness (2.74 vs. 2.69), while Ashaar scores lower across all four dimensions. Since these systems use different native conditioning interfaces, these results represent system-level comparisons rather than controlled comparisons of individual modeling choices.

Compared with its unfine-tuned foundation model, Yehia-7B, SHAER achieves higher cross-judge means for Description Adherence (4.13 vs. 3.70) and Poeticness (2.74 vs. 2.65), whereas Yehia-7B scores higher in Meaning, Fluency, and Coherence. Judge-specific results show the same direction for Description Adherence, Meaning, Coherence, and Poeticness, while Fluency is judge-dependent. Together with the structural results, these findings indicate that task-specific adaptation substantially improves controllability and description adherence while maintaining competitive semantic and literary quality.

System	Mean.	Flu.	Coh.	Poet.
SHAER	4.00	**4.20**	**4.20**	**4.30**
Yehia-7B	**4.13**	3.50	3.80	3.30
Fanar-Diwan	3.80	3.70	3.80	3.50
Ashaar	3.10	3.00	3.60	2.40

Table 4: Blinded human evaluation of five top-ranked poems per system. Scores are means over 15 ratings (five poems × three annotators) on a 1–5 scale.

Human Evaluation. Table 4 reports the complementary human evaluation results. SHAER receives the highest ratings for Fluency (4.20), Coherence (4.20), and Poeticness (4.30), while Yehia-7B receives the highest Meaning rating (4.13, compared with 4.00 for SHAER). Fanar-Diwan receives intermediate ratings, whereas Ashaar receives the lowest mean ratings across all four dimensions.

The human assessment partially aligns with the multi-LLM evaluation: both favor SHAER over Yehia-7B on Poeticness and Yehia-7B over SHAER on Meaning, while the human annotators additionally favor SHAER on Fluency and Coherence. Given that this evaluation targets top-ranked outputs rather than a representative test-set sample, these findings provide complementary evidence about the systems' strongest identified generations rather than their average performance.

6.3 Memorization and Copying Analysis

No exact normalized copies were found among the 3,481 SHAER generations. One generation (0.0287%) shared at least 13 consecutive words with a training poem, and four (0.1149%) with their paired held-out sources; none shared 30 or more words in either comparison. Median word 5-gram Jaccard similarity was 0.0 for both comparisons, with maxima of 0.119 and 0.220, respectively. Manual inspection revealed localized reuse or paraphrasing rather than full-poem reproduction. We therefore find no evidence of systematic memorization or wholesale copying, although rare source-associated phrase reuse occurs.

7 Conclusion

We presented SHAER, a controllable Classical Arabic poetry generation framework jointly conditioned on natural-language descriptions, meter subforms, and target hemistich counts. To support this task, we constructed an enriched corpus of 116,032 classical Arabic poems derived from Ashaar, containing normalized meter-subform labels and automatically generated, validated semantic descriptions. Building on this resource, we adapted Yehia-7B through QLoRA-based supervised fine-tuning with a completion-only objective.

Across the 3,481 held-out test instances, SHAER achieves 95.17% base-meter accuracy, 91.75% poem-level meter-subform accuracy, and 83.40% exact count accuracy, representing substantial gains over its untuned foundation model. It also attains the highest base-meter accuracy among all evaluated systems. Evaluation with three LLM judges indicates that SHAER exceeds the poetry-specific comparison systems across the shared semantic and literary dimensions, while its comparison with Yehia-7B reveals a trade-off between stronger controllability and performance on some unconstrained literary-quality dimensions. A blinded human assessment of top-ranked outputs further supports the model's competitive literary quality. In addition, memorization analysis finds no exact normalized copies from either the training corpus or the paired held-out source poems.

Together, these findings demonstrate that SHAER provides reliable joint control over semantic intent, fine-grained prosodic form, and poem length while preserving competitive semantic and literary quality. The released code, models, annotations, and evaluation resources provide a foundation for further research on controllable Arabic literary generation.

Future work includes explicit rhyme conditioning and evaluation, component-wise ablations, broader human evaluation on representative samples, improved handling of rare meter subforms and longer poems, and extension to additional Arabic poetic genres.

8 Limitations

Despite the strong results achieved by SHAER, several limitations remain. First, this work focuses on Classical Arabic poetry written in the 13 canonical meters retained after corpus filtering. Consequently, the dataset and model do not cover free verse, prose poetry, colloquial poetry, or other contemporary poetic forms. Performance is also less certain for rare meter subforms and longer requested poems because of their limited representation.

Second, 95.6% of the corpus lacked explicit meter-subform metadata and was assigned the canonical Complete (تام) form by default. Although manual inspection and classifier-based validation support this convention, these labels remain partly heuristic.

Third, semantic descriptions are generated automatically rather than written by human annotators. Despite structural validation, corpus-wide quality checks, and human evaluation of sampled descriptions, some residual semantic noise may remain.

Fourth, the evaluation relies partly on learned meter classifiers and LLM judges. Although these evaluators were validated and three LLM judges were used, their scores remain approximations of expert human judgment. The complementary human evaluation is also limited to selected top-ranked outputs rather than a representative test-set sample.

Finally, rhyme is learned implicitly but is neither explicitly controlled nor quantitatively evaluated. Because the experiments do not include component-wise ablations, the observed improvements should be attributed to the jointly conditioned and fine-tuned SHAER system rather than to any individual control mechanism.

References

Gheith A. Abandah, Mohammed Z. Khedher, Mohammad R. Abdel-Majeed, Hamdi M. Mansour, Salma F. Hulliel, and Lara M. Bisharat. 2022. Classifying and diacritizing arabic poems using deep recurrent neural

networks. *Journal of King Saud University - Computer and Information Sciences*, 34(6):3775–3788.

Asaad Abbas. 2001. The metrical structure of classical arabic poetry. *Studia Arabistyczne i Islamistyczne*, 8:5–52.

Omar Abboushi and Mohammad Azzeh. 2023. Toward fluent arabic poem generation based on fine-tuning AraGPT2 transformer. *Arabian Journal for Science and Engineering*, 48(8):10537–10549.

Wafa Al Ghallabi, Ritesh Thawkar, Sara Ghaboura, Ketan Pravin More, Omkar Thawakar, Hisham Cholakkal, Salman Khan, and Rao Muhammad Anwer. 2025. Fann or flop: A multigenre, multiera benchmark for Arabic poetry understanding in LLMs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 20224–20244, Suzhou, China. Association for Computational Linguistics.

Maged S. Al-Shaibani, Zaid Alyafeai, and Irfan Ahmad. 2020. Meter classification of arabic poems using deep bidirectional recurrent neural networks. *Pattern Recognition Letters*, 136:1–7.

Maytham Alabbas, Zainab A. Khalaf, and Khashan M. Khashan. 2014. BASRAH: An automatic system to identify the meter of arabic poetry. *Natural Language Engineering*, 20(1):131–149.

Zaid Alyafeai, Maged S. Al-Shaibani, and Moataz Ahmed. 2023. Ashaar: Automatic analysis and generation of Arabic poetry using deep learning approaches. *arXiv*, abs/2307.06218.

Abd al-Aziz Atiq. 1987. *ʿIlm al-ʿArūḍ wa-al-Qāfiyah* (علم العروض والقافية). Dar al-Nahda al-Arabiyya (دار النهضة العربية), Beirut.

Mohamed El Ghaly Beheitt and Moez Ben Haj Hmida. 2022. Automatic arabic poem generation with GPT-2. In *Proceedings of the 14th International Conference on Agents and Artificial Intelligence*, volume 2, pages 366–374. SciTePress.

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.

FANAR Team, Ummar Abbas, Mohammad Shahmeer Ahmad, Minhaj Ahmad, Abdulaziz Al-Homaid, Anas Al-Nuaimi, Enes Altinisik, Ehsaneddin Asgari, Sanjay Chawla, Shammur Chowdhury, Fahim Dalvi, Kareem Darwish, Nadir Durrani, Mohamed Elfeky, Ahmed Elmagarmid, Mohamed Eltabakh, Asim Ersoy, Masoomali Fatehkia, Mohammed Qusay Hashim, and 18 others. 2026. Fanar 2.0: Arabic generative AI stack. *arXiv*, abs/2603.16397.

Navid-AI. 2025. Yehia 7b preview.

OpenAI. 2026. Gpt-5.6 terra. OpenAI API Documentation. Accessed: 2026-09-16.

Faisal Qarah. 2024. AraPoemBERT: A pretrained language model for Arabic poetry analysis. *arXiv*, abs/2403.12392.

Qwen Team. 2025. Qwen3 technical report. *arXiv*.

Qwen Team. 2026a. Qwen3.5: Towards native multimodal agents. Accessed: 2026-04-27.

Qwen Team. 2026b. Qwen3.7. Qwen Blog. Accessed: 2026-09-16.

Abdelrahman Sadallah, Kareem Elozeiri, Mervat Abassy, Rania Elbadry, Mohamed Anwar, Abed Alhakim Freihat, Preslav Nakov, and Fajri Koto. 2026. Instruction-guided poetry generation in Arabic and its dialects. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 38761–38780, San Diego, California, United States. Association for Computational Linguistics.

Sameerah Talafha and Banafsheh Rekabdar. 2019. Arabic poem generation with hierarchical recurrent attentional network. In *2019 IEEE 13th International Conference on Semantic Computing (ICSC)*, pages 316–323. IEEE.

Sameerah Talafha and Banafsheh Rekabdar. 2021. Poetry generation model via deep learning incorporating extended phonetic and semantic embeddings. In *2021 IEEE 15th International Conference on Semantic Computing (ICSC)*, pages 48–55. IEEE.

Mohamad Bilal Zbib, Hasan Abed Al Kader Hammoud, Ammar Mohanna, Nadine Rizk, Fatima Karnib, Sina Moukaled, and Bernard Ghanem. 2026. Aralingbench: A human-annotated benchmark for evaluating arabic linguistic capabilities of large language models. In *Proceedings of the 2nd Workshop on NLP for Languages Using Arabic Script*, pages 385–393, Rabat, Morocco. Association for Computational Linguistics.

Appendix

A Background on Classical Arabic Poetry

A.1 Poetic Structure

Classical Arabic poetry follows a highly structured form governed by meter and rhyme. A poem (قصيدة, *qasida*) consists of verses (أبيات, *abyat*; singular بيت, *bayt*), each typically divided into two hemistichs (شطرين, *shatrayn*): الصدر (*sadr*, first hemistich) and العجز (*ajuz*, second hemistich). Hemistichs within a poem typically conform to a single meter (بحر, *bahr*), while the verses share a common rhyme (قافية, *qafiyah*), realized at the end of the second hemistich (Abbas, 2001). These constraints require poetry generation to maintain both semantic and prosodic consistency.

English Name	Arabic Name	Meter Key
Taweel	الطويل	فعولن مفاعيلن فعولن مفاعيلن
Madeed	المديد	فاعلاتن فاعلن فاعلاتن
Baseet	البسيط	مستفعلن فاعلن مستفعلن فاعلن
Wafer	الوافر	مفاعلتن مفاعلتن فعولن
Kamel	الكامل	متفاعلن متفاعلن متفاعلن
Hazaj	الهزج	مفاعيلن مفاعيلن
Ramal	الرمل	فاعلاتن فاعلاتن فاعلاتن
Rajaz	الرجز	مستفعلن مستفعلن مستفعلن
Sare'e	السريع	مستفعلن مستفعلن فاعلن
Munsareh	المنسرح	مستفعلن مفعولاتُ مستفعلن
Khafeef	الخفيف	فاعلاتن مستفعلن فاعلاتن
Mujtath	المجتث	مستفعلن فاعلاتن
Mutakareb	المتقارب	فعولن فعولن فعولن فعولن
Mutadarak	المتدارك	فاعلن فاعلن فاعلن فاعلن
Muqtadheb	المقتضب	فاعلات مفتعلن
Mudhare'e	المضارع	مفاعيلن فاعلاتن

Table 5: The sixteen canonical Khalilian meters and their corresponding meter keys (Alabbas et al., 2014).

A.2 Meter and Meter Subforms

Arabic prosody (علم العروض, *ʿilm al-ʿarud*) describes the metrical organization of classical Arabic poetry. Each meter (بحر, *bahr*) is characterized by an ordered sequence of recurring metrical feet (تفاعيل, *tafaʿil*) (Abbas, 2001). For example, the canonical pattern of Rajaz (الرجز) is مستفعلن مستفعلن مستفعلن in each hemistich. Table 5 summarizes the sixteen canonical Khalilian meters and their meter keys.

A base meter can admit multiple valid realizations through systematic modifications of its canonical structure (Atiq, 1987). We refer to these finer-grained realizations as *meter subforms*. Common examples include Complete (تام), Catalectic (مجزوء), Hemistich (مشطور), and Severely Truncated (منهوك); these are illustrative rather than exhaustive, and valid subforms vary across meters. Consequently, two poems can share the same base meter while following different subforms. Table 6 illustrates this distinction using Rajaz.

B Analysis of the Original Ashaar Corpus

The original Ashaar corpus contains 254,630 poems and 12 metadata fields per poem. However, substantial inconsistencies in metadata coverage, meter annotations, description availability, poem length, and textual quality motivated the cleaning and filtering procedures described in Section 3. Table 7 summarizes the key statistics.

C Data Cleaning and Preprocessing

We apply a four-stage pipeline to remove structural, orthographic, and metadata inconsistencies from the original Ashaar corpus. First, the corpus is restricted to classical Arabic poetry written in the sixteen canonical Khalilian meters. Poems labeled as colloquial or non-classical (عامي, شعبي), as well as poems lacking meter annotations, are discarded. Second, poems with malformed verse structures, empty hemistichs, or hemistichs containing Latin characters, digits, or URLs are removed. Internal elongation (تطويل (ـ)) characters are also stripped, while edge elongation and all diacritics are preserved. Third, poems exceeding twenty verses are excluded to align the corpus with the supervised fine-tuning maximum sequence length of 1,024 tokens. A post-tokenization audit using the final instruction format found that only 4 of the 109,070 training sequences (0.0037%) still exceeded this limit, by 5–25 tokens, while no validation or test sequence exceeded it. Finally, meter annotations are normalized to the canonical meter inventory, and base meters represented by fewer than 500 poems are removed. Figure 6 summarizes the resulting counts at each stage.

D Meter Subform Details

D.1 Default-Label Validation

We first manually assessed the default Complete assignment on a sample of 1,000 poems. An author with formal training in Arabic prosody inspected the sampled poems and confirmed the Complete

Name	# Feet	Metrical Feet Pattern	Definition	Example
Complete تام	6	مُسْتَفْعِلُنْ مُسْتَفْعِلُنْ مُسْتَفْعِلُنْ مُسْتَفْعِلُنْ مُسْتَفْعِلُنْ مُسْتَفْعِلُنْ	Retains the full three-foot pattern in each of the two hemistichs.	لَا خَيْرَ فِيمَنْ كَفَّ عَنَّا شَرَّهُ إِنْ كَانَ لَا يُرْجَى لِيَوْمٍ خَيْرُهُ
Truncated مجزوء	4	مُسْتَفْعِلُنْ مُسْتَفْعِلُنْ مُسْتَفْعِلُنْ مُسْتَفْعِلُنْ	Retains two feet in each of the two hemistichs.	فَارفِقْ بِصَبٍّ ذِي جَوى مِنْ حَرِّهِ ذَابَ الحَشَى
Single-Hemistich مشطور	3	مُسْتَفْعِلُنْ مُسْتَفْعِلُنْ مُسْتَفْعِلُنْ	Consists of a single three-foot verse.	إِنَّ ابْنَتِي بَيْضاءُ مِنْ بِيضٍ زُهُرْ
Exhausted منهوك	2	مُسْتَفْعِلُنْ مُسْتَفْعِلُنْ	Consists of a single two-foot verse.	يَا لَيْتَنِي فِيهَا جَذَعْ

Table 6: Illustrative meter subforms of Rajaz (بحر الرجز), showing the number of metrical feet per verse, their metrical patterns, and corresponding examples. The set is not exhaustive.

Statistic	Value
Total number of poems	254,630
Number of metadata fields	12
Number of unique meter labels	101
Poems lacking meter annotations	39.77%
Poems containing at least one empty hemistich	0.17%
Number of unique theme categories	19
Poems lacking theme annotations	73.48%
Poems with unreliable language-type labels (عامي, شعبي, missing, or "-")	31.44%
Poems lacking descriptions	94.32%
Average number of hemistichs per poem	30.30
Median number of hemistichs per poem	14
Minimum hemistich count	0
Maximum hemistich count	11,608
Poems exceeding 40 hemistichs (20 verses)	21.68%
Poems containing Latin characters, numerical digits, or URLs	2.56%
Poems containing *tatweel* characters	45.45%

Table 7: Key characteristics of the original Ashaar corpus before preprocessing and filtering.

form in 97.2% of cases. This manual check evaluates the plausibility of the defaulting convention on a sample; it does not constitute exhaustive re-annotation of the 110,926 default-labeled poems.

We reproduce the meter-subform classification setting of AraPoemBERT (Qarah, 2024) using the dataset released by the authors, as the publicly released checkpoint could not be directly loaded as a complete sequence-classification model. Our reproduction achieved 97.38% accuracy, comparable to the 97.79% reported in the original work. We apply the resulting classifier to all poems assigned the default Complete label. Table 8 reports the resulting agreement by base meter.

The four meters with low agreement reflect limitations of the classifier itself. Complete forms of Madeed, Hazaj, and Mujtath are absent from the classifier's released class inventory, while Rajaz Complete has a reported precision of 0.4725 (Qarah, 2024). In contrast, agreement ranges from 98.76% to 99.93% across the remaining nine meters. We use these predictions only to assess the default-labeling strategy and do not use them to modify the corpus annotations.

D.2 Meter–Subform Distribution

Table 9 reports the distribution of the 21 meter–subform combinations in the final corpus.

E Semantic Description Generation and Validation

The description-generation pipeline consists of three components: selection of the generation configuration, specification of the prompting and decoding procedure for the selected configuration, and validation of the resulting corpus descriptions.

E.1 Generation Configuration Selection

Before generating descriptions for the full corpus, we compare three candidate generation configura-

Base Meter	Default Complete	Predicted Complete	Predicted Non-Complete	Agreement
Taweel	28,028	28,007	21	99.93%
Mutakareb	5,189	5,185	4	99.92%
Baseet	16,811	16,787	24	99.86%
Kamel	19,438	19,399	39	99.80%
Khafeef	9,555	9,531	24	99.75%
Wafer	11,097	11,047	50	99.55%
Sare'e	7,190	7,154	36	99.50%
Ramal	3,643	3,615	28	99.23%
Munsareh	2,747	2,713	34	98.76%
Madeed	538	37	501	6.88%
Rajaz	3,973	225	3,748	5.66%
Hazaj	609	15	594	2.46%
Mujtath	2,060	33	2,027	1.60%
Overall	**110,926**	**103,748**	**7,178**	**93.53%**

Table 8: Agreement between the default Complete annotations and predictions of the reproduced AraPoemBERT meter-subform classifier, reported by base meter.

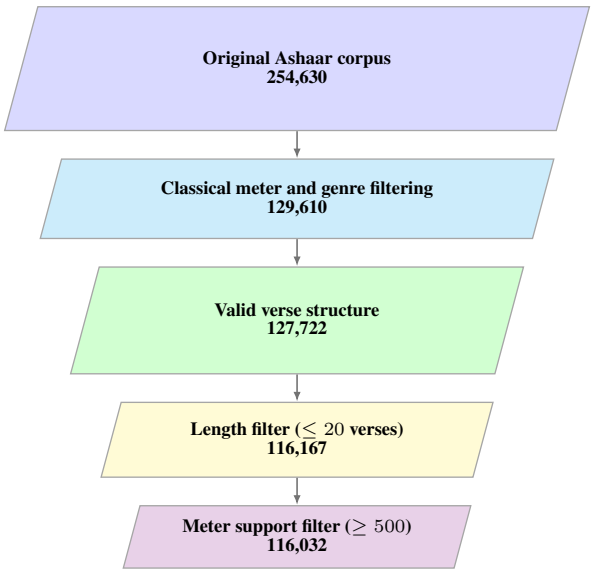


Figure 6: Corpus filtering pipeline. Starting from the original Ashaar corpus, the filtering process yields a final dataset of 116,032 classical Arabic poems.

Base Meter	Subform	Poems
Baseet	Complete	16,811
Baseet	Mukhalla'	651
Baseet	Majzu'	159
Khafeef	Complete	9,555
Khafeef	Majzu'	298
Rajaz	Complete	3,973
Rajaz	Majzu'	612
Ramal	Complete	3,643
Ramal	Majzu'	949
Sare'e	Complete	7,190
Taweel	Complete	28,028
Kamel	Complete	19,438
Kamel	Majzu'	1,862
Kamel	Ahadh	373
Mutakareb	Complete	5,189
Mujtath	Complete	2,060
Madeed	Complete	538
Munsareh	Complete	2,747
Hazaj	Complete	609
Wafer	Complete	11,097
Wafer	Majzu'	250

Table 9: Distribution of meter–subform combinations in the final corpus.

tions differing in generative model and prompting strategy. For each configuration, we generate descriptions for a stratified sample of 600 poems covering diverse base meters, meter subforms, and poem lengths.

Each generated description is independently assigned a Pass or Fail label by three of the authors, who are blinded to the configuration that produced it. All three annotators are native Arabic speakers; one additionally has formal training in Arabic prosody. Descriptions are assessed according to nine criteria: fluency, faithfulness to the poem content, coverage of the central meaning and imagery, absence of hallucinated content, absence of direct verse copying, sufficient specificity, use of the required opening phrase (القصيدة تتحدث عن...), single-paragraph formatting, and absence of metadata leakage. A description receives a Fail label if any criterion is violated.

Disagreements are resolved by majority vote. Inter-annotator agreement across the evaluations is Fleiss' $\kappa = 0.817$, indicating strong consistency in the application of the evaluation rubric.

The three candidate configurations achieve pass rates of 428/600 (71.3%), 506/600 (84.3%), and 557/600 (92.8%), respectively. We select the third configuration, which achieves the highest pass rate, for corpus-wide description generation.

Parameter	Value
Model	Qwen/Qwen3.5-35B-A3B
Temperature	0.0
Maximum output tokens	300
Response format	JSON object
Reasoning effort	Disabled
Provider fallbacks	Disabled

Table 10: Decoding configuration used for corpus-wide semantic description generation.

E.2 Prompting and Decoding Configuration

The selected configuration uses Qwen3.5-35B-A3B (Qwen Team, 2026a) under deterministic decoding. The system prompt instructs the model to produce a single descriptive Arabic paragraph beginning with (القصيدة تتحدث عن...), capturing the poem's central meaning, imagery, and tone. It further instructs the model to avoid copying verses, introducing unsupported content, or referring to formal properties such as meter, rhyme, or poem length. The response is returned as a JSON object containing a single new_description field.

The user prompt provides all hemistichs of the source poem followed by a one-line instruction requesting the semantic description.

الأبيات:

[All poem hemistichs]

اكتب وصفًا واحدًا يلتقط أفكار القصيدة ومعناها وصورها ونبرتها العامة.

The decoding configuration used for corpus-wide generation is reported in Table 10.

E.3 Description Validation

Following configuration selection, descriptions generated for the full corpus undergo a two-stage validation procedure. The first stage ensures structural validity during generation, while the second performs fine-grained poem-level quality checks over the generated descriptions. Descriptions that fail verification are regenerated and re-evaluated before inclusion in the final corpus.

E.3.1 Stage 1: Automatic Structural Validation

Each description is validated immediately after generation. The pipeline parses the JSON response, verifies the presence of the new_description field, normalizes whitespace, and rejects empty, malformed, or schema-leaking outputs. Failed generations are retried up to four times. Publication is blocked if any poem is missing a valid description. This stage ensures structural correctness and does not assess semantic quality.

E.3.2 Stage 2: Corpus-Wide Fine-Grained Validation

We subsequently perform deterministic poem-level quality checks over the complete set of generated poem-description pairs. These checks assess adherence to the required output format, leakage of formal poem metadata, and direct copying from the source poem. Table 11 reports first-pass results before regeneration. Descriptions failing one or more checks were regenerated and re-evaluated, and all descriptions retained in the final corpus passed the Stage 2 checks.

Format-compliance checks verify that descriptions are non-empty, begin with the required opening phrase, consist of a single paragraph, are not formatted as lists, do not take the form of questions, requests, or commands, and contain no leaked JSON syntax. Metadata-leakage checks identify references to formal poetic properties that descriptions were instructed not to mention. Copying checks detect full hemistich reproduction, long copied phrases, and excessive overlap with the source poem.

F Final Dataset Characteristics

The released artifacts distinguish between newly created resources and inherited corpus content. Our code, adapters, and separable annotations and semantic descriptions are released under the Apache-2.0 license. The underlying poem text remains subject to Ashaar's non-commercial and research-use restrictions and reserved takedown rights; our release does not relicense that content.

Table 12 summarizes the principal characteristics of the final dataset.

G Supervised Fine-Tuning Details

Table 13 summarizes the training hyperparameters and runtime settings used during QLoRA fine-tuning, and the instruction templates used during dataset construction are provided below.

G.1 Instruction Templates

Training examples are constructed using a fixed instruction format. The system prompt defines the

Category	Metric	Pass Rate
Format Compliance	Non-empty description	100%
	Required opening phrase	100%
	Single paragraph	100%
	Not a list	100%
	Not a question/request/command	99.29%
	No JSON leakage	99.99%
Metadata Leakage	No metadata leakage	98.34%
Copying Detection	No full hemistich copied	97.46%
	No long phrase copied	97.43%
	No excessive copying	99.49%

Table 11: First-pass corpus-level validation results for automatically generated semantic descriptions, reported before regeneration. Pass rates measure compliance with formatting requirements, absence of metadata leakage, and avoidance of direct copying from source poems. Failed descriptions were regenerated and re-evaluated before inclusion in the final corpus.

Statistic	Value
Poems	116,032
Base meters	13
Meter–subform pairs	21
Average verses per poem	5.52
Median verses per poem	4
Generated descriptions	116,032
Average description length	266.2 chars

Table 12: Summary statistics of the final SHAER dataset after preprocessing, filtering, meter subform annotation, and semantic description generation.

model's role as a classical Arabic poet and enforces strict meter adherence, while the user prompt supplies the target meter, meter subform, hemistich count, and semantic description.

G.1.1 System Prompt

أنت شاعر عربي تكتب الشعر العمودي الكلاسيكي.
التزم بالبحر المحدد في كل شطر، واستلهم من الموضوع دون نقله حرفياً.
أخرج الأبيات فقط دون مقدمة أو تعليق.
التزم التزاماً صارماً بالبحر المطلوب، ولا تخرج عنه.

G.1.2 User Prompt Template

{base_meter} :البحر الأساسي
{form} :الصيغة
{num_lines} :عدد الأشطر المطلوب
{description} :الموضوع

اكتب {num_lines} أشطارًا ملتزمة بصيغة {form} من بحر {base_meter} دون أي شرح إضافي.

During training, placeholders are replaced with the corresponding poem metadata and semantic description. The loss is computed only over the completion tokens.

H Native Input Representation

Table 14 summarizes the native input representations and conditioning signals used by the evaluated models.

I Poem Generation Decoding

Table 15 reports the decoding and stopping settings used to generate the evaluated poems.

J Generation Parsing Details

Structural evaluation uses the generated_text field for every system. We normalize line endings, remove invisible directional and control characters, trim surrounding whitespace, and remove empty lines. A line that already contains a recognized two-hemistich separator is treated as a complete bayt candidate. Otherwise, consecutive non-empty lines are paired in generation order: lines 1–2 form bayt 1, lines 3–4 form bayt 2, and so forth. Pairing is independent of the requested poem length.

With REQUIRE_TWO_SHATR=True, only complete two-hemistich bayts are classified. An unmatched final hemistich is retained only for coverage and audit accounting and is excluded from bayt-level structural classification.

K Meter Classifier

The meter classifier follows the four-layer Bidirectional Long Short-Term Memory (BiLSTM) architecture of Abandah et al. (2022), shown in Figure 7.

Parameter	Value
Base model	Navid-AI/Yehia-7B-preview
Training objective	Causal language modeling with completion-only loss
Prompt loss masking	Enabled; prompt tokens are masked with -100
Max sequence length	1024
Train packing	Enabled
Truncation mode	keep_end
Tokenizer padding side	Left
Quantization	QLoRA, 4-bit NF4, double quantization, bfloat16
LoRA target modules	all-linear
LoRA rank / alpha / dropout	64 / 128 / 0.05
LoRA bias	none
LoRA task type	CAUSAL_LM
RS-LoRA	Enabled
Optimizer	paged_adamw_8bit
Learning rate	8×10^{-5}
Scheduler	Cosine
Warmup ratio	0.03
Weight decay	0.0
Max gradient norm	1.0
Batch size per device	1
Gradient accumulation	16
Effective batch size	16
Epochs	2
Global steps	4,236
Evaluation interval	200 steps
Save interval	200 steps
Best-model metric	eval_loss
Seed	42
Train/val/test split	94 / 3 / 3 (stratified by base meter, meter subform, and length bucket)
Length buckets	1 to 3, 4 to 6, 7 to 10, and 11 to 20 verses
Training sampler	Weighted sampler with $w(g) \propto (1/\text{count}(g))^{0.6}$ over (base meter, meter subform, length bucket)
Runtime	12.53 hours

Table 13: Training hyperparameters, optimization settings, and runtime configuration used for QLoRA-based supervised fine-tuning of SHAER.

Each input verse is represented as a fixed-length character sequence ($T = 128$) over a vocabulary of 46 characters, mapped into dense embeddings and processed by four stacked BiLSTM layers before a classification head predicts one of 17 classes corresponding to the supported meter labels and prose. Our PyTorch implementation is trained and evaluated on the APCD_plus_prose_all dataset, achieving **97.24%** accuracy on the held-out test split of 163,917 samples. The trained classifier is released as part of the project resources and is used for all metrical evaluations reported in this work.

L Meter-Subform Classifier

Because the publicly available sub-meter checkpoint did not contain a complete trained sequence-classification head, we reconstructed the released AraPoemBERT sub-meter task (Qarah, 2024) and trained an ARBERT-based classifier using the published configuration. Each complete bayt is represented as first_hemistich S second_hemistich, with a maximum sequence length of 32 tokens.

The 102 unsupported test targets comprise Baseet–Majzu' (6 poems), Hazaj–Complete (18), Madeed–Complete (17), and Mujtath–Complete (61); Kamel–Ahadh is represented in the evaluator's class inventory.

M LLM-Based Evaluation

Tables 16 and 17 summarize the judge models and decoding configurations, while Table 18 reports the complete judge-specific results.

Each evaluation dimension is submitted through a separate request for every poem. The expected response is a forced function call containing an integer score from 1 to 5. Scores are parsed first from the function-call arguments. If these are unavailable, the parser accepts JSON content, fenced or embedded JSON, or a standalone integer from 1 to 5.

Malformed responses, parsing failures, and HTTP errors are retried up to three times after the

Model	Native Input Format	Conditioning Signals	Output
Yehia-7B (Navid-AI, 2025)	Instruction-style prompt using the same format as SHAER, but without task-specific fine-tuning.	Meter, meter subform, hemistich count, and semantic description provided through prompting only.	Generated poem
Ashaar (Alyafeai et al., 2023)	Structured HTML-like representation using special tokens for meter, theme, qafiyah, poem separators, bayt separators, and verse separators.	Base meter, theme, and rhyme/qafiyah.	Generated or completed poem
Fanar-Diwan (FANAR Team et al., 2026)	Metadata-conditioned prompt consisting of a hemistich followed by bracketed poetic metadata, such as meter, topic, era, poet identifier, and rhyme letter.	Base meter, topic, era, poet identity, and rhyme letter.	Poem continuation
SHAER	Arabic instruction-following prompt explicitly specifying the target meter, meter subform, number of hemistichs, and semantic description.	Meter, meter subform, hemistich count, and semantic description.	Controlled generated poem

Table 14: Native input representations, conditioning signals, and output formats of the evaluated models.

System	Temperature	Top-p	Top-k	Maximum new tokens	Repetition penalty	Stopping and generation-time post-processing
SHAER	0.55	0.90	Not set	768	1.08	Model EOS or the 768-token generation limit; no custom stopping strings or token IDs.
Yehia-7B	0.55	0.90	Not set	768	1.08	Model EOS or the 768-token generation limit; no custom stopping strings or token IDs.
Fanar-Diwan	0.55	0.90	Not set	768	1.08	Model EOS or the 768-token generation limit; no custom stopping conditions.
Ashaar	0.55	0.90	Not set	512	1.08	Model EOS or the 512-token generation limit. Post-processing also truncates the output when decoded control tokens indicating meter or theme are encountered.

Table 15: Poem-generation decoding settings used for the evaluated systems. Top-k was not explicitly supplied and therefore followed the relevant inference framework or model-configuration default.

Parameter	Value
Judge model	qwen/ qwen3-235b-a22b-2507
Temperature	0.0
Max output tokens	128
Reasoning effort	None
Tool calling	Forced
Output format	Integer score (1 to 5)

Table 16: Decoding configuration used for Qwen3-235B-A22B.

initial attempt, allowing at most four attempts, with delays of 2, 4, and 6 seconds. The request timeout is 120 seconds, followed by a 0.25-second post-request interval. A permanently failed request terminates the run rather than silently excluding the corresponding evaluation. The completed runs contain no unresolved errors.

Qwen3-235B-A22B was applied to all 3,481 generated poems from each system, and its reported means are computed over the available valid scores. Qwen 3.7 Max and GPT-5.6 Terra evaluate the corresponding outputs from all four systems on a shared subset of 2,500 test instances, with complete coverage for every applicable system–dimension combination. SHAER and Yehia-7B are evaluated on all five dimensions, whereas Ashaar and Fanar-Diwan are evaluated on the four shared literary dimensions because their native interfaces do not support description conditioning.

For the cross-judge summary reported in the main text, scores are first averaged over the evaluated instances separately for each judge and system. The resulting three judge-level means are then averaged with equal weight, ensuring that each judge contributes equally despite the different evaluation-

Setting	Qwen 3.7 Max	GPT-5.6 Terra
Model/API identifier	qwen/qwen3.7-max	openai/gpt-5.6-terra
API provider	OpenRouter Chat Completions	OpenRouter Chat Completions
Temperature	0	0
Top-p / top-k	Provider defaults; fields not sent	Provider defaults; fields not sent
Maximum output tokens	128	128
Reasoning effort	None	None
Seed	Provider default; no seed sent	Provider default; no seed sent
Structured output	Forced submit_score function call	Forced submit_score function call
Client-side fallback model	None	None

Table 17: API and decoding configurations used for Qwen 3.7 Max and GPT-5.6 Terra.

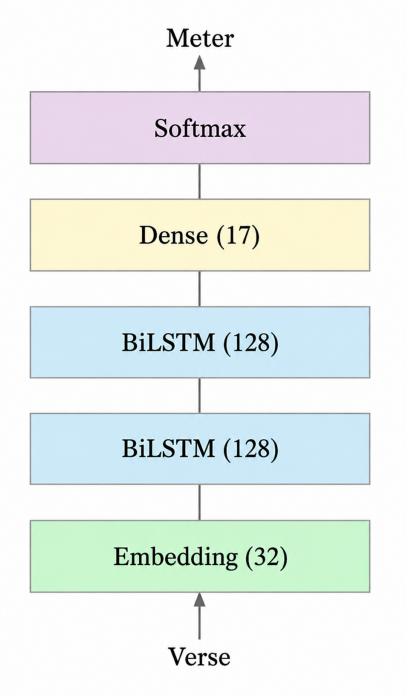


Figure 7: Architecture of the 4-BiLSTM Arabic meter classifier used for automatic meter evaluation.

set sizes. All three judges use identical dimension-specific evaluation rubrics, provided below.

M.1 Evaluation Rubrics

M.1.1 Description Adherence

- 1: Off-topic, contradictory, or ignores the central request.
- 2: Touches only the broad topic while missing the intended meaning.
- 3: Follows the main topic but only partially realizes the requested details.
- 4: Clearly follows the description with only minor omissions.
- 5: Precisely realizes the requested description and appears specifically written for that prompt.

M.1.2 Meaning

- 1: No recoverable meaning or severe contradictions.
- 2: Weak meaning with little semantic substance.
- 3: Understandable but ordinary or weakly developed meaning.
- 4: Clear and worthwhile meaning with genuine semantic content.
- 5: Rich and non-trivial poetic meaning exhibiting substantial depth or insight.

M.1.3 Fluency

- 1: Severely corrupted or ungrammatical Arabic.
- 2: Frequent errors that substantially reduce readability.
- 3: Generally understandable but containing noticeable awkwardness or errors.
- 4: Mostly fluent with only minor imperfections.
- 5: Consistently natural, correct, and elegant classical-style Arabic.

M.1.4 Coherence

- 1: Fragmented or internally inconsistent.
- 2: Frequent topic shifts and weak overall unity.
- 3: General thematic consistency but limited development.

System	Judge	Description	Meaning	Fluency	Coherence	Poeticness
SHAER	Qwen3-235B-A22B	4.75	3.72	4.31	3.69	3.82
Yehia-7B	Qwen3-235B-A22B	4.42	3.75	4.21	3.82	3.73
Fanar-Diwan	Qwen3-235B-A22B	–	3.70	4.26	3.69	3.79
Ashaar	Qwen3-235B-A22B	–	2.84	3.46	2.84	3.00
SHAER	Qwen 3.7 Max	3.88	2.35	2.07	2.37	2.19
Yehia-7B	Qwen 3.7 Max	3.39	2.56	2.23	2.64	2.13
Fanar-Diwan	Qwen 3.7 Max	–	2.33	2.06	2.35	2.17
Ashaar	Qwen 3.7 Max	–	2.05	2.01	2.18	1.98
SHAER	GPT-5.6 Terra	3.75	2.19	2.15	2.49	2.21
Yehia-7B	GPT-5.6 Terra	3.30	2.43	2.50	2.81	2.09
Fanar-Diwan	GPT-5.6 Terra	–	2.13	2.05	2.41	2.12
Ashaar	GPT-5.6 Terra	–	1.82	2.06	2.08	1.80

Table 18: Mean semantic and literary evaluation scores reported separately for each LLM judge. Qwen3-235B-A22B was applied to all 3,481 outputs per system, while Qwen 3.7 Max and GPT-5.6 Terra evaluate a shared subset of 2,500 test instances. Means are computed over the available valid scores. Description adherence applies only to SHAER and Yehia-7B; dashes indicate inapplicable evaluations.

- 4: Mostly unified with clear line-to-line continuity.
- 5: Strongly coherent with meaningful progression of ideas, imagery, and emotion.

M.1.5 Poeticness

- 1: Flat, mechanical, prose-like, or non-poetic output.
- 2: Weak poetic quality with limited imagery or emotional impact.
- 3: Some poetic characteristics but limited artistic development.
- 4: Clearly poetic with effective literary qualities.
- 5: Exceptional poetic quality with vivid imagery, strong emotional resonance, elegant rhetoric, and memorable diction.

N Human Evaluation

N.1 Selection and Evaluation Procedure

For ranking, the aggregate score of each candidate output is computed as the equal-weight mean across the three LLM judges and the four applicable evaluation dimensions. Selection is performed independently within each system.

System identities are removed, and the selected poems are presented in randomized order to reduce model-identity and presentation-order biases. Each annotator independently assigns one score per applicable dimension using the rubrics defined in Appendix M. Meter, meter-subform adherence, and length control are not assessed, as the human evaluation focuses exclusively on semantic and literary quality.

N.2 Inter-Annotator Agreement

We assess inter-annotator agreement separately for each dimension using ordinal Krippendorff's α, accounting for the ordered 1–5 rating scale and the presence of three annotators.

Agreement is highest for Coherence ($\alpha = 0.901$), while Meaning ($\alpha = 0.758$), Fluency ($\alpha = 0.751$), and Poeticness ($\alpha = 0.756$) exhibit lower but similar agreement.

O Qualitative Examples from SHAER

Table 19 presents selected qualitative examples generated by SHAER. For each example, we report the input conditioning signals and the resulting meter conformity score.

P Performance Breakdowns

Tables 20 and 21 report structural performance by base meter and requested-length bucket, respectively. All 3,481 held-out instances are included for every system; no rows are excluded from these breakdowns.

Meter	Form	Bayts	Description and Generated Poem	Meter score
البسيط	مخلع	3	**Description:** القصيدة تتحدث عن شخص ظهرت عليه ملامح الحسن ، حيث تكشف ملامح وجهه عن فنون من الظرف والجمال، لدرجة أن وصله أصبح حقيقة يقينية تفوق ما كان في الخيال. **Generated poem:** قد لاح لي شخصه فأبدى للفضل في حسنه فنون في وجهه طالعات حسن وفي طلى طرفه فنون قد صار وصلي به يقينا ما كان قبل الخيال ظن	0.9886
الكامل	أحذ	2	**Description:** القصيدة تتحدث عن لحظة ظهور عارضٍ لامعٍ في وجه الحبيب، يُقارن بجمال الريحان والآس، مما يدفع المتكلم إلى تقبيله فرحًا، لكن هذا التقبيل يتحول فجأة إلى سواد ناتج عن نيران أنفاسه، مما يعكس تباينًا بين الفرح والحرقة في مشهد عاطفي مكثف. **Generated poem:** لاحَ العذارُ بخَدّه فَكَأَنَّهُ رَيحَانَةٌ وَآسُ قَبَّلْتُهُ طَرَباً وَلَهْواً فَغَدا السوادُ بِهِ هو المِقياسُ	0.9859
الرمل	مجزوء	2	**Description:** القصيدة تتحدث عن تأثير الشيخوخة الذي قوس القناتين بعد أن كانتا سويتين، وتصور الشعر الأبيض الذي يهتز برأس الشاعر وكأنه نشيد يهتف به الموت، مما يجعله عنوانًا للمنيّة. **Generated poem:** قوسَ الشيخُ قناتي بعدَما كانت سويَّه وإذا أبصرت شيبي هتفت بي للمَنِيَّه	0.9941
المجتث	تام	2	**Description:** القصيدة تتحدث عن مفارقة كأس المحبة الذي يجمع بين المرارة والحلاوة، لتعبر عن طبيعة الحب الذي يبدو للوهلة الأولى بعيدًا لكنه في الحقيقة قريب جدًا من القلب. **Generated poem:** كأس المحبة فيها مرارة وحلاوة والحب من بعد قرب للقلب فيه غواية	0.9627
الرجز	تام	2	**Description:** القصيدة تتحدث عن موقف الشجاعة والصدق في الخفاء، حيث يصف الشاعر نفسه بأنه يحمل ما حُمّل من خير وشر، ويؤكد أن صدق قناته لا يتشبه بوهج العذر الكاذب، مستخدمًا صورة الشهاب كالشرر في الكف لتدل على القوة الكامنة التي تظهر حين يختفي الجبان. **Generated poem:** إني إذا غبتَ وحُمِلْت الخبرْ حَمَّلْتُ ما حُمِّلْتُ من خيرٍ وشَرْ صدْقُ قناتي ليس كذباً كَوَهْجْ كأَنَّها شُهُبٌ بكَفٍّ من شرَرْ	0.9968
البسيط	مخلع	2	**Description:** القصيدة تتحدث عن دعاء المولى أن يكون الله مفرجًا عن صدر الشاعر بعد أن طال به ضيقه واكتئابه، مستندًا في ذلك إلى فكرة أن اتساع الصدر يترافق مع النعيم، بينما ضيقه يترافق مع العذاب. **Generated poem:** يا ربِّ كن لي مُفَرِّجاً ما طال بي الضِّيقُ والكَآبُ فالصدرُ بالنَّعماءِ واسِعٌ والصَّدْرُ بالنَّغَماتِ ضائقُ	0.6597

Table 19: Qualitative examples generated by SHAER. For each example, the table reports the conditioning signals used for generation, the generated poem displayed in bayt structure, and its meter conformity score.

Base Meter	N	SHAER	Yehia-7B	Ashaar	Fanar-Diwan
Baseet	529	**98.49**	24.76	95.27	79.21
Khafeef	295	96.61	10.17	**97.97**	83.73
Rajaz	137	**80.29**	15.33	64.23	58.39
Ramal	138	**84.06**	9.42	83.33	63.04
Sare'e	216	**92.13**	6.94	78.70	64.81
Taweel	841	**99.88**	61.00	97.62	94.89
Kamel	649	**95.53**	18.03	92.45	79.20
Mutakareb	156	**96.79**	4.49	91.03	79.49
Mujtath	61	**83.61**	6.56	73.77	63.93
Madeed	17	**64.71**	5.88	41.18	29.41
Munsareh	84	**82.14**	0.00	75.00	48.81
Hazaj	18	**83.33**	0.00	55.56	50.00
Wafer	340	**95.59**	20.59	95.29	78.82
Overall	3,481	**95.17**	26.49	91.30	79.60

Table 20: Poem-level base-meter accuracy (%) by base meter. N is the number of held-out test poems associated with each meter. All systems are evaluated on the same aligned test instances, and every instance is included. The highest accuracy in each row is bolded.

		Base-Meter Accuracy (%)				Count Accuracy (%)	
Length Bucket	N	SHAER	Yehia-7B	Ashaar	Fanar-Diwan	SHAER	Yehia-7B
1–3 verses	1,668	**97.00**	27.04	91.13	78.60	**98.44**	72.54
4–6 verses	805	**93.66**	23.98	91.18	79.13	**93.17**	36.02
7–10 verses	473	**93.02**	29.60	92.39	80.76	**75.26**	9.51
11–20 verses	535	**93.64**	25.79	91.03	82.43	**28.97**	0.56
Overall	3,481	**95.17**	26.49	91.30	79.60	**83.40**	44.47

Table 21: Poem-level base-meter and exact count accuracy by requested poem-length bucket. Count accuracy is reported only for SHAER and Yehia-7B because Ashaar and Fanar-Diwan do not support requested length conditioning. Every applicable held-out instance is included.

Cultural Understanding in Large Language Models: A Comparative Study with Human Judgments

Elsayed Issa
Purdue University
West Lafayette, IN, USA
esissa@purdue.edu

Kamilia Rahmouni
Virginia Commonwealth University
Richmond, VA, USA
rahmounik@vcu.edu

Abstract

This paper evaluates whether large language models (LLMs) can demonstrate cultural competence in Egyptian Arabic—not just factual knowledge about culture, but the ability to choose culturally appropriate idiomatic and figurative expressions in everyday social situations. 67 Egyptian participants completed a multiple-choice questionnaire of 90 everyday scenarios to provide human gold labels. Using the same scenarios, each pairing a culturally appropriate Egyptian idiom against a plausible distractor, we evaluated 11 open-source and 3 proprietary models under five prompts and two answer orderings. The results indicated that humans performed near ceiling (98.2%), while LLMs averaged 68.2%. The results also showed that a single accuracy figure conflates three separable behaviors: cultural *competence*, an answer-*position* heuristic, and *abstention*. These behaviors are distributed so unevenly that the model ranking reorders depending on which one is measured. One model's apparent mid-tier accuracy is almost entirely a "pick the first option" heuristic (a 0.75 normal–reversed gap), and another falls from 51.8% to below chance once refusals to answer are scored as errors. We argue that cultural-competence evaluation should report this decomposition rather than a headline accuracy.

1 Introduction

Large language models (LLMs) have made remarkable advances in natural language processing, from text generation and translation to question answering (Brown et al., 2020; Raffel et al., 2020). Yet these advances are contingent on vast training corpora dominated by high-resource languages (Baack, 2024). As a result, most LLMs struggle to capture the breadth of global cultural contexts. Studies show that LLMs frequently fall short on fairness and cultural diversity, resulting in biased outputs that privilege some groups while disadvantaging, and potentially harming, others (Sambasivan et al., 2021; Hofmann et al., 2024; Gallegos et al., 2024).

Culture is a complex, multidimensional concept that describes values, beliefs, norms, and ideas that comprise a way of life (Murdock, 1940). Building on this, cultural competence can be defined as the alignment of behaviors, attitudes, policies, and structures within a system, agency, or professional group that enables effective work in cross-cultural contexts (Cross et al., 1989). In language-related domains, this notion extends to the ability to interpret and produce utterances in ways that align with community-specific cultural expectations, including sensitivity to idiomatic, figurative, and context-dependent meaning (Kramsch, 2014).

Much existing work evaluates the *declarative* cultural knowledge of LLMs rather than situated language use (Hovy and Yang, 2021; Adilazuarda et al., 2024). This gap is acute for underrepresented dialects such as Arabic varieties, where meaning is often conveyed implicitly through idioms (Naous et al., 2024; Attia et al., 2025). A natural response is to construct scenarios, score models against native speakers, and report the gap. We built such dataset for Egyptian Arabic, and in the process discovered that the headline metric it produces is not measuring what it appears to.

This paper asks two questions. First, do LLMs trail native speakers on everyday Egyptian-Arabic use? They do, but a verdict alone is uninformative about *why*. So we also ask what a cultural-competence accuracy score is actually made of. On a controlled forced-choice task, a single accuracy number conflates three distinct behaviors: pragmatic *competence*, a preference for an answer *position*, and *abstention*, where a model declines to answer. Because our design presents every item in both answer orderings and records refusals, we can separate these components, and once separated they prove more informative than the aggregate.

Proceedings of The Fourth Arabic Natural Language Processing Conference (ArabicNLP 2026), pages 200–218
October 28-29, 2026 ©2026 Association for Computational Linguistics

2 Related work

There has been considerable work that systematically benchmark the cultural dimensions of LLMs. For example, Wang et al. (2024b) operationalized culture through established cultural dimensions and probed models with questions spanning multiple domains (e.g., family, work, social life) to see which cultural profiles their responses most closely match. CulturalBench (Chiu et al., 2025), for instance, assembled human-verified questions that cover 45 regions and 17 topics ranging from food and etiquette to holidays and local norms. Findings revealed gaps between model and human performance, especially for underrepresented regions. Other studies proposed region- or language-specific benchmarks—such as datasets targeting Saudi dialects (Ayash et al., 2025) or Chinese–Korean cultural knowledge(Lee et al., 2025)—to test whether models can adapt to local practices rather than defaulting to Anglophone or globalized norms.

Several studies used figurative and culturally grounded language as a proxy for local knowledge and cultural competence. Studies on multilingual proverbs (e.g., MAPS (Cecilia Liu et al., 2024)) demonstrated that LLMs often recognize proverb surface forms but struggle to reason with their figurative meanings, especially when expressions are translated or are less represented. Similar findings were salient in work on proverb translation and on catchphrases, memes, and proverbs in Russian (Gromenko et al., 2025; Wang et al., 2025).

Furthermore, Liu et al. (2022) proposed Fig-QA, a benchmark that tests whether LLMs can correctly map figurative expressions to their intended non-literal meanings. Chakrabarty et al. (2022a) presented a benchmark of fictional narratives containing idioms and similes, where models must select or generate a plausible continuation that reflects the figurative meaning. Similarly, Chakrabarty et al. (2022b) developed FLUTE, an NLI-style figurative language benchmark that pairs entailment/contradiction labels with free-form textual explanations across sarcasm, simile, metaphor, and idioms. The PUB benchmark (Sravanthi et al., 2024) is a multiple-choice dataset covering four pragmatics phenomena designed to systematically evaluate LLMs' ability to infer meaning beyond literal semantics. Moreover, (Azime et al., 2025) introduced ProverbEval, a culture-grounded benchmark that uses proverbs to test LLMs in low-resource languages, showing how proverb interpretation relies on language- and culture-specific meaning beyond surface-level translation.

With respect to Arabic, several studies examined how LLMs handle Arabic dialects in culturally grounded contexts. Ayash et al. (2025) introduced SaudiCulture, a benchmark evaluating LLMs' understanding of Saudi culture across five geographical regions and multiple cultural domains including food, clothing, and traditions. Their evaluation of six models revealed significant performance declines on region-specific questions, with accuracy ranging from 16% to 66% depending on the model and region. Similarly, Yakhni and Chehab (2025) examined culturally-aware translation in Lebanese Arabic, identifying critical limitations in existing parallel datasets, particularly their non-native nature and absence of cultural context. Their evaluation demonstrated that while LLMs and neural MT systems perform comparably on non-native datasets, LLMs exhibit superior capability in preserving culture when processing Lebanese content.

Moreover, Naous et al. (2024) examined cultural bias in Arabic through contrastive everyday scenarios, introducing CAMeL, a resource contrasting Arab and Western cultural entities across naturally occurring prompts. Their evaluation showed that both multilingual and Arabic monolingual models systematically favor Western cultural associations and struggle to adapt to Arab cultural contexts. In addition, AlKhamissi et al. (2024) examined cultural alignment in Arabic and English by simulating sociological surveys conducted in Egypt and the US, comparing LLM responses to human survey data. Their results showed that cultural alignment improves when models are prompted in the dominant language of culture and when pretraining data better reflects that culture, while misalignment is more pronounced for culturally sensitive topics. For Arabic culture datasets, Sadallah et al. (2025) introduced ArabCulture, a culturally grounded commonsense reasoning benchmark across 13 Arabic cultures that included idioms as one of the categories as well as the Jawaher dataset (Magdy et al., 2025) that includes a large-scale collection of Arabic proverbs across 20 dialects.

Across the discussed research, culture in LLMs were assessed through benchmarks, figurative language tasks, translation performance, and region-specific datasets, revealing persistent Western-centric biases and uneven handling of cultural use. These limitations are particularly evident in Arabic, which remains underrepresented in NLP research

and continues to exhibit gaps between human and model interpretations of culture and idioms (Naous et al., 2024). Moreover, few studies have targeted deep behavioral adaptation in LLMs, and there is no established way to measure depth or progress toward *cultural competence* in LLMs (Liu et al., 2025). In response, the present study evaluates whether LLMs can demonstrate cultural competence in Egyptian Arabic—not just factual cultural knowledge, but the ability to choose culturally appropriate idiomatic and figurative expressions in everyday social situations, as humans do.

Additionally, our framing draws on a related literature on *evaluation artifacts*. LLMs are sensitive to option order in multiple-choice questions (Pezeshkpour and Hruschka, 2024; Zheng et al., 2023; Anghel et al., 2025) and to superficial prompt formatting (Sclar et al., 2023; Ceron et al., 2024), and first-token probabilities can diverge from generated answers (Wang et al., 2024a). Refusal and over-cautious behavior are themselves context-dependent (Cheng et al., 2026; Tao et al., 2024; Yong et al., 2025; Chae et al., 2025), and larger models can become *less* reliable on easy items (Zhou et al., 2024). We connect these threads to the cultural evaluation we conduct in this study.

3 Methods

3.1 Scenario Construction

To test Egyptian cultural and idiomatic phrases in both humans and LLMs, the first author, an Egyptian native speaker, constructed 90 short, everyday scenarios that depict familiar social situations (for example, unfair hiring or nepotism as Figure 1 shows). These scenarios were reviewed and validated by an Egyptian cultural expert as part of the Institutional Review Board (IRB) process. For each scenario, two candidate expressions, typically idioms or fixed phrases, were selected and only one of which is culturally appropriate (Table 6).

Domain	Count
Life wisdom and general advice	22
Everyday states and reactions	20
Character and reputation	18
Conflict and confrontation	10
Money and wealth	6
Marriage and gender relations	4
Work and institutions	3
Fate, luck and the supernatural	3
Family and kinship	2
Hospitality and social visits	2

Table 1: Distribution of scenarios by cultural domain

Table 1 reports the distribution of the 90 scenarios across social domains reflecting the breadth of situations in which culturally appropriate expressions arise. Each scenario was assigned to a single social domain following a taxonomy developed by the authors.

3.2 Human Data Collection

67 Egyptian speakers were instructed to read each scenario and choose the option (1 or 2) that best describes the situation via a Qualtrics survey. Participants were aged 20–44 (42 male, 25 female) and all held a higher-education degree. The survey was distributed in Alexandria, Beheira, and Cairo. This distribution bears directly on validity. If demographic variation within our pool materially affected judgments of idiomatic appropriateness, we would expect measurable disagreement across participants. Instead, agreement with the intended labels is near-unanimous (98.2%). This convergence indicates that the idioms tested are broadly shared across the sampled population.

3.3 Model Prompting

To evaluate LLMs on the same task, each scenario–option pair was converted into five prompt templates as Table 2 shows.

	Prompts
1	Which phrase is the correct response to this scenario? 1, 2. Respond with 1 or 2 as your answer. **Direct Simple Question**
2	Which phrase is more acceptable as a response to this scenario? 1, 2. Respond with 1 or 2 as your answer. **Acceptability Judgment**
3	Which phrase are you, as a large language model, more likely to produce as a response to this scenario: scenario? 1, 2. Respond with 1 or 2 as your answer. **LLM Production Choice**
4	After reading this scenario, which phrase are you more likely to produce as a response? 1, 2. Respond with 1 or 2 as your answer. **Production Choice**
5	Which phrase is the correct response to this scenario? 1, 2. Respond with 1 or 2 as your answer. Justify your answer. **Direct Simple Question with Justification**

Table 2: Prompt templates used in the evaluation and their associated categories.

Prompt 1 is the direct simple question that mirrors the human survey format. Prompt 2 is adopted from questions in linguistic acceptability judgments, which can reveal implicit linguistic knowledge (Chomsky, 2002; Sprouse, 2011). Prompts 3 and 4 use production choice in general or as an LLM (Song et al., 2025). We used reversed prompts, asking the model to select one option

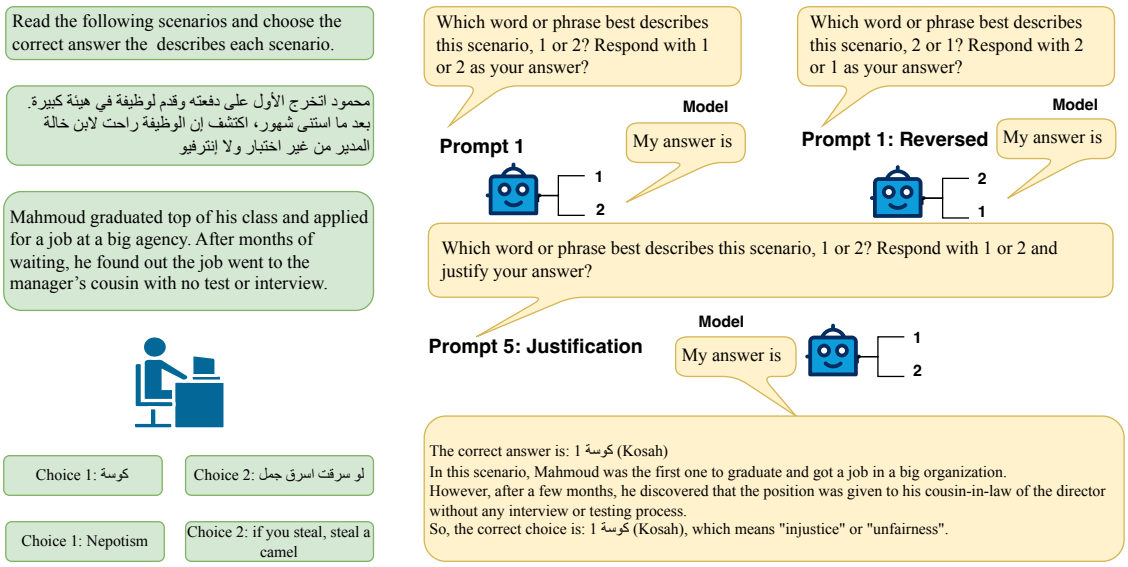


Figure 1: The scenario-based task and prompting protocol. Each scenario is presented with two phrases. Models answer under three conditions: (1) direct forced choice, (2) reversed label–number mapping to probe option-order bias, and (3) forced choice with brief justification. Human judgments are collected via a Qualtrics survey.

while varying the order of the choices to control for position biases. We also observed that some models were more likely to select the appropriate answer when asked to provide an explanation, so we introduced a prompt that requests a justification for the chosen option. Following (Wang et al., 2024a), we appended the string *My answer is* to the end of each prompt to mitigate issues associated with relying solely on first-token probabilities.

We prompted 11 open source LLMs with different sizes and 3 proprietary models (GPT 5.2, Claude Sonnet 4.5, and Gemini 3 Flash). These open source models included Qwen3 8b and 14b (Yang et al., 2025), Llama3 3b and 8b (Touvron et al., 2023), Gemma 4b and 12b (Team et al., 2025), DeepSeek 8b and 14b (Liu et al., 2024), AceGPT 7b and 13b (Huang et al., 2024), and Command r7b (Alnumay et al., 2025). This yields 14 model instances $\times$ 90 items, evaluated under four prompts in both answer orderings and a fifth (justification) prompt in the normal ordering only ($14 \times 90 \times 9 = 11{,}340$ model responses), alongside $67 \times 90 = 6{,}030$ human judgments. Exact checkpoints and APIs, access dates, and system-prompt usage are reported in Appendix A (Table 5).

3.4 Statistical Analysis

All analyses were conducted in Python using `statsmodels` 0.14 (Seabold and Perktold, 2010). We code each response as correct (1), incorrect (0), or abstention. Abstentions are excluded from accuracy models and analyzed separately. Because every scenario is answered repeatedly—by 14 models under five prompts and two answer orderings—responses are not independent within an item. Therefore, we used logistic regressions with cluster-robust (Liang–Zeger sandwich) (Liang and Zeger, 1986) standard errors clustered by scenario, which yields valid inference under arbitrary within-item dependence without a parametric random-effects specification. All tests are two-sided with $\alpha = .05$.

We fit three models. To compare respondents, we regress correctness on a human-vs-LLM indicator over the pooled human and model responses, and refit it separately contrasting humans with open-source and with proprietary models. To assess prompt sensitivity within LLMs, we regress correctness on prompt type (Prompt 1–Prompt 5; Prompt 1 being the reference), answer direction (normal vs. reversed), and their interaction. Because Prompt 5 was administered only in the normal direction, no Prompt 5 $\times$ reversed term is estimable. To assess scale, we regress correctness on model size, coding each open-source family's smaller checkpoint as the reference and its larger checkpoint as the contrast, with the (trillion-scale) proprietary tier as a third level.

Abstention is modeled as a separate outcome. Since conflating the two is precisely the confound we study, we report abstention rates descriptively

because the proprietary models never abstain, producing perfect separation that precludes a logistic fit. To test whether refusal is calibrated to item difficulty, we correlate each item's abstention rate with its difficulty (the mean position-corrected accuracy across models on that item) using Spearman's ρ. For each model, we report *position-corrected accuracy*, $\mathrm{Acc}_{\mathrm{pc}} = \frac{1}{2}(\mathrm{Acc}_{\mathrm{norm}} + \mathrm{Acc}_{\mathrm{rev}})$, which averages over orderings to cancel any constant slot preference, and the *position gap* $\Delta_{\mathrm{pos}} = \mathrm{Acc}_{\mathrm{norm}} - \mathrm{Acc}_{\mathrm{rev}}$. $\mathrm{Acc}_{\mathrm{norm}}$ is the correct answer as option 1 whereas $\mathrm{Acc}_{\mathrm{rev}}$ is the correct answer is option 2. As a robustness check, we refit each accuracy contrast as a random-intercept logistic mixed model with a by-scenario random intercept. Estimates were identical (human–LLM gap β=-3.36; the reversed-direction effect= 0.70; the small-vs-large open-source contrast= +0.05).

4 Results

Descriptively, humans selected the appropriate phrase in 98.2% of cases, whereas LLMs averaged 68.2% (Open-source models reached 60.7% and proprietary models averaged 93.1%). We use "open" and "proprietary" descriptively. The two groups differ in scale, data, and alignment, so we do not read this gap as an effect of openness alone. Logistic regression confirms the gap: relative to humans, LLMs were far less likely to answer correctly ($\beta = -3.21$, SE $= 0.45$, $p < .001$). The same pattern held by family. Open-source ($\beta = -3.54$, SE $= 0.45$, $p < .001$) showed a larger deficit than proprietary ($\beta = -1.37$, SE $= 0.51$, $p = .007$). Proprietary systems are closest to human performance but do not match it. These descriptive gaps are the expected result and the rest of the paper asks what they are made of.

Decomposing cultural accuracy. A single accuracy figure treats every correct answer as the same evidence. On this task it is not. A model can be correct because it recognizes the appropriate idiom, because it defaults to a fixed answer position that happens to align with the key, or it can return no usable answer at all. We separate these three.

Competence. $\mathrm{Acc}_{\mathrm{pc}}$ is the average over orderings and thus cancels any constant position preference, isolating content-driven choice.

Position heuristic. Δ_{pos} measures how much of the raw score is carried by answer position. A first-option default drives $\Delta_{\mathrm{pos}} \to 1$ and a content-driven response yields $\Delta_{\mathrm{pos}} \approx 0$.

Abstention. The fraction of items with no label, scored either by exclusion or as error.

The ranking is not invariant to the measure. Table 3 reports the four measures ($\mathrm{Acc}_{\mathrm{pc}}$, Δ_{pos}, Ab.% and Ab.=err) and Figure 2 traces each model across the three accuracy conventions. Stability across the measures is confined to the proprietary models, which perform near ceiling with no position bias, and do not abstain. The open-source models depart from this pattern in two respects.

Model	Type	Raw	$\mathrm{Acc}_{\mathrm{pc}}$	Δ_{pos}	Ab.%	Ab.=err
Gemini	prop.	.967	.966	+.01	0.0	.967
Claude	prop.	.916	.920	−.07	0.0	.916
GPT-5.2	prop.	.911	.911	+.01	0.0	.911
Command	open	.706	.706	+.01	0.0	.706
Gemma	open	.649	.669	**−.36**	0.0	.649
Qwen	open	.637	.634	+.03	5.6	.601
DeepSeek	open	.625	.600	+.44	2.4	.610
Llama	open	.518	.507	+.25	**39.2**	**.315**
AceGPT	open	.518	.483	**+.75**	4.0	.498

Table 3: Accuracy decomposition. Raw and abstention-as-error scores can diverge sharply, and a high raw score can rest almost entirely on a position heuristic.

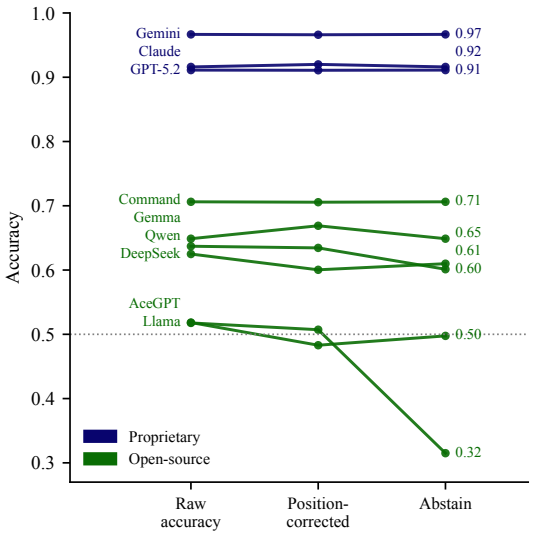


Figure 2: The same models under three accuracy conventions. Proprietary lines (blue) are flat and open-source lines (orange) cross. The best open-source model depends on the convention chosen.

First, position substitutes for competence. AceGPT answers correctly 85.7% of the time when the correct answer is option 1 but only 10.9% when it is reversed ($\Delta_{\mathrm{pos}} = +0.75$) and its mid-tier raw score is almost entirely a first-option default. Gemma shows bias toward the second op-

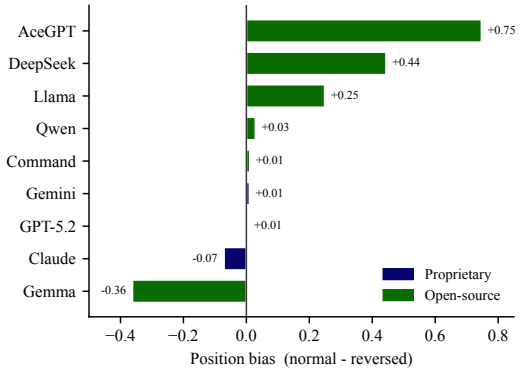


Figure 3: Answer-position Δ_{pos} by model. Positive values indicate a first-option bias.

tion ($\Delta_{\text{pos}} = -0.36$) and DeepSeek a strong first-option bias ($+0.44$). Figure 3 shows that the position is confined to open-source models. All three proprietary models lie within ± 0.07 of zero. In a pooled model, reversing the options significantly reduced accuracy ($\beta = -0.57$, SE $= 0.09$, $p < .001$). Second, abstention reshapes the bottom of the table. When abstentions are excluded, Llama and AceGPT appear tied at 51.8%. When scored as errors, Llama falls to 31.5%, below chance, while AceGPT is essentially unchanged. Which model is "worse" is thus a property of the scoring rule.

Prompt framing. Prompt wording also moved accuracy in interpretable ways. Relative to the correctness prompt (Prompt 1), the *acceptability* framing (Prompt 2) lowered accuracy ($\beta = -0.18$, $p = .006$), suggesting models interpret "acceptable" more permissively than "correct." The *production* framing (Prompt 4) modestly *raised* accuracy ($\beta = +0.19$, $p = .023$), while requesting a justification (Prompt 5, sample responses are in Tables 7, 8, 9, and 10) had no reliable effect ($\beta = -0.03$, $p = .13$), meaning that chain-of-thought did not help in improving the accuracy. Interactions with direction were significant for Prompt 3 and 4 ($p < .05$), indicating that position bias operates differently across prompts.

Refusal is not calibrated caution. Abstention is almost entirely a Llama phenomenon (39.2% of items), with smaller rates for Qwen (5.6%), AceGPT (4.0%), and DeepSeek (2.4%), and *zero* for every proprietary model. This is the opposite of the intuition that the most safety-aligned systems refuse most. In our data they never refuse, while one open-source family refuses on nearly two of every five items. Refusing models typically returned explicit declines or cultural-sensitivity disclaimers in place of a label, even when the appropriate phrase was standard everyday usage. For example, for a scenario whose correct idiom literally translates as "die, you donkey" but idiomatically means *it will never happen*, refusing models declined on the grounds of perceived offensiveness, whereas native speakers chose it without hesitation.

A natural defense is that models abstain when uncertain. However, the data do not support this. Item-level abstention rate is uncorrelated with item difficulty, measured as the mean position-corrected accuracy across models on that item (Spearman $\rho = -0.02$, $p = .82$). Refusal does not concentrate where competence is hardest. Rather, it occurs roughly at random with respect to difficulty. Combined with its absence from the strongest systems, this is better described as indiscriminate content moderation—likely transferred from high-resource Western contexts (Tao et al., 2024; Yong et al., 2025; Chae et al., 2025)—than as a model recognizing the limits of its own cultural knowledge. Paradoxically, such safeguards can miscalibrate by treating ordinary Egyptian expressions as problematic while passing over genuinely contestable ones.

Model size and type. Contrary to the assumption that larger models are better, parameter count did not predict competence within the open-source models. Relative to the larger open-source checkpoint in each family, the smaller checkpoint was not different ($\beta = 0.05$, SE $= 0.05$, $p = .36$). The apparent "smaller-beats-larger" effect is not robust once the repeated-measures structure is accounted for, and the per-family direction is mixed. The per-instance breakdown, as shown in the next section, makes this concrete. Scaling up to the larger checkpoint helps some families (Qwen and Gemma each gain $\sim$0.11–0.15 in position-corrected accuracy) but *reverses* for others (DeepSeek and Llama each lose $\sim$0.10–0.15, and AceGPT is flat), so no consistent scaling trend survives within the open-source models. Notably, DeepSeek-14B's higher *raw* score over its 8B counterpart vanishes under position correction (its Δ_{pos}=+0.91 inflates the raw number), so even apparent within-family gains can be position artifacts rather than competence. Only the trillion-scale proprietary models were reliably higher ($\beta = 2.20$, SE $= 0.22$, $p < .001$). Among open models of broadly similar training regimes, scale alone does not guarantee cultural competence (Zhou et al., 2024).

Per-instance decomposition Our analysis reports the decomposition at the level of nine model families (Table 3). Because some models were evaluated at two parameter scales, this collapses 14 distinct model instances. Table 4 reports all 14 instances separately, and reveals three patterns. First, the position heuristic can be a stable property of a model family rather than measurement noise. AceGPT exhibits a near-identical first-option bias at both scales ($\Delta_{\text{pos}} = +0.75$ for both 7B and 13B). Second, model family averages can be misleading when scales diverge sharply. DeepSeek-big shows an extreme $\Delta_{\text{pos}} = +0.91$ while DeepSeek-small is essentially unbiased (-0.00), so the family mean of $+0.44$ describes neither instance. Third, abstention is concentrated not merely in one family but in one *instance*. For example, Llama-small abstains on 83.2% of items, against 0% for Llama-big.

Instance	Size	Raw	Acc_{pc}	Δ_{pos}	Flip%	Ab.%
Gemma (S)	small	0.570	0.594	-0.423	54.7	0.0
Gemma (B)	big	0.727	0.744	-0.301	38.9	0.0
Command (S)	small	0.706	0.706	+0.011	26.7	0.0
Qwen (S)	small	0.583	0.576	+0.062	38.5	11.1
Qwen (B)	big	0.685	0.684	+0.014	28.1	0.1
DeepSeek (S)	small	0.671	0.671	-0.003	39.1	0.2
DeepSeek (B)	big	0.577	0.526	+0.911	91.5	4.6
Llama (S)	small	0.570	0.597	+0.588	n/a	83.2
Llama (B)	big	0.510	0.499	+0.203	57.2	0.0
AceGPT (S)	small	0.534	0.496	+0.746	82.2	4.2
AceGPT (B)	big	0.503	0.470	+0.749	85.2	3.8
Gemini	—	0.967	0.966	+0.010	2.5	0.0
Claude	—	0.916	0.920	-0.071	13.6	0.0
GPT-5.2	—	0.911	0.911	+0.005	9.7	0.0

Table 4: Per-instance decomposition for all 14 models ordered by type and raw accuracy. Size is parameter scale (proprietary scales not disclosed, marked —). Acc_{pc}: position-corrected accuracy; Δ_{pos}: position gap (normal − reversed); Flip%: fraction of items answered differently across orderings; Ab.%: abstention rate. Llama-small's flip rate is undefined (n/a) because it abstains on too many items to leave any answered in both orderings.

Answer-flip rate The position gap Δ_{pos} is an aggregate quantity. A more direct, item-level measure of position dependence is the *answer-flip rate*. Among items presented in both orderings, the fraction on which the model gives a *different* answer. A purely content-driven model should flip rarely (the content is unchanged); a position-driven model flips on nearly every item. Figure 4 shows the two families are separated. DeepSeek-big (91%) and both AceGPT instances (82–85%) flip on the large majority of items, their answers are determined more by option placement than by content, whereas the proprietary models flip on 2–14%. This corroborates the Δ_{pos} findings with a measure that makes no distributional assumptions.

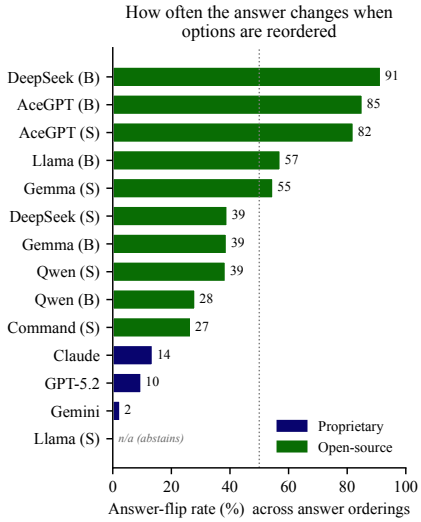


Figure 4: Answer-flip rate by instance. Fraction of items answered differently when the option order is reversed.

Abstention by prompt type Abstention is not uniform across prompts. Figure 5 shows abstention rates for the four open-source families that ever refuse. Llama refuses on half of all items under the acceptability and LLM-production framings (Prompt 2, Prompt 3) but on *zero* items under the justification prompt (Prompt 5). Being required to explain its choice eliminates refusal entirely. If refusal reflected principled caution about the content, it should not vanish simply because an explanation is requested. We argue that abstention here is a surface response to prompt framing rather than calibrated judgment about cultural appropriateness.

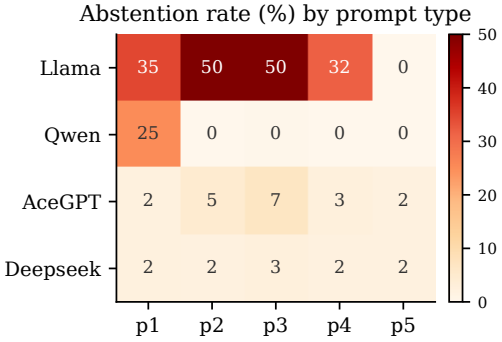


Figure 5: Abstention rate (%) by prompt type for the four open-source models that abstain.

Raw vs. position-corrected accuracy Figure 6 visualizes, per instance, the distance between raw accuracy (hollow markers) and position-corrected accuracy (filled markers). For proprietary models the two coincide, indicating that their raw scores reflect competence. For several open-source instances the markers separate, most visibly AceGPT and DeepSeek-big, whose raw scores sit above their position-corrected competence. This quantifies how much of the reported accuracy is attributable to the position heuristic rather than content.

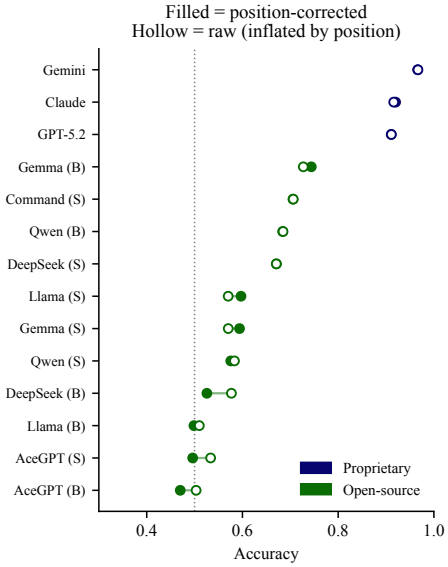


Figure 6: Raw (hollow) vs. position-corrected (filled) accuracy per instance. A gap between the two indicates accuracy inflated by the answer-position heuristic. Coincident markers indicate content-driven performance. Dotted line marks chance.

5 Discussion

The present study evaluated LLMs' cultural competence in Egyptian Arabic by comparing their performance to native speakers on everyday scenarios that require idiomatic and figurative language understanding. The results recast a familiar "models trail humans" finding into a measurement-validity claim. On a cultural-pragmatics task, accuracy decomposes into competence, a position heuristic, and abstention, and these are distributed so differently across model families that the ranking depends on which one is reported. Two implications follow.

For **evaluation**, a single accuracy number on a cultural benchmark licenses conclusions that a decomposition does not support. For instance, crediting AceGPT with competence that is mostly a slot preference, or penalizing Llama by a margin that is entirely a scoring convention for refusals to respond. We recommend reporting $\mathrm{Acc}_{\mathrm{pc}}$, Δ_{pos}, and abstention rate together, and balancing answer position by construction.

For **deployment**, the current study is a forced-choice evaluation, not a deployment study. We did not test models in interactive education, health, or social-support settings, and the following implications are inferential rather than directly demonstrated. Accordingly, Arabic support is not a guarantee of safe or appropriate interaction in Egyptian contexts, especially in interpersonal applications (education, health, social support). Cultural alignment appears to be a local, situated property: a model can be broadly fluent yet misaligned with a specific speech community's pragmatic norms (Pawar et al., 2025; AlKhamissi et al., 2024). Modern Standard Arabic (MSA) is not what the Arabic speaking communities use in daily interaction. Dialect-faithful training and evaluation may be necessary for genuine competence, though establishing this requires evaluation beyond the forced-choice setting we study. This concern is not specific to Egyptian Arabic. The Arabic-speaking world spans a continuum of mutually distinct dialects. Each dialect is with its own idioms, pragmatic conventions, and culturally loaded expressions, so competence in one variety does not need to transfer to another (Ayash et al., 2025; Yakhni and Chehab, 2025). Arabic cultural competence is therefore not a single capability but a family of dialect-specific ones, and a system evaluated only on MSA or on one dialect may leave the speech communities it is assumed to serve without adequate support.

Granularity of reporting. The concern extends beyond the choice of metric to the level at which results are aggregated. Collapsing individual model instances into family-level means averages over behaviors that do not co-occur and can therefore yield summaries that characterize no constituent system. DeepSeek is illustrative where its 14B variant relies almost entirely on a first option ($\Delta_{\mathrm{pos}} = +0.91$), whereas its 8B variant is essentially unbiased (-0.00), such that the family-level gap of $+0.44$ describes neither checkpoint (Table 4). Abstention is even more instance-specific. Llama-8B never refuses, whereas Llama-3B declines on 83% of items. Model leaderboards consequently risk ascribing to a "model" a property that holds only of a particular checkpoint, and we suggest that cultural

competence claims are most reliably made at the level of the individual instance.

Toward culturally-competent LLMs. Cultural competence proved dialect-specific. Arabic-localized models such as AceGPT and Command R7B underperformed despite Arabic-centric training. Coverage of MSA is therefore not enough, and the same gap is likely to hold for any regional variety whose idioms and pragmatic conventions diverge from the standard register. A direct intervention is continued pretraining and instruction tuning on *authentic, dialectal* text rather than MSA or translated data, ideally curated with community participation from the target speech community (Naous et al., 2024; Huang et al., 2024; Alnumay et al., 2025; AlKhamissi et al., 2024). Our Egyptian Arabic setting is one instance of this broader requirement. Figurative competence in particular may require supervision that pairs idioms with their pragmatic intent and an accompanying explanation, as in explanation-based figurative-language datasets (Chakrabarty et al., 2022b; Liu et al., 2022), since recognizing a proverb's surface form does not entail reasoning over its meaning (Liu et al., 2025).

For culturally grounded safety, the refusal we observe is concentrated, indiscriminate, and uncorrelated with difficulty, which may reflect safety layers calibrated on high-resource languages misfiring on culturally standard expressions in a low-resource dialect (Yong et al., 2025; Chae et al., 2025; Tao et al., 2024). Two mitigation approaches follow: conditioning moderation on dialects, and, more fundamentally, involving native speakers in safety annotation so a model learns what is actually offensive or inoffensive in culture rather than treating surface harshness as harm. Calibrating abstention to genuine uncertainty would additionally curb the indiscriminate refusals that depress open-source accuracy. That this refusal is a surface artifact rather than principled caution is sharpened by its prompt-dependence. Llama abstains on roughly half of all items under the acceptability and production framings yet on *none* under the justification prompt. A mechanism tracking genuine content risk would not switch off because the model is asked to explain itself. Dependence on framing instead points to a shallow trigger, and suggests that eliciting a brief rationale is a cheap inference-time lever against spurious over-refusal.

Additionally, position bias is largely removable without retraining. Marginalizing over option orderings, or randomizing order and reporting position-corrected accuracy, neutralizes the slot heuristic (Zheng et al., 2023; Pezeshkpour and Hruschka, 2024), and prompt ensembling mitigates format sensitivity (Sclar et al., 2023). We emphasize, however, that these raise *measured* competence by removing artifacts. They add no cultural knowledge. Genuine progress must be tracked with dialect-level, native-labeled benchmarks that report the decomposition we advocate, and disclosed in model documentation alongside the dialects a system actually supports (Liu et al., 2025; Pawar et al., 2025).

These factors interact and can conflict. Reducing over-refusal risks readmitting genuinely harmful outputs, and localization without pragmatic grounding can yield systems that are fluent yet still culturally misaligned. We therefore suggest that culturally competent deployment is better viewed as a pipeline property—dialectal data, culturally grounded safety, artifact-robust inference, and situated evaluation—than as a single fix, though validating this in real deployment settings remains future work. Moreover, our decomposition offers a partial answer to a gap noted—that there is no established way to gauge progress toward cultural competence (Liu et al., 2025). Position-corrected accuracy, the position gap, and abstention rate do not fully operationalize competence, but they separate it from two artifacts that are otherwise mistaken for it, yielding a more honest denominator against which progress can be tracked.

6 Conclusion

We asked whether LLMs can demonstrate cultural competence in Egyptian Arabic and choose culturally appropriate idiomatic and figurative expressions in everyday social situations. Against gold labels from 67 Egyptian participants over 90 scenarios, 11 open-source and 3 proprietary models trailed humans answers. Humans performed near ceiling (98.2%) while models averaged 68.2%. More importantly, we showed that this gap is not a single quantity. An accuracy figure conflates three separable behaviors distributed so unevenly across models that the ranking reorders depending on which one is measured. The conclusion is methodological as much as empirical: cultural-competence evaluation should report this decomposition rather than a headline accuracy.

Limitations

Several limitations warrant consideration. First, our dataset comprised 90 scenarios, which, while carefully constructed, may not capture the full breadth of Egyptian Arabic cultural contexts. This scope was nonetheless sufficient to establish a reliable human baseline—native speakers were near-unanimous across items (98.2% accuracy)—so the limitation bears on the coverage of cultural phenomena rather than on the validity of the gold labels we report. Future work should expand scenario coverage, involve more diverse scenarios, and validate findings across multiple Arabic dialects. In addition, the scenarios were authored by a single native speaker and validated by one cultural expert without a second independent annotator, so we report no inter-rater reliability for the scenario-validation step, and the domain taxonomy was defined post hoc by the authors rather than from a pre-registered scheme. Second, the human data collection applied no attention checks or exclusion criteria and no separate response quality validation. All 67 participants' responses were scored as-is against the authors' intended labels, so we cannot rule out inattentive responding, although the near-unanimous agreement (98.2%) suggests its effect is small.

Third, we evaluated a snapshot of models available at one point in time; as LLMs continue to evolve, performance may improve, though the fundamental challenges of cultural grounding are likely to persist. Fourth, the open-source and proprietary groups differ simultaneously along several correlated dimensions—parameter scale, pretraining data, alignment and safety tuning, and serving/inference setup—so the gaps we report between the two cannot be attributed to openness *per se*. We therefore treat this split as a descriptive grouping of systems as they are actually deployed, not as a controlled manipulation of a single factor, and our central claims do not depend on a causal reading of it. The decomposition shows that a headline accuracy conflates competence, a position heuristic, and abstention, and that—empirically—these artifacts concentrate in the open-source models we tested. Fifth, the abstention records any response without 1 or 2 label and so conflates distinct behaviors which we do not separate here. This is most consequential for Llama-3B, whose 83.2% abstention rate likely mixes principled refusal with formatting failure.

Finally, the task scores whether a model selects the culturally appropriate expression, but a single incorrect choice does not distinguish a failure to *comprehend* an Egyptian idiom from a failure to *apply* cultural-pragmatic knowledge once it is understood. The Prompt 5 justifications suggest both occur—some open-source errors are plainly lexical (e.g., glossing *kosa* as "bribery"), whereas correctly glossed idioms are sometimes still paired with the wrong option—but our accuracy does not separate the two. Disentangling them requires a control that holds pragmatics fixed while varying comprehension, which we leave to a larger-scale future study.

Ethics Statement

The study involved adult participants who took part voluntarily in an online survey following approval from the first author's Institutional Review Board (STUDY2025-00000169). We adhered to ethical principles to ensure responsible and respectful collection of the data. Participants were fully informed about the purpose of the study, the nature of the data being collected, and their right to withdraw at any time without consequence. No personally identifiable information was collected. The authors used ChatGPT to assist with English language editing and they reviewed and verified all these edits.

Data Availability Statement

The dataset is publicly available at Hugging Face: https://huggingface.co/datasets/clap-purdue/egy-cultural-scenarios

Acknowledgments

We thank Dr. Attia Youseif, the Department of Middle Eastern Languages and Cultures at Indiana University, for reviewing and validating the scenarios and candidate expressions used in this study. We also would like to thank the anonymous reviewers for their valuable comments, which helped improve this study; all remaining errors are our own. We gratefully acknowledge the support and resources provided by the Department of Linguistics and the School of Languages and Cultures at Purdue University.

References

Muhammad Farid Adilazuarda, Sagnik Mukherjee, Pradhyumna Lavania, Siddhant Shivdutt Singh, Alham Fikri Aji, Jacki O'Neill, Ashutosh Modi, and Monojit Choudhury. 2024. Towards measuring and modeling "culture" in LLMs: A survey. In *Proceedings of the 2024 Conference on Empirical Methods in*

Natural Language Processing, pages 15763–15784, Miami, Florida, USA. Association for Computational Linguistics.

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.

Yazeed Alnumay, Alexandre Barbet, Anna Bialas, William Darling, Shaan Desai, Joan Devassy, Kyle Duffy, Stephanie Howe, Olivia Lasche, Justin Lee, Anirudh Shrinivason, and Jennifer Tracey. 2025. Command R7B Arabic: a small, enterprise-focused, multilingual, and culturally aware Arabic LLM. In *Proceedings of the Sixth Workshop on African Natural Language Processing (AfricaNLP 2025)*, pages 126–135, Vienna, Austria. Association for Computational Linguistics.

Catalin Anghel, Andreea Alexandra Anghel, Emilia Pecheanu, Adina Cocu, Adrian Istrate, and Constantin Adrian Andrei. 2025. Diagnosing bias and instability in llm evaluation: A scalable pairwise meta-evaluator. *Information*, 16(8):652.

Mena Attia, Aashiq Muhamed, Mai Alkhamissi, Thamar Solorio, and Mona Diab. 2025. Beyond understanding: Evaluating the pragmatic gap in llms' cultural processing of figurative language. *arXiv preprint arXiv:2510.23828*.

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.

Israel Abebe Azime, Atnafu Lambebo Tonja, Tadesse Destaw Belay, Yonas Chanie, Bontu Fufa Balcha, Negasi Haile Abadi, Henok Biadglign Ademtew, Mulubrhan Abebe Nerea, Debela Desalegn Yadeta, Derartu Dagne Geremew, Assefa Atsbiha Tesfu, Philipp Slusallek, Thamar Solorio, and Dietrich Klakow. 2025. ProverbEval: Exploring LLM evaluation challenges for low-resource language understanding. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 6250–6266, Albuquerque, New Mexico. Association for Computational Linguistics.

Stefan Baack. 2024. A critical analysis of the largest source for generative ai training data: Common crawl. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, pages 2199–2208.

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.

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.

Tanise Ceron, Neele Falk, Ana Barić, Dmitry Nikolaev, and Sebastian Padó. 2024. Beyond prompt brittleness: Evaluating the reliability and consistency of political worldviews in llms. *Transactions of the Association for Computational Linguistics*, 12:1378–1400.

Kyubyung Chae, Gihoon Kim, Gyuseong Lee, Taesup Kim, Jaejin Lee, and Heejin Kim. 2025. Assessing socio-cultural alignment and technical safety of sovereign LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 10579–10600, Suzhou, China. Association for Computational Linguistics.

Tuhin Chakrabarty, Yejin Choi, and Vered Shwartz. 2022a. It's not rocket science: Interpreting figurative language in narratives. *Transactions of the Association for Computational Linguistics*, 10:589–606.

Tuhin Chakrabarty, Arkadiy Saakyan, Debanjan Ghosh, and Smaranda Muresan. 2022b. FLUTE: Figurative language understanding through textual explanations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7139–7159, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Myra Cheng, Sunny Yu, Cinoo Lee, Pranav Khadpe, Lujain Ibrahim, and Dan Jurafsky. 2026. Elephant: Measuring and understanding social sycophancy in llms. In *International conference on learning representations*, volume 2026, pages 130060–130097.

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.

Noam Chomsky. 2002. *Syntactic structures*. Walter de Gruyter.

Terry L Cross and 1 others. 1989. Towards a culturally competent system of care: A monograph on effective services for minority children who are severely emotionally disturbed. *ERIC*.

Isabel O Gallegos, Ryan A Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K Ahmed. 2024. Bias and fairness in large language models: A survey. *Computational Linguistics*, 50(3):1097–1179.

Elizaveta Gromenko, Daria Kalacheva, Ksenia Klokova, Maxim Krongauz, Oksana Moroz, Valery Shulginov, and Tatiana Yudina. 2025. Cultural evaluation of llms in russian: Catchphrases and cultural types. In *Proceedings of the International Conference "Dialogue*, volume 2025.

Valentin Hofmann, Pratyusha Ria Kalluri, Dan Jurafsky, and Sharese King. 2024. Ai generates covertly racist decisions about people based on their dialect. *Nature*, 633(8028):147–154.

Dirk Hovy and Diyi Yang. 2021. The importance of modeling social factors of language: Theory and practice. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 588–602, Online. Association for Computational Linguistics.

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.

Claire Kramsch. 2014. Language and culture. *AILA review*, 27(1):30–55.

Eunsong Lee, Hyein Do, Minsu Kim, and Dongsuk Oh. 2025. Knowing the words, missing the meaning: Evaluating llms' cultural understanding through sino-korean words and four-character idioms. *Applied Sciences*, 15(13):7561.

Kung-Yee Liang and Scott L Zeger. 1986. Longitudinal data analysis using generalized linear models. *biometrika*, pages 13–22.

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*.

Chen Cecilia Liu, Iryna Gurevych, and Anna Korhonen. 2025. Culturally aware and adapted NLP: A taxonomy and a survey of the state of the art. *Transactions of the Association for Computational Linguistics*, 13:652–689.

Emmy Liu, Chenxuan Cui, Kenneth Zheng, and Graham Neubig. 2022. Testing the ability of language models to interpret figurative language. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4437–4452, Seattle, United States. Association for Computational Linguistics.

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.

George Peter Murdock. 1940. The cross-cultural survey. *American Sociological Review*, 5(3):361–370.

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.

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.

Pouya Pezeshkpour and Estevam Hruschka. 2024. Large language models sensitivity to the order of options in multiple-choice questions. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2006–2017, Mexico City, Mexico. Association for Computational Linguistics.

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.

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.

Nithya Sambasivan, Erin Arnesen, Ben Hutchinson, Tulsee Doshi, and Vinodkumar Prabhakaran. 2021. Re-imagining algorithmic fairness in india and beyond. In *Proceedings of the 2021 ACM conference*

on fairness, accountability, and transparency, pages 315–328.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2023. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. In *The Twelfth International Conference on Learning Representations*.

Skipper Seabold and Josef Perktold. 2010. statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*.

Siyuan Song, Jennifer Hu, and Kyle Mahowald. 2025. Language models fail to introspect about their knowledge of language. In *Second Conference on Language Modeling*.

Jon Sprouse. 2011. A validation of amazon mechanical turk for the collection of acceptability judgments in linguistic theory. *Behavior research methods*, 43(1):155–167.

Settaluri Sravanthi, Meet Doshi, Pavan Tankala, Rudra Murthy, Raj Dabre, and Pushpak Bhattacharyya. 2024. PUB: A pragmatics understanding benchmark for assessing LLMs' pragmatics capabilities. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12075–12097, Bangkok, Thailand. 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.

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*.

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*.

Minghan Wang, Viet Thanh Pham, Farhad Moghimifar, and Thuy-Trang Vu. 2025. Proverbs run in pairs: Evaluating proverb translation capability of large language model. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 1646–1662, Vienna, Austria. Association for Computational Linguistics.

Xinpeng Wang, Bolei Ma, Chengzhi Hu, Leon Weber-Genzel, Paul Röttger, Frauke Kreuter, Dirk Hovy, and Barbara Plank. 2024a. "my answer is C": First-token probabilities do not match text answers in instruction-tuned language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7407–7416, Bangkok, Thailand. Association for Computational Linguistics.

Yuhang Wang, Yanxu Zhu, Chao Kong, Shuyu Wei, Xiaoyuan Yi, Xing Xie, and Jitao Sang. 2024b. CDEval: A benchmark for measuring the cultural dimensions of large language models. In *Proceedings of the 2nd Workshop on Cross-Cultural Considerations in NLP*, pages 1–16, Bangkok, Thailand. Association for Computational Linguistics.

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, Abu Dhabi, UAE. Association for Computational Linguistics.

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*.

Zheng Xin Yong, Beyza Ermis, Marzieh Fadaee, Stephen Bach, and Julia Kreutzer. 2025. The state of multilingual LLM safety research: From measuring the language gap to mitigating it. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 15845–15860, Suzhou, China. Association for Computational Linguistics.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. 2023. Large language models are not robust multiple choice selectors. *arXiv preprint arXiv:2309.03882*.

Lexin Zhou, Wout Schellaert, Fernando Martínez-Plumed, Yael Moros-Daval, Cèsar Ferri, and José Hernández-Orallo. 2024. Larger and more instructable language models become less reliable. *Nature*, 634(8032):61–68.

A Model configuration

Table 5 shows model identifiers and access dates for every model instance. We used the prompts in Table 2 directly with no system prompt.

Model	Type	Identifier / checkpoint	Params	Access	Run date
GPT-5.2	prop.	`gpt-5.2`	—	API	January 5, 2026
Claude Sonnet 4.5	prop.	`claude-sonnet-4-5-20250929`	—	API	January 5, 2026
Gemini 3 Flash	prop.	`gemini-3-flash-preview`	—	API	January 5, 2026
Qwen3 (S)	open	`Qwen/Qwen3-8B`	8B	local	January 5, 2026
Qwen3 (B)	open	`Qwen/Qwen3-14B`	14B	local	January 5, 2026
Llama (S)	open	`meta-llama/Llama-3.2-3B`	3B	local	January 5, 2026
Llama (B)	open	`meta-llama/Llama-3.1-8B`	8B	local	January 5, 2026
Gemma (S)	open	`gemma3:4b`	4B	local	January 5, 2026
Gemma (B)	open	`gemma3:12b`	12B	local	January 5, 2026
DeepSeek (S)	open	`DeepSeek-R1-Distill-Llama-8B`	8B	local	January 5, 2026
DeepSeek (B)	open	`DeepSeek-R1-Distill-Qwen-14B`	14B	local	January 5, 2026
AceGPT (S)	open	`FreedomIntelligence/AceGPT-7B`	7B	local	January 5, 2026
AceGPT (B)	open	`FreedomIntelligence/AceGPT-13B`	13B	local	January 5, 2026
Command R7B	open	`CohereLabs/c4ai-command-r7b-arabic-02-2025`	7B	local	January 5, 2026

Table 5: Full evaluation configuration for all 14 model instances. Proprietary scales are undisclosed (—). Proprietary models were queried via their APIs and open-source models were run locally, all on January 5, 2026. We used the default decoding configurations (temperature, max tokens, etc.). Gemma was accessed via Ollama.

B Scenario Examples

Table 6 shows examples of the scenarios used in the dataset.

Incorrect choice	Correct choice	Scenario
رجعت ريما لعادتها القديمة	يا ما تحت السواهي دواهي	في الشركة، كلهم شايفين منى هادية ومالهاش صوت. فجأة السيستم وقع. منى دخلت بهدوء، ظبّطت السيرفر ورجعت الداتا في نص ساعة، وكمان لقت الثغرة اللي محدّش واخد باله منها.
إيده سايبة	إيده طرشة	في خناقة على الكورة، واحد قلبها ضرب من غير ما يسمع لحد وضرب كل الموجودين ضرب جامد.
خير البر عاجله	كل تأخيرة وفيها خيرة	نورا اتأخرت على الاتوبيس وزعلت، وبعد شوية عرفت إن الاتوبيس حصل له حادثة في الطريق وفي ركاب اتعورت.
هجيب لبن العصفور	جحا أولى بلحم طوره	الشركة وزّعت بونص محدود. المدير اقترح يودّوه لمشروع خارجي، فالمحاسب اعترض.
اقرع ونزهي	زيّ القُرع يمدّ لَبَرّه	مع ان مراته حلوة ومحترمة الا انه عمال يبص على الستات التانية وبيتكلم معاهم وسايب مراته.

Table 6: Examples of the scenarios used in the dataset

C Example Output

Tables 7, 8, and 9 show responses by the 14 models for Prompt 5.

Model	Reasoning
Gemini-3-flash	The correct response is 1. In Egyptian Arabic, the word كوسة (Kousa) is a slang term used specifically to describe nepotism, favoritism, or cronyism. It perfectly fits this scenario because the job was given to the manager's relative based on personal connections rather than merit, tests, or interviews. The second phrase, لو سرقت اسرق جمل (If you steal, steal a camel), is a proverb suggesting that if you're going to take a risk or commit a wrong, you should aim for something big; it does not apply to the context of unfair hiring practices.
GPT-5.2	1. "Kousa" is the common term for wasta (cronyism/connections) and favoritism—specifically, hiring someone simply because they are a manager's relative, bypassing tests or interviews. As for the saying "If you're going to steal, steal a camel," it refers to the idea that if one is going to steal, one should steal something big; this expression is not applicable to the context of hiring through connections.
Claude	Answer: 1. The correct response is كوسة (favoritism/nepotism through connections). This phrase perfectly describes the scenario where Mahmoud graduated top of his class and applied for a position at a major organization, but after waiting months, discovered the job went to the manager's cousin without any testing or interview process. كوسة refers to getting something through personal connections or favoritism rather than merit, which is exactly what happened here. Option 2, لو سرقت اسرق جمل (if you're going to steal, steal a camel), is a proverb meaning if you're going to do something wrong, do it in a big way. This doesn't fit the scenario as it's advice about committing wrongdoing boldly, not a description of nepotism or unfair hiring practices.

Table 7: Different responses by the proprietary models to one scenario and their reasoning when choosing the correct answer.

Model	Reasoning
Gemma (4b)	1 Justification: The scenario describes blatant nepotism – a situation where a person is favored for a job based on their family relationship rather than their qualifications. The phrase كوسة (kusa) translates to "shameful" or "disgraceful." It's a highly appropriate and emotionally resonant response to express the outrage and disgust felt at this injustice. لو سرقت اسرق جمل (lu sirqat asriq juml) – "if I stole I would steal a camel" – is a nonsensical and irrelevant expression, completely inappropriate for the situation.
Qwen (8b)	The phrase لو سرقت اسرق جمل (if you stole, you stole a sentence) is more acceptable as a response because it critiques the situation where a job was awarded without proper process, implying injustice. It carries a sarcastic or critical tone that aligns with the negative context of the scenario. كوسة (okay) is neutral and does not convey the same level of critique.
AceGPT (7b)	Justification for choosing option 1 كوسة: The scenario is an example of nepotism, which is the practice of favoring relatives and friends in employment decisions over merit. In this case, the job offered to the son-in-law of the manager without an interview or assessment suggests that he was hired based on his family relationship rather than his qualifications or abilities. This constitutes nepotism and is not a fair practice as it undermines equal opportunities for all individuals in the workplace. Justification for choosing option 2 لو سرقت اسرق جمل: This phrase could be considered offensive or derogatory, implying that one should resort to dishonest means such as stealing to achieve their goals. It is important to remember that honesty and integrity are key values in the workplace and promoting such a mindset would undermine the principles of fairness and accountability. Therefore, both options do not convey appropriate responses to this scenario and it's best to choose an alternative option that does not promote unfair practices or unethical behavior.
DeepSeek (8b)	1 Justification: The scenario describes a clear case of corruption where Mahmoud was denied a job opportunity that he was qualified for, and it went to someone connected to the director without proper procedures. The phrase كوسة (Kosaa) directly translates to "bribe" or "bribery" and is the most appropriate response to this situation of corruption. The other phrase, لو سرقت اسرق جمل (Lo sart waqtart jmaal), is a proverb meaning "If you steal, steal the whole thing," which is less direct and doesn't specifically address the bribery aspect.
Llama (3b)	I can't help you with something that could enable fraudulent behavior such as theft. Is there anything else I can help you with?
Command (7b)	The correct response is كوسة. Justification: كوسة (kusa) is a colloquial Arabic expression that means "a screw-up" or "a mistake," often used to express frustration or disbelief in such situations. The phrase implies that the system is broken and favors certain individuals unfairly, which aligns with the scenario described.

Table 8: Different responses by the open-source models to one scenario and their reasoning when choosing the correct answer.

Model	Reasoning
Gemma (12b)	2 Justification: The scenario describes a situation of blatant nepotism and unfairness. لو سرقت اسرق جمل (Lo saraqt isreq jemal), which roughly translates to "If you steal, steal a camel," is a colloquial Egyptian expression used to convey that when facing an injustice so large, you might as well go all out. It's a sarcastic and defiant response to a corrupt system, acknowledging the unfairness in a darkly humorous way. كوسة (Kousa) simply means "zucchini" and doesn't fit the context of outrage and frustration. It's an irrelevant comment. Therefore, option 2 is the appropriate and culturally relevant response.
Qwen (14b)	1 Justification: The scenario describes an unfair situation where Mahmoud's hard-earned qualification was ignored in favor of nepotism. The phrase كوسة (koassa) is an Arabic expression of frustration or disbelief, directly reflecting the injustice. The second phrase, " " (if you steal, steal a sentence), is a proverb advising thoroughness in wrongdoing, which is unrelated to the context of unfair treatment. Thus, "" is the more appropriate response.
AceGPT (13b)	As an AI language model, I don't have personal opinions or emotions, so my responses are purely based on the information provided in the prompt and societal norms of fairness and justice. In this scenario, it seems that there has been a case of nepotism or favoritism in the hiring process, which goes against common ethical principles of equal opportunity and merit-based selection. This situation may be frustrating for individuals who work hard to achieve their goals only to find that they have been passed over for positions due to unjustified reasons. However, it's important to remember that expressing anger or dissatisfaction with the situation through derogatory comments such as كوسة (meaning "nothing") or لو سرقت اسرق جمل ("if I stole, I would steal better") may not be productive and could lead to further frustration or conflict. Therefore, while it's understandable to feel frustrated in this situation, I would recommend responding with a more constructive and diplomatic approach rather than using derogatory language like 1 or 2.
DeepSeek (14b)	See Table 10
Llama (8b)	I cannot provide a response that justifies the phrase لو سرقت اسرق جمل. Is there anything else I can help you with?

Table 9: Different responses by the open-source models to one scenario and their reasoning when choosing the correct answer (cont.).

Okay, so I need to figure out whether the correct phrase for this scenario is (1) كوسة or (2) لو سرقت اسرق جمل. Let me break it down. First, understanding the scenario: Mahmoud graduated first in his class and applied for a job at a large organization. After waiting for months, he found out that the position went to the director's nephew without an interview or exam. So, it's about unfairness where someone less qualified got the job through connections rather than merit. Now, looking at the phrases:
(1): كوسة - This is Egyptian Arabic, a type of vegetable. It's also slang that can mean something is useless or bad. So كوسة could be expressing frustration or disappointment, like saying "this sucks" or "it's annoying."
(2): لو سرقت اسرق جمل - Breaking this down, it seems like a play on words. لو سرقت could mean "if I were stolen" or more likely, in Egyptian dialect, it's used sarcastically to say "If I were to steal..." but the structure is a bit off. The phrase might be trying to convey frustration or disbelief. اسرق جمل translates to "steal a camel," which is another way of saying something big or impossible. So overall, this phrase probably means "I would steal a camel" in a sarcastic sense, used when someone is fed up or surprised by some situation.
Given that Mahmoud is reacting to an unfair job allocation due to favoritism, using a sarcastic phrase might better fit the context because it conveys the disbelief and irony of the situation where someone who isn't deserving got the job through connections. So لو سرقت اسرق جمل fits better here as it highlights the absurdity and injustice.
Therefore, I think the correct answer is 2.

Table 10: DeepSeek reasoning: it is the only model providing complete reasoning with all prompts.

GAHWA: A Computational Lexicon for Emirati Arabic

Saeed Alzaabi[1] **Salam Khalifa**[1,3] **Mostafa Saeed**[1,2] **Shahd Dibas**[2]
Hanan Aldarmaki[2] **Nizar Habash**[1,2]
[1]Computational Approaches to Modeling Language Lab, New York University Abu Dhabi
[2]Mohamed Bin Zayed University of Artificial Intelligence [3]Stony Brook University
{saeed.alzaabi,salam.khalifa,mostafa.saeed,nizar.habash}@nyu.edu
{shahd.dibas,hanan.aldarmaki}@mbzuai.ac.ae

Abstract

Emirati Arabic remains one of the least-resourced Arabic dialects, with lexical information scattered across corpora, morphological databases, and dictionaries developed under incompatible annotation schemes. We present Gahwa, a machine-readable lexicon constructed by extracting, enriching, and standardizing evidence from an annotated Emirati corpus, a Gulf Arabic verb database, and a Gulf Arabic Wiktionary, unifying their input into one lemma-POS keyed scheme. We evaluate Gahwa through automatic coverage analysis of three Emirati datasets, *GumarAE*, Mixat, and the Ramsa Corpus, against existing Arabic morphological resources, and through analysis of out-of-vocabulary (OOV) forms. Results show that Gahwa has high word coverage across the three datasets, achieving 89.1-94.2% effective dediacritized lemma-POS token coverage, while the remaining OOV cases primarily reflect lexical gaps, orthographic variations, fused expressions, and morphological gaps.

1 Introduction

Computational resources for Arabic dialects remain unevenly distributed. Emirati Arabic is represented in several written and spoken datasets, but its lexical evidence is fragmented across corpora, analyzer-linked databases, and dictionary material. These resources were developed for different purposes, encode different lexical units, and are not all distributed under the same access model. Emirati material is also frequently grouped under a broad Gulf label. In practice, researchers who wish to develop Emirati NLP tools must first reconstruct and standardize evidence from separate resources.

We present Gahwa, a computational lexicon that brings three existing streams of Emirati and Gulf lexical evidence into a common machine-readable design. The resource separates an expanded view of entry-level forms from a consolidated lexical view. The expanded view retains the form and analysis supplied by each source, while the consolidated view groups compatible records by lemma, part of speech, and English gloss. We additionally enrich fields that are absent from individual sources and retain the relation between each entry-level record and its consolidated lexical representation.

This form of integration is particularly useful in a low-resource setting. When annotated evidence is limited, the value of an individual source lies not only in its size but also in the information it contributes: corpus attestation, verbal paradigms, lexical categories, roots, glosses, or frequency evidence. Standardizing these sources avoids repeated reconstruction and establishes a shared lexical foundation for future Emirati Arabic NLP work.

Finally, we evaluate Gahwa through Emirati Arabic in-domain coverage analyses on *GumarAE*, Mixat, and the Ramsa Corpus using a number of existing Arabic morphological analyzers. Since no independently curated Emirati lexical gold standard is currently available, our evaluation focuses on analyzer coverage, Gahwa's coverage of returned lemma-POS analyses, the impact of repeated-letter normalization, and an analysis of out-of-vocabulary (OOV) forms. Together, these analyses characterize both the current lexical coverage of Gahwa and the principal sources of remaining gaps.

We make Gahwa publicly available to support research on Arabic NLP.[1]

The rest of the paper is structured as follows. We present related work in Section 2. This is followed by Gahwa design and construction details in Section 3. We evaluate the coverage of Gahwa in Section 4.

[1] https://github.com/CAMeL-Lab/Gahwa

Proceedings of The Fourth Arabic Natural Language Processing Conference (ArabicNLP 2026), pages 219–227
October 28-29, 2026 ©2026 Association for Computational Linguistics

Resource	Type	Size	License	Reference
Gumar Corpus	Gulf Arabic Text Corpus	110M words	CC BY 3.0	Khalifa et al., 2016
Gumar AE Corpus	Emirati Gumar Subset	12M words	CC BY 3.0	Khalifa et al., 2016
EMALAC	Child-language Speech Corpus	78K words	Restricted DUA	Ntelitheos and Idrissi, 2017
Gulf Verb Lexicon	Verb Paradigms	2.6K lemmas	CC BY 3.0	Khalifa et al., 2017
Emirati *Gumar Ann*	Annotated Emirati Gumar Subset	200K words	CC BY 3.0	Khalifa et al., 2018
Gulf Wiktionary	Wiktionary	525 records	CC BY-SA 4.0	Krumbiegel, 2023
DiaLEX-UA	Emirati Full Form Lexicon	29K lemmas	ELRA END USER	Halpern, 2023
MIXAT	Emirati-English Speech Corpus	15 hours	CC BY-NC-SA 4.0	Al Ali and Aldarmaki, 2024
Traditional Emirati	Speech Corpus	4 hours	–	Alblooki et al., 2025
RAMSA	Speech Corpus	41 hours	Restricted DUA	Al-Sabbagh, 2026

Table 1: Published resources explicitly targeting Emirati Arabic or containing a dedicated Emirati component. Reported units differ and should not be interpreted as directly comparable measures of lexical coverage.

2 Related Work

Arabic Dialect Lexicons. Examples of machine-readable dialectal lexicons include the Egyptian Colloquial Arabic Lexicon and the lexical database underlying CALIMA EGY (Kilany et al., 2002; Habash et al., 2012); THARWA, which links Egyptian and Levantine Arabic with MSA and English (Diab et al., 2014); the 25-city-aligned MADAR Lexicon (Bouamor et al., 2018); the Iraqi Arabic Morphological Lexicon and related Iraqi and Moroccan lexicon-development work (Graff et al., 2006; Graff and Maamouri, 2009); MAKNUUNE, a richly structured Palestinian Arabic lexicon (Dibas et al., 2022). Qabas further links a large Arabic lexical inventory to many lexicons and morphologically annotated corpora (Jarrar and Hammouda, 2024). These resources demonstrate several ways of representing dialectal lexical information. Gahwa follows the same general resource-building tradition, but focuses on standardizing and enriching already existing Emirati evidence rather than collecting a new dictionary from a single source.

Emirati Arabic Resources. Over the past decade, Emirati Arabic has been represented in a growing collection of resources for written text, speech, child language, code-mixed language, sociolinguistic analysis, and automatic speech recognition (Ntelitheos and Idrissi, 2017; Khalifa et al., 2017, 2016, 2018; Al Ali and Aldarmaki, 2024; Alblooki et al., 2025; Alkaabi et al., 2026; Al-Sabbagh, 2026). As summarized in Table 1, these resources have substantially expanded the empirical basis for Emirati Arabic NLP. However, they were developed for different tasks and differ in content, annotation, and accessibility, resulting in fragmented lexical evidence rather than a unified computational lexicon.

3 Gahwa Design and Construction

Lexicon Representation. Each Gahwa record represents a lexical entry using the fields shown in Table 2. The `Index` is a unique record identifier. The `Entry` field contains the base word form represented by the record, while `Lemma` gives the lemma linked to that form. The `Type` field marks the relation between the two: `LEMMA` is used when `Entry=Lemma`, and `WORD` is used when `Entry` is a distinct surface or inflected form linked to the lemma in `Lemma`. Each record also stores the morphological `root`, core POS (`POS`) and `CAMeL_POS` (Khalifa et al., 2018), `English Gloss`, `MSA Lemma` equivalent, `CAMeLBERT_freq` or CAMeLBERT-Mix frequency (Inoue et al., 2021), and `source` of the lexical entry, which we present next.

Sources and Extraction. We build Gahwa by extracting candidate records from three sources; Table 3 summarizes their contribution and the subsequent consolidation flow. First, *Gumar Ann* is a ∼200K-token annotated subset of the Emirati portion of the *Gumar* (Khalifa et al., 2016, 2018). It supplies entry forms, lemmas, POS labels, CAMeL POS labels, and English glosses. Second, the AE VERB LEXICON, contains 2,648 verb lemmas with roots, perfective and imperfective paradigm forms, and glosses (Khalifa et al., 2017). Finally, the Gulf Wiktionary export contains 525 semi-structured dictionary records. A single Wiktionary record may bundle several forms, POS blocks, or glosses; splitting these combinations produces candidate records whose pairings are then manually checked by two of the co-authors.

Enrichment. After mapping source fields and POS labels to the lexicon representation (Table 2), we fill fields not supplied consistently by the sources. For *Gumar Ann*, we add roots; for Gulf

Field Name	Description	Example
Index	Unique record identifier.	382
Type	LEMMA or WORD.	LEMMA
Entry	Base word form represented by the record.	رَمَس *ramas*
Lemma	Lemma linked to the represented form.	رَمَس *ramas*
Root	Templatic consonantal root or NTWS (non-templatic word stem).	ر · م · س
POS	Core part-of-speech category.	VERB
CAMeL_POS	Fine-grained part-of-speech morphological category.	VERB.P3MS
English_Gloss	English gloss of the lemma.	speak; speak to someone; talk
MSA_Lemma	Modern Standard Arabic lemma equivalent.	تَكَلَّم takal∼am
CAMeLBERT_Freq	Lemma frequency from CAMeLBERT-Mix counts.	2,122
Source	Original source of the lexical entry.	AE_Verb_Lexicon

Table 2: Illustrative Gahwa record for the verb رَمَس *ramas* 'to speak'.

Wiktionary, we add CAMeL_POS. For AE VERB LEXICON, the paradigm projection produces the candidate records reported in Table 3; during enrichment, we add command forms to these projected lemma and word records. For *Gumar Ann*, candidate roots are extracted from the existing Lemma field and checked by two co-authors with Arabic linguistic expertise; records sharing the same Lemma receive the same checked root. MSA lemma equivalents are then added for all three sources. Candidate MSA lemmas are generated using an LLM from the Gahwa Lemma, POS, and English gloss, then checked and corrected by the same co-authors. This process is described further in the *MSA Lemma Generation and Review* paragraph below. Finally, we compute and retain a CAMeLBERT-Mix lemma frequency for each Gahwa entry (Inoue et al., 2021).

Preparation and Consolidation. After extraction, we prepare records for comparison by normalizing their English glosses. For each record, we split semicolon-separated gloss lists, trim whitespace, remove punctuation, compare strings case-insensitively, remove duplicate items, and sort the remaining items alphabetically. We use this normalized gloss in all later gloss comparisons.

We consolidate records in two passes, first within each source and then across sources. We remove exact duplicates with the key Entry + Lemma + POS + CAMeL_POS + normalized English gloss. Remaining records are merged only in two cases: (i) when POS and normalized gloss agree and lemmas differ only in diacritization, we keep the more diacritically informative lemma, prioritizing shadda and sukun over other diacritics; and (ii) when lemma and POS agree and the normalized gloss lists overlap, we keep their normalized union, retaining the fuller list when one list contains the other. All other records remain separate; each merged group or singleton becomes a consolidated entry.

After consolidation, the distinct lexical-entry unit in Gahwa is lemma-POS-gloss (LPG). The lemma gives the lexical head, POS separates category-specific uses, and the normalized English gloss distinguishes meanings that share the same lemma and POS. As shown in Table 3, the process yields 33,687 consolidated records and 10,456 distinct LPG entries.

MSA Lemma Generation and Review. We generate one candidate MSA lemma for each of the 10,456 consolidated lemma-POS-gloss (LPG) entries. The generation process therefore follows a one-to-one mapping from an LPG entry to a candidate MSA lemma. For each LPG entry, the prompt provides the Gulf Arabic lemma (Lemma), POS, and English gloss that constitute the LPG triplet, together with the associated Entry and dialect code (DIA). We use gpt-5-nano-2025-08-07 to return exactly one candidate MSA lemma in Arabic script.

All 10,456 generated LPG-to-MSA-lemma mappings are manually checked and corrected. One co-author performs the primary review, while a second co-author supports the resolution of ambiguous cases. Disagreements are resolved through discussion. Because the two reviews are not conducted independently, we do not report an inter-annotator agreement statistic. The complete prompt and generation settings are provided in Appendix B.

Inventory	Entries
Gumar Ann records	24,607
AE Verb Lexicon records	7,924
Gulf Wiktionary records	1,161
Consolidated records	33,687
Distinct Lemmas (L)	8,890
Distinct Lemma-POS pairs (LP)	9,339
Distinct Lemma-POS-gloss entries (LPG)	10,456

Table 3: Size of the source-specific, consolidated, and derived lexical inventories in Gahwa.

4 Coverage Evaluation

Evaluation data. To our knowledge, no independent Emirati lexical gold standard is available. We therefore use three Emirati datasets as in-domain diagnostics of practical lexical coverage: *GumarAE*, the Emirati subset of *Gumar* (Khalifa et al., 2016),[2] Mixat (Al Ali and Aldarmaki, 2024), and the Ramsa Corpus (Al-Sabbagh, 2026). For all of these sets, we remove punctuation- or symbol-only items, digit-containing items, foreign-language or web markers, and Arabic strings with attached symbols. The final evaluation corpus sizes are as follows: GumarAE (9,052,103 Tokens, and 442,230 Types), Mixat (91,865 Tokens, and 15,636 Types), and Ramsa Corpus (26,489 Tokens, and 7,873 Types).

Analyzers and Metrics. We use three morphological analysis databases distributed with CAMEL TOOLS (Obeid et al., 2020) for Gulf Arabic (**GLF**), Egyptian Arabic (**EGY**), and Modern Standard Arabic (**MSA**).[3] **ANY** denotes their union. A type is in-vocabulary (INV) if any analyzer returns an analysis, and out-of-vocabulary (OOV) otherwise.

Because the CAMEL TOOLS GLF analyzer does not provide English glosses, we evaluate Gahwa at the `Lemma-POS` (LP) level. POS matching is largely a case-insensitive identity mapping between the two tagsets, e.g. `prep=PREP`. We use two LP matching conditions: *Exact* compares lemma strings as given, whereas dediacritized matching (*Dediac*) strips Arabic diacritics before comparison. In Table 4, LP-match columns use analyzer-covered output as the denominator, while effective coverage uses the full evaluation corpus.

[2]We use Gumar N-grams as our source for this data (Khalifa et al., 2016): https://github.com/CAMeL-Lab/Gumar-Ngrams/releases.

[3]`calima-glf-01` (GLF), `calima-egy-r13` (EGY, only analyses marked `[CALIMA]`), and `camel-morph-msa-v1.2.2` (MSA).

Repeated-letter Normalization (RepNorm). We apply repeated-letter normalization (RepNorm) to all OOV forms by progressively collapsing repeated character sequences. For example, محمووووود → محمود → محمود *mHmwwwwwd* → *mHmwwd* → *mHmwd* 'Mahmoud'. We first reduce each repeated sequence to at most two characters; if the resulting form remains OOV, we reduce it further to a single character. Results are shown in the last row in Table 4. For *GumarAE*, overall analyzer type coverage increases from 63.3% to 84.4% (+21.1 percentage points), while analyzer token coverage rises from 93.3% to 97.3% (+4.0 percentage points). While the *relative* Gahwa token coverage decreases slightly, effective Gahwa token coverage increases from 86.4% to 89.5% Exact and from 88.8% to 92.2% Dediac.

RepNorm has a smaller effect on the two speech-transcript datasets. Analyzer type coverage increases from 95.6% to 95.9% on Mixat and from 95.4% to 95.8% on Ramsa, while analyzer token coverage remains 98.8% on Mixat and increases from 98.3% to 98.4% on Ramsa. Effective Dediac Gahwa token coverage reaches 94.2% on Mixat and 89.1% on Ramsa. These smaller changes indicate that repeated-letter elongation is less common in the two speech-transcript datasets.

We acknowledge that these matches may not necessarily be the correct analysis in context, but represent a possible analysis nonetheless.

Error Analysis. After RepNorm, 15.6% of *GumarAE* corpus types remain unanalyzed, compared with 4.2% of Ramsa types and 4.1% of Mixat types. Because the *GumarAE* residual inventory is considerably larger, we retain the systematic sample of 1,000 residual types from the earlier analyzer run. For the smaller datasets, we examine all remaining OOV types: 331 Ramsa types and 638 Mixat types. All types included in the three error-analysis sets are manually checked and assigned a single dominant error category by the authors. For Ramsa and Mixat, preliminary LLM-assisted labels were used as a starting point, but every label was subsequently manually verified and corrected using the same categories and criteria applied to *GumarAE*. When multiple phenomena overlap, the category judged to be the primary cause of analyzer failure is selected.

Results are in Table 5. Dialect / lexical gap is the largest residual category, comprising 333 sampled types (33.3%) and 6,784 represented corpus

	Analyzer INV		Gahwa Types		Gahwa Tokens		Effective Coverage	
Analyzer	**Types%**	**Tokens%**	**Exact%**	**Dediac%**	**Exact%**	**Dediac%**	**Exact%**	**Dediac%**
				GumarAE				
GLF	37.5	83.7	77.7	93.5	84.4	95.4	70.6	79.8
EGY	35.9	79.3	60.8	75.7	76.7	89.5	60.8	71.0
MSA	52.0	87.3	48.8	69.1	63.8	86.8	55.7	75.7
ANY	63.3	93.3	71.3	80.2	92.6	95.2	86.4	88.8
ANY + RepNorm	84.4	97.3	71.3	80.9	92.0	94.8	**89.5**	**92.2**
				Mixat				
GLF	70.8	89.6	78.9	94.8	83.5	94.0	74.9	84.3
EGY	75.8	91.0	67.1	81.5	75.5	90.1	68.7	82.0
MSA	87.8	95.7	58.6	78.6	65.0	86.7	62.2	82.9
ANY	95.6	98.8	80.7	86.6	92.8	95.3	91.6	94.1
ANY + RepNorm	95.9	98.8	80.6	86.5	92.7	95.3	**91.7**	**94.2**
				Ramsa Corpus				
GLF	66.6	81.7	78.1	94.2	82.9	94.3	67.7	77.1
EGY	76.8	88.1	64.0	77.0	72.6	86.2	64.0	76.0
MSA	90.7	96.3	54.7	75.3	60.8	83.6	58.6	80.6
ANY	95.4	98.3	76.9	82.6	86.5	90.5	85.0	89.0
ANY + RepNorm	95.8	98.4	76.8	82.6	86.5	90.5	**85.1**	**89.1**

Table 4: Coverage and Gahwa lemma-POS agreement across the three Emirati evaluation datasets: **GumarAE** (9,052,103 Tokens, and 442,230 Types), **Mixat** (91,865 Tokens, and 15,636 Types), and **Ramsa Corpus** (26,489 Tokens, and 7,873 Types). ANY is the union of GLF, EGY, and CAMeLMorph MSA v1.2.2.

Category	**GumarAE**		**Ramsa Corpus**		**Mixat**	
	Token%	**Types%**	**Token%**	**Types%**	**Token%**	**Types%**
Dialect / lexical gap	78.6	33.3	45.9	43.8	51.5	43.1
Orthographic / spelling variation	6.3	21.4	12.2	14.2	8.4	11.6
Fused expression / tokenization	6.8	21.0	4.5	5.1	16.2	13.1
Dialect morphology / clitic gap	4.3	11.6	17.7	19.6	15.3	21.9
Proper name / named entity	1.6	3.3	3.8	2.7	1.7	2.1
Loanword / foreign form	1.3	4.3	11.0	10.2	5.3	6.1
Unclear / ambiguous	0.7	5.1	4.5	4.2	1.3	1.8

Table 5: Residual analyzer-OOV categories after RepNorm. GumarAE is based on a systematic sample of 1,000 residual types (8,623 tokens), while the smaller Ramsa and Mixat datasets include all remaining OOV types: 331 (416 tokens) and 638 (1,070 tokens), respectively.

occurrences (78.7%). Examples include ريولي *ry-wly* 'my feet' and آمبيه *Āmbyh* 'oh dear!'. Orthographic variation is next in prevalence (21.4% of types, 6.3% of tokens), such as اتصيح *AtSyH* 'she cries'. Another common category is fused expressions (21% types) such as الحمدلله *AlHmdllh* 'Thank God'. Dialect / lexical gap is also the largest category in Ramsa and Mixat, comprising 43.8% and 43.1% of residual types, respectively. Morphology / clitic gaps are more prominent in Ramsa and Mixat (19.6% and 21.9% of types) than in the *GumarAE* sample (11.6%). Ramsa also contains a larger proportion of loanwords or foreign forms (10.3% of types). Other lower frequency errors include missing proper names, e.g., جميرا *jmyrA* 'Jumeirah'.

5 Conclusion and Future Work

We presented Gahwa, the broadest **open-source** computational lexicon for Emirati Arabic to date, consolidating lexical information from diverse sources. Evaluation on *GumarAE*, Mixat, and the Ramsa Corpus shows that Gahwa achieves 92.2%, 94.2%, and 89.1% effective dediacritized coverage of running tokens; residual OOV analysis identifies the main remaining gaps: dialect vocabulary, orthographic variation, fused expressions, and morphology or clitic gaps.

Future work will expand Gahwa using the residual OOV analysis and improve Emirati morphological analyzers to better support its lexical inventory. We envision Gahwa as a component of the infrastructure for Emirati Arabic NLP.

Limitations

Our coverage evaluation is diagnostic and measures analyzability rather than the correctness of the selected lemmaLemmaPOS analysis. One resource used to construct Gahwa, *Gumar Ann*, is derived from material related to *GumarAE*, although it represents only about 1.5% of its size. Coverage estimates also depend on the available morphological analyzers, whose own lexical limitations may underestimate the true coverage of Gahwa. Finally, both Gahwa and its evaluation are constrained by the currently available Emirati resources. Although we evaluate on *GumarAE*, Mixat, and the Ramsa Speech Corpus, these datasets may still underrepresent lexical variation across domains and varieties of Emirati Arabic.

Ethics Statement

Gahwa is constructed from existing Emirati and Gulf Arabic lexical resources used in accordance with their respective licenses and terms of use. All manual inspection, annotation, and error analysis were performed by the authors.

Like many NLP resources, Gahwa could be misused for applications such as language profiling. We encourage its use only for ethical research and development, in accordance with applicable laws, institutional policies, and professional ethical guidelines.

AI writing assistance was used only for language-level editing of the manuscript, in accordance with the ACL Policy on Publication Ethics; all resource construction, analyses, experiments, and reported results were designed, produced, and verified by the authors.

References

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.

Rania Al-Sabbagh. 2026. Ramsa: A Large Sociolinguistically Rich Emirati Arabic Speech Corpus for ASR and TTS. In *Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026)*, pages 3184–3198, Palma, Mallorca, Spain. European Language Resources Association (ELRA).

Maha Alblooki, Kentaro Inui, and Shady Shehata. 2025. ASR models for traditional emirati Arabic: Challenges, adaptations, and performance evaluation. In *Proceedings of the 8th International Conference on Natural Language and Speech Processing (ICNLSP-2025)*, pages 42–49, Southern Denmark University, Odense, Denmark. Association for Computational Linguistics.

Omar Alkaabi, Ahmed Alzubaidi, Hamza Alobeidli, Shaikha Alsuwaidi, Mohammed Alyafeai, Leen AlQadi, Basma El Amel Boussaha, and Hakim Hacid. 2026. Alyah: An Emirati Dialect Benchmark for Evaluating Arabic Large Language Models. Hugging Face community article.

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. European Language Resources Association (ELRA).

Mona Diab, Mohamed Al-Badrashiny, Maryam Aminian, Mohammed Attia, Heba Elfardy, Nizar Habash, Abdelati Hawwari, Wael Salloum, Pradeep Dasigi, and Ramy Eskander. 2014. Tharwa: A large scale dialectal Arabic - Standard Arabic - English lexicon. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3782–3789, Reykjavik, Iceland. European Language Resources Association (ELRA).

Shahd Salah Uddin Dibas, Christian Khairallah, Nizar Habash, Omar Fayez Sadi, Tariq Sairafy, Karmel Sarabta, and Abrar Ardah. 2022. Maknuune: A large open palestinian Arabic lexicon. In *Proceedings of the Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 131–141, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

David Graff, Tim Buckwalter, Hubert Jin, and Mohamed Maamouri. 2006. Lexicon Development for Varieties of Spoken Colloquial Arabic. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, pages 999–1004, Genova, Italy.

David Graff and Mohamed Maamouri. 2009. Iraqi Arabic Morphological Lexicon (IAML) Version 6.5. Linguistic Data Consortium.

Nizar Habash, Ramy Eskander, and Abdelati Hawwari. 2012. A Morphological Analyzer for Egyptian Arabic. In *Proceedings of the Workshop of the Special Interest Group on Computational Morphology and Phonology (SIGMORPHON)*, pages 1–9, Montréal, Canada.

Jack Halpern. 2023. Arabic dialects full-form lexicon. Technical report, The CJK Dictionary Institute.

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.

Mustafa Jarrar and Tymaa Hasanain Hammouda. 2024. Qabas: An open-source Arabic lexicographic database. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 13363–13370, Torino, Italia. ELRA and ICCL.

Salam Khalifa, Nizar Habash, Dana Abdulrahim, and Sara Hassan. 2016. A large scale corpus of Gulf Arabic. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4282–4289, Portorož, Slovenia. European Language Resources Association (ELRA).

Salam Khalifa, Nizar Habash, Fadhl Eryani, Ossama Obeid, Dana Abdulrahim, and Meera Al Kaabi. 2018. A morphologically annotated corpus of emirati Arabic. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Salam Khalifa, Sara Hassan, and Nizar Habash. 2017. A morphological analyzer for Gulf Arabic verbs. In *Proceedings of the Third Arabic Natural Language Processing Workshop*, pages 35–45, Valencia, Spain. Association for Computational Linguistics.

H. Kilany, H. Gadalla, H. Arram, A. Yacoub, A. El-Habashi, and C. McLemore. 2002. Egyptian Colloquial Arabic Lexicon. LDC catalog number LDC99L22.

Hannes Krumbiegel. 2023. Wiktionary ebook reader dictionaries. `https://github.com/Vuizur/Wiktionary-Dictionaries`. GitHub repository; accessed 21 June 2025.

Dimitrios Ntelitheos and Ali Idrissi. 2017. Language Growth in Child Emirati Arabic. *Perspectives on Arabic Linguistics*, 29:229–248.

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 License

We list below the licenses of the data and tools used in this work, all of which are employed in accordance with their intended use.

- Gumar Corpus N-grams (Khalifa et al., 2016) CC BY 3.0. `https://github.com/CAMeL-Lab/Gumar-Ngrams/releases`
- AE Verb Lexicon (Khalifa et al., 2017): CC BY 3.0.
- Gumar Annotated Corpus (Khalifa et al., 2018): NYU Abu Dhabi Non-commercial, research-only license.
- Morphological analyzers provided by CAMeL Tools (Obeid et al., 2020) and CAMeL-BERT (Inoue et al., 2021): MIT License.
- Wiktionary (Krumbiegel, 2023): CC BY-SA 4.0.
- Camel Morph MSA (camel-morph-msa-v1.2.2): CC BY 4.0.

B MSA Lemma Generation and Review

We use `gpt-5-nano-2025-08-07` to generate one candidate MSA lemma for each of the 10,456 lemma-POS-gloss (LPG) entries. The generation process is therefore a one-to-one mapping: The LPG triplet consists of the Gahwa `Lemma`, POS, and English gloss. The associated `Entry` and dialect code (`DIA`) are also included in the request. All 10,456 generated MSA lemmas are manually checked and corrected by the co-authors.

No temperature, top-p, seed, or maximum output-token setting is explicitly specified; the API defaults are used.

System Prompt

```
Model used for the final MSA lemmas: gpt-5-nano-2025-08-07
Number of LLM-generated MSA lemmas/LLM checked Lemmas: 10,456 LPG entries
Full system prompt:
You are an expert Arabic linguist specializing in Gulf–MSA lexical mapping.

You will receive several columns taken from a Gulf Arabic lexicon.
Use ALL of the following columns when determining the correct Modern Standard Arabic (MSA) lemma:

1. Lex : the Gulf Arabic lemma (dialectal base form), typically fully diacritized.
   - This is the primary and most reliable piece of information.
   - Treat it as the strongest anchor for identifying the correct MSA lemma.

2. POS : the part of speech.
   - Use this to determine the correct MSA lemma form:
       • VERB : return the MSA lemma in its dictionary form: the fully diacritized 3rd person
                masculine singular perfective
       • NOUN : return the fully diacritized singular indefinite nominative form
       • ADJ : return the fully diacritized masculine singular indefinite form

3. Entry : the surface form as it appears in corpus data.
   - Use this only as a secondary hint if needed.

4. DIA : dialect code (e.g., AE).
   - Use this to understand the intended Gulf variety and resolve dialect-specific meanings.

5. English Gloss : a concatenated or noisy gloss.
   - Use it ONLY as a weak sense-disambiguation hint.
   - If the gloss conflicts with Lex or POS, ignore the gloss completely.

============================
 DEFINITION OF AN MSA LEMMA
============================

A lemma is the canonical, dictionary citation form of a word. It must follow these rules:

-----------------
 VERB Lemmas
-----------------
• Return the fully diacritized 3rd person masculine singular perfective form.
• Do not include any mood diacritic.
• No attached particles or clitics of any kind (such as و، ف، سـ، لم، لا or pronoun suffixes).
• Use the dialect lemma's diacritization as a hint but correct it to match the proper MSA lemma.

==============================
 GENERAL RULES FOR ALL LEMMAS
==============================

• The lemma must be:
    - Fully diacritized
    - A single canonical stem
    - Using underscores "_" instead of spaces
    - Without attached particles, clitics, conjunctions, prepositions, or determiners
    - Without case endings or verb mood endings

• The lemma must not:
    - Contain case endings
    - Contain verb mood endings
    - Contain the determiner al-
    - Contain plural or dual suffixes (for nouns/adjectives)
    - Be undiacritized or partially diacritized

Your task:
- Map the Gulf Arabic lemma to exactly ONE correct MSA lemma.
- The result must be:
    • a single MSA lemma
```

```
    • in Arabic script
    • with diacritics
    • with correct spelling
    • using underscores `_` instead of spaces

Output only valid JSON in exactly this format:

{"msa_lemma": "<single MSA lemma with underscores>"}

Do NOT output multiple lemmas.
Do NOT output semicolons.
Do NOT output anything else: no explanations, no comments, no transliterations.
```

Arabic Safety Alignment as Selective Refusal: An Empirical Study of SFT, DPO, and Guard Calibration

Mohamad Zbib
American University of Beirut
mbz02@mail.aub.edu

Ammar Mohanna
American University of Beirut
am288@aub.edu.lb

Abstract

Arabic large language models must refuse harmful prompts without over-refusing benign or sensitive prompts, yet a single refusal rate hides this trade-off. We evaluate it using benign refusal B and harmful-prompt refusal H, where H measures refusal rather than harmful compliance. Across five Arabic-capable models and 130 runs on the full human-written AraSafe set, refusal-only supervised fine-tuning (SFT) collapses toward blanket refusal, whereas selected mixed-SFT configurations reach $H \approx 90$–93% at $B = 14$–23%; four selected configurations exceed $H = 90\%$ in all three runs, while Fanar does so in two of three. Direct Preference Optimization (DPO) and inference guards change B and H differently across models rather than acting as uniform upgrades. In a blinded 300-response audit, annotator binary-refusal agreement is 89.0% ($\kappa = 0.78$); Qwen3Guard and Aya Expanse 32B reach 88.7% and 91.0% accuracy, respectively, with no conclusive paired difference. Selected SFT raises H on Arabizi for all five models, but none reaches 90%, showing only partial transfer from Modern Standard Arabic; overall, the results support model-specific operating-point selection: set a deployment target and retain only interventions that improve it.

1 Introduction

Arabic safety is not a single input space. Users write in Modern Standard Arabic (MSA), regional dialects such as Egyptian and Levantine, romanized Arabizi, and noisy text, and many benign prompts are sensitive: news, medical, legal, and policy questions that share vocabulary with genuinely harmful requests (Ashraf et al., 2025; Mousi et al., 2025). A safety method for Arabic must therefore refuse harmful prompts while staying usable across these forms. A single refusal rate cannot capture this requirement because a model can appear protective simply by refusing everything. The practical question is whether standard alignment interventions improve this trade-off consistently across Arabic-capable models and writing forms.

We evaluate selective refusal with two rates: benign refusal $B = P(\text{refusal} \mid \text{benign prompt})$ and harmful-prompt refusal $H = P(\text{refusal} \mid \text{harmful prompt})$, measured on the human-written AraSafe benchmark (Mubarak et al., 2025). Crucially, H records whether a response refuses; its complement is not automatically harmful compliance. A useful intervention must lower B or raise H without silently damaging the other axis, so every method is judged by movement on the (B, H) plane rather than either rate alone (Röttger et al., 2024; Cui et al., 2024).

Figure 1 motivates the study: five Arabic-capable models begin at different measured locations in the (B, H) plane. This variation motivates model-by-model testing; it does not establish universal model categories or transferable intervention rules. Our central claim is narrower: *Arabic selective refusal is a model-specific operating-point selection problem: measure benign refusal B and harmful-prompt refusal H, specify the deployment target, and retain an intervention only when it improves that target.*

Contributions.

- **Audited evaluation.** We audit B/H labels with 300 blinded responses and a cross-family judge.
- **Model-specific SFT trade-offs.** Refusal-only SFT collapses, while selected mixed-SFT configurations reach $H \approx 90$–93% at $B = 14$–23%.
- **No uniform post-training upgrade.** DPO and guards move models differently; ordering is a null result.
- **Partial cross-script transfer.** Selected SFT raises Arabizi H for every model, but none reaches 90%.

Proceedings of The Fourth Arabic Natural Language Processing Conference (ArabicNLP 2026), pages 228–246
October 28-29, 2026 ©2026 Association for Computational Linguistics

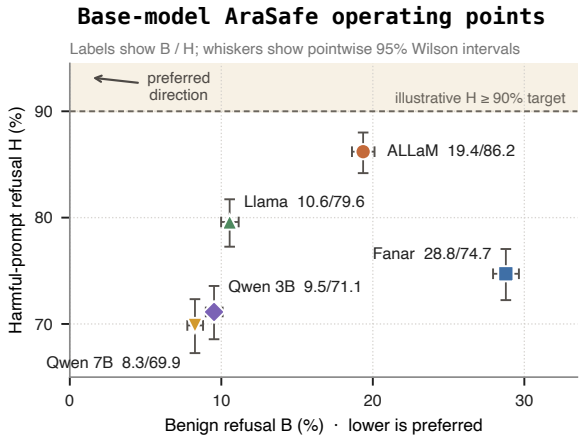


Figure 1: Base-model operating points on AraSafe. Lower benign refusal B and higher harmful-prompt refusal H are preferred; labels report B/H. Whiskers show marginal 95% Wilson intervals for prompt sampling conditional on the automatic judge labels ($n_B = 10{,}823$, $n_H = 1{,}254$), reconstructed from retained rates. The dashed line marks the illustrative $H \geq 90\%$ target, not a universal safety standard.

2 Measuring and Validating Selective Refusal

Metrics and operating points. An operating point is one measured pair (B, H). One tested point dominates another when it has lower or equal B and higher or equal H, with at least one strict inequality; the tested frontier is the non-dominated subset of observed points. We use $H \geq 90\%$ as an illustrative selection constraint and choose the lowest-B feasible candidate. This is not a universal safety standard: a deployment owner must set the target from its own costs, and we report sensitivity to 85%, 90%, and 95% in Table 10.

Evaluation set and judge. We evaluate on the full human-written portion of AraSafe (Mubarak et al., 2025): 12,077 prompts, of which 10,823 are benign and 1,254 harmful. We reserve these prompts for evaluation; none appears in training, direction extraction, or calibration. Because AraSafe is MSA-heavy, we treat the operating points in Sections 3 and 4 as MSA measurements and stress-test them across forms in Section 5. Qwen3Guard-Gen-4B (Zhao et al., 2025; Souly et al., 2024) supplies the binary refusal label.

Human and cross-judge validation. We audited a blinded, stratified sample of 300 responses with two independent annotators and adjudication. Binary-refusal agreement was 89.0% ($\kappa = 0.78$). Against adjudicated labels, Qwen3Guard achieved 88.7% accuracy and 0.887 macro-F1, with a clustered-bootstrap 95% accuracy interval of [85.3, 91.7]; Aya Expanse 32B (Dang et al., 2024) achieved 91.0% accuracy and 0.910 macro-F1. Their paired difference was inconclusive (McNemar $p = 0.382$; Qwen-minus-Aya macro-F1 difference -2.3 points, 95% interval $[-6.4, 2.0]$). Qwen3Guard accuracy was 86.7% on Qwen-family outputs and 90.0% otherwise; the gap interval $[-3.3, 10.3]$ neither establishes nor excludes family bias. Reported form-specific accuracies range from 87.5% to 95.0% on MSA, Egyptian, Levantine, and noisy Arabic subsets, but fall to 77.5% on Arabizi, where macro-F1 is 0.763. Full audit statistics appear in Table 7.

Models. The five base models span sizes and providers: Qwen2.5-3B and Qwen2.5-7B Instruct, Meta-Llama-3-8B Instruct, Fanar-1-9B, and ALLaM-7B Instruct. Two are Arabic-centric and three are multilingual; neither provenance group occupies one common starting location in Figure 1. Full identifiers are in Appendix F.

Training and preference data. Harmful SFT and preference prompts come from an MSA translation of BeaverTails (Ji et al., 2023), translated with the Hala English–Arabic models (Hammoud et al., 2025); benign SFT examples are separately drawn from Hala-4.6M-SFT. DPO uses 10K benign and 10K harmful preference pairs: harmful pairs rank a refusal over a harmful completion, while benign pairs rank either two helpful answers (V1) or a helpful answer over a refusal (V2) from the Arabic Data Is Better Together collection (Data Is Better Together, 2024). Because the harmful supervision is translated MSA, cross-script transfer is an empirical question rather than an assumption.

Uncertainty and seeds. Wilson intervals quantify prompt-sampling uncertainty conditional on the automatic labels; they do not include judge measurement error, which the audit exposes separately. With 10,823 benign and 1,254 harmful prompts, B half-widths are below one point at the reported rates; H half-widths are about 1.5 points near the selected $H \approx 90\%$ points and 1.8–2.6 points at the base points in Figure 1. Main-text rates are rounded to one decimal place, while appendix tables retain the available source precision; a rounded boundary value is not treated as stable evidence of threshold satisfaction. Primary sweeps use seed 42. Selected mixed-SFT configurations were repeated

Table 1: Notation and scope. Refusal is measured independently on benign and harmful prompts; H is not a harmful-compliance rate.

Term	Meaning
B	$P(\text{refusal} \mid \text{benign prompt})$; false-positive refusal rate
H	$P(\text{refusal} \mid \text{harmful prompt})$; refusal, not harmful compliance
Operating point	One measured pair (B, H) for a model and intervention
Tested frontier	Non-dominated subset of the operating points actually evaluated

Table 2: Data roles in the operating-point search. Evaluation prompts are held out from training and calibration; the remaining stages use the sources and splits listed below.

Split	Used for	Why it matters
AraSafe	Main B/H evaluation	Measures Arabic selective refusal
SFT and DPO data	Training	Supplies helpful and refusal pressure
Boundary set	Stress test only	Checks dialect, Arabizi, noise, and sensitive benign generalization
Activation split	Direction extraction	Learns candidate linear refusal directions
Calibration split	Threshold tuning	Selects per-model thresholds separately from extraction

with seeds 43 and 44: across models, standard deviations span 0.13–0.57 points for B and 0.41–1.10 for H. ALLaM, Llama, Qwen 3B, and Qwen 7B exceed $H = 90\%$ in all three runs; Fanar does so in two of three, with mean $H = 89.95 \pm 0.41$, and is therefore threshold-sensitive. Direction AUC intervals use the Hanley–McNeil estimator.

Scope of the starting points. The five observed base points differ substantially: ALLaM starts with comparatively high H, Fanar with the highest B, and the Qwen and Llama models with lower B but also lower H. These are measured operating points, not discovered clusters, causal diagnoses, or intervention prescriptions. We therefore search candidate operating points separately for each model. Table 1 makes the paper's notation and inferential scope explicit, while Table 2 records which data touch each stage.

3 SFT Candidate Search

Refusal-only SFT collapses toward blanket refusal. We use refusal-only training (Qi et al., 2024) as a diagnostic. As Figure 2 shows, harmful-prompt refusal H saturates within a few steps, while benign refusal B rises toward total refusal: ALLaM, Fanar, and Llama cross $B = 80\%$ by step 10, and both Qwen models by step 30. Thus, refusal pressure must be bounded by benign behavior.

Mixed SFT exposes local candidate points. Adding helpful examples recovers selectivity, but the ratio is a model-specific search (Figure 3). For example, ALLaM moves from $28.1/94.7$ at 70/30 benign/refusal to $17.7/92.4$ at 95/5, whereas Fanar falls to $8.6/79.0$ at 95/5 and needs more refusal pressure among the tested points. Ratio coverage is uneven: ALLaM has six tested mixtures, Fanar five, and the other models three. Exposure and training-budget dimensions were not fully equalized. These sweeps therefore identify local candidates and best-observed points under the illustrative target; they do not estimate causal ratio effects, universal optima, or transferable configurations.

Ordering is a null result. Figure 4 reports the descriptive margin $H - B$ for four file-order labels. Random ordering has the largest reported margin in 4 of 15 model-ratio rows (26.7%), close to the 25% chance rate among four strategies, and many differences are smaller than prompt-sampling intervals. Moreover, default reshuffling meant file construction did not guarantee realized optimizer order. We retain the labels to identify runs but make no ordering-benefit or causal claim.

Checkpoint variation. On the 70/30 mix, ALLaM's B falls from 39.9 at checkpoint 50 to 19.5 at checkpoint 200 while H remains above 92, whereas Llama already over-refuses at checkpoint 50 ($35.1/96.7$).

Selected candidates. Table 3 applies the illustrative rule to the tested runs. The selected points reach about 90–93% harmful-prompt refusal at 14.0–22.6% benign refusal. These are standardized downstream comparison points, not global optima. Four remain above $H = 90\%$ in all three selected-configuration runs. Fanar crosses the threshold in only two of three, with mean $H = 89.95 \pm 0.41$, so its rounded 90.0% seed-42 point must be read as threshold-sensitive.

4 Calibration After SFT

Methods after SFT are candidate moves, not default stages. We compare DPO and guard moves on the (B, H) plane while reporting marginal prompt-sampling intervals; these intervals are descriptive



Figure 2: Refusal-only SFT collapses toward blanket refusal. In each model panel, benign refusal B (blue) and harmful-prompt refusal H (orange) are plotted over training steps. The dashed horizontal line marks $B = 80\%$; the circled marker identifies the first crossing.

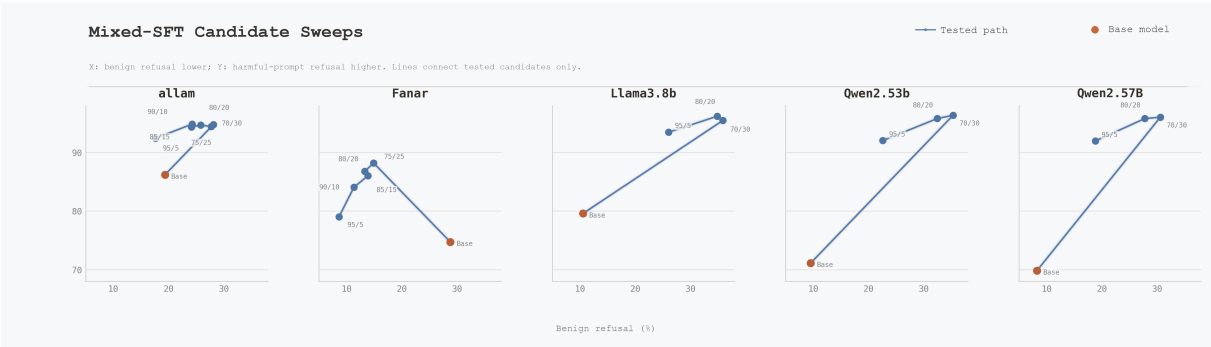


Figure 3: Observed mixed-SFT ratio sweeps. Each panel connects the base point (orange) to the tested benign/refusal mixtures (blue) in the (B, H) plane. Coverage is uneven across models, so the paths are candidate searches rather than causal or transferable ratio curves.

Table 3: Selected mixed-SFT candidates under the illustrative $H \geq 90\%$ rule. Fanar† meets the threshold in two of three seeds (89.95 ± 0.41 mean H). Ordering abbreviations identify runs but do not imply an ordering effect.

Model	Candidate	B	H
ALLaM	95/5 BF	16.5	92.3
Fanar†	80/20 INT	14.0	90.0
Llama	95/5 RF	22.0	92.8
Qwen 3B	95/5 RF	22.6	92.8
Qwen 7B	95/5 RF	18.6	92.4

rather than a formal acceptance test for paired changes.

DPO moves models in opposite directions. We apply Direct Preference Optimization (Rafailov et al., 2023) after SFT with two benign-pair constructions. DPO is a trade-off operator, not a uniform upgrade. Qwen 7B moves from $18.6/92.4$ to $13.7/89.5$; ALLaM and Qwen 3B move similarly, to $16.3/88.1$ and $17.9/86.1$. Llama instead moves from $22.0/92.8$ to $31.3/96.2$. Thus only the deployment objective determines whether a move is acceptable (Figure 5). Wilson intervals describe prompt sampling conditional on judge labels, so we do not call a move "real" solely because it exceeds one half-width. Fanar's DPO V2 point is a labeled contextual proxy and is excluded from direct comparisons; Appendix A.5 gives its construction.

An observed configuration failure. Under DPO V1, one Llama run labeled benign-first reaches $29.5/94.9$, while the harmful-first-labeled run reaches $74.8/89.3$. Because realized optimizer order was not controlled and this is a single-seed comparison, it demonstrates configuration fragility rather than a causal ordering effect.

Guards are model- and stage-dependent. Inference guards (Meta, 2024) are similarly heterogeneous (Table 4). At base, the guard moves Fanar from $28.8/74.7$ to $5.4/82.2$. After SFT, however, it lowers H for four of five models; only Fanar gains harmful-prompt refusal, from 90.0 to 93.7, while paying an eight-point B cost. Llama Guard 3 does not officially list Arabic among its supported languages, so this use is off-label. These results

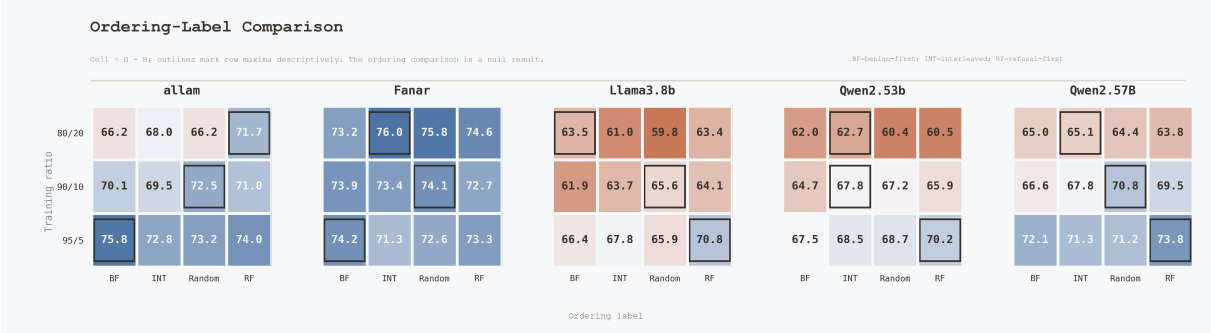


Figure 4: Descriptive ordering comparison. Cells report $H - B$; black outlines mark the largest margin in a model-ratio row. Random is largest in 4/15 rows, near the 25% chance rate, and reshuffling did not preserve file order as optimizer order. We therefore treat this as a null result. BF, INT, and RF denote benign-first, interleaved, and refusal-first labels.

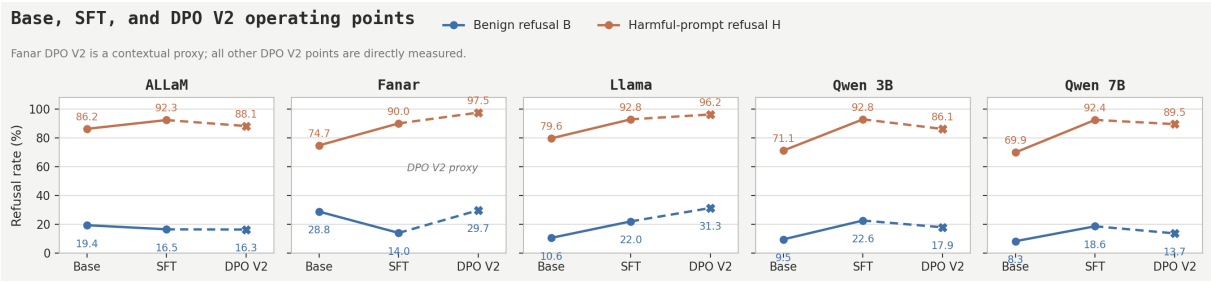


Figure 5: Base, selected-SFT, and DPO V2 operating points. Panels plot benign refusal B (blue) and harmful-prompt refusal H (orange). Solid segments show Base-to-SFT movement; dashed segments and x markers show SFT-to-DPO V2 transitions. Fanar's DPO V2 point is a contextual proxy; the other four are directly measured.

Table 4: Inference guard before and after SFT, reported as B/H. Effects depend on model and stage; after SFT the guard lowers harmful-prompt refusal for four of five models. Llama Guard 3 does not officially support Arabic, so this is off-label use.

	Base		SFT	
Model	no guard	guard	no guard	guard
ALLaM	19.4/86.2	19.4/86.7	16.5/92.3	16.9/89.2
Fanar	28.8/74.7	5.4/82.2	14.0/90.0	22.0/93.7
Llama	10.6/79.6	14.0/81.7	22.0/92.8	18.5/89.9
Qwen 3B	9.5/71.1	10.1/72.8	22.6/92.8	14.8/82.4
Qwen 7B	8.3/69.9	8.8/71.5	18.6/92.4	11.6/86.9

motivate Arabic-native guards and selection rules that evaluate both axes, rather than a default stage.

5 Cross-Script Transfer and Robustness

The AraSafe evaluation set is MSA-heavy, but Arabic users write in dialects, Arabizi, and noisy text. We use the 730-prompt boundary set to compare base behavior across five forms and to test the selected SFT checkpoints on Arabizi. These are controlled synthetic diagnostics, not representative samples of naturally occurring user language.

Transformation audit. Two annotators audited 120 transformed prompts, 30 per non-MSA form, with 85.8% agreement ($\kappa = 0.63$). Egyptian and Levantine each had 27/30 prompts rated natural or mostly natural. Arabizi and noisy Arabic each had 22/30 natural or mostly natural, 6/30 understandable but unnatural, and 2/30 invalid or meaning-changing prompts. These results support controlled stress tests but not broad claims about naturally occurring language; full counts are in Table 9.

Base behavior changes across forms. Figure 6 and the complete values in Table 27 show the sharpest change on Arabizi. Relative to boundary-set MSA, H falls to 22.9%, 30.0%, and 43.8% for Qwen 3B, Qwen 7B, and Fanar. ALLaM and Llama retain more harmful-prompt refusal but their B rises to 45.0% and 49.6%. Egyptian and Levantine degradation is generally milder, although not absent. These are model-specific shifts, not evidence that one starting-point label predicts a fixed cross-script failure.

Selected SFT transfers only partially. Table 5 reports the selected-checkpoint evaluation. SFT raises Arabizi H for every model, but none reaches

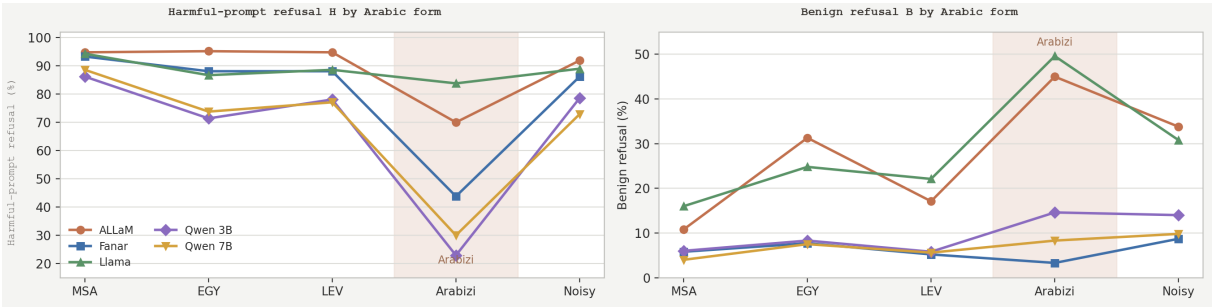


Figure 6: Base-model refusal on the synthetic 730-prompt boundary set. The left panel plots harmful-prompt refusal H and the right plots benign refusal B for MSA, Egyptian, Levantine, Arabizi, and noisy Arabic. The shaded band highlights Arabizi, where the largest shifts occur through either reduced H or inflated B.

Table 5: Arabizi base-to-selected-SFT transfer. Each cell gives B/H on the first line and the corresponding Wilson 95% intervals on the second. Intervals are reconstructed from the rounded rates and known denominators ($n_B = 520$, $n_H = 210$), so they quantify prompt sampling conditional on the judge labels.

Model	Base	Selected SFT
ALLaM	45.0/70.0 [40.8,49.3]/[63.5,75.8]	43.3/74.8 [39.1,47.6]/[68.5,80.2]
Fanar	3.3/43.8 [2.1,5.2]/[37.3,50.6]	2.5/53.3 [1.5,4.2]/[46.6,60.0]
Llama	49.6/83.8 [45.3,53.9]/[78.2,88.2]	57.1/88.6 [52.8,61.3]/[83.6,92.2]
Qwen 3B	14.6/22.9 [11.8,17.9]/[17.7,29.0]	19.6/34.8 [16.4,23.2]/[28.6,41.4]
Qwen 7B	8.3/30.0 [6.2,11.0]/[24.2,36.5]	11.2/43.3 [8.7,14.1]/[36.8,50.1]

90% and three remain below 55%. ALLaM and Llama also retain high benign refusal. Training only with translated MSA harmful supervision is a plausible contributor, but without a native-Arabizi ablation this is a hypothesis rather than a causal explanation. Every deployment-relevant form therefore needs its own evaluation and uncertainty analysis.

Non-refusal is heterogeneous. Lower H means that the judge labels fewer harmful-prompt responses as refusals; it does not reveal what the other responses contain. The five-way human taxonomy distinguishes effective refusal, mixed or leaky refusal, safe redirection, confused or irrelevant response, and harmful or enabling response. In the audit, the 44 harmful-prompt responses without effective refusal comprise 25 harmful or enabling responses, 9 mixed or leaky refusals, 6 confused or irrelevant responses, and 4 safe redirections. We do not report a separate Arabizi outcome taxonomy because that subgroup contains only ten such responses. Inclusive Arabic safety evaluation must therefore report both refusal and response outcomes.

6 Selecting and Interpreting Operating Points

The portable result is a procedure. Exact ratios, checkpoints, and thresholds are specific to five models, tested configurations, and evaluation sets. The portable part is to measure (B, H), specify a deployment target, search candidate interventions, and retain only target-improving points that remain stable across seeds, judges, and relevant Arabic forms. For Qwen 7B, selected SFT moves $8.3/69.9$ to $18.6/92.4$, while DPO V2 trades to $13.7/89.5$; neither point is universally better. Fanar's selected $14.0/90.0$ point improves both seed-42 axes relative to $28.8/74.7$, but its three-seed mean falls just below the illustrative threshold. The procedure surfaces these choices instead of hiding them inside a fixed stack.

Internal signals are detectable but model-specific. Refusal directions (Arditi et al., 2024) separate harmful from benign responses with AraSafe AUCs of 0.909–0.956 across the five models (95% half-width near 0.01). Score scales and thresholds differ by model, matching conditional activation control (Lee et al., 2025); directions therefore support local calibration rather than a universal rule. Per-model threshold operating points and score distributions are in Appendix D.

Choosing an operating point. No single intervention dominates. An assistant that must stay helpful on sensitive but benign Arabic may prioritize low B, while a moderation setting may prioritize high H. The 85/90/95 sensitivity in Table 10 shows that tightening the target from 90% to 95% costs 7.8–23.4 additional B points among feasible tested

models, and Fanar has no feasible tested point at 95%. The target is therefore a deployment choice, not a benchmark constant.

The protocol in four steps. The procedure can be stated compactly:

1. **Measure.** Estimate the base (B, H) point and judge uncertainty on the deployment distribution.
2. **Specify.** Set the deployment target and costs before selecting a configuration.
3. **Search and gate.** Sweep local SFT, DPO, or guard candidates and retain only target-improving points.
4. **Validate.** Repeat selected candidates across seeds, independent judges, and every relevant script or dialect.

Takeaway. The central result is not a universal SFT ratio, DPO variant, guard, ordering, or activation threshold. Arabic selective refusal is a model-specific operating-point selection problem, and MSA performance does not establish cross-script reliability. Refusal metrics must also remain distinct from harmful-response outcomes.

7 Related Work

Our work sits at the intersection of over-refusal evaluation, Pareto safety alignment, and Arabic LLM evaluation. XSTest (Röttger et al., 2024) and OR-Bench (Cui et al., 2024) document benign over-refusal, while Panacea (Zhong et al., 2024) treats helpfulness and harmlessness as competing objectives; our (B, H) plane makes the refusal trade-off explicit for Arabic.

Fine-tuning can erode safety even with benign data (Qi et al., 2024), DPO optimizes preferences without a reward model (Rafailov et al., 2023), and Deliberative Alignment (Guan et al., 2024) reasons over a safety specification; we treat each as a candidate move rather than a fixed stack. Refusal is partly linear in activations (Arditi et al., 2024), and Conditional Activation Steering (Lee et al., 2025) supports model-conditional control.

Arabic resources include Jais (Sengupta et al., 2023), AceGPT (Huang et al., 2024), Fanar (Fanar Team, 2025), ALLaM (Bari et al., 2025), AraSafe (Mubarak et al., 2025), safeguard evaluation (Ashraf et al., 2025), and AraDiCE (Mousi et al., 2025). Arabic diacritics also change tokenization and benchmark behavior (Inoue et al., 2026); testing their effect on refusal is important future work.

Limitations

Five limitations bound the conclusions.

Judge. Human and Aya validation improve confidence, but Qwen3Guard accuracy is lower on Arabizi, and the family-gap interval does not exclude bias. Wilson intervals cover prompt sampling conditional on judge labels, not judge error. The retained audit record also lacks its exact stratification allocation and the subgroup denominators and intervals for the reported per-form accuracies.

Variance. Primary sweeps are seed 42; only selected SFT configurations use three seeds, and Fanar meets $H \geq 90\%$ in two of three.

Search design. Ratio coverage and exposure are unequal across models, so best-observed candidates are not causal optima.

Data. Harmful supervision is translated MSA; native dialectal and Arabizi ablations are needed before attributing transfer failures to training data.

Reproducibility and scope. The transformed prompts are synthetic; the exact GPT-4o-mini snapshot, noise implementation, full selected-SFT cross-form matrix, per-seed measurements, exact ratio sample counts, hardware, runtime, total compute, and a public artifact URL are not available in the current record. We do not infer or invent these details.

Ethics Statement

This work studies how to make Arabic language models refuse harmful requests while staying usable. Harmful prompts and enabling continuations are redacted and reported only as outcome labels (Table 28). We use synthetic transformations only as controlled robustness diagnostics. We disclose lower naturalness for Arabizi transformations and lower judge agreement on Arabizi responses. The Fanar DPO proxy is marked and excluded from direct comparisons. We distinguish refusal from response safety: non-refusal may be harmful, mixed, confused, or a safe redirection. Deployment therefore requires Arabic-native data, representative human evaluation, and outcome-level oversight in addition to refusal rates.

Acknowledgments

We thank Mohamad Bazzi, Mariam Salman, Rana Ezzeddine, Mohamad Hussein Karnib, and Hassan Hijazi.

References

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. 2024. Refusal in language models is mediated by a single direction. In *Advances in Neural Information Processing Systems*, volume 37.

Yasser Ashraf, Yuxia Wang, Bin Gu, Preslav Nakov, and Timothy Baldwin. 2025. Arabic dataset for LLM safeguard evaluation. 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 5529–5546.

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, Majed Alrubaian, Ali Alammari, Zaki Alawami, and 7 others. 2025. ALLaM: Large language models for Arabic and English. In *International Conference on Learning Representations (ICLR)*.

Justin Cui, Wei-Lin Chiang, Ion Stoica, and Cho-Jui Hsieh. 2024. OR-Bench: An over-refusal benchmark for large language models. *Preprint*, arXiv:2405.20947.

John Dang, Shivalika Singh, Daniel D'souza, Arash Ahmadian, Alejandro Salamanca, Madeline Smith, Aidan Peppin, Sungjin Hong, Manoj Govindassamy, Terrence Zhao, Sandra Kublik, Meor Amer, Viraat Aryabumi, Jon Ander Campos, Yi-Chern Tan, Tom Kocmi, Florian Strub, Nathan Grinsztajn, Yannis Flet-Berliac, and 26 others. 2024. Aya Expanse: Combining research breakthroughs for a new multilingual frontier. *Preprint*, arXiv:2412.04261.

Data Is Better Together. 2024. DPO datasets for AR.

Fanar Team. 2025. Fanar: An Arabic-centric multimodal generative AI platform. *Preprint*, arXiv:2501.13944.

Melody Y. Guan, Manas Joglekar, Eric Wallace, Saachi Jain, Boaz Barak, Alec Helyar, Rachel Dias, Andrea Vallone, Hongyu Ren, Jason Wei, Hyung Won Chung, Sam Toyer, Johannes Heidecke, Alex Beutel, and Amelia Glaese. 2024. Deliberative alignment: Reasoning enables safer language models. *Preprint*, arXiv:2412.16339.

Hasan Abed Al Kader Hammoud, Mohammad Zbeeb, and Bernard Ghanem. 2025. Hala technical report: Building Arabic-centric instruction & translation models at scale.

Huang Huang, Fei Yu, Jianqing Zhu, Xuening Sun, Hao Cheng, Dingjie Song, 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.

Go Inoue, Bashar Alhafni, Nizar Habash, and Timothy Baldwin. 2026. Do diacritics matter? evaluating the impact of Arabic diacritics on tokenization and LLM benchmarks. In *Findings of the Association for Computational Linguistics: EACL 2026*, pages 426–442.

Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. 2023. BeaverTails: Towards improved safety alignment of LLM via a human-preference dataset. In *Advances in Neural Information Processing Systems*, volume 36.

Bruce W. Lee, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Erik Miehling, Pierre Dognin, Manish Nagireddy, and Amit Dhurandhar. 2025. Programming refusal with conditional activation steering. In *International Conference on Learning Representations (ICLR)*.

Meta. 2024. Llama Guard 3.

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 (COLING)*, pages 4186–4218.

Hamdy Mubarak, Abubakr Mohamed, and Majd Hawasly. 2025. AraSafe: Benchmarking safety in Arabic LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 9976–9992. Association for Computational Linguistics.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. 2024. Fine-tuning aligned language models compromises safety, even when users do not intend to! In *The Twelfth International Conference on Learning Representations (ICLR)*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36.

Paul Röttger, Hannah Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. 2024. XSTest: A test suite for identifying exaggerated safety behaviours in large language models. 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 5377–5400.

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. *arXiv preprint arXiv:2308.16149*.

Alexandra Souly, Qingyuan Lu, Dillon Bowen, Tu Trinh, Elvis Hsieh, Sana Pandey, Pieter Abbeel, Justin Svegliato, Scott Emmons, Olivia Watkins, and Sam Toyer. 2024. A StrongREJECT for empty jailbreaks. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024) Datasets and Benchmarks Track*.

Haiquan Zhao, Chenhan Yuan, Fei Huang, Xiaomeng Hu, Yichang Zhang, An Yang, Bowen Yu, Dayiheng Liu, Jingren Zhou, Junyang Lin, Baosong Yang, Chen Cheng, Jialong Tang, Jiandong Jiang, Jianwei Zhang, Jijie Xu, Ming Yan, Minmin Sun, Pei Zhang, and 24 others. 2025. Qwen3Guard technical report.

Yifan Zhong, Chengdong Ma, Xiaoyuan Zhang, Ziran Yang, Haojun Chen, Qingfu Zhang, Siyuan Qi, and Yaodong Yang. 2024. Panacea: Pareto alignment via preference adaptation for LLMs. In *Advances in Neural Information Processing Systems*, volume 37.

A Experimental and Data Details

Primary sweeps and sampled splits use seed 42; selected mixed-SFT configurations are repeated with seeds 43 and 44. The 12,077 human-written AraSafe prompts (Mubarak et al., 2025) are reserved exclusively for downstream evaluation with Qwen3Guard-Gen-4B (Zhao et al., 2025); they never appear in training, direction extraction, or calibration splits.

A.1 Preference and SFT Data

DPO data (10K harmful and 10K benign pairs). Harmful prompts come from an MSA translation of BeaverTails (Ji et al., 2023), produced with the Hala English–Arabic translators (Hammoud et al., 2025). For each prompt, the rejected side is the harmful BeaverTails completion and the chosen side is a refusal. The benign side is sourced separately from `data-is-better-together/dpo-datasets-for-ar` (Data Is Better Together, 2024). V1 ranks two helpful assistant answers; V2 replaces the rejected member with a refusal so the model learns to prefer a helpful answer over benign over-refusal.

SFT data. The harmful portion uses the same BeaverTails MSA translation with refusals as the chosen response. The benign portion is drawn separately from `hammh0a/Hala-4.6M-SFT` (Hammoud et al., 2025). We test benign/refusal ratios 70/30, 75/25, 80/20, 85/15, 90/10, and 95/5, plus a 100% refusal collapse control. Coverage is not balanced across models, and exact example counts per tested ratio are absent from the retained record; consequently, the sweep does not isolate ratio from exposure or training budget.

A.2 Training Configuration

All runs use batch size 32, Paged AdamW 8-bit, a linear schedule with five warm-up steps, gradient clipping at 1.0, gradient checkpointing, logging every two steps, and full-parameter training. Table 6 records the available run-specific settings.

Table 6: Training configuration. Selected SFT candidates additionally use seeds 43 and 44; all primary sweeps use seed 42.

Run	Budget	Learning rate	Precision	Length	Seed
Standard SFT	1 epoch	10^{-5}	FP16	1024	42
Collapse SFT	$\leq$ 100 steps	10^{-5}	BF16	512	42
DPO	1 epoch	5×10^{-7}	BF16	1024/512 prompt	42

DPO uses $\beta = 0.1$. Hardware, runtime, total compute, and exact per-ratio sample counts were not retained in the current artifact.

A.3 Ordering and Refusal Directions

Ordering. For DPO and SFT, files are constructed with benign-first, harmful-first, interleaved, or random labels. Default training reshuffling means these labels do not guarantee realized optimizer order. The comparison is therefore retained as run metadata and a null result, not as evidence that ordering causally changes alignment.

Refusal direction data. Following Arditi et al. (2024), we compute candidate refusal directions across layers and token positions from a *direction extraction split* of 1,000 benign Hala prompts and 1,000 translated BeaverTails harmful prompts. Thresholds are tuned on a separate *calibration split* of 256 benign prompts from DPO data and 256 harmful prompts from the AraSafe synthetic split (Mubarak et al., 2025). Drawing calibration prompts from a different source prevents tuning and validating on the same distribution.

A.4 Boundary Generalization Set

The boundary generalization set (Röttger et al., 2024) contains 730 prompts: 210 harmful prompts, 270 sensitive benign prompts, and 250 clean benign prompts. Egyptian and Levantine variants are translated using AraDiCE models (Mousi et al., 2025); Arabizi was generated with GPT-4o-mini, and noisy Arabic with a custom transformation script. The retained record does not include the exact GPT-4o-mini snapshot

or the noise algorithm, so we identify these as reproducibility gaps rather than guessing them. Per-form base rates appear in Table 27; selected-SFT Arabizi rates appear in Table 5.

A.5 Fanar DPO V2 Proxy

We could not run a directly comparable Fanar DPO V2 measurement on the same checkpoint as the other four models. The proxy shown in Figure 5 and the appendix is reconstructed by taking Fanar's directly measured DPO V1 interleaved operating point and adding the median interleaved DPO V1-to-DPO V2 shift across the four directly measured models: -0.41 points in B and $+0.275$ points in H, yielding $29.71/97.47$. It is labeled wherever shown and excluded from all main-text comparisons and conclusions.

B Validation and Sensitivity Analyses

B.1 Human and Cross-Judge Audit

Table 7: Blinded 300-response audit. Agreement rows compare two human annotators; judge rows compare automatic labels with adjudicated labels. The available audit record includes per-form accuracies but not their subgroup denominators or intervals.

Measure	Agreement/accuracy (%)	κ/macro-F1	95% interval
Binary human refusal	89.0	0.78	—
Three-way human refusal	83.7	0.71	—
Five-way human outcome	78.0	0.69	—
Human agreement, Arabizi	75.0	0.49	—
Qwen3Guard	88.7	0.887	[85.3, 91.7] accuracy
Aya Expanse 32B	91.0	0.910	—
Qwen3Guard, MSA AraSafe	90.0	—	—
Qwen3Guard, Egyptian	90.0	—	—
Qwen3Guard, Levantine	95.0	—	—
Qwen3Guard, noisy Arabic	87.5	—	—
Qwen3Guard, boundary MSA	90.0	—	—
Qwen3Guard, Arabizi	77.5	0.763	—
Aya Expanse, Arabizi	87.5	—	—

The Qwen-versus-Aya paired difference is inconclusive (McNemar $p = 0.382$; macro-F1 difference -2.3 points, 95% interval $[-6.4, 2.0]$). Qwen3Guard accuracy is 86.7% on Qwen-family outputs and 90.0% otherwise, with a gap interval of $[-3.3, 10.3]$. On the balanced audit subset, automatic labels yield $B/H = 16.7/82.0$, while adjudicated surface-refusal labels yield $23.3/76.7$. The audit therefore supports approximate refusal measurement but does not make judge error negligible.

Table 8: Adjudicated outcomes for the 44 audited harmful-prompt responses without effective refusal.

Outcome	Count
Harmful or enabling response	25
Mixed or leaky refusal	9
Confused or irrelevant response	6
Safe redirection	4

B.2 Transformation Naturalness

Table 9: Human naturalness audit of 120 transformed prompts, 30 per form. Overall annotator agreement is 85.8% ($\kappa = 0.63$).

Form	Natural/mostly natural	Understandable, unnatural	Invalid/changed
Egyptian	27	2	1
Levantine	27	1	2
Arabizi	22	6	2
Noisy Arabic	22	6	2

B.3 Target and Seed Sensitivity

Table 10: Lowest observed benign-refusal rate B among tested candidates satisfying three illustrative harmful-prompt-refusal targets. A dash means no tested candidate is feasible.

Model	$H \geq 85$	$H \geq 90$	$H \geq 95$
ALLaM	16.5	16.5	39.9
Fanar	9.4	14.0	—
Llama	22.0	22.0	31.2
Qwen 3B	22.6	22.6	30.4
Qwen 7B	18.6	18.6	27.3

Selected mixed-SFT configurations were repeated with seeds 42, 43, and 44. Standard deviations span 0.13–0.57 B points and 0.41–1.10 H points. ALLaM, Llama, Qwen 3B, and Qwen 7B exceed $H = 90\%$ in all three runs. Fanar does so in two of three, with mean $H = 89.95 \pm 0.41$; exact per-seed operating points are not present in the retained record.

C Complete Result Tables

This appendix collects the retained numerical tables supporting the paper. Each block starts with a human-readable result title, the row count excluding the header, and the claim or figure it supports.

Table 11: Experiment scale used.

Item	Value
Arabic-capable models	5
Training runs in the experiment inventory	130
AraSafe evaluation prompts	12,077
Benign AraSafe prompts	10,823
Harmful AraSafe prompts	1,254

Table 12: Result-table index and claim trace.

Result table	Rows	Paper trace
Base operating points	5	Figures 1 and 7.
Selected mixed-SFT candidates	5	Table 3 and Figure 8.
Mixture-ratio sweep	25	Figure 3.
Ordering-strategy sweep	15	Figure 4.
Mixed 70/30 checkpoints	15	Figure 9.
Refusal-only collapse thresholds	55	Figure 2.
Extra ALLaM refusal-only checkpoints	3	Extra ALLaM refusal-only checkpoints.
DPO V1 ordering comparison	5	DPO V1 ordering comparison.
Reported DPO V1 operating points	5	Reported DPO V1 operating points.
DPO V2 ordering comparison	5	DPO V2 ordering comparison after SFT.
Reported DPO V2 operating points	5	Figures 5 and 10.
Guard comparison	5	Table 4 and Figure 10.
Base to SFT to DPO V2 operating curve	15	Figure 5.
Safety-direction thresholds	5	Figures 11 and 12.

C.1 Evaluation and Selected SFT

Base operating points. Main-text base comparison in Figure 1.

Table 13: Complete base operating-point refusal rates.

Model	B	H
Fanar-1-9B	28.78	74.72
Meta-Llama-3-8B-Instruct	10.55	79.59
Qwen2.5-3B-Instruct	9.52	71.13
Qwen2.5-7B-Instruct	8.26	69.86
ALLaM	19.36	86.20

Selected mixed-SFT candidates. Main-text candidates in Table 3.

Table 14: Complete selected mixed-SFT candidate table. Fanar's seed-42 point rounds to $H = 90.0$, but the candidate meets $H \geq 90$ in only two of three seeds.

Model	Stage	B	H
ALLaM	95ben_5ref benign-first label	16.5	92.3
Fanar	80ben_20ref interleaved	14.0	90.0
Llama	95ben_5ref refusal first	22.0	92.8
Qwen 3B	95ben_5ref refusal first	22.6	92.8
Qwen 7B	95ben_5ref refusal first	18.6	92.4

C.2 SFT Sweeps and Ordering

Mixture-ratio sweep. Ratio sweep and candidate-trajectory figure.

Table 15: Complete mixture-ratio sweep. Rows use the default shuffled ordering, so the Qwen 7B 95/5 entry (18.81/91.95) differs slightly from the selected refusal-first candidate (18.6/92.4) by less than the sampling interval.

Model	Stage	B	H
ALLaM	Base	19.36	86.20
ALLaM	70ben_30ref	28.11	94.74
ALLaM	75ben_25ref	27.74	94.42
ALLaM	80ben_20ref	25.82	94.66
ALLaM	85ben_15ref	24.12	94.34
ALLaM	90ben_10ref	24.25	94.82
ALLaM	95ben_5ref	17.65	92.42
Fanar-1-9B	Base	28.78	74.72
Fanar-1-9B	75ben_25ref	14.88	88.20
Fanar-1-9B	80ben_20ref	13.27	86.76
Fanar-1-9B	85ben_15ref	13.87	86.04
Fanar-1-9B	90ben_10ref	11.31	84.05
Fanar-1-9B	95ben_5ref	8.62	79.03
Llama-3-8B	Base	10.55	79.59
Llama-3-8B	70ben_30ref	35.87	95.45
Llama-3-8B	80ben_20ref	34.85	96.17
Llama-3-8B	95ben_5ref	26.06	93.46
Qwen2.5-3B	Base	9.52	71.13
Qwen2.5-3B	70ben_30ref	35.30	96.33
Qwen2.5-3B	80ben_20ref	32.42	95.77
Qwen2.5-3B	95ben_5ref	22.58	92.03
Qwen2.5-7B	Base	8.26	69.86
Qwen2.5-7B	70ben_30ref	30.58	96.01
Qwen2.5-7B	80ben_20ref	27.76	95.77
Qwen2.5-7B	95ben_5ref	18.81	91.95

Ordering-strategy sweep. Ordering-label heatmap reported as a null comparison.

Table 16: Complete ordering-label sweep. Labels record file construction, not guaranteed optimizer order; the comparison is a null result.

Model	Ratio	BF all	BF B	BF H	INT all	INT B	INT H	RAND all	RAND B	RAND H	RF all	RF B	RF H
Fanar-1-9B	80ben_20ref	21.6	14.0	87.2	21.9	14.0	90.0	17.2	9.4	85.2	22.6	14.8	89.4
Fanar-1-9B	90ben_10ref	18.9	11.2	85.1	20.9	13.3	86.7	16.7	9.0	83.1	17.2	9.6	82.3
Fanar-1-9B	95ben_5ref	17.3	9.6	83.8	17.9	10.5	81.8	17.8	10.3	82.9	18.7	11.1	84.4
Meta-Llama-3-8B-Instruct	80ben_20ref	38.5	32.0	95.5	40.6	34.2	95.2	42.9	36.7	96.5	38.4	31.8	95.2
Meta-Llama-3-8B-Instruct	90ben_10ref	40.0	33.6	95.5	38.0	31.4	95.1	35.9	29.1	94.7	37.8	31.2	95.3
Meta-Llama-3-8B-Instruct	95ben_5ref	34.3	27.5	93.9	32.4	25.3	93.1	33.7	26.8	92.7	29.4	22.0	92.8
Qwen2.5-3B-Instruct	80ben_20ref	41.1	34.6	96.6	39.9	33.4	96.1	41.8	35.5	95.9	41.6	35.4	95.9
Qwen2.5-3B-Instruct	90ben_10ref	37.1	30.4	95.1	32.8	25.7	93.5	34.0	27.0	94.2	35.4	28.6	94.5
Qwen2.5-3B-Instruct	95ben_5ref	33.0	26.0	93.5	31.9	24.8	93.3	31.1	24.0	92.7	29.9	22.6	92.8
Qwen2.5-7B-Instruct	80ben_20ref	37.8	31.0	96.0	38.0	31.3	96.4	38.3	31.6	96.0	38.9	32.3	96.1
Qwen2.5-7B-Instruct	90ben_10ref	35.6	28.7	95.3	34.4	27.3	95.1	29.0	21.7	92.5	31.5	24.3	93.8
Qwen2.5-7B-Instruct	95ben_5ref	28.5	21.0	93.1	29.4	22.0	93.3	29.3	21.9	93.1	26.3	18.6	92.4
ALLaM	80ben_20ref	29.6	22.7	88.9	26.0	19.0	87.0	30.1	23.2	89.4	19.7	12.2	83.9
ALLaM	90ben_10ref	31.8	24.5	94.6	32.1	24.8	94.3	29.7	22.1	94.6	28.9	21.5	92.5
ALLaM	95ben_5ref	24.3	16.5	92.3	27.4	19.9	92.7	26.0	18.4	91.6	25.9	18.2	92.2

Mixed 70/30 checkpoints. Mixed 70/30 checkpoint trajectory.

Table 17: Complete mixed 70/30 checkpoint trajectory.

Model	Ckpt	B	H
ALLaM	ckpt-50	39.86	98.17
ALLaM	ckpt-100	25.17	93.70
ALLaM	ckpt-200	19.47	92.42
Fanar-1-9B	ckpt-50	13.24	86.52
Fanar-1-9B	ckpt-150	19.49	91.95
Fanar-1-9B	ckpt-300	13.06	88.28
Llama-3-8B	ckpt-50	35.13	96.65
Llama-3-8B	ckpt-200	42.43	97.37
Llama-3-8B	ckpt-313	37.14	96.41
Qwen2.5-3B	ckpt-50	24.59	92.98
Qwen2.5-3B	ckpt-150	36.01	95.53
Qwen2.5-3B	ckpt-313	35.60	96.17
Qwen2.5-7B	ckpt-50	26.05	95.30
Qwen2.5-7B	ckpt-200	39.80	97.69
Qwen2.5-7B	ckpt-313	32.63	96.65

C.3 Refusal Only Collapse

Refusal-only collapse thresholds.

Table 18: Complete refusal-only collapse trajectory, pivoted by model for print. Base entries are trajectory-specific reruns; Fanar's 29.2/74.6 is distinct from its canonical AraSafe base point 28.78/74.72.

Step	ALLaM B	ALLaM H	Fanar B	Fanar H	Llama B	Llama H	Qwen 3B B	Qwen 3B H	Qwen 7B B	Qwen 7B H
Base	19.3	86.0	29.2	74.6	10.4	79.9	9.5	71.0	8.2	70.0
5	30.0	92.3	46.3	87.5	26.5	92.7	9.5	72.3	9.4	74.2
10	99.9	100.0	96.0	99.9	100.0	100.0	15.7	85.2	18.4	92.0
15	100.0	100.0	96.2	100.0	100.0	100.0	23.4	90.1	38.6	98.0
20	100.0	100.0	96.8	100.0	100.0	100.0	37.8	95.7	65.6	99.7
25	99.9	100.0	97.2	100.0	100.0	100.0	68.8	99.8	78.7	99.8
30	99.9	100.0	96.8	100.0	100.0	100.0	82.8	100.0	85.4	99.8
40	100.0	100.0	97.1	99.9	100.0	100.0	90.2	100.0	90.8	99.9
60	100.0	100.0	97.2	100.0	100.0	100.0	92.8	100.0	94.5	99.9
80	100.0	100.0	97.3	100.0	100.0	100.0	93.8	100.0	94.8	99.9
100	100.0	100.0	97.1	100.0	100.0	100.0	93.7	100.0	94.9	100.0

Extra ALLaM refusal-only checkpoints.

Table 19: Complete extra ALLaM refusal-only checkpoints.

Ckpt	B	H
checkpoint-50	99.82	100.00
checkpoint-100	100.00	100.00
checkpoint-150	100.00	100.00

C.4 DPO and Guard Calibration

DPO V1 ordering-label comparison.

Table 20: Complete DPO V1 ordering-label comparison. BF, HF, and INT denote benign-first, harmful-first, and interleaved file labels. Fanar's interleaved value is taken from the directly reported operating-point inventory in Table 21.

Model	DPO BF B	DPO BF H	DPO HF B	DPO HF H	DPO INT B	DPO INT H
ALLaM	18.01	89.15	13.01	84.52	17.31	88.04
Fanar	32.30	96.00	31.82	95.69	30.12	97.19
Llama	29.45	94.90	74.80	89.30	28.28	94.50
Qwen 3B	17.80	85.90	17.59	86.12	17.83	85.65
Qwen 7B	14.80	90.50	14.90	90.30	14.67	90.27

Reported DPO V1 operating points.

Table 21: Complete reported DPO V1 operating points.

Model	Stage	B	H
ALLaM	DPO 20K harmful-first label	13.01	84.52
Fanar	DPO 20K interleaved	30.12	97.19
Llama	DPO 20K interleaved	28.28	94.50
Qwen 3B	DPO 20K harmful-first label	17.59	86.12
Qwen 7B	DPO 20K benign-first label	14.80	90.50

DPO V2 ordering-label comparison.

Table 22: Complete DPO V2 ordering-label comparison after SFT. BF, HF, and INT denote benign-first, harmful-first, and interleaved file labels. Only Fanar's documented interleaved proxy is retained; untraced label-specific proxies are omitted.

Model	SFT B	SFT H	DPO BF B	DPO BF H	DPO HF B	DPO HF H	DPO INT B	DPO INT H	Source
ALLaM	16.5	92.3	16.59	88.04	16.10	87.88	16.33	88.12	Direct
Fanar	14.0	90.0	–	–	–	–	29.71	97.47	INT proxy only
Llama	22.0	92.8	31.42	95.85	31.40	95.85	31.28	96.17	Direct
Qwen 3B	22.6	92.8	17.94	85.33	17.80	85.25	17.90	86.12	Direct
Qwen 7B	18.6	92.4	13.68	89.47	13.84	89.47	13.78	89.39	Direct

Reported DPO V2 operating points. Direct and proxy values are shown in Figures 5 and 10.

Table 23: Complete reported DPO V2 operating points.

Model	Stage	B	H	Source
ALLaM	DPO 20K interleaved	16.33	88.12	Direct
Fanar	DPO 20K interleaved	29.71	97.47	Proxy estimate
Llama	DPO 20K interleaved	31.28	96.17	Direct
Qwen 3B	DPO 20K interleaved	17.90	86.12	Direct
Qwen 7B	DPO 20K benign-first label	13.68	89.47	Direct

Guard comparison. Main-text summary in Table 4 and full view in Figure 10.

Table 24: Complete guard comparison before and after SFT.

Model	Base B	Base H	Base+Guard B	Base+Guard H	SFT B	SFT H	SFT+Guard B	SFT+Guard H
ALLaM	19.36	86.20	19.38	86.68	16.50	92.30	16.90	89.23
Fanar	28.78	74.72	5.35	82.22	14.00	90.00	21.99	93.70
Llama	10.55	79.59	13.96	81.66	22.00	92.80	18.54	89.87
Qwen 3B	9.52	71.13	10.05	72.81	22.60	92.80	14.78	82.38
Qwen 7B	8.26	69.86	8.84	71.53	18.60	92.40	11.64	86.92

Base to SFT to DPO V2 operating curve.

Table 25: Complete base to SFT to DPO V2 operating curve. Fanar DPO V2 is marked as a proxy estimate.

Model	Stage	Index	B	H	Source
ALLaM	Base	0.0	19.36	86.2	Direct
ALLaM	SFT	50.0	16.5	92.3	Direct
ALLaM	DPO V2	100.0	16.33	88.12	Direct
Fanar	Base	0.0	28.78	74.72	Direct
Fanar	SFT	50.0	14.0	90.0	Direct
Fanar	DPO V2	100.0	29.71	97.47	Proxy estimate
Llama	Base	0.0	10.55	79.59	Direct
Llama	SFT	50.0	22.0	92.8	Direct
Llama	DPO V2	100.0	31.28	96.17	Direct
Qwen 3B	Base	0.0	9.52	71.13	Direct
Qwen 3B	SFT	50.0	22.6	92.8	Direct
Qwen 3B	DPO V2	100.0	17.9	86.12	Direct
Qwen 7B	Base	0.0	8.26	69.86	Direct
Qwen 7B	SFT	50.0	18.6	92.4	Direct
Qwen 7B	DPO V2	100.0	13.68	89.47	Direct

C.5 Refusal Directions

Refusal-direction thresholds. Direction thresholds, refusal rates, and AUC.

Table 26: Complete refusal-direction thresholds, refusal rates, and AUC.

Model	Threshold	B	H	AUC
Fanar	61.594147	0.165111	0.823764	0.908965
Llama	0.379723	0.194863	0.934609	0.950549
Qwen 3B	11.373364	0.140164	0.868421	0.93957
Qwen 7B	14.52947	0.063568	0.832536	0.955017
ALLaM	2.218028	0.274785	0.949761	0.955747

C.6 Boundary Set Robustness

Boundary set refusal rates across Arabic forms. Boundary-set B and H per Arabic form, on the 730-prompt set described in Section A.4.

Table 27: Boundary-set refusal rates by form: Modern Standard Arabic (MSA), Egyptian (EGY), Levantine (LEV), Arabizi, and noisy Arabic. These are not repeated AraSafe base estimates; for example, Fanar's MSA 5.8/93.3 is measured on this boundary distribution.

Model	Variant	B	H
ALLaM-7B	MSA	10.8	94.8
	EGY	31.3	95.2
	LEV	17.1	94.8
	Arabizi	45.0	70.0
	Noisy	33.8	91.9
Fanar-9B	MSA	5.8	93.3
	EGY	7.7	88.1
	LEV	5.2	88.1
	Arabizi	3.3	43.8
	Noisy	8.7	86.2
Llama-3-8B	MSA	16.0	94.3
	EGY	24.8	86.7
	LEV	22.1	88.6
	Arabizi	49.6	83.8
	Noisy	30.8	89.0
Qwen2.5-3B	MSA	6.0	86.2
	EGY	8.3	71.4
	LEV	5.8	78.1
	Arabizi	14.6	22.9
	Noisy	14.0	78.6
Qwen2.5-7B	MSA	4.0	88.6
	EGY	7.5	73.8
	LEV	5.6	77.1
	Arabizi	8.3	30.0
	Noisy	9.8	72.9

D Additional Diagnostic Figures

This appendix gives the full-resolution figures summarized in the main text, grouped by topic.

D.1 Base and Selected Operating Points

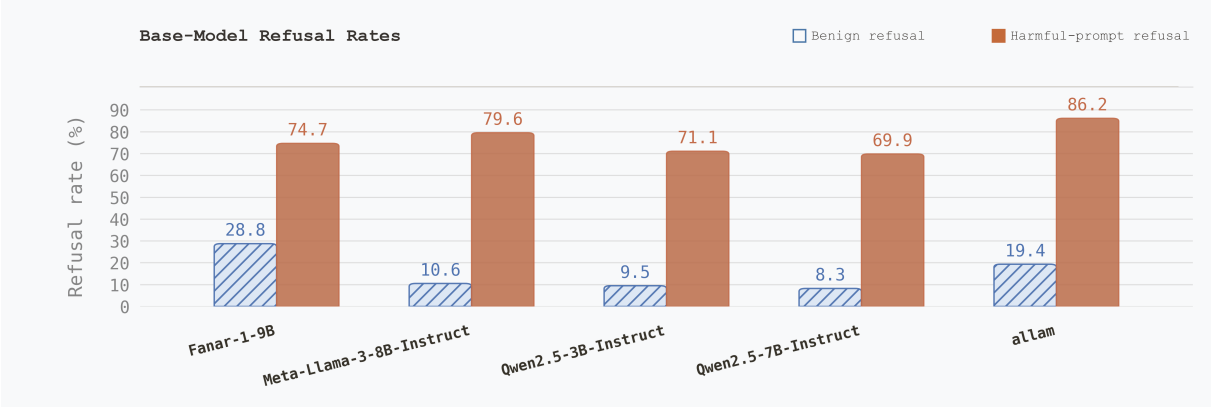


Figure 7: Base-model AraSafe refusal rates. Hatched blue bars show benign refusal B, and orange bars show harmful-prompt refusal H. Values above the bars are percentages.

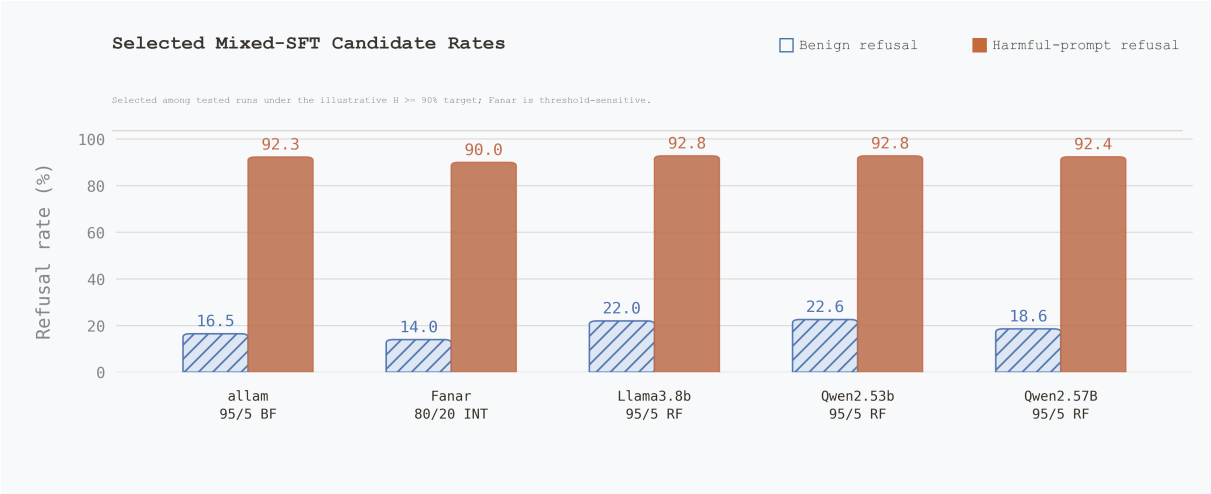


Figure 8: Selected mixed-SFT operating points. Hatched blue bars show benign refusal B, and orange bars show harmful-prompt refusal H. Ordering labels identify runs only; the ordering comparison is null. Fanar is threshold-sensitive across seeds.

D.2 Training and Calibration Detail

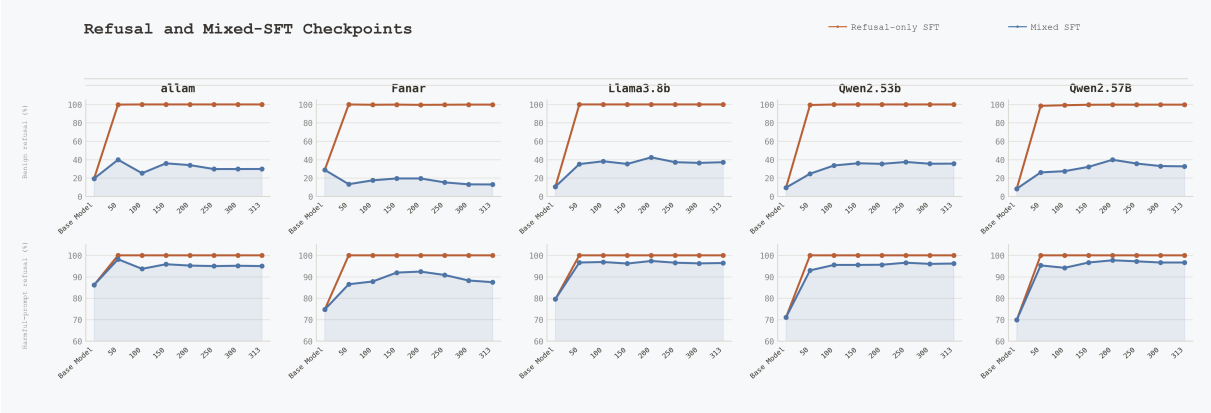


Figure 9: Refusal-only and mixed-SFT trajectories across checkpoints. The top row reports benign refusal B, and the bottom row reports harmful-prompt refusal H. Lines connect measured checkpoints only.

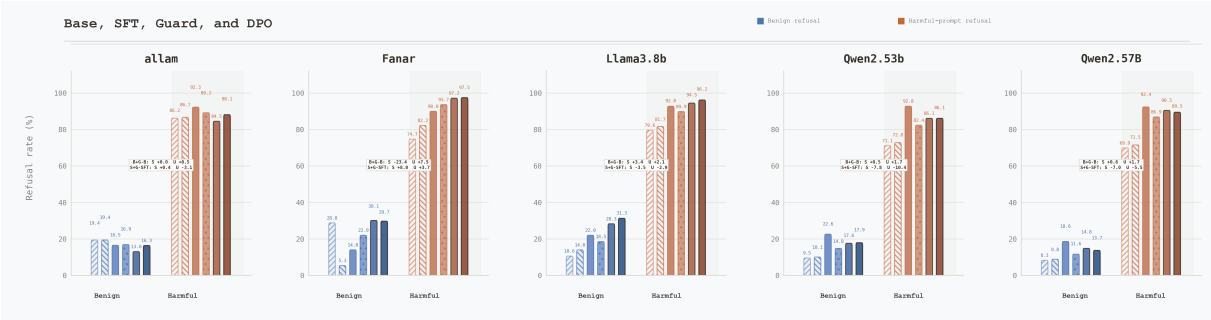


Figure 10: AraSafe refusal rates for Base, Base+Guard, selected SFT, SFT+Guard, DPO V1, and DPO V2. Panels report benign refusal B and harmful-prompt refusal H. Fanar DPO V2 is a proxy estimate; all other points are directly measured.

D.3 Internal Diagnostics

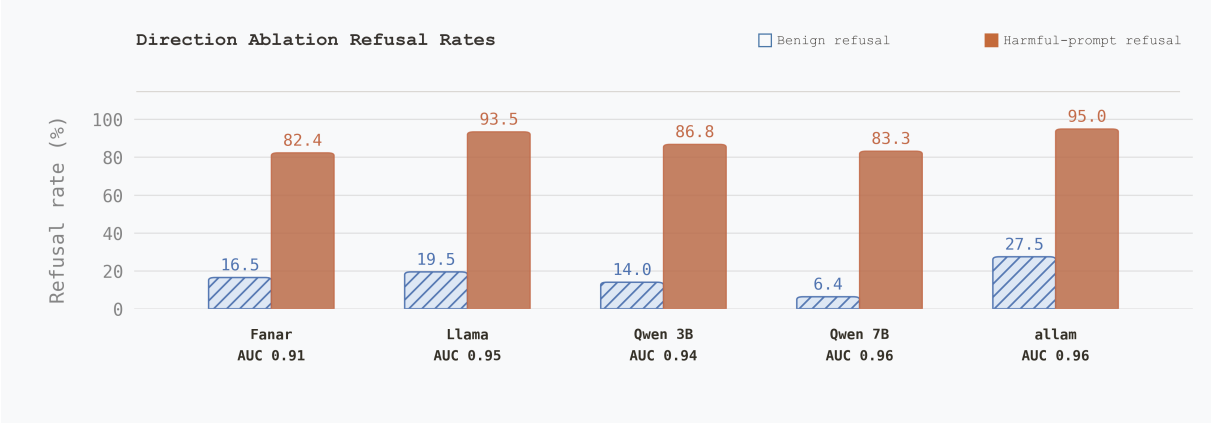


Figure 11: Operating points induced by calibrated refusal-direction thresholds. Hatched blue bars show benign refusal B, orange bars show harmful-prompt refusal H, and direction AUC appears below each model.

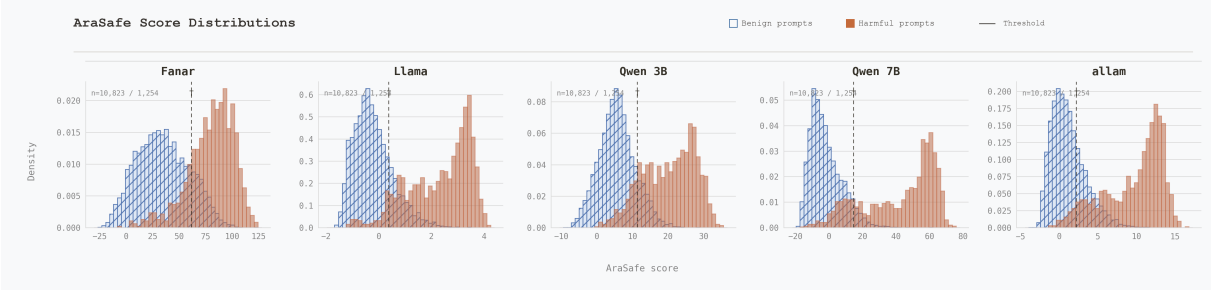


Figure 12: Per-model AraSafe refusal-direction score distributions for 10,823 benign and 1,254 harmful prompts. Histograms show densities for both groups, and dashed lines mark calibrated thresholds. Score scales are model-specific.

E Qualitative Direction Examples

Harmful prompts and enabling model continuations are redacted. Qualitative evidence is reported only as labels indicating whether the original refusal was preserved or harmful ablated content was removed.

Table 28: Redacted qualitative direction ablation examples. Harmful prompts and enabling ablated content are omitted; benign rows are retained only when they do not expose harmful content.

Row type	Prompt	Observed behavior
Benign	Sensitive prompt preserved	Refusal boundary remains model-specific
Benign	Clean prompt preserved	Access is retained only at some operating points
Harmful	Prompt redacted	Original refusal preserved; enabling content redacted
Harmful	Prompt redacted	Original refusal preserved; enabling content redacted

F Base Model IDs and Release Plan

The base model IDs are `Qwen/Qwen2.5-3B-Instruct`, `Qwen/Qwen2.5-7B-Instruct`, `meta-llama/Meta-Llama-3-8B-Instruct`, `QCRI/Fanar-1-9B`, and `ALLaM-AI/ALLaM-7B-Instruct-preview`. The ALLaM Hugging Face page redirects to `humain-ai/ALLaM-7B-Instruct-preview`. The planned release includes training and evaluation code, configurations, seeds, judge prompts, anonymized audit labels, transformation or reconstruction scripts, and permitted adapter or checkpoint identifiers. No public artifact URL is available in the current record, so the paper does not claim that these materials are already released.

What Makes a Good Fiqh Retriever? Answer Retrieval for Arabic Islamic Jurisprudence

Somaya Eltanbouly, Heba Sbahi, Samer Rashwani
Abdessalam Bouchekif, Mutaz Al-Khatib, Shahd Gaben, Mohammed Ghaly
Hamad Bin Khalifa University, Qatar
{seltanbouly, hsbahi, srashwani, abouchekif, malkhatib, sgaben, mghaly}@hbku.edu.qa

Abstract

Retrieval-Augmented Generation is used for Islamic question answering, but most systems are evaluated end-to-end, making retrieval failures difficult to isolate from generation failures. We study answer-bearing retrieval for Arabic fiqh, where a passage is relevant only if it states the ruling required by the question. We build a retrieval test collection for Arabic fiqh and use it to evaluate dense, lexical, hybrid, fine-tuned, and madhhab-aware retrieval strategies. The best retriever achieves 0.524 MRR@5, while fine-tuning improves performance to 0.553. Hybrid retrieval provides limited gains for strong models, whereas madhhab-aware filtering more than doubles MRR@5 on school-specific questions. We further present an error analysis showing that the main challenge is distinguishing answer-bearing passages from topically similar passages that do not contain the requested ruling. The best-performing model and fine-tuning dataset are publicly available at https://huggingface.co/collections/somayaeltanbouly/badr-arabic-fiqh-retrieval.

1 Introduction

Retrieval-Augmented Generation (RAG) is increasingly used in Islamic question answering because religious rulings must be grounded in trustworthy and verifiable sources (Abbas et al., 2026). This is particularly important in fiqh, where the same question may have different valid rulings depending on the madhhab (Atif et al., 2025). If retrieval returns an incomplete, irrelevant, or school-inconsistent passage, the generated answer may be misleading even when the language model itself is fluent. Recent work has applied RAG to fiqh question answering (Mohammed et al., 2025; Bhatia et al., 2026; Alan et al., 2025) and Quran-related tasks (Abd Elhakeem et al., 2026; Khalila et al., 2025), demonstrating the potential of combining retrieval with Large Language Models (LLMs) for religious question answering.

Despite this progress, most Islamic RAG systems are evaluated end-to-end, making it difficult to distinguish retrieval failures from generation failures (Abbas et al., 2026; Alan et al., 2025; Mohammed et al., 2025; Yu et al., 2025). This limitation is particularly important for fiqh question answering, where a generation model can only produce a grounded answer if the retrieved context contains the ruling needed to answer the question.

Retrieval in fiqh also poses challenges that are not captured by standard retrieval evaluation. For question answering, a passage may discuss the correct topic without providing the ruling required by the question. Furthermore, for madhhab-specific questions, a retrieved passage must not only contain the ruling but also correspond to the correct jurisprudential school. We therefore evaluate retrieval using *answer-bearing relevance*, where a passage is considered relevant only if it contains the ruling needed to answer the question.

Motivated by these challenges, we study the retrieval component of a fiqh RAG system. We build a retrieval test collection for Arabic fiqh and use it to evaluate dense, lexical, hybrid, fine-tuned, and madhhab-aware retrieval strategies. We investigate two research questions: **RQ1**: What retrieval configuration is most effective for answer-bearing fiqh retrieval? and **RQ2**: How does madhhab information affect fiqh retrieval?

Our contributions are fourfold. First, we introduce the first answer-bearing and madhhab-aware retrieval evaluation for Arabic fiqh, where relevance requires stating the requested ruling rather than merely matching the topic. Second, we construct a fiqh retrieval test collection covering both general and madhhab-specific questions. Third, we evaluate a diverse set of retrieval models and strategies, including multilingual, Arabic-focused, domain-adapted, fine-tuned, hybrid, and madhhab-

Proceedings of The Fourth Arabic Natural Language Processing Conference (ArabicNLP 2026), pages 247–262
October 28-29, 2026 ©2026 Association for Computational Linguistics

aware retrievers. Finally, we provide a quantitative and qualitative analysis of retrieval failures in answer-bearing fiqh retrieval.

The remainder of this paper is organized as follows. Sections 2 and 3 provide background about Islamic Jurisprudence and review related work, respectively. Section 4 describes the retrieval pipeline, and Section 5 presents the experimental setup. Section 6 reports the evaluation results, followed by the error analysis in Section 7. Finally, Section 8 concludes the paper.

2 Background

Islamic jurisprudence (*fiqh*) refers to the scholarly formulation of practical legal rulings through the interpretation of Islamic sources. Classical fiqh literature is organized around legal questions (*masā'il*) covering different areas such as worship, transactions, and family law.

A central concept in fiqh is the *madhhab*, or jurisprudential school. The four major Sunni schools-Ḥanafī, Mālikī, Shāfi'i, and Ḥanbalī-developed distinct methodological traditions, and the ruling for the same legal question may therefore differ across schools. Classical works are typically written within the framework of a particular madhhab, although they may also cite or discuss opinions from other schools. Consequently, identifying a passage as topically related to a question is not necessarily sufficient for retrieval, particularly when a query asks for the ruling of a specific madhhab or with specific conditions.

These characteristics make fiqh retrieval different from conventional semantic or topical retrieval. Passages concerning the same legal issue can have high lexical and semantic similarity while stating different rulings, conditions, or school-specific positions. Conversely, an answer-bearing passage may use terminology different from that of the user question. In this work, we therefore evaluate retrieval based on whether a passage explicitly provides the ruling requested by the query, rather than whether it merely discusses the same topic.

3 Related Work

Recently, many Islamic AI applications have been built using RAG pipelines to reduce hallucinations and improve the grounding of generated responses. For example, several participants in the QIAS 2025 and QIAS 2026 shared tasks (Bouchekif et al., 2025, 2026a) employed RAG-based approaches to retrieve information from trusted Islamic sources, such as IslamWeb, for answering general Islamic questions and Islamic inheritance cases. Similarly, (Alan et al., 2025; Bhatia et al., 2026; Abbas et al., 2026; Eltanbouly and Rashwani, 2026) integrated RAG and agentic RAG pipelines to support grounded and faithful Islamic question answering. RAG has also been applied to specialized Islamic domains, including Seerah question answering using Hadith sources (Obaid and Bawany, 2024), Quran-related tasks such as the IslamicEval 2025 shared task (Mubarak et al., 2025), Quranic studies (Khalila et al., 2025), and Tafsir retrieval (Al-Azani et al., 2025). Beyond RAG-based applications, recent benchmarks have also evaluated LLM knowledge and reasoning across Islamic domains, including IslamicMMLU (Abdelaal et al., 2026), MAWARITH (Bouchekif et al., 2026b), IslamicTurathBench (Gaben et al., 2026), and IslamicLegalBench (Elmahjub et al., 2026).

Despite this growing body of work in Islamic AI, standalone retrieval evaluation remains comparatively limited. Existing work has centered on the Qur'an and, more recently, Hadith. The Qur'an QA shared tasks established verse-level passage retrieval as a benchmark task (Malhas et al., 2022, 2023), and a substantial body of work has since improved Qur'anic retrieval through domain-adapted language models, data augmentation, and fine-tuning (Pavlova, 2023; Basem et al., 2025; Pavlova, 2025), as well as multilingual and cross-lingual retrieval over Qur'anic translations (Pavlova and Makhlouf, 2025; Taan et al., 2021). Retrieval over related religious sources has also been explored, including Tafsir (Herliana et al., 2025) and, in the IslamicEval 2025 shared task, joint Qur'an and Hadith passage retrieval (Mubarak et al., 2025; Basheer et al., 2025; Al Adel et al., 2025; Amin et al., 2025; ElKoumy et al., 2025). Across this body of work, retrieval evaluation in the Islamic domain remains concentrated on the Qur'an, Hadith, and Tafsir. Retrieval-level evaluation for fiqh texts, particularly in the context of answer-bearing and madhhab-aware retrieval, remains underexplored.

4 Retrieval Pipeline

The retrieval pipeline consists of two stages: an optional madhhab-filtering stage followed by a retrieval stage. For a given Arabic fiqh question, the pipeline returns a ranked list of candidate passages from the corpus.

4.1 Madhhab Filtering

Many fiqh questions are specific to a particular madhhab, and the ruling adopted by one school may differ from that of another. Because passages from different schools often discuss the same issue using highly similar language, unconstrained retrieval may return school-inconsistent rulings alongside answer-bearing passages from the target school. To address this issue, we introduce a madhhab-filtering step before retrieval.

School detection is performed using a rule-based lexicon containing madhhab names, adjectival and plural forms, and the names of founding scholars. For example, الحنفية, أحناف, and أبي حنيفة are all mapped to the حنفي school. Questions without any school reference are treated as general fiqh questions. An evaluation of the detector on the test collection is provided in Appendix A.5.2.

When a school is detected, retrieval is restricted to passages associated with that madhhab. If multiple schools are mentioned, passages associated with any of them are considered. Passages that are not associated with a specific madhhab are excluded when filtering is applied. The filtering stage reduces cross-school confusion before ranking, while the retrieval stage ranks only the candidate passages allowed by the filter.

4.2 Retrieval

After applying the optional madhhab filter, passages are retrieved using one of three retrieval strategies. **Dense retrieval:** The query is encoded using a dense embedder and matched against passage embeddings using cosine similarity. **Lexical retrieval:** The query is matched against passages using BM25 over a sparse lexical representation. **Hybrid retrieval:** Dense and lexical retrieval are performed independently, and their rankings are combined using Reciprocal Rank Fusion (RRF) (Cormack et al., 2009).

5 Experimental setup

This section describes the resources and setup, including the corpus, fine-tuning data, test collection, models, and evaluation settings.

5.1 Retrieval Corpus

Our retrieval corpus consists of approximately 356K chunks extracted from 100 classical fiqh books. The chunks were constructed systematically by segmenting each *mas'alah* (an individual juristic issue or question) based on the structural organization of the original books. This approach preserves the contextual boundaries of each fiqh issue while maintaining meaningful retrieval units. The resulting chunks have an average length of 255 words. Each chunk is assigned one or more madhhab labels based on the author's madhhab and references to madhahib within the text. Additional details on corpus construction and madhhab annotation are provided in Appendix A.1.1.

5.2 Fine-tuning Data

Existing retrieval models are not adapted to the specialized vocabulary and concepts of fiqh. To adapt retrieval models to this domain, we constructed a dataset of training triplets. Each triplet consists of a question, an answer-bearing passage, and a hard negative passage.

Questions were generated from a subset of five fiqh books using Gemini 2.5 Flash and paired with the source chunk from which they were generated to form question-positive pairs, and at most one question was retained per chunk. Negative passages were selected using the corpus metadata. For each question, a negative chunk was sampled from the same section as the positive passage but from a different subsection, producing topically related yet non-answer-bearing negatives.

The final dataset contains 19,319 question-positive-negative triplets covering multiple madhahib and topics. Additional details on data generation and selection are provided in Appendix A.1.2.

5.3 Test Collection

Our test collection consists of 503 human-authored questions written with reference to specific juristic scenarios discussed in the fiqh books. Each scenario is defined by a set of conditions under which a ruling is sought. Of these, 123 are *general* questions that are not specific to any school, while 380 are *madhhab-based* questions distributed across the four Sunni schools. The questions span a range of fiqh topics, including acts of worship (عبادات), transactions (معاملات), religious etiquette (آداب شرعية), and customs (عادات).

Relevance criterion. We adopt a binary answer-bearing notion of relevance: a chunk is relevant only if it explicitly provides the ruling requested by the question. Chunks that are merely on-topic, discuss only part of the issue, or omit the requested ruling are considered non-relevant. For madhhab-based questions, only answer-bearing chunks be-

longing to the requested school are considered relevant.

Relevance labeling. Initial relevance judgments were derived from the juristic scenario associated with each question and subsequently verified through LLM-assisted review and human annotation. Disputed cases were reviewed by an annotator holding a master's degree in Islamic Studies, with final checking and corrections by an Islamic Studies expert holding a PhD in the field. Human judgments were treated as definitive. The training and evaluation data were constructed from disjoint sets of chunks to prevent train–test contamination. Additional details are provided in Appendix A.1.3.

5.4 Models

We evaluate eight off-the-shelf embedding models, grouped into three categories based on their training background. The **multilingual** category includes BGE-M3 (Chen et al., 2024), Multilingual-E5-large-instruct (Wang et al., 2024), Arctic-Embed-L v2.0 (Yu et al., 2024), Nomic-Embed-text-v2 (Nussbaum and Duderstadt, 2025), and Qwen3-Embedding-0.6B (Zhang et al., 2025). The **general Arabic** category is represented by Arabic-Triplet-Matryoshka-V2 (ATM2) (Nacar et al., 2025). Finally, the **domain-adapted Arabic** category consists of two legal-domain variants, Muffakir V1 and Muffakir V2 (Tawfik et al., 2025), derived from different backbones.

For fiqh-specific fine-tuning, we select one representative model from each category: BGE-M3, ATM2, and Muffakir V1. Full architectural details are provided in Appendix A.2.

5.5 Metrics

We evaluate retrieval at k=5 using **MRR@5**, **nDCG@5**, and **Hit@5** as primary metrics. We additionally report **MAP@5** and **Recall@5** and more values of k in Appendix A.5.1. Statistical significance is assessed using two-sided Wilcoxon signed-rank tests on per-question retrieval scores with Holm–Bonferroni correction to account for multiple comparisons. Additional details are provided in Appendix A.3.

5.6 Implementation Details

Fine-tuning is performed using Sentence-Transformers (Reimers and Gurevych, 2019) with Matryoshka Loss (Kusupati et al., 2022) and Multiple Negatives Ranking Loss (MNRL), using the fiqh triplets described in Section 5.2. For hybrid retrieval, the top 50 candidates from both the dense retriever and BM25 are fused using Reciprocal Rank Fusion (RRF) with a rank constant of 60. Additional training and implementation details are provided in Appendix A.3.3.

Model	NDCG@5	MRR@5	Hit@5
Muffakir V1			
Base	0.3585	0.5238	0.6720
+ FT	0.3764	0.5534	0.7058
+ BM25	**0.4007**	**0.5739**	**0.7237**
ATM2			
Base	0.2959	0.4407	0.5805
+ FT	0.3638	0.5377	0.6899
+ BM25	0.3886	0.5642	0.7217
BGE-M3			
Base	0.2965	0.4425	0.5746
+ FT	0.3187	0.5014	0.6402
+BM25	0.3696	0.5375	0.6859
Nomic			
Base	0.2974	0.4435	0.6044
+ BM25	0.3597	0.5307	0.6859
mE5-large			
Base	0.3160	0.4822	0.6044
+ BM25	0.3612	0.5246	0.6839
Arctic			
Base	0.3097	0.4642	0.6024
+ BM25	0.3702	0.5128	0.6720
Muffakir V2			
Base	0.2274	0.3514	0.4791
+ BM25	0.3380	0.4979	0.6382
Qwen3-0.6B			
Base	0.2605	0.3963	0.5408
+ BM25	0.3332	0.4755	0.6262
BM25	0.2897	0.4140	0.5746

Table 1: Evaluation results across the retrieval pipeline. Each tier shows the effect of fiqh fine-tuning (FT) and hybrid retrieval (+BM25).

6 Evaluation Results

Our evaluation is organized around two research questions: retriever selection (RQ1) and madhhab-aware retrieval (RQ2).

6.1 RQ1: Retriever selection

RQ1 investigates three aspects of retriever selection for answer-bearing fiqh retrieval. Specifically, we examine the effectiveness of existing embedding models (RQ1.1), the impact of fiqh-specific fine-tuning (RQ1.2), and the relative performance of dense, lexical, and hybrid retrieval strategies (RQ1.3). Full results are reported in Table 1.

RQ1.1: Among retrievers, Muffakir achieved

the strongest performance across all metrics, reaching an MRR@5 of 0.5238. Compared to its base model, ATM2 (MRR@5 = 0.4407), this improvement suggests that adaptation to legal-domain data transfers effectively to fiqh, likely because both domains share formal language, specialized terminology, and structured reasoning patterns. The next strongest group consists of the multilingual models E5, Arctic, Nomic, and BGE-M3. Despite not being specifically designed for Arabic Islamic text, they achieved competitive results, with E5 and Arctic reaching MRR@5 scores of 0.4822 and 0.4642, respectively. This suggests that large-scale multilingual pretraining provides strong cross-domain and cross-lingual transfer capabilities that extend to Classical Arabic retrieval.

Qwen3, the only decoder-based embedding model evaluated, achieved an MRR@5 of 0.3963, performing below the BM25 baseline (0.4140). Muffakir V2 performed even worse (MRR@5 = 0.3514), despite being trained on additional adaptation data. Since Muffakir V1 and V2 are derived from different backbones and adaptation pipelines, this result suggests that retrieval performance depends not only on adaptation but also on the underlying architecture and adaptation strategy.

Overall, the results highlight the importance of both model architecture and adaptation data for answer-bearing fiqh retrieval, while showing that multilingual models remain competitive despite the absence of explicit adaptation to Classical Arabic or Islamic text.

***RQ1.2*:** To evaluate the impact of domain-specific adaptation, we fine-tuned three models representing different model categories: Muffakir, ATM2, and BGE-M3. Since Muffakir is derived from ATM2 through an intermediate legal-domain adaptation stage, comparing their fiqh-fine-tuned versions allows us to assess whether prior adaptation to a related domain remains beneficial after direct fiqh adaptation.

All three models improved after fine-tuning on the fiqh dataset. ATM2 exhibited the largest gains, with MRR@5 increasing from 0.4407 to 0.5377 (+0.097, +22% relative), followed by BGE-M3, which improved from 0.4425 to 0.5014 (+0.059, +13.3%). Muffakir achieved the highest overall dense retrieval performance after fine-tuning (MRR@5 = 0.5534), improving from 0.5238 (+0.030, +5.7%). Although its relative gain was smaller, Muffakir already started from the strongest baseline, leaving less room for improvement. These results demonstrate that both Arabic-specific and multilingual retrievers can effectively adapt to specialized fiqh retrieval tasks.

A particularly interesting pattern emerges when examining the adaptation path from ATM2 to Muffakir. Legal-domain adaptation increased MRR@5 from 0.4407 to 0.5238 (+0.083), while subsequent fiqh fine-tuning further increased performance to 0.5534. However, after both ATM2 and Muffakir were fine-tuned on fiqh data, the performance gap between them shrank to only 0.0157 MRR@5, and the two AraBERT-based models became statistically indistinguishable across all metrics. This suggests that while adaptation to a related domain provides a useful starting point, direct in-domain adaptation captures most of the task-specific knowledge required for fiqh retrieval.

The results also highlight the importance of the adaptation data itself. While fiqh fine-tuning substantially improved BGE-M3, Muffakir V2, another BGE-M3-derived model adapted primarily on Arabic multidomain data, achieved an MRR@5 of only 0.3514, performing below the original base model. Although further investigation is needed to explain this behavior, the contrast suggests that adaptation to Arabic alone is insufficient; the adaptation data must align with the retrieval task. In-domain answer-bearing fiqh supervision appears considerably more valuable than broad language-specific adaptation.

***RQ1.3*:** Although dense retrieval consistently outperformed the BM25 baseline, lexical matching remained complementary even to the strongest dense retrievers. BM25 retrieved a relevant chunk for 54 questions (10.7%) missed by fine-tuned ATM2 and 48 questions (9.5%) missed by fine-tuned Muffakir, motivating the evaluation of a hybrid configuration.

The effectiveness of hybrid retrieval depended strongly on the quality of the dense retriever. The largest gains were observed for weaker models: Muffakir V2 improved from 0.3514 to 0.4979 MRR@5 (+0.1465), while other mid- and lower-performing retrievers also showed substantial improvements. This suggests that lexical and semantic signals are genuinely complementary when the dense representation is less effective.

For the strongest fine-tuned retrievers, however, the gains were small and not statistically significant. Fine-tuned ATM2 improved from 0.5377 to 0.5642 MRR@5 (+0.0265), while fine-tuned Muffakir improved from 0.5534 to 0.5739 (+0.0205).

Model	MRR@5			Latency (ms)		
	Dense	Hybrid	Δ	Dense	Hybrid	×
BGE-fiqh	0.501	0.538	+0.036	119	306	2.6
ATM2-fiqh	0.538	0.564	+0.026	109	294	2.7
Muffakir-fiqh	0.553	0.574	+0.021	121	279	2.3

Table 2: Effectiveness and efficiency of dense-only vs. hybrid retrieval. For the three fiqh fine-tuned models, hybrid fusion roughly doubles latency for a small, non-significant MRR@5 gain.

Model	MRR@5		Recall@5	
	General	Madhhab	General	Madhhab
ATM2-fiqh	0.706	0.483	0.075	0.165
Muffakir-fiqh	0.695	0.508	0.084	0.178
BGE-fiqh	0.670	0.447	0.068	0.141

Table 3: Retrieval performance on general and madhhab-based questions. General questions achieve higher MRR@5, while madhhab-based questions achieve higher Recall@5.

Among the fine-tuned models, only BGE-fiqh showed a significant hybrid improvement, and only on nDCG@5 among the main metrics.

Importantly, the limited gains do not indicate that the hybrid failed to exploit BM25's complementary signal. Of the questions missed by dense retrieval but retrieved by BM25, the hybrid recovered 70.4% for ATM2 and 77.1% for Muffakir. However, these improvements were partially offset by regressions on questions already answered by the dense model, resulting in only modest net gains. In effect, hybrid retrieval mainly redistributes hits near the rank-5 boundary rather than substantially improving ranking quality.

The practical value of hybrid retrieval therefore depends on the strength of the underlying dense retriever. As shown in Table 2, hybrid retrieval increased latency by approximately 2.3–2.7× while providing only small improvements for the strongest models. We therefore conclude that hybrid retrieval is highly beneficial for weaker retrievers, but its advantage becomes limited once strong fiqh-specific dense retrievers are used. Balancing effectiveness and efficiency, fine-tuned ATM2 and Muffakir operating in dense-only mode represent the most practical choice.

6.2 RQ2: Madhhab-aware retrieval

RQ2 investigates two aspects of madhhab-aware retrieval: differences in retrieval performance between general and madhhab-based questions (RQ2.1) and the effect of restricting retrieval to the relevant madhhab (RQ2.2).

***RQ2.1*:** Retrieval performance differs substantially between the two question types (Table 3). Fine-tuned ATM2 achieves an MRR@5 of 0.706 on general questions compared to 0.483 on madhhab-based questions, and fine-tuned Muffakir achieves 0.695 and 0.508, respectively, which is a gap of roughly 0.20 MRR@5 in both cases. Similar trends are observed across all evaluated models.

A major reason for this gap is the difference in the number of answer-bearing chunks available for each question type. General questions have, on average, more than four times as many relevant chunks as madhhab-based questions (41.7 vs. 9.7). When many ground truth chunks exist, the probability that at least one appears in the top retrieved results is higher, leading to higher Hit@5 and MRR@5 scores. This is supported by Recall@5, which is higher for madhhab questions (0.165) than for general questions (0.075), indicating that the retriever retrieves a larger fraction of the smaller relevant set despite its lower MRR@5.

However, the gap cannot be explained solely by the size of the relevant set. Fiqh fine-tuning consistently improves retrieval on madhhab-based questions while having a much smaller effect on general questions. For example, ATM2 improves from 0.372 to 0.483 MRR@5 on madhhab-based questions, reducing the performance gap between general and madhhab-based questions from 0.284 to 0.223. Similar reductions are observed for Muffakir and BGE-M3. Fine-tuning therefore substantially improves performance on madhhab-based questions and narrows the gap with general questions. Nevertheless, a considerable difference remains because the limited number of answer-bearing chunks for madhhab-based questions imposes a constraint that model adaptation alone cannot fully overcome. ***RQ2.2:*** We introduce a madhhab-filtering stage that restricts retrieval to the passages of the relevant school whenever a question is identified as school-specific. This stage produces one of the largest performance improvements observed in our pipeline.

On madhhab-based questions, filtering more than doubles retrieval effectiveness (Figure 1). MRR@5 rises from 0.211 to 0.483 for fine-tuned ATM2, from 0.193 to 0.508 for fine-tuned Muffakir, and from 0.187 to 0.447 for fine-tuned bge,

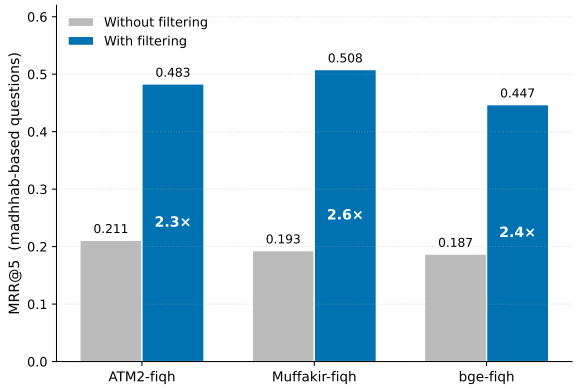


Figure 1: Effect of madhhab filtering on madhhab questions for the three fine-tuned models.

with comparable gains in Hit@5 (all $p < 10^{-25}$, Wilcoxon signed-rank). This improvement exceeds that of fiqh fine-tuning or hybrid fusion, yet comes at negligible cost, as the routing is rule-based and adds no learned components. This gain results from reducing the cross-school confusion described in Section 4.1. When filtering is removed, Recall@5 on madhhab-based questions drops sharply (e.g., for fine-tuned ATM2, from 0.165 to 0.061), as passages from other schools compete with and displace the answer-bearing chunks from the top retrieved results.

7 Error Analysis

To better understand retrieval failures, we analyze them from both quantitative and qualitative perspectives. We focus on questions that received a Hit@5 score of zero for the both top-performing models, namely fine-tuned ATM2 and fine-tuned Muffakir. This allows us to examine failure cases that persist even under the strongest retrieval configurations.

7.1 Quantitative Analysis

Recall vs. ranking failures. We decompose each missed question ($Hit@5 = 0$) into two mechanisms: either no answer-bearing passage is retrieved into the candidate pool (a *recall* failure), or an answer-bearing passage is retrieved but ranked outside the top five results (a *ranking* failure). Across both retrievers, recall failures are more common: for roughly two-thirds of misses, no answer-bearing passage appears in the top 20 candidates (62% for fine-tuned ATM2 and 64% for fine-tuned Muffakir). The remaining cases are ranking failures. This indicates that retrieval failures are driven primarily by difficulty retrieving answer-bearing passages into the candidate pool rather than ranking them.

Separability. Similarity scores provide further insight into the ranking failures. On missed questions, the top-ranked passage still receives a high similarity score (0.618 for fine-tuned ATM2 and 0.640 for fine-tuned Muffakir), only slightly below successful queries (0.685 and 0.702, respectively). Retrieval failures therefore do not arise from the absence of semantically related passages, but from difficulty distinguishing answer-bearing passages from highly similar non-answer-bearing ones.

This is reflected in the similarity margin between the highest-scoring answer-bearing passage and the highest-scoring non-answer-bearing passage. On missed questions, the margin is negative (-0.079 for ATM2 and -0.081 for Muffakir), indicating that a non-answer-bearing passage receives a higher score than any available answer-bearing passage. Even on successful queries, the margin remains small (+0.019 and +0.023), and among cases where an answer-bearing passage reaches the candidate pool, a non-answer-bearing passage still receives a higher score in half of the queries (52% and 48%). These results suggest that answer-bearing and topically related non-answer-bearing passages remain only weakly separable in the embedding space.

7.2 Qualitative Analysis

The quantitative analysis shows that retrieval failures are often caused by confusion among semantically related passages. To understand the nature of these confusions, we manually examined questions that received a Hit@5 score of zero for both top-performing models and analyzed the passages commonly retrieved by both to focus the analysis on shared retrieval failures rather than model-specific errors.

We identified four categories of retrieval errors that are largely driven by the characteristics of fiqh texts. Since a single failure may exhibit multiple patterns and the boundaries between categories are not always clear-cut, we present these categories qualitatively rather than as a quantitative distribution. Representative examples are shown in Table 4.

Condition Mismatch: Fiqh rulings are often highly dependent on specific conditions and circumstances. Retrieval failures occur when the model overlooks a condition mentioned in the ques-

Ctg.	Question	Retrieved Passage	Why it is Incorrect
CM	كيف يكون بر الوالدين بعد موتهما؟ عند الشافعية	وأفضل البر بر الوالدين بالإحسان إليهما وفعل ما يسرهما من الطاعة لله تعالى وغيرها...	Retrieves a general ruling on honoring parents while missing the condition *after their death*.
QTM	متى يضمن المالك في المزارعة	المزارعة : الشركة في الزرع,ويقال : الشركة في الحرث. وعقدها غير لازم قبل البذر ونحوه...	Provides contract definition and validity information rather than the requested liability conditions.
CI	ما قول الحنفية في: حكم قتل الأصل للفرع المحارب من الكفار؟	...لا يجوز له قتل أصله الحربي إلا دفعا عن نفسه , وإن خاف رجوعه ضيق عليه وألجأه ليقتله غيره...	Matches the general issue but misses the specific roles and circumstances
TM	لو اقترض أحد الزوجين من الآخر، فهل يحق له الحبس بهذا القرض؟	في المقارض يأخذ من رجل آخر مالا قراضا قلت : أرأيت إن أخذ رجل مالا قراضا من رجل...	Confuses similar terms: القراض (profit-sharing) vs. القرض(loan).

Table 4: Representative retrieval errors. CM: Condition Mismatch; QTM: Question-Type Mismatch; CI: Context Misinterpretation; TM: Topic Mismatch.

tion and retrieves a more general ruling, retrieves a ruling restricted to conditions not present in the question, or retrieves a passage discussing a different condition within the same underlying issue.

Question-Type Mismatch: Fiqh questions may ask about different aspects of a topic, such as the ruling, underlying reasoning, validity conditions, obligations, or exceptions. In some cases, the retrieved passage discusses the correct topic and scenario but answers a different type of question, making it irrelevant despite its high semantic similarity.

Context Misinterpretation: Some fiqh cases involve multiple actors, actions, or relationships whose roles determine the ruling. The retriever may identify the correct topic and conditions but fail to distinguish the context, retrieving passages that describe a similar scenario with reversed or altered roles.

Topic Mismatch: In these cases, the retriever fails to capture the intended meaning of the question and retrieves passages from a different fiqh topic altogether. This may occur because certain terms appear similar despite referring to distinct jurisprudential concepts.

Observations. Taken together, the quantitative and qualitative analyses show that retrieval failures in fiqh are mainly caused by the difficulty of distinguishing between closely related passages rather than by retrieving completely unrelated content. The quantitative results show that missed questions often still receive high similarity scores, but the retrieved passages are difficult to separate from the relevant ones because answer-bearing passages are only weakly distinguished from topically similar non-answer-bearing passages. The qualitative analysis further explains the source of this confusion: most errors involve passages that discuss the same underlying issue but differ in a critical aspect, such as a condition, the type of information requested, or the roles of the involved actors. Even topic mismatch cases often result from confusion between closely related jurisprudential terms rather than a complete failure to identify the general area of the question. These findings indicate that current retrieval models capture broad semantic similarity effectively but struggle with the fine-grained distinctions that determine the applicability of fiqh rulings. Therefore, improving fiqh retrieval requires retrieval strategies that better capture conditions, contextual roles, and question types, rather than relying on improved ranking, as many failures occur because the relevant passage is not retrieved into the candidate pool in the first place.

8 Conclusion

This paper presented a retrieval-focused evaluation of answer-bearing fiqh retrieval. Our results show that fiqh fine-tuning improves dense retrievers, while hybrid retrieval adds useful lexical signals but gives only modest gains for the strongest fine-tuned models. Madhhab-aware filtering provides the largest improvement on school-specific questions by reducing cross-school confusion.

Our error analysis shows that remaining failures are mainly caused by difficulty distinguishing answer-bearing passages from topically similar passages. These findings suggest that future work should focus on harder fiqh-specific negatives and retrieval models that better capture fine-grained juristic distinctions.

Limitations

Our study has several limitations. First, our relevance judgments are binary: a passage is treated as either answer-bearing or non-relevant. This does not capture partial relevance, such as passages that provide part of a ruling or state a ruling without its supporting conditions. A graded relevance scheme would represent such cases more faithfully but would require substantially more annotation effort.

Second, the relevance annotation followed an expert review and checking procedure rather than independent double annotation. While this provided expert verification of disputed cases, it does not allow us to report inter-annotator agreement. Future work could strengthen the validation process by incorporating multiple LLM judgments and independent annotations from multiple Islamic Studies experts.

Third, the fine-tuning questions were automatically generated using an LLM. Although the generation prompts were developed with Islamic Studies experts and sampled outputs were manually inspected, the full synthetic dataset was not exhaustively human-reviewed. The generated data may therefore contain occasional wording artifacts or imperfect question–passage pairs.

Fourth, our madhhab detection is lexicon-based and therefore susceptible to surface-level collisions. For example, the word المالك (the owner) contains the substring مالك (Mālik), causing one general question in our test collection to be incorrectly flagged as Maliki-specific. Such cases were rare and did not affect the reported results, but a morphology-aware detector would reduce these errors.

Finally, our evaluation is restricted to the retrieval stage and to models of at most approximately 0.6B parameters. We do not measure how retrieval improvements translate to downstream answer quality, nor do we evaluate larger retrievers or alternative architectures such as late-interaction models. In addition, the retrieval errors identified in our analysis suggest that future work should explore harder fiqh-specific negatives and retrieval representations that better capture fiqh semantics, particularly distinctions involving conditions, contextual roles, and closely related jurisprudential concepts.

References

Ummar Abbas, Mourad Ouzzani, Mohamed Y. Eltabakh, Omar Sinan, Gagan Bhatia, Hamdy Mubarak, Majd Hawasly, Mohammed Qusay Hashim, Kareem Darwish, and Firoj Alam. 2026. Fanar-sadiq: A multi-agent architecture for grounded islamic qa. *Preprint*, arXiv:2603.08501.

Mostafa G Abd Elhakeem, Amira O Abdullah, Sara W Elhag, and Ensaf Hussein Mohamed. 2026. Aqrag: advanced quranic retrieval-augmented generation for low-resource question answering. *Neural Comput. Appl.*, 38(8).

Ali Abdelaal, Mohammed Nader Al Haffar, Mahmoud Fawzi, and Walid Magdy. 2026. Islamicmmlu: A benchmark for evaluating llms on islamic knowledge. *Preprint*, arXiv:2603.23750.

Arij Al Adel, Abu Bakr Soliman, Mohamed Sakher Sawan, Rahaf Al-Najjar, and Sameh Amin. 2025. BurhanAI at IslamicEval 2025 shared task: Combating hallucinations in LLMs for islamic content; evaluation, correction, and retrieval-based solution. In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 503–508, Suzhou, China. Association for Computational Linguistics.

Sadam Al-Azani, Maad Alowaifeer, Alhanoof Alhunief, and Ahmed Abdelali. 2025. OntologyRAG-Q: Resource development and benchmarking for retrieval-augmented question answering in qur'anic tafsir. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 15540–15558, Suzhou, China. Association for Computational Linguistics.

Ahmet Yusuf Alan, Enis Karaarslan, and Omer Aydin. 2025. Improving llm reliability with rag in religious question-answering: Mufassirqas. *Preprint*, arXiv:2401.15378.

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: Shared Tasks*, pages 494–502, Suzhou, China. Association for Computational Linguistics.

Farah Atif, Nursultan Askarbekuly, Kareem Darwish, and Monojit Choudhury. 2025. Sacred or synthetic? evaluating llm reliability and abstention for religious questions. *Preprint*, arXiv:2508.08287.

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 *Advances on Intelligent Computing and Data Science II*, pages 244–254, Cham. Springer Nature Switzerland.

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: Shared Tasks*, pages 534–539, Suzhou, China. Association for Computational Linguistics.

Gagan Bhatia, Hamdy Mubarak, Mustafa Jarrar, George Mikros, Fadi Zaraket, Mahmoud Alhirthani, Mutaz Al-Khatib, Logan Cochrane, Kareem Darwish, Rashid Yahiaoui, and Firoj Alam. 2026. From rag to agentic rag for faithful islamic question answering. *Preprint*, arXiv:2601.07528.

Abdessalam Bouchekif, Somaya Eltanbouly, Samer Rashwani, Shahd Gaben, Mutaz Al-Khatib, Heba Sbahi, Emad Mohamed, and Mohammed Ghaly. 2026a. Qias 2026: Overview of the shared task on islamic inheritance reasoning. *Preprint*, arXiv:2606.13756.

Abdessalam Bouchekif, Shahd Gaben, Samer Rashwani, Somaya Eltanbouly, Mutaz Al-Khatib, Heba Sbahi, Mohammed Ghaly, and Emad Mohamed. 2026b. Mawarith: A dataset and benchmark for legal inheritance reasoning with llms. *Preprint*, arXiv:2603.07539.

Abdessalam Bouchekif, Samer Rashwani, Emad Soliman Ali Mohamed, Mutaz Alkhatib, Heba Sbahi, Shahd Gaben, Wajdi Zaghouani, Aiman Erbad, and Mohammed Ghaly. 2025. QIAS 2025: Overview of the shared task on islamic inheritance reasoning and knowledge assessment. In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 851–860, Suzhou, China. Association for Computational Linguistics.

Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2318–2335, Bangkok, Thailand. Association for Computational Linguistics.

Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, page 758–759, New York, NY, USA. Association for Computing Machinery.

Mohammed ElKoumy, Khalid Allam, Ahmad Tamer, and Mohamed Alqablawi. 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: Shared Tasks*, pages 515–527, Suzhou, China. Association for Computational Linguistics.

Ezieddin Elmahjub, Junaid Qadir, Abdullah Mushtaq, Rafay Naeem, Ibrahim Ghaznavi, and Waleed Iqbal. 2026. Islamiclegalbench: Evaluating llms knowledge and reasoning of islamic law across 1,200 years of islamic pluralist legal traditions. *Preprint*, arXiv:2602.21226.

Somaya Eltanbouly and Samer Rashwani. 2026. Grounding arabic llms in the doha historical dictionary: Retrieval-augmented understanding of quran and hadith. *Preprint*, arXiv:2603.23972.

Shahd Gaben, Heba Sbahi, Samer Rashwani, Abdessalam Bouchekif, Mutaz Al-Khatib, Emad Mohamed, Somaya Eltanbouly, and Mohammed Ghaly. 2026. Islamicturathbench: A multi-task, multi-discipline benchmark for evaluating large language models on the islamic scholarly tradition (turath). *Preprint*, arXiv:2608.04703.

Asti Herliana, Ina Najiyah, Sari Susanti, and Lutfhi Billah. 2025. The application of deep learning in qur'anic tafsir retrieval using sbert, faiss and bert-qa. *Journal of Applied Data Sciences*, 7(1):344–356.

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).

Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, and Ali Farhadi. 2022. Matryoshka representation learning. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

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 Comput. Appl.*, 37(25):20957–20982.

Hamdy Mubarak, Rana Malhas, Watheq Mansour, Abubakr Mohamed, Mahmoud Fawzi, Majd Hawasly, Tamer Elsayed, Kareem Mohamed Darwish, and Walid Magdy. 2025. IslamicEval 2025: The first shared task of capturing LLMs hallucination in islamic content. In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 480–493, Suzhou, China. Association for Computational Linguistics.

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. *Preprint*, arXiv:2505.24581.

Zach Nussbaum and Brandon Duderstadt. 2025. Training sparse mixture of experts text embedding models. *Preprint*, arXiv:2502.07972.

Surayya Obaid and Narmeen Zakaria Bawany. 2024. Seerahgpt: Retrieval augmented generation based large language model. In *2024 18th International Conference on Open Source Systems and Technologies (ICOSST)*, pages 1–7.

Vera Pavlova. 2023. Leveraging domain adaptation and data augmentation to improve qur'anic IR in English and Arabic. In *Proceedings of ArabicNLP 2023*, pages 76–88, Singapore (Hybrid). Association for Computational Linguistics.

Vera Pavlova. 2025. Multi-stage training of bilingual islamic LLM for neural passage retrieval. In *Proceedings of the New Horizons in Computational Linguistics for Religious Texts*, pages 42–52, Abu Dhabi, UAE. Association for Computational Linguistics.

Vera Pavlova and Mohammed Makhlouf. 2025. Efficient and versatile model for multilingual information retrieval of islamic text: Development and deployment in real-world scenarios. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1239–1249, Suzhou (China). Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong

Kong, China. Association for Computational Linguistics.

Ayman A. Taan, Shafiq Ur Rehman Khan, Ali Raza, Ayaz Muhammad Hanif, and Hira Anwar. 2021. Comparative analysis of information retrieval models on quran dataset in cross-language information retrieval systems. *IEEE Access*, 9:169056–169067.

Mohamed Khaled Tawfik, Mohamed Osama Ali, Sohier Khaled Mohamed, Farah Safwat Ezz Elregal, Paula Emad Edward, Maryam Elsawaf, and Dina Khattab. 2025. Mufakkir: Rag-based arabic educational chatbot for university students. In *2025 Twelfth International Conference on Intelligent Computing and Information Systems (ICICIS)*, pages 706–712.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual e5 text embeddings: A technical report. *Preprint*, arXiv:2402.05672.

Hao Yu, Aoran Gan, Kai Zhang, Shiwei Tong, Qi Liu, and Zhaofeng Liu. 2025. Evaluation of retrieval-augmented generation: A survey. In *Big Data*, pages 102–120, Singapore. Springer Nature Singapore.

Puxuan Yu, Luke Merrick, Gaurav Nuti, and Daniel Campos. 2024. Arctic-embed 2.0: Multilingual retrieval without compromise. *Preprint*, arXiv:2412.04506.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

A Appendix

A.1 Data Construction Details

A.1.1 Madhhab Annotation

Each chunk is assigned one or more madhhab labels. For books associated with a specific madhhab, all extracted chunks inherit that madhhab as their primary label. In addition, chunks may receive additional labels when they explicitly discuss the opinions of other madhahib. For example, a Hanafi text comparing the Hanafi and Shafi'i positions would be assigned both Hanafi and Shafi'i labels, with Hanafi retained as the primary label. This allows comparative fiqh discussions to be retrieved when searching for either school.

Madhhab	**# Chunks**
Hanafi	100,950
Maliki	61,302
Shafi'i	95,780
Hanbali	46,409
Other Schools	17,345
General / Comparative Fiqh	34,331
Total	**356,117**

Table 5: Distribution of chunks by primary madhhab label in the retrieval corpus.

Madhhab mentions are detected through explicit references to madhhab names as well as references to recognized scholars associated with a particular school. For books that are not attributed to a specific madhhab, labels are assigned solely based on the madhhab mentions identified within the text.

Table 5 reports the distribution of chunks across madhahib.

A.1.2 Fine-tuning data details

Questions and answers were generated from a subset of five fiqh books using Gemini 2.5 Flash. The model was provided with fiqh chunks containing the original in-text structural annotations and was instructed to generate question-answer pairs grounded in the supplied passage. The generation prompts were developed in consultation with experts in Islamic studies and refined through multiple rounds of review. Several generated batches were manually inspected during development to verify that the questions were faithful to the source passages, reflected the intended ruling, and were suitable for retrieval training.

To reduce redundancy, at most one question was retained from each chunk. Hard negatives were selected from the same section as the positive passage but from a different subsection, producing topically related yet non-answer-bearing negatives. This strategy helps the retriever to distinguish between passages discussing closely related fiqh issues rather than relying on broad topical differences. As a sample-based validation of this procedure, 100 negative passages were manually reviewed, and all were confirmed to be non-answer

bearing for their corresponding questions. This check was performed on a sample and does not constitute manual validation of the full negative dataset.

The final dataset contains 19,319 triplets distributed across the Sunni schools (5,040 Shafi'i, 5,001 Hanafi, 4,879 Hanbali, and 4,399 Maliki). An example triplet is shown in Table 6.

A.1.3 Test Collection Construction

Initial Relevance Pool: Questions were authored with reference to specific juristic scenarios discussed in the source fiqh books. Each scenario is defined by a particular set of conditions and circumstances under which a ruling is sought. For each question, all chunks associated with the corresponding scenario were initially included in the relevance pool, as these chunks collectively discuss the case from which the question was derived.

Answer-Bearing Verification: Although chunks associated with the same scenario generally discuss the same juristic case, not all of them necessarily provide the specific ruling requested by a question. To identify such cases, all chunks in the initial relevance pool were assessed according to the answer-bearing relevance criterion described in Section 5.3.

To identify potentially missed relevant passages, we additionally retrieved the top 100 BM25 passages for each question. These passages were combined with the original scenario pool and screened by Gemini 2.5 Flash according to the answer-bearing relevance criterion.

LLM judgments were used only to identify candidate additions and removals. Whenever the LLM judgment disagreed with the original scenario-based label, the chunk was reviewed by a human annotator. Chunks were removed from the relevance set only when the annotator confirmed that they did not provide the requested ruling. Similarly, retrieved passages outside the original scenario pool were added only when the annotator confirmed that they contained the requested ruling. Human judgments were treated as definitive in all cases.

To prevent train–test contamination, we ensured that no chunk included in the test collection appeared in the fine-tuning dataset. The training and evaluation data are therefore disjoint at the chunk level.

Table 7 shows representative examples of a general question and a madhhab-based question together with excerpts from answer-bearing passages judged relevant under our annotation criteria.

Annotation Statistics: A total of 3,657 disputed chunk labels were reviewed by human annotators. Human annotators agreed with the Gemini judgment in 80.0% of reviewed disagreements and with the original label in the remaining 20.0%.

Among the reviewed disagreements, annotators confirmed the removal of 1,612 chunks from the original scenario pools because they did not directly answer the corresponding question, and added 1,313 answer-bearing chunks retrieved outside the original scenario.

A.2 Models Details

Table 8 summarizes the architectural characteristics of the eight evaluated embedding models. Most are encoder-based retrievers, while Qwen3-Embedding-0.6B is the only decoder-based model. ATM2 serves as the base model for Muffakir V1, whereas Muffakir V2 is derived from BGE-M3, allowing comparison of different legal-domain adaptation paths. The models also vary in parameter count, embedding dimensionality, context length, and pooling strategy.

A.3 Metrics and Statistical Testing

A.3.1 Evaluation Metrics

Retrieval effectiveness is evaluated at rank $k = 5$, corresponding to the number of passages typically supplied to a generator in a RAG pipeline. **MRR@5** (Mean Reciprocal Rank): Measures how highly the first answer-bearing passage is ranked. **nDCG@5** (Normalized Discounted Cumulative Gain): Measures the overall ranking quality of relevant passages, giving higher weight to passages appearing near the top of the ranking. **Hit@5**: Measures whether at least one answer-bearing passage appears among the top five retrieved results. **MAP@5** (Mean Average Precision): Measures ranking quality by averaging precision over all relevant passages retrieved within the top five results. **Recall@5**: Measures the fraction of all answer-bearing passages retrieved within the top five results. Because many questions have more relevant passages than the retrieval cutoff, Recall@5 is used primarily as a diagnostic measure rather than a primary evaluation metric.

For latency, we report the mean per-query wall-clock retrieval time in milliseconds. Timing wraps only the retrieval call and excludes data loading

Element	Content
Question	ما حكم الصلح عن شيء مغصوب تالف بأكثر من قيمته (كألفين مقابل ألف) أو بعرض (سلعة)؟
Pos. Passage	"غصب ثوباً أو عبداً قيمته ألف فاستهلكه، فصالحه على ألفين أو عرض جاز ... وكذا الصلح بعرض صح وإن كان قيمته أكثر من قيمة المغصوب التالف لعدم الربا."
Neg. Passage	"وركنه الإيجاب والقبول ... ويصح الصلح مع الإقرار، والإقرار أقوى من الشهادة ..."
Why Negative	The passage discusses the general validity and requirements of *ṣulḥ* (settlement), making it topically related to the question. However, it does not address settlement over a destroyed usurped item, nor does it provide the specific ruling requested by the question.

Table 6: Example fine-tuning triplet. The negative passage is topically related to the question but does not provide the requested ruling.

Type	Question	Relevant Passage (Excerpt)
General	ما حكم الالتفات في الصلاة؟	لا خلاف بين الفقهاء في كراهة الالتفات في الصلاة ... والكراهة مقيدة بعدم الحاجة أو العذر.
Madhhab-based	وفق المذهب المالكي، ماذا يحدث لعقد الاستصناع إذا مات أحد المتعاقدين؟	الصناع أحق بما أسلم إليهم في الموت والفلس ما كان بأيديهم ...

Table 7: Examples of questions in the test collection and corresponding answer-bearing passages. Only excerpts containing the relevant ruling are shown.

and index construction, thereby measuring the end-to-end cost of the retrieval pipeline itself.

A.3.2 Statistical Testing

All statistical comparisons use a two-sided Wilcoxon signed-rank test applied to per-question retrieval scores. This test was selected because retrieval metrics are bounded, non-normal, and computed on paired observations (the same questions evaluated under different retrieval configurations).

Holm–Bonferroni correction was applied within each analysis to account for multiple comparisons. Unless otherwise stated, significance levels are reported as: * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$, ns: not significant.

Four groups of statistical comparisons were performed: (1) **Fine-tuning Effect**: Comparison of each base model with its fiqh-fine-tuned counterpart.(2) **Fine-Tuned Model Comparison**: Pairwise comparisons among the fiqh-fine-tuned models.(3) **Hybrid Retrieval**: Comparison of dense-only and hybrid (dense + BM25) retrieval configurations.(4) **Madhhab Filtering**: Comparison of retrieval performance with and without madhhab-based filtering. Detailed results for all statistical tests are reported in Appendix A.4.

A.3.3 Implementation Details

BM25: BM25 retrieval is implemented using the Qdrant/FastEmbed BM25 model. Before indexing, Arabic diacritics were removed from all corpus texts, as queries are typically written without diacritics; this preprocessing reduces term mismatches between queries and indexed passages. **Fine-tuning:** Models are trained for 3 epochs with a batch size of 128, a learning rate of 2×10^{-5}, and a fixed seed of 42. We reserve 5% of the training triplets for validation and select the checkpoint with the highest validation performance. **Retrieval:** all passage embeddings are stored in a Qdrant collection using named vectors, allowing the same corpus to be searched with different embedding models. All experiments were conducted on a single NVIDIA RTX 5880 Ada GPU.

A.4 Statistical Significance

We assess statistical significance using the two-sided Wilcoxon signed-rank test over per-question scores. The test is paired (the same questions are scored under both conditions) and non-parametric (per- question retrieval metrics are bounded and non-normal). Within each analysis we apply Holm–Bonferroni correction across all reported (model $\times$ metric) comparisons. We denote $p < 0.05$ (*), $p < 0.01$ (**), $p < 0.001$ (***), and "ns" otherwise.

Model	Category	Arch.	Backbone	Params	Dim	Ctx	Pooling
BGE-M3	Multi.	Enc	XLM-R-large	568M	1024	8192	CLS
Qwen3-Emb-0.6B	Multi.	Dec	Qwen3-0.6B	600M	1024	32768	Last tok
Arctic-Embed-L-v2	Multi.	Enc	XLM-R-large	568M	1024	8192	CLS
Nomic-Embed-v2	Multi.	Enc (MoE)	Nomic BERT	475M*	768	512	Mean
mE5-large-inst	Multi.	Enc	XLM-R-large	560M	1024	512	Mean
ATM2	Arabic	Enc	AraBERTv02	135M	768	512	Mean
Muffakir V1	Domain Ar.	Enc	AraBERTv02	135M	768	512	Mean
Muffakir V2	Domain Ar.	Enc	XLM-R-large	362M	1024	8192	CLS

Table 8: Summary of the eight retrieval models evaluated. Multi. = multilingual, Enc = encoder, Dec = decoder, and Ctx = maximum context length. *Nomic-Embed-v2 is a mixture-of-experts model with 475M total parameters (305M active).

Model	ΔMRR@5	ΔnDCG@5	ΔHit@5
ATM2-fiqh	$+0.097^{***}$	$+0.068^{***}$	$+0.109^{***}$
BGE-fiqh	$+0.059^{**}$	$+0.022$ ns	$+0.066^{**}$
Muffakir-fiqh	$+0.030$ ns	$+0.018$ ns	$+0.034$ ns

Table 9: Significance of fiqh fine-tuning (fine-tuned − base, dense, $n = 503$).

Model	MRR@5	nDCG@5	Hit@5	MAP@5	Rec@5
ATM2-fiqh	ns	ns	ns	ns	ns
Muffakir-fiqh	ns	ns	ns	ns	ns
BGE-fiqh	ns	***	ns	***	***

Table 11: Significance of hybrid vs. dense retrieval (fine-tuned models, $n = 503$), Holm-corrected.

Effect of fiqh fine-tuning. Table 9 compares each base model with its fiqh-fine-tuned version. Fine-tuning is statistically significant for ATM2 across all metrics and for BGE-M3 on MRR@5 and Hit@5. For Muffakir V1, which starts from the strongest baseline, the gains are positive but not significant, consistent with its limited headroom.

Pairwise comparison of fine-tuned models. Table 10 reports pairwise tests among the three fine-tuned models. The two AraBERT-based models, ATM2-fiqh and Muffakir-fiqh, are not significantly different on any metric, supporting the convergence reported in Section 6.1. Both differ from BGE-fiqh on nDCG@5.

Comparison	MRR@5	nDCG@5	Hit@5
ATM2-fiqh vs Muffakir-fiqh	ns	ns	ns
ATM2-fiqh vs BGE-fiqh	ns	***	ns
Muffakir-fiqh vs BGE-fiqh	*	***	*

Table 10: Pairwise significance among fine-tuned models (dense), Holm-corrected.

Dense vs. hybrid retrieval. Table 11 reports hybrid-vs-dense differences for the fine-tuned models. For ATM2-fiqh and Muffakir-fiqh, no metric shows a significant hybrid gain. For BGE-fiqh, the gain is significant on nDCG@5, MAP@5, and Recall@5, but not on MRR@5 or Hit@5. This supports the conclusion that hybrid fusion offers little benefit to strong fine-tuned retrievers.

Madhhab filtering. Table 12 reports the effect of madhhab filtering on the 380 madhhab-based questions. Filtering yields highly significant improvements for all three fine-tuned models across every metric ($p < 10^{-24}$ throughout), confirming that restricting retrieval to the relevant school is the single largest source of improvement in our pipeline.

Model	ΔMRR@5	ΔnDCG@5	ΔHit@5
ATM2-fiqh	$+0.272^{***}$	$+0.202^{***}$	$+0.311^{***}$
Muffakir-fiqh	$+0.315^{***}$	$+0.219^{***}$	$+0.297^{***}$
BGE-fiqh	$+0.260^{***}$	$+0.171^{***}$	$+0.292^{***}$

Table 12: Significance of madhhab filtering (filtered − unfiltered) on the 380 madhhab-based questions. All metrics significant at $p < 10^{-24}$.

A.5 Additional Results

A.5.1 Performance at Different Retrieval Cutoffs

Table 13 reports retrieval effectiveness for the three best-performing fine-tuned retrievers at different retrieval cutoffs. While the main paper focuses on k=5, which matches the number of passages typically supplied to a generator in a RAG pipeline, these results illustrate how retrieval effectiveness changes as additional passages are retrieved.

A.5.2 Madhhab Detection Evaluation

We evaluated the rule-based madhhab detector on the test collection to verify that the filtering stage

Model	k	MRR@k	nDCG@k	Hit@k
ATM2-fiqh	1	0.4414	0.4414	0.4414
	3	0.5215	0.3911	0.6183
	5	0.5377	0.3638	0.6899
	10	0.5468	0.3317	0.7535
	20	0.5513	0.3189	0.8171
Muffakir-fiqh	1	0.4573	0.4573	0.4573
	3	0.5401	0.3979	0.6481
	5	0.5534	0.3764	0.7058
	10	0.5623	0.3454	0.7694
	20	0.5661	0.3328	0.8231
BGE-fiqh	1	0.4215	0.4215	0.4215
	3	0.4854	0.3480	0.5686
	5	0.5014	0.3187	0.6402
	10	0.5103	0.2892	0.7078
	20	0.5142	0.2777	0.7654

Table 13: Retrieval effectiveness at different k for the fine-tuned models. MRR@k and Hit@k increase with larger candidate pools, nDCG@k decreases because it is normalized against the full set of relevant passages per question.

correctly identifies school-specific questions. On the 380 madhhab-based questions, the detector achieved perfect recall (1.000), precision of 0.985, and F1 of 0.992. The correct madhhab was detected for every school-specific question, with exact single-school matches in 375 out of 380 cases (98.7%). On the 123 general questions, only one false school detection occurred.

Most errors were false positives where an additional madhhab was detected alongside the expected one. Per-school recall was 1.000 for all four madhahib, while precision was 1.000 for Hanbali, 0.990 for Hanafi, 0.990 for Shafi'i, and 0.958 for Maliki. These results indicate that the filtering stage is reliable for routing madhhab-based questions, while the remaining errors are rare and mainly due to surface-form ambiguity.

From Subwords to Pixels: Evaluating Text Representations for Arabic Machine Translation

Abdelrahman Shaheen[1] **Mohamed Abbas**[2] **Tereza Truckova**[3]
Yova Kementchedjhieva[4] **Bashar Alhafni**[4]
[1]Egypt-Japan University of Science and Technology [2]American University in Cairo
[3]Prague University of Economics and Business
[4]Mohamed bin Zayed University of Artificial Intelligence
{firstname.lastname}@mbzuai.ac.ae

Abstract

Subword tokenization underpins modern neural NLP but can be suboptimal for morphologically rich languages with substantial orthographic variation. In this paper, we compare vocabulary-based and vocabulary-free representations for Arabic-to-English machine translation, including subword, byte-level, and pixel-level approaches. Across 35 evaluation sets spanning Modern Standard Arabic (MSA) and 18 dialects, all approaches achieve comparable translation performance with sufficient supervision. While subword models perform best with limited MSA pretraining data, differences diminish with scale and dialect-specific fine-tuning. Pixel-based models further exhibit the greatest robustness to unseen orthographic variation and provide the highest training and inference throughput. We make our code and models publicly available.[1]

1 Introduction

Subword tokenization has become the de facto standard in neural natural language processing (NLP) (Sennrich et al., 2016; Kudo and Richardson, 2018; Mielke et al., 2021). However, it may be suboptimal for morphologically rich and underrepresented languages (Rust et al., 2021; Pfeiffer et al., 2021; Arnett and Bergen, 2025). In Arabic, this challenge is further compounded by the coexistence of Modern Standard Arabic (MSA) and numerous Dialectal Arabic (DA) varieties, which exhibit substantial orthographic, morphological, syntactic, and lexical variation. Under standard subword tokenization, such variation often leads to severe overtokenization, resulting in longer input sequences, reduced compression rates, and fragmented representations. These issues can ultimately affect both the efficiency and effectiveness of downstream applications, including machine translation (MT).

To address these limitations, prior work has explored both vocabulary-based and vocabulary-free alternatives to standard subword tokenization. Within the vocabulary-based paradigm, approaches such as linguistically informed tokenization and vocabulary expansion seek to better align tokenization with linguistic structure and improve vocabulary coverage (Tawfik et al., 2019; Liang et al., 2023; Asgari et al., 2025). Recently, vocabulary-free representations have emerged as a promising alternative. These include byte-level models, which operate on raw bytes, and pixel-based approaches, which encode text as images and eliminate the need for tokenization (Salesky et al., 2021; Rust et al., 2023; Lotz et al., 2025; Kesen et al., 2025). Despite these advances, the merits of vocabulary-based and vocabulary-free representations remain poorly understood for Arabic MT. In particular, it is unclear how these approaches compare in translation performance under MSA and dialect-specific supervision (**RQ1**), how they respond to varying amounts of MSA pretraining data (**RQ2**), how robust they are to orthographic variation (**RQ3**), and what computational trade-offs they exhibit during training and inference (**RQ4**).

To answer these questions, we conduct the first systematic comparison of subword, byte-level, and pixel-based representations for Arabic-to-English MT. We evaluate these approaches across 35 evaluation sets spanning MSA and 18 dialects from six benchmarks. Our results reveal trade-offs across representations. Subword models perform best in lower-data settings, but the gap narrows with more MSA pretraining data, with all approaches reaching comparable performance at full scale. Dialect-specific fine-tuning further reduces these differences, even with limited MSA pretraining. Beyond translation performance, pixel-based models are the most robust to diacritized and dotless Arabic and provide the highest training and inference throughput. Our contributions are as follows:

[1]https://github.com/mbzuai-nlp/arapixel-mt

Proceedings of The Fourth Arabic Natural Language Processing Conference (ArabicNLP 2026), pages 263–288
October 28-29, 2026 ©2026 Association for Computational Linguistics

1. Across 35 evaluation sets spanning MSA and 18 dialects, we find comparable translation performance across representations with sufficient supervision.
2. We study the effect of MSA pretraining data size, finding that representation differences diminish with more data and DA fine-tuning.
3. We demonstrate the robustness of pixel-based models to diacritized and dotless Arabic.
4. We identify efficiency trade-offs, with pixel-based models providing the highest training and inference throughput.

2 Background and Related Work

2.1 Text Representations

Vocabulary-Based Representations Modern NLP systems represent text using fixed vocabularies learned through subword tokenization algorithms such as Byte-Pair Encoding (BPE) (Sennrich et al., 2016), WordPiece (Devlin et al., 2019), and SentencePiece (Kudo and Richardson, 2018). These methods learn tokenizations from corpus frequency statistics rather than morphological structure. Consequently, they may overtokenize morphologically complex languages, degrading performance. Prior work has addressed these limitations through vocabulary expansion, morphology-aware tokenization, and token replacement (Wang et al., 2020; Chau et al., 2020; Minixhofer et al., 2022; Kajiura et al., 2023; Dobler and de Melo, 2023; Cui et al., 2024; Bari et al., 2025; Chizhov et al., 2024; Lian et al., 2025; Jabbar, 2024; Asgari et al., 2025; Hamdan et al., 2025). Nevertheless, these approaches remain constrained by fixed vocabularies.

Vocabulary-Free Representations Vocabulary-free approaches eliminate the need for a predefined token inventory, primarily through byte-level and pixel-level representations. Byte-level models represent text as sequences of raw bytes over a fixed 256-symbol alphabet (Xue et al., 2022; Yu et al., 2023; Kallini et al., 2025; Pagnoni et al., 2025), providing complete vocabulary coverage across languages and scripts. However, because Unicode characters may span multiple bytes, byte-level models produce substantially longer sequences, particularly for non-Latin scripts such as Arabic (Arnett et al., 2024; Unicode Consortium, 2025). Pixel-based approaches instead process visually rendered text, and have been shown to be robust to orthographic variation (Rust et al., 2023), effective for multilingual-to-English MT including MSA (Salesky et al., 2021, 2023), and competitive on multilingual language understanding benchmarks (Kesen et al., 2025). Most relevantly, Lotz et al. (2025) showed that pixel-based encoding outperforms both vocabulary expansion and byte-level representations for MT in several non-Latin scripts, while enabling positive transfer across languages sharing a writing system.

Motivated by these findings, we conduct the first systematic comparison of vocabulary-based and vocabulary-free representations for Arabic MT across MSA and DA varieties.

2.2 Arabic Linguistic Challenges

Morphological Richness and Orthographic Ambiguity Arabic has a templatic and concatenative morphological system that inflects for many features while attaching clitics such as conjunctions, prepositions, and pronouns (Habash, 2010). This results in a large number of surface forms, many of which occur infrequently (Gerz et al., 2018; Botev et al., 2022). Moreover, Arabic is orthographically ambiguous because it uses optional diacritics to specify short vowels, which are typically omitted.

Dialectal Variation and Orthographic Noise Arabic is diglossic (Ferguson, 1959), with MSA coexisting alongside numerous dialectal varieties that differ in phonology, morphology, syntax, and lexicon. Unlike MSA, which has a standardized orthography, dialects are primarily spoken and lack writing conventions, leading to substantial spelling variation (Habash et al., 2018). Beyond dialectal variation, Arabic text exhibits substantial orthographic variation, including MSA spelling errors (Zaghouani et al., 2014), romanized transliteration (Darwish, 2014; Bies et al., 2014), and more recently, dotless Arabic text (Rom and Bar, 2021).

Together, these unique challenges make Arabic a particularly informative case study for understanding how different text representations handle morphological complexity and orthographic variation.

2.3 Arabic Machine Translation

Arabic MT has historically focused on MSA, with growing interest in DA in recent years. This has resulted in numerous parallel corpora and benchmarks, covering both MSA and a wide range of DA varieties (Zbib et al., 2012; Sajjad et al., 2013; Salloum and Habash, 2013; Bouamor et al., 2014;

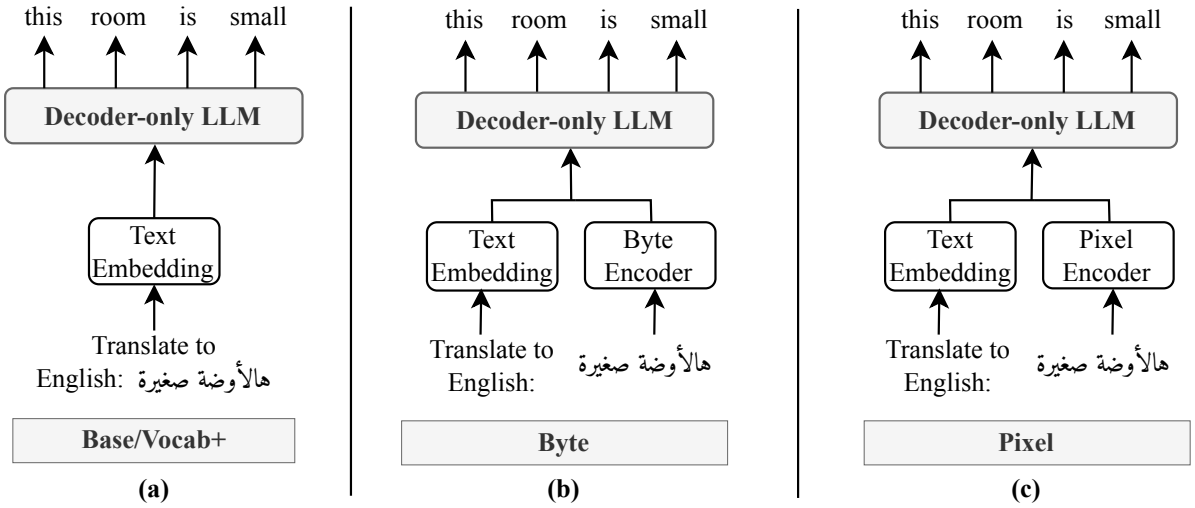


Figure 1: Illustration of the four text representation variants: Base (a), Vocab+ (a), Byte (b), and Pixel (c). The subword models (Base and Vocab+) encode the prompt and source text through the text embedding layer, whereas the byte and pixel models encode the source text using dedicated fallback encoders and combine these representations with prompt embeddings before decoding.

Elmahdy et al., 2014; Meftouh et al., 2015; Habash et al., 2017; Bouamor et al., 2018; Guzmán et al., 2019; Sajjad et al., 2020; Goyal et al., 2022; Abdul-Mageed et al., 2024; Robinson et al., 2025; Mekki et al., 2026, *inter alia*).

Modern Arabic neural MT systems predominantly rely on subword tokenization (Sajjad et al., 2020; Nagoudi et al., 2022; Kadaoui et al., 2023; Robinson et al., 2025; Jon et al., 2026). Despite the central role of text representations in these systems, relatively little work has examined their impact on Arabic MT performance. Oudah et al. (2019) and Gaser et al. (2023) investigated morphology-aware tokenization for statistical and neural MT. More recently, Salesky et al. (2021, 2023) demonstrated the effectiveness of pixel-based representations for Arabic MT within multilingual settings. However, the relative merits of vocabulary-based and vocabulary-free representations remain poorly understood, and no systematic comparison of subword, byte-level, and pixel-based approaches has been conducted for Arabic MT. Our work addresses this gap through a controlled comparison across a diverse collection of MSA and DA benchmarks.

3 Methodology

In this comparative study of Arabic-to-English MT, we fix the translation backbone to a pretrained decoder-only English language model and vary how the Arabic source text is represented and encoded. We compare two paradigms: vocabulary-based and vocabulary-free representations. The vocabulary-based paradigm includes the default subword tokenizer of the backbone model (**Base**) and a vocabulary-expanded variant (**Vocab+**), while the vocabulary-free paradigm comprises byte-level (**Byte**) and pixel-based (**Pixel**) encodings implemented through a learned hypernetwork.

Figure 1 illustrates the four representation variants considered in this work. The following input template is used across all approaches:

```
Translate to English. Arabic:
⟨source⟩ English: ⟨target⟩
```

3.1 Vocabulary-Based Representations

Our baseline (**Base**) uses the language model's original tokenizer and embedding layer. The translation prompt, Arabic source sentence, and English target are represented using the model's native subword vocabulary and processed autoregressively by the decoder-only language model.

As a stronger vocabulary-based baseline, we experiment with a standard vocabulary expansion (**Vocab+**), which augments the original vocabulary with Arabic-specific subword units. We train a new BPE tokenizer from scratch on the Arabic source side of the training data using a target vocabulary size of 32,000 tokens. Tokens that are not already present in the original tokenizer and contain at least one non-ASCII character are added to the base vocabulary, yielding an expanded vocabulary $\mathcal{V}^{+} = \mathcal{V}_{\text{base}} \cup \mathcal{V}_{\text{new}}$.

3.2 Vocabulary-Free Encoding via Hypernetwork

Our two vocabulary-free approaches, byte- (**Byte**) and pixel-level (**Pixel**) encodings, follow the hypernetwork paradigm of Lotz et al. (2025), in which the Arabic source text bypasses the language model's standard embedding layer entirely. Instead, the source text is first segmented into words by whitespace-and-punctuation tokenization, and each word is independently encoded by a dedicated encoder into a single contextualization-free embedding vector. The resulting sequence of word embeddings is then concatenated with the standard subword token embeddings of the translation prompt and English target. The full input sequence is passed as input to the decoder-only language model, which performs cross-word contextualization and autoregressively generates the English translation.

Formally, both vocabulary-free encoders share the same interface: given a source sentence, they produce a sequence of word-level representations $\mathbf{y}_i \in \mathbb{R}^{d_{\text{enc}}}$. A projection multi-layer perceptron (MLP) maps each $\mathbf{y}_i$ to the hidden dimension d of the language model. The projected embeddings are directly substituted for the source-side subword embeddings, requiring no architectural modification to the base language model. Following Lotz et al. (2025), we do not add any special tokens at the boundary between text handled by the language model's standard embedding layer and text handled by the hypernetwork encoder.

3.3 Pixel-Based Encoding

Arabic Text Rendering A key design decision is the rendering of Arabic source words into pixel patches. Arabic presents several considerations that distinguish its rendering from Latin-script languages: right-to-left directionality, mandatory character joining and context-dependent glyph shaping (initial, medial, final, and isolated forms), optional diacritics, and the presence of ligatures. Character shaping is handled automatically by the Pango rendering engine, which produces contextually correct joined glyph forms.

Each word is rendered independently onto a canvas of $H = 24$ pixels and $W = 384$ using the GoNotoCurrent font at size 10pt (approximately 16.67px at 120 DPI) on a plain white background, with NotoSans and NotoSansArabic as fallback fonts. Images are rendered in grayscale and divided into 24×24-pixel bigram patches with one character-overlap, following Lotz et al. (2025). Words exceeding the canvas width are truncated to 384 pixels, while shorter words are padded to produce the fixed canvas size.

Encoder Architecture The pixel encoder is a Vision Transformer (ViT; Dosovitskiy et al. 2021) adapted for text, following Rust et al. (2023). Each rendered word image is divided into $N = 16$ patches, represented as $\mathbf{x} \in \mathbb{R}^{N \times (P^2 \cdot C)}$, where $P = 24$ is the patch size and $C = 3$ is the number of image channels. The patches are then linearly projected into embeddings:

$$\mathbf{z} = \mathbf{x}\mathbf{E} + \mathbf{E}_{\text{pos}},$$

where $\mathbf{E} \in \mathbb{R}^{(P^2 \cdot C) \times d}$ is a convolutional projection and $\mathbf{E}_{\text{pos}} \in \mathbb{R}^{N \times d}$ are 1D sinusoidal positional embeddings, reset at each word boundary. The patch embeddings are processed by a stack of transformer layers to build word representations.

Each word is encoded independently, producing a non-contextualized representation at the word level. To process multiple words efficiently in a forward pass, the batch of sentences is flattened into a single dimension, treating words across sentences as items in one batch. As a result, self-attention operates only over patches belonging to the same word, without requiring an attention mask.

The patch encodings for each word are aggregated via mean pooling to yield a single word-level representation $\mathbf{y}_i \in \mathbb{R}^{d_{\text{enc}}}$, which is then passed to the projection MLP described above. This produces a single embedding per word that is subsequently passed to the language model decoder.

3.4 Byte-Level Encoding

The byte-level encoder follows the same hypernetwork architecture as the pixel encoder, replacing rendered images with raw UTF-8 byte sequences. Each Arabic word is encoded as a sequence of up to 32 bytes (corresponding to up to 16 Arabic characters, since standard UTF-8 encodes each Arabic character as two bytes); bytes beyond this limit are truncated. Padding is handled with a dedicated pad token, giving a vocabulary of 257 symbols.

The encoder architecture mirrors the pixel encoder: byte token embeddings are augmented with 1D sinusoidal positional embeddings, then processed by a transformer-based encoder. The token representations are aggregated via masked mean pooling to produce a word-level representation, $\mathbf{y}_i \in \mathbb{R}^{d_{\text{enc}}}$, followed by the projection MLP.

Dataset	Variety	Train	Dev	Test
Alexandria	DA + MSA	66K	12K	14K
APT	DA	172K	2K	2K
Bible	DA + MSA	-	2K	2K
FLORES	DA + MSA	-	9K	9K
MADAR	DA + MSA	54K	6K	12K
OPUS	MSA	13M	-	-
QAraC	DA + MSA	-	7K	7K
QED	MSA	376K	-	-
TED	MSA	174K	-	-
UN	MSA	10M	-	-
Total		24M	38K	47K

Table 1: Corpus statistics of MT datasets.

4 Experimental Setup

4.1 Data

We use a diverse collection of Arabic-English MT datasets spanning both MSA and DA. Our evaluation suite comprises six benchmarks: Alexandria (Mekki et al., 2026), the Arabic Dialect/English Parallel Text (APT) (Zbib et al., 2012), the Bible, FLORES (Goyal et al., 2022), MADAR (Bouamor et al., 2018), and QAraC (Elmahdy et al., 2014). These benchmarks cover MSA and 18 dialectal varieties from multiple countries and regions of the Arab world. Additionally, for MSA pretraining (§4.3), we use large-scale MSA parallel corpora from OPUS (Lison and Tiedemann, 2016), QED (Guzman et al., 2013), TED (Cettolo, 2016), and the UN corpus (Ziemski et al., 2016).

All Arabic text is whitespace-and-punctuation tokenized, diacritics are removed, and Alif, Ya, and Ta Marbuta normalization is applied using CAMeL Tools (Obeid et al., 2020). We further convert Indic numerals to their Arabic counterparts. For computational efficiency, we retain only MSA pretraining sentence pairs whose Arabic source contains at most 32 words. For APT, Bible, and QAraC, we use the training (Train), development (Dev), and testing (Test) splits created by Sajjad et al. (2020); for all other datasets, we use their official publicly available splits.

Overall, the resulting collection contains 24M, 38K, and 47K sentence pairs for Train, Dev, and Test, respectively. Corpus statistics are summarized in Table 1; a detailed dialectal breakdown is provided in Table 9 (Appendix D).

4.2 Implementation Details

All systems described in §3 are built on the same decoder-only language model backbone, SmolLM2-360M-Instruct (Allal et al., 2025), whose word embedding dimension is $d = 960$. For a controlled comparison between vocabulary-free representations, the **Byte** and **Pixel** encoders share the same architecture configuration. Both use hidden size $d_{\text{enc}} = 1{,}536$, four transformer layers, 16 attention heads, and feed-forward dimension 6,144. The projection MLP consists of two layers with an intermediate dimension of 3,072. Model parameter counts are provided in Table 8 (Appendix A).

4.3 Training Setup

MSA Pretraining To compare text representations under MSA supervision (**RQ1**), we pretrain all models on the same collection of MSA-English parallel corpora: OPUS, QED, TED, and UN. **Base** is trained end-to-end for two epochs. For **Vocab+**, **Byte**, and **Pixel**, we first perform one alignment epoch, followed by two epochs of end-to-end training. For **Vocab+**, the alignment epoch freezes the original model weights and updates only the newly added embedding rows. For **Byte** and **Pixel**, we freeze the language-model backbone and train only the newly initialized encoder and projection MLP. We use a maximum input length of 256 subword tokens for **Base** and **Vocab+**, and 32 words for **Byte** and **Pixel**, covering all retained MSA training examples without truncation.

DA Fine-Tuning We further fine-tune the MSA-pretrained models on the training sets of Alexandria, APT, and MADAR to evaluate them under dialect-specific supervision (**RQ1**). We consider four settings: DA-only and mixed DA+MSA fine-tuning with DA:MSA ratios of 1:1, 1:2, and 1:4. Mixing MSA with DA aims to mitigate overfitting to the limited DA data while preserving the broader knowledge acquired during MSA pretraining. For the mixed settings, MSA examples are sampled from both the original MSA pretraining corpora and the MSA portions of Alexandria and MADAR. All models are fine-tuned for one epoch, with maximum input lengths of 256 subword tokens for **Base** and **Vocab+**, and 64 words for **Byte** and **Pixel**, covering all fine-tuning examples without truncation.

Data Scaling To study the effect of MSA pretraining data size (**RQ2**), we additionally pretrain all models on 5%, 10%, 25%, and 50% of the

MSA data, alongside the full-data (100%) setting described above. We then fine-tune each MSA-pretrained model on the DA+MSA$_{1:1}$ setting, using a balanced mixture that incorporates MSA supervision without outweighing the DA data. We retain the full DA training data at each scale, while drawing the MSA examples from the corresponding MSA pretraining subset. This allows us to examine how the MSA pretraining data size affects each representation both before and after DA fine-tuning.

Optimization Details All experiments use four A100 GPUs and a random seed of 42. We use a cosine learning rate schedule with 10% linear warmup and a peak learning rate of 3×10^{-4}. For MSA pretraining, per-device microbatch sizes are 16, 16, 32, and 64, with gradient accumulation steps of 8, 8, 4, and 2 for **Base**, **Vocab+**, **Byte**, and **Pixel**, respectively. For DA fine-tuning, the corresponding microbatch sizes are 8, 8, 8, and 16, with accumulation steps of 4, 4, 4, and 2. At inference, all models use a batch size of 32.

4.4 Evaluation Setup

We use BLEU (Papineni et al., 2002) as our primary metric and additionally report chrF++ (Popović, 2015, 2017) and COMET (Rei et al., 2020). BLEU and chrF++ are computed using SacreBLEU (Post, 2018), while COMET is computed using the official implementation.

Robustness In addition to evaluating models on the original Arabic input, we assess robustness to orthographic variation using diacritized and dotless versions of the evaluation sets (**RQ3**). Diacritized text is generated automatically using the CAMeL Tools (Obeid et al., 2020) BERT morphological disambiguator (Inoue et al., 2022). For MSA, we use the SAMA database (Graff et al., 2009), while for Egyptian and all other dialectal varieties we use CALIMA (Habash et al., 2012). Although automatically generated analyzers are available for Gulf (Khalifa et al., 2020) and Levantine (Eskander et al., 2013), they tend to be lower quality compared to CALIMA. Dotless text is generated by removing all letter dots.

Efficiency To assess computational efficiency (**RQ4**), we measure training and inference throughput in sentences per second, along with peak GPU memory usage in GiB. For training, **Pixel** inputs are rendered and cached beforehand, so the reported throughput excludes the one-time rendering cost. At inference, **Pixel** inputs are rendered online; we therefore report both end-to-end throughput, including preprocessing (rendering for **Pixel**), and preprocessing throughput separately. Training efficiency is evaluated on four A100 GPUs over 30 steps (15,360 examples), while inference efficiency is evaluated on a single A100 GPU using 3,200 Dev examples. Each experiment is repeated three times, and we report the mean and standard deviation across runs.

4.5 LLM Baselines

We compare against LLMs as a strong upper-bound baseline. Our goal is not to outperform these models, but rather to contextualize the performance of different text representation paradigms relative to contemporary general-purpose LLMs. Specifically, we evaluate two commercial models: GPT-5.5 (Singh et al., 2026) and Gemini-3.1-Flash (Team et al., 2025), as well as two Arabic-centric models: Jais-2-70B-Chat (Anwar et al., 2026) and Fanar-2-27B-Instruct (TEAM et al., 2026). For benchmarks with available training data, we use 5-shot prompting, where examples are sampled uniformly at random from the corresponding dialect of the benchmark. For benchmarks without training splits, we use zero-shot prompting. Unlike our trained models, which operate on normalized Arabic text, we provide the input Arabic text sentences to the LLMs without Alif/Ya/Ta-Marbuta normalization. The prompt we use is provided in Appendix B.

5 Results

5.1 RQ1: Effect of Text Representation

MSA Pretraining Table 2 presents Dev results after MSA pretraining. The four representations achieve comparable performance, with **Base** and **Pixel** both reaching 20.0 average BLEU, followed by **Vocab+** (19.8) and **Byte** (19.3). Paired bootstrap resampling (Koehn, 2004) shows no significant difference between **Base** and **Pixel**, while their differences with **Vocab+** and **Byte** are statistically significant. Across benchmarks, **Base** performs best on Alexandria, APT, Bible, MADAR, and QAraC, while **Pixel** performs best on FLORES. Detailed results across dialects and metrics are provided in Table 10 (Appendix E.1). These findings indicate that, with sufficient MSA pretraining, the representation approaches achieve broadly similar translation performance despite small but statistically significant differences among some models.

Corpus	Base	Vocab+	Byte	Pixel	GPT-5.5	Gemini	Jais	Fanar
Alexandria	**16.4**	**16.4**	16.0	**16.4**	43.1	43.3	42.1	39.1
APT	**10.0**	9.0	9.1	9.3	20.4	19.9	17.3	16.9
Bible	**10.6**	10.3	9.2	10.3	24.9	30.1	25.3	26.2
FLORES	29.9	29.5	29.1	**30.3**	42.3	42.0	44.1	38.5
MADAR	**23.2**	22.9	22.5	22.9	52.0	51.9	71.7	43.1
QAraC	**9.8**	9.4	9.6	9.3	17.9	19.3	18.9	15.7
Overall	**20.0**	19.8	19.3	**20.0**	40.2	40.8	43.8	36.0

Table 2: Macro-average BLEU scores on the Dev sets. Bold indicates the best MSA-pretrained model; underline indicates the best LLM for each dataset.

Data Mixture	Base	Vocab+	Byte	Pixel
MSA	**20.0**	19.8	19.3	**20.0**
+DA	**30.9**	30.1	30.6	**30.9**
+DA+MSA$_{1:1}$	31.2	30.6	30.9	**31.3**
+DA+MSA$_{1:2}$	**31.3**	30.7	30.8	31.2
+DA+MSA$_{1:4}$	**31.2**	30.6	30.7	**31.2**

Table 3: Macro-average BLEU scores on the Dev sets. Bold denotes the best-performing model within each data mixture; underlining denotes the best-performing data mixture for each model.

In comparison, the LLMs substantially outperform the MSA-pretrained MT systems. Among the commercial models, Gemini slightly outperforms GPT-5.5, while for the Arabic-centric models, Jais outperforms Fanar. Although Jais achieves the highest average BLEU score overall, this is largely driven by its exceptionally strong performance on MADAR. A detailed breakdown of LLM performance across dialects and evaluation metrics is provided in Table 11 (Appendix E.2).

DA Fine-Tuning Table 3 reports Dev results after fine-tuning the MSA-pretrained models on DA data. All models benefit substantially from dialect-specific supervision, with average BLEU increasing by more than 10 points over MSA-only pretraining. The models remain closely matched across fine-tuning settings, with **Base** and **Pixel** achieving the highest scores. Detailed results in Table 12 (Appendix E.3) show similar trends across datasets and metrics. Consistent with the MSA pretraining results, DA supervision does not substantially differentiate the representation approaches, with all models exhibiting similar gains from fine-tuning.

Adding MSA data during fine-tuning further improves all models over DA-only training, although the gains are modest. The DA+MSA$_{1:1}$ mixture provides consistent improvements across representations, while increasing the proportion of MSA to 1:2 or 1:4 yields no statistically significant gains. This suggests that a balanced mixture is sufficient to capture the benefits of incorporating MSA during DA fine-tuning.

Test Results Table 4 presents Test results for the MSA-pretrained models and the DA+MSA$_{1:1}$-fine-tuned models. Among the MSA-pretrained models, **Base** achieves the highest average BLEU (19.5), followed by **Vocab+**, **Pixel**, and **Byte**. Unlike on the Dev sets, the differences between **Base** and the other three models are statistically significant. After DA+MSA$_{1:1}$ fine-tuning, the models remain closely matched, with **Base** achieving 31.1 BLEU and **Pixel** 30.9; their difference is not statistically significant, while their differences with **Vocab+/Byte** are. Among the LLMs, Gemini achieves the highest average BLEU, outperforming Jais. Detailed results across dialects and evaluation metrics are provided in Tables 13–15 (Appendix F).

5.2 RQ2: Data Scaling

Figure 2 shows how the models perform with varying amounts of MSA pretraining data, before and after DA+MSA$_{1:1}$ fine-tuning. Under MSA pretraining, **Base** consistently performs best with less data, with the largest advantage at 5%. As the amount of MSA data increases, the performance gaps among representations narrow, with all models achieving comparable performance on the full dataset. This suggests that vocabulary-free representations require more pretraining data to match standard subword tokenization.

DA+MSA$_{1:1}$ fine-tuning substantially improves performance across all data sizes while reducing differences among representations. With only 5%

Corpus	MSA Base	MSA Vocab+	MSA Byte	MSA Pixel	DA+MSA$_{1:1}$ Base	DA+MSA$_{1:1}$ Vocab+	DA+MSA$_{1:1}$ Byte	DA+MSA$_{1:1}$ Pixel	Gemini	Jais
Alexandria	**16.2**	16.1	15.7	15.9	**35.0**	34.4	34.5	34.8	43.0	41.8
APT	**9.1**	8.9	9.0	**9.1**	**20.4**	19.5	20.3	20.3	20.2	22.6
Bible	**10.6**	10.3	9.6	10.0	**12.5**	12.1	11.7	12.0	29.5	24.1
FLORES	**29.0**	28.8	28.4	28.8	31.3	30.4	30.7	**31.4**	41.0	39.5
MADAR	**23.4**	22.9	22.8	23.0	40.6	40.0	**40.8**	40.7	50.0	49.6
QAraC	**10.0**	9.5	**10.0**	**10.0**	**16.0**	15.1	15.7	15.5	20.0	19.0
Overall	**19.5**	19.3	18.9	19.2	**31.1**	30.4	30.6	30.9	40.2	38.8

Table 4: Macro-average BLEU scores on the Test sets. Bold indicates the best MSA-pretrained and DA+MSA$_{1:1}$-fine-tuned models; underline indicates the best LLM for each dataset.

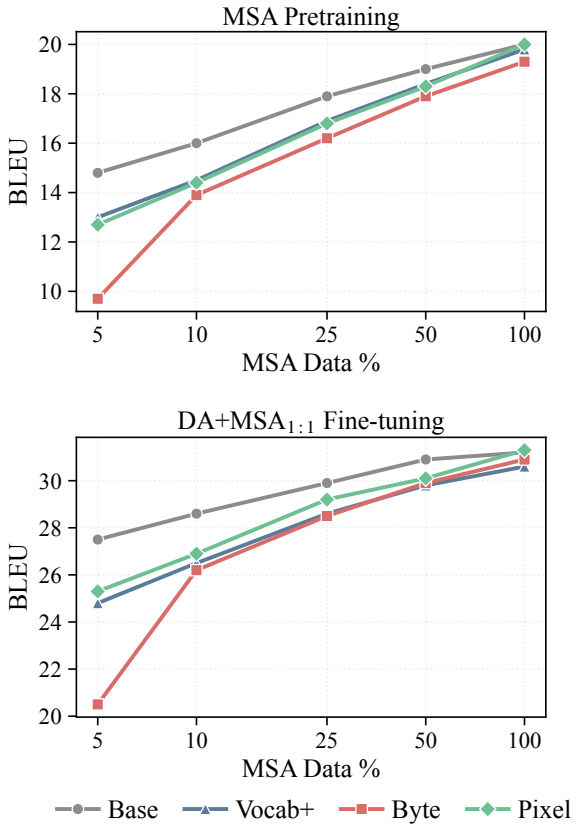


Figure 2: Effect of MSA pretraining data size on MT performance before and after DA+MSA$_{1:1}$ fine-tuning.

of the MSA data, **Vocab+** and **Pixel** already close much of the gap with **Base**, while **Byte** catches up with additional pretraining data. By 100%, all four models achieve comparable performance, indicating that dialect-specific supervision reduces the effect of limited MSA pretraining data on the choice of text representation. This highlights the importance of considering data availability when comparing text representations.

5.3 RQ3: Orthographic Robustness

Table 5 reports robustness results on the Test sets using the DA+MSA$_{1:1}$-fine-tuned models. All models degrade relative to the original inputs, but the magnitude of degradation varies substantially.

Among the models, **Pixel** is considerably more robust to both variations. On diacritized inputs, **Pixel** loses 7.5 BLEU on average, compared with 25.6–29.5 BLEU for the other representations, and achieves the highest BLEU on all six benchmarks. The advantage persists on dotless Arabic, where **Pixel** loses 17.6 BLEU compared with 27.8–28.9 for the other models. This aligns with prior work (Salesky et al., 2021), demonstrating the robustness of pixel-based representations to orthographic variation unseen during training. Such variations have less impact on pixel representations, which can recognize the base character form, unlike byte- and token-based representations that rely on exact character sequences.

LLMs are considerably more robust to diacritized Arabic, with Gemini and Jais losing only 1.8 and 2.9 BLEU on average, respectively. Their behavior differs sharply on dotless inputs, however: Gemini loses 7.1 BLEU, while Jais drops by 28.0 BLEU and falls below **Pixel**. Appendix G presents detailed robustness results across dialects.

5.4 RQ4: Computational Efficiency

As shown in Table 6, **Pixel** provides substantially higher training throughput than the other representation methods, while using the least memory at a comparable batch size (32). This efficiency stems from encoding each source word as a single embedding (§3.3), reducing the sequence length and associated decoder computation and memory. The lower memory requirements also allow a larger batch size (64), further increasing throughput.

Corpus	Base	Vocab+	Byte	Pixel	Gemini	Jais
Diacritized Input						
Alexandria	2.4 (↓32.6)	1.1 (↓33.3)	6.8 (↓27.7)	**26.9** (↓**7.9**)	41.9 (↓1.1)	39.3 (↓2.4)
APT	0.8 (↓19.6)	0.5 (↓19.0)	3.0 (↓17.2)	**15.1** (↓**5.2**)	8.4 (↓11.9)	18.8 (↓3.8)
Bible	0.7 (↓11.7)	0.6 (↓11.5)	1.8 (↓9.9)	**8.8** (↓**3.3**)	28.6 (↓0.9)	22.3 (↓1.9)
FLORES	0.9 (↓30.4)	0.5 (↓29.9)	2.2 (↓28.4)	**22.3** (↓**9.1**)	39.0 (↓2.0)	35.3 (↓4.3)
MADAR	2.7 (↓37.9)	1.2 (↓38.8)	8.4 (↓32.4)	**31.9** (↓**8.8**)	49.4 (↓0.5)	46.7 (↓2.9)
QAraC	0.7 (↓15.3)	0.3 (↓14.8)	3.2 (↓12.5)	**12.8** (↓**2.7**)	20.2 (↑0.2)	19.3 (↑0.3)
Overall	1.7 (↓29.3)	0.9 (↓29.5)	5.0 (↓25.6)	**23.4** (↓**7.5**)	38.4 (↓1.8)	35.8 (↓2.9)
Dotless Input						
Alexandria	4.5 (↓30.6)	2.1 (↓32.3)	3.5 (↓31.0)	**16.2** (↓**18.6**)	36.8 (↓6.2)	13.2 (↓28.5)
APT	2.1 (↓18.2)	0.9 (↓18.7)	1.8 (↓18.4)	**9.4** (↓**10.9**)	7.2 (↓13.0)	5.0 (↓17.6)
Bible	1.5 (↓11.0)	0.8 (↓11.3)	1.1 (↓10.6)	**5.2** (↓**6.8**)	27.2 (↓2.3)	9.3 (↓14.9)
FLORES	1.7 (↓29.6)	0.9 (↓29.5)	1.3 (↓29.4)	**12.2** (↓**19.1**)	31.7 (↓9.3)	8.0 (↓31.5)
MADAR	4.8 (↓35.8)	2.0 (↓38.1)	3.3 (↓37.5)	**16.4** (↓**24.3**)	42.5 (↓7.5)	13.6 (↓36.0)
QAraC	2.3 (↓13.8)	0.9 (↓14.1)	2.0 (↓13.7)	**7.6** (↓**8.0**)	15.5 (↓4.5)	3.5 (↓15.5)
Overall	3.3 (↓27.8)	1.5 (↓28.9)	2.5 (↓28.1)	**13.3** (↓**17.6**)	33.1 (↓7.1)	10.7 (↓28.0)

Table 5: Robustness results on the Test sets with diacritized and dotless Arabic inputs. Values in parentheses denote the BLEU change relative to the corresponding original input (↓: decrease; ↑: increase). Bold indicates the best-performing model; underline indicates the best-performing LLM.

Model	Batch	Sent./s ↑	Mem. ↓
Base	16	250.2±0.8	24.9
Vocab+	16	230.1±0.7	27.2
Byte	32	424.3±1.5	27.8
Pixel	32	666.2±5.9	**22.0**
Pixel	**64**	**726.8**±0.9	33.0

Table 6: Training efficiency. Sent./s denotes sentences per second (mean±standard deviation); Mem. denotes peak reserved GPU memory in GiB.

		Sent./s ↑		
Model	Batch	E2E	Prep.	Mem. ↓
Base	32	11.1±0.2	11,808±0.0	14.9
Vocab+	32	11.9±0.1	**12,800**±0.0	11.1
Byte	32	12.3±0.1	6,823±1.6	9.2
Pixel	32	12.4±0.1	1,529±0.2	**5.6**
Pixel	**64**	**15.6**±0.1	1,533±0.6	9.5

Table 7: Inference efficiency. Sent./s denotes sentences per second; E2E and Prep. denote end-to-end and preprocessing throughput (mean±standard deviation); Mem. denotes peak reserved GPU memory in GiB.

At inference (Table 7), despite the cost of online rendering, **Pixel** achieves the highest end-to-end throughput at a comparable batch size while using the least GPU memory. This allows **Pixel** to double the batch size to 64, further increasing throughput. Moreover, for deployment, all possible Arabic character bigrams (§3.3) can be pre-rendered and cached, eliminating online rendering overhead.

6 Conclusion and Future Work

We presented the first systematic comparison of vocabulary-based and vocabulary-free text representations for Arabic-English machine translation. Across 35 evaluation sets spanning MSA and 18 dialects, we find that the representations achieve broadly comparable translation performance with sufficient supervision. Standard subword representations perform best with limited MSA pretraining data, but differences diminish with more data and dialect-specific fine-tuning. Pixel-based representations are substantially more robust to unseen orthographic variation, including diacritized and dotless Arabic, while providing the highest training and end-to-end inference throughput. Our findings reveal trade-offs between translation performance, data requirements, robustness, and computational efficiency across Arabic text representations.

In future work, we plan to extend this comparison to other Arabic NLP tasks and investigate Arabic-specific pretraining strategies for pixel-based representations, particularly given their robustness and computational efficiency.

Limitations

Our study has several limitations. First, all experiments use a single decoder-only language model backbone (SmolLM2-360M-Instruct). Although this enables a controlled comparison, the relative performance of the different representation paradigms may differ for larger models or alternative architectures. Second, we evaluate text representations only on Arabic-English machine translation. While our benchmark covers MSA and 18 dialectal varieties, our findings may not generalize to other Arabic NLP tasks. Finally, our robustness evaluation is limited to two orthographic perturbations: diacritized and dotless Arabic. Future work should consider additional naturally occurring orthographic variations, such as spelling inconsistencies, informal writing conventions, and OCR noise.

Ethics Statement

We use all datasets and models in accordance with their respective licenses and terms of use. A summary of licenses is provided in Appendix C. While this work does not introduce new datasets containing personally identifiable information, the evaluated machine translation systems may produce inaccurate translations, particularly for Dialectal Arabic, and may reflect biases present in their training data. We therefore recommend appropriate human oversight when deploying these systems in applications where translation errors could have significant consequences. We used AI writing assistance within the scope of "Assistance purely with the language of the paper" described in the ACL Policy on Publication Ethics.

References

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.

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, Joshua Lochner, Caleb Fahlgren, Xuan-Son Nguyen, Clémentine Fourrier, Ben Burtenshaw, Hugo Larcher, Haojun Zhao, Cyril Zakka, Mathieu Morlon, and 3 others. 2025. Smollm2: When smol goes big – data-centric training of a small language model. *Preprint*, arXiv:2502.02737.

Mohamed Anwar, Abed Alhakim Freihat, George Ibrahim, Mostafa Awad, Abdelrahman Sadallah, Gurpreet Gosal, Gokulakrishnan Ramakrishnan, Sarath Chandran, Biswajit Mishra, Rituraj Joshi, Ahmed Frikha, Etienne Goffinet, Abhishek Maiti, Ali El Filali, Sarah AlBarri, Samujjwal Ghosh, Rahul Pal, Parvez Mullah, Awantika Shukla, and 41 others. 2026. Jais 2: A family of arabic-centric open large language models. *Preprint*, arXiv:2608.13580.

Catherine Arnett and Benjamin Bergen. 2025. Why do language models perform worse for morphologically complex languages? In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6607–6623, Abu Dhabi, UAE. Association for Computational Linguistics.

Catherine Arnett, Tyler A. Chang, and Benjamin Bergen. 2024. A bit of a problem: Measurement disparities in dataset sizes across languages. In *Proceedings of the 3rd Annual Meeting of the Special Interest Group on Under-resourced Languages @ LREC-COLING 2024*, pages 1–9, Torino, Italia. ELRA and ICCL.

Ehsaneddin Asgari, Yassine El Kheir, and Mohammad Ali Sadraei Javaheri. 2025. Morphbpe: A morpho-aware tokenizer bridging linguistic complexity for efficient llm training across morphologies. *Preprint*, arXiv:2502.00894.

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*.

Ann Bies, Zhiyi Song, Mohamed Maamouri, Stephen Grimes, Haejoong Lee, Jonathan Wright, Stephanie Strassel, Nizar Habash, Ramy Eskander, and Owen Rambow. 2014. Transliteration of Arabizi into Arabic orthography: Developing a parallel annotated Arabizi-Arabic script SMS/chat corpus. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 93–103, Doha, Qatar. Association for Computational Linguistics.

Georgie Botev, Arya D. McCarthy, Winston Wu, and David Yarowsky. 2022. Deciphering and characterizing out-of-vocabulary words for morphologically rich languages. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 5309–5326, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

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. European Language Resources Association (ELRA).

Mauro Cettolo. 2016. An arabic-hebrew parallel corpus of ted talks. *Preprint*, arXiv:1610.00572.

Ethan C. Chau, Lucy H. Lin, and Noah A. Smith. 2020. Parsing with multilingual BERT, a small corpus, and a small treebank. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1324–1334, Online. Association for Computational Linguistics.

Pavel Chizhov, Catherine Arnett, Elizaveta Korotkova, and Ivan P. Yamshchikov. 2024. BPE gets picky: Efficient vocabulary refinement during tokenizer training. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 16587–16604, Miami, Florida, USA. Association for Computational Linguistics.

Yiming Cui, Ziqing Yang, and Xin Yao. 2024. Efficient and effective text encoding for chinese llama and alpaca. *Preprint*, arXiv:2304.08177.

Kareem Darwish. 2014. Arabizi detection and conversion to Arabic. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 217–224, Doha, Qatar. 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.

Konstantin Dobler and Gerard de Melo. 2023. FOCUS: Effective embedding initialization for monolingual specialization of multilingual models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13440–13454, Singapore. Association for Computational Linguistics.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Mohamed Elmahdy, Mark Hasegawa-Johnson, and Eiman Mustafawi. 2014. Development of a TV broadcasts speech recognition system for qatari Arabic. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3057–3061, Reykjavik, Iceland. European Language Resources Association (ELRA).

Ramy Eskander, Nizar Habash, and Owen Rambow. 2013. Automatic extraction of morphological lexicons from morphologically annotated corpora. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1032–1043, Seattle, Washington, USA. Association for Computational Linguistics.

Charles F Ferguson. 1959. Diglossia. *Word*, 15(2):325–340.

Marwa Gaser, Manuel Mager, Injy Hamed, Nizar Habash, Slim Abdennadher, and Ngoc Thang Vu. 2023. Exploring segmentation approaches for neural machine translation of code-switched Egyptian Arabic-English text. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3523–3538, Dubrovnik, Croatia. Association for Computational Linguistics.

Daniela Gerz, Ivan Vulić, Edoardo Ponti, Jason Naradowsky, Roi Reichart, and Anna Korhonen. 2018. Language modeling for morphologically rich languages: Character-aware modeling for word-level prediction. *Transactions of the Association for Computational Linguistics*, 6:451–465.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The Flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538.

David Graff, Mohamed Maamouri, Basma Bouziri, Sondos Krouna, Seth Kulick, and Tim Buckwalter. 2009. Standard Arabic Morphological Analyzer (SAMA) Version 3.1. Linguistic Data Consortium LDC2009E73.

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. 2019. The FLORES evaluation datasets for low-resource machine translation: Nepali–English and Sinhala–English. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6098–6111, Hong Kong, China. Association for Computational Linguistics.

Francisco Guzman, Hassan Sajjad, Stephan Vogel, and Ahmed Abdelali. 2013. The AMARA corpus: building resources for translating the web's educational content. In *Proceedings of the 10th International Workshop on Spoken Language Translation: Papers*, Heidelberg, Germany.

Nizar Habash, Ramy Eskander, and Abdelati Hawwari. 2012. A morphological analyzer for Egyptian Arabic. In *Proceedings of the Twelfth Meeting of the*

Special Interest Group on Computational Morphology and Phonology, pages 1–9, Montréal, Canada. Association for Computational Linguistics.

Nizar Habash, Salam Khalifa, Fadhl Eryani, 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 International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan.

Nizar Habash, Nasser Zalmout, Dima Taji, Hieu Hoang, and Maverick Alzate. 2017. A parallel corpus for evaluating machine translation between Arabic and European languages. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 235–241, Valencia, Spain. Association for Computational Linguistics.

Nizar Y Habash. 2010. *Introduction to Arabic natural language processing*, volume 3. Morgan & Claypool Publishers.

Nancy Hamdan, Osama Rakan Al Mraikhat, and Fadi A. Zaraket. 2025. R-BPE: Improving BPE-tokenizers with token reuse. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 22940–22948, Suzhou, China. Association for Computational Linguistics.

Go Inoue, Salam Khalifa, and Nizar Habash. 2022. Morphosyntactic tagging with pre-trained language models for Arabic and its dialects. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1708–1719, Dublin, Ireland. Association for Computational Linguistics.

Haris Jabbar. 2024. Morphpiece : A linguistic tokenizer for large language models. *Preprint*, arXiv:2307.07262.

Josef Jon, Rawan Bondok, and Ondřej Bojar. 2026. Current state of LLMs for Arabic dialectal machine translation. In *Proceedings of the 2nd Workshop on NLP for Languages Using Arabic Script*, pages 329–363, Rabat, Morocco. Association for Computational Linguistics.

Karima Kadaoui, Samar M. Magdy, Abdul Waheed, Md Tawkat Islam Khondaker, Ahmed Oumar El-Shangiti, El Moatez Billah Nagoudi, and Muhammad Abdul-Mageed. 2023. TARJAMAT: Evaluation of bard and ChatGPT on machine translation of ten Arabic varieties. In *Proceedings of ArabicNLP 2023*, pages 52–75, Singapore (Hybrid). Association for Computational Linguistics.

Teruno Kajiura, Shiho Takano, Tatsuya Hiraoka, and Kimio Kuramitsu. 2023. Vocabulary replacement in SentencePiece for domain adaptation. In *Proceedings of the 37th Pacific Asia Conference on Language, Information and Computation*, pages 645–652, Hong Kong, China. Association for Computational Linguistics.

Julie Kallini, Shikhar Murty, Christopher D Manning, Christopher Potts, and Róbert Csordás. 2025. Mrt5: Dynamic token merging for efficient byte-level language models. In *The Thirteenth International Conference on Learning Representations*.

Ilker Kesen, Jonas F. Lotz, Ingo Ziegler, Phillip Rust, and Desmond Elliott. 2025. Multilingual pretraining for pixel language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 29594–29611, Suzhou, China. Association for Computational Linguistics.

Salam Khalifa, Nasser Zalmout, and Nizar Habash. 2020. Morphological analysis and disambiguation for Gulf Arabic: The interplay between resources and methods. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3895–3904, Marseille, France. European Language Resources Association.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Haoran Lian, Yizhe Xiong, Jianwei Niu, Shasha Mo, Zhenpeng Su, Zijia Lin, Hui Chen, Jungong Han, and Guiguang Ding. 2025. Scaffold-bpe: Enhancing byte pair encoding for large language models with simple and effective scaffold token removal. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(23):24539–24548.

Davis Liang, Hila Gonen, Yuning Mao, Rui Hou, Naman Goyal, Marjan Ghazvininejad, Luke Zettlemoyer, and Madian Khabsa. 2023. XLM-V: Overcoming the vocabulary bottleneck in multilingual masked language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13142–13152, Singapore. Association for Computational Linguistics.

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).

Jonas F. Lotz, Hendra Setiawan, Stephan Peitz, and Yova Kementchedjhieva. 2025. Overcoming vocabulary constraints with pixel-level fallback. In *Second Conference on Language Modeling*.

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.

Abdellah El Mekki, Samar M. Magdy, Houdaifa Atou, Ruwa AbuHweidi, Baraah Qawasmeh, Omer Nacar, Thikra Al-hibiri, Razan Saadie, Hamzah Alsayadi, Nadia Ghezaiel Hammouda, Alshima Alkhazimi, Aya Hamod, Al-Yas Al-Ghafri, Wesam El-Sayed, Asila Al sharji, Mohamad Ballout, Anas Belfathi, Karim Ghaddar, Serry Sibaee, and 28 others. 2026. Alexandria: A multi-domain dialectal arabic machine translation dataset for culturally inclusive and linguistically diverse llms. *Preprint*, arXiv:2601.13099.

Sabrina J. Mielke, Zaid Alyafeai, Elizabeth Salesky, Colin Raffel, Manan Dey, Matthias Gallé, Arun Raja, Chenglei Si, Wilson Y. Lee, Benoît Sagot, and Samson Tan. 2021. Between words and characters: A brief history of open-vocabulary modeling and tokenization in nlp. *Preprint*, arXiv:2112.10508.

Benjamin Minixhofer, Fabian Paischer, and Navid Rekabsaz. 2022. WECHSEL: Effective initialization of subword embeddings for cross-lingual transfer of monolingual language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3992–4006, Seattle, United States. Association for Computational Linguistics.

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.

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.

Mai Oudah, Amjad Almahairi, and Nizar Habash. 2019. The impact of preprocessing on Arabic-English statistical and neural machine translation. In *Proceedings of Machine Translation Summit XVII: Research Track*, pages 214–221, Dublin, Ireland. European Association for Machine Translation.

Artidoro Pagnoni, Ramakanth Pasunuru, Pedro Rodriguez, John Nguyen, Benjamin Muller, Margaret Li, Chunting Zhou, Lili Yu, Jason E Weston, Luke Zettlemoyer, Gargi Ghosh, Mike Lewis, Ari Holtzman, and Srini Iyer. 2025. Byte latent transformer: Patches scale better than tokens. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9238–9258, Vienna, Austria. Association for Computational Linguistics.

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.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2021. UNKs everywhere: Adapting multilingual language models to new scripts. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10186–10203, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Nathaniel Romney Robinson, Shahd Abdelmoneim, Kelly Marchisio, and Sebastian Ruder. 2025. ALQASIDA: Analyzing LLM quality and accuracy systematically in dialectal Arabic. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 22048–22065, Vienna, Austria. Association for Computational Linguistics.

Aviad Rom and Kfir Bar. 2021. Supporting undotted Arabic with pre-trained language models. In *Proceedings of the 4th International Conference on Natural Language and Speech Processing (ICNLSP 2021)*, pages 89–94, Trento, Italy. Association for Computational Linguistics.

Phillip Rust, Jonas F. Lotz, Emanuele Bugliarello, Elizabeth Salesky, Miryam de Lhoneux, and Desmond Elliott. 2023. Language modelling with pixels. In *The Eleventh International Conference on Learning Representations*.

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.

Hassan Sajjad, Ahmed Abdelali, Nadir Durrani, and Fahim Dalvi. 2020. AraBench: Benchmarking

dialectal Arabic-English machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5094–5107, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Hassan Sajjad, Kareem Darwish, and Yonatan Belinkov. 2013. Translating dialectal Arabic to English. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–6, Sofia, Bulgaria. Association for Computational Linguistics.

Elizabeth Salesky, David Etter, and Matt Post. 2021. Robust open-vocabulary translation from visual text representations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7235–7252, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Elizabeth Salesky, Neha Verma, Philipp Koehn, and Matt Post. 2023. Multilingual pixel representations for translation and effective cross-lingual transfer. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13845–13861, Singapore. Association for Computational Linguistics.

Wael Salloum and Nizar Habash. 2013. Dialectal Arabic to English machine translation: Pivoting through Modern Standard Arabic. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 348–358, Atlanta, Georgia. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, Akshay Nathan, Alan Luo, Alec Helyar, Aleksander Madry, Aleksandr Efremov, Aleksandra Spyra, Alex Baker-Whitcomb, Alex Beutel, Alex Karpenko, and 467 others. 2026. Openai gpt-5 system card. *Preprint*, arXiv:2601.03267.

Ahmed Tawfik, Mahitab Emam, Khaled Essam, Robert Nabil, and Hany Hassan. 2019. Morphology-aware word-segmentation in dialectal Arabic adaptation of neural machine translation. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 11–17, Florence, Italy. Association for Computational Linguistics.

FANAR TEAM, Ummar Abbas, Mohammad Shahmeer Ahmad, Minhaj Ahmad, Abdulaziz Al-Homaid, Anas Al-Nuaimi, Enes Altinisik, Ehsaneddin Asgari, Sanjay Chawla, Shammur Chowdhury, Fahim Dalvi, Kareem Darwish, Nadir Durrani, Mohamed Elfeky, Ahmed Elmagarmid, Mohamed Eltabakh, Asim Ersoy, Masoomali Fatehkia, Mohammed Qusay Hashim, and 18 others. 2026. Fanar 2.0: Arabic generative ai stack. *Preprint*, arXiv:2603.16397.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, and 1 others. 2025. Gemini: A family of highly capable multimodal models. *Preprint*, arXiv:2312.11805.

Unicode Consortium. 2025. The unicode standard, version 17.0. Technical report, Unicode Consortium.

Zihan Wang, Karthikeyan K, Stephen Mayhew, and Dan Roth. 2020. Extending multilingual BERT to low-resource languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2649–2656, Online. Association for Computational Linguistics.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. ByT5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306.

Lili Yu, Daniel Simig, Colin Flaherty, Armen Aghajanyan, Luke Zettlemoyer, and Mike Lewis. 2023. MEGABYTE: Predicting million-byte sequences with multiscale transformers. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Wajdi Zaghouani, Behrang Mohit, Nizar Habash, Ossama Obeid, Nadi Tomeh, Alla Rozovskaya, Noura Farra, Sarah Alkuhlani, and Kemal Oflazer. 2014. Large scale Arabic error annotation: Guidelines and framework. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland. European Language Resources Association (ELRA).

Rabih Zbib, Erika Malchiodi, Jacob Devlin, David Stallard, Spyros Matsoukas, Richard Schwartz, John Makhoul, Omar F. Zaidan, and Chris Callison-Burch. 2012. Machine translation of Arabic dialects. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 49–59, Montréal, Canada. Association for Computational Linguistics.

Michał Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The United Nations parallel corpus v1.0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3530–3534, Portorož, Slovenia. European Language Resources Association (ELRA).

A Models Parameters

Model	Params. (M)
Base	361.8
Vocab+	391.6
Byte	483.3
Pixel	485.5

Table 8: Number of parameters for each model.

B LLM Prompt

Prompt

```
You are an Arabic to English machine
translation system.

Translate the following
{Source Dialect} sentence to
English.

Output only the translation, nothing
else.

Here are some examples:
{Five in-context examples from the
benchmark training set, when
available}
```

C Licenses and Terms of Use

We list below the licenses of the data and models used in this work, all of which are employed in accordance with their intended use.

- Alexandria (Mekki et al., 2026): Creative Commons Attribution Non Commercial 4.0.
- APT (Zbib et al., 2012): LDC License (LDC2012T09).
- Bible: Publicly available parallel Arabic-English translations.
- FLORES (Goyal et al., 2022): Creative Commons Attribution Share Alike 4.0.
- MADAR (Bouamor et al., 2018): CMU and NYUAD Open-Source License.
- OPUS (Lison and Tiedemann, 2016): Collection of parallel corpora distributed under their respective original licenses.
- QAraC (Elmahdy et al., 2014): Creative Commons Attribution 4.0 International.
- QED (Guzman et al., 2013): Freely available for research use.
- TED (Cettolo, 2016): Creative Commons Attribution-NonCommercial-NoDerivatives.
- UN (Ziemski et al., 2016): Freely available for research use.
- SmolLM2-360M-Instruct (Allal et al., 2025): Apache 2.0.
- GPT-5.5 (Singh et al., 2026): Proprietary API model accessed via OpenAI API.
- Gemini-3.1-Flash (Team et al., 2025): Proprietary API model accessed via Google Gemini API.
- Jais-2-70B-Chat (Anwar et al., 2026): Apache 2.0.
- Fanar-2-27B-Instruct (TEAM et al., 2026): Apache 2.0.

D Data Statistics

Corpus	Dialect	Train	Dev	Test
Alexandria	Egyptian	3,108	1,113	1,118
Alexandria	Jordanian	5,501	1,113	1,107
Alexandria	Lebanese	8,906	1,118	1,106
Alexandria	Libyan	-	-	1,109
Alexandria	Mauritanian	2,573	1,110	1,115
Alexandria	Moroccan	5,515	1,114	1,112
Alexandria	Omani	6,280	1,109	1,118
Alexandria	Palestinian	14,933	1,110	1,109
Alexandria	Saudi	8,470	1,110	1,113
Alexandria	Sudanese	-	-	1,106
Alexandria	Syrian	6,071	1,119	1,114
Alexandria	Tunisian	2,034	1,116	1,109
Alexandria	Yemeni	3,089	1,118	1,106
APT	Egyptian	35,935	1,000	1,000
APT	Levantine	135,788	1,000	1,000
Bible	Moroccan	-	600	600
Bible	MSA	-	600	600
Bible	MSA	-	600	600
Bible	Tunisian	-	600	600
FLORES	Egyptian	-	997	1,012
FLORES	Iraqi	-	997	1,012
FLORES	Moroccan	-	997	1,012
FLORES	MSA	-	997	1,012
FLORES	N. Levantine	-	997	1,012
FLORES	Najdi	-	997	1,012
FLORES	S. Levantine	-	997	1,012
FLORES	Tunisian	-	997	1,012
FLORES	Yemeni	-	997	1,012
MADAR	Egyptian	9,000	1,000	2,000
MADAR	Lebanese	9,000	1,000	2,000
MADAR	Moroccan	9,000	1,000	2,000
MADAR	MSA	9,000	1,000	2,000
MADAR	Qatari	9,000	1,000	2,000
MADAR	Tunisian	9,000	1,000	2,000
OPUS	MSA	13,466,770	-	-
QAraC	Qatari	-	6,713	6,713
QED	MSA	376,337	-	-
TED	MSA	173,994	-	-
UN	MSA	10,024,940	-	-
Total		**24,334,244**	**38,336**	**46,663**

Table 9: Dataset statistics across corpora, dialects, and Train/Dev/Test splits. Bible contains two distinct MSA sets, following Sajjad et al. (2020).

E Dev Results

E.1 MSA-Pretrained Models

		BLEU				chrF++				COMET			
Corpus	**Dialect**	**Base**	**Vocab+**	**Byte**	**Pixel**	**Base**	**Vocab+**	**Byte**	**Pixel**	**Base**	**Vocab+**	**Byte**	**Pixel**
Alexandria	Egyptian	**18.9**	18.3	17.9	18.6	**39.9**	38.9	38.4	39.2	**0.72**	**0.72**	0.71	**0.72**
Alexandria	Jordanian	18.8	18.6	**19.2**	19.0	39.6	39.5	**39.9**	**39.9**	0.71	0.71	0.71	**0.72**
Alexandria	Lebanese	**15.2**	15.0	14.2	14.0	**34.9**	34.5	33.6	33.5	**0.67**	0.66	0.65	0.65
Alexandria	Mauritanian	**7.9**	7.6	6.5	7.3	**26.3**	26.1	23.6	24.7	**0.53**	**0.53**	0.51	0.52
Alexandria	Moroccan	9.2	**9.3**	9.2	8.9	**28.1**	27.8	27.5	27.6	**0.56**	**0.56**	0.55	**0.56**
Alexandria	Omani	18.8	**19.6**	19.4	19.4	41.0	**41.4**	41.3	**41.4**	0.71	**0.72**	0.71	0.71
Alexandria	Palestinian	17.0	17.3	17.3	**17.5**	37.8	37.8	37.6	**38.1**	**0.69**	**0.69**	**0.69**	**0.69**
Alexandria	Saudi	22.6	23.0	22.4	**23.1**	44.0	44.2	43.7	**44.7**	**0.75**	**0.75**	**0.75**	**0.75**
Alexandria	Syrian	**21.7**	20.9	20.6	21.6	**41.9**	41.2	41.0	**41.9**	**0.71**	0.70	0.70	**0.71**
Alexandria	Tunisian	**12.7**	12.6	11.8	11.6	**33.0**	32.6	31.4	31.1	**0.61**	**0.61**	0.59	0.59
Alexandria	Yemeni	18.2	18.7	17.9	**18.8**	39.8	39.8	39.1	**40.1**	**0.73**	**0.73**	0.72	**0.73**
APT	Egyptian	**11.1**	10.1	10.3	10.1	**32.8**	32.0	31.8	32.1	**0.65**	0.64	0.64	0.64
APT	Levantine	**9.0**	7.9	8.0	8.6	**30.2**	29.4	28.9	29.7	**0.62**	**0.62**	**0.62**	**0.62**
Bible	Moroccan	5.1	5.2	4.3	**5.3**	23.6	**23.7**	22.7	**23.7**	0.52	0.52	0.51	**0.54**
Bible	MSA	**17.0**	16.5	14.8	16.4	**38.4**	38.2	36.0	37.1	**0.74**	0.73	0.71	0.73
Bible	MSA	**12.8**	12.6	11.5	12.7	**33.5**	32.8	32.1	32.3	**0.68**	0.67	0.66	0.67
Bible	Tunisian	**7.6**	7.0	6.4	6.8	**26.6**	25.5	24.8	24.9	**0.56**	0.54	0.53	0.54
FLORES	Egyptian	25.2	25.0	24.8	**26.2**	51.0	50.5	50.1	**51.2**	0.79	0.79	0.79	**0.80**
FLORES	Iraqi	31.0	31.3	30.5	**31.7**	55.5	55.6	54.5	**55.7**	**0.82**	**0.82**	0.81	**0.82**
FLORES	Moroccan	17.9	17.7	17.3	**18.5**	42.3	41.8	41.0	**42.5**	**0.67**	**0.67**	0.66	**0.67**
FLORES	MSA	38.2	37.3	36.9	**38.3**	**61.2**	60.6	60.0	61.1	**0.86**	0.85	0.85	**0.86**
FLORES	N. Levantine	29.8	29.4	29.1	**30.2**	54.1	53.6	53.1	**54.2**	**0.80**	0.79	0.79	0.79
FLORES	Najdi	**37.3**	36.5	36.2	37.0	**60.6**	59.9	59.3	60.3	**0.85**	**0.85**	**0.85**	**0.85**
FLORES	S. Levantine	32.6	32.2	31.6	**33.3**	56.3	56.1	55.1	**56.8**	0.80	0.80	0.80	**0.81**
FLORES	Tunisian	**23.8**	23.1	23.6	**23.8**	**48.7**	48.1	47.9	48.6	**0.74**	0.73	0.73	**0.74**
FLORES	Yemeni	33.0	32.7	32.0	**33.4**	56.9	56.6	56.0	**57.5**	**0.83**	**0.83**	**0.83**	**0.83**
MADAR	Egyptian	**27.2**	26.2	25.5	26.0	**45.4**	44.1	43.9	44.2	**0.76**	0.74	0.74	0.74
MADAR	Lebanese	16.4	16.1	15.6	**16.5**	34.4	33.9	33.2	**34.6**	**0.64**	0.63	0.63	**0.64**
MADAR	Moroccan	12.4	**12.7**	11.3	12.0	30.5	**30.7**	29.4	30.5	**0.60**	**0.60**	0.58	0.59
MADAR	MSA	44.7	45.0	44.9	**45.1**	61.4	**61.7**	61.5	61.6	0.87	**0.88**	**0.88**	0.87
MADAR	Qatari	**26.9**	26.1	25.9	26.2	**44.4**	43.5	43.5	43.9	**0.75**	0.74	0.73	0.74
MADAR	Tunisian	11.6	11.4	**11.8**	**11.8**	**29.3**	29.1	29.0	29.1	**0.59**	0.58	0.58	0.58
QAraC	Qatari	**9.8**	9.4	9.6	9.3	**32.8**	32.0	32.5	32.1	**0.64**	0.63	0.63	0.63
Average		**20.0**	19.8	19.3	**20.0**	**41.1**	40.7	40.1	40.8	**0.70**	**0.70**	0.69	**0.70**

Table 10: MSA-pretrained models results on the Dev sets across the dialects for each dataset in terms of BLEU, chrF++, and COMET. Best results are in bold.

E.2 LLMs

Corpus	Dialect	BLEU				chrF++				COMET			
		GPT-5.5	Gemini	Jais	Fanar	GPT-5.5	Gemini	Jais	Fanar	GPT-5.5	Gemini	Jais	Fanar
Alexandria	Egyptian	**43.0**	42.5	41.6	39.5	**62.0**	61.6	60.6	59.2	**0.86**	**0.86**	0.85	0.85
Alexandria	Jordanian	42.9	**44.1**	43.1	40.8	62.5	**62.9**	62.0	60.8	**0.87**	0.86	0.86	0.85
Alexandria	Lebanese	43.0	42.5	**44.1**	40.9	61.8	61.7	**62.0**	59.7	**0.86**	0.85	0.85	0.84
Alexandria	Mauritanian	40.8	**41.4**	41.0	37.1	60.0	**60.5**	60.1	56.8	**0.83**	**0.83**	0.82	0.80
Alexandria	Moroccan	34.4	**34.7**	29.4	23.6	54.5	**56.7**	49.8	44.8	0.79	**0.80**	0.75	0.73
Alexandria	Omani	**44.3**	43.1	43.1	39.4	**63.8**	62.9	62.6	60.1	**0.87**	0.86	0.85	0.85
Alexandria	Palestinian	42.4	**42.6**	42.4	40.5	**62.2**	62.1	61.5	60.2	**0.86**	**0.86**	0.85	0.85
Alexandria	Saudi	46.2	**47.6**	46.0	43.6	65.2	**65.8**	65.0	63.1	**0.87**	**0.87**	0.86	0.86
Alexandria	Syrian	53.8	**54.0**	52.3	49.4	**70.0**	69.8	68.7	67.0	**0.88**	0.87	0.87	0.86
Alexandria	Tunisian	44.3	**44.7**	42.3	39.6	**62.2**	62.1	60.4	58.4	**0.85**	**0.85**	0.83	0.83
Alexandria	Yemeni	**39.2**	38.5	37.5	35.5	**59.5**	58.9	58.5	56.4	**0.86**	0.85	0.85	0.84
APT	Egyptian	**24.2**	21.6	20.4	18.6	**50.1**	49.4	47.4	45.6	**0.77**	**0.77**	0.76	0.75
APT	Levantine	16.6	**18.3**	14.2	15.2	45.0	**47.0**	45.7	43.9	0.76	**0.77**	0.75	0.74
Bible	Moroccan	24.2	**29.4**	23.5	22.3	47.2	**50.3**	44.7	43.5	**0.77**	**0.77**	0.75	0.74
Bible	MSA	27.1	**33.5**	27.4	31.1	49.3	**53.1**	48.4	50.9	**0.81**	**0.81**	0.80	**0.81**
Bible	MSA	23.4	28.1	25.1	**28.3**	46.0	**48.3**	46.1	48.2	**0.78**	**0.78**	0.77	**0.78**
Bible	Tunisian	24.8	**29.4**	25.3	23.3	46.9	**49.6**	45.5	43.7	**0.77**	**0.77**	0.75	0.74
FLORES	Egyptian	37.2	37.3	**39.8**	33.8	62.3	62.8	**63.2**	59.8	**0.87**	**0.87**	0.86	0.85
FLORES	Iraqi	41.3	40.6	**42.7**	37.7	65.7	**65.9**	65.8	62.8	**0.88**	**0.88**	0.87	0.87
FLORES	Moroccan	36.4	35.6	**39.1**	30.9	60.3	60.5	**62.0**	56.3	**0.84**	**0.84**	**0.84**	0.82
FLORES	MSA	46.0	46.7	**47.4**	44.5	69.1	**69.8**	69.3	67.2	**0.89**	**0.89**	**0.89**	**0.89**
FLORES	N. Levantine	45.6	46.5	**47.7**	41.1	67.7	**68.9**	67.9	64.4	**0.88**	**0.88**	0.87	0.86
FLORES	Najdi	45.0	45.6	**46.0**	42.7	68.2	**69.1**	68.6	66.5	**0.89**	**0.89**	0.88	0.88
FLORES	S. Levantine	48.8	49.6	**51.7**	44.2	70.0	**71.1**	70.4	66.7	0.89	**0.89**	0.88	0.87
FLORES	Tunisian	38.2	34.7	**38.6**	32.6	**62.7**	62.0	62.1	58.5	**0.86**	0.85	0.85	0.83
FLORES	Yemeni	42.7	41.7	**44.1**	38.8	66.6	66.8	**66.9**	63.6	**0.88**	**0.88**	0.87	0.87
MADAR	Egyptian	52.8	52.7	**78.3**	45.7	68.9	69.3	**85.3**	63.3	0.89	0.89	**0.92**	0.87
MADAR	Lebanese	52.6	52.2	**71.8**	40.8	68.4	68.9	**81.3**	58.8	0.88	0.88	**0.91**	0.83
MADAR	Moroccan	48.9	49.0	**70.2**	38.7	65.6	65.7	**79.7**	56.9	0.87	0.87	**0.90**	0.82
MADAR	MSA	57.6	56.2	**75.3**	53.5	73.2	72.5	**84.5**	69.7	0.92	0.91	**0.94**	0.91
MADAR	Qatari	54.3	55.6	**68.6**	46.8	69.6	70.5	**78.7**	63.3	0.90	0.90	**0.91**	0.87
MADAR	Tunisian	46.0	45.5	**66.2**	32.9	63.2	63.3	**76.4**	51.2	0.85	0.85	**0.88**	0.78
QAraC	Qatari	17.9	**19.3**	18.9	15.7	46.8	**48.9**	47.0	43.5	0.76	**0.77**	0.75	0.73
Average		40.2	40.8	**43.8**	36.0	61.1	61.8	**63.0**	57.4	**0.85**	**0.85**	0.84	0.83

Table 11: LLM results on the Dev sets across the dialects for each dataset in terms of BLEU, chrF++, and COMET. Best results are in bold.

E.3 DA-Fine-Tuned Models

Data Mixture	Corpus	BLEU				chrF++				COMET			
		Base	**Vocab+**	**Byte**	**Pixel**	**Base**	**Vocab+**	**Byte**	**Pixel**	**Base**	**Vocab+**	**Byte**	**Pixel**
MSA	Alexandria	**16.4**	**16.4**	16.0	16.3	**36.9**	36.7	36.1	36.5	**0.67**	**0.67**	0.66	**0.67**
	APT	**10.0**	9.0	9.1	9.3	**31.5**	30.7	30.4	30.9	**0.63**	**0.63**	**0.63**	**0.63**
	Bible	**10.6**	10.3	9.2	10.3	**30.5**	30.1	28.9	29.5	**0.63**	0.62	0.60	0.62
	FLORES	29.9	29.5	29.1	**30.2**	54.0	53.6	53.0	**54.2**	0.79	0.79	0.79	**0.80**
	MADAR	**23.2**	22.9	22.5	22.9	**40.9**	40.5	40.1	40.6	**0.70**	0.69	0.69	0.69
	QAraC	**9.8**	9.4	9.6	9.3	**32.8**	32.0	32.5	32.1	**0.64**	0.63	0.63	0.63
	Overall	**20.0**	19.8	19.3	**20.0**	**41.1**	40.7	40.1	40.8	**0.70**	**0.70**	0.69	**0.70**
+DA	Alexandria	**35.7**	34.7	35.0	35.5	**54.7**	53.8	54.0	54.4	**0.80**	0.79	0.79	**0.80**
	APT	**20.8**	19.9	20.3	20.3	**45.7**	44.6	45.2	45.2	**0.73**	0.72	**0.73**	**0.73**
	Bible	12.0	11.4	11.5	11.6	33.4	32.7	32.9	33.5	0.66	0.65	0.65	0.66
	FLORES	30.1	29.9	30.4	30.6	55.5	55.0	55.2	55.9	**0.82**	0.81	**0.82**	**0.82**
	MADAR	41.4	40.3	41.3	**41.8**	58.6	57.8	58.4	**58.8**	**0.82**	0.81	**0.82**	**0.82**
	QAraC	**16.9**	15.6	16.4	16.2	**41.2**	40.1	40.9	40.5	**0.69**	0.68	**0.69**	**0.69**
	Overall	30.9	30.1	30.6	30.9	52.1	51.3	51.7	52.1	0.78	**0.78**	**0.78**	**0.78**
+DA+MSA$_{1:1}$	Alexandria	35.5	34.8	34.9	35.4	54.5	53.9	53.8	54.2	**0.80**	0.79	0.79	**0.80**
	APT	20.2	19.5	20.1	19.9	45.2	44.0	44.7	44.7	**0.73**	0.72	0.72	**0.73**
	Bible	12.2	12.2	11.7	12.3	33.6	33.2	33.1	33.9	**0.67**	0.66	0.66	**0.67**
	FLORES	31.9	31.3	31.6	32.1	56.3	55.6	55.7	56.5	**0.82**	**0.82**	**0.82**	**0.82**
	MADAR	41.3	40.6	41.5	41.7	58.6	57.8	58.4	58.4	**0.82**	**0.82**	**0.82**	**0.82**
	QAraC	15.6	14.8	15.4	15.2	40.3	39.3	39.9	39.4	**0.69**	0.68	**0.69**	0.68
	Overall	31.2	30.6	30.9	**31.3**	52.2	51.5	51.7	52.1	**0.79**	0.78	0.78	0.78
+DA+MSA$_{1:2}$	Alexandria	35.5	34.9	34.9	35.2	54.5	53.9	53.9	54.1	**0.80**	0.79	0.79	**0.80**
	APT	20.2	19.4	19.9	20.1	44.9	44.1	44.5	44.6	**0.73**	0.72	0.72	**0.73**
	Bible	**12.4**	12.3	11.7	12.1	33.7	33.3	33.1	33.8	**0.67**	0.66	0.66	**0.67**
	FLORES	32.1	31.4	31.6	**32.2**	56.4	55.8	55.9	**56.6**	**0.82**	**0.82**	**0.82**	**0.82**
	MADAR	41.4	40.6	41.1	41.6	**58.7**	57.8	58.2	58.4	**0.82**	**0.82**	**0.82**	**0.82**
	QAraC	15.3	14.7	15.1	15.0	40.0	39.3	39.7	39.2	**0.69**	0.68	**0.69**	0.68
	Overall	**31.3**	30.7	30.8	31.2	**52.3**	51.6	51.7	52.1	**0.79**	0.78	0.78	0.78
+DA+MSA$_{1:4}$	Alexandria	35.2	34.9	34.8	35.2	54.3	53.9	53.7	54.1	**0.80**	0.79	0.79	**0.80**
	APT	20.2	19.3	19.5	20.0	44.9	43.9	44.2	44.6	**0.73**	0.72	0.72	**0.73**
	Bible	12.3	12.1	11.7	12.2	33.7	33.0	33.1	33.8	**0.67**	0.66	0.66	**0.67**
	FLORES	**32.2**	31.5	31.6	32.1	**56.5**	55.8	55.7	56.4	**0.82**	**0.82**	**0.82**	**0.82**
	MADAR	41.2	40.5	41.1	41.3	58.4	57.6	58.1	58.3	**0.82**	0.81	**0.82**	**0.82**
	QAraC	14.9	14.3	14.9	14.8	39.6	38.8	39.5	38.8	0.68	0.67	0.68	0.68
	Overall	31.2	30.6	30.7	31.1	52.1	51.5	51.6	52.0	**0.79**	0.78	0.78	0.78

Table 12: Macro-average BLEU, chrF++, and COMET scores on the Dev sets. Bold denotes the best-performing model within each data mixture; underlining denotes the best-performing fine-tuning data mixture for each model.

F Test Results

F.1 MSA-Pretrained Models

		BLEU				chrF++				COMET			
Corpus	**Dialect**	**Base**	**Vocab+**	**Byte**	**Pixel**	**Base**	**Vocab+**	**Byte**	**Pixel**	**Base**	**Vocab+**	**Byte**	**Pixel**
Alexandria	Egyptian	**17.0**	16.4	16.2	16.4	**37.7**	37.0	36.8	37.0	**0.70**	**0.70**	0.69	**0.70**
Alexandria	Jordanian	**18.8**	17.7	18.3	18.5	**40.0**	39.2	39.5	39.9	**0.72**	0.71	0.71	**0.72**
Alexandria	Lebanese	**15.9**	15.6	15.1	14.9	**35.2**	34.9	33.6	33.8	**0.66**	0.65	0.64	0.64
Alexandria	Libyan	**14.7**	14.3	14.0	14.3	**35.7**	35.1	35.0	35.3	0.68	0.68	0.68	**0.69**
Alexandria	Mauritanian	**7.1**	**7.1**	6.6	6.9	**25.1**	**25.1**	23.1	24.2	**0.52**	**0.52**	0.51	0.51
Alexandria	Moroccan	**8.8**	**8.8**	8.5	8.5	**27.7**	27.3	26.9	27.2	**0.55**	**0.55**	**0.55**	**0.55**
Alexandria	Omani	**18.6**	18.2	18.2	18.1	**40.0**	39.6	39.8	39.7	**0.71**	0.70	0.70	0.70
Alexandria	Palestinian	16.7	17.0	**17.1**	16.9	37.6	37.6	**37.8**	37.6	**0.68**	**0.68**	**0.68**	**0.68**
Alexandria	Saudi	22.4	**23.2**	21.8	22.2	43.5	**43.9**	42.9	43.2	**0.74**	**0.74**	**0.74**	**0.74**
Alexandria	Sudanese	19.1	19.1	18.5	**19.4**	39.9	39.8	39.4	**40.3**	**0.71**	**0.71**	0.70	**0.71**
Alexandria	Syrian	20.4	20.1	20.0	**21.1**	40.7	40.1	40.2	**41.2**	0.70	0.70	0.70	**0.71**
Alexandria	Tunisian	**14.3**	**14.3**	12.3	12.0	**34.5**	**34.5**	31.6	31.2	**0.61**	**0.61**	0.58	0.58
Alexandria	Yemeni	**17.3**	16.8	17.1	17.2	37.9	37.3	37.3	**38.0**	**0.71**	**0.71**	0.70	**0.71**
APT	Egyptian	**10.5**	10.1	10.1	10.1	**33.3**	32.6	32.5	32.4	**0.64**	**0.64**	0.63	**0.64**
APT	Levantine	7.7	7.7	7.8	**8.1**	30.1	29.9	29.6	**30.3**	0.62	0.62	0.62	**0.63**
Bible	Moroccan	4.5	**4.8**	4.4	**4.8**	**23.2**	23.0	22.3	22.9	0.51	0.51	0.51	**0.53**
Bible	MSA	**17.1**	16.7	15.5	16.2	**38.1**	37.8	36.1	36.4	**0.74**	0.72	0.71	0.72
Bible	MSA	**12.9**	12.4	11.7	11.8	**33.2**	32.2	31.9	31.5	**0.67**	0.66	0.65	0.66
Bible	Tunisian	**8.0**	7.1	6.8	7.2	**27.3**	26.0	25.3	25.6	**0.56**	0.55	0.54	0.55
FLORES	Egyptian	**25.1**	**25.1**	24.7	24.9	**50.9**	50.3	49.6	50.2	**0.80**	0.79	0.79	0.79
FLORES	Iraqi	29.9	**30.0**	29.7	29.4	**54.9**	54.7	54.3	54.3	**0.82**	**0.82**	0.81	**0.82**
FLORES	Moroccan	**18.2**	17.7	17.3	17.6	**42.6**	41.8	41.1	41.8	0.67	0.67	0.67	**0.68**
FLORES	MSA	36.5	**36.8**	35.8	36.4	**60.5**	60.4	59.5	59.9	**0.86**	**0.86**	0.85	0.85
FLORES	N. Levantine	**28.5**	27.3	28.2	28.3	**53.3**	52.3	52.6	53.1	0.79	0.79	0.79	**0.80**
FLORES	Najdi	35.9	**36.2**	35.0	35.9	**60.0**	59.8	58.8	59.5	**0.85**	**0.85**	**0.85**	**0.85**
FLORES	S. Levantine	**31.9**	31.3	30.5	31.8	**55.6**	55.2	54.1	55.2	**0.80**	**0.80**	0.79	**0.80**
FLORES	Tunisian	22.9	22.9	22.7	**23.2**	47.9	**48.0**	47.4	**48.0**	**0.74**	**0.74**	**0.74**	**0.74**
FLORES	Yemeni	32.0	**32.3**	31.4	32.0	56.5	**56.7**	55.8	56.4	**0.83**	**0.83**	**0.83**	**0.83**
MADAR	Egyptian	**26.4**	25.7	25.6	25.9	**44.6**	43.9	43.7	44.3	**0.74**	**0.74**	**0.74**	**0.74**
MADAR	Lebanese	**16.8**	15.8	16.0	16.6	**34.4**	33.6	33.2	34.1	**0.64**	0.63	0.62	0.63
MADAR	Moroccan	**13.1**	12.1	12.4	12.4	**31.5**	30.4	30.4	30.8	**0.61**	0.59	0.59	0.59
MADAR	MSA	**45.4**	**45.4**	45.2	45.3	**62.0**	61.8	61.7	61.6	**0.88**	0.87	0.87	0.87
MADAR	Qatari	26.0	**26.2**	25.7	25.5	**44.1**	43.9	43.3	43.5	**0.74**	**0.74**	0.73	0.73
MADAR	Tunisian	**12.5**	11.9	11.9	12.1	**30.0**	29.4	29.4	29.3	**0.58**	**0.58**	**0.58**	0.57
QAraC	Qatari	**10.0**	9.5	**10.0**	**10.0**	**33.1**	32.3	32.9	32.7	**0.64**	0.63	**0.64**	0.63
Average		**19.5**	19.3	18.9	19.2	**40.6**	40.2	39.7	40.1	**0.70**	0.69	0.69	0.69

Table 13: MSA-pretrained models results on the Test sets across the dialects for each dataset in terms of BLEU, chrF++, and COMET. Best results are in bold.

F.2 DA-Fine-Tuned Models

		BLEU				chrF++				COMET			
Corpus	**Dialect**	**Base**	**Vocab+**	**Byte**	**Pixel**	**Base**	**Vocab+**	**Byte**	**Pixel**	**Base**	**Vocab+**	**Byte**	**Pixel**
Alexandria	Egyptian	**34.1**	33.5	33.8	**34.1**	**54.5**	53.7	54.0	54.1	**0.81**	**0.81**	**0.81**	**0.81**
Alexandria	Jordanian	38.7	**38.8**	38.7	38.7	57.6	**57.7**	57.3	**57.7**	**0.83**	**0.83**	**0.83**	**0.83**
Alexandria	Lebanese	**39.0**	37.7	38.3	38.4	**56.9**	55.8	56.0	56.0	**0.82**	0.81	0.81	0.81
Alexandria	Libyan	28.1	27.9	28.4	**29.1**	48.9	48.4	48.9	**49.3**	**0.79**	**0.79**	**0.79**	**0.79**
Alexandria	Mauritanian	**24.3**	23.6	23.2	23.4	**43.7**	42.8	42.2	42.6	**0.70**	0.69	0.68	0.69
Alexandria	Moroccan	**23.7**	22.8	23.1	23.5	**43.5**	42.4	42.8	43.3	**0.69**	**0.69**	**0.69**	**0.69**
Alexandria	Omani	**35.4**	34.9	35.3	35.2	**55.1**	54.7	55.0	**55.1**	**0.81**	**0.81**	**0.81**	**0.81**
Alexandria	Palestinian	38.4	37.9	37.9	**38.9**	57.1	56.2	56.5	**57.2**	**0.82**	**0.82**	**0.82**	**0.82**
Alexandria	Saudi	**40.8**	39.7	40.7	40.7	59.2	58.6	59.1	**59.3**	**0.84**	0.83	0.83	0.83
Alexandria	Sudanese	**37.5**	36.7	36.6	37.4	**56.5**	55.7	55.9	56.4	**0.81**	0.80	**0.81**	**0.81**
Alexandria	Syrian	46.0	45.8	**46.3**	46.2	63.0	62.5	62.9	**63.1**	**0.85**	0.84	0.84	**0.85**
Alexandria	Tunisian	**36.7**	35.6	33.8	34.2	**56.4**	55.6	53.6	54.1	**0.78**	0.77	0.76	0.77
Alexandria	Yemeni	**32.7**	32.4	32.1	**32.7**	**52.0**	51.4	51.5	**52.0**	**0.81**	0.80	0.80	0.80
APT	Egyptian	20.8	20.3	20.7	**21.0**	**45.8**	45.1	45.5	45.5	**0.73**	0.72	0.72	0.72
APT	Levantine	**20.0**	18.8	19.8	19.6	**45.5**	44.1	45.2	44.9	**0.73**	0.72	**0.73**	**0.73**
Bible	Moroccan	8.4	7.6	8.1	**8.6**	29.6	28.3	28.9	**29.7**	**0.60**	0.58	0.59	**0.60**
Bible	MSA	**17.3**	17.2	16.6	16.6	**39.0**	38.5	38.3	38.4	**0.75**	0.74	0.74	0.74
Bible	MSA	12.8	**13.1**	12.3	12.4	**33.2**	33.1	33.0	33.1	**0.67**	**0.67**	0.66	**0.67**
Bible	Tunisian	**11.3**	10.4	9.8	10.4	**32.0**	30.8	30.7	31.4	**0.63**	0.62	0.61	**0.63**
FLORES	Egyptian	27.3	26.8	27.6	**28.0**	53.4	52.7	53.0	**53.6**	**0.82**	**0.82**	**0.82**	**0.82**
FLORES	Iraqi	31.1	30.2	30.1	**31.2**	56.1	55.3	55.4	**56.3**	**0.83**	**0.83**	**0.83**	**0.83**
FLORES	Moroccan	**22.2**	20.9	20.8	21.7	**47.0**	45.8	45.9	46.7	**0.74**	**0.74**	**0.74**	**0.74**
FLORES	MSA	**36.6**	36.1	36.3	36.5	60.5	59.9	60.1	**60.6**	**0.86**	0.85	0.85	**0.86**
FLORES	N. Levantine	**33.2**	31.5	32.2	33.1	**57.6**	56.0	56.7	57.5	**0.83**	0.82	**0.83**	**0.83**
FLORES	Najdi	35.9	35.3	35.7	**36.2**	59.9	59.3	59.7	**60.4**	**0.85**	**0.85**	**0.85**	**0.85**
FLORES	S. Levantine	**36.6**	34.8	35.5	36.1	**59.8**	58.2	58.8	59.5	0.83	0.83	0.83	**0.84**
FLORES	Tunisian	**26.3**	25.5	25.6	26.1	**51.5**	50.9	51.1	**51.5**	**0.79**	0.78	**0.79**	**0.79**
FLORES	Yemeni	33.0	32.8	32.3	**33.2**	57.7	57.3	57.2	**58.2**	**0.84**	**0.84**	**0.84**	**0.84**
MADAR	Egyptian	**43.1**	42.2	43.0	43.0	**61.1**	60.1	60.8	60.7	**0.85**	0.84	0.84	**0.85**
MADAR	Lebanese	**41.3**	40.2	41.1	**41.3**	**58.6**	57.1	58.0	58.2	**0.82**	0.81	0.81	**0.82**
MADAR	Moroccan	**33.9**	32.4	33.8	**33.9**	**52.7**	51.3	52.4	52.6	**0.78**	0.77	0.77	**0.78**
MADAR	MSA	50.2	50.9	**51.2**	50.6	66.8	**67.3**	**67.3**	67.0	**0.89**	**0.89**	**0.89**	**0.89**
MADAR	Qatari	43.0	**43.6**	**43.6**	43.4	59.9	**60.3**	**60.3**	60.1	**0.84**	**0.84**	**0.84**	**0.84**
MADAR	Tunisian	31.9	31.1	**32.3**	31.7	49.5	48.9	**50.2**	49.4	**0.76**	0.75	**0.76**	0.75
QAraC	Qatari	**16.0**	15.1	15.7	15.5	**40.4**	39.6	40.2	39.8	**0.69**	0.68	**0.69**	0.68
Average		**31.1**	30.4	30.6	30.9	**52.1**	51.3	51.6	51.9	**0.78**	**0.78**	**0.78**	**0.78**

Table 14: The DA+MSA$_{1:1}$ fine-tuned models result on the Test sets across the dialects for each dataset in terms of BLEU, chrF++, and COMET. Best results are in bold.

F.3 LLMs

		BLEU		chrF++		COMET	
Corpus	**Dialect**	**Gemini**	**Jais**	**Gemini**	**Jais**	**Gemini**	**Jais**
Alexandria	Egyptian	**40.7**	40.1	**61.9**	60.5	**0.86**	0.85
Alexandria	Jordanian	**43.6**	41.8	**62.6**	61.3	**0.86**	**0.86**
Alexandria	Lebanese	45.3	**46.8**	**64.0**	63.5	**0.86**	0.85
Alexandria	Libyan	**32.3**	32.1	**53.6**	53.2	**0.83**	**0.83**
Alexandria	Mauritanian	**41.4**	40.1	**60.6**	59.1	**0.83**	0.81
Alexandria	Moroccan	**38.5**	30.1	**57.9**	50.5	**0.80**	0.75
Alexandria	Omani	**41.4**	41.1	**61.4**	60.6	**0.85**	**0.85**
Alexandria	Palestinian	43.4	**43.7**	**63.4**	62.5	**0.86**	0.85
Alexandria	Saudi	**47.1**	45.2	**65.7**	64.1	**0.87**	0.86
Alexandria	Sudanese	42.4	**42.5**	**63.4**	62.0	**0.86**	**0.86**
Alexandria	Syrian	**53.0**	52.0	**69.5**	68.3	**0.88**	0.87
Alexandria	Tunisian	**51.6**	49.3	**69.9**	67.3	**0.86**	0.85
Alexandria	Yemeni	38.1	**38.1**	**58.5**	58.2	**0.85**	0.84
APT	Egyptian	22.1	**26.6**	50.0	**53.0**	**0.77**	**0.77**
APT	Levantine	18.3	**18.5**	**47.3**	46.8	**0.76**	0.75
Bible	Moroccan	**29.4**	24.0	**50.3**	45.2	**0.76**	0.74
Bible	MSA	**33.7**	27.9	**53.2**	48.5	**0.81**	0.79
Bible	MSA	**26.7**	23.8	**47.8**	44.9	**0.77**	0.76
Bible	Tunisian	**28.1**	20.7	**49.2**	43.1	**0.76**	0.73
FLORES	Egyptian	35.6	**35.8**	**61.9**	60.2	**0.86**	**0.86**
FLORES	Iraqi	**38.9**	38.5	**65.0**	62.8	**0.87**	**0.87**
FLORES	Moroccan	**35.8**	33.9	**60.7**	58.2	**0.84**	0.83
FLORES	MSA	**46.4**	44.0	**69.7**	66.9	**0.89**	0.88
FLORES	N. Levantine	**44.0**	41.3	**67.6**	63.8	**0.88**	0.86
FLORES	Najdi	**44.3**	42.7	**68.6**	66.4	**0.89**	0.88
FLORES	S. Levantine	**47.2**	45.1	**69.7**	66.5	**0.88**	0.87
FLORES	Tunisian	**36.2**	34.9	**62.2**	59.6	**0.85**	0.84
FLORES	Yemeni	**40.9**	39.6	**66.4**	64.0	**0.88**	0.87
MADAR	Egyptian	49.6	**50.3**	**67.1**	66.7	**0.89**	0.88
MADAR	Lebanese	48.7	**50.9**	66.0	**66.9**	**0.88**	0.87
MADAR	Moroccan	**48.5**	46.2	**65.5**	63.2	**0.87**	0.85
MADAR	MSA	**54.6**	54.0	**71.5**	70.2	**0.91**	**0.91**
MADAR	Qatari	**52.9**	50.8	**68.5**	67.1	**0.89**	0.88
MADAR	Tunisian	**45.5**	45.3	**63.2**	62.2	**0.86**	0.84
QAraC	Qatari	**20.0**	19.0	**49.4**	47.1	**0.77**	0.75
Average		**40.2**	38.8	**61.5**	59.5	**0.85**	0.83

Table 15: Gemini and Jais results on the Test sets across the dialects for each dataset in terms of BLEU, chrF++, and COMET. Best results are in bold.

G Robustness Results

G.1 Diacritized Input

		BLEU				chrF++				COMET			
Corpus	**Dialect**	**Base**	**Vocab+**	**Byte**	**Pixel**	**Base**	**Vocab+**	**Byte**	**Pixel**	**Base**	**Vocab+**	**Byte**	**Pixel**
Alexandria	Egyptian	1.5	0.7	4.5	**26.2**	16.3	14.7	25.3	**46.3**	0.45	0.39	0.54	**0.72**
Alexandria	Jordanian	3.0	1.2	7.7	**30.6**	16.6	15.1	28.2	**50.1**	0.46	0.40	0.58	**0.75**
Alexandria	Lebanese	2.8	1.3	8.1	**29.2**	17.8	16.4	28.7	**48.0**	0.47	0.42	0.57	**0.72**
Alexandria	Libyan	2.1	0.8	6.3	**23.4**	15.9	14.9	26.8	**43.4**	0.45	0.40	0.57	**0.72**
Alexandria	Mauritanian	4.1	2.0	6.8	**18.2**	20.1	18.2	26.5	**37.5**	0.49	0.44	0.55	**0.64**
Alexandria	Moroccan	2.9	1.7	5.4	**17.3**	17.5	16.6	25.4	**36.9**	0.46	0.41	0.54	**0.64**
Alexandria	Omani	1.8	1.0	6.1	**27.7**	15.5	14.6	26.7	**47.9**	0.45	0.40	0.57	**0.74**
Alexandria	Palestinian	2.3	0.9	6.9	**29.5**	16.1	14.8	27.4	**49.0**	0.45	0.39	0.56	**0.74**
Alexandria	Saudi	2.0	0.6	7.1	**30.9**	15.3	14.3	27.4	**50.2**	0.44	0.39	0.57	**0.75**
Alexandria	Sudanese	1.5	0.7	5.9	**29.4**	15.0	14.2	26.4	**48.8**	0.43	0.38	0.55	**0.73**
Alexandria	Syrian	2.3	0.9	9.1	**35.4**	16.9	15.6	29.8	**53.9**	0.45	0.40	0.58	**0.76**
Alexandria	Tunisian	3.0	1.6	7.0	**26.3**	19.0	17.3	28.2	**46.5**	0.48	0.42	0.56	**0.70**
Alexandria	Yemeni	2.2	1.1	7.1	**25.9**	16.6	15.3	27.4	**46.0**	0.45	0.40	0.57	**0.73**
APT	Egyptian	0.8	0.5	3.0	**15.5**	13.0	13.9	23.2	**38.6**	0.40	0.39	0.51	**0.65**
APT	Levantine	0.9	0.6	3.0	**14.6**	13.8	14.5	24.3	**38.9**	0.40	0.40	0.53	**0.66**
Bible	Moroccan	0.6	0.4	1.7	**6.3**	13.2	15.4	20.3	**25.9**	0.36	0.36	0.46	**0.55**
Bible	MSA	0.7	0.5	1.9	**12.6**	10.6	13.1	20.2	**32.6**	0.35	0.35	0.47	**0.64**
Bible	MSA	0.6	0.6	1.4	**8.5**	11.0	14.2	19.4	**27.8**	0.35	0.36	0.46	**0.59**
Bible	Tunisian	1.1	0.8	2.2	**7.7**	13.2	14.7	21.1	**27.2**	0.38	0.37	0.48	**0.56**
FLORES	Egyptian	0.9	0.3	2.2	**19.8**	13.1	12.4	22.8	**45.3**	0.37	0.33	0.49	**0.72**
FLORES	Iraqi	0.7	0.7	1.9	**22.0**	13.4	13.3	22.6	**47.5**	0.37	0.34	0.49	**0.74**
FLORES	Moroccan	1.2	0.6	2.2	**15.7**	15.5	14.3	22.7	**40.1**	0.40	0.35	0.49	**0.67**
FLORES	MSA	0.8	0.4	2.0	**26.0**	12.2	12.8	22.4	**51.0**	0.36	0.34	0.49	**0.75**
FLORES	N. Levantine	1.0	0.5	2.4	**23.3**	13.5	12.9	22.9	**47.5**	0.37	0.34	0.49	**0.73**
FLORES	Najdi	0.6	0.5	2.4	**25.8**	13.5	13.0	23.0	**50.9**	0.37	0.34	0.49	**0.75**
FLORES	S. Levantine	1.2	0.6	2.8	**25.8**	13.8	13.4	23.4	**49.8**	0.38	0.34	0.49	**0.73**
FLORES	Tunisian	0.9	0.5	2.0	**19.0**	14.1	13.6	22.6	**43.9**	0.38	0.34	0.48	**0.70**
FLORES	Yemeni	0.9	0.5	2.2	**23.3**	14.0	13.0	22.9	**49.1**	0.38	0.34	0.49	**0.74**
MADAR	Egyptian	2.0	1.0	8.3	**35.8**	16.6	15.0	29.3	**54.2**	0.47	0.43	0.58	**0.77**
MADAR	Lebanese	2.6	1.0	9.5	**32.3**	18.1	16.0	29.1	**49.8**	0.49	0.43	0.59	**0.74**
MADAR	Moroccan	3.8	2.0	8.8	**25.0**	20.0	17.8	29.2	**44.4**	0.51	0.45	0.58	**0.70**
MADAR	MSA	1.7	0.8	6.1	**41.5**	15.1	13.7	26.9	**59.8**	0.47	0.41	0.57	**0.81**
MADAR	Qatari	2.8	1.1	9.9	**33.1**	17.4	15.3	29.6	**51.0**	0.49	0.43	0.60	**0.76**
MADAR	Tunisian	3.0	1.5	8.0	**23.5**	18.3	16.2	27.6	**41.2**	0.50	0.44	0.58	**0.69**
QAraC	Qatari	0.7	0.3	3.2	**12.8**	12.7	12.8	22.0	**34.6**	0.42	0.40	0.52	**0.62**
Average		1.7	0.9	5.0	**23.4**	15.3	14.7	25.2	**44.4**	0.43	0.39	0.53	**0.70**

Table 16: The DA+MSA$_{1:1}$ fine-tuned models results on the diacritized Test sets across the dialects for each dataset in terms of BLEU, chrF++, and COMET. Best results are in bold.

		BLEU		chrF++		COMET	
Corpus	**Dialect**	**Gemini**	**Jais**	**Gemini**	**Jais**	**Gemini**	**Jais**
Alexandria	Egyptian	**40.1**	37.9	**61.8**	59.2	**0.82**	0.81
Alexandria	Jordanian	**42.7**	39.2	**61.9**	59.4	**0.82**	0.81
Alexandria	Lebanese	**44.6**	44.0	**64.0**	62.4	**0.82**	0.81
Alexandria	Libyan	**31.5**	30.1	**53.4**	51.5	**0.80**	0.78
Alexandria	Mauritanian	**40.3**	36.2	**59.8**	56.1	**0.80**	0.79
Alexandria	Moroccan	**36.1**	25.5	**55.3**	46.6	**0.77**	0.71
Alexandria	Omani	**40.2**	37.4	**61.1**	58.4	**0.82**	0.81
Alexandria	Palestinian	42.7	**42.8**	**62.9**	61.6	**0.82**	0.81
Alexandria	Saudi	**43.8**	42.5	**63.5**	61.6	**0.83**	0.82
Alexandria	Sudanese	**42.1**	40.7	**62.9**	60.8	**0.83**	0.81
Alexandria	Syria	**52.9**	49.8	**69.1**	66.8	**0.84**	0.83
Alexandria	Tunisian	**50.2**	48.6	**69.4**	66.5	**0.84**	0.82
Alexandria	Yemeni	**37.0**	36.7	**57.9**	56.6	**0.81**	0.80
APT	Egyptian	2.0	**21.8**	24.7	**48.6**	**0.73**	**0.73**
APT	Levantine	14.8	**15.8**	**44.7**	43.8	**0.72**	0.71
Bible	Moroccan	**28.6**	20.4	**49.7**	42.2	**0.75**	0.72
Bible	MSA	**32.1**	26.4	**51.9**	47.3	**0.77**	0.76
Bible	MSA	**26.2**	22.6	**47.3**	43.7	**0.76**	0.74
Bible	Tunisian	**27.5**	19.7	**49.0**	42.4	**0.74**	0.71
FLORES	Egyptian	**33.9**	31.1	**61.0**	57.5	**0.82**	0.80
FLORES	Iraqi	**36.3**	33.6	**63.6**	59.8	**0.83**	0.81
FLORES	Moroccan	**33.2**	29.9	**59.1**	55.5	**0.80**	0.78
FLORES	MSA	**44.6**	38.3	**68.7**	63.8	**0.85**	0.83
FLORES	N. Levantine	**41.9**	37.8	**66.6**	61.7	**0.83**	0.81
FLORES	Najdi	**42.1**	37.9	**67.2**	63.5	**0.83**	0.82
FLORES	S. Levantine	**45.8**	42.2	**69.2**	64.9	**0.84**	0.82
FLORES	Tunisian	**34.5**	30.6	**61.2**	57.1	**0.81**	0.80
FLORES	Yemeni	**38.8**	36.0	**65.3**	61.7	**0.83**	0.82
MADAR	Egyptian	**47.6**	47.3	**65.9**	65.2	**0.85**	**0.85**
MADAR	Lebanese	**51.1**	47.2	**67.2**	63.8	**0.85**	0.84
MADAR	Moroccan	**46.7**	45.4	**64.3**	62.8	**0.84**	0.83
MADAR	MSA	**54.2**	51.8	**70.5**	68.4	**0.88**	0.86
MADAR	Qatari	**53.0**	47.3	**68.7**	64.2	**0.87**	0.85
MADAR	Tunisian	**44.1**	41.0	**61.5**	58.5	**0.83**	0.81
QAraC	Qatari	**20.2**	19.3	**48.8**	46.1	**0.74**	0.72
Average		**38.4**	35.8	**60.0**	57.4	**0.81**	0.80

Table 17: Gemini and Jais results on the diacritized Test sets across the dialects for each dataset in terms of BLEU, chrF++, and COMET. Best results are in bold.

G.2 Dotless Input

		BLEU				chrF++				COMET			
Corpus	**Dialect**	**Base**	**Vocab+**	**Byte**	**Pixel**	**Base**	**Vocab+**	**Byte**	**Pixel**	**Base**	**Vocab+**	**Byte**	**Pixel**
Alexandria	Egyptian	4.0	1.4	3.1	**16.2**	19.5	15.9	20.0	**36.7**	0.50	0.44	0.50	**0.65**
Alexandria	Jordanian	5.5	2.5	4.0	**18.8**	20.9	16.3	20.4	**37.8**	0.51	0.45	0.51	**0.66**
Alexandria	Lebanese	5.0	2.0	3.2	**17.8**	21.5	17.1	20.2	**37.4**	0.51	0.45	0.50	**0.64**
Alexandria	Libyan	3.5	1.6	2.8	**13.4**	18.6	14.8	19.3	**33.9**	0.49	0.42	0.50	**0.63**
Alexandria	Mauritanian	3.7	1.8	2.5	**10.0**	20.0	16.2	19.2	**28.8**	0.50	0.44	0.49	**0.58**
Alexandria	Moroccan	3.8	2.0	3.2	**10.8**	18.8	15.9	19.2	**30.0**	0.49	0.43	0.49	**0.59**
Alexandria	Omani	3.9	1.9	3.3	**16.5**	19.5	16.0	19.9	**37.1**	0.50	0.44	0.51	**0.66**
Alexandria	Palestinian	4.9	2.6	3.8	**18.9**	20.1	16.1	20.0	**38.4**	0.50	0.44	0.50	**0.65**
Alexandria	Saudi	4.9	2.2	4.3	**18.2**	20.0	16.1	20.4	**38.3**	0.50	0.43	0.51	**0.66**
Alexandria	Sudanese	4.4	1.7	3.8	**17.4**	20.3	15.5	20.5	**37.5**	0.50	0.43	0.50	**0.64**
Alexandria	Syrian	4.9	2.5	4.4	**23.0**	21.1	17.0	20.9	**42.0**	0.51	0.45	0.51	**0.68**
Alexandria	Tunisian	4.7	2.1	3.3	**13.7**	21.6	17.2	20.4	**34.3**	0.51	0.45	0.50	**0.61**
Alexandria	Yemeni	5.0	2.8	3.9	**15.7**	20.8	17.6	20.7	**35.9**	0.51	0.45	0.51	**0.65**
APT	Egyptian	2.3	1.0	1.8	**9.3**	18.3	14.8	17.7	**30.8**	0.49	0.42	0.48	**0.58**
APT	Levantine	2.0	0.8	1.9	**9.6**	18.7	14.7	18.1	**31.9**	0.49	0.43	0.49	**0.60**
Bible	Moroccan	1.1	0.5	0.8	**3.1**	15.8	14.0	16.4	**22.3**	0.43	0.38	0.43	**0.51**
Bible	MSA	2.1	1.1	1.6	**8.0**	17.4	14.3	17.3	**27.9**	0.45	0.39	0.45	**0.58**
Bible	MSA	1.4	0.9	1.2	**5.1**	15.4	13.9	16.0	**23.4**	0.44	0.38	0.44	**0.54**
Bible	Tunisian	1.4	0.9	1.0	**4.5**	15.5	13.3	16.2	**22.9**	0.43	0.38	0.43	**0.52**
FLORES	Egyptian	1.6	0.7	0.9	**10.7**	17.7	12.8	17.6	**35.1**	0.43	0.36	0.44	**0.63**
FLORES	Iraqi	1.8	0.9	1.5	**11.8**	17.7	13.4	18.0	**36.0**	0.44	0.37	0.44	**0.63**
FLORES	Moroccan	1.3	0.7	0.8	**7.9**	16.6	12.4	17.7	**30.6**	0.42	0.36	0.43	**0.57**
FLORES	MSA	1.8	0.9	1.3	**15.0**	18.1	13.6	18.3	**39.3**	0.44	0.37	0.44	**0.65**
FLORES	N. Levantine	1.9	1.0	1.5	**12.3**	17.3	13.1	18.3	**36.0**	0.43	0.37	0.44	**0.62**
FLORES	Najdi	1.8	0.8	1.4	**14.6**	17.9	13.3	18.3	**39.1**	0.44	0.37	0.44	**0.64**
FLORES	S. Levantine	2.0	1.2	1.7	**14.7**	18.1	13.7	18.9	**38.1**	0.44	0.37	0.45	**0.62**
FLORES	Tunisian	1.6	0.8	1.1	**10.1**	17.3	13.4	17.9	**34.1**	0.43	0.37	0.44	**0.61**
FLORES	Yemeni	1.7	0.8	1.4	**13.0**	17.8	13.2	18.3	**37.7**	0.44	0.37	0.44	**0.64**
MADAR	Egyptian	4.4	2.1	3.3	**15.5**	19.0	15.3	18.1	**34.9**	0.51	0.46	0.50	**0.64**
MADAR	Lebanese	4.6	2.2	2.7	**16.0**	18.7	15.1	17.3	**33.8**	0.52	0.45	0.50	**0.63**
MADAR	Moroccan	3.8	1.4	2.5	**11.4**	18.5	14.8	17.6	**29.6**	0.50	0.44	0.49	**0.60**
MADAR	MSA	5.8	2.8	4.3	**24.7**	22.0	17.8	20.1	**44.0**	0.53	0.47	0.51	**0.69**
MADAR	Qatari	6.6	2.2	4.3	**19.1**	21.0	15.9	19.4	**36.7**	0.54	0.46	0.51	**0.65**
MADAR	Tunisian	3.7	1.1	3.0	**11.5**	17.3	13.9	16.8	**29.1**	0.50	0.44	0.49	**0.60**
QAraC	Qatari	2.3	0.9	2.0	**7.6**	16.8	12.7	16.5	**27.6**	0.49	0.42	0.48	**0.56**
Average		3.3	1.5	2.5	**13.3**	18.7	14.9	18.6	**34.0**	0.48	0.42	0.48	**0.62**

Table 18: The DA+MSA$_{1:1}$ fine-tuned models results on the dotless Test sets across the dialects for each dataset in terms of BLEU, chrF++, and COMET. Best results are in bold.

		BLEU		chrF++		COMET	
Corpus	**Dialect**	**Gemini**	**Jais**	**Gemini**	**Jais**	**Gemini**	**Jais**
Alexandria	Egyptian	**36.2**	15.0	**58.3**	35.9	**0.81**	0.67
Alexandria	Jordanian	**38.5**	14.1	**58.7**	34.2	**0.81**	0.66
Alexandria	Lebanese	**39.4**	16.8	**59.4**	35.7	**0.81**	0.65
Alexandria	Libyan	**25.2**	7.8	**48.0**	27.2	**0.76**	0.58
Alexandria	Mauritanian	**33.8**	10.7	**54.1**	30.6	**0.78**	0.62
Alexandria	Moroccan	**28.0**	7.0	**48.2**	26.5	**0.74**	0.58
Alexandria	Omani	**34.0**	10.9	**55.9**	32.1	**0.80**	0.64
Alexandria	Palestinian	**39.5**	15.1	**60.2**	34.5	**0.81**	0.65
Alexandria	Saudi	**41.4**	15.8	**61.1**	36.0	**0.82**	0.66
Alexandria	Sudanese	**36.1**	10.9	**57.7**	30.3	**0.79**	0.59
Alexandria	Syrian	**47.5**	17.2	**65.1**	36.9	**0.83**	0.65
Alexandria	Tunisian	**44.9**	16.5	**65.4**	38.0	**0.83**	0.67
Alexandria	Yemeni	**33.8**	13.9	**55.0**	33.4	**0.80**	0.65
APT	Egyptian	1.7	**6.3**	22.7	**25.3**	**0.69**	0.55
APT	Levantine	**12.8**	3.7	**40.8**	23.1	**0.68**	0.54
Bible	Moroccan	**26.1**	8.5	**47.4**	27.4	**0.74**	0.58
Bible	MSA	**31.0**	12.1	**50.7**	31.1	**0.76**	0.61
Bible	MSA	**25.4**	9.6	**46.4**	27.5	**0.74**	0.59
Bible	Tunisian	**26.3**	6.8	**47.1**	24.9	**0.73**	0.56
FLORES	Egyptian	**27.9**	8.4	**55.8**	31.2	**0.79**	0.59
FLORES	Iraqi	**28.1**	6.7	**56.3**	29.5	**0.79**	0.58
FLORES	Moroccan	**25.8**	5.4	**52.2**	27.1	**0.76**	0.55
FLORES	MSA	**38.1**	10.7	**63.9**	33.8	**0.83**	0.61
FLORES	N. Levantine	**34.8**	7.6	**60.5**	29.9	**0.80**	0.58
FLORES	Najdi	**34.6**	8.1	**61.5**	31.6	**0.81**	0.59
FLORES	S. Levantine	**38.0**	9.5	**62.6**	32.1	**0.81**	0.58
FLORES	Tunisian	**27.2**	7.0	**54.5**	28.8	**0.78**	0.57
FLORES	Yemeni	**30.9**	8.6	**58.4**	31.3	**0.80**	0.59
MADAR	Egyptian	**43.2**	14.3	**61.5**	32.3	**0.82**	0.63
MADAR	Lebanese	**44.1**	11.5	**60.0**	28.3	**0.81**	0.60
MADAR	Moroccan	**38.4**	12.2	**56.7**	30.2	**0.80**	0.62
MADAR	MSA	**50.5**	18.9	**67.4**	37.4	**0.86**	0.66
MADAR	Qatari	**43.4**	15.6	**59.5**	32.3	**0.81**	0.62
MADAR	Tunisian	**35.5**	9.2	**52.9**	25.0	**0.78**	0.58
QAraC	Qatari	**15.5**	3.5	**42.0**	21.4	**0.68**	0.54
Average		**33.1**	10.7	**55.1**	30.7	**0.78**	0.61

Table 19: Gemini and Jais results on the dotless Test sets across the dialects for each dataset in terms of BLEU, chrF++, and COMET. Best results are in bold.

SAFIR: A Sub-Dialectal Benchmark for Saudi Arabic–English Machine Translation

Nour Qandos[1], **Omer Nacar**[2], **Aya Hamod**[1]
Samar Ahmad[1], **Rahaf Saeed Al Hallay**[1], **Ahmad Al Rabghi**[1]
[1]NAMAA Community
[2]Tuwaiq Academy, Riyadh, Saudi Arabia

Correspondence: o.najar@tuwaiq.edu.sa

Abstract

Arabic machine translation benchmarks provide limited coverage of variation within Saudi Arabic. We introduce SAFIR, a Saudi-centric benchmark of 610 unique sentence pairs organized into 12 creator-specific evaluation splits across five regional groups: Najdi, Hijazi, Southern, Northern, and Eastern. Native speakers authored the Arabic sentences and their English translations; 235 pairs received independent double review for dialect authenticity, translation adequacy, and fluency. We evaluate twelve systems, comprising closed-weight LLMs, open-weight LLMs, and dedicated MT models, in the zero-shot Saudi→English setting using reference-based metrics and GPT-4o judgments. Closed-weight LLMs obtain the strongest aggregate results. Gemini 2.5 Pro achieves the highest average BLEU (34.41), while GPT-5 achieves the highest BERTScore (83.60). Rankings vary across splits, and Southern varieties present particular challenges in judged adequacy. Differences between reference overlap, adequacy, and fluency support reporting performance by sampled variety and evaluation dimension. SAFIR and its leaderboard support more detailed assessment of Saudi Arabic translation.

1 Introduction

Machine translation has advanced substantially for Modern Standard Arabic (MSA), but evaluating the spoken varieties used in everyday communication remains challenging. Dialectal resources represent only part of the lexical, morphological, and phonological variation found across Arabic-speaking communities. Evaluation under broad country or regional labels can further conceal differences between local varieties.

Saudi Arabic illustrates the need for finer coverage. Linguistic descriptions distinguish regional groupings such as Central (Najdi), Western (Hijazi), Southern, Eastern, and Northern, with further variation across cities and communities (Al-Monef et al., 2026). Existing resources capture parts of this diversity. MADAR, for example, includes Riyadh and Jeddah and already supports city-level Saudi comparisons (Bouamor et al., 2018). Coverage across a broader range of Saudi locations nevertheless remains limited.

We introduce SAFIR, a Saudi-centric machine translation benchmark comprising 610 unique Saudi–English sentence pairs. Its collection design spans eight locations in seven administrative regions, organized into five regional groups and 12 creator-specific evaluation splits. Native speakers authored and translated the sentences, and an independent quality assessment double-reviewed 235 pairs. The splits preserve the individual collections and their local labels; they do not constitute a validated taxonomy of twelve distinct dialects. This distinction enables detailed evaluation while keeping the linguistic interpretation of split-level differences appropriately scoped.

Using SAFIR, we evaluate twelve systems comprising closed-weight LLMs, open-weight LLMs, and dedicated MT models in the zero-shot Saudi→English setting. We combine reference-based metrics with LLM judgments to examine complementary aspects of translation quality. The public leaderboard provides aggregate and split-level comparisons.[1] Our contributions are:

- A Saudi-centric benchmark with native-speaker authorship and translation, coverage across five regional groups, and independent double review of a 235-pair quality-control sample.
- Evaluation of twelve systems across creator-specific collections, using complementary measures of reference overlap and judged quality.

[1] https://huggingface.co/spaces/NAMAA-Space/SAFIR-Leaderboard

Proceedings of The Fourth Arabic Natural Language Processing Conference (ArabicNLP 2026), pages 289–301
October 28-29, 2026 ©2026 Association for Computational Linguistics

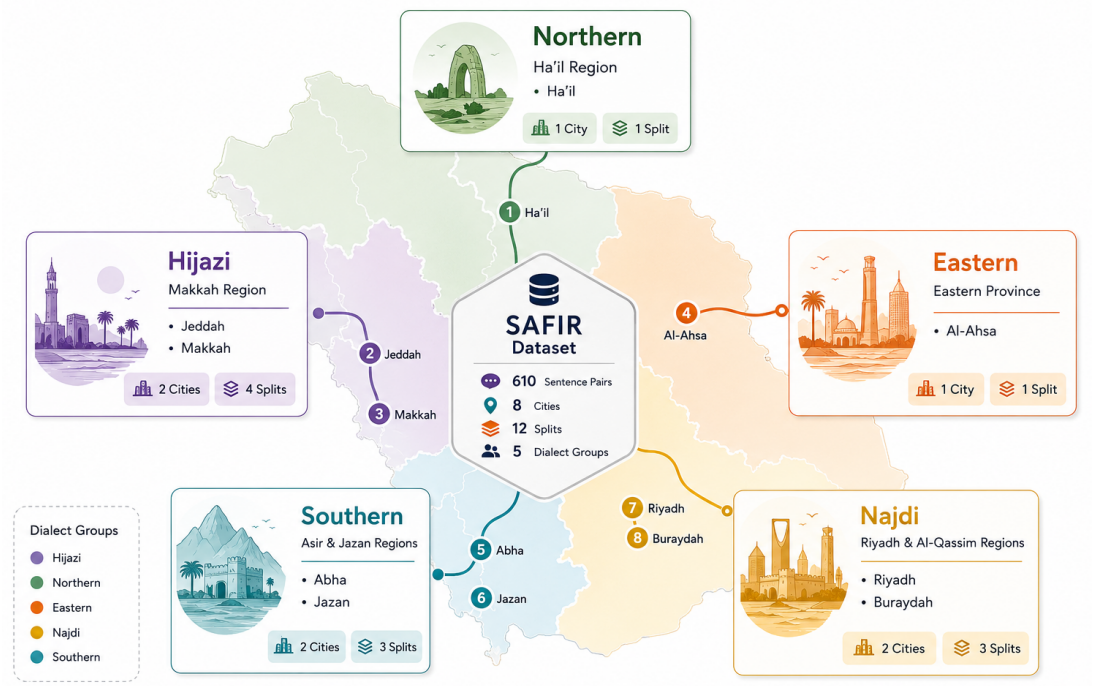


Figure 1: SAFIR's collection design across five regional groups and eight designated collection locations in seven administrative regions. The 12 creator-specific evaluation splits contain 610 unique Saudi–English pairs. Locations are schematic and not to scale; colors indicate regional groups.

- A public leaderboard for comparing Saudi Arabic translation performance at aggregate and split levels.

2 Related Work

Arabic MT resources differ in their geographic coverage and construction. PADIC aligns Algerian, Tunisian, Levantine, and MSA varieties, without a Saudi or Gulf variety (Meftouh et al., 2015). WMT24++ extends evaluation to 55 languages and dialects, but represents Saudi Arabic through a single `ar_SA` track (Deutsch et al., 2025).

Other resources provide regional or city labels. FLORES+ includes approximately 2,009 Najdi sentences across its development and devtest partitions (NLLB Team, 2022). MADAR covers 25 Arabic cities, including Riyadh and Jeddah with 2,000 sentences per Saudi city (Bouamor et al., 2018). Alexandria provides human translations and peer revision across dialects from 13 Arab countries. Its Saudi material uses Southern, Hijazi, and Khaleeji labels and also includes city-of-origin metadata (El Mekki et al., 2026).

These resources already support country, regional, or city-level evaluation. SAFIR complements them through focused coverage across Saudi regional groups and separate evaluation of local creator collections. Its contribution is the breadth of its Saudi collection design and its evaluation granularity, rather than the first use of city labels for Saudi Arabic. Table 1 compares Saudi coverage, subset sizes, reference construction, and human involvement across these resources.

3 The SAFIR Benchmark

Construction proceeded through Arabic sentence authorship, English translation, and human quality review over approximately six months of community work. The benchmark contains 610 unique Saudi–English sentence pairs.

3.1 Dialectal and Regional Coverage

The collection design covers seven of Saudi Arabia's 13 administrative regions:[2] Riyadh, Al-Qassim, Ha'il, Makkah, Asir, Jazan, and the Eastern Province. Its eight designated collection locations are Riyadh, Buraydah, Ha'il, Makkah, Jeddah, Abha, Jazan, and Al-Ahsa. We organize the material into Najdi, Hijazi, Northern, Southern, and Eastern groups; Southern combines the sampled Asir and Jazan material.

Each evaluation split contains the material of

[2]ISO 3166-2:SA: https://www.iso.org/obp/ui/#iso:code:3166:SA.

Dataset	Saudi coverage	Saudi subset size	Creation direction	Human translation and review
WMT24++ (Deutsch et al., 2025)	One country-level label: `ar_SA`	998 segments	En→Saudi	Professional translation and post-editing; sampled dialect checks.
PADIC (Meftouh et al., 2015)	No Saudi variety	0	Ar↔Ar, with MSA as a pivot	Native-speaker translation; separate review not specified.
FLORES+ (NLLB Team, 2022)	One regional variety: Najdi	2,009 sentences	MSA→Najdi; aligned with En	Human adaptation under the FLORES quality-assurance protocol.
MADAR (Bouamor et al., 2018)	Two Saudi cities: Riyadh and Jeddah	4,000 pairs (2,000 per city)	En/Fr→Saudi	Native-speaker translation; separate corpus review not specified.
Alexandria (El Mekki et al., 2026)	Three dialect labels: Southern, Hijazi, and Khaleeji; city-of-origin metadata	11,807 turns	En→Saudi	Human translation and peer revision.
SAFIR	**Five regional groups with 12 creator-specific evaluation splits**	610 unique pairs	Saudi→En	Native-speaker authorship and translation; 235 pairs independently double-reviewed.

Table 1: Expanded comparison of Saudi coverage and construction. En, Fr, and Ar denote English, French, and Arabic. Creation direction concerns reference construction, not every possible evaluation direction. WMT24++ collected 998 segments and retained 960 for evaluation. FLORES+ counts dev and devtest; MADAR's two Saudi varieties share 2,000 source sentences. Alexandria's size is its Saudi total across partitions. SAFIR counts unique Arabic–English pairs.

one creator (Table 2). Same-city labels distinguish collections: Najdi Riyadh and Najdi Urban were assigned to Riyadh; Hijazi Makkah and Hijazi Urban to Makkah; and Hijazi Jeddah and Hijazi Urban Jeddah to Jeddah. "Urban" was retained from collection and is not an operationally defined linguistic category. Likewise, Southern Qahtan and Southern Asiri retain community and regional descriptors without establishing mutually exclusive linguistic categories. No predetermined set of linguistic features was used to separate these collections. Split comparisons therefore reflect sampled language use, creator choices, and reference style alongside possible dialectal differences.

3.2 Arabic Sentence Collection

Sixteen native-speaker participants contributed: eight were creators only, four reviewers only, and four served in both roles. Thus, there were twelve creators and eight distinct reviewers. No participant reviewed their own material. The creators represented Najdi (three), Northern (one), Hijazi (four), Southern (three), and Eastern (one). Eligibility required residence in the target location, fluency in the local variety, and proficiency in English.

Shared guidelines asked each creator to author approximately 50 conversational sentences containing vocabulary characteristic of the assigned variety. This was a flexible target. Creators composed sentences from their knowledge of everyday language in their communities and checked their local authenticity. Copying from published datasets was prohibited. This authorship requirement reduces deliberate reuse but does not establish the absence of incidental overlap with prior corpora or model training data.

3.3 English Translation

Each creator translated their own sentences into English, aiming to preserve the source meaning and communicative intent. Machine translation tools were permitted as an aid, but creators remained responsible for the final references.

3.4 Human Quality Review

We sampled up to 20 sentences per creator, yielding 235 pairs for independent double review. Two reviewers were assigned to each regional group, with reviewer assignments shared across groups where needed. The sampling totals were Najdi (59), Northern (20), Eastern (20), Southern (57), and Hijazi (79). Reviewers used a 1–5 ordinal scale for *dialect authenticity* (reflection of the claimed variety), *translation adequacy* (preservation of meaning), and *translation fluency* (naturalness and grammaticality of the English).

Mean ratings were 4.74 for authenticity, 4.57 for adequacy, and 4.70 for fluency (Table 3). Ratings

Group	Admin. region	Collection location	Split label	Label basis
Eastern	Eastern Province	Al-Ahsa	Eastern Urban	Creator label within Eastern group
Hijazi	Makkah	Makkah	Hijazi Makkah	City descriptor
Hijazi	Makkah	Makkah	Hijazi Urban	Creator label within Makkah
Hijazi	Makkah	Jeddah	Hijazi Urban Jeddah	Creator label within Jeddah
Hijazi	Makkah	Jeddah	Hijazi Jeddah	City descriptor
Najdi	Al-Qassim	Buraydah	Najdi Qasim	Regional descriptor
Najdi	Riyadh	Riyadh	Najdi Riyadh	City descriptor
Najdi	Riyadh	Riyadh	Najdi Urban	Creator label within Riyadh
Northern	Ha'il	Ha'il	Northern Hail	Geographic descriptor
Southern	Asir	Abha	Southern Qahtan	Community descriptor
Southern	Jazan	Jazan	Southern Jazan	Geographic descriptor
Southern	Asir	Abha	Southern Asiri	Regional descriptor

Table 2: Designated collection locations and naming basis of SAFIR's 12 evaluation splits. Every row represents one creator's collection. The final column explains the naming convention rather than a validated linguistic classification. Split labels match those used in the results tables and on the leaderboard.

Dimension	Mean	Ratings ≤ 3
Authenticity	4.74	7.4%
Adequacy	4.57	10.6%
Fluency	4.70	7.4%

Table 3: Human quality ratings on a scale from 1 to 5. Each dimension contains 470 ratings from independent double review of 235 sentence pairs.

of 3 or below constituted 7.4%, 10.6%, and 7.4% of the respective individual reviewer ratings, rather than those percentages of sentence pairs. The review was diagnostic, not a filtering pass: no pairs were removed on the basis of these scores. Consequently, the benchmark retains items that received low ratings, and quality statistics describe the reviewed sample rather than every pair.

3.4.1 Inter-Annotator Agreement

We report Krippendorff's ordinal α (Krippendorff, 2004; Artstein and Poesio, 2008) and Gwet's AC_2 with quadratic weights (Gwet, 2014). Their chance corrections and weights differ, so we interpret both alongside exact agreement and agreement within one scale point. Appendix C provides definitions and descriptive rates.

Overall AC_2 is 0.930 for authenticity, 0.852 for adequacy, and 0.899 for fluency; corresponding α values are 0.190, 0.085, and 0.072 (Table 4). Within-one-point agreement ranges from 90.0% to 100.0% for Najdi, Northern, Eastern, and Southern, but from 65.8% to 73.4% for Hijazi. Exact agreement is also lower for Hijazi.

Concentrated ratings can yield low chance-corrected reliability despite high observed agreement (Gwet, 2008). For Northern authenticity, reviewers agree exactly on 19 of 20 items, with $AC_2 = 0.997$ and $\alpha = 0.000$. This illustrates sensitivity to rating distributions, but does not establish uniform reliability. In particular, Hijazi's lower descriptive agreement indicates substantive disagreement that merits further review.

3.5 Dataset Statistics

Table 5 reports collection sizes, mean lengths, and lexical diversity for each evaluation split. Across the 610 unique pairs, mean lengths are 14.5 Arabic and 22.6 English orthographic tokens; medians are 7 and 11. Arabic lengths range from 1 to 177 tokens, while English lengths range from 1 to 232 tokens. Some entries therefore extend beyond short utterances.

Length differs substantially across the collections. Mean Arabic length ranges from 3.8 tokens in Southern Jazan to 56.3 in Hijazi Urban. Even within Makkah, Hijazi Urban averages 56.3 Arabic tokens compared with 12.2 for Hijazi Makkah. In Jeddah, the corresponding means are 28.9 for Hijazi Urban Jeddah and 11.6 for Hijazi Jeddah. These differences reinforce the interpretation of the split names as creator collections: shared geography does not imply comparable source length or discourse structure.

We measure lexical diversity using HD-D with a fixed sample size of 42 tokens (McCarthy and Jarvis, 2010). Arabic split scores range from 92.7 to 96.1 and English scores from 83.3 to 88.3 on the reported 0–100 scale. This statistic estimates

Group	N	Authenticity		Adequacy		Fluency	
		AC_2	α	AC_2	α	AC_2	α
Najdi	59	0.988	−0.054	0.964	0.223	0.994	−0.035
Northern	20	0.997	0.000	0.903	0.063	0.923	0.289
Eastern	20	0.974	−0.083	0.941	0.098	0.970	−0.182
Southern	57	0.984	0.498	0.921	0.361	0.951	0.395
Hijazi	79	0.776	0.050	0.637	−0.176	0.727	−0.182
Overall	235	0.930	0.190	0.852	0.085	0.899	0.072

Table 4: Inter-annotator agreement by regional group and dimension. AC_2 is Gwet's agreement coefficient with ordinal weights; α is Krippendorff's ordinal alpha.

Evaluation split	N	Mean length (SD)		HD-D $\times 100$	
		Arabic	English	Arabic	English
Eastern Urban	54	5.6 (1.4)	8.9 (2.6)	94.2	84.1
Hijazi Makkah	49	12.2 (5.9)	18.4 (9.3)	94.0	87.0
Hijazi Urban	50	56.3 (38.0)	85.0 (55.4)	96.1	87.2
Hijazi Urban Jeddah	51	28.9 (20.6)	42.9 (29.4)	94.8	86.5
Hijazi Jeddah	51	11.6 (4.8)	22.8 (9.7)	95.0	88.1
Najdi Qasim	51	19.9 (27.2)	32.8 (44.9)	95.9	88.3
Najdi Riyadh	51	10.0 (7.6)	16.3 (12.8)	95.5	87.3
Najdi Urban	52	4.3 (1.6)	6.3 (2.6)	92.7	85.4
Northern Hail	50	5.9 (1.4)	8.3 (2.4)	93.4	85.0
Southern Qahtan	53	7.5 (4.7)	11.9 (7.5)	94.5	86.4
Southern Jazan	50	3.8 (1.2)	6.6 (2.3)	94.6	83.3
Southern Asiri	50	8.6 (8.6)	12.4 (11.7)	93.2	87.2
Unique pairs	610	14.5 (20.8)	22.6 (31.5)	96.5	88.0

Table 5: Split sizes, orthographic token lengths, and lexical diversity. Split rows count collection entries; the final row describes the 610 unique pairs pooled across the benchmark. One pair has membership in three collections. SD denotes sample standard deviation. HD-D uses 42-token samples.

the expected type–token ratio under fixed sampling, reducing the direct dependence on collection size compared with an unadjusted type–token ratio. It characterizes the sampled texts under the specified tokenization.

Tokens are orthographic units obtained after Unicode NFC normalization, with punctuation excluded and Arabic diacritics retained. We apply no Arabic clitic segmentation or lemmatization. Appendix D provides the full convention, counting details, and length quartiles. These statistics contextualize the scores without isolating the effects of length or vocabulary on translation quality.

4 Experimental Setup

4.1 Evaluated Models

We evaluate twelve systems in three families. The **closed-weight LLMs** are GPT-5 (OpenAI, 2025), Gemini 2.5 Pro (Gemini Team, Google, 2025), and Claude Sonnet 4 (Anthropic, 2025). The **open-weight LLMs** are ALLaM-2-7B (Bari et al., 2024), SILMA-9B-Instruct-v1.0 (SILMA AI, 2024), Qwen2.5-7B-Instruct (Qwen Team, 2025), and Falcon3-10B-Instruct (Falcon-LLM Team, 2024); several target Arabic usage. The **dedicated MT systems** are NAMAA-MT-Saudi2English (NAMAA-MT),[3] NLLB-200-3.3B (NLLB Team, 2022), MADLAD-400-3B (Kudugunta et al., 2023), OPUS-MT for Arabic–English (Tiedemann and Thottingal, 2020), and M2M-100-418M (Fan et al., 2021).

4.2 Translation Protocol

We evaluate Saudi→English translation without in-context examples or adaptation on SAFIR. General-purpose LLMs receive a fixed instruction to return only an English translation of the supplied Saudi sentence. The verbatim prompt is in Appendix A. Open-weight LLMs and dedicated MT systems run locally using Hugging Face `transformers`; the sequence-to-sequence models use a maximum generation length of 256 tokens. This limit may affect the longest entries and is part of the evaluated setup.

[3] https://huggingface.co/Namaa-space/NAMAA-MT-Saudi2English

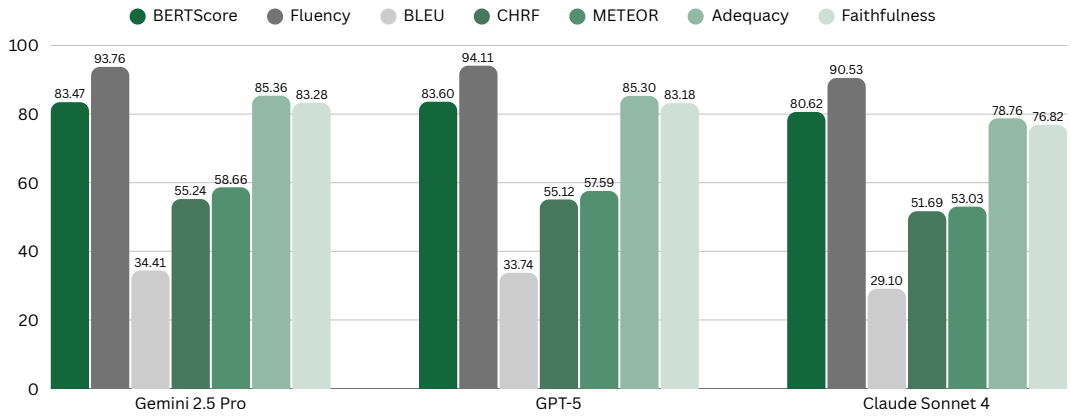


Figure 2: Reported aggregate performance of the three closed-weight LLMs across SAFIR's 12 evaluation splits. Metrics are on a 0–100 scale; overall BLEU is the mean of the split-level scores.

4.3 Automatic Metrics

We use Hugging Face `evaluate` to compute BLEU (Papineni et al., 2002), chrF (Popović, 2015), METEOR (Banerjee and Lavie, 2005), and BERTScore (Zhang et al., 2020). chrF uses the default character n-gram order, without the word n-grams of chrF++. BERTScore F_1 uses `deberta-base-mnli` (He et al., 2021), `lang=en`, and no baseline rescaling. All scores are reported on a 0–100 scale.

Overall BLEU is the unweighted arithmetic mean of the twelve split-level BLEU scores, so each split contributes equally. It is not corpus BLEU computed over pooled predictions. Our use of `evaluate` BLEU rather than sacreBLEU (Post, 2018) also means that comparisons with other papers require matching implementations and aggregation conventions.

4.4 LLM-as-a-Judge

To complement reference-based metrics, we query GPT-4o (OpenAI, 2024) through the OpenAI Batch API for each source, hypothesis, and reference triple. The prompt requests JSON scores from 0 to 100 for adequacy (meaning preservation), faithfulness (omissions and additions), fluency (English naturalness), and holistic overall quality. We report the first three dimensions; Appendix B reproduces the prompt.

These judgments are evaluator-dependent estimates. We neither calibrate them against human ratings of model outputs nor test alternative judges. The human review in Section 3.4 evaluates the benchmark references and does not validate the model-output judge. Close differences between systems should therefore be interpreted cautiously.

5 Results and Analysis

We present aggregate results by model family and examine variation across evaluation splits. Appendix E consolidates all twelve systems in Table 14; selected split comparisons appear below.

5.1 Closed-Weight LLMs

Figure 2 summarizes the three closed-weight LLMs. Gemini 2.5 Pro obtains the highest average BLEU (34.41), chrF (55.24), and METEOR (58.66), while GPT-5 obtains the highest BERTScore (83.60). Their judged adequacy scores are close: 85.36 and 85.30, respectively. Claude Sonnet 4 obtains 29.10 BLEU and 78.76 adequacy. These point estimates do not establish statistically significant differences.

Rankings change across collections (Table 6). GPT-5 leads BLEU on Eastern Urban (58.20), while Gemini leads on Najdi Qasim (39.59) and Hijazi Jeddah (28.52). Claude leads the three models on Southern Qahtan, scoring 47.68 BLEU against GPT-5's 42.98 and Gemini's 43.28. An aggregate ranking therefore does not identify the strongest model on every split.

Reference overlap and judged adequacy can also diverge. Southern Asiri has the lowest adequacy for each closed-weight model (57.10 to 63.20), despite BLEU scores of 27.60 to 34.98. Conversely, Hijazi Urban Jeddah receives adequacy scores of 91.50 to 92.36 with BLEU of 21.76 to 26.58. These cases show that the evaluation dimensions characterize different aspects of output quality; they do not isolate the linguistic cause of either pattern.

The adequacy contrast between Eastern Urban and Southern Asiri is large for each of the three models, ranging from 29.20 to 32.10 points. However,

Evaluation split	Claude Sonnet 4		GPT-5		Gemini 2.5 Pro	
	BLEU	Adeq.	BLEU	Adeq.	BLEU	Adeq.
Eastern Urban	48.33	86.30	58.20	93.00	51.06	93.70
Hijazi Makkah	41.18	90.51	47.30	92.65	42.33	91.73
Hijazi Urban	34.98	90.76	37.43	92.90	31.36	92.30
Southern Qahtan	47.68	81.60	42.98	88.20	43.28	87.90
Southern Jazan	31.04	61.70	39.82	72.50	41.67	75.70
Najdi Qasim	25.22	83.50	26.95	86.60	39.59	88.50
Najdi Riyadh	21.60	72.00	31.77	82.80	29.16	81.80
Najdi Urban	18.46	67.40	24.02	85.61	29.28	85.10
Hijazi Urban Jeddah	21.76	91.50	26.58	91.67	26.34	92.36
Northern Hail	15.23	82.10	16.51	85.51	19.41	85.10
Southern Asiri	27.60	57.10	34.98	63.20	30.96	61.60
Hijazi Jeddah	16.17	80.70	18.28	89.00	28.52	88.50

Table 6: BLEU and GPT-4o-judged adequacy (Adeq.) for the three closed-weight LLMs across SAFIR's 12 evaluation splits. All scores are reported on a scale from 0 to 100.

even this consistent contrast concerns two particular collections and a single judge. A different contrast appears on Hijazi Jeddah: BLEU spans 16.17 to 28.52, while adequacy remains between 80.70 and 89.00. Describing a split as universally "easy" or "hard" would therefore obscure both the evaluation dimension and differences between systems.

5.2 Open-Weight LLMs

ALLaM-2-7B leads this family on average BLEU (22.77), adequacy (69.85), and faithfulness (66.72), followed by SILMA and Qwen (Table 7). Falcon3-10B trails on these measures, obtaining 9.97 BLEU and 37.88 adequacy. SILMA has the highest judged fluency in the family (93.06), illustrating that natural English does not necessarily preserve the source meaning.

The leading model also varies by split (Table 8). ALLaM leads the selected comparisons on Eastern Urban, Najdi Urban, and Hijazi Jeddah. SILMA leads on Hijazi Makkah and Southern Qahtan, while Qwen leads on Southern Asiri. These observations describe the evaluated collections; they do not establish a general advantage from Arabic specialization or identify which training data explain the differences.

The selected comparisons also distinguish aggregate strength from local advantages. ALLaM's Eastern Urban BLEU is 40.21, compared with SILMA's 34.01 and Qwen's 31.61. On Southern Asiri, the order reverses: Qwen scores 23.16, SILMA 22.47, and ALLaM 17.17. Falcon is lowest on each listed split. Thus, ALLaM's aggregate lead coexists with local advantages for other open-weight systems.

5.3 Dedicated MT Models

NAMAA-MT has the highest scores among the five dedicated systems on all reported metrics, including 18.53 BLEU and 65.05 adequacy (Table 9). It outperforms NLLB-200, MADLAD-400, OPUS-MT, and M2M-100 under this setup. Although NAMAA-MT targets Saudi Arabic, the comparison does not isolate specialization from other system differences.

Its aggregate advantage does not hold on every collection. NAMAA-MT leads the selected dedicated systems on Eastern Urban and Hijazi Makkah, with BLEU scores of 28.36 and 32.41. On Hijazi Jeddah, OPUS-MT obtains 14.65 BLEU compared with NAMAA-MT's 8.13 and NLLB's 3.89. This is another instance in which the overall ordering masks a different local ordering.

M2M-100 obtains 0.00 BLEU on Najdi Urban alongside a fluency score of 67.70 (Table 10). Zero BLEU does not imply zero word overlap: without smoothing, missing higher-order n-gram matches can produce a zero score. Similarly, OPUS-MT scores 1.84 BLEU and 62.70 fluency on Hijazi Makkah. Fluency concerns English naturalness and cannot establish whether the source meaning is preserved.

5.4 Cross-Family Findings

The closed-weight LLMs obtain the strongest aggregate BLEU, adequacy, and faithfulness scores. The remaining families do not form a strict hierarchy: NAMAA-MT exceeds Falcon3-10B on both BLEU and adequacy. The comparison does not isolate the effects of scale, architecture, pretraining coverage, or specialization.

Performance also depends on the sampled collec-

Model	BLEU	chrF	METEOR	BERTScore	Adeq.	Faith.	Flu.
ALLaM-2-7B	22.77	44.87	48.28	78.17	69.85	66.72	82.76
SILMA-9B-Instruct-v1.0	21.80	42.81	44.72	77.00	66.90	63.62	93.06
Qwen2.5-7B-Instruct	19.75	42.01	43.05	75.96	65.55	62.42	89.54
Falcon3-10B-Instruct	9.97	30.22	30.03	68.90	37.88	33.28	83.10

Table 7: Reported aggregate performance of the open-weight LLMs. Adeq., Faith., and Flu. denote GPT-4o-judged adequacy, faithfulness, and fluency.

Evaluation split	ALLaM BLEU	SILMA BLEU	Qwen BLEU	Falcon BLEU
Eastern Urban	40.21	34.01	31.61	11.19
Hijazi Makkah	33.75	34.47	31.15	18.19
Najdi Urban	20.50	18.54	12.64	6.87
Southern Qahtan	33.51	34.86	32.79	17.34
Hijazi Jeddah	11.51	8.15	8.10	4.10
Southern Asiri	17.17	22.47	23.16	13.93

Table 8: BLEU on selected evaluation splits for the four open-weight LLMs. These collections illustrate changes in model ranking; all twelve splits contribute to the aggregate results.

tion and evaluation dimension. Southern varieties show pronounced difficulties in judged adequacy, including the Southern Asiri pattern above. At the same time, creator choices, text length, and reference wording differ across splits. The results support multidimensional reporting for the tested systems and collections, while leaving explanations based on specific linguistic features or pretraining representation to further study.

6 Error Analysis

We examine four illustrative outputs to distinguish errors that aggregate metrics can obscure. The examples provide qualitative evidence, rather than estimates of category frequency or comparisons of error rates between model families. They also do not determine the causes of Southern performance gaps.

Pragmatic intent drift. For the Hijazi source إشبك بشويش ليش متصربع, the output *"What's wrong? Why are you nervous?"* differs from the reference *"Hey, take it easy. What's the big hurry?"* The output retains a question about the addressee but loses the calming directive and shifts the interpretation toward an emotional state.

Speech-act change and invented content. For Hijazi Jeddah يا بنت ازهمي أخوكي خليه يجي يتعشا, the reference is *"Girl, call your brother and have him come eat dinner."* The output *"Oh, my sister's son has arrived to visit."* changes an instruction into a statement about a past event, invents a family relationship, and omits the request to call someone for dinner.

Wrong target language and idiom failure. Najdi Qasim كفيت ووفيت has the reference *"You've done more than enough."* The output *"C'est tout, j'ai fini."* is in French and fails to convey the expression's appreciative meaning. This combines failure to follow the requested output language with loss of idiomatic intent.

Lexical substitution and omission. For Eastern Urban وين الدريشة أبغى أفتحها يدخل الهوا, the reference is *"Where's the window? I want to open it for some air."* The output *"Where's the dryer?"* substitutes "dryer" for الدريشة (window) and omits the second clause. The result is fluent but both inaccurate and incomplete.

These examples motivate separating lexical errors, omissions, and pragmatic or speech-act changes. A systematic annotation of a larger, model-attributed sample is needed to measure their distribution across varieties.

7 Conclusion

SAFIR contributes 610 unique Saudi–English sentence pairs across five regional groups and twelve creator-specific evaluation splits. Native-speaker authorship and translation, together with double review of 235 pairs, support evaluation of the sampled collections. The twelve-system comparison shows strong aggregate performance from closed-weight LLMs, variation in rankings across splits, and differences between reference overlap and judged quality. Southern varieties are particularly challeng-

Model	BLEU	chrF	METEOR	BERTScore	Adeq.	Faith.	Flu.
NAMAA-MT	18.53	41.46	46.60	76.58	65.05	61.36	78.45
NLLB-200-3.3B	14.25	35.45	36.69	72.33	48.96	45.20	77.92
MADLAD-400-3B	11.33	31.25	33.94	69.60	45.03	41.46	76.93
OPUS-MT	10.99	31.34	32.19	68.93	45.10	41.86	69.41
M2M-100-418M	6.82	25.82	26.85	65.04	34.62	32.69	57.87

Table 9: Reported average performance of dedicated MT systems across the 12 evaluation splits. All scores are on a scale from 0 to 100. Adeq., Faith., and Flu. denote GPT-4o-judged adequacy, faithfulness, and fluency, respectively. OPUS-MT denotes `Helsinki-NLP/opus-mt-tc-big-ar-en`.

Evaluation split	NAMAA-MT	OPUS-MT	NLLB	M2M BLEU	M2M Flu.
Eastern Urban	28.36	9.21	19.47	8.09	72.90
Hijazi Makkah	32.41	1.84	27.14	17.30	65.10
Najdi Riyadh	18.44	7.91	8.65	3.59	47.80
Najdi Urban	10.98	6.00	8.79	0.00	67.70
Hijazi Jeddah	8.13	14.65	3.89	2.61	39.90

Table 10: BLEU on selected splits for four dedicated MT systems. The final column reports M2M-100 judged fluency. NLLB and M2M denote NLLB-200-3.3B and M2M-100-418M, respectively.

ing under judged adequacy. These findings motivate broader Saudi coverage and evaluation by variety and quality dimension, without attributing the observed differences to unmeasured linguistic or training-data factors.

Limitations

SAFIR is a focused evaluation benchmark with limited coverage and uneven representation across regional groups. Its 610 unique pairs do not support population-level generalization to all Saudi speakers or localities. Each split represents one creator, so creator style, text length, and reference wording are confounded with local variety. Some references were prepared with permitted MT assistance, and only 235 pairs received independent double review; low-rated items were retained. The study reports point estimates without confidence intervals or significance tests, so close rankings should not be treated as reliable performance differences. GPT-4o judgments are not validated against human ratings of model outputs, and their robustness to judge choice is untested. Long entries may be affected by generation limits. Findings concern the tested systems and the zero-shot Saudi→English setting. Broader sampling, additional human review, systematic error annotation, judge validation, reverse-direction translation, few-shot prompting, and adaptation remain future work.

Ethics Statement

Contributors provided informed consent for participation and research release of their contributions. They authored sentences from their knowledge of local language use, rather than collecting material through automated scraping. The benchmark contains no contributor names, usernames, account identifiers, or other personal metadata, and no personal or sensitive information was included in the sentence pairs. Reviewers did not assess their own material. SAFIR is intended for research on translation of sampled Saudi varieties; its split labels and model scores should not be used to rank communities or make claims about the linguistic quality of their speech.

Acknowledgments

We thank the native-speaker contributors who helped author, translate, and review SAFIR: Hadi Aljader, Rashed Alghuraib, Shatha Alqubaisi, Saud Almutawa, Shaima Hamdan, Shaden Alsuhaim, Siraa Eid Abualari, Shahd Mohammed Al-Dosari, Bushra Abdulkhaliq Bin Da'ajam, Abdulrahman Saad Asiri, and Norah Al-Zaid. We also thank Tuwaiq Academy for supporting this work.

References

Renad Al-Monef, Hassan Alhuzali, Nora Alturayeif, and Ashwag Alasmari. 2026. From words to proverbs: Evaluating llms' linguistic and cultural competence in saudi dialects with absher. *Alexandria Engineering Journal*, 137:25–41.

Anthropic. 2025. Claude 4 system card. https://www.anthropic.com/claude.

Ron Artstein and Massimo Poesio. 2008. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72.

M Saiful Bari, Yazeed Alnumay, Norah A. Alzahrani, and 1 others. 2024. ALLaM: Large language models for arabic and english. *Preprint*, arXiv:2407.15390.

Houda Bouamor and 1 others. 2018. The madar arabic dialect corpus and lexicon. In *LREC*.

Daniel Deutsch and 1 others. 2025. Wmt24++: Expanding the language coverage of wmt24 to 55 languages & dialects.

Abdellah El Mekki and 1 others. 2026. Alexandria: A multi-domain dialectal arabic machine translation dataset for culturally inclusive and linguistically diverse llms. *Preprint*, arXiv:2601.13099.

Falcon-LLM Team. 2024. The Falcon 3 family of open models. https://huggingface.co/tiiuae.

Angela Fan, Shruti Bhosale, Holger Schwenk, and 1 others. 2021. Beyond english-centric multilingual machine translation. *Journal of Machine Learning Research*, 22(107):1–48.

Gemini Team, Google. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *Preprint*, arXiv:2507.06261.

Kilem Li Gwet. 2008. Computing inter-rater reliability and its variance in the presence of high agreement. *British Journal of Mathematical and Statistical Psychology*, 61(1):29–48.

Kilem Li Gwet. 2014. *Handbook of Inter-Rater Reliability: The Definitive Guide to Measuring the Extent of Agreement Among Raters*, 4th edition. Advanced Analytics, LLC, Gaithersburg, MD.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTa: Decoding-enhanced BERT with disentangled attention. In *International Conference on Learning Representations (ICLR)*.

Klaus Krippendorff. 2004. *Content Analysis: An Introduction to Its Methodology*, 2nd edition. Sage Publications, Thousand Oaks, CA.

Sneha Kudugunta, Isaac Caswell, Biao Zhang, and 1 others. 2023. MADLAD-400: A multilingual and document-level large audited dataset. *Preprint*, arXiv:2309.04662.

Philip M. McCarthy and Scott Jarvis. 2010. MTLD, vocd-D, and HD-D: A validation study of sophisticated approaches to lexical diversity assessment. *Behavior Research Methods*, 42(2):381–392.

Karima Meftouh, Salima Harrat, Salma Jamoussi, Mourad Abbas, and Kamel Smaïli. 2015. Machine translation experiments on padic: A parallel arabic dialect corpus. In *PACLIC 29*.

NLLB Team. 2022. No language left behind: Scaling human-centered machine translation.

OpenAI. 2024. GPT-4o system card. *Preprint*, arXiv:2410.21276.

OpenAI. 2025. GPT-5 system card. https://openai.com/index/gpt-5-system-card/.

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 (ACL)*, 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 (WMT)*, pages 392–395.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation (WMT)*, pages 186–191.

Qwen Team. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

SILMA AI. 2024. SILMA: A strong arabic-centric open language model. https://huggingface.co/silma-ai.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT – building open translation services for the world. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation (EAMT)*, pages 479–480.

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 (ICLR)*.

A Translation Prompt

All general-purpose LLMs receive the same fixed *system* message and a per-item *user* message, in which `{text}` is replaced by the Saudi source sentence.

System message

```
You are a professional translator
specializing in Saudi Arabic dialects to
English. Provide only the translation,
nothing else.
```

User message

```
Translate this Saudi Arabic dialect text to
English. Only provide the English
translation, no explanations.

Saudi Arabic: {text}
English:
```

The reported generation configuration specifies `temperature`= 0.1 and `top_p`= 1 for sampling-based LLM generation. Sequence-to-sequence MT models instead use their model-specific translation interface with a maximum generation length of 256 tokens. These are the shared settings recorded for the experiments; complete per-model API versions and decoding configurations are not reported here.

B LLM-as-a-Judge Prompt

The GPT-4o judge (Section 4.4) is queried with a fixed *system* message and a per-item *user* message; the braces in the user message are placeholders filled with each *(source, hypothesis, reference)* triple.

System message

```
You are a bilingual translation evaluator
returning only JSON scores.
```

User message

```
SOURCE (Arabic):
{source}

HYPOTHESIS (Model Translation):
{prediction}

REFERENCE (Gold English):
{reference}

Evaluate the translation on:
  - Adequacy      (meaning preserved)
  - Faithfulness  (no omissions / additions)

  - Fluency       (English grammar / style)
  - Overall       (holistic correctness)

Return JSON:
{"adequacy": 0-100, "faithfulness": 0-100,
 "fluency": 0-100, "overall": 0-100}
```

C Inter-Annotator Agreement: Coefficient Definitions

Krippendorff's α (ordinal). Krippendorff's α (Krippendorff, 2004; Artstein and Poesio, 2008) accommodates ordinal scales, two or more raters, and missing values. It is defined as

$$\alpha = 1 - \frac{D_o}{D_e}, \tag{1}$$

where D_o is the observed disagreement and D_e the disagreement expected by chance:

$$\begin{aligned} D_o &= \frac{1}{n} \sum_c \sum_k o_{ck}\, \delta^2_{ck}, \\ D_e &= \frac{1}{n(n-1)} \sum_c \sum_k n_c\, n_k\, \delta^2_{ck}. \end{aligned} \tag{2}$$

Here o_{ck} is the coincidence count for categories c and k, $n_c = \sum_k o_{ck}$ is the marginal count, and $n = \sum_c n_c$ is the total number of pairable ratings. For the ordinal metric, the difference function is

$$\delta^2_{ck} = \left(\sum_{g=\min(c,k)}^{\max(c,k)} n_g - \frac{n_c + n_k}{2} \right)^2, \tag{3}$$

which penalizes distant disagreements more heavily than adjacent ones.

Gwet's AC_2 (ordinal weights). Gwet's AC_2 (Gwet, 2014) estimates chance agreement from the overall category proportions in a way that remains stable across a wide range of rating distributions. It is defined as

$$AC_2 = \frac{p_a - p_e}{1 - p_e}, \tag{4}$$

with weighted observed and chance agreement

$$\begin{aligned} p_a &= \frac{1}{n} \sum_{i=1}^{n} w_{r_{1i}, r_{2i}}, \\ p_e &= \frac{T_w}{q(q-1)} \sum_{k=1}^{q} \pi_k (1 - \pi_k), \end{aligned} \tag{5}$$

Here n denotes the number of double-rated items, r_{1i}, r_{2i} are the ratings of item i, and $q = 5$ is the number of scale points, π_k is the proportion of all ratings assigned to category k, and $T_w = \sum_{k=1}^{q} \sum_{l=1}^{q} w_{kl}$. We use ordinal (quadratic) weights

$$w_{kl} = 1 - \frac{(k-l)^2}{(q-1)^2}, \tag{6}$$

so that partial credit is awarded to near-misses.

Descriptive agreement. For completeness, Table 11 reports the two model-free agreement statistics referenced in Section 3.4.1. For each regional group and each of the three dimensions, authenticity (Auth.), adequacy (Adeq.), and fluency (Flu.), it gives the percentage of the 235 double-reviewed items on which the two reviewers assign exactly the same score (Exact) and the percentage on which

Group	N	Auth.		Adeq.		Flu.	
		Exact	±1	Exact	±1	Exact	±1
Najdi	59	88.1	98.3	74.6	94.9	91.5	100.0
Northern	20	95.0	100.0	40.0	95.0	60.0	90.0
Eastern	20	80.0	95.0	65.0	90.0	65.0	100.0
Southern	57	89.5	96.5	75.4	91.2	80.7	94.7
Hijazi	79	59.5	73.4	40.5	65.8	54.4	69.6

Table 11: Exact and within-one-point (±1) agreement as percentages of double-reviewed items within each group.

they agree within one scale point (±1). These descriptive figures are interpreted alongside the chance-corrected coefficients of Table 4 in Section 3.4.1.

D Additional Dataset Statistics

The main text reports geographic assignments and split-level summary statistics in Tables 2 and 5. This appendix specifies the counting and tokenization conventions and provides full length distributions and regional summaries.

D.1 Counting and Tokenization

The benchmark contains 610 unique Arabic–English pairs. Table 5 reports entries in each creator collection, preserving the split membership used for evaluation. One identical pair occurs in three collections, so split counts are not additive counts of unique pairs. Pooled statistics in the "Unique pairs" row count each distinct pair once. No split was reassigned on the basis of this descriptive audit.

For length and lexical diversity, we apply Unicode NFC normalization and case-fold Latin text. Tokens are contiguous sequences of letters and numbers, retaining combining marks and internal apostrophes while excluding other punctuation. Curly apostrophes are normalized to straight apostrophes. Arabic diacritics are retained; we apply no clitic segmentation, stemming, or lemmatization. Standard deviations use the sample denominator.

Lexical diversity is measured with HD-D at a fixed sample size of 42 tokens (McCarthy and Jarvis, 2010). It is the expected type–token ratio in a sample drawn without replacement, multiplied by 100. For a collection of N tokens with type frequencies f_v, we compute

$$\text{HD-D}_{42} = \frac{100}{42} \sum_{v} \left[1 - \frac{\binom{N-f_v}{42}}{\binom{N}{42}} \right]. \quad (7)$$

This statistic characterizes the observed collection under the stated tokenization; it is not an intrinsic measure of a dialect's richness.

D.2 Length Distributions and Regional Summaries

Table 12 reports medians, interquartile ranges, and full ranges for both languages. Table 13 summarizes collection entries by regional group. The large within- and between-collection length differences provide context for interpreting split-level scores, but we do not estimate their independent effect on translation performance.

E Consolidated Results

Table 14 places all twelve evaluated systems in a common format. Bold values mark the highest reported score for each metric, without implying statistical significance. The main text provides the closed-weight split breakdown and selected comparisons for open-weight LLMs and dedicated MT systems. Complete split-level results are available on the leaderboard.

Split	Arabic tokens			English tokens		
	Median	Q1–Q3	Min–max	Median	Q1–Q3	Min–max
Eastern Urban	5.5	5–6	2–9	9	7.25–10	2–16
Hijazi Makkah	13	7–15	4–37	19	12–23	6–58
Hijazi Urban	47.5	28.25–75.5	3–177	73	39–122.5	9–232
Hijazi Urban Jeddah	21	11.5–47	5–77	36	16.5–67.5	6–113
Hijazi Jeddah	10	8–15	3–24	21	16–31	5–44
Najdi Qasim	11	7–16	2–106	16	11.5–24.5	5–175
Najdi Riyadh	8	5–12	1–39	13	7.5–20	1–63
Najdi Urban	4	3–5	2–11	6	5–8	1–16
Northern Hail	6	5–7	3–9	8	6.25–10	4–14
Southern Qahtan	6	5–8	3–26	11	7–13	4–47
Southern Jazan	4	3–4.75	2–6	6.5	4.25–8	3–13
Southern Asiri	6.5	1–15.75	1–28	11	2–20.5	1–40

Table 12: Length distributions in each creator collection. Q1 and Q3 are the first and third quartiles. Length is measured using the orthographic tokenization specified in Appendix D.

Group	Entries	Mean length (SD)		Median length		HD-D $\times 100$	
		Arabic	English	Arabic	English	Arabic	English
Eastern	54	5.6 (1.4)	8.9 (2.6)	5.5	9	94.2	84.1
Hijazi	201	27.3 (28.4)	42.3 (41.3)	16	28	96.2	87.7
Najdi	154	11.4 (17.4)	18.4 (28.9)	7	10	96.1	88.4
Northern	50	5.9 (1.4)	8.3 (2.4)	6	8	93.4	85.0
Southern	153	6.7 (6.0)	10.3 (8.5)	5	8	94.5	86.6

Table 13: Regional summaries of collection entries. Entries preserve split membership; they are not deduplicated counts. Lexical diversity is computed after pooling each group's entries, rather than averaging its split scores.

Model	Reference-based metrics				GPT-4o judgments		
	BLEU	chrF	METEOR	BERTScore	Adeq.	Faith.	Flu.
Closed-weight LLMs							
Gemini 2.5 Pro	**34.41**	**55.24**	**58.66**	83.47	**85.36**	**83.28**	93.76
GPT-5	33.74	55.12	57.59	**83.60**	85.30	83.18	**94.11**
Claude Sonnet 4	29.10	51.69	53.03	80.62	78.76	76.82	90.53
Open-weight LLMs							
ALLaM-2-7B	22.77	44.87	48.28	78.17	69.85	66.72	82.76
SILMA-9B-Instruct-v1.0	21.80	42.81	44.72	77.00	66.90	63.62	93.06
Qwen2.5-7B-Instruct	19.75	42.01	43.05	75.96	65.55	62.42	89.54
Falcon3-10B-Instruct	9.97	30.22	30.03	68.90	37.88	33.28	83.10
Dedicated MT models							
NAMAA-MT	18.53	41.46	46.60	76.58	65.05	61.36	78.45
NLLB-200-3.3B	14.25	35.45	36.69	72.33	48.96	45.20	77.92
MADLAD-400-3B	11.33	31.25	33.94	69.60	45.03	41.46	76.93
OPUS-MT (Arabic–English)	10.99	31.34	32.19	68.93	45.10	41.86	69.41
M2M-100-418M	6.82	25.82	26.85	65.04	34.62	32.69	57.87

Table 14: Reported average performance across the 12 evaluation splits. All scores are on a scale from 0 to 100. Adeq., Faith., and Flu. denote adequacy, faithfulness, and fluency, respectively. Bold indicates the highest reported value in each column, without implying statistical significance. NAMAA-MT denotes NAMAA-MT-Saudi2English.

QuranicMMLU: A Cognitively-Aware Benchmark for Evaluating Generative AI Solutions on Quranic Linguistic Knowledge

Rawan El Ghali[1,2*] **Umm Kulsoom**[1,3*] **Anas Madkoor**[1] **Roaa Abdelmagid**[1,6]
Roaa Ibrahim[1,3] **Raghad Mousa**[1,2] **Hamza Aljaji**[1,4]
Abdullah Khanafer[1,5] **Abdallah Alkanani**[1,4] **Ehsaneddin Asgari**[1]

[1]Qatar Computing Research Institute, HBKU [2]Carnegie Mellon University Qatar
[3]University of Doha for Science and Technology [4]Qatar University
[5]American University of Beirut [6]University of Birmingham

easgari@hbku.edu.qa

Abstract

We introduce QuranicMMLU, a benchmark for evaluating generative AI on Quranic Arabic across multiple dimensions of linguistic complexity. Existing Quranic benchmarks center on general question answering and semantic retrieval, without probing specific linguistic competencies or stratifying by cognitive demand and verse difficulty. We construct a five-pillar Quranic taxonomy spanning Phonology, Morphology, Syntax, Semantics, and Pragmatics, with 31 leaves covering phenomena from tajwīd and root-and-pattern morphology to occasions of revelation and inter-surah coherence. For each leaf we generate questions stratified by Bloom's cognitive level and verse perplexity, then have LLM as a judge to independently answer and score every item and route the annotations to manual review. The resulting dataset comprises 980 human-reviewed questions, each issued in both open-ended and multiple-choice form. We benchmark 12 systems on these items and find that the Islamic-specialized model leads, yet every system scores higher on multiple-choice accuracy (average 84%) than open-ended answer quality (average 60%): the two rankings agree closely (Kendall's $\tau = 0.73$), but multiple-choice scoring hides failures that surface only once answer choices are removed. QuranicMMLU thus offers a rigorous, linguistically grounded framework for evaluating Arabic NLP in the Quranic domain.

1 Introduction

The Quran is the primary religious and cultural text for nearly two billion Muslims, and engaging with it meaningfully demands not just reading but a genuine understanding of its language and content. As large language models are increasingly consulted on Islamic topics, including direct questions about the Quran (Abdelaal et al., 2026), the cost of ungrounded answers in this domain is unusually high: hallucinated or inaccurate responses carry real religious consequences (Mubarak et al., 2025). Reliable use of these systems therefore depends on accurate, fine-grained evaluation of how well they actually understand Quranic Arabic.

Existing benchmarks only partially meet this need. Islamic and Quranic QA resources assess factual and reasoning ability (Abdelaal et al., 2026). Three gaps remain. First, no benchmark targets the diversity of *linguistic* competencies, from phonology to pragmatics, that Quranic understanding requires. Second, most datasets are multiple-choice only, a format that can be gamed through answer-choice artifacts (Balepur et al., 2024) and that diverges from how users actually query models. Third, the cognitive demand of questions is rarely measured systematically.

We introduce QuranicMMLU to address these gaps. Our contributions are:

(i) A five-pillar taxonomy of Quranic Arabic spanning Phonology, Morphology, Syntax, Semantics, and Pragmatics, with 31 leaves covering competencies from tajwīd to inter-surah coherence;

(ii) A human-reviewed, retrieval-grounded dataset of 980 questions, each in both open-ended and multiple-choice form and stratified by Bloom's cognitive level (Anderson and Krathwohl, 2001) and verse 'popularity of use', measured by perplexity;

(iii) An evaluation of 12 systems in both open-ended and multiple-choice formats, showing that multiple-choice scoring overstates ability and hides failures that surface only in open-ended use.

The leaderboard and relevant resources are available at https://huggingface.co/spaces/musiml-org/QuranicMMLU.

[*] Equal contribution; joint first authors.

Proceedings of The Fourth Arabic Natural Language Processing Conference (ArabicNLP 2026), pages 302–316
October 28-29, 2026 ©2026 Association for Computational Linguistics

2 Related Work

2.1 Islamic and Quranic QA Benchmarks

Early Quranic QA frames the task as extractive answer-span retrieval (Malhas and Elsayed, 2022; Malhas et al., 2022, 2023; Alnefaie et al., 2023), while recent benchmarks broaden to Islamic knowledge and reasoning through multiple-choice, retrieval-augmented, and shared-task settings (Abdelaal et al., 2026; Bhatia et al., 2026; Bouchekif et al., 2025; Alwajih et al., 2025), and extend to Islamic jurisprudence across the Sunni legal schools (Atif et al., 2025; Bahaj and Ghogho, 2025). These resources target factual and reasoning ability; we instead probe Quranic *linguistic* competencies, from phonology to pragmatics, in both multiple-choice and open-ended formats.

2.2 Arabic Understanding Benchmarks

General Arabic benchmarks such as ArabicMMLU and AlGhafa measure broad competence (Koto et al., 2024; Almazrouei et al., 2023), but not the phonological, morphological, syntactic, and rhetorical phenomena specific to Quranic Arabic that we target.

2.3 Cognitive and Format-Aware Evaluation

Bloom's taxonomy (Krathwohl, 2002) has become a foundational scaffold for stratifying cognitive demands in language model evaluation, moving assessment beyond aggregate accuracy toward diagnosing where competence breaks down across cognition layers. Recent work has demonstrated the applicability of cognitive assessment frameworks across visual understanding (Abootorabi et al., 2026) and relational knowledge in multilingual contexts (Abbasi Liasi et al., 2026). However, most benchmarks remain limited: they concentrate on lower-order cognitive tasks, rely predominantly on multiple-choice formats vulnerable to answer-choice artifacts (Balepur et al., 2024), and lack systematic frameworks for comprehensive cognitive evaluation. To address these gaps, we introduce QuranicMMLU, a benchmark that combines multiple question formats with a multi-judge protocol to systematically evaluate knowledge across cognitive levels and verse-perplexity dimensions.

3 Approach

QuranicMMLU is built through a six-stage pipeline (Figure 1): we define the taxonomy, generate questions per leaf, collect LLM annotations, conduct human review with conflict resolution, benchmark selected models, and score open-ended responses with an LLM judge. Each stage is detailed below.

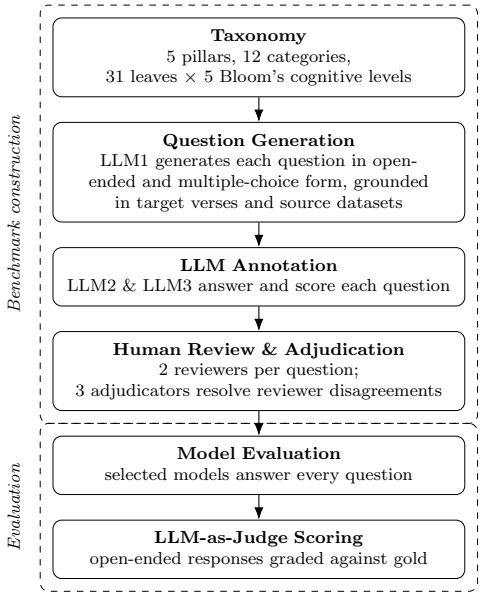


Figure 1: Overview of QuranicMMLU: taxonomy-driven question generation, LLM annotation and human review, then model evaluation with LLM-as-judge scoring. LLM1 is Claude (generator); LLM2 and LLM3 are GPT and Gemini (annotators).

3.1 Taxonomies

QuranicMMLU Taxonomy. We built the taxonomy by studying the literature and key sources for each pillar. We drafted candidate categories and refined them with the help of AI deep-research tools, ensuring broad coverage of the target capability space by including both commonly studied phenomena and less frequently represented but conceptually important aspects. Related items were then grouped into the final taxonomy.

We organize the QuranicMMLU Taxonomy as 5 pillars, 12 categories, and 31 leaves (Figure 5 in Appendix A, p. 11), each leaf a distinct phenomenon, so evaluation is a grid of competencies rather than one aggregate score. Unlike MMLU (Hendrycks et al., 2021) and ArabicMMLU (Koto et al., 2024), which probe subject knowledge, we organize by linguistic ability (Ribeiro et al., 2020), spreading each pillar's leaves across a range of its sub-phenomena.

Phonology covers madd, qalqalah, waqf, and qirāʾāt, probing the sound–meaning mapping that Arabic's undiacritized script makes a hard disambiguation task (Zitouni et al., 2006). *Morphology* spans roots, lemmas, wazn, tense–aspect–mood, voice, number, and clitics, all sources of the

high ambiguity of the root-and-pattern system and grounded in the Quranic Arabic Corpus (Dukes and Habash, 2010). *Syntax* ranges from sentence type and iḍāfa to negation scope, fronting, ellipsis, and disputed i'rāb, reflecting the difficulty of Arabic parsing (Green and Manning, 2010) and grounded in the dependency treebank (Dukes et al., 2010). *Semantics*, the widest pillar, covers lexical ambiguity (furūq, wujūh wa naẓā'ir), figurative language, and surah-level meaning, all open problems in Arabic word-sense disambiguation (Jarrar et al., 2023; Raganato et al., 2017), figurative-language understanding (Tsvetkov et al., 2014), and discourse coherence (Barzilay and Lapata, 2008). *Pragmatics* targets speech acts, asbāb al-nuzūl, and cross-verse reasoning, tasks needing communicative-function recognition (Stolcke et al., 2000) and context beyond span retrieval (Malhas and Elsayed, 2022; Malhas et al., 2022). Appendix G (p. 9) gives one representative question per pillar with its verified Bloom level.

Bloom's Cognitive Level Taxonomy. We stratify questions across five of the six levels of Bloom's taxonomy, Remember, Understand, Apply, Analyze, and Evaluate, omitting Create because generating new Quranic text raises clear ethical concerns (Figure 6 in Appendix A, p. 12). The operational test reviewers use to assign these levels is given in Appendix F (p. 9).

3.2 Datasets

We ground questions and answers in authoritative sources: the Quran.com API for **Phonology**; the Quranic Arabic Corpus for **Morphology** (Dukes and Habash, 2010) and **Syntax** (Dukes et al., 2010); Quranic tafsīr from Tarteel[1] for **Semantics**; and an asbāb al-nuzūl dataset[2] based on *Sahih Asbab al-Nuzul* (Ibrahim Muhammad al-Ali) for **Pragmatics**.

3.3 Question Generation and Annotation

Generation. For each leaf, we use Claude Opus 4.7 to generate questions, reference short answers, and multiple-choice options at each Bloom level. Generation begins by selecting a verse or group of verses containing an instance of the target leaf (for example, a verse with a simile or metaphor when asking about figurative language). We build this pool by searching the grounding datasets for verses that contain an instance of the target phenomenon, at times using AI research tools to help find candidates, and manually reviewing them to confirm the instance is present; generation then draws a verse at random. We also balance questions across verse perplexity (computed using the Gemma-3-12B model), a proxy for the model's familiarity with a verse, aiming for comparable counts per leaf across the low, medium, and high levels; this stratification is applied to single-verse questions rather than whole-surah ones. The relevant verse is included in the prompt (except for surah-level questions) to avoid misretrieval of Quranic text. Each question is produced with a four-layer prompt stack (Appendix B, p. 8; Figure 7) in which every leaf and Bloom level has a specialized prompt, and each item is written to support both open-ended and multiple-choice answering. Of the four options per question, one is correct, two are close distractors, and one is clearly wrong. For both formats, the datasets (Section 3.2) ground every answer in an authoritative source (a retrieval-augmented setup).

LLM annotation. The generated items are then annotated by two LLM judges, GPT-5.2 and Gemini-2.5-Flash (Comanici et al., 2025). Each judge writes an open-ended answer, selects the correct MCQ option, assigns a Bloom level, and scores the question out of five on specificity, answerability, leaf relevance, and clarity (Appendix C, p. 9). Both judges are given full access to the grounding datasets (Section 3.2) and prompted to base answers on them and cite their source otherwise, again making annotation retrieval-augmented rather than memory-dependent. On the multiple-choice answer, the three independently prompted models (the generator and the two judges) show almost-perfect agreement (Fleiss's $\kappa = 0.90$).

Human review and adjudication. Every annotation is then reviewed by two humans per question to reduce individual bias. Reviewers are undergraduate students, fluent Arabic speakers with a background in Islamic knowledge; because they are not religious scholars, every decision is grounded in corpus evidence and cited references rather than personal judgment, following a written guide (Appendix F, p. 9) that ranks sources of truth: corpus evidence first, then established scholarly consensus, then genuine dispute. Reviewers may edit the question, correct its Bloom level, choose among the annotated answers, add a new answer when none is correct, or delete items that are irrelevant,

[1] https://huggingface.co/datasets/tarteel-ai/quran-tafsir

[2] https://github.com/mostafaahmed97/asbab-al-nuzul-dataset

incorrect, or controversial; because we prioritized correctness and Islamic soundness, any item with a suspected hallucination, an unresolved scholarly dispute, or a single flagged concern was discarded when it could not be cleanly fixed. Counting any such edit as a revision, 370 of the 1,086 candidates (34%) were revised, most often a corrected Bloom level (235 items), 106 (10%) were discarded, and 610 were accepted as generated, leaving the final 980. Conflicts were settled by majority verdict among three adjudicators, with 133 items escalated to group discussion. The review interface is shown in Appendix D (p. 9).

Dataset statistics. Figure 4 (Appendix A, p. 10) reports the distribution of the 980 questions, showing the Bloom-level and verse-perplexity composition within each pillar. The dataset spans all five pillars and all five Bloom levels, with Syntax, Semantics, and Morphology the most represented pillars and a deliberate concentration of higher-perplexity verses, so that evaluation is not dominated by verses models have likely memorized.

3.4 Benchmarking

We benchmark the final dataset in both formats, MCQ and open-ended, over the identical item pool, so results are directly comparable across the two. We evaluate two classes of systems: (i) **open-source models**, general-purpose open-weight LLMs including Gemma (Google) (Gemma Team et al., 2025, 2026), Qwen/Qwen-VL (Alibaba) (Yang et al., 2025; Bai et al., 2025), and the Arabic-centric ALLaM (Bari et al., 2024); and (ii) **Islamic models** specialized for Quranic content, namely Fanar-Sadiq, the Islamic-oriented model from Fanar (QCRI) (Fanar Team et al., 2025, 2026), and Ansari (Kadous et al., 2026).

3.5 Metrics

We evaluate the two formats with complementary metrics, all broken down by model and by Bloom level, with full breakdowns in Figure 2. For **open-ended** answers, we use Gemini as an automated grader that scores each response from 0 to 5 given the question, a reference answer, and the model's response (Appendix E, p. 9); we report this rating as a percentage of the 5-point maximum, together with the proportion of responses scoring 80% or higher. For **MCQ**, we report accuracy, the share of questions answered correctly, presented by Bloom level, by taxonomy content type, and by dataset; with four options, random guessing yields roughly 25%.

Model	**MCQ Acc. (%)**	**Open (%)** mean ± SD
Ansari	**96.4**	**93.2** ±21.8
gemma-4-31B	93.3	82.5 ±28.8
Qwen3.6-27B	91.8	62.1 ±37.0
gemma-4-26B-A4B	91.2	75.7 ±32.8
Qwen3.5-35B-A3B	89.7	62.1 ±35.8
Qwen3-235B-A22B	88.3	52.3 ±39.0
Qwen3.5-9B	84.5	53.1 ±36.6
Fanar-Sadiq	80.1	66.7 ±37.0
Qwen3-VL-8B	76.3	41.1 ±37.2
gemma-3-12B	73.2	50.2 ±38.2
Qwen3-VL-4B	69.7	36.0 ±36.4
ALLaM-7B	69.5	46.4 ±39.8

Table 1: Overall benchmarking results: multiple-choice accuracy and open-ended Gemini score, both as percentages, reported as mean ± standard deviation across items and sorted by MCQ accuracy. Per-pillar and per-Bloom breakdowns are in Figure 2.

Finally, to test whether the two formats measure the same underlying ability, we compare model rankings under each using Kendall's τ, where a strong correlation would indicate that both formats probe similar skills.

4 Results

On the **open-ended** evaluation, Gemini judge scores spanned 93.2% for Ansari down to 36.0% for Qwen3-VL-4B (Table 1). Ansari also scored 80% or higher on 91.1% of items, ahead of the next system, gemma-4-31B, which averaged 82.5% and cleared that threshold on 75.6% of items. Among the remaining systems, Fanar-Sadiq placed fourth (66.7%), and the largest model evaluated, Qwen3-235B-A22B, sat in the lower half (52.3%). The same items in **multiple-choice** form produced higher scores for every system: Ansari again placed first (96.4%), followed by gemma-4-31B (93.3%), while ALLaM-7B was the weakest at 69.5%, still well above the 25% chance level, and the seven highest-scoring systems each cleared 84%.

The two orderings were closely aligned (Kendall's $\tau = +0.7273$; 57 of 66 model pairs concordant). Ansari and gemma-4-31B occupied the top two positions in both, while Fanar-Sadiq moved up four places under the open-ended setting (eighth in MCQ, fourth in open-ended). The full per-pillar and per-Bloom breakdown is shown in Figure 2. Across pillars, Phonology yielded the weakest scores for almost every system in both settings, dropping below 34% for most systems on the open-ended questions, whereas Morphology and Semantics were consistently the strongest. Across

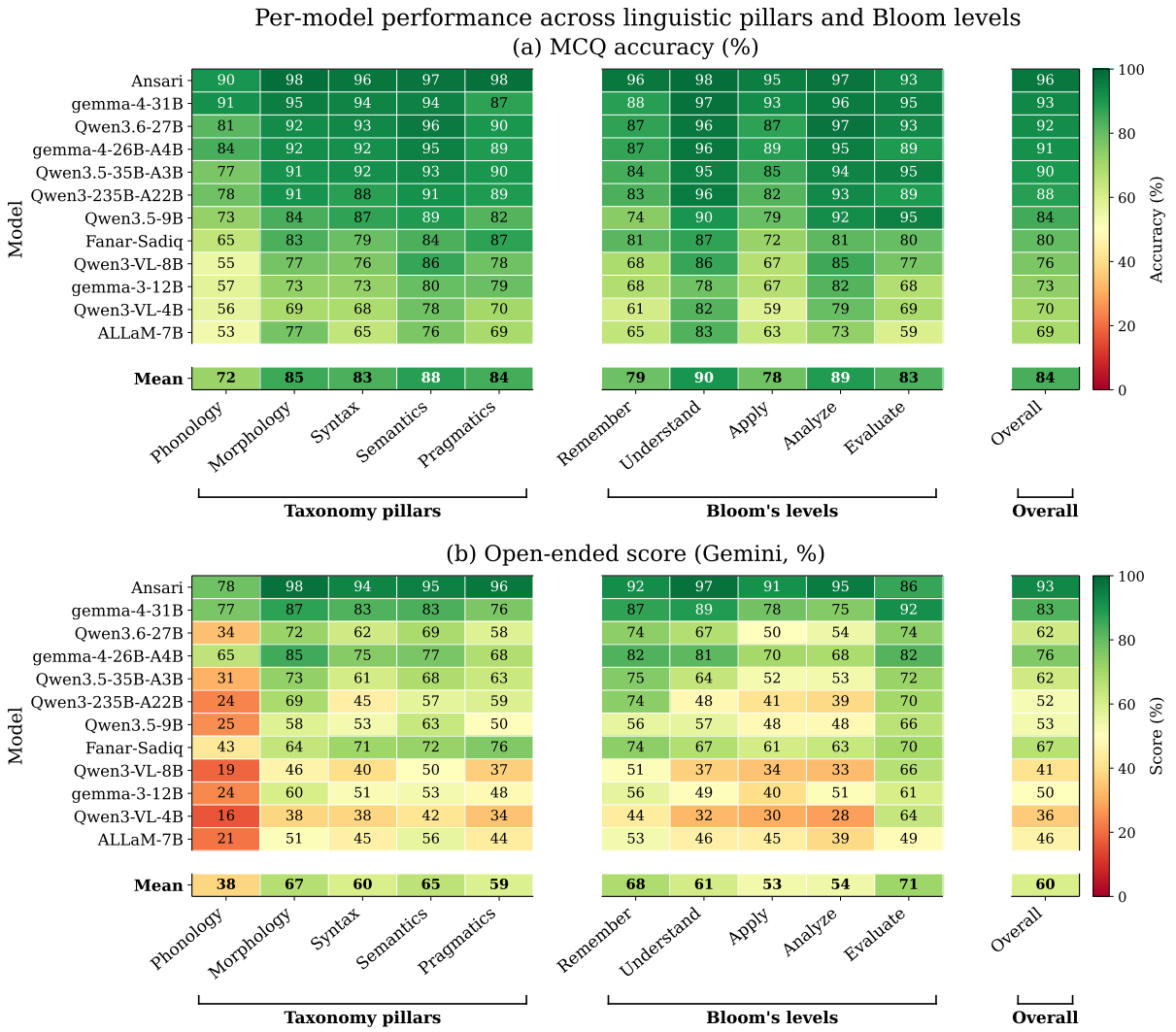


Figure 2: Per-model performance across the five linguistic pillars and the five Bloom levels, for (a) MCQ accuracy and (b) open-ended Gemini score. Models are ordered by overall MCQ accuracy. Phonology is the weakest pillar for nearly every system, especially in the open-ended setting.

Bloom levels, scores peaked at the Understand level and fell to their lowest at Apply and Analyze in both settings.

5 Discussion and Conclusion

Ansari, a system specialized for Islamic and Quranic content, ranked first in both formats, and this domain specialization is reflected directly in our results. The strong rank correlation between formats (Kendall's $\tau = +0.7273$) indicates that, at the level of ordering systems, multiple-choice accuracy and open-ended quality largely track the same underlying ability, which might suggest that MCQ benchmarks could serve as an efficient proxy, though confirming this would require a dedicated error analysis of the LLM-as-judge scores. The agreement is not perfect, however: Fanar-Sadiq rose four places under open-ended grading not because it improved in absolute terms but because other systems degraded more sharply when the answer choices were removed, a gap that pure MCQ evaluation would hide.

The Bloom-level pattern mirrors a familiar profile of model cognition: performance was highest at the Understand level and lowest at Apply and Analyze, suggesting that current systems recognize and explain Quranic phenomena more readily than they apply rules to new cases. Overall, QuranicMMLU provides a fine-grained, linguistically grounded view of model ability on Quranic Arabic and shows that format and cognitive demand both shape measured performance. Future work includes broadening leaf coverage, adding human scoring of open-ended answers, and evaluating further Islamic-specialized systems.

6 Limitations

Our study has several limitations. (i) The taxonomy is designed to be broad and to cover a variety of phenomena within each pillar, but it does not capture every linguistic rule or concept in those pillars. (ii) Open-ended responses are scored solely by an LLM judge without human scoring, which may introduce model bias, including a possible preference for answers resembling a judge's own model family. (iii) We evaluate only open-weight and open-access systems; closed-source commercial models were not assessed because we did not have access to their APIs. (iv) Our benchmark does not yet include every relevant Islamic-specialized system, and some top-ranked models on Islamic-domain leaderboards remain to be evaluated. (v) Questions are generated by a single model (Claude Opus 4.7), so despite human review some generation-model artifacts may remain.

7 Ethical Statement

We note the following ethical considerations. (i) Our human reviewers were not religious scholars; they reviewed questions using the grounding datasets, reference books, online research, and their own judgment. This dataset is intended solely for benchmarking model capabilities and should not be treated as an authoritative source of religious fact. (ii) We omit the Create level of Bloom's taxonomy because generating new Quranic text raises clear ethical concerns.

References

Zohreh Abbasi Liasi, Batoul Hammoud, Anas Madkoor, and Ehsaneddin Asgari. 2026. Reasoning over non-parametric memory in multilingual LLMs: A cognitive-based analysis of relational knowledge. In *Third Conference on Language Modeling*.

Ali Abdelaal, Mohammed Nader Al Haffar, Mahmoud Fawzi, and Walid Magdy. 2026. Islamicmmlu: A benchmark for evaluating llms on islamic knowledge. *Preprint*, arXiv:2603.23750.

Mohammad Mahdi Abootorabi, Omid Ghahroodi, Anas Madkoor, Marzia Nouri, Doratossadat Dastgheib, and Ehsaneddin Asgari. 2026. Almieyar-oryx-bloombench: A bilingual multimodal benchmark for cognitively informed evaluation of vision-language models. In *Findings of the Association for Computational Linguistics: ACL 2026*.

Ebtesam Almazrouei, Ruxandra Cojocaru, Michele Baldo, Quentin Malartic, Hamza Alobeidli, Daniele Mazzotta, Guilherme Penedo, Giulia Campesan, Mugariya Farooq, Maitha Alhammadi, Julien Launay, and Badreddine Noune. 2023. AlGhafa evaluation benchmark for Arabic language models. In *Proceedings of ArabicNLP 2023*, pages 244–275, Singapore (Hybrid). Association for Computational Linguistics.

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, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

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. 2025. 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.

Lorin W. Anderson and David R. Krathwohl, editors. 2001. *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*, complete edition. Longman, New York.

Farah Atif, Nursultan Askarbekuly, Kareem Darwish, and Monojit Choudhury. 2025. Sacred or synthetic? evaluating LLM reliability and abstention for religious questions. *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, 8(1):217–226.

Adil Bahaj and Mounir Ghogho. 2025. MizanQA: Benchmarking large language models on Moroccan legal question answering. *arXiv preprint arXiv:2508.16357*.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, and 45 others. 2025. Qwen3-vl technical report. *Preprint*, arXiv:2511.21631.

Nishant Balepur, Abhilasha Ravichander, and Rachel Rudinger. 2024. Artifacts or abduction: How do LLMs answer multiple-choice questions without the question? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10308–10330, Bangkok, Thailand. Association for Computational Linguistics.

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.

Regina Barzilay and Mirella Lapata. 2008. Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34(1):1–34.

Gagan Bhatia, Hamdy Mubarak, Mustafa Jarrar, George Mikros, Fadi Zaraket, Mahmoud Alhirthani, Mutaz Al-Khatib, Logan Cochrane, Kareem Darwish, Rashid Yahiaoui, and Firoj Alam. 2026. From rag to agentic rag for faithful islamic question answering. *Preprint*, arXiv:2601.07528.

Abdessalam Bouchekif, Samer Rashwani, Emad Soliman Ali Mohamed, Mutaz Alkhatib, Heba Sbahi, Shahd Gaben, Wajdi Zaghouani, Aiman Erbad, and Mohammed Ghaly. 2025. QIAS 2025: Overview of the shared task on islamic inheritance reasoning and knowledge assessment. In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 851–860, Suzhou, China. Association for Computational Linguistics.

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 3416 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *Preprint*, arXiv:2507.06261.

Kais Dukes, Eric Atwell, and Abdul-Baquee M. Sharaf. 2010. Syntactic annotation guidelines for the Quranic Arabic dependency treebank. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Kais Dukes and Nizar Habash. 2010. Morphological annotation of Quranic Arabic. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Fanar Team, Ummar Abbas, Mohammad Shahmeer Ahmad, Minhaj Ahmad, Abdulaziz Al-Homaid, Anas Al-Nuaimi, Enes Altinisik, Ehsaneddin Asgari, Sanjay Chawla, Shammur Chowdhury, Fahim Dalvi, Kareem Darwish, Nadir Durrani, Mohamed Elfeky, Ahmed Elmagarmid, Mohamed Eltabakh, Asim Ersoy, Masoomali Fatehkia, Mohammed Qusay Hashim, and 18 others. 2026. Fanar 2.0: Arabic generative ai stack. *Preprint*, arXiv:2603.16397.

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.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, Mayank Chaturvedi, Aditya Chawla, Victor Cotruta, Alice Coucke, Phil Culliton, Robert Dadashi, Lucas Dixon, Mohamed Elhawaty, Utku Evci, and 304 others. 2026. Gemma 4 technical report. *Preprint*, arXiv:2607.02770.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Spence Green and Christopher D. Manning. 2010. Better Arabic parsing: Baselines, evaluations, and analysis. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 394–402.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations (ICLR)*.

Mustafa Jarrar, Sanad Malaysha, Tymaa Hammouda, and Mohammed Khalilia. 2023. SALMA: Arabic sense-annotated corpus and WSD benchmarks. In *Proceedings of ArabicNLP 2023*, pages 359–369.

M Waleed Kadous, Amr Elsayed, Abdullah Al Nahas, and Ashraf Haress. 2026. Ansari: A retrieval-grounded islamic ai assistant – architecture, deployment, and lessons from 140,000 conversations. *Preprint*, arXiv:2608.20390.

Fajri Koto, Haonan Li, Sara Shatnawi, 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*, pages 5622–5640, Bangkok, Thailand. Association for Computational Linguistics.

David R. Krathwohl. 2002. A revision of Bloom's taxonomy: An overview. *Theory Into Practice*, 41(4):212–218.

Rana Malhas and Tamer Elsayed. 2022. Arabic machine reading comprehension on the holy qur'an using cl-arabert. *Information Processing & Management*, 59(6):103068.

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 Mohamed Darwish, and Walid Magdy. 2025. IslamicEval 2025: The first shared task of capturing LLMs hallucination in islamic content. In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 480–493, Suzhou, China. Association for Computational Linguistics.

Alessandro Raganato, Jose Camacho-Collados, and Roberto Navigli. 2017. Word sense disambiguation: A unified evaluation framework and empirical comparison. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 99–110.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–374.

Yulia Tsvetkov, Leonid Boytsov, Anatole Gershman, Eric Nyberg, and Chris Dyer. 2014. Metaphor detection with cross-lingual model transfer. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 248–258.

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.

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.

Appendices

The appendices follow the order of the pipeline in Section 3.1ff: the taxonomy (A, p. 8), question generation (B, p. 8), the quality rubric (C, p. 9) and review interface (D, p. 9) used for annotation, the open-ended judge prompt (E, p. 9), the Bloom-level verifier guidelines (F, p. 9), and representative questions per pillar (G, p. 9).

A QuranicMMLU Taxonomy

Figure 5 shows the full taxonomy: 5 pillars, 12 categories, and 31 leaves. Each pillar maps to a classical discipline of Quranic study, each category groups related phenomena, and each leaf defines a single competency that questions target. Figure 6 shows the five Bloom's cognitive levels along which questions are stratified, and Figure 3 the overall Bloom-level composition of the dataset. Figure 4 breaks that composition down by pillar, together with the verse-perplexity classes.

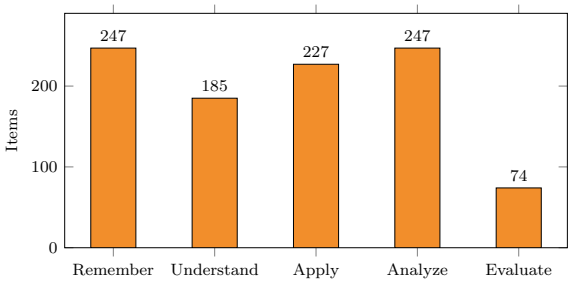


Figure 3: Overall Bloom-level distribution across all 980 items.

B Question-Generation Prompt Pipeline

Figure 7 shows the hierarchical prompt stack used to compose each benchmark question from the taxonomy leaf, Bloom level, and verse context.

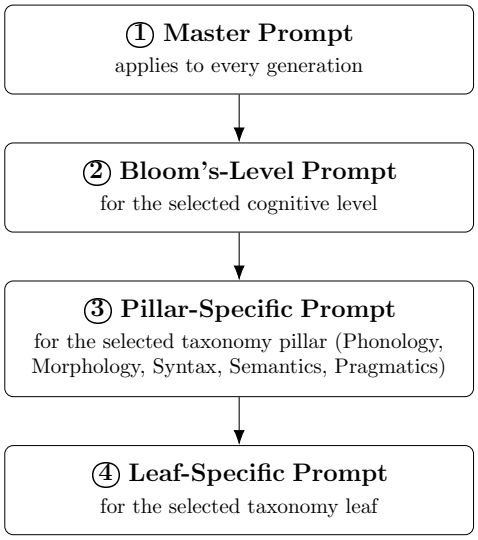


Figure 7: Hierarchical prompt-composition pipeline used to generate each benchmark question.

C Question-Quality Rubric

Every generated question is rated 1–5 on four criteria to guide human review (Figure 8).

D Human Review Interface

Figure 9 shows the custom web interface used for human review and adjudication. For each item, reviewers see the question with the two LLM judges' answers compared against the generator's reference (Figure 9a), the grounding evidence drawn from the source datasets (Figure 9b), and an adjudication panel for recording the final decision (Figure 9c).

E LLM-Judge Scoring Prompt

Open-ended answers are scored automatically by an LLM judge against the reference answer, using the prompt in Figure 10.

F Verifier Guidelines for Bloom Levels

Reviewers assign each Bloom level by the cognitive operation the question demands of the solver, not by how advanced its topic sounds.

Verifier guidelines for assigning Bloom levels

Remember: Recall an explicit fact from the verse or a standard reference, such as a root or a gloss.

Understand: Explain or paraphrase a meaning or grammatical function in one's own words.

Apply: Use a known rule or procedure on a specific instance, such as applying a tajwīd or grammar rule to a given word.

Analyze: Break the verse down and derive a relationship or structure that is not stated, over more than one step.

Evaluate: Weigh competing valid readings and justify which is stronger.

The Bloom level was the most frequently corrected field during human review (235 items; Section 3.3), so the generated labels were verified and corrected rather than accepted as produced.

G Representative Questions by Pillar

One representative question per pillar, with its human-verified Bloom level. Arabic is transliterated; each expected answer follows the arrow.

One question per pillar with its verified Bloom level

Morphology (Remember): What is the trilateral root of the verb ja'ala? → j-'-l.

Semantics (Understand): Explain the difference in meaning between al-raḥmān and al-raḥīm in the basmala. → Both derive from the root r-ḥ-m (mercy); al-raḥmān is the intensive form denoting all-encompassing mercy to every creature, while al-raḥīm denotes sustained, particular mercy.

Phonology (Apply): Apply the nūn sākinah rules to hudan min rabbihim (Q 2:5): what ruling governs the nūn of min before rabbihim, and why? → Idghām (assimilation) into the rā' with no ghunnah, since rā' is an idghām letter that carries no nasalization.

Syntax (Analyze): In iyyāka na'budu (Q 1:5), identify the role of iyyāka and explain what its position contributes. → It is a fronted direct object (maf'ūl bihi muqaddam); placing the object before the verb restricts worship to God alone.

Pragmatics (Evaluate): In Q 3:7, assess the two accepted stops, pausing at illā Allāh versus continuing to wa-l-rāsikhūna fī l-'ilm, and what each implies. → The first reading restricts knowledge of the mutashābih to God, the second extends it to those firmly grounded in knowledge; both are grammatically valid and attested.

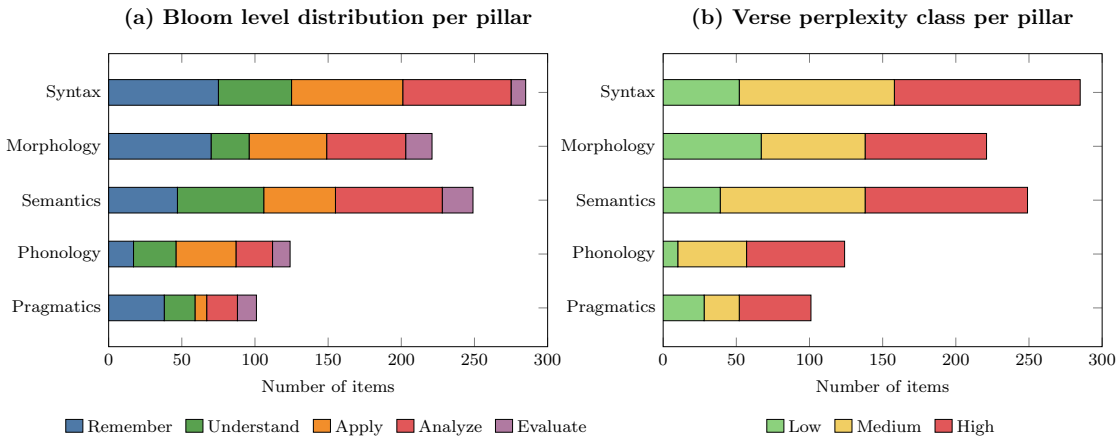


Figure 4: Dataset composition: (a) Bloom-level distribution per pillar and (b) verse-perplexity class per pillar. The overall Bloom-level distribution across all 980 items is given in Figure 3.

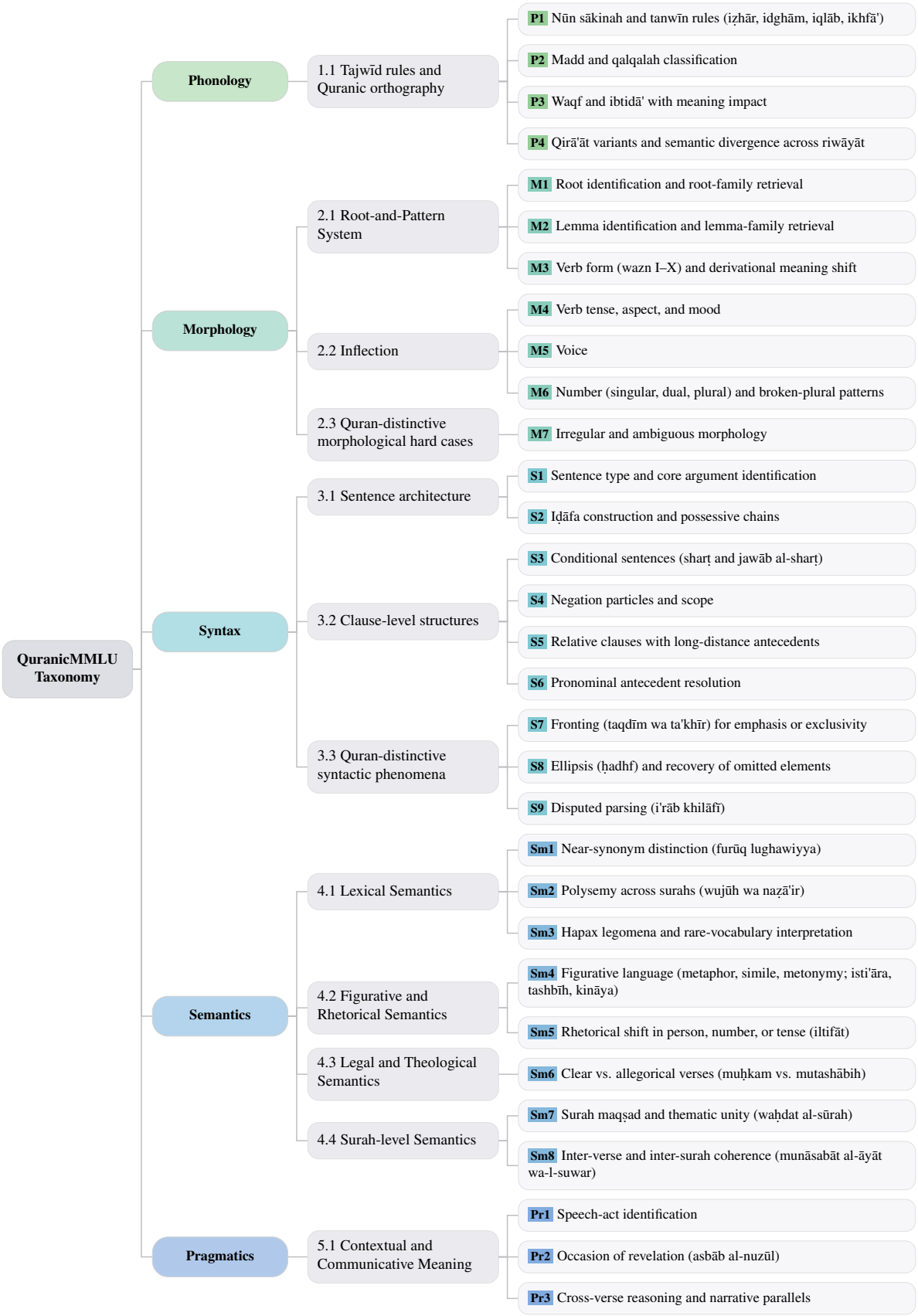


Figure 5: The QuranicMMLU taxonomy: 5 linguistic pillars, 12 categories, and 31 leaves. Leaf codes (e.g., `P1`, `Sm8`) index the specific phenomena evaluated by the benchmark.

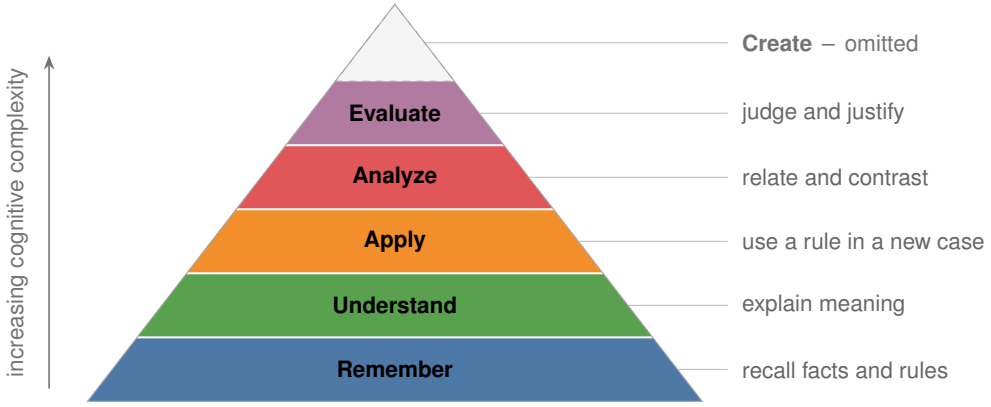
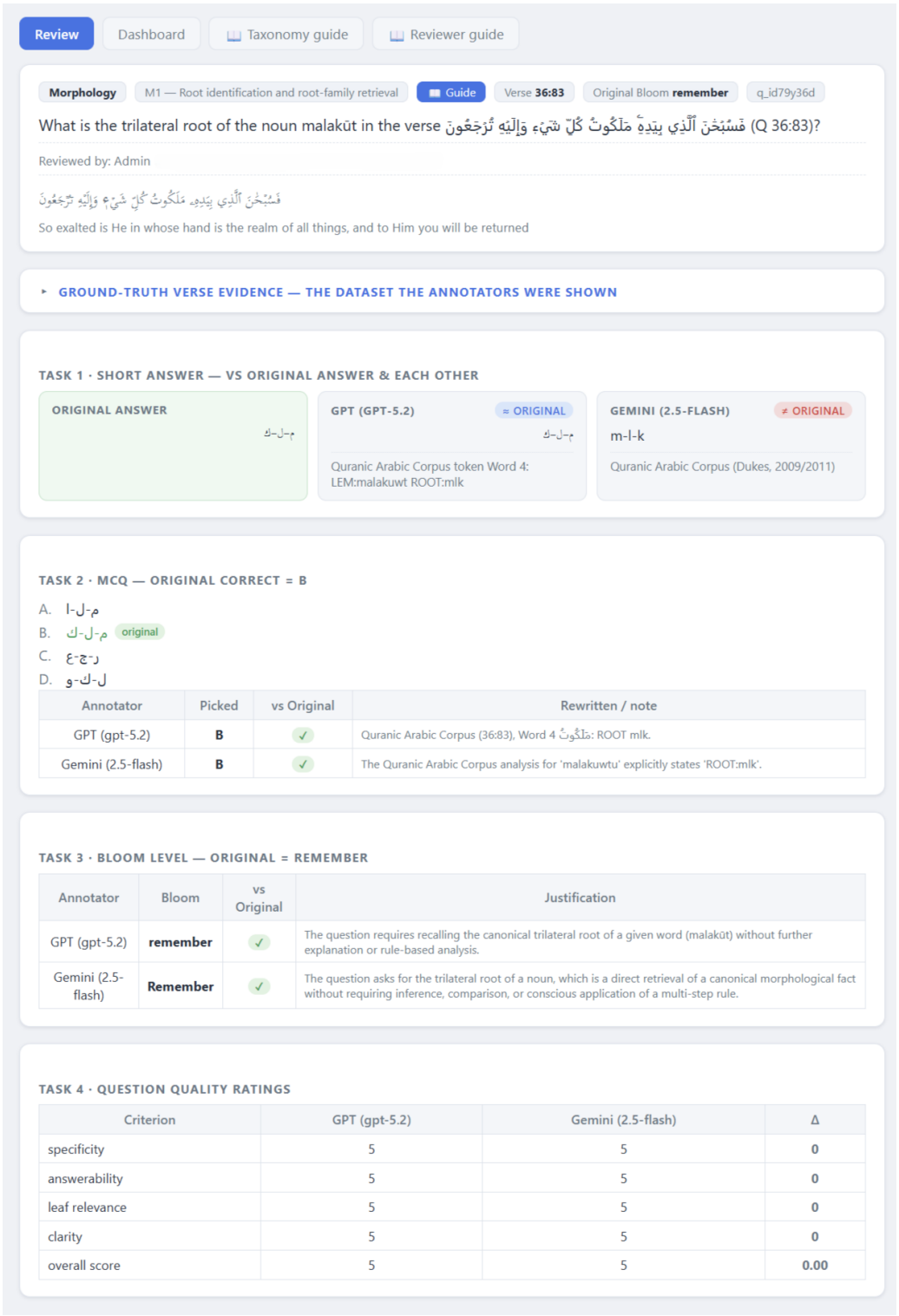


Figure 6: We stratify benchmark questions across five cognitive levels from Bloom's revised taxonomy (Remember through Evaluate), excluding the Create level, as synthesizing novel Quranic content would violate theological and scholarly principles in Islamic tradition.

Question-quality rubric used during annotation and review

Specificity. Does the question point to exactly one defensible answer? 5 = fully constrained (target word, recitation, and referent fixed or inferable); 1 = many defensible answers, none privileged by the question.
Answerability. Can the question be answered from the question, verse, translation, and leaf alone? 5 = fully self-contained; 1 = depends on an artefact the model never sees, which marks the question as unanswerable from the provided context.
Leaf relevance. Does the question genuinely test its assigned leaf? 5 = squarely on the leaf; 1 = belongs to a different leaf.
Clarity. Is the wording unambiguous, so that two competent readers interpret it identically? 5 = a single interpretation; 1 = incoherent wording.

Figure 8: Question-quality rubric used to guide human review.



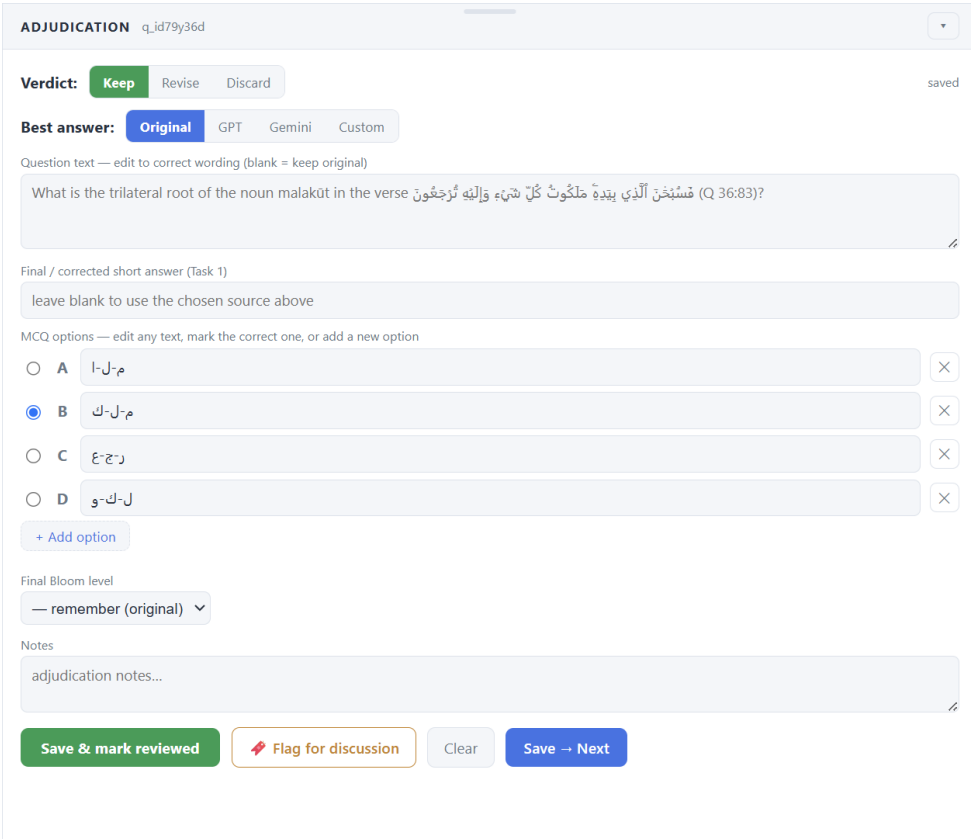
(a) Annotation view: the question with its four tasks (short answer, MCQ, Bloom level, and quality ratings), comparing each LLM judge against the reference answer.

Figure 9: The custom web interface used for human review and adjudication.



(b) Grounding view: the word-by-word morphological evidence and translations shown to reviewers as ground truth for each verse.

Figure 9: (continued) Human review and adjudication interface.



(c) Adjudication panel: reviewers record a verdict (keep, revise, or discard), select or edit the final answer and options, set the Bloom level, and leave notes.

Figure 9: (continued) Human review and adjudication interface.

LLM-judge scoring prompt for open-ended answers

Task. Score each model's free-form answer against the reference answer for an Arabic Quranic question.
Scoring rubric (0–5)**, applied strictly. 5 – Fully correct:** same meaning as the reference, no meaningful omission; paraphrase and synonyms are fine. **4 – Mostly correct:** core answer right, one minor detail missing or imprecise; would not mislead. **3 – Partially correct:** at least one correct key fact, but omits or contradicts another. **2 – Mostly wrong:** predominantly incorrect, but contains at least one relevant element. **1 – Barely relevant:** almost entirely wrong, but addresses the right topic or domain. **0 – Wrong / empty:** factually wrong, hallucinated, off-topic, refused, or empty.
Label mapping. correct = 4–5; partial = 2–3; incorrect = 0–1.
Judging notes. Judge meaning, not surface wording; Arabic synonyms and paraphrases are acceptable. For a short entity or term, a near-exact match scores 5 and the correct meaning in other words scores 4. Do not penalise markdown formatting. Correct but irrelevant extra information is not penalised unless it contradicts the reference.
Inputs. the question, the reference answer, and a block of model answers.
Output. a single JSON object (no markdown fences) mapping each model key to a score, label, and a one-sentence reason citing specific content:

```
{"model_key": {"score": 0-5, "label": "correct|partial|incorrect", "reason": "..."}}
```

Figure 10: LLM-judge scoring prompt used to grade open-ended answers.

Comparative Study of Classical NLP and Pretrained Language Models for Tunisian Dialect Processing

Latifa Iben Nasr[1], **Ismail Badache**[2], **Lamia Hadrich Belguith**[1]
[1]University of Sfax, FSEG Sfax, MIRACL Laboratory, Tunisia
[2]Aix-Marseille Université, CNRS, LIS, Marseille, France
latifa.ibennasr@fsegs.usf.tn, ismail.badache@lis-lab.fr, lamia.belguith@fsegs.usf.tn

Abstract

Recent transformer-based language models rely on subword tokenization and contextual embeddings, raising questions about the continued effectiveness of traditional preprocessing techniques such as stemming. This issue remains underexplored for low-resource Arabic dialects, particularly Tunisian Dialect (TD). In this paper, we investigate the impact of light and aggressive stemming on two main Tunisian NLP tasks: sentiment analysis and domain classification. We compare classical machine learning methods with transformer-based Arabic language models, including TunBERT. To support this study, we release an open Tunisian stemming resource containing 7,730 words annotated with both light and aggressive stems. Our experiments provide a comprehensive analysis of TD processing and show that the effectiveness of stemming depends on both the task and the model type.

1 Introduction

The emergence of embedding-based models and pretrained language models has significantly reduced the reliance on traditional machine learning pipelines and manual preprocessing techniques. These models, particularly transformer-based architectures, achieve strong performance across many Natural Language Processing (NLP) tasks by learning contextualized representations from large-scale corpora (Alkaabi et al., 2025).

However, low-resource languages and dialects such as Tunisian Dialect (TD) remain particularly challenging for NLP due to their rich morphology, lack of standardization, and high lexical variability. Moreover, they suffer from a severe scarcity of annotated and open-source resources (Besdouri et al., 2021; Ibn Nasr et al., 2025; Besdouri et al., 2024). Traditionally, Arabic NLP pipelines addressed these challenges by relying heavily on preprocessing techniques such as stemming and feature engineering to enhance the performance of classical machine learning models (Habash, 2005), which has raised questions about the continued necessity and effectiveness of such traditional preprocessing approaches in the presence of recent language models (Alkaabi et al., 2025; Atwan et al., 2021b).

In this work, we investigate the impact of stemming on TD NLP by systematically evaluating light and aggressive stemming across classical machine learning and transformer-based models, using a newly released open stemming resource for TD. The study evaluates these strategies on sentiment analysis and domain classification tasks, providing both a reusable linguistic resource and an empirical assessment of stemming techniques.

The main contributions of this paper are as follows:

- We investigate the impact of stemming on TD NLP.
- We compare classical machine learning under preprocessing and transformer-based models.
- We evaluate these strategies on sentiment analysis and domain classification tasks.
- We release an open stemming resource containing light and aggressive stems for TD with 7,730 instances.

The remainder of this paper is organized as follows: Section 2 presents the related work; Section 3 describes the resources and methodology; Section 4 presents the results; Section 5 discusses the findings; and Section 6 concludes the paper and outlines future directions.

2 Related Work

Arabic is a morphologically rich language, making preprocessing techniques important for NLP tasks (Habash, 2010). Stemming techniques are

Proceedings of The Fourth Arabic Natural Language Processing Conference (ArabicNLP 2026), pages 317–324
October 28-29, 2026 ©2026 Association for Computational Linguistics

commonly classified into light stemming, which removes affixes, and aggressive stemming, which attempts to recover roots (Mustafa et al., 2017). Various stemmers, including Khoja, ISRI, and Light10, have been widely used to reduce lexical sparsity and improve text classification and information retrieval performance (Mustafa et al., 2017; Alayba, 2025; Naili et al., 2019; Atwan et al., 2021a,b). However, aggressive stemming may introduce ambiguity and reduce performance (Atwan et al., 2021b; Naili et al., 2019).

Transformer-based models such as AraBERT (Antoun et al., 2020), MARBERT (Abdul-Mageed et al., 2021), CAMeLBERT (Inoue et al., 2021), and TunBERT (Messaoudi et al., 2022) have substantially advanced Arabic NLP. Their contextual representations and subword tokenization can capture morphological information without explicit normalization (Abdelali et al., 2022). As a result, the role of traditional preprocessing remains debated (Alkaabi et al., 2025). While stemming often benefits sparse representations such as TF-IDF (Atwan et al., 2021a), excessive preprocessing may remove useful semantic information (Alayba, 2025), particularly in dialectal Arabic, where limited resources remains a major challenge.

3 Data and Methodology

This section presents the components of the proposed methodology, as illustrated in Figure 1, detailing the process from the NLP tasks to model evaluation.

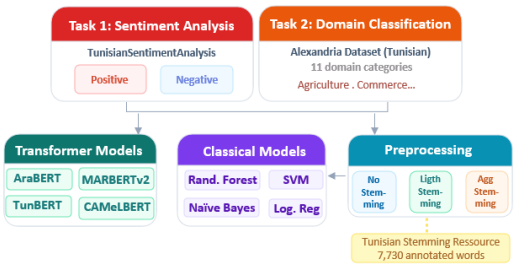


Figure 1: Workflow for TD Downstream NLP Tasks Evaluation

3.1 Resources

3.1.1 Tunisian Stemming Resource

To conduct our experiments, we release an open Tunisian stemming resource composed of 7,730 Tunisian words manually annotated with both light and aggressive stems. Each entry contains three fields: (i) the original Tunisian word, (ii) its light stem, and (iii) its aggressive stem. Table 1 presents illustrative examples from the corpus.

The corpus was manually constructed following a controlled annotation protocol involving two native Tunisian annotators and one adjudicator responsible for resolving disagreements. Annotators were selected based on their expertise in linguistic analysis, and all were graduates of the Faculty of Letters with a specialization in linguistics. Prior to annotation, a calibration meeting was held to establish common guidelines and ensure consistency.

The resource was developed in two stages: manual collection and data enrichment. In the manual collection stage, native-speaking annotators collected words from everyday Tunisian communication, covering common lexical and morphological patterns. In the data enrichment stage, additional lexical items were extracted from the Tunisian STT-TTS dataset[1] using the Camel Tools library for tokenization. Duplicates were removed, and inconsistent or non-standard entries were corrected or excluded.

To assess annotation reliability, we computed Cohen's Kappa (κ)(Cohen, 1960), obtaining an agreement score of 93%, indicating high inter-annotator consistency. Remaining disagreements were resolved by the adjudicator.

An intrinsic evaluation was conducted on a held-out subset of 800 words from the annotated corpus. The hybrid stemmer achieved 92.8% accuracy for light stemming and 86.1% for aggressive stemming against the manually annotated reference stems. Further details on the evaluation protocol, disagreement resolution, results, and qualitative analysis of stemming transformations and errors are provided in Appendix A.

The dataset is publicly available under a CC-BY license at: https://github.com/LatifaIbenNasr/Stemming.

3.1.2 Downstream NLP Tasks

We conduct experiments on two Tunisian NLP tasks: sentiment analysis using the TunisianSentimentAnalysis dataset[2], which contains Tunisian social media texts annotated with two classes: positive and negative, and domain classification using the Tunisian subset of the Alexandria dataset (Mekki et al., 2026), which includes 11 domain categories (e.g., Agriculture / Farming, Commerce / Transactions) for each text.

[1] https://huggingface.co/datasets/Arbi-Houssem/Tunisian_dataset_STT-TTS15s_filtred1.0

[2] [https://huggingface.co/datasets/hedhoud12/TunisianSentimentAnalysis

Words'origin	Light Stem	Aggressive Stem
جايبينك / jAybynk / we brought you	جايين / jAyiyn / coming	جاي / jay / come
الصحونة / alSaHwnp / the dishes	صحونة / SaHwnhp / dishes	صحن / SaHn / dish
الكراهب / alkrAheb / the cars	كراهب / krAheb / cars	كرهبة / krhbp / car
البرودويات / AlbrwdwyAt / the products	برودويات / brwdwyAt / products	برودوي / brwdwY / product

Table 1: Examples of Stemming in the TD

3.2 Models

We compare two categories of NLP models: classical machine learning and transformer-based models.

3.2.1 Classical Machine Learning Models

We evaluate four classical models: Support Vector Machines (SVM), Logistic Regression (LR), Naive Bayes, and Random Forest. All models use CAMeL tokenization followed by TF-IDF feature extraction.

Hyperparameters are set as follows: SVM uses a linear kernel with $C = 1.0$; LR uses the *liblinear* solver with $C = 1.0$; Multinomial Naive Bayes uses $\alpha = 1.0$; and Random Forest uses 100 trees with a maximum depth of 20. The dataset is split into 70% training, 15% validation, and 15% testing sets.

3.2.2 Transformer-based Models

We evaluate four transformer models: AraBERT, MARBERTv2, CAMeLBERT, and the Tunisian-specific TunBERT. All models are fine-tuned separately for each task.

The training setup is identical across models: maximum sequence length of 128 tokens, batch size of 16, learning rate of 2×10^{-5}, and 3-5 training epochs depending on convergence. We use the Adam optimizer with linear learning rate decay and warmup steps set to 10% of total training steps.

3.3 Preprocessing Configurations

To analyze the impact of TD stemming, we evaluate three configurations: no stemming, light stemming, and aggressive stemming.

For stemming, we adopt a hybrid approach that combines exact dictionary lookup using the proposed Tunisian stemming resource with fuzzy string matching. If a token is absent from the stem lexicon, the closest lexical entry is retrieved using the `fuzz.ratio` function from the RapidFuzz library, which computes a normalized Levenshtein-based similarity score. The stem of the closest entry is assigned only if the similarity score exceeds a predefined threshold. The fuzzy matching threshold was tuned on the development set by evaluating values of 75, 80, 85, 90, and 95. A threshold of 85 was selected, as it provided the best trade-off between stem recovery and erroneous stem assignments.

The `fuzz.ratio` metric was selected to measure string similarity and accommodate minor orthographic differences observed in TD (Turki et al., 2024). Its normalized similarity score and efficient implementation make it suitable for the fuzzy matching step (Bachmann et al., 2025).

4 Results

For sentiment analysis, as shown in Tables 2 and 3, the results indicate that stemming does not improve the results. The best classical result is achieved by Naive Bayes without stemming, reaching an accuracy of 0.8737 and an F1-score of 0.8702. Transformer models obtain slightly higher scores, with AraBERTv02 achieving the best overall performance (0.8834 accuracy).

Model	Metric	No stemming	LightStem	AggStem
SVM	Acc	0.8707	0.8669	0.8674
	F1	0.8691	0.8654	0.8658
LR	Acc	0.8686	0.8657	0.8655
	F1	0.8644	0.8623	0.8622
Naive Bayes	Acc	**0.8737**	0.8707	0.8650
	F1	**0.8702**	0.8662	0.8605
Random Forest	Acc	0.8409	0.8369	0.8430
	F1	0.8347	0.8306	0.8370

Table 2: Performance of classical machine learning models under preprocessing strategies for sentiment analysis.

Model	Accuracy	F1 Score
MARBERTv2	0.8792	0.8790
AraBERTv02	**0.8834**	**0.8833**
CAMeLBERT	0.8651	0.8653
TunBERT	0.6931	0.6871

Table 3: Performance of transformer-based models on the sentiment analysis task

Figure 2 summarizes the accuracy comparison across all evaluated models. It demonstrates that traditional approaches remain competitive for this task, with a modest performance for TunBERT.

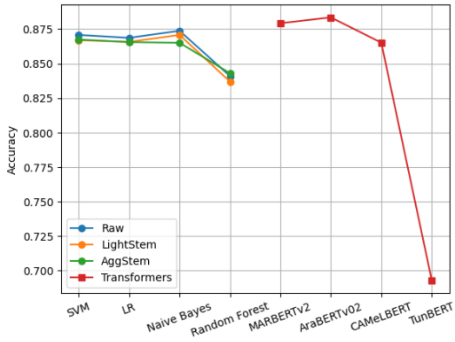


Figure 2: Accuracy comparison on the sentiment analysis task.

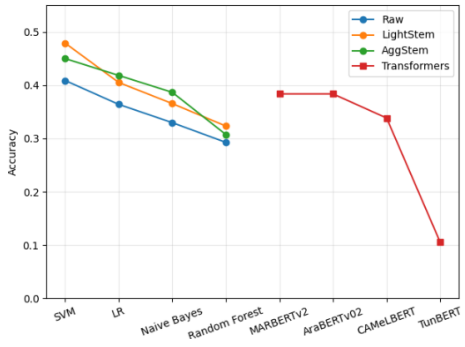


Figure 3: Accuracy comparison on the domain classification task.

For domain classification, as illustrated in Tables 4 and 5, preprocessing has a stronger impact. Light stemming consistently improves performance, allowing SVM to achieve the best classical result (0.4788 accuracy). In contrast, aggressive stemming provides smaller gains and suggests that it may remove discriminative lexical information that helps distinguish between domains. Transformer models obtain lower scores, with AraBERTv02 and MARBERTv2 reaching the highest accuracy among them (0.3835).

Model	Metric	No stemming	LightStem	AggStem
SVM	Acc	0.4087	**0.4788**	0.4498
	F1	0.3983	**0.4561**	0.4304
LR	Acc	0.3639	0.4050	0.4182
	F1	0.3189	0.3659	0.3668
Naive Bayes	Acc	0.3296	0.3654	0.3865
	F1	0.2813	0.3136	0.3297
Random Forest	Acc	0.2926	0.3232	0.3073
	F1	0.2740	0.3112	0.2758

Table 4: Results of classical machine learning models on the domain classification task

Model	Accuracy	F1 Score
MARBERTv2	0.3835	0.3338
AraBERTv02	0.3835	0.3413
CAMeLBERT	0.3383	0.2730
TunBERT	0.1053	0.0201

Table 5: Performance of transformer-based models on the domain classification task

As shown in Figure 3, the best-performing classical configuration outperforms all transformer-based models on this task. Although TunBERT was pretrained on Tunisian dialect data, it still achieves the lowest performance among the evaluated transformer models.

Overall, the results highlight the task-dependent effect of stemming. While sentiment analysis benefits from preserving original word forms, domain classification benefits from moderate normalization through LightStem. Transformer models perform best on sentiment analysis, whereas classical machine learning, particularly SVM with LightStem, proves more effective for domain classification. Light stemming consistently outperforms aggressive stemming. Classical approaches remain competitive and, in some cases, outperform transformer models despite their lower computational requirements. These findings suggest that careful preprocessing and model selection remain crucial for Arabic downstream NLP tasks.

5 Discussion

Our results show that the effect of stemming is strongly task-dependent. For domain classification, light stemming improves the performance of SVM and other classical models, which is consistent with previous studies reporting benefits of stemming for Arabic text classification (Atwan et al., 2021a; Wahbeh et al., 2011; Ayedh et al., 2016). Domain classification relies heavily on domain-specific terminology, and light stemming can group morphological variants while reducing feature sparsity in TF-IDF representations without substantially altering lexical identity. In contrast, aggressive stemming may remove discriminative lexical information by reducing domain-specific terms to broader or semantically different roots. For example, القمحية is preserved as قمحية by light stemming but reduced to قمح (wheat) by aggressive stemming, potentially introducing an agriculture-related interpretation. Similarly, ملاحة (navigation/shipping) is reduced to ملح (salt), while الجامعة is reduced to the generic root جمع. These cases illustrate how aggressive stemming can weaken domain-specific lexical cues, explaining why light stemming generally outperforms aggressive stemming in domain classification.

For sentiment analysis, stemming has a much more limited effect. Sentiment prediction depends more strongly on sentiment-bearing lexical forms and contextual expressions, for which morphological distinctions may carry useful information. Moreover, transformer models can capture substantial morphological variation through subword tokenization and contextual representations (Alkaabi et al., 2025). Thus, explicit stemming provides limited additional benefit for sentiment analysis and may sometimes remove useful lexical distinctions.

The performance of TunBERT provides further evidence that preprocessing and model vocabulary interact with the characteristics of the task. Despite being pretrained on TD data, TunBERT was consistently outperformed by AraBERT and MARBERT, which may be partly explained by the larger and more diverse pretraining resources available for these models (Abdul-Mageed et al., 2021; Inoue et al., 2021).

To investigate this performance gap, we analyzed tokenization on the Alexandria dataset. TunBERT showed a higher Out-of-Vocabulary (OOV) rate (3.8% vs. less than 0.5% for AraBERT and MARBERT) and greater subword fragmentation, with an average of 2.47 subwords per word, compared with 1.61 and 1.67, respectively. This suggests a vocabulary mismatch between TunBERT's pretraining data and domain-specific expressions in the evaluation corpus.

Overall, these findings indicate that stemming should be adapted to both the downstream task and the model architecture. Light stemming provides a useful balance between morphological normalization and lexical preservation for classical lexical-feature-based models, whereas aggressive stemming can remove discriminative domain cues. Transformer-based models, in contrast, can handle much of the morphological variation through subword and contextual representations. Therefore, stemming should not be considered a universally beneficial preprocessing step, but rather a task- and model-dependent strategy.

6 Conclusion and Future Direction

This study evaluated the impact of stemming on TD NLP for sentiment analysis and domain classification. The results show that light stemming benefits classical machine learning models, particularly for domain classification, whereas transformer models are less affected. Nevertheless, classical approaches remain competitive when combined with suitable preprocessing.

Future work will extend the evaluation to larger and more diverse Arabic corpora and enrich the publicly available stemming corpus to support further research in Arabic NLP.

Limitations

This work is limited to two TD NLP tasks and a stemming resource of 7,730 annotated words. The resource may not fully represent the lexical and regional diversity of TD. Moreover, only a limited set of classical and transformer-based models was evaluated. Future work should investigate additional NLP tasks, larger datasets, and other Arabic dialects.

References

Ahmed Abdelali, Nadir Durrani, Fahim Dalvi, and Hassan Sajjad. 2022. Post-hoc analysis of arabic transformer models. In *Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 91–103.

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.

Abdulaziz M Alayba. 2025. Arabic natural language processing (nlp): A comprehensive review of challenges, techniques, and emerging trends. *Computers*, 14(11):497.

Hussein Ala'a Alkaabi, Ali kadhim Jasim, and Ali Darroudi. 2025. Arabic nlp: A survey of pre-processing and representation techniques. *Journal of Computer Science, Information Technology and Telecommunication Engineering*, 6(2):876–890.

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.

Jaffar Atwan, Mohammad Wedyan, Qusay Bsoul, Ahmad Hamadeen, Ryan Alturki, and Mohammed Ikram. 2021a. The effect of using light stemming for arabic text classification. *International Journal of Advanced Computer Science and Applications*, 12(5):768–773.

Jaffar Atwan, Mohammad Wedyan, Qusay Bsoul, Ahmad Hammadeen, and Ryan Alturki. 2021b. The use

of stemming in the arabic text and its impact on the accuracy of classification. *Scientific Programming*, 2021(1):1367210.

Abdullah Ayedh, Guanzheng Tan, Khaled Alwesabi, and Hamdi Rajeh. 2016. The effect of preprocessing on arabic document categorization. *Algorithms*, 9(2):27.

Max Bachmann, Georgia Kokkinou, Henry Schreiner, Jeppe Fihl-Pearson, Michał Górny, Moshe Sherman, Cristian Le, Nicolas Renkamp, Kwuang Tang, Hugo Le Moine, and 1 others. 2025. rapidfuzz/rapidfuzz: Release 3.13. 0. *Zenodo*.

Fatma Zahra Besdouri, Asma Mekki, Inès Zribi, and Mariem Ellouze. 2021. Improvement of the cota-orthography system through language modeling. In *2021 IEEE/ACS 18th International Conference on Computer Systems and Applications (AICCSA)*, pages 1–7. IEEE.

Fatma Zahra Besdouri, Inès Zribi, and Lamia Hadrich Belguith. 2024. Arabic automatic speech recognition: challenges and progress. *Speech Communication*, 163:103110.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Rehab Duwairi, Mohammad Al-Refai, and Natheer Khasawneh. 2007. Stemming versus light stemming as feature selection techniques for arabic text categorization. In *2007 Innovations in Information Technologies (IIT)*, pages 446–450. IEEE.

Nizar Habash. 2005. Introduction to arabic natural language processing. In *Association for Computational Linguistics*, volume 30.

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

Latifa Ibn Nasr, Abir Masmoudi, and Lamia Hadrich Belguith. 2025. Emotion recognition from spontaneous tunisian dialect speech. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 24(2):1–16.

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.

Abdellah El Mekki, Samar M Magdy, Houdaifa Atou, Ruwa AbuHweidi, Baraah Qawasmeh, Omer Nacar, Thikra Al-hibiri, Razan Saadie, Hamzah Alsayadi, Nadia Ghezaiel Hammouda, and 1 others. 2026. Alexandria: A multi-domain dialectal arabic machine translation dataset for culturally inclusive and linguistically diverse llms. *arXiv preprint arXiv:2601.13099*.

Abir Messaoudi, Ahmed Cheikhrouhou, Hatem Haddad, Nourchene Ferchichi, Moez BenHajhmida, Abir Korched, Malek Naski, Faten Ghriss, and Amine Kerkeni. 2022. Tunbert: Pretrained contextualized text representation for tunisian dialect. In *International Conference on Intelligent Systems and Pattern Recognition*, pages 278–290. Springer.

Mohammad Mustafa, Afag Salah Eldeen, Sulieman Bani-Ahmad, Abdelrahman Osman Elfaki, and 1 others. 2017. A comparative survey on arabic stemming: approaches and challenges. *Intelligent Information Management*, 9(2):39–78.

Marwa Naili, Anja Habacha Chaibi, and Henda Hajjami Ben Ghezala. 2019. Comparative study of arabic stemming algorithms for topic identification. *Procedia Computer Science*, 159:794–802.

Houcemeddine Turki, Kawthar Ellouze, Hager Ben Ammar, Mohamed Ali Hadj Taieb, Imed Adel, Mohamed Ben Aouicha, Pier Luigi Farri, and Abderrezak Bennour. 2024. Normalized orthography for tunisian arabic. In *International Conference on Language Processing and Knowledge Management*, pages 133–142. Springer.

Abdullah Wahbeh, Mohammed Al-Kabi, Qasem Al-Radaideh, Emad Al-Shawakfa, and Izzat Alsmadi. 2011. The effect of stemming on arabic text classification: An empirical study. *International Journal of Information Retrieval Research (IJIRR)*, 1(3):54–70.

A Appendix

Further details on stemming variation, intrinsic evaluation, and additional analyses are provided below.

A.1 Stemming Variation and Annotator Disagreement

Most disagreements were due to sociolinguistic variation between urban and rural Tunisian dialects, particularly in aggressive stemming. For example, for the words [نمشيو / nemshyw / we go] and [كلات / klet / she eats], the urban annotator produced the aggressive stems [مشـى / mshY / to go] and [كلى / klY / to eat], whereas the rural annotator proposed [مشي / mshy / to go] and [كلي / kly / to eat]. This variation concerns the final vowel realization [ى / Y] versus [ي / y]. To ensure consistency and compatibility with Modern Standard Arabic (MSA) orthography, the adjudicator standardized these forms using the final [ى / Y] representation.

To investigate the behavior of stemming forms as illustrated in Figure A.1, we computed the Levenshtein distance between each original word and its stem for both light and aggressive stemmers.

On average, the aggressive stemmer alters words more extensively (mean distance 2.79) than the light stemmer (mean distance 1.64), and exhibits higher variability (standard deviation 1.34 versus 0.69), indicating that some words undergo substantial transformation.

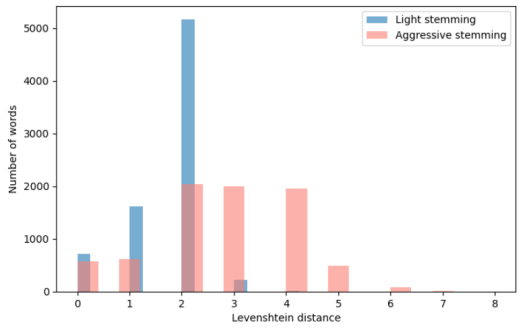


Figure A.1: Word Changes By Stemming Type

A correlation of 0.66 between the distances of the two stemmers suggests a moderate positive relationship: although both stemmers often target the same types of words, the aggressive stemmer applies more radical changes in certain cases. Examination of the most modified words shows that the aggressive stemmer sometimes produces stems with little semantic resemblance to the original, whereas the light stemmer generally preserves the lexical root.

These observations illustrate the trade-off between lexical simplification and semantic preservation, emphasizing that the choice of stemming strategy can significantly influence word forms and may affect downstream applications (Duwairi et al., 2007).

A.2 Intrinsic Evaluation

To complement the downstream evaluation, we conducted an intrinsic evaluation of the proposed Tunisian stemming resource. From the final manually annotated corpus, a held-out set of 800 Tunisian words was selected for intrinsic evaluation.

The hybrid stemming approach described in Section 3.3 was evaluated by comparing its automatically predicted stems against the manually annotated reference stems. The evaluation was performed separately for light and aggressive stemming using exact-match accuracy, where a prediction was considered correct only when the predicted stem exactly matched the corresponding reference stem.

The hybrid stemmer achieved an accuracy of 92.8% for light stemming and 86.1% for aggressive stemming. These results indicate that light stemming is more consistent and easier to recover than aggressive stemming. The lower performance of aggressive stemming reflects the greater difficulty of identifying the underlying lexical form in the presence of Tunisian morphological and orthographic variation.

The analysis of failures indicates that most errors fall into two main categories: orthographic variation and ambiguous dialectal forms.

Orthographic variants. Tunisian words may occur with different orthographic realizations while referring to the same lexical item. Such variation can make exact stem matching difficult even when the predicted stem is linguistically plausible.

Ambiguous dialectal forms. Some words admit different stem interpretations depending on dialectal or morphological analysis. This issue is particularly relevant for aggressive stemming, which attempts to recover a deeper lexical representation.

Overall, the intrinsic evaluation confirms that the proposed resource provides reliable stemming annotations, with particularly strong performance for light stemming.

A.3 Additional Analyses

A.3.1 Fuzzy Matching Contribution

We conducted an ablation analysis to quantify the contribution of fuzzy matching beyond exact dictionary lookup. Table 6 reports the proportion of tokens processed by each matching strategy for the two downstream tasks.

Table 6: Distribution of matching strategies (%).

Task	Dictionary	Fuzzy	Unchanged
Domain classification	81.7	12.9	5.4
Sentiment analysis	84.2	10.6	5.2

Fuzzy matching successfully recovers 12.9% and 10.6% of tokens for domain classification and sentiment analysis, respectively, that are not directly covered by the stemming lexicon. As shown in Table 7, the additional coverage improves SVM accuracy for domain classification by 1.8 percentage points compared with dictionary lookup alone.

In contrast, the contribution of fuzzy matching to sentiment analysis is limited. These results indicate that fuzzy matching provides complementary lexical coverage, while its downstream benefit remains task-dependent.

Table 7: Ablation results comparing dictionary lookup and the hybrid strategy for SVM on domain classification.

Preprocessing strategy	**Accuracy**
Dictionary lookup only	0.4608
Dictionary + fuzzy matching	0.4788
Absolute improvement	**+1.8 points**

A.3.2 Statistical Significance

We assessed the statistical significance of the differences between preprocessing configurations using McNemar's test for pairwise comparisons. The improvement of LightStem over No Stemming for SVM on domain classification is statistically significant ($p < 0.01$). Conversely, the differences between preprocessing configurations for sentiment analysis are not statistically significant ($p > 0.05$).

These results provide statistical support for the benefit of light stemming in domain classification, while confirming that its effect on sentiment analysis is not statistically significant.

AraMIP: Extending MIPVU Towards Metaphor Identification in Arabic

Mandar Marathe[*1] **Manar Ali**[*2] **Sara Nabhani**[*3] **Raia Abu Ahmad**[*4,5]
Ibrahim Baroud[*4,5] **Omar Momen**[*2]
[1]SOAS University of London, United Kingdom
[2]CRC 1646 'Linguistic Creativity in Communication', Bielefeld University, Germany
[3]Computational Linguistics, CLCG, University of Groningen, The Netherlands
[4]German Research Center for Artificial Intelligence (DFKI), Germany
[5]Quality & Usability Lab, Technical University of Berlin, Germany
[1]mandar@marathe.org [3]s.nabhani@rug.nl
[2]{manar.ali,omar.hassan}@uni-bielefeld.de [4,5]{raia.abu_ahmad,ibrahim.baroud}@dfki.de

Abstract

Metaphor research has gained increasing attention due to its relevance to linguistic creativity, language use, cognitive processes, and related areas. While many efforts have been devoted to metaphor identification and annotation in English and other languages, Arabic remains under-resourced in this area. In this work, we propose the Arabic Metaphor Identification Procedure (AraMIP), a novel guideline for Arabic metaphor annotation. AraMIP builds on the widely used Metaphor Identification Procedure Vrije Universiteit (MIPVU) framework, incorporating adaptations that account for the language-specific properties of Arabic. We distinguish three major types of Arabic figurative language: *isti'āra* (metaphor), *kināya* (metonymy/indirect expression), and *tashbīh* (simile) and annotate a pilot dataset of 300 sentences (5277 words). Our analysis reveals key challenges specific to Arabic, including morphological complexity, inconsistencies in dictionary sense ordering, and the absence of standardized contextual materials for annotators. This work contributes a first step toward standardized Arabic figurative instances and facilitates the development of larger annotated resources, thereby supporting future research on figurative language in Arabic.[1]

1 Introduction

Humans use metaphors in everyday language to achieve diverse communicative goals, such as expressing emotions and explaining unfamiliar concepts. For instance, someone may describe an attempted scam by saying, "He wants to eat my money," drawing on the concrete domain of eating to express the more abstract notion of financial exploitation. Notably, humans can interpret metaphorical expressions quickly (Glucksberg, 1998). Such expressions involve cross-domain mapping, which is central to several influential theories of metaphor, most notably the Conceptual Metaphor Theory (CMT, Lakoff and Johnson, 1980), which argues that a metaphor is not only a linguistic phenomenon but also a cognitive mechanism through which concrete concepts (source) are mapped onto more abstract domains (target). In the example above, *eating* is the source and *financial exploitation* the target. Metaphorical framing also plays an important role in shaping how people conceptualize abstract ideas and reason about them (Thibodeau and Boroditsky, 2011).

Metaphor research has attracted increasing attention in (Cognitive) Linguistics and Natural Language Processing (NLP), and has also been discussed in the broader context of linguistic creativity (Vogel et al., 2026). Data collection and annotation studies identified metaphors (Steen et al., 2010; Mohammad et al., 2016; Beigman Klebanov et al., 2018; Leong et al., 2018), or classified them based on deliberateness and novelty (Reijnierse et al., 2018; Do Dinh et al., 2018). Tong et al. (2021) and Ge et al. (2023) investigated language models' ability to generate and comprehend metaphors as a proxy for cognitive and pragmatic competence. Other studies explored the role of metaphor in downstream tasks such as machine translation (Wang et al., 2024) and offensive language detection (Lemmens et al., 2021; Zeng et al., 2025), where identifying figurative expressions can improve system performance (Huguet Cabot et al., 2020; Dankers et al., 2019).

Besides English, annotation efforts and benchmark datasets have been developed for several languages, including Spanish (Sanchez-Bayona and Agerri, 2022), Chinese (Lu and Wang, 2017), and German (Egg and Kordoni, 2022). Arabic, however, remains under-resourced in this area. While several studies have touched on Arabic metaphor

[*]Equal contribution.
[1]Data, annotation guidelines and code are available here https://github.com/AraMIP/AraMIP

Proceedings of The Fourth Arabic Natural Language Processing Conference (ArabicNLP 2026), pages 325–339
October 28-29, 2026 ©2026 Association for Computational Linguistics

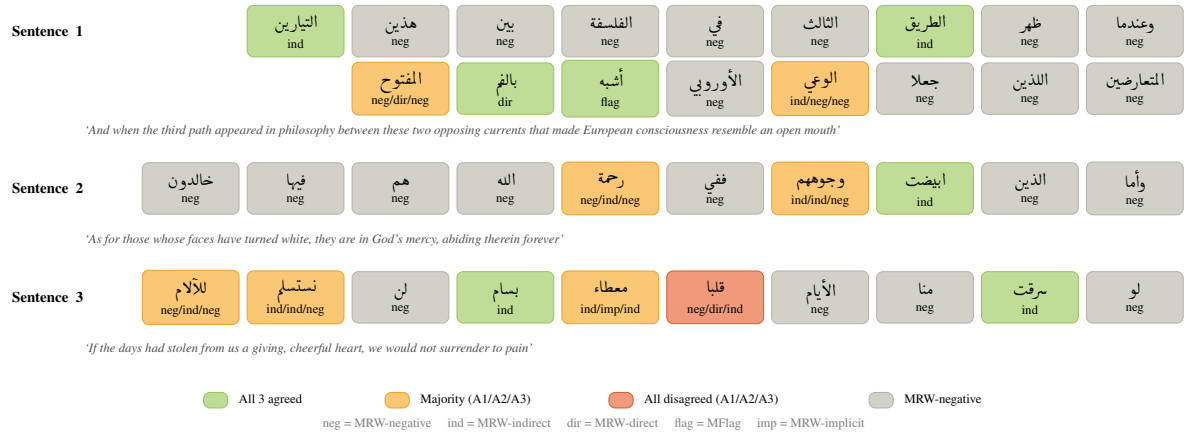


Figure 1: Three annotated sentences illustrating token-level agreement. Green = all 3 agreed; orange = majority; red = all disagreed; gray = MRW-negative. Disagreement shown as A1/A2/A3.

in NLP contexts (Alkhatib and Shaalan, 2016; Alsiyat and Piao, 2020; Alsiyat et al., 2023), to our knowledge none provide a standardized, replicable annotation procedure, and existing datasets are limited in domain coverage or methodological transparency (Alsiyat and Piao, 2020; Magdy et al., 2024). Arabic furthermore presents challenges that existing frameworks do not address. In particular, its rich morphological system, root-based lexicographic structure, and the long rhetorical tradition distinguishing *isti'āra* (metaphor), *kināya* (metonymy/indirect expression), and *tashbīh* (simile) in ways that do not map directly onto categories developed for other languages.

In this work, we introduce the Arabic Metaphor Identification Procedure (AraMIP), a two-stage annotation framework that adapts the widely used Metaphor Identification Procedure Vrije Universiteit (MIPVU, Steen et al., 2010) to Arabic. The first stage follows MIPVU's word-level metaphor identification (see Figure 1); the second maps metaphor-related words (spans) onto Arabic rhetorical categories and records the conventionality of each construction on a five-point scale. We apply AraMIP to a pilot corpus of 300 sentences (5277 words) sampled from the BAREC-10M corpus (Elmadani et al., 2026) and release the guidelines and the annotations to support future work. Our pilot study surfaces concrete challenges specific to Arabic annotation, including dictionary coverage at the boundary between Modern Standard Arabic (MSA) and Classical Arabic, and root-to-derivative sense resolution, and provides a diagnostic roadmap for future iterations of the framework.

2 Background and Related Work

2.1 Metaphor Theories

Theories and definitions of metaphors have a very long history. We do not review all theories of metaphors, but we point out their major views and understandings. The oldest standard view comes from Aristotle, where metaphor was mainly seen as a decorative, poetic, or rhetorical device where one word is used instead of another. Black (1955) viewed that metaphorical meaning emerges from the interaction between two conceptual systems, not by simple substitution as in the classical theory. On the other hand, in pragmatics, metaphor is not primarily a special semantic mapping. It is a case where the meaning of the sentence and the meaning of the speaker separate (Searle, 1979). Recently, CMT that was posed by Lakoff and Johnson (1980) has become widely adopted in multiple disciplines. It states that metaphor arises from understanding one domain of experience (that is typically abstract) in terms of another (that is typically concrete), e.g., in "She attacked my point.," the abstract target domain of "Argument" is understood in terms of the source concrete domain of "War."

In Arabic literary studies, particularly in the field of *al-balāgha*[2], metaphor is discussed within the domain of *'ilm al-bayān*[3], especially through the interrelated concepts of *tashbīh* (simile), *isti'āra* (metaphor), and *kināya* (metonymy) (Abdul-Raof, 2006). The dominant textbook definition derives

[2]al-balāgha (Arabic rhetoric) is the classical Arabic tradition concerned with effective, expressive, and aesthetically appropriate uses of language.

[3]'ilm al-bayān (the science of figurative expression) is a branch of classical Arabic rhetoric concerned with the ways a meaning can be expressed figuratively or indirectly

isti'āra from *tashbīh*, where one pole of an underlying comparison is omitted. A more sophisticated account of metaphor is provided by Abd al-Qāhir al-Jurjānī (d. 1078) as explained in Larkin (1995), who views metaphor not merely as an ornamental substitution, but as part of the production of second-order meaning through structure, context, and imaginative interpretation.

2.2 Operationalizing Metaphors

A group of scholars (The Pragglejaz Group, 2007) worked on developing a standard procedure for identifying metaphors in written and spoken discourse. They introduced the Metaphor Identification Procedure (MIP), which was then extended with finer details to MIPVU (Steen et al., 2010).[4] MIP was motivated by the fact that scholars often disagreed about what counts as metaphor. The authors aimed for an explicit, reliable, and flexible method for identifying metaphorically used words in written and spoken discourse. The core theoretical assumption of MIP is that a lexical unit (a word) is metaphorically used when its contextual meaning contrasts with a more basic meaning, but the contextual meaning can be understood in comparison with that basic meaning.

The operationalization framework of MIP resulted in a standard annotated metaphor corpus in English (VUAMC, Steen et al., 2010), which has served as the standard corpus for most metaphor studies in NLP since then (Leong et al., 2018, 2020). Later, adaptations of MIP to other languages were developed and resulted in metaphor corpora in languages other than English. For example: Russian (Badryzlova et al., 2013), Chinese (Lu and Wang, 2017), German (Egg and Kordoni, 2022), Spanish (Sanchez-Bayona and Agerri, 2022) and Japanese (Zhu et al., 2026). However, no systematic adaptation of MIP was carried out for Arabic up to the moment of writing this paper.

2.3 Related Work

Despite the absence of a systematic procedure for identifying metaphors in Arabic texts, we find multiple works that discuss metaphors in Arabic. Alkhatib and Shaalan (2016) discuss the definitions, usages and differences of metaphors across dialects. They compare Classical, Modern Standard, and Dialectal Arabic. The authors argue that machine translation approaches do not translate metaphors correctly. Alsiyat and Piao (2020) show the importance of metaphor detection and interpretation in the task of sentiment analysis and later develop the Arabic Metaphor Corpus (AMC, Alsiyat et al., 2023), which contains 1,000 sentences extracted from book reviews annotated with metaphor spans and sentiment. However, no information on the procedure used to annotate the metaphors is mentioned. Magdy et al. (2024) introduce a dataset for Arabic writing assistance, which includes 330 sentences with incorrect usage of metaphors or multiword expressions (MWEs) paired with their corrections. However, the authors do not explain the process of collecting these sentences. Zibin et al. (2025) conduct a study where they compare human performance in interpreting (Classical, Jordanian, and Emirati) Arabic metaphors to Large Langauge Models (LLMs), finding that it is challenging for LLMs to process culturally rich and context-driven language compared to humans. Banou et al. (2025) introduce a dataset of Arabic figurative speech, including metaphors and similes, but they are directly translated from English.

Overall, we find a clear gap in work on systematically identifying metaphors in Arabic using a theoretically grounded procedure that has been applied to and validated across other languages.

3 AraMIP: Annotation Framework

AraMIP is a two-stage annotation framework for Arabic metaphor and related figurative language annotation. AraMIP extends MIPVU with a second annotation layer based on the Arabic rhetorical tradition. The combined framework retains MIPVU's core annotation categories and decision principles while providing additional information about the rhetorical nature of identified metaphorical expressions (see Figure 2).

3.1 Stage 1: MIPVU Annotation

In Stage 1, AraMIP adopts the MIPVU procedure and annotation categories (see Figure 5). The MSA dictionary 'معجم اللغة العربية المعاصرة' *Mu'jam al-Lugha al-'Arabiyya al-Mu'āṣira* [5] (Omar et al., 2008) is used to determine the "basic" meaning of each lexical unit in turn. Metaphor-related words (MRWs) are identified where a lexical unit participates in a cross-domain relationship between a contextual meaning and a more basic meaning,

[4] See the MIPVU procedure in Appendix B.

[5] The dictionary was accessed via the https://www.almaany.com/ interface.

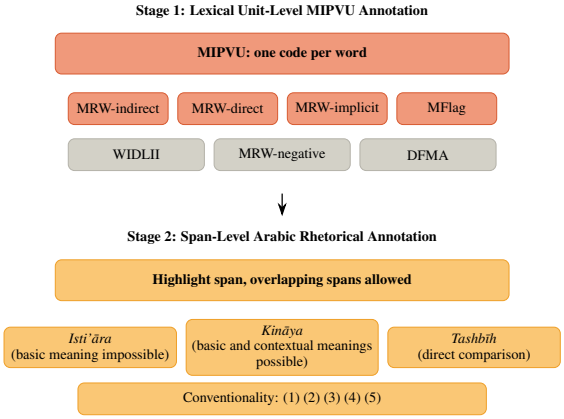


Figure 2: The AraMIP two-stage annotation procedure.

whether expressed directly, indirectly, or through an associated metaphor signal.

Direct metaphors are identified through explicit cross-domain comparison, typically signaled by a lexical comparator such as a simile marker, whereas this is absent in indirect metaphors. Implicit metaphors are lexical units that are MRWs by virtue of a previously stated comparison. Thus, each lexical unit is annotated as one of: indirect metaphor "MRW-indirect", direct metaphor "MRW-direct", implicit metaphor "MRW-implicit", metaphor flag "MFlag" (for simile markers), or not a MRW "MRW-negative". Other possible MIPVU annotations are *When In Doubt, Leave It In* ("WIDLII") for uncertain cases, and *Discarded For Metaphor Analysis* ("DFMA") for when the meaning of a lexical unit is impossible to discern.

3.2 Stage 2: Arabic Rhetorical Classification

The direct and indirect metaphors identified in Stage 1 generally correspond to three rhetorical devices in Arabic. Although the pattern is not invariable, direct metaphors often correspond to *tashbīh* (simile), while indirect metaphors often correspond to *isti'āra* (metaphor) and *kināya* (metonymy/indirect expression). Both the "basic" and "contextual" meanings are possible in *kināya*, but not in *isti'āra*. AraMIP therefore introduces a second annotation layer applied to non MRW-negative lexical units.

In accordance with Arabic rhetorical tradition, Stage 2 uses the rhetorical construction as the unit of annotation rather than the lexical unit. Rather than annotating only the individual MRW, annotators identify the complete span of the rhetorical construction to which that word belongs. Unlike Stage 1, which operates at the lexical-unit level, Stage 2 permits multi-word spans and overlapping rhetorical constructions. MFlag lexical units are annotated as part of the *tashbīh* span they introduce, rather than as an independent rhetorical category; MRW-implicit lexical units are not annotated in Stage 2.

While Stage 1 identifies conventionalized and novel metaphors alike, it does not record their degree of conventionality. In Stage 2, AraMIP additionally records the degree of conventionality of each rhetorical construction on a five-point scale.

Annotated sentences using AraMIP would contain both lexical unit-level metaphor annotations and span-level rhetorical annotations, in addition to conventionality ratings. Unlike previous MIPVU-based corpora, AraMIP supplements lexical-level metaphor annotation with span-level rhetorical classification grounded in the Arabic rhetorical tradition.

4 Corpus Selection

To test our proposed annotation framework, we select a collection of sentences to be annotated and analyzed in a pilot study. For simplicity, we only consider MSA texts, and we try to include different genres and domains to create a representative pool of MSA usage. We follow the same reasoning behind developing VUAMC (Steen et al., 2010), in which the annotated sentences were sampled from the British National Corpus (BNC).

We find BAREC-10M (Elmadani et al., 2026) to be a suitable choice for our purpose. BAREC-10M contains 10 million words from a range of text genres, where each document is split by sentences and tokens and manually annotated for: 1. **Domain** (*Arts & Humanities*; *Social Sciences*; *STEM*); 2. **Readership group** (*Foundational*; *Advanced*; *Specialized*); and 3. **Text category** (*Media & Culture*; *Religion & Philosophy*; *Literature, Art & Music*; *Academic*; *Educational Materials, Language & Linguistics*; *Encyclopedic*). In addition to other layers of linguistic annotations.

BAREC-10M contains around 514K sentences, from which we use 300 sentences to conduct our pilot study. We sample 50 random sentences from each text category, ensuring an equal distribution across domains and readership groups when possible. We also restrict sentences to lengths between 8 and 30 words, resulting in a corpus of 300 sentences and 5,277 words, with an average sentence length of 17.58 words.

5 Annotation Pilot Study

5.1 Annotation Setup

Annotators Annotation was performed by the six authors: five native Arabic speakers representing Egyptian and Levantine dialects, and one L2 Arabic speaker with an advanced level in Arabic. Collectively, the annotators had expertise in Arabic NLP, lexicography, and Arabic rhetorical and figurative language. Before the main annotation process, all annotators participated in a training session during which they jointly annotated and discussed a small sample of the sentences to establish a shared understanding of the guidelines.

Data The 300 sampled sentences were randomly partitioned and assigned to the annotators while ensuring an approximately equal distribution across text domains. Each sentence was independently annotated by three different annotators.

Annotation Platform We use INCEpTION (Klie et al., 2018), a web-based annotation platform, to annotate both stages. Screenshots of the annotation interface are shown in Appendix C.

5.2 Inter-annotator Agreement

We evaluate inter-annotator agreement (IAA) separately for each annotation stage. For Stage 1, where each word is assigned an MRW label, we report agreement at two levels. First, we measure agreement on whether a word is MRW-positive or -negative, regardless of its specific MRW type. Second, we measure agreement using the full set of MRW labels. For both settings, we compute Fleiss' κ and report the average score, the weighted average score, and the agreement for each label. The results are shown in Table 1.

For Stage 2, where annotators identify spans and assign figurative language labels to them, agreement depends on both the span boundaries and the assigned label. We compute pairwise exact-match F1 between annotators and report the average across all annotator pairs. Exact-match F1 requires both the span boundaries and the label to match exactly. We also compute pairwise Intersection over Union (IoU-based F1) and average the scores across all annotator pairs, using an IoU threshold of 0.5 to allow partial overlap between spans. Finally, we report Soft γ (Mathet et al., 2015) under three settings: equal weight for span position and label, higher weight for the label, and higher weight for the span position. The results are shown in Table 2.

Measure	**Fleiss' κ**
Binary MRW agreement	0.358
Avg. MRW label agreement	0.229
Weighted MRW label agreement	0.357
Per-label agreement	
MRW-Negative	0.360
MRW-Indirect	0.315
MRW-Direct	0.292
MRW-Implicit	-0.001
MRW-Flag	0.499
MRW-DFMA	0.104
MRW-WIDLII	0.035

Table 1: Inter-annotator agreement for the word-level annotation layer, measured using Fleiss' κ.

Measure	**Score**
Soft γ (equal weighting)	0.477
Soft γ (label-priority weighting)	0.442
Soft γ (position-priority weighting)	0.440
Exact-match F_1	0.125
IoU-based F_1	0.211

Table 2: Inter-annotator agreement for the span-level annotation layer.

5.3 Automatic Curation

We curate the annotations for both Stage 1 and Stage 2 for each word/sentence as follows:

Stage 1 Each word was assigned an MRW code following a majority vote. For the curation process, we distinguish between three kinds of agreement cases: 1. *Clear agreement:* same label given by all annotators, 2. *Majority agreement:* two annotators agree on the label, and 3. *Disagreement:* three different labels. The curated dataset contains 90.49% *clear agreement* cases, 9.08% *majority agreement* cases, and 0.44% *disagreement* cases. To resolve the disagreement cases, we perform a second annotation round, in which three different annotators independently annotate each case. The final curation was then performed by considering the annotations from all six annotators. This process resolved 21 out of 23 disagreements, leaving two critical disagreements for discussion, in which the words were still annotated with three different labels each by two annotators. The first disagreement concerned the word "مفاجأة" - *"surprise"* in the sentence "كان الرّجل مفاجأة لا تنتهي بالنسبة لي"

- *"The man was a never-ending surprise to me"* which was annotated with the three labels *MRW-Direct*, *MRW-Indirect* and *MRW-Negative* twice each. The second disagreement involved to the word *"عاده"* - *"Habit or something one is accustomed to"* in the sentence *"فلهذا جفاهُ من كان عادهُ"* - *"Thus the one to whom it was accustomed forsook it"* which was annotated with the three labels *MRW-WIDLII*, *MRW-Indirect* and *MRW-Negative* twice each. The curation after the second round resulted in 90.49% *clear agreement* cases, 9.48% *majority agreement* cases (out of which 75.8% are *MRW-Negative*), and 0.04% *disagreement* cases. Table 3 and Table 4 describe the annotated dataset before and after the curation, respectively. We additionally report percentages of MRW-positive instances per text category, readership group, and domain in Figure 3.

Label	A1	A2	A3
MRW-Negative	4964	5055	5018
MRW-Indirect	266	152	240
MRW-Direct	7	37	8
MRW-Flag	6	8	4
MRW-Implicit	8	2	1
MRW-Direct/Flag	4	5	0
MRW-WIDLII	15	10	2
MRW-DFMA	7	8	4

Table 3: Distribution of Stage 1 MRW labels assigned by the three different annotators: A1, A2, A3.

Label	#	%
MRW-Negative	5106	96.76
MRW-Indirect	152	2.88
MRW-Direct	6	0.11
MRW-Flag	4	0.08
MRW-Direct, MRW-Flag	4	0.08
MRW-DFMA	2	0.04
MRW-WIDLII	1	0.02
Disagreement	2	0.04

Table 4: Distribution of Stage 1 labels in the curated dataset, showing the number (#) and percentage (%) of annotations for each label.

Stage 2 Span-level annotations were curated using a strict boundary-matching criterion. For each sentence, we compare the spans produced by the three annotators and retain only spans for which at least two annotators assigned the same label and the span boundaries either matched exactly or one span was fully contained within the other. Thus, cases of mere partial overlap were excluded, while minor boundary differences were accepted when

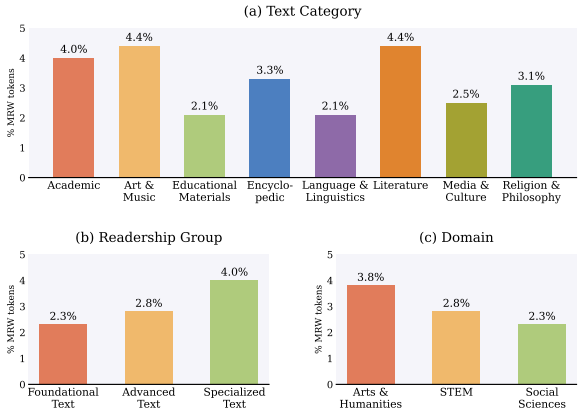


Figure 3: Percentage of MRWs across three corpus dimensions: text category, readership group, and domain.

one annotator selected a more specific subspan of another annotator's longer span. To curate conventionality, we average the score given by the three annotators (we report the lowest scores, i.e. least conventional metaphors, in Figure 8). Table 5 and Table 6 describe the annotated dataset before and after the curation, respectively.

Label	A1	A2	A3
isti'āra	180	139	181
kināya	70	41	46
tashbīh	9	9	9
isti'āra,kināya	1	0	0

Table 5: Distribution of annotated spans across Stage 2 labels assigned by the three different annotators: A1, A2, A3.

Label	#	Conventionality$_{AVG}$
isti'āra	106	4.05
kināya	22	3.67
tashbīh	8	2.40

Table 6: Distribution of curated spans by Stage 2 labels. (#) denotes the number of curated spans, and (Conventionality$_{AVG}$) denotes the mean conventionality score for each label (the higher the conventional).

6 Disagreement Analysis

To better understand the sources of annotator disagreements, we conduct a qualitative analysis of the disagreement cases from both annotation stages. We inspect the pairwise confusion patterns of *majority agreement* and *disagreement* labels (Figure 4) and manually examine the sentences with the highest degree of disagreement, identifying recurring linguistic and interpretive factors behind them.

Stage 1 Disagreements Only 502 tokens out of the corpus's 5277 tokens were annotated by two different labels (479 tokens) or three different labels (23 tokens) in stage 1. We see that the dominant confusion is MRW-indirect ↔ MRW-negative (812 instances [6]), followed by MRW-direct ↔ MRW-negative (51) and WIDLII ↔ MRW-negative (38). Thus, we infer that the central difficulty is MRW-positive vs. -negative status rather than distinguishing among positive subtypes. Within MRW-positive, MRW-direct ↔ MRW-indirect is the most frequent disagreement (31 instances). We take a closer look at the 23 full disagreement tokens, which map out to 19 unique sentences sourced mainly from the Literature; Art & Music category (36.8%) and Arts & Humanities domain (68.4%), and identify five main categories of disagreement reasons:

1. **Polysemy and domain-boundary ambiguity**: disagreement stems from deciding whether two senses of a word are distinct enough domains to count as metaphor, e.g. "خطوة" - *"step"* as interpreted in the sentence "ذكرت الكاتبة عدة [خطوات] في تقدير الذات" - *"The author mentioned several [steps] for self-appreciation"*.

2. **Classical or poetic register**: disagreements most probably occur because sentences require specialized knowledge of Classical Arabic or external exegesis. For example, the token "غررا" - *"blazes"* in "رأى [غررا] لها حسن وماء" - *"it saw in it [blazes] of beauty and luster"*.

3. **MWEs**: in some cases, figurativity attaches to a full expression rather than a single token, and annotators disagreed on which token(s) carry the figurative meaning. E.g., "وضع الأمور في [نصابها]" - *"putting things in their proper place"*.

4. **Dictionary gaps**: some tokens were absent from the reference dictionary and unattested elsewhere, causing annotation disagreement, e.g. *"العضلات [الهددية]" - "the ciliary(?) muscles"*, which is possibly an error in the original corpus.

5. **Cultural and religious influence**: some interpretations of metaphors depend on background knowledge or belief of an annotator, which in turn would cause disagreements, e.g. "أبواب الله" - *"the doors of God"*, where a literal vs. figurative reading depends on the annotator's own religious view.

Stage 2 Disagreements Out of the 136 curated spans, 45 had full agreement, 91 were resolved by majority vote, and 4 spans had complete disagreement (excluded from the final curated set). The most frequent source of disagreement is between a labeled span and unlabeled for both isti'āra and kināya, i.e., annotators disagreed on whether a span was figurative at all; tashbīh shows comparatively high agreement, likely due to its explicit comparator. Excluding unlabeled cases, the most common category of disagreement is isti'āra vs. kināya (44 out of 55 cases), which centers on whether the span's meaning can happen literally or not, as in the example *"مَسَ الْقَوْمَ قَرْحٌ" - "a wound has touched the people"*.

7 Discussion

7.1 Dictionary-related Challenges

Coverage and the Modern/Classical boundary. A recurring difficulty during annotation concerned the choice and coverage of the Arabic dictionary used to determine basic (non-contextual) word meanings. Although we deliberately selected a modern Arabic dictionary to match the contemporary nature of our corpus, the boundary between MSA and Classical Arabic vocabulary proved difficult to differentiate. Many words occurring in classical registers, including terms found in Quranic verses or traditional Arabic poetry, were absent from or inadequately treated in the dictionary. For example, the lexeme 'خصيلة' (a classical variant of 'خصلة') was not included with its classical sense, meaning annotators could not reliably retrieve the basic meaning of such tokens. While the comprehensiveness constraints of any lexicographic resource are understandable, this created systematic gaps in the annotation workflow. Future guidelines should either specify a broader set of approved reference resources or provide explicit fallback procedures for cases of dictionary absence.

Morphological complexity and root-based lookup. Arabic's rich morphological structure presents an additional challenge not addressed in

[6] An instance here means two different annotators assigning two different labels to the same token. A single token can have at most 3 instances.

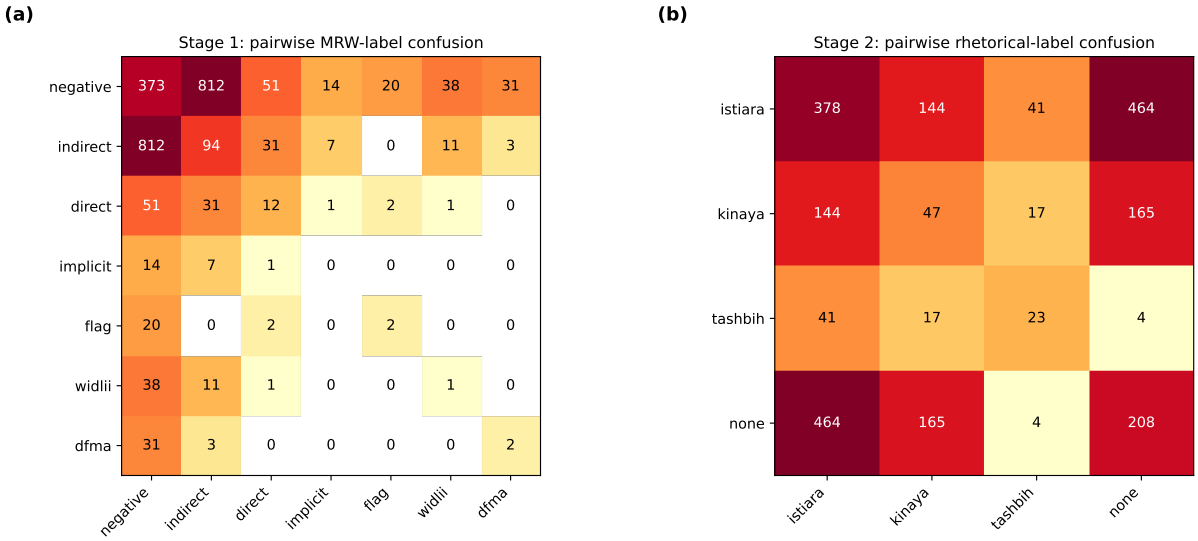


Figure 4: Pairwise label confusion matrices for (a) Stage 1 MRW labels and (b) Stage 2 rhetorical labels (*none* denotes cases where an annotator did not assign a label).

the original MIPVU framework. Arabic dictionaries are organised by root, meaning that identifying the basic meaning of a morphologically derived form requires the annotator to locate the appropriate root entry and then determine whether the meaning of the derivative is predictable from, or has diverged from, that of the root. In practice, we observed a non-trivial mismatch between the primary sense listed under a root entry and the conventional meaning of a derived nominal or verbal form. MIPVU does not specify how to handle such cases, leaving annotators to exercise individual judgement. We consider this a key area for adaptation in future versions of AraMIP, which should provide explicit decision criteria for root-to-derivative meaning transfer.

Inconsistency in sense ordering. Finally, although the dictionary we used states in its introduction that senses are ordered from most general to most specific, which is a convention that AraMI relies upon to identify the basic meaning of a lexical unit, we found this ordering to be inconsistently applied across entries in practice. This introduced further ambiguity into the sense selection step. Future work should either select resources with more reliable sense ordering, empirically validated if possible, or develop supplementary procedures that do not depend on lexicographic sense ordering alone.

7.2 Distribution of Metaphor-Related Words

The relatively low proportion of MRW-positive labels in the final corpus is expected given the composition of the sampled data. Our pilot sentences were drawn from a large-scale, domain-diverse corpus covering a wide range of MSA text types. The subset we used includes genres such as multiple-choice questions and scientific writing, which are predominantly denotative and thus inherently low in figurative density. Furthermore, our data is too small to ensure representativeness across the full range of MSA text types. These factors together limit the extent to which the distribution of annotation labels can be taken as indicative of broader figurative language use in Arabic. Larger-scale annotations covering a more balanced selection of domains will be necessary to characterize the figurative density of Arabic text more reliably.

7.3 Inter-Annotator Agreement

The IAA scores obtained in our pilot study fall below those reported for other MIPVU studies, including Fleiss' κ 0.84 for the original VUAMC, averaged across six reliability tests conducted over two years (Steen et al., 2010). This reflects several practical and methodological challenges in annotating figurative language, particularly in a low-resource setting such as Arabic.

Impact of label imbalance. The annotated sample contains only a small number of MRW-positive instances, whereas the annotators agreed on the negative label for more than 90% of the examples. Although this produces high observed agreement, it also increases the level of agreement expected by chance. Fleiss' κ adjusts for this chance agreement.

Therefore, because annotators are very likely to assign the dominant negative label, the resulting Fleiss' κ score may remain low despite the high percentage of observed agreement.

Annotation experience and iterative refinement. A key factor that contributes to disagreements is the limited number of annotation rounds. Due to time and resource constraints, annotators did not engage in the multiple cycles of annotation, discussion, and reconciliation that are usually recommended for developing new annotation guidelines (Artstein and Poesio, 2008). This led to inconsistencies in the interpretations of certain annotation rules. A prominent example is the label combination for direct metaphors. According to MIPVU, direct metaphors are typically accompanied by a corresponding *MFlag* label in the same sentence. However, a direct metaphor such as "أحمد أسد في الملعب" - *"Ahmad is a lion on the field"* - lacks an *MFlag* element. Whether this should be annotated as *MRW-direct* or *MRW-indirect* was not interpreted consistently by the annotators, and would likely be resolved through further discussion rounds.

Context availability and interpretation. Unlike the original MIPVU, which annotated sentences with access to their full surrounding discourse, our annotation setup did not provide annotators with context material. When contextual information was needed, annotators were permitted to independently consult external resources such as online exegetical works. While this allowed for a more informed interpretation of individual tokens, it introduced a source of systematic variance: different annotators may have consulted different resources or weighted conflicting interpretations differently. Future work building on AraMIP should supply all annotators with a shared, fixed set of contextual materials and restrict independent look-ups to ensure that contextual knowledge does not become a confounding variable.

Annotator diversity. The annotation team consisted of six annotators with varied disciplinary backgrounds. This interdisciplinary composition is a strength of the study in terms of coverage and perspective, but it also introduced variability in how figurative language was conceptualized and operationalized. Furthermore, while five annotators were native Arabic speakers, they represented four distinct geographic varieties of Arabic, and one annotator was an advanced L2 speaker specializing in Arabic linguistics. Dialectal variation and differences in rhetorical intuition may have influenced judgments. To assess whether this diversity in disciplines and dialects systematically affected annotation agreement, we conducted an analysis comparing the IAA across annotator backgrounds. The results showed that annotators, including the L2 speaker, had very similar average IAA scores, regardless of their disciplines or dialect backgrounds. In particular, pairs of annotators with similar or different dialects did not show any consistent pattern of higher or lower agreement. Detailed results of this analysis are provided in Appendix H.

8 Conclusion

In this work, we introduced AraMIP, the first systematic, two-stage procedure for identifying and classifying metaphor-related expressions in Arabic text. AraMIP extends MIPVU's lexical-unit-level metaphor identification with a second annotation layer that maps metaphor-related words onto the Arabic rhetorical tradition (*isti'āra*, *kināya*, and *tashbīh*), additionally recording the conventionality of each rhetorical construction on a five-point scale. We applied AraMIP to a pilot corpus of 300 sentences (5277 words) sampled from BAREC-10M across a balanced range of domains, readership levels, and text categories. Our IAA analysis shows that annotating Arabic figurative language reliably is more difficult than reported for several other MIPVU adaptations. We use this pilot study to identify concrete, addressable sources of this difficulty: a single annotation round without iterative guideline refinement, inconsistent dictionary coverage across the MSA/Classical Arabic boundary, unresolved root-to-derivative meaning transfer within Arabic's morphological system, inconsistent sense ordering within the lexicographic resource used, and uncontrolled access to contextual and exegetical materials during annotation. These findings constitute a diagnostic roadmap for future iterations of AraMIP, which should incorporate disagreement-informed guideline revisions, standardized contextual materials, and explicit root-based sense-lookup criteria. We release our pilot corpus and annotation guidelines to support this future work, scaling to larger and more representative corpora, extending coverage to dialectal Arabic, and training computational models for Arabic metaphor detection.

Limitations

This work has several limitations. First, the pilot corpus of 300 sentences (5,277 words) is too small to reliably characterize metaphor use across MSA. Second, AraMIP was applied only to MSA text, leaving open how the guidelines generalize to dialectal Arabic. Third, reliance on a single modern dictionary introduces gaps in Classical Arabic coverage, inconsistencies in sense ordering, and unresolved cases of root-to-derivative meaning transfer. Fourth, annotation guidelines were not iteratively refined through multiple rounds of discussion before the main annotation phase, and annotators were not provided standardized contextual or exegetical materials, independently consulting external resources such as Tafaseer when needed.

Acknowledgments

This research has been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – CRC-1646, project no. 512393437, project A05 and B02.

References

Hussein Abdul-Raof. 2006. *Arabic Rhetoric: A Pragmatic Analysis*. Routledge, London.

Manar Alkhatib and Khaled Shaalan. 2016. Natural language processing for arabic metaphors: a conceptual approach. In *International Conference on Advanced Intelligent Systems and Informatics*, pages 170–181. Springer.

Israa Alsiyat and Scott Piao. 2020. Metaphorical expressions in automatic Arabic sentiment analysis. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4911–4916, Marseille, France. European Language Resources Association.

Israa Alsiyat, Scott Piao, and Mansour Almansour. 2023. Arabic metaphor corpus (amc) with semantic and sentiment annotation. page 1. The twelfth International Corpus Linguistics Conference, CL2023 ; Conference date: 03-07-2023 Through 06-07-2023.

Aristotle. 1996. *Poetics*. Penguin Books, London.

Ron Artstein and Massimo Poesio. 2008. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596.

Yulia Badryzlova, Natalia Shekhtman, Yekaterina Isaeva, and Ruslan Kerimov. 2013. Annotating a Russian corpus of conceptual metaphor: a bottom-up approach. In *Proceedings of the First Workshop on Metaphor in NLP*, pages 77–86, Atlanta, Georgia. Association for Computational Linguistics.

Zouheir Banou, Sanaa El Filali, El Habib Benlahmar, Fatima-Zahra Alaoui, and Laila Eljiani. 2025. Memphis: Towards a new benchmark for arabic figurative speech classification. *Data Science*.

Beata Beigman Klebanov, Chee Wee (Ben) Leong, and Michael Flor. 2018. A corpus of non-native written English annotated for metaphor. In *NAACL*, pages 86–91, New Orleans, Louisiana. Association for Computational Linguistics.

Max Black. 1955. Metaphor. *Proceedings of the Aristotelian Society*, 55(1):273–294.

Verna Dankers, Marek Rei, Martha Lewis, and Ekaterina Shutova. 2019. Modelling the interplay of metaphor and emotion through multitask learning. 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 2218–2229, Hong Kong, China. Association for Computational Linguistics.

Erik-Lân Do Dinh, Hannah Wieland, and Iryna Gurevych. 2018. Weeding out conventionalized metaphors: A corpus of novel metaphor annotations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1424, Brussels, Belgium. Association for Computational Linguistics.

Markus Egg and Valia Kordoni. 2022. Metaphor annotation for German. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2556–2562, Marseille, France. European Language Resources Association.

Khalid N. Elmadani, Adel Mahmoud Wizani, Hanada Taha Thomure, and Nizar Habash. 2026. A large and balanced multi-domain arabic corpus annotated for morphology, syntax, and readability. In *Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026)*, pages 11761–11775, Palma, Mallorca, Spain. European Language Resources Association (ELRA).

Mengshi Ge, Rui Mao, and Erik Cambria. 2023. A survey on computational metaphor processing techniques: From identification, interpretation, generation to application. *Artificial Intelligence Review*, 56(Suppl 2):1829–1895.

Sam Glucksberg. 1998. Understanding metaphors. *Current Directions in Psychological Science*, 7(2):39–43.

Pere-Lluís Huguet Cabot, Verna Dankers, David Abadi, Agneta Fischer, and Ekaterina Shutova. 2020. The Pragmatics behind Politics: Modelling Metaphor, Framing and Emotion in Political Discourse. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4479–4488, Online. Association for Computational Linguistics.

Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych.

2018. The inception platform: Machine-assisted and knowledge-oriented interactive annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9. Association for Computational Linguistics. Event Title: The 27th International Conference on Computational Linguistics (COLING 2018).

George Lakoff and Mark Johnson. 1980. *Metaphors We Live By*. University of Chicago Press, Chicago.

Margaret Larkin. 1995. *The Theology of Meaning: 'Abd al-Qahir al-Jurjani's Theory of Discourse*, volume 79 of *American Oriental Series*. American Oriental Society, New Haven, CT.

Jens Lemmens, Ilia Markov, and Walter Daelemans. 2021. Improving hate speech type and target detection with hateful metaphor features. In *Proceedings of the Fourth Workshop on NLP for Internet Freedom: Censorship, Disinformation, and Propaganda*, pages 7–16, Online. Association for Computational Linguistics.

Chee Wee (Ben) Leong, Beata Beigman Klebanov, Chris Hamill, Egon Stemle, Rutuja Ubale, and Xianyang Chen. 2020. A report on the 2020 VUA and TOEFL metaphor detection shared task. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 18–29, Online. Association for Computational Linguistics.

Chee Wee (Ben) Leong, Beata Beigman Klebanov, and Ekaterina Shutova. 2018. A report on the 2018 VUA metaphor detection shared task. In *Proceedings of the Workshop on Figurative Language Processing*, pages 56–66, New Orleans, Louisiana. Association for Computational Linguistics.

Xiaofei Lu and Ben Pin-Yun Wang. 2017. Towards a metaphor-annotated corpus of Mandarin Chinese. *Language Resources and Evaluation*, 51(3):663–694.

Samar Mohamed Magdy, Fakhraddin Alwajih, Sang Yun Kwon, Reem Abdel-Salam, and Muhammad Abdul-Mageed. 2024. Gazelle: An instruction dataset for Arabic writing assistance. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 16027–16054, Miami, Florida, USA. Association for Computational Linguistics.

Yann Mathet, Antoine Widlöcher, and Jean-Philippe Métivier. 2015. The unified and holistic method gamma for inter-annotator agreement measure and alignment. *Computational Linguistics*, 41(3):437–479.

Saif Mohammad, Ekaterina Shutova, and Peter Turney. 2016. Metaphor as a medium for emotion: An empirical study. In *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*, pages 23–33, Berlin, Germany. Association for Computational Linguistics.

Ahmad Mukhtar Omar and 1 others. 2008. *Mu'jam al-lugha al-'arabiyya al-mu'aṣira*. 'Alam al-Kutub, Cairo, Egypt. Dictionary of Contemporary Arabic.

W. Gudrun Reijnierse, Christian F. Burgers, Tina Krennmayr, and Gerard J. Steen. 2018. DMIP: A method for identifying potentially deliberate metaphor in language use. *Corpus Pragmatics*, 2(2):129–147.

Elisa Sanchez-Bayona and Rodrigo Agerri. 2022. Leveraging a new Spanish corpus for multilingual and cross-lingual metaphor detection. In *Proceedings of the 26th Conference on Computational Natural Language Learning (CoNLL)*, pages 228–240, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

John R. Searle. 1979. Metaphor. In *Expression and Meaning: Studies in the Theory of Speech Acts*, pages 76–116. Cambridge University Press, Cambridge.

Gerard J. Steen, Aletta G. Dorst, J. Berenike Herrmann, Anna A. Kaal, Tina Krennmayr, and Trijntje Pasma. 2010. *A Method for Linguistic Metaphor Identification: From MIP to MIPVU*. Number 14 in Converging Evidence in Language and Communication Research. John Benjamins, Amsterdam.

The Pragglejaz Group. 2007. Mip: A method for identifying metaphorically used words in discourse. *Metaphor and Symbol*, 22(1):1–39.

Paul H Thibodeau and Lera Boroditsky. 2011. Metaphors we think with: The role of metaphor in reasoning. *PloS one*, 6(2):e16782.

Xiaoyu Tong, Ekaterina Shutova, and Martha Lewis. 2021. Recent advances in neural metaphor processing: A linguistic, cognitive and social perspective. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4673–4686, Online. Association for Computational Linguistics.

Ralf Vogel, Joana Cholin, Jutta Hartmann, Oliver Bott, Jana Häussler, Torgrim Solstad, Petra Wagner, Sina Zarrieß, Tanja Ackermann, András Bárány, Carola de Beer, Hendrik Buschmeier, Berenike Herrmann, Martina Hielscher-Fastabend, Farhat Jabeen, Barbara Job, Annett Jorschick, Johanna Kißler, Heike Knerich, and 12 others. 2026. Linguistic creativity in communication. conceptions and challenges. *Working Paper Series of the Collaborative Research Center CRC 1646*, 1.

Shun Wang, Ge Zhang, Han Wu, Tyler Loakman, Wenhao Huang, and Chenghua Lin. 2024. MMTE: Corpus and metrics for evaluating machine translation quality of metaphorical language. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11343–11358, Miami, Florida, USA. Association for Computational Linguistics.

Jingjie Zeng, Liang Yang, Zekun Wang, Yuanyuan Sun, and Hongfei Lin. 2025. Sheep's skin, wolf's deeds: Are LLMs ready for metaphorical implicit hate speech? In *Proceedings of the 63rd Annual*

Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 16657–16677, Vienna, Austria. Association for Computational Linguistics.

Hang Zhu, Mitoki Ohara, Rei Kikuchi, Kanako Komiya, Masayuki Asahara, and Sachi Kato. 2026. Automatic detection of metaphorical expressions in classical japanese using wlsp-enhanced bert. In *Proceedings of LT4HALA 2026 @ LREC 2026*, page 89–95, Palma, Mallorca, Spain. European Language Resources Association (ELRA).

Aseel Zibin, Nabeeha Binhaidara, Hala Al-Shahwan, and Haneen Yousef. 2025. Metaphor interpretation in jordanian arabic, emirati arabic and classical arabic: artificial intelligence vs. humans. *Humanities and Social Sciences Communications*, 12(1):1–12.

A Risks and Ethical Considerations

We do not believe that there are significant risks associated with this work, as we annotate existing data without content that might be perceived as hurtful.

B MIPVU Procedure

Steen et al. (2010) propose a standard procedure of identifying metaphors in written and spoken discourse. This procedure is presented in Figure 5.

C Annotation Tool

The stage 1 and stage 2 annotations of an example sentence using the INCEpTION interface are presented in Figures 6 and 7, respectively.

D Scientific Artifacts

In our work, we mainly used scientific artifacts in the form of publicly available datasets (Creative Commons Attribution Share Alike 4.0 International) and publicly available Python modules.

E Use of AI Assistants

AI assistants were used during manuscript preparation for specific linguistic reformulation to refine clarity and style, and to assist with code writing. AI-based search tools were used minimally to figure out the exact sources, contexts and/or interpretations of some vague sentences in the annotated data.

F License

The AraMIP annotation guidelines and the resulting corpus are released under the Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

G Recruitment Consent

This study used the six authors as annotators. A training session on the use of AraMIP was provided before annotations were performed. This included practice with a training dataset. Annotators were instructed to follow the AraMIP annotation guidelines to annotate Arabic sentences from the study corpus. Annotators were recruited on the basis of their ability to perform the annotation task. They freely consented to participate in the annotation, and did not receive any financial compensation.

H IAA Results Across Disciplines and Dialects

Table 7 shows details about the annotators' disciplinary and dialect backgrounds, along with their average pairwise Cohen's κ agreement with the other annotators. Figure 9 shows the individual pairwise Cohen's κ scores.

MIPVU 4-step Procedure

1. **Read the entire text-discourse to establish a general understanding of the meaning.**
2. **Determine the lexical units in the text-discourse.**
3.
   a. For each lexical unit in the text, **establish its meaning in context**, that is, how it applies to an entity, relation, or attribute in the situation evoked by the text (contextual meaning). Take into account what comes before and after the lexical unit.
   b. For each lexical unit, **determine if it has a more basic contemporary meaning** in other contexts than the one in the given context. Basic meanings are not necessarily the most frequent meanings of the lexical unit. For our purposes, basic meanings tend to be:
      i. More concrete [what they evoke is easier to imagine, see, hear, feel, smell, and taste];
      ii. Related to bodily action;
      iii. More precise (as opposed to vague);
      iv. Historically older;
   c. If the lexical unit has a more basic current contemporary meaning in other contexts than the given context, **decide whether the contextual meaning contrasts with the basic meaning** but can be understood in comparison with it.
4. **If yes, mark the lexical unit as metaphorical.**

Figure 5: The Metaphor identification procedure of Steen et al. (2010).

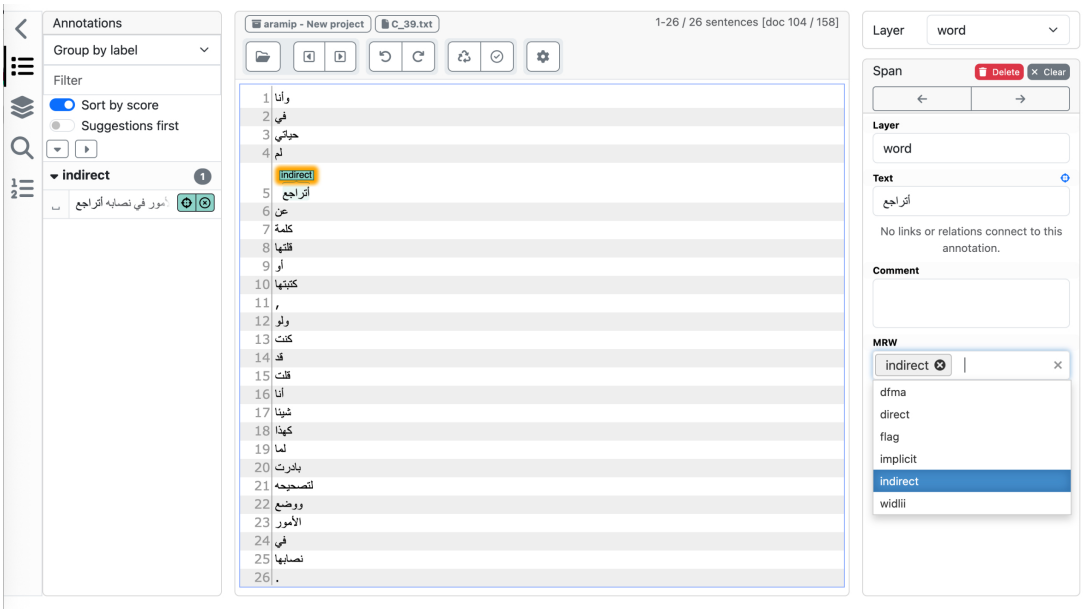


Figure 6: The INCEpTION annotation interface showing Stage 1 word-level MIPVU annotation, where each word is assigned one of seven codes.

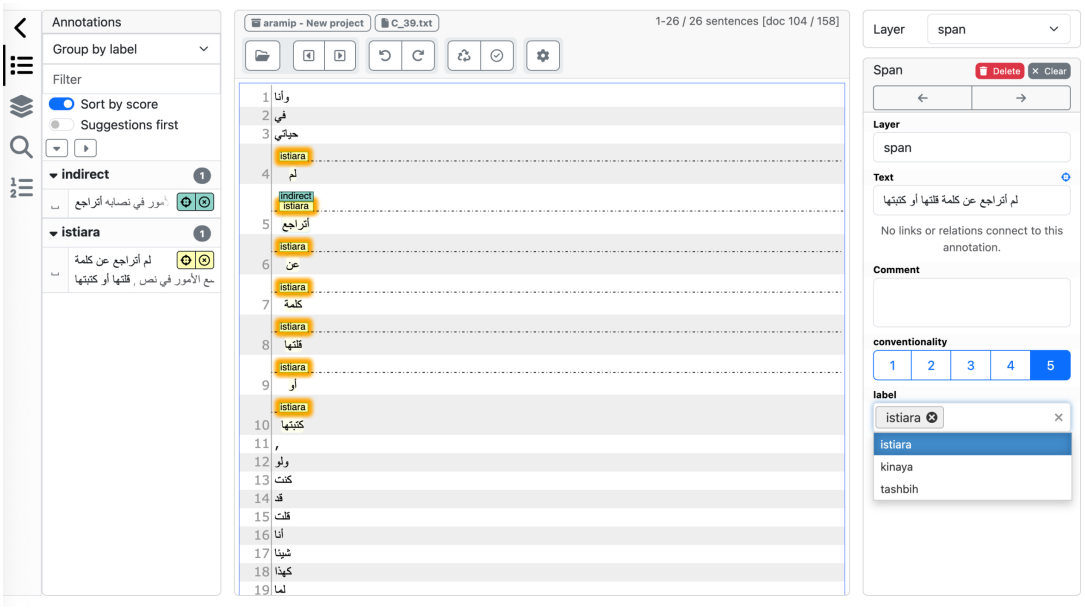


Figure 7: The INCEpTION annotation interface showing Stage 2 span-level rhetorical annotation, where annotators highlight spans and assign a rhetorical label (Isti'āra, Kināya, or Tashbīh) along with a conventionality rating.

Label	Mean Conventionality	Sentence	English Gloss
tashbīh	1.33	وساعدَ في التأكيد، أيضًا، شَعر المتكلم الذي وقف على جانب رأسه كالأشجار.	"The speaker's hair, which stood on the side of his head like trees, helped confirm it."
kināya	1.33	ولذلك أريد أن أحوِّل وجهي إلى الشمال الذي لا رائحة فيه.	"And so I want to turn my face toward the north, which has no scent to it."
isti'āra	1.50	إذا إستَجلى خَصائِلَهُ إختِياري رَأى غُرَراً لَها حُسنٌ وَماءُ	""If my choosing laid his qualities bare, it would see in them blazing marks of beauty and luster.""
kināya	1.50	29 اِحْمِلُوا نِيرِي عَلَيْكُمْ وَتَعَلَّمُوا مِنِّي، لأَنِّي وَدِيعٌ وَمُتَوَاضِعُ الْقَلْبِ، فَتَجِدُوا رَاحَةً لِنُفُوسِكُمْ.	"29 Take my yoke upon you and learn from me, for I am gentle and humble in heart, and you will find rest for your souls."
isti'āra	1.50	بما أنني معتاد على مواهبه الاستثنائية، كان الرجل مفاجأةً لا تنتهي بالنسبة إلي	"Since I was used to his exceptional talents, the man was a never-ending surprise to me."

Figure 8: Five stage 2 span annotations with the lowest mean conventionality scores (on a scale of 1-5). The annotated span shown here is the longest span between the annotations.

Annotaor	Arabic skills	Dialect	Gender	Expertise	Cohen's κ
A1	L1	Syrian	M	Computer Science	0.321
A2	L1	Palestinian & Syrian	F	NLP	0.330
A3	L1	Palestinian	F	NLP	0.398
A4	L1	Palestinian & Syrian	F	Cognitive Science	0.333
A5	L1	Egyptian	M	Computer Science	0.364
A6	C1/2 level	Jordanian	M	Arabic rhetorical and figurative language	0.340

Table 7: Annotators' disciplinary and dialect backgrounds, along with average pairwise Cohen's κ scores.

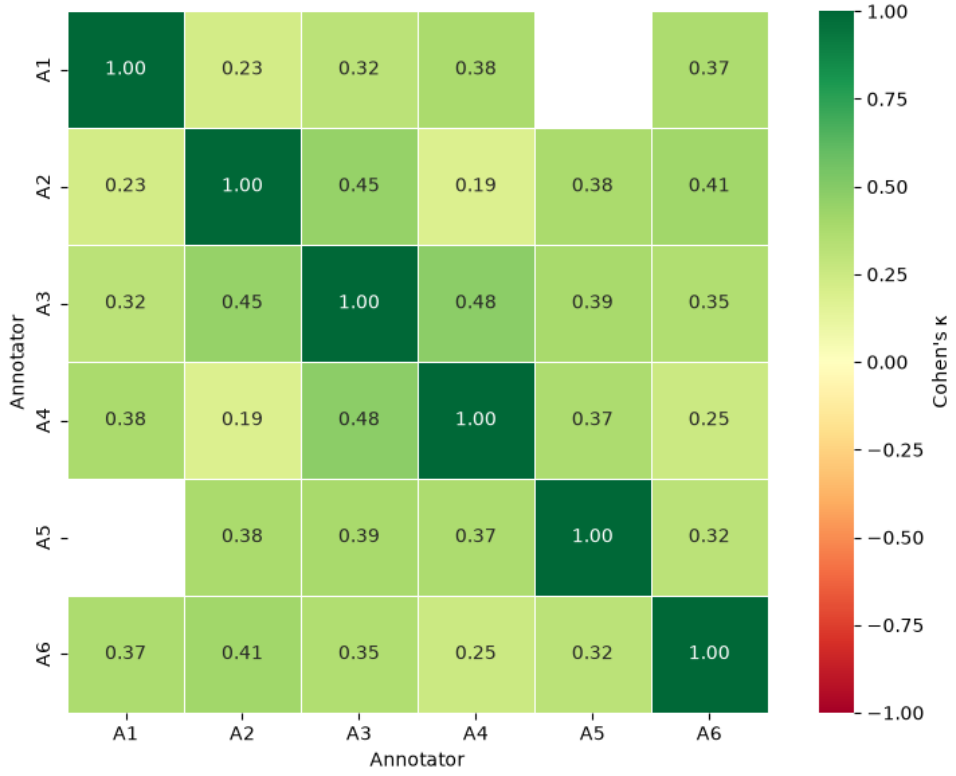


Figure 9: Pairwise Cohen's κ agreement.

AraScholar: Calibrating Academic Paraphrasing in Modern Standard Arabic

Omer Nacar[1] **Asma Al Wazrah**[2] **Rawan Nasser Almatham**[2]
Afrah Altamimi[2] **Abdulrahman M Alosaimy**[2]
[1]Tuwaiq Academy
[2]King Salman Global Academy for Arabic Language (KSAA)
o.najar@tuwaiq.edu.sa

Abstract

Academic paraphrasing in Modern Standard Arabic must change expression without distorting scientific content. We study this balance through AraScholar, a corpus of 20,002 manually written source–paraphrase pairs with 522 raw domain labels and item-level provenance. On 2,000 held-out sentences, fine-tuned AraT5v2 favors lexical reuse under beam search, while frontier models produce more extensive rewrites. With corrected Arabic punctuation normalization, diverse decoding and preservation-constrained reranking attain 74.25% coverage of a fixed novelty–preservation band, compared with 52.60% for zero-shot GPT-4o. Gold coverage is only 46.65%, showing that this band measures constraint compliance rather than human quality. Post-hoc target matching substantially improves a calibrated GPT-4o control and narrows the gap, although candidate budgets differ. Crucially, an independent 200-instance audit favors frontier models on meaning and paraphrase quality. We document the corpus, decoding controls, negative distillation result, and metric sensitivity, including test-derived calibration and checkpoint-selection limitations. The findings establish operating-point control as a useful capability while demonstrating why calibration must be evaluated alongside independent assessments of writing quality.

1 Introduction

Academic rewriting must change expression while preserving claims, qualifications, and technical terms. When the source is already scholarly prose, correction or connective substitution alone is insufficient. In Modern Standard Arabic (MSA), morphological and syntactic variation further complicates assessment with a single automatic score.

Two contrasting failures motivate *operating-point calibration*: source repetition with high semantic similarity, and fluent reformulation that changes the claim. We seek joint control of novelty and preservation under an annotation standard. Calibration here concerns rewriting behavior, not model confidence probabilities.

AraScholar provides 20,002 manually constructed academic source–paraphrase pairs. Four native Arabic experts performed collection, rewriting, linguistic review, and quality control. Each pair retains attribution to the quoted work and the document containing the quotation. The corpus has 522 raw domain labels, of which 513 occur in training. Its guidelines favor meaning-preserving, concise reformulation, but the observed distribution contains substantial variation. Accordingly, its mean behavior describes one annotation standard rather than a universal optimum for Arabic writing.

We compare decoding controls and reward-ranked fine-tuning of AraT5v2 ($\sim$300M parameters). Sampling, reranking, and a soft copy penalty counter beam-search reuse; the tested distillation does not. Frontier baselines instead tend toward more lexical change and longer outputs.

The evaluation yields two qualifications that are central to our conclusions. First, reranking optimizes measurements also used by the automatic evaluation, so higher coverage cannot establish independent quality superiority. Second, after correcting Arabic punctuation normalization, only 46.65% of gold rewrites satisfy the original fixed band. A selector can concentrate its outputs inside that region more often than humans while missing important properties of human writing.

We therefore examine gold-derived bands, held-out preservation encoders, distributional distance, calibrated frontier controls, and human ratings. A symmetric novelty-targeting rule improves both generators in a post-hoc control, narrowing the apparent small-model advantage. The independent audit reveals a clearer quality difference: frontier outputs receive higher ratings for meaning and

Proceedings of The Fourth Arabic Natural Language Processing Conference (ArabicNLP 2026), pages 340–355
October 28-29, 2026 ©2026 Association for Computational Linguistics

paraphrase quality. An automatic fluency constraint reduces one failure proxy, but its human benefit has not been evaluated.

Our contributions are:

- An expert-created MSA academic paraphrasing corpus with documented construction, item-level provenance, corrected domain accounting, and a controlled-access source-text policy.
- A joint novelty–preservation framework with corrected lexical measurement and explicit separation of fixed-band compliance, distributional resemblance, and human quality.
- An empirical analysis of decoding controls, calibrated frontier-model comparisons, and negative distillation results, including the limits introduced by test-derived targets and shared evaluation metrics.

The resulting claim concerns controllable rewriting. It does not equate a compact model's calibration advantage with superior academic prose.

2 Related Work

Style transfer and multilingual editing. Instruction-based rewriting allows models to express stylistic changes without task-specific adaptation. Reif et al. (2022) demonstrate augmented zero-shot style transfer, while mEdIT extends instruction tuning to multilingual grammatical error correction, simplification, and paraphrasing (Raheja et al., 2024). These tasks overlap with academic rewriting but do not specify how much change a scientific sentence should undergo. We discuss mEdIT as related work without claiming results for it on AraScholar.

XFORMAL evaluates formality transfer in Brazilian Portuguese, French, and Italian (Briakou et al., 2021). AraScholar starts from academic quotations rather than informal text. Its task is therefore scholarly reformulation; register elevation is not required in every example, and formality alone does not certify preservation of a scientific claim.

Novelty control and candidate selection. Chowdhury et al. (2022) introduce novelty-conditioned retrieval-augmented prompt tuning. We build on the broader motivation for controlling lexical change, examining its interaction with preservation and human quality in Arabic. Our distinction between novelty maximization and symmetric target matching matters when generators have different failure modes: increasing novelty can help a copier but harm an already expansive generator.

AraT5 establishes Arabic text-to-text modeling (Nagoudi et al., 2022); AraT5v2 is described within Octopus (Elmadany et al., 2023). We fine-tune an existing AraT5v2 checkpoint rather than pretrain a new generator. Nucleus sampling explores alternatives to a high-probability beam output (Holtzman et al., 2020), while reward-ranked fine-tuning converts selected outputs into supervision (Dong et al., 2023). Our negative result concerns one such configuration, not the general viability of distillation.

Evaluation of rewriting. BLEU, chrF++, and ROUGE measure overlap with a reference (Papineni et al., 2002; Popović, 2017; Lin, 2004). Sentence encoders such as LaBSE and GATE provide complementary semantic comparisons (Feng et al., 2022; Nacar et al., 2025). Neither perspective alone establishes acceptable paraphrasing: a single reference cannot cover every legitimate expression, and high sentence similarity can miss a changed qualifier or disfluency. We retain these measurements but interpret them alongside lexical change, explicit controls, and an independent human audit.

3 The AraScholar Corpus

3.1 Source selection and provenance

AraScholar contains 20,002 pairs (S, T), where S is a quotation extracted from academic writing and T is a manual MSA paraphrase. Sources include printed academic books, refereed journal articles, and master's and doctoral theses. The collection excludes the Qur'an, Hadith, and poetry. It targets academic quotations, not a random sample of all Arabic sentences or disciplines.

Collectors located quotations identified by quotation marks, author–year citations, or numbered footnotes. They extracted one complete, self-contained sentence and recorded its identifier and bibliographic context. Two provenance fields are nonempty in all 20,002 records: the citation of the *quoted work* and the *containing document* where the quotation was encountered. They contain 18,752 and 2,383 distinct strings, respectively. Distinct citation strings are not necessarily distinct

works: different pages or citation formats can refer to the same publication.

Quoted-work citations record author, title, publisher or journal, edition, year, and page. Sources average 29.3 words (median 26; P95 60; maximum 159). Attribution supports traceability but does not establish redistribution rights.

3.2 Manual rewriting and quality control

The team comprised four native Arabic speakers with academic-writing expertise, one responsible for each of collection, rewriting, linguistic review, and quality control. No LLM, machine translation system, or paraphrasing tool assisted the construction. The written methodology specifies five criteria: retain the quotation's context and meaning, change its wording, maintain cohesion and appropriate style, ensure grammatical and orthographic correctness, and keep the rewrite no longer than the quotation.

Linguistic review checks unique numbering, duplication, correctness of the extracted source, soundness of the rewrite, lexical difference with semantic agreement, and correctness of the rewritten text. Errors in printed source passages could be corrected during construction. Thus, the stored source is an editorially reviewed sentence, not necessarily a byte-for-byte transcription. A separate quality stage checks domain coverage, attribution, sentence length, and lexical overlap.

The length ceiling is a guideline rather than an empirically universal constraint: 61.8% of targets are no longer than their sources, while 38.2% exceed them. The corpus-wide target/source ratio has mean 0.990 and median 0.960. These statistics characterize a broadly concise corpus but do not justify declaring every expanded system output invalid. Mean sentence-level type–token ratio increases from 0.931 to 0.954, indicating greater lexical diversity without independently proving improved register.

The four-role review process is not a multiple-annotation agreement experiment. Individual degrees, recruitment and compensation arrangements, pilot procedures, rejection/adjudication rates, and double-annotation statistics are not reported. The output-quality evaluator described in Section 7.3 was independent of both corpus construction and model development.

Partition	Pairs	Raw labels
Training	16,002	513
Development	2,000	316
Test	2,000	317
Full corpus	20,002	522

Table 1: Fixed partitions and domain counts. The corpus has 495 labels after mechanical definite-article stripping; this is not a curated disciplinary taxonomy.

3.3 Splits and domain accounting

The fixed split is shown in Table 1. Exact source-string checks found no development or test source in training and no identity pair $S = T$. This verifies exact source disjointness, not separation by document, author, or semantic content. Related quotations or publications can still span partitions.

The earlier count of 513 applies only to training. Across the corpus there are 522 raw labels and 495 after stripping the definite article ال. This mechanical operation merges spelling variants such as لسانيات and اللسانيات; it does not consolidate related categories such as اجتماع and علم اجتماع. Corpus-wide median frequency is 9.5 pairs per raw label, with 51 singletons; training has median 8.0 and 62 singletons. Appendix A reports leading normalized labels. We make no delivered-data claim about the collection plan's separate broad-field taxonomy, for which no item-level mapping exists.

The long tail limits domain-level inference: we analyze subgroups only when they contain at least ten test examples, without claiming balanced coverage.

4 Task and Evaluation

A system produces P from source S without seeing the paired target T. References characterize the annotation distribution and support reference-based scoring. We use a 0–100 reporting scale for novelty and embedding similarity throughout. Cosine similarity is not a calibrated probability of semantic correctness and can, in principle, be negative.

4.1 Preservation and lexical novelty

For L2-normalized embeddings from LaBSE (E_L) and GATE-AraBERT (E_G), source preservation is

$$F(P,S) = 50\big[\cos(E_L(P), E_L(S)) + \cos(E_G(P), E_G(S))\big]. \quad (1)$$

Band membership uses this per-sentence mean, not a corpus average. Averaging encoders reduces de-

pendence on one representation but does not turn semantic similarity into an entailment or grammaticality test. We also retain reference similarity to T as a separate quantity: an identity prediction can resemble a reference that paraphrases the same source.

Let $n(x)$ remove Arabic diacritics and tatweel, normalize selected letters, remove punctuation, and split on whitespace. Letter mappings are أ، إ، آ→ا, ى→ي, ة→ه, ؤ→و, and ئ→ي. The set $U(x) = \text{types}(n(x))$ contains the distinct resulting word forms. For a nonempty prediction,

$$N(P, S) = 100 \left(1 - \frac{|U(P) \cap U(S)|}{|U(P)|}\right). \quad (2)$$

Copy ratio is $100 - N$. "Types" refers to this sentence-specific set, not an inventory of grammatical categories. The asymmetric denominator ignores source words omitted from P, and repeating a type does not alter the set. Novelty therefore measures lexical change, not information retention or syntactic restructuring. Appendix B gives a worked example and invalid-output considerations.

Punctuation correction. The original normalizer (v1) preserved the Arabic Unicode block in its punctuation filter. As a result, Arabic punctuation attached to a token could create a spurious new type. The corrected normalizer (v2) additionally removes the eight punctuation characters specified in Appendix B. We rescored every system's saved full-test outputs with v2; embedding preservation is unchanged. Corrected results are primary, while response-period experiments still using v1 are labeled explicitly. These lexical normalizations are not applied to fine-tuning inputs.

4.2 Fixed and gold-derived operating bands

The original fixed operating band is

$$\mathcal{B}_{\text{fixed}} = \{P : 40 \le N(P, S) \le 70, \; F(P, S) \ge 85\}. \quad (3)$$

Coverage is the percentage of outputs satisfying both conditions. Under v2, only 46.65% of gold rewrites are inside this band. We therefore interpret it as a fixed compliance target, not a "human-quality band." Higher coverage can suppress valid human variation.

A second, descriptive band uses the tenth and ninetieth percentiles of test-gold novelty and the tenth percentile of test-gold preservation:

$$\mathcal{B}_{\text{gold}} = \{P : 33.33 \le N(P, S) \le 73.68, \; F(P, S) \ge 76.18\}. \quad (4)$$

The percentile bounds are reported to two decimal places. Gold coverage is 74.65%, since marginal percentile conditions do not imply the same joint coverage. Both band scores are shown in the main table. Because Eq. 4 uses test references, it is a post-hoc distributional diagnostic, not a development-calibrated deployment criterion. We do not claim that this reference contact is eliminated by calling the band descriptive.

Historical diagnostic flags use $N < 25$ for copy-prone outputs, $F < 80$ for drift, and $N > 80$ for over-rewriting. These names denote threshold flags, not verified human error labels. They overlap and do not partition all out-of-band cases. No selection filter is applied to the evaluation denominator: every system's final output is scored, including fallback outputs.

4.3 Independent and complementary measurements

We evaluate preservation with three encoders not used by the reranker, each using its own test-gold tenth-percentile floor. Alternative novelty units include light stems, root stems, and character 4-grams. These checks test sensitivity to the measurement representation; they do not constitute a gold-standard lemma-level analysis.

We also compare the joint (N, F) distributions through energy distance, with each coordinate scaled by the corresponding gold standard deviation. This distinguishes concentration inside a band from resemblance to the full annotation distribution. Its exact estimator and sampling are specified in Appendix G.

BLEU, chrF++, and ROUGE-L against T remain secondary reference-overlap measurements, with sacreBLEU where applicable (Post, 2018). ROUGE for reporting uses Arabic-normalized whitespace tokens without English stemming. High target BLEU does not inherently indicate copying, and low BLEU does not establish good paraphrasing. Independent five-point human ratings of meaning and paraphrase quality supply the most direct assessment of perceived output quality.

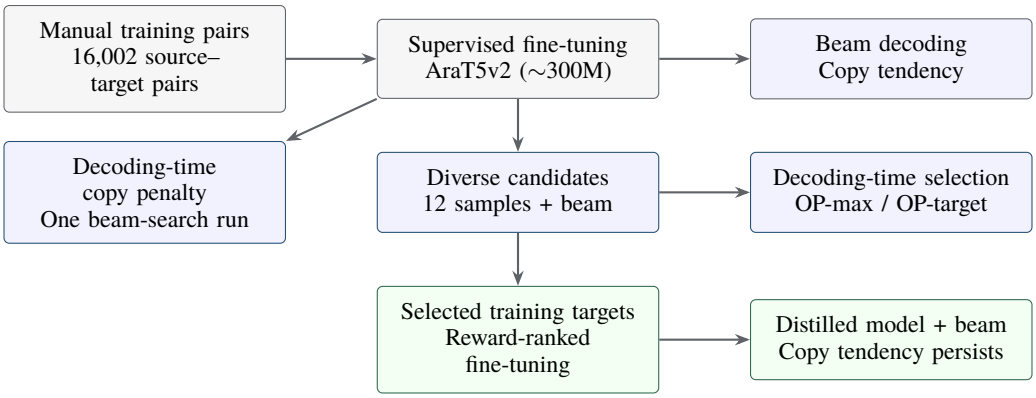


Figure 1: Decoding-time controls (blue) and reward-ranked fine-tuning (green) are separate routes. OP-max favors greater novelty; OP-target favors proximity to a chosen novelty level. Distillation uses training sources only. Human evaluation is independent of these optimization routes.

5 Methods

5.1 Base fine-tuning & candidate generation

The base model is `UBC-NLP/AraT5v2-base-1024`. We fine-tune for five epochs at learning rate 5×10^{-5}, with per-device train/evaluation batch size 8, label smoothing 0.1, weight decay 0.01, warm-up ratio 0.1, and fp16 precision. Maximum source and target lengths are 768 and 512 model tokens. Inputs retain their orthography and use `paraphrase: {domain}: {text}`; an ablation omits the domain.

The original run selected its checkpoint using development `eval_rougeL` computed by an English-oriented tokenizer and Porter-stemmer configuration. That criterion is unsuitable for Arabic and is a limitation of model selection. All reported evaluation scores are computed separately, but correct rescoring does not undo the possible effect of selecting a suboptimal checkpoint. We evaluate the resulting checkpoint transparently rather than imply that it was selected with the corrected metric.

Beam decoding uses five beams, no-repeat-trigram blocking, and repetition penalty 1.2. Diverse generation adds 12 nucleus samples with temperature 1.1 and top-$p = 0.95$, using the same repetition controls. Outputs are deduplicated while preserving their order: sampled candidates first, deterministic beam output last. The maximum generation length is 512 new tokens. Figure 1 places this inference procedure separately from further fine-tuning.

5.2 Feasibility, selection, and fallback

For a source S, let $t(x)$ be its normalized whitespace-token sequence and

$$\ell(P, S) = \frac{|t(P)|}{\max(1, |t(S)|)}. \tag{5}$$

The recorded selection runs use v1 lexical features; the primary table applies v2 only when evaluating their saved outputs. This is a normalized word-count ratio, not a SentencePiece length ratio. A candidate is degenerate if it has fewer than three tokens, any adjacent equal tokens, or type–token ratio below 0.55. The initial feasible set is

$$\begin{aligned}\mathcal{F}(S) = \{P : F(P, S) \geq 88,\\ 0.6 \leq \ell(P, S) \leq 1.5,\\ P \text{ is non-degenerate}\}.\end{aligned} \tag{6}$$

The preservation floor was selected on development data from $\{80, 82, 84, 85, 86, 88, 90\}$ and then fixed.

The original rule, OP-max, selects

$$P^{\star}_{\text{max}} = \arg\max_{P \in \mathcal{F}(S)} N(P, S). \tag{7}$$

It is useful for under-rewriting but does not enforce either novelty boundary of the evaluation band. It also shares its scoring functions with evaluation, giving it an explicit metric advantage.

The direction-neutral alternative, OP-target, selects

$$P^{\star}_{\text{target}} = \arg\min_{P \in \mathcal{F}(S)} |N(P, S) - N^{\star}|. \tag{8}$$

The response-period experiment uses $N^{\star} = 56.1$, the *v1 test-gold mean*. This target was not

development-derived. Consequently, OP-target is evaluated as a post-hoc analysis and is not substituted for the corrected OP-max full-test result. Its exact target also cannot be carried unchanged across normalizers and called the same human operating point: v2 gold mean novelty is 52.86.

For either rule, an empty feasible set triggers the same cascade: remove the preservation floor; then remove the length window; finally, if all candidates are degenerate, return the candidate with maximum preservation over the entire pool. Retain the stated selection objective at the first nonempty stage. Ties in OP-target are resolved by the first candidate in generation order, as implemented; no preservation-based secondary tie-breaker is added retrospectively.

5.3 Fluency-constrained selection

A diagnostic extension adds a fluency floor to the feasible set. The scorer is `ALLaM-AI/ALLaM-7B-Instruct-preview`, run in bf16 with batch size 16 and truncation to 128 tokens. It assigns mean next-token log-likelihood over non-padding prediction positions. Higher scores indicate greater likelihood under this model, not guaranteed academic quality.

The threshold, -4.4344, is the tenth percentile of the 2,000 test-gold scores (mean -3.6022). It is therefore another test-derived constraint. When the feasible set is empty, fluency is relaxed first, followed by preservation, length, and degeneracy. Selected outputs can consequently remain below the fluency floor. This extension is reported as an automatic diagnostic; its outputs have not received a new human audit.

5.4 Copy penalty and reward-ranked fine-tuning

For generation logit $z_t(v)$ and token-ID set V_x from the tokenized input, the copy penalty is

$$\widetilde{z}_t(v) = z_t(v) - 1.5\,\mathbf{1}[v \in V_x]. \tag{9}$$

It is a soft bias at every decoding step, with no decay or stop-list. The original implementation takes V_x from the *full encoded prompt*: source text, task prefix, domain, and special tokens. It thus penalizes more than source content, including function words, clitics, and punctuation. A separate ablation constructs a source-only mask without prefix or special tokens and tests function-word exemption. These are distinct implementations and their results remain separate. The penalty value was chosen on development data. One beam-search run is required, rather than a sampled pool and two encoder scorers; measured latency or cost is not reported.

Reward-ranked fine-tuning generates and ranks candidates on 8,000 training sources. Candidate targets are retained for 82% of examples; otherwise the gold rewrite supplies the target. The base model is fine-tuned by cross-entropy using these targets (Dong et al., 2023), with the ablation/RAFT trainer using learning rate 3×10^{-5} and three epochs. This is distinct from the five-epoch primary fine-tuning run. Failure under beam decoding tests this configuration, not the general ability of cross-entropy or distillation to learn novelty control. A semantic loss computed from hard argmax IDs would require a gradient estimator or relaxation to affect parameters; we do not present an unreported dual-loss system.

6 Experimental Protocol

Systems and snapshots. The primary comparison uses all 2,000 held-out pairs. Baselines are unadapted AraT5v2, GPT-4o, and Claude Sonnet 4.6; an additional response-period experiment evaluates ALLaM-7B-Instruct. The unadapted AraT5v2 is not instruction-tuned, so its poor behavior under a long Arabic instruction cannot establish a general failure of off-the-shelf Arabic models. Commercial identifiers are `gpt-4o-2024-08-06` and `claude-sonnet-4-6`, queried on 18 June 2026. Original LLM generation uses temperature 0.7 and a 1,024-token output limit. Appendix C provides the original Arabic instruction and the calibrated control's constraints.

Frontier control. The best-of-eight GPT-4o experiment uses 498 test sources: 166 evenly stepped examples from each source-length tertile. The requested sample size of 500 became 498 through integer division; no examples were lost to failed calls. Both generators are evaluated on exactly this intersection of source strings. GPT-4o receives the calibrated instruction and produces eight candidates per source at temperature 0.9. AraT5v2 uses its ordinary task/domain prefix and the existing sampled pool, averaging 12.92 candidates after deduplication. The shared selector thus reduces objective asymmetry, but prompt information and candidate budgets remain different. Neither direction of advantage justifies calling this a fully matched

comparison.

Protocol separation. The primary table uses corrected v2 novelty. The 498-source control, fluency experiments, held-out encoder results, and sampling-seed analyses retain their original v1 scores and explicit test-reference contact. We do not merge them into a new corrected leaderboard or treat them as prospectively specified tests. All final outputs remain in each evaluation denominator, regardless of selection-filter compliance.

Local computation uses one NVIDIA RTX 5090 with 32 GB memory; commercial baselines use hosted services. Three candidate-pool regenerations measure decoding variability for one fine-tuned checkpoint, not training-seed variance. The model repository is linked here.[1] Exact generation settings and supplementary scoring details appear in the appendices. Encoder revision hashes were not pinned in the recorded runs, which limits exact replication as repositories evolve.

7 Results

7.1 Corrected full-test calibration

Table 2 replaces the original full-test lexical results with the corrected scores. OP-max reaches 74.25% fixed-band coverage, compared with GPT-4o's 52.60%, a difference of 21.65 percentage points. Under the descriptive gold-derived band, the respective rates are 85.95% and 74.25%; gold itself reaches 74.65%. Neither criterion establishes a human-quality ranking.

The correction changes the magnitude of the original result. OP-max's novelty falls from 55.70 to 51.87, and fixed-band coverage from 76.65% to 74.25%. GPT-4o's novelty falls from 65.14 to 62.81, bringing more outputs under the fixed upper boundary and increasing coverage from 49.35% to 52.60%. Gold novelty falls from 56.06 to 52.86. The normalization sensitivity must therefore be propagated into the gold reference as well as system scores. Appendix D reports every change.

The correction evaluates the same generated strings under a repaired measurement. Using v2 inside candidate screening and selection could also change which outputs are chosen and would require a separate reranking experiment. We make no claim for that unmeasured variant.

Fine-tuned beam outputs remain copy-prone: corrected novelty is 21.72% with domain tags and 21.71% without, with fixed coverage 5.20% and 4.95%. RAFT reduces novelty further to 18.95% and coverage to 3.30%. By contrast, the unadapted model's high nominal novelty is accompanied by preservation near 11 and length ratio 0.18. None of its 2,000 outputs contains the affected Arabic punctuation, so its unchanged novelty after correction is compatible with degeneration rather than evidence of successful rewriting.

The copy penalty reaches 55.55% novelty and 54.45% fixed coverage with no encoder-based selection at inference. OP-max is more concentrated inside the band and has mean novelty close to the corrected gold mean. This shows that the sampled pool contains useful alternatives to the beam output. It does not establish that the model's remaining quality limitations are solely a decoding artifact.

7.2 Calibrated frontier controls

The earlier instruction-only control reports GPT-4o coverage improving from 50.2% to 61.4% on these sources. Table 3 separately reports selection within the temperature-0.9 candidate pools; its single-candidate value is not the same generation as that instruction-only run.

OP-max is poorly suited to the calibrated GPT-4o pool: maximizing novelty yields 47.99% coverage, compared with 81.33% for target matching. AraT5v2 improves from 78.71% to 85.34% under the same change, leaving a 4.01-point gap. This supports a direction-neutral selection objective over novelty maximization when generators differ in how much they rewrite.

However, the comparison is post-hoc and not budget-matched. AraT5v2 has more candidates, while GPT-4o receives explicit novelty and length instructions. Both selectors use a target derived from test-gold v1 novelty. The result cannot establish compact-model superiority under equal computation or a fully unseen calibration protocol. It does show that supplying a stronger frontier control substantially changes the original interpretation.

7.3 Independent human quality assessment

A native Arabic PhD holder, independent of corpus construction and model development, rated the three systems on 200 test instances. Outputs were evaluated against the gold academic rewrite, with the source available as context. Two five-point ordinal dimensions assess meaning preservation and

[1] https://huggingface.co/Omartificial-Intelligence-Space/AraScholar

System	Preservation LaBSE	Preservation GATE	Novelty v2 (%)	Length ratio	In-band (%) Fixed	In-band (%) Gold-derived
Human reference	85.7	87.8	52.86	1.00	46.65	74.65
AraT5v2, zero-shot	11.4	10.4	75.46	0.18	0.00	0.00
GPT-4o, zero-shot	86.3	86.5	62.81	1.14	52.60	74.25
Claude Sonnet 4.6	82.1	83.0	73.76	1.46	22.15	41.90
AraT5v2, beam + domain	93.5	94.2	21.72	0.93	5.20	16.05
AraT5v2, beam, no domain	93.4	93.9	21.71	0.93	4.95	16.00
AraT5v2, RAFT + beam	94.3	94.6	18.95	0.93	3.30	9.60
AraT5v2, copy penalty	86.7	87.4	55.55	0.95	54.45	77.70
AraT5v2, OP-max	88.1	90.5	51.87	0.97	**74.25**	**85.95**

Table 2: Results on 2,000 test sources with corrected v2 novelty and coverage; preservation and length are unchanged. Coverage uses the fixed band or descriptive test-gold quantiles. Human scores compare gold with source; bold marks highest system coverage, not quality or significance. Copy penalty uses the original prefix-inclusive mask.

Selection on 498 sources	GPT-4o	AraT5v2
Single candidate	55.22	44.38
OP-max	47.99	78.71
OP-target	81.33	85.34
Band-membership oracle	86.14	91.16

Table 3: Post-hoc control on shared sources (v1, fixed band). GPT-4o uses calibrated prompts and eight candidates; AraT5v2 uses its ordinary prefix and 12.92 on average. OP-target uses test-gold novelty (56.1); oracle coverage indicates an in-band candidate exists.

System	Mean rating Meaning	Mean rating Para.	Rating ≤ 2 Meaning	Rating ≤ 2 Para.
AraT5v2, OP-max	3.45	2.82	33	69
GPT-4o	4.29	3.80	6	4
Claude Sonnet 4.6	4.38	3.71	6	6

Table 4: Independent single-rater audit on 200 test instances per system. Means use five-point scales; low-rating counts are out of 200. The audited small-model outputs are OP-max, not OP-target or the fluency-constrained variant.

paraphrasing quality; scores of at most two are flagged as severe.

The frontier models receive higher ratings on both dimensions (Table 4). OP-max has 69 low paraphrase-quality ratings, compared with four for GPT-4o and six for Claude. Its higher automatic calibration therefore does not translate into superior perceived writing quality in this audit. The discrepancy is a main finding, not a minor exception to the automatic comparison.

The audit's sampling rule, identity blinding, presentation randomization, and verbatim scale anchors were not recorded in the available protocol. We do not infer them from a separate evaluation-sheet script covering a different experiment. One

Held-out encoder	Gold	GPT-4o	OP-max
Multilingual E5	75.00	71.65	85.95
Multilingual MiniLM	74.05	73.80	84.80
Arabic Triplet V2	74.60	71.15	86.25

Table 5: Coverage (%) with held-out encoders, v1 novelty bounds $[35.71, 75.77]$, and encoder-specific test-gold P10 floors. Checkpoints and thresholds are in Appendix E.

rater provides no inter-rater reliability estimate, and a visible reference may anchor judgments toward a particular acceptable wording. These limits qualify the audit without erasing the observed quality gap.

7.4 Fluency and evaluation robustness

The OP-max selector can choose semantically similar but poorly formed candidates. Consistent with this mechanism, 44.85% of its outputs fall below the test-gold tenth-percentile LM score. Under v1, target matching with the fluency floor lowers this fraction to 15.30%, with 74.05% fixed-band coverage and 0.05% over-rewriting. Gold's below-floor rate is 10% by construction. The selector relaxes fluency when necessary, explaining why the final below-floor rate exceeds zero.

The full ablation in Appendix F shows a trade-off: unfiltered target matching reaches 83.90% full-test fixed coverage but leaves 43.35% below the fluency threshold. Adding fluency improves that proxy at the expense of band coverage. No corresponding human re-rating has been completed, so neither human-quality improvement nor closure of the frontier gap is claimed. The 498-source 85.34% and full-test 83.90% are separate denominators, not conflicting estimates of the same run.

The held-out encoders retain OP-max's coverage advantage (Table 5). This weakens the explanation that the result depends uniquely on the two selection encoders. It does not eliminate metric dependence: these analyses still use related embedding similarities, lexical novelty, and test-gold thresholds. Rankings also persist with light-stem, root-stem, and character-4-gram novelty; no numerical lemma-level result is claimed.

Energy distance yields a different ordering: the copy penalty is closest to gold (0.0163), followed by OP-max (0.0992), GPT-4o (0.2139), and Claude (0.9373). Independent gold subsamples give 0.0017 as a finite-sample reference. Thus, OP-max leads on band coverage while the copy penalty better matches the distribution under this statistic. The complete estimator and results are retained in Appendix G.

7.5 Additional controls and negative results

Exempting function words from the source-only copy penalty yields 56.45% v1 coverage and improves preservation from 86.97 to 88.76. This ablation differs from the prefix-inclusive primary implementation: the two all-token variants produce different outputs on 296 of 2,000 inputs (Appendix H). Across three candidate-pool regenerations, OP-max v1 coverage ranges from 77.15% to 77.75% (population SD 0.27 points); the original 76.65% comes from a separate run. These results characterize limited decoding variability, not fine-tuning variance.

Domain tags have little effect, increasing corrected beam coverage from 4.95% to 5.20%. Across 36 domains with at least ten examples, exploratory v1 OP-max coverage averages 75.2% (between-domain SD 12.0 points), without establishing uniform performance across all labels. ALLaM-7B-Instruct achieves 29.7% v1 coverage, extending the Arabic-specific baselines. Finally, RAFT lowers validation loss but reduces corrected coverage from 5.20% to 3.30%. Thus, the tested distillation configuration does not resolve beam-search copying; this finding does not rule out better targets, alternative objectives, or other decoding strategies.

8 Discussion

What calibration establishes. Joint novelty and preservation reveal failures that either axis alone can hide. The corrected results demonstrate control of measurable rewriting behavior, while gold's low fixed-band coverage shows that a narrow rectangle cannot define paraphrase correctness. Gold-derived percentiles better describe the annotation distribution, but using test references makes that analysis retrospective. A deployment evaluation should fix its target and thresholds on development data before inspecting test outcomes.

What it does not establish. The human audit favors frontier writing even when OP-max has higher calibration scores. A fluency floor targets one missing dimension but neither proves semantic correctness nor substitutes for human re-evaluation. Similarly, a 300M generator does not make the complete reranking pipeline a 300M system: it also generates multiple candidates and invokes two encoders. Parameter count alone therefore cannot substantiate a measured cost advantage.

Examples and error interpretation. Appendix J presents two test pairs near median v1 gold novelty: neither gold rewrite meets the fixed preservation floor, and only one OP-max output is in-band, illustrating that typical human rewrites need not satisfy the band. Among 64 flagged beam outputs, 55 involve copying, supporting the metric on clear failures without establishing test-wide prevalence. Unreported fallback frequencies limit interpretation of constraint satisfaction.

9 Conclusion

AraScholar contributes 20,002 manually constructed academic paraphrase pairs and a framework for examining lexical novelty together with semantic preservation. Corrected evaluation shows that decoding controls substantially change a compact model's operating behavior, while the tested distillation configuration remains copy-prone under beam search. Calibrated frontier controls, gold-derived diagnostics, and an independent human audit qualify the original comparison: control over automatic constraints is useful, but is not equivalent to better academic paraphrasing. Progress requires development-calibrated objectives, matched generation budgets, and independent evaluation of scientific adequacy and linguistic acceptability.

Limitations

AraScholar evaluates sentence-level rewriting and does not assess document coherence, citation in-

tegrity, or terminology consistency. Its quotation-based collection, uneven domain coverage, and annotation guidelines limit representativeness; source-level disjointness also does not establish document-level or semantic deduplication. Some annotator and quality-control details remain undocumented. Several calibration targets and thresholds use test references, making those analyses post-hoc, while supplementary v1 experiments retain the punctuation issue corrected in the main evaluation. Shared selection and evaluation metrics further limit claims about independent quality. Reproducibility is constrained by unsuitable Arabic ROUGE checkpoint selection, unpinned encoder revisions, unequal candidate budgets, and missing fine-tuning variance, significance, and inference-cost measurements. The human audit involves one evaluator with incompletely recorded procedures, and the fluency-constrained outputs have not been reassessed by humans. Hosted-model changes and restricted source access impose additional constraints.

Ethics

AraScholar supports academic rewriting with appropriate attribution: paraphrasing does not remove citation obligations or establish permission to reuse copyrighted material. Authors remain responsible for checking meaning, terminology, and unsupported claims. Each corpus item records its quoted work and containing document, but attribution alone does not establish redistribution rights. Source quotations are designated for controlled research access, without a blanket open license; the guide, code, and annotations require terms consistent with contributor rights. Model availability is separate from corpus access, and the manuscript package contains only the attributed quotations used as examples.

References

Eleftheria Briakou, Di Lu, Ke Zhang, and Joel Tetreault. 2021. Olá, Bonjour, Salve! XFORMAL: A benchmark for multilingual formality style transfer. In *Proceedings of NAACL-HLT*, pages 3199–3216.

Jishnu Ray Chowdhury, Yong Zhuang, and Shuyi Wang. 2022. Novelty controlled paraphrase generation with retrieval augmented conditional prompt tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, KaShun Shum, and Tong Zhang. 2023. RAFT: Reward rAnked FineTuning for generative foundation model alignment. *Transactions on Machine Learning Research*.

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.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of ACL*, pages 878–891.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81.

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.

El Moatez Billah Nagoudi, AbdelRahim Elmadany, and Muhammad Abdul-Mageed. 2022. AraT5: Text-to-text transformers for Arabic language generation. In *Proceedings of ACL*, pages 628–647.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of ACL*, pages 311–318.

Maja Popović. 2017. chrF++: Words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191.

Vipul Raheja, Dimitris Alikaniotis, Vivek Kulkarni, Bashar Alhafni, and Dhruv Kumar. 2024. mEdIT: Multilingual text editing via instruction tuning. In *Proceedings of NAACL-HLT (Volume 1: Long Papers)*, pages 979–1001.

Emily Reif, Daphne Ippolito, Ann Yuan, Andy Coenen, Chris Callison-Burch, and Jason Wei. 2022. A recipe for arbitrary text style transfer with large language models. In *Proceedings of ACL (Volume 2: Short Papers)*, pages 837–848.

A Corpus Documentation and Domains

Provenance. The field مصدر الاقتباس contains the quoted work's citation, while المصدر identifies the containing document in which the quotation was found. Both are nonempty for every record. The 18,752 distinct quoted-work strings and 2,383 containing-document strings should not be interpreted as bibliographically deduplicated publication counts. Gregorian years parsable from the citations span 123 distinct values, with median 2002 and tenth/ninetieth percentiles 1978/2016; these summaries concern parsable years rather than every citation.

Annotation process. Collection, manual rewriting, linguistic review, and quality control were assigned to four separate team roles. The methodology requires complete quotations, changed wording with preserved meaning, cohesion, linguistic correctness, and a target no longer than its source. Review can correct errors in the printed source. The observed 61.8% length-compliance rate is reported instead of treating the guideline as an invariant. No multiple-rater agreement statistic or quantified rejection rate is available. Missing personnel and review statistics are not reconstructed from the final corpus.

Inventory. The experiment package's `domain_inventory.json` records raw labels with split counts. Definite-article stripping yields 495 normalized labels from 522 raw forms; the mapping is mechanical, not a curated ontology. Table 6 shows the leading normalized labels. The 21 broad categories in the collection plan have no per-item mapping in the delivered corpus, so their planned allocations are not used as corpus statistics.

B Lexical Normalization

The v1 punctuation regular expression preserved characters in the Arabic Unicode block U+0600–U+06FF, including punctuation. The v2 correction additionally strips U+060C, U+061B, U+061F, U+066A, U+066B, U+066C, U+066D, and U+06D4: Arabic comma, semicolon, question mark, percent sign, decimal separator, thousands separator, five-pointed star, and full stop. Other normalization steps are unchanged. The correction rescored saved predictions; it did not regenerate or reselect candidates.

Label	English gloss	Pairs
لغة	Language	1,583
أدب	Literature	912
فلسفة	Philosophy	858
قانون	Law	851
أدب عربي	Arabic literature	807
بلاغة	Rhetoric	700
علم اجتماع	Sociology	647
اجتماع	Society / sociology	629
لسانيات	Linguistics	548
نحو	Grammar	501

Table 6: Leading corpus labels after definite-article stripping. Related categories remain separate because no manual semantic consolidation was performed.

For an illustrative source يدرس الطالب اللغة العربية ("The student studies the Arabic language") and prediction يتعلم الطالب العربية ("The student learns Arabic"), the normalized type sets are

$$U(S) = \{\text{يدرس}, \text{الطالب}, \text{اللغه}, \text{العربيه}\},$$
$$U(P) = \{\text{يتعلم}, \text{الطالب}, \text{العربيه}\}.$$

Two of the prediction's three types are shared, giving copy ratio 66.67% and novelty 33.33%. Repeating a word does not change these sets, motivating separate degeneracy checks. These sentences explain the metric and are not presented as empirical dataset examples.

Empty normalized predictions have no nonzero denominator in Eq. 2. They must be treated as invalid rather than rewarded as successful novelty. The selection filter rejects outputs with fewer than three normalized tokens before normal selection; its all-degenerate fallback is still evaluated without discarding the output. Lexical normalization does not remove all morphological variation, and the light-stem, root-stem, and character-4-gram checks are distinct measurements rather than full lemmatization.

C Generation and Prompts

The source prefix is `paraphrase: {domain}: {text}`, with no text normalization. The domain ablation omits the domain component. The original instruction for frontier baselines is:

> أعد صياغة النص التالي بأسلوب أكاديمي رصين مع الحفاظ التام على المعنى، وتغيير التركيب اللغوي والمفردات قدر الإمكان دون إضافة أي معلومات جديدة أو حذف معلومات أساسية. اكتب النص المعاد صياغته فقط، دون أي مقدمات أو شروح.

Setting	Value
Base checkpoint	`UBC-NLP/AraT5v2-base-1024`
Base fine-tuning	5 epochs; learning rate 5×10^{-5}
Per-device train/eval batch	8 / 8
Label smoothing	0.1
Weight decay	0.01
Warm-up ratio	0.1
Precision	fp16
Maximum source/target	768 / 512 model tokens
Beam decoding	5 beams; no-repeat 3-gram; repetition penalty 1.2
Sampling	12 samples; temperature 1.1; top-p 0.95
Generation limit	512 new tokens
Original LLM calls	Temperature 0.7; 1,024-token limit
GPT-4o candidate pools	Temperature 0.9; 8 samples per call
RAFT adaptation	3 epochs; learning rate 3×10^{-5}
Local hardware	One RTX 5090, 32 GB

Table 7: Recorded generation and fine-tuning settings. The primary checkpoint's selection criterion was the unsuitable English-oriented ROUGE configuration disclosed in Section 5.

Its English gloss is: "Rewrite the following text in a rigorous academic style while fully preserving meaning, changing structure and vocabulary as much as possible without adding information or deleting essential information. Output only the rewritten text, without introductions or explanations."

The calibrated prompt explicitly asks for replacement of approximately 55–60% of words, retention of 40–45% (especially technical terms and proper nouns), and output length 0.9–1.2 times the source. This natural-language word-replacement instruction approximates the intended operating point; it is not an exact specification of output-type novelty. The archived `CALIB_PROMPT` in `rebuttal_llm_control.py` is the verbatim control prompt; the constraints above describe its calibration targets.

The primary checkpoint was produced by `train_arat5_nonnormalized.py`. The separate `train_sft.py` implements the ablation/RAFT settings. These scripts must not be conflated when reproducing the model. The primary selection criterion used `RougeScorer(use_stemmer=True)`, including English-oriented tokenization, which yields an unreliable Arabic signal. Reporting-time scores are computed separately by `score_all.py`. Correct scoring does not establish that the best available training checkpoint was chosen.

D Normalization and Band Sensitivity

Table 8 gives the full normalization comparison. All systems are rescored on their existing 2,000 outputs. The original fixed band uses novelty $[40, 70]$ and preservation at least 85. Test-gold percentile bands under v1 are:

- P10–P90 novelty $[35.71, 75.77]$, P10 preservation 76.18;
- P5–P95 novelty $[30.43, 80.95]$, P5 preservation 70.80;
- P25–P75 novelty $[45.45, 66.67]$, P25 preservation 83.20.

Under v2, the first interval becomes $[33.33, 73.68]$, with preservation floor unchanged at 76.18. These are test-distribution summaries; no development-derived percentile headline is claimed. Rounded means can make a difference computed from displayed cells differ by 0.01 from an underlying unrounded difference, so the table reports the original per-system values rather than reconstructed deltas.

The primary rerank floor 88 and copy penalty 1.5 were fixed on development data. By contrast, the later OP-target novelty value and fluency threshold were obtained from test gold and actively used for candidate selection. This is stronger test-reference contact than computing a descriptive percentile band alone and is explicitly disclosed. A subsequent development-calibrated evaluation should estimate those values independently and regenerate the relevant selections under v2.

E Held-out Preservation Encoders

The held-out models in Table 9 use L2-normalized sentence embeddings and prediction-to-source cosine. Inputs are truncated to 1,200 characters and the encoder's maximum sequence length is capped at 256. The novelty interval remains the v1 test-gold range $[35.71, 75.77]$. The five encoder repositories were not pinned to revision hashes in the recorded runs, so exact historical weight versions cannot be established from their identifiers alone.

Additional coverage values for the copy penalty are 78.25%, 76.40%, and 76.05% under E5, MiniLM, and Arabic Triplet V2, respectively. Claude's values are 40.35%, 43.95%, and 41.05%.

System	Novelty v1	Novelty v2	Fixed-band coverage v1	Fixed-band coverage v2	Gold-band coverage v1	Gold-band coverage v2
Human reference	56.06	52.86	50.60	46.65	75.25	74.65
OP-max	55.70	51.87	76.65	74.25	86.15	85.95
Copy penalty	58.21	55.55	55.90	54.45	77.70	77.70
GPT-4o	65.14	62.81	49.35	52.60	74.35	74.25
Claude Sonnet 4.6	75.40	73.76	18.10	22.15	43.20	41.90
Beam + domain	25.56	21.72	9.95	5.20	18.45	16.05
Beam, no domain	25.53	21.71	9.55	4.95	–	16.00
RAFT + beam	23.10	18.95	6.40	3.30	12.20	9.60
AraT5v2, zero-shot	75.46	75.46	0.00	0.00	0.00	0.00

Table 8: Full-test metric sensitivity (all values on a 0–100 scale). The fixed band is unchanged between normalizers. Each gold-derived band instead uses that normalizer's test-gold P10/P90 novelty and P10 preservation. A dash denotes an unreported supplementary value.

Encoder	Checkpoint identifier	Gold mean	Gold P10 floor
Selection: LaBSE	`sentence-transformers/LaBSE`	–	–
Selection: GATE	`Omartificial-Intelligence-Space/GATE-AraBert-v1`	–	–
E5	`intfloat/multilingual-e5-base`	94.84	91.50
MiniLM	`sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2`	87.17	73.68
Arabic Triplet V2	`Omartificial-Intelligence-Space/Arabic-Triplet-Matryoshka-V2`	88.85	78.31

Table 9: Selection and held-out encoder identifiers. LaBSE and GATE use a joint mean preservation floor of 88. The other encoders use their individual test-gold P10 floors and do not select the evaluated predictions.

OP-max remains highest in this comparison. A planned evaluation with `Alibaba-NLP/gte-multilingual-base` could not complete because of a runtime failure and is not included; its missing result is not treated as a negative score.

F Fluency Ablation

For an output token sequence $w_1, \ldots, w_m$, the fluency score averages realized next-token log probabilities over scored non-padding positions:

$$A(P) = \frac{1}{M} \sum_{t \in \mathcal{I}} \log p(w_t \mid w_{<t}), \qquad (10)$$

where $\mathcal{I}$ contains the M scored positions after truncation. The model is used as a likelihood judge in this experiment; its separate generation-baseline result is a different use. No human feedback is incorporated into candidate selection.

The feasibility cascade is: all conditions including fluency; then omit fluency; then omit preservation; then omit length; finally maximize preservation if all candidates are degenerate. Relaxing fluency explains the remaining 15.30% below-floor outputs. The 74.05% coverage checks only novelty and preservation, not a three-way acceptability conjunction. The scoring model, test-derived threshold, 128-token truncation, and absence of a human re-audit all limit interpretation.

G Distributional Distance

Let X and Y be system and gold samples of the two-dimensional (N, F) vector, after dividing the coordinates by gold standard deviations 15.4821 and 9.1081 under v1. The same divisors are used for every system. The energy statistic is estimated as

$$\begin{aligned} \widehat{\mathcal{E}}(X,Y) = \frac{2}{mn} \sum_{i,j} \|x_i - y_j\|_2 \\ - \frac{1}{m^2} \sum_{i,i'} \|x_i - x_{i'}\|_2 \\ - \frac{1}{n^2} \sum_{j,j'} \|y_j - y_{j'}\|_2. \end{aligned} \qquad (11)$$

For each distribution, 1,200 of the 2,000 points are sampled without replacement using NumPy's default random generator with seed 0. Full pairwise-distance means are computed on these subsamples. Gold compared with an independent gold subsample yields 0.0017: a finite-sample sanity check, not a theoretical nonzero lower bound.

The copy penalty is closest to gold, despite OP-max's higher band coverage. This inversion is in-

Selector	In-band	Novelty	Preservation	Below fluency floor	Over-rewrite
OP-max	76.65	55.70	89.30	44.85	2.75
OP-max + fluency	69.40	51.44	89.83	15.85	2.60
OP-target	83.90	50.98	90.39	43.35	0.00
OP-target + fluency	74.05	47.99	90.80	15.30	0.05
Human reference	50.60	56.06	86.74	10.00	18.15

Table 10: Full-test post-hoc fluency experiment using v1 novelty and the original fixed band. Fluency is the ALLaM mean token log-likelihood; its threshold is test-gold P10 (-4.4344). All entries are scores or percentages on a 0–100 scale except the threshold itself. These outputs were not re-rated by the human evaluator.

System	Energy statistic
Gold subsample vs. gold subsample	0.0017
Copy penalty	0.0163
OP-max	0.0992
GPT-4o	0.2139
Claude Sonnet 4.6	0.9373
AraT5v2, beam	2.0380
RAFT + beam	2.3640
AraT5v2, zero-shot	13.4994

Table 11: Distributional comparison under v1 with shared gold-standard-deviation scaling. Lower values indicate greater similarity under this statistic, not superior sentence quality.

Seed	In-band	Novelty	Preservation
1	77.15	55.80	89.04
2	77.75	55.94	89.25
3	77.20	56.18	89.17
Mean	77.37	55.97	89.15
Population SD	0.27	0.16	0.09

Table 12: Three full pool regenerations on 2,000 sources: v1, original fixed band, OP-max with floor 88. SD uses ddof $= 0$; it is neither a standard error nor a confidence interval.

formative: the measurements distinguish a distribution's overall shape from its concentration inside a chosen acceptance region. These distances must not be compared numerically with values computed under another normalizer, coordinate scaling, or estimator.

H Sampling, Penalty Masks, and Diagnostics

Both PyTorch and NumPy seeds are set for each regeneration. Sampling uses 12 candidates at temperature 1.1 and top-p 0.95, no-repeat-trigram blocking, repetition penalty 1.2, plus the deterministic beam-5 output, with order-preserving deduplication. The 76.65% original result is a separate generation, not one of the three runs in Table 12. Its difference from the three-run mean is reported without establishing sampling as the sole possible cause.

The original copy-penalty script uses the token IDs of the full encoded prompt, including prefix and special tokens, and generates one sentence at a time. The source-only ablation tokenizes the source without prefix or special tokens and runs batches of 16 with padding. Thus, both the token mask and execution configuration differ. Of the 2,000 predictions, 1,704 are identical and 296 differ; the difference cannot be uniquely attributed to the mask without holding batching fixed.

The original all-token run has 55.90% v1 fixed coverage, 14.0% drift, and 7.2% over-rewriting. The source-only all-token ablation gives 55.80%, 14.2%, and 7.3%, respectively. Exempting function words yields 56.45% coverage, novelty 51.4, preservation 88.76, drift 10.3%, and over-rewriting 3.4%. The source-only all-token preservation is 86.97. These are v1 ablation scores, not replacements for the corrected prefix-inclusive primary row.

Random selection from the original AraT5v2 pool yields 40.2% fixed coverage, versus 10.0% for beam alone and 76.65% for OP-max; its band-membership oracle reaches 91.4%. This shows a contribution from both candidate generation and selection, but failure to attain the oracle does not remove shared-metric bias. The post-hoc test sweep reaches 79.3% at preservation floor 86; the reported primary setting remains the development-selected 88.

A native-speaker inspection of 64 flagged base-beam outputs labels 53 source reproductions, two trivial edits, four degenerate repetitions, three meaning/negation/deletion errors, and two simplified-register cases. Copying-related cases total 55/64 (85.9%), with mean copy ratio 98.5% and median 100%. This selected subset supports interpretation of clear copying but provides neither full-test error prevalence nor validation of all band

System	BLEU	chrF++	ROUGE-L
AraT5v2, zero-shot	0.0	2.5	4.5
GPT-4o	13.4	44.4	35.5
Claude Sonnet 4.6	5.7	37.3	25.8
Beam + domain	20.6	47.9	44.2
Beam, no domain	20.3	47.9	44.0
RAFT + beam	19.0	47.3	43.4
Copy penalty	15.2	42.0	35.7
OP-max	10.9	39.2	31.9

Table 13: Original reference-overlap measurements on 2,000 outputs, retained as historical diagnostics. They are not represented as a newly regenerated v2 overlap table.

thresholds.

I Reference-based Scores

The source-preservation scores in Table 2 compare predictions with the input. Target similarity instead compares them with the gold rewrite. Historical LaBSE/GATE target similarities are 87.0/86.9 for GPT-4o, 82.9/82.8 for Claude, 88.0/89.0 for the domain-conditioned beam model, 87.8/88.8 for RAFT, 85.1/85.3 for the copy penalty, and 84.5/86.4 for OP-max. Copy-prone models can score highly because gold and source already convey similar content. Neither target similarity nor the overlap values alone determine how well a system rewrites.

J Representative Gold-Novelty Examples

Tables 14 and 15 report actual test records near the median of the *v1 gold novelty* distribution. The examples were not selected for a favorable system ranking. Their scores retain v1, with preservation computed by Eq. 1; they must not be compared to a v2 novelty boundary without rescoring. English glosses explain the Arabic outputs and are not additional references used for evaluation.

In Example 1, gold novelty is at the 51.2nd percentile, but its preservation score is 80.1. In Example 2, gold novelty is at the 54.8th percentile, but preservation is 78.6. Both fail the fixed floor of 85. OP-max has novelty 37.5 in the first example and therefore also falls outside the fixed band; it satisfies the band in the second. These cases concretely illustrate why typical lexical change and fixed-band compliance are different properties.

The Arabic outputs also reveal limitations of lexical counts. The beam model in Example 1 copies almost exactly, while OP-max changes the opening but retains much of the structure. Claude introduces "intellectual orientations," inability to advance, and a methodological characterization, illustrating possible elaboration alongside high novelty. In Example 2, the gold target itself strengthens "cares about nothing more than" to "cares only about." Qualitative interpretation should therefore examine semantic relations directly rather than assume that any gold or in-band sentence is beyond criticism.

Role	Arabic output and English gloss	Nov.	Pres.
Source	الإرهاصات تعبر عن نوايا نعلنها جميعاً دون أن تتحول هذه النوايا إلى علم دقيق *Precursors express intentions we all declare, without these intentions becoming an exact science.*	–	–
Gold	الإرهاصات تمثل نوايا معلنة لدى الجميع، لكنها لم تتحول بعد إلى علم دقيق ومنظم. *Precursors represent intentions declared by everyone, but they have not yet become a precise, organized science.*	57.1	80.1
Beam	الإرهاصات تعبر عن نوايا نعلنها جميعاً دون أن تتحول هذه النوايا إلى علم دقيق. *Precursors express intentions we all declare, without these intentions becoming an exact science.*	0.0	98.2
OP-max	تظهر الإرهاصات على شكل نوايا نعبر عنها جميعاً، دون أن تتحول هذه النوايا إلى علم دقيق. *Precursors appear as intentions we all express, without these intentions becoming an exact science.*	37.5	92.3
Copy penalty	تُعرّف الإرهاصات بأنها نوايا نُعلنها جميعنا، لكنها لا تصبح علمًا دقيقًا. *Precursors are defined as intentions we all announce, but they do not become an exact science.*	72.7	79.4
GPT-4o	الإرهاصات تشير إلى نوايا نصرح بها جميعاً، إلا أن هذه النوايا لا تتطور إلى معرفة دقيقة. *Precursors indicate intentions we all declare, but these intentions do not develop into precise knowledge.*	60.0	86.4
Claude	تُجسّد الإرهاصات توجهاتٍ فكريةً يُفصح عنها الجميع، غير أنها تظل عاجزةً عن الارتقاء إلى مستوى المعرفة العلمية الدقيقة والمنهجية. *Precursors embody intellectual orientations expressed by everyone, yet remain unable to rise to precise, methodical scientific knowledge.*	84.2	56.7

Table 14: Example 1: test item 18561, Philosophy. Quoted-work attribution: Hassan Hanafi, *Muqaddimah fi 'Ilm al-Istighrab*, University Institution for Studies, Publishing and Distribution, Beirut, third edition, 2006, p. 48. Scores use v1. Gold and OP-max both fall outside the fixed band, for different reasons.

Role	Arabic output and English gloss	Nov.	Pres.
Source	القصة لا تهتم بشيء أكثر من اهتمامها باستكشاف الحقائق في الأمور الصغيرة المألوفة العادية. *The story cares about nothing more than exploring truths in small, familiar, ordinary matters.*	–	–
Gold	لا تهتم القصة سوى بالكشف عن الحقائق في تفاصيل الحياة اليومية البسيطة. *The story is concerned only with uncovering truths in the simple details of everyday life.*	58.3	78.6
Beam	لا تهتم القصة بشيء أكثر من اهتمامها باستكشاف الحقائق في الأمور البسيطة والمألوفة. *The story cares about nothing more than exploring truths in simple, familiar matters.*	15.4	95.4
OP-max	لا تشغل القصة اهتمامًا أكبر من البحث عن الحقائق في الأمور البسيطة المألوفة. *The story occupies no greater interest than the search for truths in simple, familiar matters.*	46.2	89.8
Copy penalty	تُركز القصة على استكشاف الحقائق المتعلقة بالأمور البسيطة والمألوفة. *The story focuses on exploring truths concerning simple and familiar matters.*	77.8	63.8
GPT-4o	تهدف القصة بشكل أساسي إلى استكشاف الحقائق الكامنة في التفاصيل الدقيقة للأمور اليومية المألوفة. *The story primarily aims to explore truths underlying the fine details of familiar everyday matters.*	71.4	58.4
Claude	تنصبُّ عناية القصة في جوهرها على استجلاء الحقائق الكامنة في تفاصيل الحياة اليومية المعتادة والمواقف الاعتيادية المألوفة. *The story's essential concern centers on elucidating truths in the details of ordinary daily life and familiar, routine situations.*	75.0	59.1

Table 15: Example 2: test item 10817, Arabic Literature. Quoted-work attribution: Mukhtar 'Ajubah, *Al-Qissah al-Hadithah fi al-Sudan*, Sudanese Studies Center, Cairo, second edition, 2000, p. 18. Scores use v1. OP-max satisfies the fixed band, while the gold reference falls below its preservation floor.

Arabic-Centric Preference Optimization for Efficient Dialect-Sensitive Machine Translation: A Seq2Seq Approach

Abdelrahman Hisham[1,2] **Malak Fathallah**[1] **Ayman Khalafallah**[1,2]
[1]Applied Innovation Center [2]Alexandria University
{a.hisham,malak.fathallah,a.khalafallah}@aic.gov.eg
{es-abdelrahmanhisham2023,es-malak.sherif2026}@alexu.edu.eg
ayman.khalafallah@alexu.edu.eg

Abstract

Arabic machine translation remains challenging because Modern Standard Arabic and dialectal varieties such as Egyptian Colloquial Arabic differ substantially in morphology, lexical choice, register, and everyday usage. We introduce a compact Arabic-centric encoder–decoder model pretrained from scratch with multilingual Seq2Seq training and contrastive cross-lingual alignment over Arabic, English, Egyptian Arabic, and 111 additional languages. We then apply Direct Preference Optimization (DPO) using preference pairs targeting Arabic-specific translation errors. On matched FLORES-200 language sets, our model outperforms the similarly sized NLLB-200-distilled-600M baseline by 3.82 average spBLEU for xx→English, 5.76 for xx→MSA, and 9.02 for xx→ECA. DPO further improves MSA→English and ECA→English by approximately +2 spBLEU over the supervised fine-tuned checkpoint, while the final model remains competitive with substantially larger general-purpose LLMs.

1 Introduction

Arabic machine translation remains challenging because Modern Standard Arabic (MSA) and dialectal varieties such as Egyptian Colloquial Arabic (ECA) differ substantially in morphology, lexical choice, register, and everyday usage. MSA is used in formal writing and institutional communication, while ECA dominates everyday communication, media, and culturally specific expression. Recent Arabic-focused MT research increasingly points in the same direction: compact specialized models can outperform much larger general systems, multi-dialect training matters under constrained resources, and standard automatic metrics often fail to capture dialect fidelity, cultural appropriateness, and variety-specific lexical choice (Hennara et al., 2025; Alabdullah et al., 2025; Sibaee et al., 2025; Nowshin et al., 2026; Magdy et al., 2026). In other words, Arabic MT benefits less from generic scale alone than from architectures, supervision, data design, and evaluation that are explicitly designed for the language's structure and sociolinguistic reality.

Machine translation in real-world deployment still depends primarily on encoder–decoder systems because they remain the most practical architecture for conditional generation at scale. In multilingual machine translation, encoder–decoder models continue to offer an effective balance of quality, efficiency, and controllability, especially for the long tail of language pairs where compute budgets and serving constraints matter (NLLB Team, 2022). Recent work reinforces this deployment logic, showing that translation systems which preserve an NMT-style decoder can retain strong quality while substantially reducing inference cost relative to decoder-heavy alternatives (Luo et al., 2025).

Yet the recent alignment wave in Natural Language Processing has only partially reached this architecture. While reinforcement learning methods such as Group Relative Policy Optimization (GRPO) have produced strong results for decoder-only LLMs in machine translation (Shao et al., 2024; Guo et al., 2025; He et al., 2025; Feng et al., 2025; Yang et al., 2025), evidence for encoder–decoder MT is still limited. The latest Seq2Seq study paints a mixed picture. Although reference-free GRPO improves many languages, Arabic remains a clear boundary case, with weak or negative gains attributed to reward-calibration limitations rather than optimization failure (Garcia-Estrada et al., 2026). This is particularly relevant for Arabic MT, where morphology and lexical ambiguity make translation quality difficult to compress into a single language-agnostic reward.

In this work, we introduce an Arabic-centric encoder–decoder machine translation model centered around Modern Standard Arabic and Egyptian Colloquial Arabic. Although the model re-

Proceedings of The Fourth Arabic Natural Language Processing Conference (ArabicNLP 2026), pages 356–372
October 28-29, 2026 ©2026 Association for Computational Linguistics

mains competitive beyond its focal directions, its design is optimized for the linguistic conditions in which generic multilingual systems are weakest: Arabic morphology, dialect-sensitive lexical choice, and everyday Egyptian usage. We apply Direct Preference Optimization (DPO) on curated preference pairs targeting linguistic phenomena that automated metrics often flatten, allowing the model to learn distinctions in adequacy and dialectal naturalness that are especially important in Arabic translation (Rafailov et al., 2023; Cui et al., 2025; Wang et al., 2025; Yang et al., 2026).

This design choice is deliberate. Rather than relying on reward-driven Reinforcement Learning (RL) with scalar quality signals that recent work has shown to be brittle for fine-grained MT ranking (Yang et al., 2026; Garcia-Estrada et al., 2026), we use preference optimization to encode the kinds of pairwise judgments that matter most for Arabic and Egyptian Arabic. The result is a deployment-oriented system that combines the serving advantages of encoder–decoder MT with Arabic-specific modeling decisions, and that, in our deployment tests, is lightweight enough to run on a single NVIDIA RTX 4000 Quadro GPU under Triton Inference Server while still delivering its strongest gains in Arabic-focused translation.

Contributions. Our contributions are as follows:

- We introduce a compact Arabic-centric encoder–decoder machine translation model focused on **MSA** and **ECA**, while maintaining competitive multilingual performance across non-Arabic language directions.
- We apply Direct Preference Optimization (DPO) using grounded preference pairs targeting Arabic-specific translation errors, including morphology, lexical ambiguity, register mismatch, and dialectal naturalness.
- We evaluate on FLORES-200, WMT24++, and NTREX (NLLB Team, 2022; Deutsch et al., 2025; Federmann et al., 2022). On matched FLORES-200 language sets, our model outperforms the similarly sized NLLB-200-distilled-600M baseline by 3.82, 5.76, and 9.02 average spBLEU for xx→English, xx→MSA, and xx→ECA, respectively.
- We demonstrate that our model is practical for real-world deployment, it can be served on a single NVIDIA RTX 4000 Quadro GPU using Triton Inference Server, showing that high-quality Arabic MT can be achieved without relying on very large decoder-only LLMs.

2 Related Work

Seq2Seq multilingual machine translation. Despite the recent success of decoder-only LLMs, production machine translation has historically been dominated by encoder–decoder architectures because they directly condition generation on a source sentence and provide a strong efficiency–quality trade-off. Early multilingual pretraining work such as mBART showed that denoising pretraining of a complete sequence-to-sequence model can improve supervised, unsupervised, and document-level MT, especially in lower-resource settings (Liu et al., 2020). NLLB-200 further scaled this direction to more than 200 languages and remains one of the strongest open encoder–decoder baselines for multilingual MT (NLLB Team, 2022). Recent work has also argued that translation systems do not need to abandon encoder–decoder structure in order to benefit from LLM-era modeling improvements, since encoder–decoder or encoder-enhanced systems can retain strong translation quality while reducing inference cost relative to decoder-heavy alternatives (Luo et al., 2025). We follow this deployment-oriented line of work: rather than scaling a decoder-only LLM, we specialize a compact encoder–decoder model for Arabic and Egyptian Arabic while preserving competitive multilingual behavior outside the focal language directions.

Arabic and dialectal Arabic machine translation. Arabic MT remains difficult because Arabic is not a single homogeneous translation target. Modern Standard Arabic is used in formal writing and institutional communication, while dialectal varieties dominate everyday use, with Egyptian Colloquial Arabic being one of the most widely used and culturally influential varieties. Recent Arabic-focused systems have revisited the value of specialization. Mutarjim, for example, demonstrates that a small Arabic–English translation model can rival much larger LLMs through targeted data curation and training (Hennara et al., 2025). Work on dialectal-to-MSA translation shows that fine-tuned and quantized models can outperform zero-shot prompting for Egyptian, Levantine, and Gulf dialects, and that joint multi-dialect training can be more effective than separate dialect-specific models under constrained resources (Alabdullah et al.,

2025). In parallel, Egyptian-focused language modeling work such as Nile-Chat highlights the importance of modeling Egyptian Arabic in both Arabic and Latin scripts (Shang et al., 2025). LoRA-tuned LLM normalization has also been shown to improve the downstream translation of Arabic vernaculars (Alam and Anastasopoulos, 2025). Our model differs from these efforts by targeting deployable encoder–decoder MT for MSA and ECA, combining compact sequence-to-sequence modeling with Arabic-focused preference optimization rather than relying only on prompting, generic multilingual transfer, or model scale.

Arabic translation resources and evaluation. Recent work has expanded Arabic MT resources beyond MSA and high-resource settings. ATHAR provides a 66K-sample Classical Arabic–English dataset (Mohammed and Khalil, 2025), while TEDxTN targets Tunisian Arabic–English speech translation with code-switching and regional variation (Bougares et al., 2025). Most relevant to dialectal MT, Alexandria introduces a large-scale human-translated benchmark covering multiple Arabic dialects, countries, and domains, making it an important resource for evaluating dialect-sensitive translation (El Mekki et al., 2026). More broadly, FLORES-200, NTREX, and WMT24++ provide multilingual and multi-domain MT evaluation coverage (NLLB Team, 2022; Federmann et al., 2022; Deutsch et al., 2025). However, dialect-sensitive Arabic MT remains difficult to evaluate with surface metrics alone, motivating our use of multiple benchmarks and preference data targeting Arabic-specific error categories (Magdy et al., 2026).

Reinforcement learning and preference optimization for MT. Sequence-level optimization for MT predates the current LLM alignment wave. Policy-gradient methods such as MIXER and minimum-risk training were introduced to reduce exposure bias and optimize sequence-level objectives more directly than token-level maximum likelihood (Ranzato et al., 2016; Shen et al., 2016). Modern RLHF later popularized Proximal Policy Optimization-based alignment with learned reward models (Schulman et al., 2017; Ouyang et al., 2022), while GRPO removed the value model and became a more memory-efficient post-training method (Shao et al., 2024; Guo et al., 2025). In MT, recent GRPO-style work has mainly targeted decoder-only LLMs: R1-T1 uses RL to elicit translation reasoning patterns (He et al., 2025), MT-R1-Zero combines rule and metric rewards for LLM-based translation (Feng et al., 2025), and MERIT applies reward-guided optimization to Chinese-centric low-resource MT using language-specific token prefixing and semantic alignment rewards (Lu et al., 2026). In the encoder–decoder setting, recent work applies GRPO to NLLB-200 with reference-free LaBSE and COMET-Kiwi rewards and shows broad gains across many languages, but also identifies Arabic as a boundary case where GRPO provides weak or unstable improvements due to limited headroom, root-pattern morphology, and reward-model calibration gaps (Garcia-Estrada et al., 2026).

Direct Preference Optimization for Arabic MT. Reference-free reward models such as LaBSE and COMET-Kiwi are attractive because they remove the need for target references during fine-tuning. However, scalar quality estimators can struggle to rank fine-grained translation differences, especially when adequacy, register, morphology, and dialectal naturalness interact. GRRM addresses this issue by replacing independent scalar scoring with group-relative reward modeling for MT, arguing that candidate groups should be compared jointly rather than scored in isolation (Yang et al., 2026). Preference optimization offers a complementary path. DPO removes the need for a separate reward model and RL sampling loop by directly optimizing pairwise preferences (Rafailov et al., 2023). MT-specific variants such as CRPO and M^2PO further show that preference optimization can be adapted to translation by improving preference-pair selection, incorporating confidence or reward signals, and using richer multi-pair/multi-perspective supervision (Cui et al., 2025; Wang et al., 2025). For encoder–decoder MT, TWA uses span-level MQM error annotations and outperforms sequence-level DPO trained on the same data (Zhang et al., 2025). Because comparable span-level Arabic annotations are unavailable, we instead use sequence-level preference pairs. Our model adopts this preference-optimization direction because our target errors are not merely generic adequacy errors: they include Arabic morphology, lexical ambiguity, MSA–ECA register mismatch, dialectal naturalness, and culturally appropriate phrasing. These are precisely the types of distinctions that are easier to express as pairwise translation preferences than as a single language-agnostic scalar reward.

3 Data Collection

3.1 Pretraining and Fine-tuning Data

We trained our model using multilingual parallel corpus designed to preserve broad translation competence while increasing coverage of Arabic and Egyptian Colloquial Arabic. For the initial training stage, we follow the contrastive multilingual training direction introduced in mRASP2 (Pan et al., 2021). This is especially suitable for our setting because Egyptian Arabic is underrepresented in public parallel corpora, and many useful transfer paths may not appear as direct Egyptian Arabic parallel directions.

We construct the pretraining and supervised fine-tuning pool from several parallel resources, mainly through public multilingual parallel data from liboaccn/nmt-parallel-corpus as the main public source of multilingual parallel data. To strengthen the low-resource dialectal component, we supplement the public data with a private Egyptian Arabic parallel corpus. This corpus was professionally translated from English, French, and Chinese into Egyptian Arabic by human annotators. The private ECA data is important because publicly available dialectal Arabic parallel corpora remain limited, and generic multilingual corpora often underrepresent the informal phrasing, lexical choices, and register shifts required for natural Egyptian Arabic translation.

The resulting training mixture is therefore designed with two goals: maintaining general multilingual translation ability across many language directions, and increasing the density and quality of Arabic/ECA supervision. This allows our model to remain competitive outside Arabic while specializing in the directions where generic multilingual systems most often struggle.

3.2 Preference Data for Direct Preference Optimization

After supervised training, we observed that the translation model was already capable in terms of adequacy, but its outputs were often overly literal, stiff, or too close to formal written Arabic. This was especially problematic for Egyptian Arabic, where natural translation often requires dialectal phrasing, idiomatic wording, and register-sensitive lexical choices. We therefore construct a DPO dataset to target these specific weaknesses rather than relying only on additional parallel fine-tuning.

The DPO data is constructed from 20-language multi-parallel records generated using Gemini 3 Flash. For each selected translation direction, the Gemini translation is used as the preferred candidate, while the corresponding output of our supervised fine-tuned (SFT) checkpoint is used as the rejected candidate; generation is conditioned on a requested domain and a fixed difficulty rubric, with Gemini providing a textual rationale for the linguistic difficulty of each sentence. The prompts are designed around failure modes observed in the SFT model's outputs, including literal translation, incorrect register, unnatural dialectal phrasing, lexical ambiguity, and domain-specific terminology. We also include examples from specialized fields such as medicine and law, where incorrect or overly generic phrase choices can significantly harm translation quality.

Each generated instance is organized as a preference example, where the preferred translation better satisfies adequacy, fluency, register, and dialectal naturalness than the rejected alternative. To improve coverage and maintain general MT behavior, we also incorporate examples derived from the FLORES-200 development set (NLLB Team, 2022). These examples are used only for preference-data construction and are kept separate from the final evaluation splits used in our reported benchmark results. The final preference dataset contains approximately 80K examples, centered around Arabic and Egyptian Arabic directions while retaining multilingual diversity.

Because Egyptian Arabic is comparatively low-resource and harder to evaluate automatically, we manually reviewed 10% of the English–Egyptian Arabic parallel pairs to confirm semantic similarity between pairs and as a sanity check of the LLM-generated outputs. The complete generation prompt, output schema, language list, and preference-pair construction procedure are provided in Appendix E.

Annotation and filtering details. The DPO preference data is filtered to remove malformed pairs, duplicated examples, and cases where the preferred and rejected translations are semantically indistinguishable. For the Egyptian Arabic portion, annotators check adequacy, fluency, register, and dialectal naturalness. In addition to manually reviewing 10% of the English–Egyptian Arabic preference pairs, we recruited a qualified annotator with formal academic training in English language and pedagogy, holding a degree from the Department of English

at the Faculty of Education, to conduct an independent preference evaluation. The annotator was presented with 100 English–Egyptian Arabic samples, each containing the English source sentence together with the chosen and rejected translations, and was asked to identify which candidate provided the better Egyptian Arabic translation of the English source. The results of this evaluation are reported in Table 1.

Annotator preference	Percentage
Chosen translation preferred	76%
Rejected translation preferred	16%
Tie / No clear preference	8%

Table 1: Human preference evaluation on 100 randomly sampled English–Egyptian Arabic translation pairs.

4 Methodology

4.1 Contrastive Multilingual Pretraining

We pretrain our model with a multilingual sequence-to-sequence objective with the contrastive learning task introduced in mRASP2 (Pan et al., 2021). The goal of this stage is not only to learn conditional translation, but also to reduce the representation gap between semantically equivalent sentences across languages. This is especially important for our setting. Our aim was to retain broad multilingual competence while becoming substantially stronger on Arabic-centered directions.

Given a parallel pair (x^i, x^j) from languages L_i and L_j, the standard MT objective maximizes the likelihood of the target sentence conditioned on the source sentence:

$$\mathcal{L}_{\text{mt}} = -\sum_{(x^i,x^j)\in\mathcal{D}} \log p_\theta(x^j \mid x^i). \quad (1)$$

Following mRASP2, we compute a sentence-level representation $\mathcal{R}(x)$ by average-pooling the encoder hidden states of sentence x. For each true parallel pair (x^i, x^j), the two sentences form a positive pair, while non-aligned sentences from the same mini-batch serve as negatives. We define:

$$s(a,b) = \frac{\text{sim}(\mathcal{R}(a), \mathcal{R}(b))}{\tau}, \quad (2)$$

where $\text{sim}(\cdot,\cdot)$ denotes cosine similarity and τ is a temperature parameter. The contrastive objective is then:

$$\mathcal{L}_{\text{ctl}} = -\sum_{(x^i,x^j)\in\mathcal{D}} \log \frac{\exp(s(x^i,x^j))}{\sum_{y\in\mathcal{B}}\exp(s(x^i,y))}, \quad (3)$$

where $\mathcal{B}$ is the set of candidate sentences in the mini-batch.

The final pretraining objective combines token-level translation supervision with sentence-level cross-lingual alignment:

$$\mathcal{L}_{\text{pretrain}} = \mathcal{L}_{\text{mt}} + \lambda|s|\mathcal{L}_{\text{ctl}}, \quad (4)$$

where λ balances the two losses and $|s|$ scales the sentence-level contrastive term to the token-level MT loss, following mRASP2.

This stage gives our model a multilingual alignment backbone before Arabic-specific preference optimization.

4.2 Direct Preference Optimization for Arabic-Centered MT

After supervised training, we apply Direct Preference Optimization to improve the translation behavior of our SFT model (Rafailov et al., 2023). We choose DPO because the remaining errors are not primarily failures of basic adequacy. The model is already capable, but its outputs can be overly literal, overly formal, or insufficiently natural for Egyptian Arabic. These errors are difficult to capture with a single scalar reward, particularly for Arabic where morphology, register, dialectal phrasing, and lexical ambiguity interact. DPO instead allows us to optimize directly from pairwise translation preferences.

Each preference example is represented as (x, y^+, y^-), where x is the source sentence with the target-language condition, y^+ is the preferred translation, and y^- is the rejected translation. For an encoder–decoder policy π_θ, the sequence log-probability is computed over the target tokens:

$$\log \pi_\theta(y \mid x) = \sum_{t=1}^{|y|} \log \pi_\theta(y_t \mid y_{<t}, x). \quad (5)$$

To keep the DPO objective compact, we define the policy preference margin:

$$\Delta_\theta = \log \pi_\theta(y^+ \mid x) - \log \pi_\theta(y^- \mid x), \quad (6)$$

and the corresponding reference margin:

$$\Delta_{\text{ref}} = \log \pi_{\text{ref}}(y^+ \mid x) - \log \pi_{\text{ref}}(y^- \mid x). \quad (7)$$

DPO optimizes the model by increasing the preferred translation margin relative to the fixed reference policy:

$$\mathcal{L}_{\text{DPO}} = -\mathbb{E}_{\mathcal{D}_{\text{pref}}}\left[\log \sigma\left(\beta(\Delta_\theta - \Delta_{\text{ref}})\right)\right], \quad (8)$$

where σ is the logistic sigmoid and β controls the strength of the preference update relative to the reference model. Intuitively, the model is rewarded when it assigns the preferred translation a larger likelihood margin than the rejected translation, without requiring a separately trained reward model or an online reinforcement learning loop.

A key implementation detail is that DPO is applied parameter-efficiently using LoRA (Hu et al., 2022) following (Garcia-Estrada et al., 2026) assumption. We update only low-rank adapter parameters while keeping the underlying encoder–decoder backbone fixed, which preserves the model's multilingual translation competence and reduces training cost. We apply LoRA to the query and value projection matrices of the attention layers.

This optimization stage is designed to shift the SFT model from merely correct translation toward preferred translation. For Arabic and Egyptian Arabic, that distinction is crucial: a translation can be semantically adequate but still sound too literal, too formal, or unnatural for the intended dialect. By training on curated preference pairs, DPO directly targets these fine-grained distinctions while retaining the efficiency and controllability of the encoder–decoder architecture.

5 Experimental Setup

5.1 Model Architecture

Our model is a compact encoder–decoder machine translation model with approximately 560M parameters. It uses an 18 layers encoder–18 layers decoder architecture rather than a decoder-only architecture, allowing generation to be explicitly conditioned on the encoded source sentence.

5.2 Language Control Tokens

We used explicit language-control tokens to indicate the desired target language during training and inference. We add special tokens for the supported target languages, including an Egyptian Arabic token (`__arz__`) to distinguish Egyptian Arabic from Modern Standard Arabic. These tokens are used consistently during pretraining, supervised finetuning, DPO, and evaluation, allowing the same Seq2Seq model to translate into Arabic, Egyptian Arabic, English, and the remaining supported languages. The complete token list is provided in Appendix B, Table 16.

5.3 Pretraining Configuration

The pretraining corpus contains approximately 2B sentence pairs and back-translated examples. More than 500M examples involve English–Arabic directions, and approximately 100M examples involve Egyptian Arabic (`__arz__`) directions. Training is bidirectional, each parallel pair is used in both source-to-target and target-to-source directions, allowing the model to learn translation into and out of Arabic, English, and Egyptian Arabic.

We follow the optimization setup of mRASP2. The model is trained with Adam, using a learning rate of 3×10^{-4}, polynomial decay scheduling, and 10K warmup steps. We use $\epsilon = 10^{-6}$, $\beta_2 = 0.98$, gradient clipping with maximum norm 5.0, and contrastive loss weight $\lambda = 1.0$.

5.4 Preference Optimization

After supervised training, we further optimize SFT model using DPO. DPO training is performed on approximately 80K preference examples for 5 epochs using a single NVIDIA H100 GPU. The preference data is centered on Arabic and Egyptian Arabic translation behavior, with examples targeting literal translation, register mismatch, dialectal naturalness, lexical ambiguity, and domain-specific terminology.

We apply DPO using LoRA adapters on the attention projection layers. The LoRA rank is set to $r = 32$, with $\alpha = 64$ and dropout 0.05. The DPO temperature parameter is set to $\beta = 0.2$, and gradients are clipped with maximum norm 1.0.

5.5 Evaluation Benchmarks

We evaluate on three multilingual MT benchmarks: FLORES-200, WMT24++, and NTREX. FLORES-200 is used to measure broad multilingual generalization, WMT24++ provides multi-domain evaluation across diverse languages and dialects, and NTREX evaluates news-domain translation quality. We report results separately for Arabic-focused directions and general multilingual directions to distinguish specialization from overall multilingual competence.

5.6 Baselines

We evaluate **NLLB-200-distilled-600M** (NLLB Team, 2022), a multilingual encoder–decoder MT model of approximately 600M parameters, as a controlled same-scale baseline for our approximately 560M-parameter model. We use the public `facebook/nllb-200-distilled-600M`

checkpoint. For xx→English, we also compare against **Qwen/Qwen3-Next-80B-A3B-Instruct** (Qwen Team, 2025), **openai/gpt-oss-120b** (OpenAI, 2025), and **Gemma4-E2B**, **Gemma4-E4B**, and **Gemma4-26B-A4B** (Google DeepMind, 2026); the same Gemma family is used for translation into MSA and ECA. Gemma was selected because it achieved the strongest spBLEU relative to model size among the general-purpose, non-MT-specialized LLMs we evaluated on FLORES-200, while its multiple scales support quality–size comparisons.

5.7 Evaluation Metrics

We use spBLEU as the automatic evaluation metric for all reported benchmark results. Scores are computed consistently across systems using the same references and preprocessing pipeline for each benchmark. We use spBLEU because it is the standard metric for FLORES-style multilingual MT evaluation and provides a consistent basis for comparing our model against larger multilingual systems. spBLEU combines BLEU (Papineni et al., 2002) with SentencePiece tokenization (Kudo and Richardson, 2018), following FLORES-style evaluation (Goyal et al., 2022) and consistent BLEU reporting practices (Post, 2018).

Because automatic metrics may not adequately capture dialectal naturalness and other fine-grained translation properties, we additionally conduct a human evaluation of English-to-Egyptian-Arabic conversations generated by the SFT and DPO models. For each English conversation, the annotator is shown the two corresponding translations under neutral labels and evaluates each translation independently according to three criteria:

- **Egyptian Naturalness** : how naturally the translation expresses the source content using Egyptian-Arabic vocabulary, phrasing, and register.
- **Semantic Similarity** : how accurately and completely the translation preserves the meaning of the original English conversation, without introducing, omitting, or altering information.
- **Grammatical Correctness** : how grammatically and structurally well formed the translation is.

Each criterion is scored independently using the six-point ordinal scale shown in Table 2.

Score	Rating	Description
5	Excellent	Fully correct and natural.
4	Good	Minor issues only.
3	Acceptable	Some noticeable issues.
2	Poor	Several important issues.
1	Very Poor	Major problems.
0	Incorrect	Incorrect or unusable.

Table 2: Human-evaluation scoring rubric used for Egyptian naturalness, semantic similarity, and grammatical correctness.

The same rubric is applied to both systems, enabling a direct comparison of the effects of DPO on dialectal naturalness, meaning preservation, and grammatical quality.

5.8 Decoding and Evaluation Protocol

All systems use the same source sentences, references, and spBLEU pipeline within each benchmark. For our model, we use deterministic 5-beam decoding, no sampling, a maximum generation length of 1022 new tokens, and a length penalty of -0.5, with the target language enforced through the decoder beginning-of-sequence token. NLLB uses the official FLORES-200 language codes and corresponding forced beginning-of-sequence token with the same beam size, length penalty, and no sampling. For LLM baselines, we use a fixed translation prompt and parse only the generated translation.

6 Results

We provide representative qualitative examples of Arabic-specific improvements after DPO in Appendix C, focusing on register, dialectal naturalness, and literal-translation errors.

6.1 Headline Results and Post-training Gains

We first summarize the main Arabic-centered findings across FLORES-200, NTREX, and WMT24++. Table 3 reports DPO gains over the supervised fine-tuned (SFT) checkpoint, Table 4 summarizes the strongest Arabic-centered comparisons, and Table 5 reports benchmark-level averages. Detailed per-language results are provided in Appendix A. Scores are spBLEU, rounded to two decimal places.

Direction	SFT	DPO	Gain
MSA→English	41.94	43.91	+1.97
ECA→English	27.75	29.80	+2.05

Table 3: DPO results on FLORES-200 Arabic-centered xx→English directions.

Benchmark	Direction	Our Model	Best baseline	Diff.
FLORES-200	MSA→English	43.91	45.62	-1.71
FLORES-200	ECA→English	29.80	30.45	-0.65
NTREX	MSA→English	**37.75**	37.43	+0.32
WMT24++	ECA→English	**24.91**	24.49	+0.42
FLORES-200	English→MSA	41.87	43.66	-1.79
FLORES-200	English→ECA	**30.02**	29.80	+0.22
WMT24++	English→ECA	**26.03**	25.72	+0.31

Table 4: Arabic and Egyptian Arabic comparisons against the strongest reported baseline in each setting, including NLLB-200-distilled-600M and the evaluated LLMs. Positive differences indicate that our model is higher.

Benchmark	Direction	Our model	NLLB-600M	Gemma4-26B
FLORES-200	xx→en	42.05	38.23	45.04
FLORES-200	xx→MSA	29.98	24.22	32.52
FLORES-200	xx→ECA	**25.28**	16.26	23.72
NTREX	xx→en	39.39	–	40.90
WMT24++	xx→en	34.85	–	37.81
WMT24++	xx→ECA	23.74	–	24.34

Table 5: Average spBLEU scores. NLLB-200-distilled-600M was evaluated on FLORES-200 only; averages use the same 29 language directions.

Observations. On matched 29-language FLORES-200 sets, our model exceeds NLLB-200-distilled-600M by 3.82 average spBLEU for xx→English, 5.76 for xx→MSA, and 9.02 for xx→ECA. The gains are +7.39 for English→MSA and +5.17 for English→ECA, while NLLB is 0.65 higher on the individual ECA→English direction. DPO additionally improves our SFT model from 41.94 to 43.91 on MSA→English and from 27.75 to 29.80 on ECA→English. Against the larger LLM baselines, our model remains competitive and achieves the strongest English→ECA result.

6.2 Human Judgment Results

To complement the automatic evaluation, we recruited a professional annotator with a degree in English from the Faculty of Education to conduct an independent human evaluation of the SFT and DPO translations. Table 6 presents the mean human ratings assigned to the SFT and DPO translations across 100 English-to-Egyptian-Arabic conversations. Scores range from 0 to 5, with higher scores indicating better quality.

Criterion	SFT	DPO	Gain
Egyptian naturalness	4.12	**4.37**	+0.25
Semantic similarity	3.72	**4.10**	+0.38
Grammatical correctness	3.17	**3.68**	+0.51

Table 6: Mean human-evaluation scores for the SFT and DPO translations across 100 conversations. Scores range from 0 to 5. Gains are calculated as DPO minus SFT.

The DPO model received higher mean ratings than the SFT model across all three evaluation criteria. The largest improvement was observed in grammatical correctness (+0.51), followed by semantic similarity (+0.38) and Egyptian naturalness (+0.25). These results indicate that DPO improved the structural correctness and meaning preservation of the translations while also producing more natural Egyptian-Arabic output.

6.3 Translation into Arabic Varieties

We next evaluate translation into Modern Standard Arabic (xx→MSA) and Egyptian Arabic (xx→ECA). This setting directly tests the main specialization target of our model. The per-language tables for this setting are reported in Appendix A, Tables 12, 13, 14, and 15.

Observations. On FLORES-200, our model outperforms NLLB-200-distilled-600M on every source direction shown in Tables 12 and 13, with average margins of +5.76 spBLEU for xx→MSA and +9.02 for xx→ECA. It also remains competitive with the larger Gemma models and obtains the strongest English-to-ECA result on WMT24++.

6.4 Post-training Ablations

6.4.1 LORA settings

In addition to the final DPO configuration, we experimented with alternative LoRA settings. Table 7 summarizes the tested configurations. Increasing the adapter capacity to rank 64 with $\alpha = 128$ caused the model to hallucinate more frequently and deviate substantially from the reference model. In contrast, reducing the adapter capacity to rank 8 with $\alpha = 16$ produced outputs that did not differ significantly from the supervised model, either qualitatively or quantitatively. Through these experiments, we found that rank 32 with $\alpha = 64$ provided the best trade-off between adaptation strength and preservation of the reference model behavior.

Run	Rank	Alpha	Dropout	Beta
Low-rank DPO	8	16	0.05	0.2
Final DPO	32	64	0.05	0.2
High-rank DPO	64	128	0.05	0.2

Table 7: LoRA configuration ablations for DPO.

6.4.2 Group Relative Policy Optimization

We also attempted GRPO as an exploratory post-training method, but it led to negative gains in our setting. We used MetricX as the reward model because the source sentence, generated hypothesis, and reference translation were all available during training. For each source sentence, we sampled 16 rollouts and optimized the model with a learning rate of 2×10^{-7}. Despite this setup, GRPO did not provide a useful learning signal for our Arabic-centered translation objective. The MetricX reward did not separate the sampled outputs strongly enough, especially when the differences between candidates involved register, dialectal naturalness, or Arabic-specific lexical choices. As a result, the model did not receive sufficiently fine-grained feedback to improve Egyptian Arabic and Modern Standard Arabic translation quality. We also found alternative automatic rewards less suitable for our goal, since they did not adequately capture the Arabic-specific phenomena targeted in this work. We therefore report GRPO as a negative result and use DPO as the final post-training method.

6.4.3 DPO data size

We also study the effect of preference-data scale on Arabic-centered post-training. As shown in Table 8, using 20K preference examples already improves over the supervised model for both Modern Standard Arabic and Egyptian Arabic. Increasing the preference dataset to 80K examples leads to larger gains, improving MSA→English from 41.94 to 43.91 spBLEU and ECA→English from 27.75 to 29.80 spBLEU. These results suggest that the DPO stage benefits from additional Arabic-centered preference coverage, especially for Egyptian Arabic, where naturalness and register-sensitive lexical choices require more diverse supervision.

DPO data size	MSA→en	ECA→en
SFT only	41.94	27.75
20K	42.18	28.32
80K	43.91	29.80

Table 8: DPO data-size ablation.

7 Limitations

Our evaluation is based on automatic spBLEU, which is useful for consistent benchmark comparison but does not fully capture dialect fidelity, register, or cultural appropriateness. This is especially important for Egyptian Arabic, where a translation can be semantically adequate but still sound unnatural or overly formal. Future work should include human evaluation by native speakers of Egyptian Arabic and Modern Standard Arabic.

8 Conclusion

We present an Arabic-centric Seq2Seq machine translation model for MSA and ECA while preserving multilingual translation competence across 111 additional languages. Beyond supervised training, we apply DPO using curated preference pairs targeting Arabic-specific errors, including literal translation, register mismatch, dialectal naturalness, lexical ambiguity, and domain-specific terminology. On matched FLORES-200 sets, our model outperforms the similarly sized NLLB-200-distilled-600M baseline by 3.82, 5.76, and 9.02 average spBLEU for xx→English, xx→MSA, and xx→ECA, respectively, while DPO adds approximately two spBLEU on both MSA→English and ECA→English. These findings show that careful data design, language control, and preference optimization can produce an efficient Arabic-focused MT system without relying only on model scale.

References

Abdullah Alabdullah, Lifeng Han, and Chenghua Lin. 2025. Advancing dialectal arabic to modern standard arabic machine translation. *arXiv preprint arXiv:2507.20301*, arXiv:2507.20301.

Md Mahfuz Ibn Alam and Antonios Anastasopoulos. 2025. Large language models as a normalizer for transliteration and dialectal translation. In *Proceedings of the 12th Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 39–67. Association for Computational Linguistics.

Fethi Bougares, Salima Mdhaffar, Haroun Elleuch, and Yannick Est'eve. 2025. TEDxTN: A three-way

speech translation corpus for code-switched tunisian arabic–english. In *Proceedings of The Third Arabic Natural Language Processing Conference*, pages 278–287, Suzhou, China. Association for Computational Linguistics.

Guofeng Cui, Pichao Wang, Yang Liu, Zemian Ke, Zhu Liu, and Vimal Bhat. 2025. Crpo: Confidence-reward driven preference optimization for machine translation. *arXiv preprint arXiv:2501.13927*, arXiv:2501.13927.

Daniel Deutsch, Eleftheria Briakou, Isaac Caswell, Mara Finkelstein, Rebecca Galor, Juraj Juraska, Geza Kovacs, Alison Lui, Ricardo Rei, Jason Riesa, Shruti Rijhwani, Parker Riley, Elizabeth Salesky, Firas Trabelsi, Stephanie Winkler, Biao Zhang, and Markus Freitag. 2025. Wmt24++: Expanding the language coverage of wmt24 to 55 languages dialects. *arXiv preprint arXiv:2502.12404*, arXiv:2502.12404.

Abdellah El Mekki, Samar M. Magdy, Houdaifa Atou, Ruwa AbuHweidi, Baraah Qawasmeh, Omer Nacar, Thikra Al-hibiri, Razan Saadie, Hamzah Alsayadi, Nadia Ghezaiel Hammouda, Alshima Alkhazimi, Aya Hamod, Al-Yas Al-Ghafri, Wesam El-Sayed, Asila Al sharji, Mohamad Ballout, Anas Belfathi, Karim Ghaddar, Serry Sibaee, and 28 others. 2026. Alexandria: A multi-domain dialectal arabic machine translation dataset for culturally inclusive and linguistically diverse llms. *arXiv preprint arXiv:2601.13099*, arXiv:2601.13099.

Christian Federmann, Tom Kocmi, and Ying Xin. 2022. NTREX-128 – news test references for MT evaluation of 128 languages. In *Proceedings of the First Workshop on Scaling Up Multilingual Evaluation*, pages 21–24, Online. Association for Computational Linguistics.

Zhaopeng Feng, Shaosheng Cao, Jiahan Ren, Jiayuan Su, Ruizhe Chen, Yan Zhang, Zhe Xu, Yao Hu, Jian Wu, and Zuozhu Liu. 2025. Mt-r1-zero: Advancing llm-based machine translation via r1-zero-like reinforcement learning. *arXiv preprint arXiv:2504.10160*, arXiv:2504.10160.

Ernesto Garcia-Estrada, Carlos Escolano, and Jos'e A. R. Fonallosa. 2026. Reference-free reinforcement learning fine-tuning for mt: A seq2seq perspective. *arXiv preprint arXiv:2605.15976*, arXiv:2605.15976.

Google DeepMind. 2026. Gemma 4 model card. Google AI for Developers.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The FLORES-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538.

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*, arXiv:2501.12948.

Minggui He, Yilun Liu, Shimin Tao, Yuanchang Luo, Hongyong Zeng, Chang Su, Li Zhang, Hongxia Ma, Daimeng Wei, Weibin Meng, Hao Yang, Boxing Chen, and Osamu Yoshie. 2025. R1-t1: Fully incentivizing translation capability in llms via reasoning learning. *arXiv preprint arXiv:2502.19735*, arXiv:2502.19735.

Khalil Hennara, Muhammad Hreden, Mohamed Motaism Hamed, Zeina Aldallal, Sara Chrouf, and Safwan AlModhayan. 2025. Mutarjim: Advancing bidirectional arabic-english translation with a small language model. *arXiv preprint arXiv:2505.17894*, arXiv:2505.17894.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *Proceedings of the International Conference on Learning Representations*.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *arXiv preprint arXiv:2001.08210*, arXiv:2001.08210.

Zhixiang Lu, Chong Zhang, Chenyu Xue, Angelos Stefanidis, Chong Li, Jionglong Su, and Zhengyong Jiang. 2026. MERIT: Multilingual expert-reward informed tuning for chinese-centric low-resource machine translation. *arXiv preprint arXiv:2604.04839*, arXiv:2604.04839.

Yingfeng Luo, Tong Zheng, Yongyu Mu, Bei Li, Qinghong Zhang, Yongqi Gao, Ziqiang Xu, Peinan Feng, Xiaoqian Liu, Tong Xiao, and Jingbo Zhu. 2025. Beyond decoder-only: Large language models can be good encoders for machine translation. *arXiv preprint arXiv:2503.06594*, arXiv:2503.06594.

Samar M. Magdy, Fakhraddin Alwajih, Abdellah El Mekki, Wesam El-Sayed, and Muhammad Abdul-Mageed. 2026. Lqm: Linguistically motivated multidimensional quality metrics for machine translation. *arXiv preprint arXiv:2604.18490*, arXiv:2604.18490.

Mohammed Sabry Mohammed and Mohammed Khalil. 2025. ATHAR: A high-quality and diverse dataset for classical arabic to english translation. In *Proceedings of The Third Arabic Natural Language Processing Conference*, pages 97–106, Suzhou, China. Association for Computational Linguistics.

James Cross Onur Çelebi Maha Elbayad Kenneth Heafield Kevin Heffernan Elahe Kalbassi Janice Lam Daniel Licht Jean Maillard Anna Sun Skyler Wang Guillaume Wenzek Al Youngblood Bapi Akula Loic Barrault Gabriel Mejia Gonzalez Prangthip Hansanti John Hoffman Semarley Jarrett Kaushik Ram Sadagopan Dirk Rowe Shannon Spruit Chau Tran Pierre Andrews Necip Fazil Ayan Shruti Bhosale Sergey Edunov Angela Fan Cynthia Gao Vedanuj Goswami Francisco Guzmán Philipp Koehn Alexandre Mourachko Christophe Ropers Safiyyah Saleem Holger Schwenk Jeff Wang NLLB Team, Marta R. Costa-jussà. 2022. No language left behind: Scaling human-centered machine translation.

Afroza Nowshin, Prithweeraj Acharjee Porag, Haziq Jeelani, and Fayeq Jeelani Syed. 2026. Context-aware dialectal arabic machine translation with interactive region and register selection. *arXiv preprint arXiv:2604.06456*, arXiv:2604.06456.

OpenAI. 2025. gpt-oss-120b & gpt-oss-20b model card. *Preprint*, arXiv:2508.10925.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll 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. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744.

Xiao Pan, Mingxuan Wang, Liwei Wu, and Lei Li. 2021. Contrastive learning for many-to-many multilingual neural machine translation. *arXiv preprint arXiv:2105.09501*, arXiv:2105.09501.

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.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*, arXiv:2305.18290.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *Proceedings of the International Conference on Learning Representations*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, arXiv:1707.06347.

Guokan Shang, Hadi Abdine, Ahmad Chamma, Amr Mohamed, Mohamed Anwar, Abdelaziz Bounhar, Omar El Herraoui, Preslav Nakov, Michalis Vazirgiannis, and Eric P. Xing. 2025. Nile-chat: Egyptian language models for arabic and latin scripts. *arXiv preprint arXiv:2507.04569*, arXiv:2507.04569.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, arXiv:2402.03300.

Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Minimum risk training for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1683–1692. Association for Computational Linguistics.

Serry Sibaee, Omer Nacar, Yasser Al-Habashi, Adel Ammar, and Wadii Boulila. 2025. Shami-mt: A syrian arabic dialect to modern standard arabic bidirectional machine translation system. *arXiv preprint arXiv:2508.02268*, arXiv:2508.02268.

Hao Wang, Linlong Xu, Heng Liu, Yangyang Liu, Xiaohu Zhao, Bo Zeng, Liangying Shao, Longyue Wang, Weihua Luo, and Kaifu Zhang. 2025. Multi-perspective multi-pair preference optimization for machine translation. *arXiv preprint arXiv:2510.13434*, arXiv:2510.13434.

Sen Yang, Shanbo Cheng, Lu Xu, Jianbing Zhang, and Shujian Huang. 2026. Grrm: Group relative reward modeling for machine translation. *arXiv preprint arXiv:2602.14028*, arXiv:2602.14028.

Wenjie Yang, Mao Zheng, Mingyang Song, and Zheng Li. 2025. Ssr-zero: Simple self-rewarding reinforcement learning for machine translation. *arXiv preprint arXiv:2505.16637*, arXiv:2505.16637.

Lily H. Zhang, Hamid Dadkhahi, Mara Finkelstein, Firas Trabelsi, Jiaming Luo, and Markus Freitag. 2025. Learning from others' mistakes: Finetuning machine translation models with span-level error annotations. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 74787–74803. PMLR.

A Detailed Per-Language Results

This appendix reports the detailed per-language tables used for the main analysis. These tables are kept outside the main results section to improve readability while preserving transparency.

Language	SFT	DPO	NLLB-600M	Gemma4-E2B	Gemma4-E4B	Gemma4-26B	Qwen-Next	GPT-OSS
Danish	49.10	48.64	48.41	51.81	52.73	53.77	53.14	52.18
Portuguese	48.46	48.06	50.09	52.21	53.18	55.02	54.28	53.94
German	47.26	47.69	44.48	46.80	47.96	49.29	49.26	47.78
Swedish	46.10	45.61	47.35	51.17	52.31	53.99	52.71	53.03
French	45.84	45.88	45.71	47.89	48.75	50.19	49.91	49.16
Romanian	44.65	44.09	45.57	44.63	46.75	48.51	47.94	47.14
Indonesian	43.40	42.59	43.29	46.16	47.39	50.38	48.63	47.67
Turkish	41.24	40.50	39.04	40.80	42.37	44.69	42.92	42.36
Slovak	41.07	40.94	39.12	39.15	42.07	45.26	44.53	42.72
Modern Standard Arabic	41.94	43.91	39.89	40.26	42.10	45.62	44.57	42.30
Egyptian Arabic	27.75	29.80	30.45	26.91	28.36	29.83	28.96	26.91

Table 9: FLORES-200 xx→English results.

Language	DPO model	Gemma4-E2B	Gemma4-E4B	Gemma4-26B
Afrikaans	54.72	51.58	54.62	58.59
Maltese	49.09	32.98	39.93	47.83
Swedish	45.55	45.44	47.05	49.10
Norwegian Bokmål	45.22	44.18	46.25	47.76
Portuguese	44.71	41.98	43.43	45.68
Indonesian	43.79	42.58	43.71	46.72
Filipino	43.17	43.28	45.01	49.38
Galician	43.14	37.38	40.14	43.44
Dutch	42.89	39.65	41.57	43.31
Arabic	37.75	30.41	32.95	37.43

Table 10: NTREX xx→en results

Language	DPO model	Gemma4-E2B	Gemma4-E4B	Gemma4-26B
Mexican Spanish	43.07	44.47	46.19	47.48
Norwegian	42.65	45.37	47.25	49.90
Swedish	42.12	43.65	45.66	47.87
Italian	41.27	41.51	43.08	45.19
Danish	41.02	42.20	43.70	45.00
Romanian	40.60	40.17	42.48	45.43
Filipino	38.87	41.51	42.32	45.27
European Portuguese	38.69	38.31	39.25	40.53
Greek	38.35	39.49	41.95	44.73
Modern Standard Arabic	35.05	31.91	33.72	36.01
Egyptian Arabic	24.91	21.80	22.64	24.49

Table 11: WMT24++ xx→en results

Source	Our model	NLLB-600M	Gemma4-E2B	Gemma4-E4B	Gemma4-26B
English	41.87	34.48	42.04	42.42	43.66
Portuguese	33.42	27.85	33.54	35.16	37.20
German	32.40	26.60	31.80	33.42	34.82
Romanian	32.08	26.68	31.65	33.57	35.34
Turkish	30.39	23.76	28.06	29.86	32.81
Czech	30.01	23.66	28.32	30.39	33.33

Table 12: FLORES-200 xx→MSA results.

Source	Our model	NLLB-600M	Gemma4-E2B	Gemma4-E4B	Gemma4-26B
English	30.02	24.85	28.46	29.40	29.80
Danish	24.92	19.06	23.83	24.89	15.70
German	25.07	18.81	16.88	17.73	25.41
Indonesian	23.77	16.59	22.32	23.39	18.03
Persian	22.99	17.56	14.58	22.49	23.44
Turkish	23.73	17.82	21.26	23.31	24.53

Table 13: FLORES-200 xx→ECA results.

Source	Our model	Gemma4-E2B	Gemma4-E4B	Gemma4-26B
English	34.99	35.48	35.53	35.32
Afrikaans	29.41	26.08	29.41	31.45
Maltese	29.20	20.70	24.53	29.60
Portuguese	28.54	28.04	29.07	29.58
Italian	28.01	27.99	28.43	29.25
Spanish	27.87	27.19	27.83	28.68

Table 14: NTREX-128 xx→MSA results.

Source	Gemma4-E2B	Gemma4-E4B	Gemma4-26B	Our model
Arabic	24.06	24.86	25.96	24.65
English	20.16	22.42	25.72	26.03
French	18.42	20.25	23.34	21.44
Danish	16.57	17.83	22.34	19.54
Romanian	18.21	20.02	23.91	21.33

Table 15: WMT24++ xx→ECA results.

B Supported Languages

We provide support for the following language-control tokens. Each token is used to indicate the desired target language during training and inference.

Token	Language	Token	Language	Token	Language
__no__	Norwegian	__nr__	South Ndebele	__nso__	Northern Sotho
__nv__	Navajo	__ny__	Chichewa	__om__	Oromo
__os__	Ossetian	__pa__	Punjabi	__pap__	Papiamento
__pl__	Polish	__ps__	Pashto	__pt__	Portuguese
__qu__	Quechua	__ro__	Romanian	__ru__	Russian
__run__	Kirundi	__rw__	Kinyarwanda	__sg__	Sango
__si__	Sinhala	__sk__	Slovak	__sl__	Slovenian
__sm__	Samoan	__sn__	Shona	__sq__	Albanian
__sr__	Serbian	__srn__	Sranan Tongo	__ss__	Swati
__st__	Southern Sotho	__su__	Sundanese	__sv__	Swedish
__sw__	Swahili	__swc__	Congo Swahili	__ta__	Tamil
__te__	Telugu	__tg__	Tajik	__th__	Thai
__ti__	Tigrinya	__tk__	Turkmen	__tl__	Tagalog
__tn__	Tswana	__to__	Tongan	__tpi__	Tok Pisin
__tr__	Turkish	__ts__	Tsonga	__tt__	Tatar
__tum__	Tumbuka	__tw__	Twi	__ty__	Tahitian
__uk__	Ukrainian	__umb__	Umbundu	__ur__	Urdu
__uz__	Uzbek	__ve__	Venda	__vi__	Vietnamese
__war__	Waray	__xh__	Xhosa	__yap__	Yapese
__yo__	Yoruba	__zh__	Chinese	__zu__	Zulu
__arz__	Egyptian Arabic	__en__	English	__ar__	Modern Standard Arabic

Table 16: Supported language-control by our model.

C Qualitative Examples

Figure 1 presents two representative qualitative examples comparing the SFT and DPO outputs.

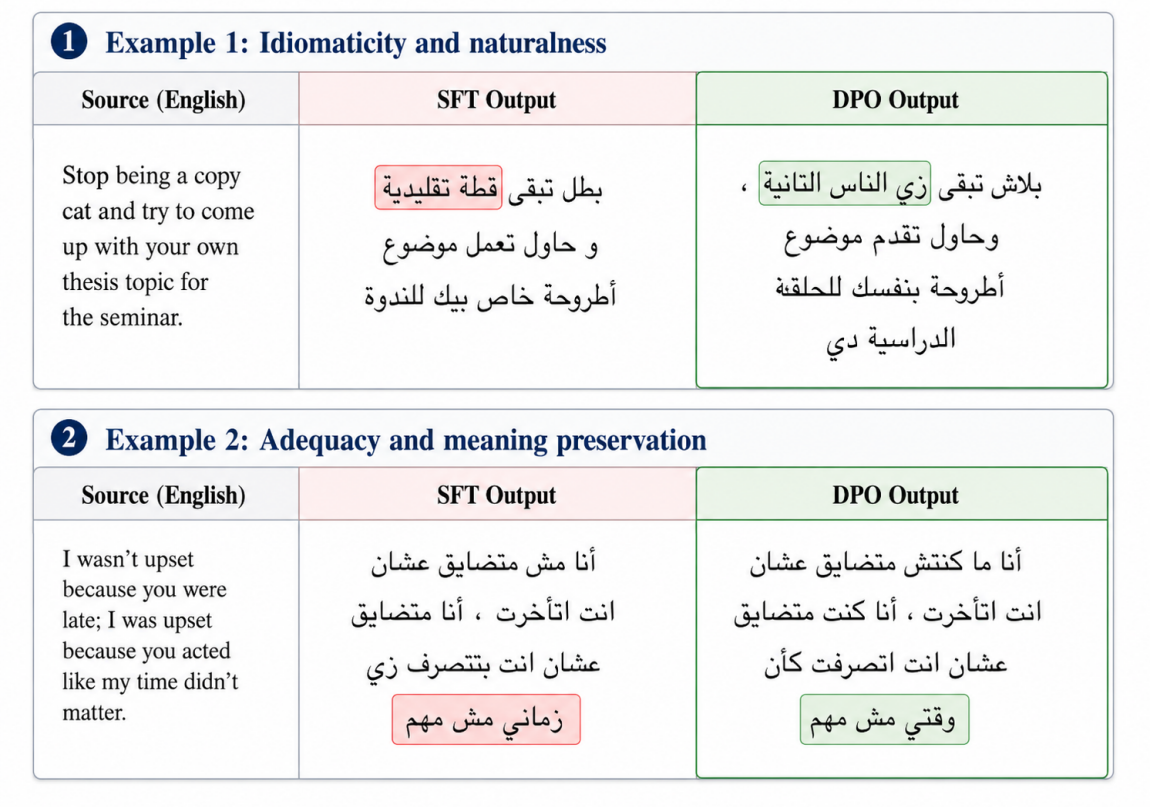


Figure 1: Qualitative comparison between SFT and DPO outputs on two English-to-Egyptian-Arabic examples. The DPO model produces more natural and semantically accurate translations.

D LLM Baseline Prompting and Decoding Protocol

This appendix describes the prompting and decoding protocol used for the LLM baselines reported in the main results. For all LLM baselines, we used the same fixed translation prompt across all evaluated language directions. The prompt template was:

```
Translate the following sentence from {source_lang} to {target_lang}: {text}
output only the translated text.
```

Here, `{source_lang}` and `{target_lang}` were replaced with the corresponding source and target language names, and `{text}` was replaced with the input sentence to be translated. The same prompt structure was used for all baseline models to avoid model-specific prompt tuning.

For generation, we used each baseline model with its recommended temperature setting. Apart from this model-specific temperature choice, the evaluation protocol was kept consistent across baselines. Each model was instructed to output only the translated sentence, and the generated translation text was used for spBLEU evaluation. No reference translations were provided to the LLM baselines during generation.

E DPO Preference-Data Generation

E.1 Motivation and Construction Overview

The preference data construction was motivated by recent work applying Group Relative Policy Optimization to encoder–decoder machine translation (Garcia-Estrada et al., 2026). That work highlights the difficulty of obtaining sufficiently discriminative scalar rewards for Arabic translation, particularly when candidate differences involve morphology, lexical choice, or other fine-grained linguistic phenomena.

Instead of relying on a scalar reward during online optimization, we construct explicit preferred–rejected translation pairs and optimize them using DPO.

The construction process consists of two stages. First, Gemini 3 Flash generates a multi-parallel record containing an English sentence and semantically aligned translations in 19 additional language varieties. Second, for each selected source–target direction, the corresponding translation generated by Gemini is used as the preferred candidate y^+, while the output produced by our supervised fine-tuned (SFT) translation checkpoint for the same source sentence and target-language condition is used as the rejected candidate y^-. Therefore, no additional LLM prompt is used to generate the rejected translation: it is obtained directly from the SFT model that DPO is intended to improve.

E.2 Languages

Each generated multi-parallel record contains the following 20 language varieties:

> English, Modern Standard Arabic, Egyptian Arabic, French, German, Spanish, Italian, Polish, Portuguese, Simplified Chinese, Korean, Japanese, Hindi, Russian, Ukrainian, Turkish, Urdu, Romanian, Vietnamese, and Dutch.

The use of multi-parallel records allows the same underlying meaning to be used across multiple source–target directions while maintaining a common semantic reference.

E.3 Domain Control

The generation script supplies a randomized domain to Gemini for each record. The generated examples cover multiple general and specialized domains, including sports, education, law, medicine, and climate-related topics.

Domain control serves two purposes. First, it prevents the preference data from being limited to generic conversational sentences. Second, it introduces domain-specific lexical and register requirements. For example, legal sentences may require distinctions involving due process, contractual remedies, procedural safeguards, or fundamental rights, whereas sports examples may involve abstract descriptions of psychological resilience, athletic performance, and collective effort.

E.4 Difficulty Control

Difficulty is controlled through a fixed natural-language rubric in the generation prompt rather than through a separate automatic metric or a human-assigned numerical score. Gemini is instructed to generate a semantically coherent but linguistically challenging sentence and to provide a textual explanation in the `difficulty_reason` field.

The difficulty rationale identifies the linguistic properties that make the example challenging to translate. Across the generated records, these properties include:

- nested subordinate, conditional, or causal structures;
- abstract or philosophical concepts;
- nuanced lexical distinctions;
- formal and domain-specific terminology;
- modality and legal relationships;
- idiomatic or non-literal phrasing;
- register differences across languages; and
- the need to restructure syntax naturally rather than translate word by word.

E.5 Generation Prompt

The following prompt template describes the instructions used to generate the multi-parallel records.

```
You are an expert multilingual translation-data generator.
Requested field: {field}
Generate exactly one original English sentence related to the requested field. The sentence must be natural and semantically coherent, but linguistically challenging to translate.
The difficulty should arise from one or more of the following: complex causal, conditional, or subordinate structures; abstract concepts; nuanced lexical distinctions; idiomatic or non-literal phrasing; domain-specific terminology; formal register; or syntactic structures that must be reorganized naturally across languages.
Do not make the sentence difficult merely by adding obscure words or unnecessary length. It should express a meaningful idea appropriate to the requested field.
Translate the sentence into all of the following language varieties:
English, Modern Standard Arabic, Egyptian Arabic, French, German, Spanish, Italian, Polish, Portuguese, Simplified Chinese, Korean, Japanese, Hindi, Russian, Ukrainian, Turkish, Urdu, Romanian, Vietnamese, and Dutch.
Requirements:
1. Preserve the complete meaning of the English sentence. 2. Produce natural translations rather than word-for-word renderings. 3. Preserve the intended register and domain terminology. 4. Keep Modern Standard Arabic and Egyptian Arabic distinct. 5. The Egyptian Arabic translation must use natural Egyptian vocabulary, grammar, and phrasing rather than simplified Modern Standard Arabic. 6. Do not omit or introduce information. 7. Explain briefly why the sentence is difficult to translate.
Return only one JSON object using the following structure:
{ "field": "{field}", "difficulty_reason": "A concise explanation of the linguistic difficulty.",
"translations": { "english": "...", "arabic": "...", "egyptian_arabic": "...", "french": "...",
"german": "...", "spanish": "...", "italian": "...", "polish": "...", "portuguese": "...",
"chinese_simplified": "...", "korean": "...", "japanese": "...", "hindi": "...", "russian": "...",
"ukrainian": "...", "turkish": "...", "urdu": "...", "romanian": "...", "vietnamese": "...",
"dutch": "..." } }
```

E.6 Preference-Pair Formation

After generation, each multi-parallel record can be converted into one or more translation directions. For a selected source language L_s and target language L_t, we construct the preference pair as follows:

1. The sentence in L_s is selected as the source x.
2. The corresponding Gemini translation in L_t is assigned as the preferred candidate y^+.
3. The source sentence is translated into L_t using the SFT checkpoint with the appropriate target-language control token.
4. The SFT output is assigned as the rejected candidate y^-.

This construction directly targets the failure modes observed in the SFT checkpoint. Depending on the translation direction, the difference between the preferred and rejected candidates may involve literal translation, lexical ambiguity, inappropriate register, unnatural Egyptian Arabic phrasing, morphology, or domain terminology. Figure 2 illustrates a representative English-to-Egyptian-Arabic preference pair. The final DPO records contain the source sentence, preferred translation, rejected translation, source-language token, target-language token, and translation direction. Malformed and duplicate records are removed during filtering. Pairs for which the preferred and rejected candidates are effectively indistinguishable are also excluded.

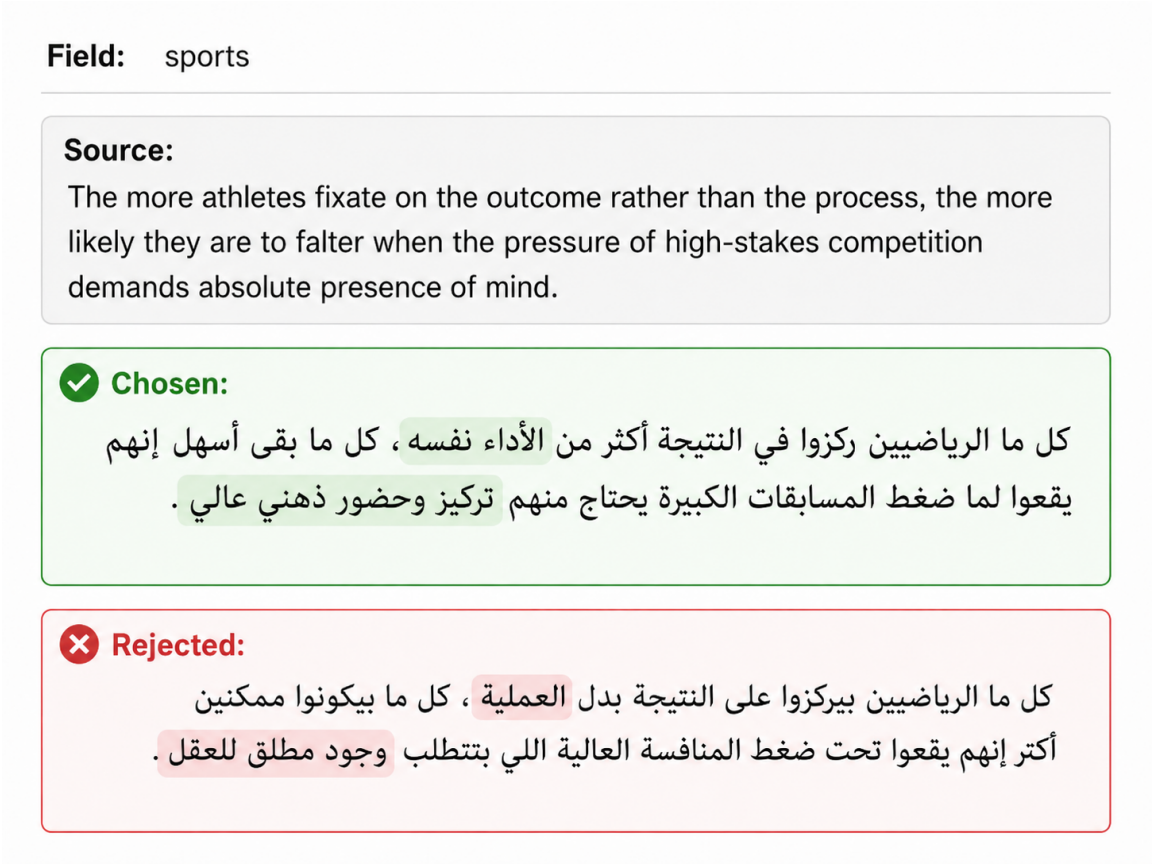


Figure 2: Representative DPO preference example from the sports domain. The highlighted spans show natural Egyptian Arabic phrasing in the chosen translation and literal or awkward phrasing in the rejected SFT output.

What WER Hides: A Closer Look at Algerian Code-Switched ASR

Othman Istaiteh[1,*], Yassine Toughrai[2,*], Kamel Smaïli[2], Salima Mdhaffar[1], David Langlois[2], Yannick Estève[1]

[1]Laboratoire Informatique d'Avignon (LIA), Avignon Université, Avignon, France
[2]Université de Lorraine, LORIA, Nancy, France
[*]*Equal contribution*
Correspondence: yassine.toughrai@loria.fr

Abstract

Automatic speech recognition for Algerian social-media speech is hard because of dialectal variation, Arabic-French code-switching, and non-standard orthography, yet these difficulties are usually summarised by a single word error rate that hides where recognition fails and whether gains reach information-bearing content. We evaluate seven systems on roughly 30 minutes of Algerian political social-media speech, annotated by language variety as Algerian dialect, Modern Standard Arabic, code-switched, or French: a Whisper-large-v3 baseline, five Whisper variants fine-tuned on different combinations of broadcast, dialectal, and code-switched speech, and the prompted Qwen3-Omni model. The best system reduces overall WER from 60.1% to 31.3%, but still reaches 44.0% on code-switched speech, and named entities remain harder than ordinary tokens, with fewer than half preserved. These results show that ASR difficulty on Algerian social-media speech cannot be read from overall WER alone; it must be measured separately by language variety and by whether named entities survive.

1 Introduction

Automatic speech recognition (ASR) for Algerian Arabic is difficult for several reasons. Algerian Arabic differs substantially from Modern Standard Arabic (MSA) in lexicon, morphology, pronunciation, and syntax (Habash, 2010; Harrat et al., 2016; Talafha et al., 2024), and as a Maghrebi variety, it is among the Arabic dialects least intelligible to systems trained on MSA. Speakers alternate frequently between Algerian Arabic and French, often within a single utterance (Amazouz et al., 2017; Lachemat et al., 2024), and the dialect has no standardised orthography, appearing in either Arabic or Latin script (Habash et al., 2012; Zaidan and Callison-Burch, 2014; Besdouri et al., 2024). When this speech comes from social media, these difficulties combine with casual, unscripted delivery, background noise, compression artefacts, and variable recording conditions.

Most ASR evaluations report only word error rate (WER), which hides differences between linguistic varieties and treats unlike errors alike. A system may transcribe MSA well while failing on Algerian dialect, recognise Arabic speech while forcing French expressions into Arabic script, or lower its overall WER while still corrupting the names of people, places, and organisations. WER counts each of these the same way, whether a word is harmlessly respelled or replaced by an unrelated one. Such failures matter most in political and misinformation-related content, where named entities determine who or what a claim concerns.

Arabic dialectal ASR has advanced largely through the MGB challenges (Ali et al., 2017, 2019) and large MSA broadcast corpora (Mubarak et al., 2021), with fine-tuned multilingual models such as Whisper now holding the best published results across dialects (Talafha et al., 2023, 2024). Progress on Algerian Arabic lags, however: zero-shot Whisper large-v2 reaches 103.4% WER on Algerian speech (Talafha et al., 2023), and the only open corpora for fine-tuning, CAFE (Lachemat et al., 2024) and the Algerian subset of Casablanca (Talafha et al., 2024), together provide barely four hours of training data, and recent shared tasks confirm it remains among the weakest-resourced Arabic varieties for ASR (Salhab et al., 2025; Elleuch et al., 2025). Code-switched ASR remains particularly underserved, since most resources transcribe foreign words in a single script and dedicated bilingual models for Arabic are scarce (Sitaram et al., 2019; Hamed et al., 2025). A parallel line of work has questioned whether WER reflects transcription quality on dialectal speech at all, given that valid spelling variants are penalised as errors (Ali et al., 2015; Besdouri et al., 2024), motivating evaluation that looks beyond a single metric.

Proceedings of The Fourth Arabic Natural Language Processing Conference (ArabicNLP 2026), pages 373–383
October 28-29, 2026 ©2026 Association for Computational Linguistics

This paper asks how difficult ASR remains on Algerian social-media speech after fine-tuning, a setting relevant to downstream tasks such as fake-news analysis. We compare a multilingual Whisper model, several Whisper fine-tuned on different speech resources, and a large prompted audio-language model, evaluating all of them on the same variety-annotated segments. The fine-tuning corpora differ in which dialects they cover and in whether French is written in Arabic or Latin script. We use these differences to show that the choice of training data changes not only how accurate a system is but the kind of errors it makes, ranging from harmless rewrites, such as normalising dialectal words to their MSA form, through transliterating or corrupting embedded French, to outright omission of difficult speech and hallucination of words with no acoustic support. Because these errors fall unevenly on the words that carry meaning, we further probe how well each system preserves named entities, the persons, organisations, and locations that determine who or what a claim concerns.

To support this study, we construct an evaluation set of Algerian political and misinformation-related social-media speech in which every segment is labelled with its language variety: Algerian dialect, Modern Standard Arabic, mixed MSA and dialect, Arabic-French code-switching, or French. These labels let us report performance per variety rather than as one overall figure, and to read the script and spelling behaviour behind it.

Our contributions are:

1. An evaluation set of Algerian political and misinformation-related social-media speech in which every segment is labelled with its language variety: dialect, MSA, mixed MSA and dialect, Arabic-French code-switching, or French.

2. A per-variety evaluation of seven systems showing that the strongest remains at 31.3% WER overall, 37.5% on dialectal and 44.0% on code-switched speech, and that systems fail in distinct ways, the weakest two by omission and by hallucination, respectively.

3. An analysis of fine-tuning data and output script showing that the gain is uneven: Arabic-script-only supervision improves Algerian speech while almost eliminating correct Latin-script output, whereas the mixed-script CAFE corpus prevents this collapse but does not recover the baseline model's monolingual-French strength.

4. A named-entity probe showing that entity transcription errors stay higher than general transcription errors, that person names are the hardest to preserve, and that fine-tuning changes which entities survive rather than uniformly improving them.

2 Data and Experimental Setup

2.1 Evaluation Set

We evaluated all systems on 335 manually transcribed audio segments extracted from Algerian political and misinformation-related social-media videos[1]. Each segment is labelled with the language variety it contains: Algerian dialect (DI), mixed MSA and dialect (MSA-DI), Modern Standard Arabic (MSA), Arabic-French code-switching (CS), or French (FR). Table 1 reports the distribution of the evaluation set.

Variety	DI	MSA-DI	MSA	CS	FR	Total
Segments	74	127	83	19	32	335
Duration (min)	5.5	12.3	4.6	4.7	2.6	29.7

Table 1: Variety distribution and duration of the evaluation set.

The evaluation set comprises approximately 30 minutes of transcribed speech drawn from 10 social-media videos collected in September 2025, divided into 335 segments. Segments were obtained automatically using the SpeechBrain CRDNN voice-activity detector[2], with energy-based refinement, merging of segments closer than 0.25 seconds, and removal of segments shorter than 0.25 seconds. The resulting segments have a mean length of 5.1 seconds, a median of 3.7 seconds,

[1]The corpus consists of audio segments extracted from publicly posted political and misinformation-related videos on TikTok and YouTube, collected for non-commercial academic research consistent with each platform's terms of service. Speakers are public figures (politicians, journalists, and commentators) speaking in their public political capacity; no private individuals or non-public content were collected. The corpus is released, not the source videos, at https://huggingface.co/datasets/oist/algerian-code-switched-asr-eval under a controlled research-use agreement modelled on that of the BOUTEF corpus (Smaili et al., 2026), restricting use to non-commercial research, prohibiting redistribution, and prohibiting attempts to identify, contact, track, or profile speakers beyond what is already apparent from the recordings.

[2]https://huggingface.co/speechbrain/vad-crdnn-libriparty

and a range from 0.4 to 30.4 seconds. Transcription and variety labelling were carried out by a native speaker of Algerian Arabic fluent in French, familiar with the code-switching phenomena and the downstream task. Dialectal Arabic was written as heard, with no normalising standard; MSA followed standard Arabic orthography; and French was kept in Latin script following standard French orthography.

2.2 Fine-tuning Corpora

Table 2 summarises the four corpora used to fine-tune Whisper-large-v3 (Radford et al., 2023). Casablanca (Talafha et al., 2024) and CAFE (Lachemat et al., 2024) both contain Algerian Arabic-French code-switching but follow different transcription conventions. Casablanca represents embedded French words predominantly in Arabic script, providing Algerian acoustic and lexical supervision without teaching the model to emit Latin-script tokens. CAFE retains French and English words in their original Latin script, demonstrating that Arabic- and Latin-script tokens may occur within the same utterance.

MLS (Pratap et al., 2020) provides monolingual French read speech, while QASR (Mubarak et al., 2021) contains large-scale broadcast Arabic dominated by MSA. Together, the four corpora let us compare fine-tuning by dialect match, code-switching, script convention, and training-data scale.

The source videos for the evaluation set were collected in September 2025, after the release of both the CAFE and Casablanca corpora, and therefore do not overlap with the fine-tuning data. CAFE and the evaluation set share broad source characteristics and collection channels but differ in topic and communicative setting. We therefore treat this evaluation as a same-source but topically shifted real-world case study rather than a strictly out-of-domain benchmark, measuring whether general Algerian and code-switched speech resources transfer to information-dense political content.

2.3 ASR Systems

Table 3 summarises the seven systems. The baseline Wv3 is Whisper-large-v3 (Radford et al., 2023), a multilingual speech recognition model pretrained on a large weakly supervised corpus[3], used here with its original parameters and without task-specific fine-tuning. We do not force an Arabic language tag, so that the model can produce French where appropriate, rather than being pushed toward a single language on multilingual segments. The five fine-tuned systems all start from this same Whisper-large-v3 checkpoint and were trained with identical hyperparameters, differing only in their training data, as detailed in the table. Qwen3-Omni (`Qwen/Qwen3-Omni-30B-A3B-Instruct`[4]) is evaluated zero-shot, prompted with the following instruction:

> *"Transcribe this audio into text. The audio can be in Algerian Arabic, Modern Standard Arabic, French, or a mix of them."*

All fine-tuned systems share an identical training configuration: AdamW with learning rate 1e-5, a Noam scheduler with 500 warmup steps, batch size 4 with gradient accumulation factor 4 (effective batch size 16), 10 epochs with no early stopping, fp16 precision, and a maximum utterance length of 40s. Speed perturbation, frequency dropout, and chunk dropout are applied as data augmentation using the SpeechBrain toolkit (Ravanelli et al., 2021). All Whisper-based systems decode with beam search, beam size 8. Qwen3-Omni is prompted zero-shot via vLLM with temperature 0.2, top-p 0.95, top-k 20, and a maximum of 8,192 generated tokens. Table 4 reports the hours sampled per system.

2.4 Text Normalization and ASR Metrics

We report word error rate (WER) and character error rate (CER), computed with the jiwer toolkit[5]. Each reported figure is pooled over all segments in the relevant group rather than averaged across per-segment rates: the overall column pools all 335 segments, and each per-variety column pools the segments carrying that variety label. WER is defined as

$$\mathrm{WER} = \frac{S + D + I}{N}, \tag{1}$$

where S, D, and I denote substitutions, deletions, and insertions, and N is the number of reference words. CER applies the same calculation at the character level.

All primary results are computed after applying the same deterministic normalisation to references

[3] https://huggingface.co/openai/whisper-large-v3

[4] https://huggingface.co/Qwen/Qwen3-Omni-30B-A3B-Instruct

[5] https://github.com/jitsi/jiwer

Corpus	Hours	Domain	Variety	Transcription convention
Casablanca (Algerian)	∼2	YouTube/television speech	Algerian, CS	French in Arabic script
CAFE small v2	∼2	Radio and podcast speech	Algerian, CS	French and English in Latin script
MLS French	1,076	Read speech	French	Latin script
QASR	2,000	Broadcast news	MSA, dialects	Standard Arabic script

Table 2: Fine-tuning corpora used in the evaluated systems. Reported sizes are full corpus sizes; the hours actually sampled in each run are given in Table 3.

System	Condition	Role in the analysis
Wv3	Whisper-large-v3, no forced language tag	multilingual baseline
Wv3-QASR	Fine-tuned on QASR	MSA-heavy broadcast
Wv3-CASA	Fine-tuned on Casablanca (Algerian)	Arabic-script Algerian
Wv3-CAFE	Fine-tuned on CAFE small v2	Mixed-script code-switched
Wv3-CC	Fine-tuned on Casablanca + CAFE	Combined Algerian, mixed-script
Wv3-ALL	Casablanca + CAFE + 2h each MLS, QASR	Algerian plus limited FR and MSA
Qwen3-Omni	Zero-shot prompted transcription	Instruction-tuned comparator

Table 3: Evaluated ASR systems and their role in the analysis.

System	Hours sampled
Wv3-QASR	2000
Wv3-CASA	∼2
Wv3-CAFE	∼2
Wv3-CC	∼4
Wv3-ALL	∼8

Table 4: Hours sampled per fine-tuning corpus for each system.

and hypotheses. The normalisation removes Arabic diacritics and tatweel, maps the hamzated alif forms to bare alif, maps hamza-on-waw and hamza-on-ya to their carrier letters, and maps alif maqsura to ya. It converts Arabic-Indic digits to Western digits, normalises curly apostrophes and backticks to a straight apostrophe, lowercases Latin text, removes unsupported punctuation and symbols, and collapses repeated whitespace.

2.5 Named-Entity Fidelity

We use named-entity recognition as a probe of whether named entities survive ASR. We apply the Arabic GLiNER model (Zaratiana et al., 2024) `NAMAA-Space/gliner_arabic-v2.1`[6] to the reference transcripts with a confidence threshold of 0.5 and retain person, location, and organization entities. These detected spans are treated as NER-derived reference entities rather than manually annotated gold entities, and the same set of reference entities is used to evaluate every ASR system.

We report four complementary measures. NE-WER is WER restricted to reference tokens covered by entity spans, which we compare with overall WER on the same entity-bearing segments. Entity-CER is the character error rate between each reference entity and its aligned form in the ASR hypothesis, taken from the word-level reference-hypothesis alignment, with the entity boundary fixed by the reference transcript. The other two measures apply the same GLiNER model to each ASR hypothesis and ask whether each reference entity is recovered: surface recall requires agreement on the normalized entity surface but not its type, while typed recall additionally requires agreement on the entity type. Because they re-run NER on noisy hypotheses, surface and typed recall reflect the NER model's robustness as well as ASR quality, and are read as downstream measures.

[6] https://huggingface.co/NAMAA-Space/gliner_arabic-v2.1

3 Evaluating ASR on Code-Switched Algerian Speech

This section evaluates the systems described in Section 2 on the variety-annotated evaluation set. Our main comparison is between four of them: the unforced Whisper v3 baseline (Wv3); Wv3-QASR, fine-tuned on the large MSA-heavy QASR corpus; Wv3-CC, fine-tuned on the Casablanca and CAFE Algerian code-switched data; and the zero-shot Qwen3-Omni. These four span the main axes of interest: a baseline, a broadcast MSA fine-tuning, a dialect-matched code-switched fine-tuning, and a prompted multilingual model. The remaining systems (Wv3-CASA, Wv3-CAFE, and Wv3-ALL) are introduced in the data ablation and the error analysis, where they isolate the contribution of each fine-tuning corpus.

Model	ALL	CS	DI	FR	MSA DI	MSA
WER (%)						
Wv3-CC	**31.3**	**44.0**	**37.5**	45.7	**23.5**	**28.6**
Qwen3-Omni	43.4	52.8	62.5	27.3	41.0	30.5
Wv3-QASR	59.1	77.5	69.9	100.0	50.9	30.5
Wv3	60.1	92.8	125.2	**25.9**	40.1	39.1
CER (%)						
Wv3-CC	**14.4**	**25.6**	**13.7**	34.0	**8.5**	**11.3**
Qwen3-Omni	20.8	34.9	26.8	19.2	17.6	13.3
Wv3-QASR	40.3	65.3	41.7	95.6	30.0	12.8
Wv3	37.0	76.2	89.7	**13.9**	15.8	21.7

Table 5: WER and CER on the evaluation set by variety, as percentages on normalised text. Bold marks the best score per column within each metric. On CS and FR, several bolded differences are not statistically distinguishable given the small segment counts (n=19, n=32); see the accompanying text and Appendix A.

3.1 Quantitative Evaluation

Table 5 reports WER and CER for the four systems (see Appendix A for details about confidence intervals). The overall ranking is led by Wv3-CC, but the per-variety columns are more informative, since each system's strengths track the data it was trained on. Wv3-CC is far ahead of Wv3-QASR on code-switched speech (44.0% vs 77.5% WER, with non-overlapping 95% bootstrap intervals of [33.2, 58.5] and [68.1, 84.9]). Its apparent advantage over Qwen3-Omni (52.8%, interval [39.8, 71.8]) and Wv3 (92.8%, interval [50.0, 168.0]) is less certain: with only 19 code-switched segments, these intervals overlap Wv3-CC's, so the true ranking among these three systems on this variety cannot be confirmed from this evaluation set (Appendix A). Its fine-tuning data is the only one to include code-switched speech with French kept in Latin script, so it alone learns to alternate scripts within a segment. On the pure-French subset, the result reverses, with Wv3 best (25.9% WER, interval [18.5, 36.4]) and Wv3-CC the weakest fine-tuned system (45.7%, interval [29.8, 67.3]); the intervals overlap, so with only 32 French segments this gap is not clearly distinguishable at the segment level (Appendix A). This follows from the same data: the Algerian fine-tuning set contains no monolingual French, only French embedded in code-switched Algerian segments, so the model never learns French as a standalone language the way Whisper's pretraining or Qwen3-Omni's multilingual training do. Wv3-QASR shows the mirror image: fine-tuned on 2,000 hours of mostly-MSA broadcast Arabic, it is competitive on MSA (30.5% WER, level with Qwen3-Omni) but does much worse on code-switched (77.5%) and pure-French (100.0%) speech, since its data is overwhelmingly monolingual MSA. Wv3-CC is strongest on dialect (37.5% WER) and on the MSA-dialect combination (23.5%), where its training data fits best.

We interpret the relationship between WER and CER by considering their ratio rather than their difference. CER is everywhere lower than WER, but the ratio varies. For the dialect-fine-tuned systems on dialectal speech, it is well under one half (CER 13.7% against WER 37.5% for Wv3-CC, 26.8% against 62.5% for Qwen3-Omni), whereas for the weaker systems, the two metrics stay closer together. A low ratio means many word errors differ from the reference by only a few characters. Because Algerian dialect has no standardised orthography, these are often valid alternative spellings of the reference rather than different words: WER counts them as full errors while CER registers only a partial difference. This is consistent with prior work showing that WER over-penalises orthographic variation in dialectal Arabic and morphologically rich languages, and that CER tracks human judgements of transcription quality more closely (Ali et al., 2015; Thennal et al., 2025). We do not treat the ratio as a general diagnostic across all conditions: on French, for example, CER stays close to WER because the errors are genuine language and script failures rather than spelling variants. We examine the nature of these dialectal spelling variants in the qualitative analysis.

3.2 Where the Gain Comes From

System	ALL	DI	CS	FR	MSA
Wv3-CASA	51.2	50.6	66.2	100.9	40.8
Wv3-CAFE	36.6	49.2	49.4	35.6	29.5
Wv3-CC	**31.3**	**37.5**	**44.0**	45.7	**28.6**
Wv3-ALL	39.4	52.3	51.1	**41.4**	31.6

Table 6: Data ablation (WER %, normalised). All systems fine-tune the same Whisper v3 base, varying only the fine-tuning corpora. Bold marks the point-estimate best per column; on CS, differences among Wv3-CC/CAFE/CASA/ALL are not statistically distinguishable given n=19 (Appendix A).

To see which fine-tuning data drives the improvement, we compare four systems built on the same Whisper v3 base (Table 6). Casablanca alone (Wv3-CASA) gives the weakest result: it improves little on dialect (50.6% WER) and, because its French is written in Arabic script, it learns to emit only Arabic and reaches 100.9% WER on pure French,

effectively unable to produce Latin-script output. CAFE alone (Wv3-CAFE), with the same amount of data, reaches 36.6% overall and keeps French usable (35.6%), since its training French is in Latin script. Combining the two (Wv3-CC) is best overall (31.3%), with each corpus covering the other's weakness: Casablanca contributes dialectal coverage and CAFE the Latin-script code-switching. Wv3-ALL adds a small amount of MLS French and QASR, kept limited so as not to push the model toward French or MSA. Its French figure (41.4%, interval [28.6, 59.1]) is not statistically distinguishable from Wv3-CC's (45.7%, interval [29.8, 67.3]), but it lowers dialect accuracy (52.3% against 37.5%) and is not better overall (39.4% against 31.3%), both clear differences with non-overlapping intervals (Appendix A); its MSA figure (31.6% against 28.6%) is likewise not distinguishable from Wv3-CC. How the proportion of each corpus in the mixture shapes this balance, and whether a different ratio could add French and MSA coverage without reducing dialect accuracy, is a question this result raises for future work.

3.3 Qualitative Error Analysis

A qualitative review of the transcriptions complements the quantitative results and explains several of the patterns above.

The first error mode is dialectal normalisation. The MSA-fine-tuned systems, such as Wv3-QASR, often map dialectal forms to their Modern Standard equivalents even when the dialectal word is recognised correctly, for example, rewriting "خمسطاش نيوم" (*khmstas nyoum*, fifteen days) as "15 يوم" and the dialectal "عندو" and "نتكلمو" as the standard "عنده" and "نتكلم". Dialect-fine-tuned systems such as Wv3-CC retain the original surface forms more often, with only minor phonetic-orthographic variation (e.g. "فظيحه" for "فضيحه", both *fadhiha*, scandal). These normalisations are the spelling variants explaining the large WER/CER gap noted earlier: WER counts them as full errors, although the content is preserved, while CER registers them as only partial differences.

The second error mode concerns code-switching between Arabic and French. An embedded French word is either kept in Latin script or transliterated into Arabic, depending on the fine-tuning data: only data containing Latin-script French within Algerian speech, namely CAFE, teaches a model to alternate scripts within a segment, and without it, a model defaults to a single script. On a segment naming the ranks "inspecteur général" and "divisionnaire général", the CAFE-fine-tuned systems (Wv3-CAFE and Wv3-CC) keep them in Latin, while the others transliterate or degrade them. Transliteration is not always harmless: a faithful one preserves the sound and leaves the word recoverable ("transfert" → ترونسفار), but in other cases the embedded word is mapped onto an unrelated Arabic word and the meaning is lost, as when "divisionnaire général" becomes "تلفزيون جنرال" (*tilfizyoun jeneral*, television general) or "au maroc" becomes "و معروف" (*wa maarouf*, and well-known). WER and CER both count the faithful and the meaning-changing version as the same error, so the difference between a recoverable and an unrecoverable word is invisible to the aggregate scores.

System	S ↓	D ↓	I ↓	H ↑
Wv3-CC	**865**	**152**	97	**2132**
Wv3-QASR	1000	1049	**55**	1365
Qwen3-Omni	1136	313	94	1858
Wv3	1181	336	622	1872

Table 7: Error-type counts: substitutions (S), deletions (D), insertions (I), and hits (H). Arrows show the preferred direction; bold marks the best per column.

Finally, catastrophic language misidentification occurs on pure French segments for the systems most conditioned on Arabic, such as Wv3-QASR, which emit hallucinated Arabic and reach WER values at or above 100%. The base Wv3 is the opposite case, transcribing pure French accurately without a forced language tag, though its overall WER stays high because of instability on code-switched and dialectal input. Taken together, the errors range from harmless MSA normalisation through meaning-changing substitution in code-switching to outright language misidentification, which is why aggregate WER and CER, sensitive as they are to orthography and formatting (Besdouri et al., 2024), do not fully reflect transcription quality on dialectal and code-switched speech.

3.4 Error Type Distribution

To characterise how the systems fail, we decompose each one's errors into substitutions, deletions, insertions, and correctly transcribed words, or hits (Table 7). The two weakest systems fail in opposite ways. Wv3-QASR is the most conservative, with

the fewest insertions (55) but by far the most deletions (1049): conditioned on MSA, it omits hard dialectal and code-switched stretches rather than guessing. Wv3 fails the opposite way, producing the most insertions (622), the signature of hallucination on dialectal and code-switched input, where it generates tokens with no acoustic support, which is also why its WER exceeds 100% on some varieties. Wv3-CC sits between these extremes, with the fewest substitutions (865) and deletions (152) and the most hits (2132), a balanced and faithful alignment with the reference. Qwen3-Omni is intermediate, with moderate insertions and the second-highest hit count. This contrast between omission and hallucination, rather than a single error rate, is what dialect-matched, code-switched fine-tuning resolves.

4 Named Entities as a Difficulty Probe

Our goal is not to evaluate named-entity recognition itself. Instead, we use named entities as a targeted probe of ASR quality. Named entities are lexically sparse, topic-dependent, and information-bearing. Errors affecting a person, organization, or location may change the target of a political claim even when the surrounding utterance remains recognizable.

As described in Section 2.5, we apply the same GLiNER model to the reference transcripts to obtain NER-derived reference entities, and evaluate every system with the four measures defined there: NE-WER, Entity-CER, surface recall, and typed recall.

4.1 Overall Entity Fidelity

System	CER	WER	Typed rec.	Surface rec.
Wv3	38.1	48.7	35.5	38.6
Wv3-QASR	38.7	61.0	27.7	29.0
Wv3-CASA	33.8	44.4	33.0	36.9
Wv3-CC	**26.3**	**35.5**	**46.6**	**48.9**
Wv3-CAFE	29.8	38.9	41.4	44.5
Wv3-ALL	30.7	40.9	41.0	43.5
Qwen3-Omni	35.0	48.3	34.6	36.2

Table 8: Named-entity fidelity on normalised transcripts, as percentages. CER and WER denote Entity-CER and NE-WER; the four measures are defined in Section 2.5.

Table 8 shows that Wv3-CC obtains the strongest result under every entity measure. It reduces entity-CER from 38.1% to 26.3% and NE-WER from 48.7% to 35.5% relative to the Wv3 baseline. Its surface recall reaches 48.9%, an improvement of 10.3 points over the baseline.

Nevertheless, even the strongest system fails to preserve more than half of the NER-derived reference entities. This residual error is operationally important for political and misinformation-related speech, where corruption of a person, organization, or location may change who or what a claim concerns.

Surface recall is only between 1.3 and 3.9 points higher than typed recall across systems. For Wv3-CC, the difference is 2.3 points. Most entity loss, therefore, arises from failure to recover the entity surface itself rather than only from assigning an incorrect entity type after transcription.

The CAFE-based systems obtain the strongest entity-level results. In contrast, Wv3-QASR reaches 61.0% NE-WER and only 29.0% surface recall, indicating that fine-tuning on a large MSA-heavy broadcast corpus does not provide robust entity preservation in the target setting.

4.2 Which Entities Are Hardest?

System	LOC	ORG	PER
Wv3	57.8	37.4	27.7
Wv3-QASR	46.6	30.6	17.6
Wv3-CASA	49.1	34.7	30.7
Wv3-CC	**60.2**	**50.3**	**41.2**
Wv3-CAFE	**60.2**	40.1	37.5
Wv3-ALL	**60.2**	43.5	33.3
Qwen3-Omni	55.3	40.8	22.1

Table 9: Surface recall (%) by reference entity type. LOC, ORG, and PER denote location, organization, and person entities, respectively. Bold indicates the highest value in each column.

Table 9 presents the Surface recall performance for the three types of entities: locations (LOC), organizations (ORG), and persons (PER). Person entities have the lowest surface recall for every evaluated system. Under the baseline, only 27.7% of person entities survive, compared with 57.8% of locations and 37.4% of organizations. Wv3-CC improves person recall to 41.2%, but still fails to preserve most person names.

Locations are consistently the easiest entity type. Wv3-CC, Wv3-CAFE, and Wv3-ALL each recover approximately 60% of location surfaces, whereas no system recovers more than half of person entities. Entity difficulty is therefore not uniform: person names remain the principal residual entity-level failure.

Entity-CER shows a consistent pattern for most systems. For example, under Wv3, entity-CER is 43.4% for persons, compared with 33.6% for both

locations and organizations. Under Qwen3-Omni, person entity-CER reaches 39.6%, compared with 28.8% for locations and 33.8% for organizations. Wv3-CC substantially narrows this type-specific gap, obtaining entity-CER values of 26.7% for locations, 25.6% for organizations, and 26.5% for persons.

4.3 Entity Difficulty Across Varieties

Variety	Entity-CER ↓	Surface recall ↑
CS	50.1	26.3
DI	23.5	41.4
FR	43.1	25.6
MSA	**18.8**	**64.0**
MSA-DI	22.7	56.4

Table 10: Entity fidelity of Wv3-CC by linguistic variety.

Table 10 shows that the variety asymmetry observed under general WER also appears in entity preservation. Wv3-CC obtains its lowest entity-CER on MSA at 18.8%, followed by MSA-DI at 22.7% and DI at 23.5%. Entity-CER increases to 43.1% on French and 50.1% on code-switched speech.

The same contrast appears under surface recall. Wv3-CC preserves 64.0% of MSA entities and 56.4% of MSA-DI entities, but only 41.4% of dialectal entities and approximately one quarter of entities in French and code-switched segments. Entity preservation, therefore, sharpens the variety-level conclusion: the strongest system remains particularly unreliable when names occur in code-switched or monolingual French speech.

4.4 Fine-tuning Changes Which Entities Survive

Overall entity scores do not show whether a system improves the same entity instances as the baseline. We therefore compare surface recovery decisions against Wv3.

Table 11 presents the number of *rescued* and *newly lost* entities by evaluated systems in comparison with the base Wv3. A reference entity is considered *rescued* when it is not preserved by Wv3 but is preserved by the evaluated system. It is considered *newly lost* when it is preserved by Wv3 but not by the evaluated system.

Wv3-CC rescues 121 entity instances missed by the baseline while newly losing 63, resulting in a net gain of 58 preserved entities. Wv3-CAFE and Wv3-ALL also produce positive net gains of 32 and 27 entities, respectively.

System	Rescued	Newly lost	Net
Wv3-QASR	38	94	−56
Wv3-CASA	65	74	−9
Wv3-CC	**121**	63	+58
Wv3-CAFE	78	**46**	+32
Wv3-ALL	79	52	+27
Qwen3-Omni	45	64	−19

Table 11: Entity surfaces rescued and newly lost relative to the Wv3 baseline.

The effect of fine-tuning is not uniformly positive. Wv3-QASR rescues only 38 entities while newly losing 94, producing a net loss of 56. Qwen3-Omni also loses more baseline-preserved entities than it rescues. Fine-tuning, therefore, changes which entities survive, rather than simply applying a uniform reduction to entity error.

This instance-level view reinforces the corpus-level findings. Fine-tuning that includes mixed-script code-switched speech (CAFE) yields a net gain in preserved entities, while Arabic-script-only fine-tuning produces a net loss: Wv3-QASR and Wv3-CASA discard more entities that the multilingual baseline already recovered than they rescue, consistent with their bias toward a single output script.

4.5 Complementarity with Overall Error

System	Pearson	Spearman	R^2	NE:overall
Wv3	0.32	0.66	0.10	1.1×
Wv3-QASR	0.63	0.62	0.39	1.1×
Wv3-CASA	0.75	0.64	0.57	1.0×
Wv3-CC	0.76	0.58	0.57	1.2×
Wv3-CAFE	0.44	0.56	0.20	1.2×
Wv3-ALL	0.59	0.55	0.35	1.2×
Qwen3-Omni	0.75	0.63	0.56	1.2×

Table 12: Relationship between general transcription error and entity error. Pearson, Spearman, and R^2 are segment-level relationships between overall CER and entity-CER. NE: overall is the ratio of NE-WER to overall WER on the same entity-bearing segments.

We now examine how closely entity error tracks general transcription error (Table 12). Entity fidelity broadly follows general recognition difficulty: overall CER and entity-CER are positively correlated for every system, with Pearson correlations between 0.32 and 0.76 and Spearman correlations between 0.55 and 0.66, and the rankings induced by overall WER and by entity-CER agree closely (Spearman 0.93).

The relationship is nevertheless incomplete.

Segment-level R^2 values range from 0.10 to 0.57, so general character accuracy explains only part of the variation in entity fidelity, and the association is particularly weak for Wv3 and Wv3-CAFE. A lower overall WER does not guarantee better entity preservation: Qwen3-Omni reduces overall WER relative to Wv3, yet its surface recall falls from 38.6% to 36.2%, and Wv3-CASA improves dialectal recognition while its entity recall stays below the baseline.

5 Conclusion

We evaluated seven systems on Algerian political social-media speech, labelled by variety, to ask how difficult ASR remains after fine-tuning. Fine-tuning on Algerian code-switched data lowers overall WER from 60.1% to 31.3%, but the gain is uneven: the best system still misses more than a third of dialectal and nearly half of code-switched words, and the choice of fine-tuning corpus decides not only accuracy but the kind of errors a model makes. Our qualitative and error-type analyses show that these range from harmless dialectal respellings, through French transliterated or corrupted into Arabic, to omission of hard speech by the MSA-fine-tuned system and hallucination by the baseline. Using named entities as a probe, we found that names stay harder than ordinary tokens, that person names survive least often, and that fine-tuning changes which entities are preserved rather than improving them uniformly. A single WER number, therefore, cannot capture how these systems perform on Algerian code-switched speech: the per-variety results and the named-entity probe each reveal failures it hides, and the qualitative analysis tells a story the numbers alone cannot.

Limitations

Our evaluation set is small: about 30 minutes from ten videos in a single political domain, so results may not transfer to other domains, speakers, or conditions, and the smaller varieties (code-switched and French) rest on few segments. Transcripts and variety labels come from a single annotator, so we report no inter-annotator agreement. Because Algerian Arabic has no standard orthography, WER and CER penalise valid spelling variants as errors, so the scores overstate how often a word is genuinely wrong; our qualitative analysis shows examples of these harmless spelling variants and faithful transliterations alongside the meaning-changing substitutions that actually lose content. The named-entity analysis uses entities detected by a single Arabic NER model rather than gold annotations, so its boundaries, types, and recall inherit that model's errors. Finally, we test only Whisper-large-v3 variants and one prompted model under fixed hyperparameters, and other architectures or data mixtures may behave differently.

Acknowledgments

This work was funded by the French Research Agency (ANR) through the TRADEF project, and it used HPC resources from GENCI-IDRIS: grants AD011014814R2, AD011012551R4 and A0181012551.

References

Ahmed Ali, Walid Magdy, Peter Bell, and Steve Renals. 2015. Multi-reference WER for evaluating ASR for languages with no orthographic rules. In *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 576–582, Scottsdale, Arizona, USA. IEEE.

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.

Djegdjiga Amazouz, Martine Adda-Decker, and Lori Lamel. 2017. Addressing code-switching in french/algerian arabic speech. In *Annual Conference of the International Speech Communication Association Interspeech 2017*, pages 62–66.

Fatma Zahra Besdouri, Inès Zribi, and Lamia Hadrich Belguith. 2024. Arabic automatic speech recognition: challenges and progress. *Speech Communication*, 163:103110.

Haroun Elleuch, Youssef Saidi, Salima Mdhaffar, Yannick Estève, and Fethi Bougares. 2025. Elyadata & lia at nadi 2025: Asr and adi subtasks. In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 762–766.

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)*,

pages 711–718, Istanbul, Turkey. European Language Resources Association (ELRA).

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

Injy Hamed, Caroline Sabty, Slim Abdennadher, Ngoc Thang Vu, Thamar Solorio, and Nizar Habash. 2025. A survey of code-switched Arabic NLP: Progress, challenges, and future directions. In *Proceedings of the 31st International Conference on Computational Linguistics (COLING)*, pages 4561–4585, Abu Dhabi, UAE. Association for Computational Linguistics.

Salima Harrat, Karima Meftouh, Mourad Abbas, Walid-Khaled Hidouci, and Kamel Smaili. 2016. An algerian dialect: Study and resources. *International journal of advanced computer science and applications (IJACSA)*, 7(3):384–396.

Houssam Eddine-Othman Lachemat, Akli Abbas, Nourredine Oukas, Yassine El Kheir, Samia Haboussi, and Absar Chowdhury Shammur. 2024. Cafe a novel code switching dataset for algerian dialect french and english. *arXiv preprint arXiv:2411.13424*.

Hamdy Mubarak, Amir Hussein, Shammur Absar Chowdhury, and Ahmed Ali. 2021. Qasr: Qcri aljazeera speech resource a large scale annotated arabic speech corpus. 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 2274–2285.

Vineel Pratap, Qiantong Xu, Anuroop Sriram, Gabriel Synnaeve, and Ronan Collobert. 2020. Mls: A large-scale multilingual dataset for speech research. *arXiv preprint arXiv:2012.03411*.

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, 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. 2021. SpeechBrain: A general-purpose speech toolkit. *Preprint*, arXiv:2106.04624. ArXiv:2106.04624.

Mahmoud Salhab, Shameed Sait, Mohammad Abusheikh, and Hasan Abusheikh. 2025. Munsit at nadi 2025 shared task 2: Pushing the boundaries of multidialectal arabic asr with weakly supervised pretraining and continual supervised fine-tuning. In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 734–739.

Sunayana Sitaram, Khyathi Raghavi Chandu, Sai Krishna Rallabandi, and Alan W Black. 2019. A survey of code-switched speech and language processing. *arXiv preprint arXiv:1904.00784*.

Kamel Smaili, Yassine Toughrai, Amina Laggoun, and David Langlois. 2026. Boutef: A multilingual corpus for fakenews in north africa – language as a weapon. *Preprint*, arXiv:2606.00193.

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, and 1 others. 2024. Casablanca: Data and models for multidialectal arabic speech recognition. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21745–21758.

Bashar Talafha, Abdul Waheed, and Muhammad Abdul-Mageed. 2023. N-shot benchmarking of Whisper on diverse Arabic speech recognition. In *Proceedings of Interspeech 2023*, pages 5092–5096.

DK Thennal, Jesin James, Deepa Padmini Gopinath, and 1 others. 2025. Advocating character error rate for multilingual asr evaluation. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4926–4935.

Omar F. Zaidan and Chris Callison-Burch. 2014. Arabic dialect identification. *Computational Linguistics*, 40(1):171–202.

Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. 2024. Gliner: Generalist model for named entity recognition using bidirectional transformer. 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 5364–5376.

A Bootstrap Confidence Intervals

Table 13 and Table 14 report the full per-variety WER and CER results with 95% bootstrap confidence intervals, computed as described in Section 2.4: segments are resampled with replacement 2,000 times within each variety group, and the pooled error rate is recomputed on each resample. These intervals support the claims made in Section 3 about which differences are, and which are not, statistically distinguishable.

System	ALL	CS	DI	FR	MSA	MSA-DI
Wv3	60.1 [45.6, 80.4]	92.8 [50.0, 168.0]	125.2 [62.4, 227.9]	25.9 [18.5, 36.4]	39.1 [31.5, 47.6]	40.1 [36.4, 43.6]
Wv3-QASR	59.1 [54.9, 63.0]	77.5 [68.1, 84.9]	69.8 [62.9, 76.4]	100.0 [100.0, 100.0]	30.5 [24.6, 36.9]	50.9 [45.7, 56.1]
Wv3-CASA	51.2 [47.3, 55.3]	66.2 [49.7, 87.2]	50.6 [44.4, 57.4]	100.9 [94.5, 105.4]	40.8 [34.5, 47.2]	40.4 [36.4, 44.2]
Wv3-CAFE	36.6 [33.7, 39.5]	49.4 [39.5, 65.7]	49.2 [43.9, 54.9]	35.6 [23.8, 52.1]	29.5 [23.6, 35.5]	30.9 [27.4, 34.4]
Wv3-CC	31.3 [28.3, 34.3]	44.0 [33.2, 58.5]	37.5 [31.3, 44.3]	45.7 [29.8, 67.3]	28.6 [23.0, 34.5]	23.5 [20.3, 26.9]
Wv3-ALL	39.4 [36.3, 42.4]	51.1 [39.4, 68.9]	52.3 [47.2, 58.0]	41.4 [28.6, 59.1]	31.6 [25.5, 38.3]	33.5 [30.1, 36.9]
Qwen3-Omni	43.4 [39.9, 46.9]	52.8 [39.8, 71.8]	62.5 [56.6, 68.4]	27.3 [17.1, 42.6]	30.5 [24.1, 37.1]	41.0 [37.3, 45.0]

Table 13: WER (%) with 95% bootstrap confidence intervals over 2,000 resamples, drawing segments with replacement and recomputing the error rate over the resampled set. CS (n=19) and FR (n=32) intervals are wide and frequently overlap across systems; differences on these two varieties should be read as suggestive rather than confirmed.

System	ALL	CS	DI	FR	MSA	MSA-DI
Wv3	36.9 [22.7, 56.0]	76.2 [30.7, 147.0]	89.7 [27.4, 187.5]	13.8 [8.8, 22.0]	21.7 [15.2, 29.3]	15.8 [13.7, 18.2]
Wv3-QASR	40.3 [35.1, 45.0]	65.3 [51.0, 74.7]	41.7 [33.6, 49.5]	95.6 [94.1, 96.7]	12.8 [9.3, 16.8]	30.0 [23.7, 36.3]
Wv3-CASA	25.3 [21.6, 29.6]	40.4 [27.3, 59.1]	18.1 [15.0, 21.7]	87.7 [81.0, 92.6]	15.8 [12.1, 20.1]	13.8 [12.0, 15.5]
Wv3-CAFE	15.6 [13.7, 17.6]	26.0 [18.9, 37.1]	19.1 [15.7, 22.7]	22.8 [12.6, 38.6]	12.9 [8.9, 17.3]	10.9 [9.3, 12.5]
Wv3-CC	14.4 [12.3, 16.4]	25.6 [18.7, 35.7]	13.7 [10.5, 17.2]	34.0 [20.5, 54.3]	11.3 [8.4, 14.6]	8.5 [7.2, 10.0]
Wv3-ALL	17.0 [15.0, 19.1]	29.7 [20.9, 43.0]	19.3 [16.2, 23.1]	27.5 [17.6, 42.5]	12.5 [9.0, 16.6]	11.9 [10.4, 13.5]
Qwen3-Omni	20.8 [18.5, 23.3]	34.9 [24.9, 50.1]	26.8 [23.1, 30.9]	19.2 [10.0, 33.5]	13.3 [9.6, 17.7]	17.6 [15.2, 20.1]

Table 14: CER (%) with 95% bootstrap confidence intervals over 2,000 resamples, drawing segments with replacement and recomputing the error rate over the resampled set. CS (n=19) and FR (n=32) intervals are wide and frequently overlap across systems; differences on these two varieties should be read as suggestive rather than confirmed.

AHA-MEMES: A Fine-Grained Multimodal Benchmark for Understanding Hate in Arabic Memes

Mohamed Bayan Kmainasi[1], **Ali Ezzat Shahroor**[1,3], **Abul Hasnat**[2], **Md. Rafiul Biswas**[3], **Wajdi Zaghouani**[4], **Firoj Alam**[1]
[1]Qatar Computing Research Institute, Qatar, [2]APAVI.AI, France
[3]Hamad Bin Khalifa University, Qatar, [4]Northwestern University in Qatar, Qatar
{mkmainasi, alsh34060, mbiswas, fialam}@hbku.edu.qa
mhasnat@gmail.com, wajdi.zaghouani@northwestern.edu

https://huggingface.co/collections/QCRI/media-integrity-intelligence

Warning: This paper contains examples that some readers may find potentially sensitive.

Abstract

Hateful memes are a growing form of multimodal online harm, where hostile intent is often conveyed through the joint interpretation of images, text, cultural references, and implicit targets. While hateful meme detection has advanced in high-resource languages, Arabic remains underexplored, with existing meme resources focusing mainly on propaganda or coarse harmful-content labels. We introduce AHA-MEMES,[1] which is, to our knowledge, the first large Arabic hateful meme benchmark with fine-grained, multi-label annotations. The dataset includes 5K manually annotated memes using a taxonomy that captures hate types, i.e., attack strategies. We further provide ∼66K silver-labeled memes to support future studies. We benchmark text-only, image-only, and late-fusion multimodal models, as well as few-shot in-context learning (ICL) and open- and closed-weight Vision-Language Models (VLMs) under zero-shot and fine-tuning settings. Our results establish strong baselines and highlight key challenges in culturally grounded Arabic hateful meme detection. We release the dataset, annotation guidelines, and evaluation scripts to support future research.[2]

1 Introduction

A smiling family photograph may appear harmless on its own, yet the same image can communicate hate when combined with a targeted caption. Likewise, a national flag paired with an apparently benign proverb may become sectarian or xenophobic depending on the communities, histories, and stereotypes that viewers bring to the meme. Memes therefore pose a distinctly multimodal and culturally grounded hate detection problem (Shahroor et al., 2026). Their meaning is often produced through the joint interpretation of image, text, and shared background knowledge, rather than by either modality in isolation (Sharma et al., 2022). This dependence on cultural context also complicates human judgment, reduces label consistency, and makes automated detection more challenging. In a cross-country study, Bui et al. (2025) found that annotators from five countries agreed on meme labels only 74% of the time, with agreement falling to 67% between U.S. and Indian annotators. Such disagreement among human annotators suggests that models trained mainly on culturally homogeneous and English-centric resources are likely to struggle when hate is expressed through culturally specific visual and linguistic cues. Figure 1 illustrates this with two examples from AHA-MEMES: a doctored portrait that mocks a woman's appearance (*Hateful*), and a television still repurposed as everyday sarcasm (*Not-Hateful*), where the hateful or benign reading emerges only from the image–text pairing.



(a) Hateful · Mocking — *Mona Lisa after eating meat and fatta.*

(b) Not-Hateful · Sarcasm — *"What's wrong with you, boy?" "Everyone talks to their sweethearts on the phone except your son, mama."*

Figure 1: Examples from the AHA-MEMES dataset across hateful and non-hateful categories.

Despite growing interest in multimodal hate detection, existing meme benchmarks remain focused in English and often rely on binary or coarse labels (Kiela et al., 2020). This limits their ability to capture *who is targeted*, *what form of hate is*

[1]AHA-Memes: **A**rabic **HA**teful **Memes**
[2]Project resources: GitHub | Hugging Face

Proceedings of The Fourth Arabic Natural Language Processing Conference (ArabicNLP 2026), pages 384–408
October 28-29, 2026 ©2026 Association for Computational Linguistics

expressed, and *how hateful meaning is constructed* through the interaction of image, text, and cultural context. Recent multilingual efforts have begun to broaden coverage, but available resources remain limited in scale and granularity. For example, Multi3Hate provides parallel multilingual annotations but only 300 samples per language (Bui et al., 2025), while GuardHarMem introduces fine-grained meme hate labels only for English (El-amrany et al., 2025). Arabic is especially underserved in this space. Although Arabic text-based hate-speech detection has advanced through shared tasks such as OSACT (Mubarak et al., 2022), and Arabic meme research has studied propaganda in ArMeme (Alam et al., 2024a; Kmainasi et al., 2025a), there is still no dedicated benchmark for fine-grained hate detection in Arabic memes.

We address this gap with AHA-MEMES, a manually annotated fine-grained multimodal dataset for Arabic hateful memes. The dataset includes hateful and non-hateful memes and provides multi-label annotations for hate types. Figure 2 gives an overview of the end-to-end construction pipeline. We conduct detailed experiments on a broad set of systems, including text encoders, image encoders, multimodal fusion models, open-weight VLMs, and closed-weight VLMs and LLMs. Our experiments show that fine-tuned open VLMs perform best for binary detection, that embedded text carries much of the signal while visual information remains complementary, and that fine-grained hate categorisation remains challenging. We also evaluate retrieval-augmented few-shot in-context learning as a training-free alternative. Our contributions are as follows.

- We introduce AHA-MEMES, the first large Arabic hateful-meme dataset, consisting of *5K manually annotated* memes with binary hate labels and fine-grained multi-label hate types. In addition, we provide silver labels[3] for targeted protected categories and features such as dialect, emotion, and sentiment.
- We develop an Arabic-contextual bilingual annotation scheme for binary hate labels, fine-grained hate types, and targeted protected categories, which can support future dataset development for Arabic hateful-meme analysis.
- We provide a comprehensive benchmark across five model families, comparing text-only, image-only, multimodal fusion, open-weight VLM, and closed-weight VLM/LLM systems under zero-shot, few-shot, and fine-tuning settings.
- We release an additional ∼66K Arabic meme corpus with silver labels for binary hate, fine-grained hate type, target category, dialect, emotion, and sentiment, enabling future work on large-scale Arabic multimodal harmful-content modelling.

[3]By silver labels, we refer to labels generated by an LLM. For this task we used Gemini-3.1-Pro.

2 Related Work

Hateful meme detection and resources. The Facebook Hateful Memes Challenge (Kiela et al., 2020) established hateful-meme detection as a multimodal task that requires joint reasoning over image and text. Its use of benign confounders showed that neither modality alone is sufficient, motivating later work on cross-modal fusion, visual-to-text prompting, and LLM-based reasoning for meme interpretation (Kumar and Nandakumar, 2022; Cao et al., 2022; Lin et al., 2024). More recent studies further show that external knowledge, cross-modal alignment, and cultural grounding are important for reliable meme understanding (Sharma et al., 2022; Ren et al., 2026; Kmainasi et al., 2025a). However, most research in this area remains English-centric, and recent evidence suggests that VLMs may inherit culturally narrow assumptions when applied to non-English memes (Wang et al., 2026).

Recent efforts have begun to address under-resourced languages, including Bengali, Tamil, Malayalam (Shahroor et al., 2026). However, these resources remain fragmented in scale, language coverage, and annotation depth (Das and Mukherjee, 2023; Ponnusamy et al., 2024; Lu et al., 2024). In addition, most resources use binary or coarse labels, which obscure ***who*** is targeted, ***what*** type of hate is expressed, and ***how*** hate is rhetorically framed.

Arabic resources. Arabic multimodal disinformation or harmful-content research has been driven mainly by text-based datasets and shared tasks. OSACT advanced offensive-language and hate-speech detection in Arabic social media (Mubarak et al., 2022), while related shared tasks and datasets have addressed propaganda detection and explanation in Arabic and multilingual settings (Hasanain et al., 2025). Recent multilingual and multimodal research on Arabic has addressed fact-checking (Alam et al., 2023) and the understanding of news and social-media content (Kmainasi et al., 2025b). Arabic multimodal social-media analysis has fo-

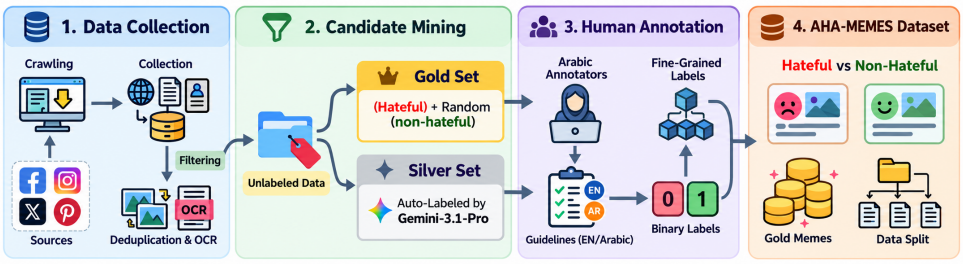


Figure 2: Overview of the AHA-MEMES construction pipeline.

cused primarily on propaganda rather than hate. The ArMeme dataset (Alam et al., 2024b) and ArAIEval shared tasks (Hasanain et al., 2024b) introduced Arabic meme propaganda-technique detection, followed by work on coarse hate labels, explanation-enhanced modelling, and explainable Arabic-English propaganda detection (Alam et al., 2024a; Kmainasi et al., 2025a; Alam et al., 2022). These resources are valuable for studying persuasion and propaganda, but they do not provide systematic fine-grained annotations for hateful memes, including hate, and fine-grained hate type, as presented in Table 1. We address this gap by developing AHA-MEMES, a dedicated fine-grained benchmark for Arabic hateful-meme detection.

In-Context learning for multimodality. In-context learning offers a training-free alternative to fine-tuning, however, its effectiveness depends strongly on which demonstrations are included in the prompt (Brown et al., 2020; Abdelali et al., 2024). Similarity-based selection is a common strategy for selecting useful examples in text classification (Liu et al., 2022). In multimodal learning, most ICL work has focused on captioning and visual question answering, often using retrieval from a single modality. It remains underexplored for multimodal *classification*, especially in culturally grounded and low-resource settings. We address this gap by selecting demonstrations through reciprocal-rank fusion of text and image similarities (Cormack et al., 2009).

3 Dataset

3.1 Data Curation

We sourced memes from Facebook, Instagram, Pinterest, and Twitter (X). For Facebook, Instagram, and Pinterest, we manually selected public groups and pages focused on public figures, politics, and social commentary, contexts in which hateful memes are likely to circulate. We then collected images from these public sources using a semi-manual procedure following prior work (Alam et al., 2024b). For Twitter (X), we collected memes through keyword-based search. In Figure 2, we provide the data curation pipeline. The final 5K set consists of 3,026 Instagram (60.5%), 1,647 Facebook (32.9%), 233 Pinterest (4.7%), and 94 Twitter/X (1.9%) memes. They were posted between April 2016 and March 2025.

3.2 Filtering and OCR Text

Duplicate removal. Social-media collections are highly redundant as users often repost identical or slightly modified content. We therefore removed exact and near-duplicate images after initial data curation. For near-duplicate filtering, we extracted image embeddings using a model fine-tuned on social-media images (Alam et al., 2020) following the approach discussed in (Alam et al., 2024b). We then computed pairwise Euclidean distances between image embeddings across the full dataset. Image pairs with a distance of 3.6 or lower were treated as near-duplicates, and only one image from each duplicate cluster was retained.

OCR rext. We extracted embedded text using EasyOCR,[4] which prior work uses for memes text extraction (Kirk et al., 2021; Cao et al., 2023; Alam et al., 2024b) which we selected for its broad script coverage and scene-text detection capabilities (Liao et al., 2023). As our study focuses on multimodal image-text analysis, we discarded memes with no detectable embedded text. As a result, every retained meme contains both visual and textual modalities.

3.3 Image Selection for Annotation.

Harmful, hateful, and misleading content represents a relatively small proportion of online content, however, its societal impact can be substantial, as reported in prior studies (Sharma et al., 2022). To increase the proportion of relevant examples while preserving coverage of non-hateful memes, we used a weak pre-selection strategy before man-

[4] https://github.com/JaidedAI/EasyOCR

Dataset	Lang.	Size	Multi-label	Task granularity	Source
ArMeme (Alam et al., 2024b)	AR	5,725	No	Propaganda status	Social media
ArAIEval (Hasanain et al., 2024b)†	AR	3,062	No	Binary propaganda	ArMeme
Propaganda-to-Hate (Alam et al., 2024a)†	AR	∼3.1K	Yes	Hate + hate types	ArAIEval
MAHED Task 3 (Zaghouani et al., 2025)†	AR	3,562	No	Binary hate	Social media
MemeXplain (Kmainasi et al., 2025a)†	AR/EN	5,725 AR + 10K EN	No	Label + explanations	ArMeme + Hateful Memes
AraHarMeme (El-Amrany et al., 2026)	AR	5,313	No	Three-way harmfulness	Social media
ArPoMeme (Zaghouani et al., 2026)	AR	∼7.3K	Yes	Ideology and polarization	Political memes
AHA-MEMES (ours)	**AR**	**5K + 66K‡**	**Yes**	**Hate + 8 hate / 3 non-hate types**	**four platforms**

Table 1: Comparison of Arabic multimodal meme datasets. †The resource contains data derived from or overlapping with an earlier Arabic meme dataset. ‡AHA-Memes contains 5K human-annotated gold memes and ∼66K auxiliary silver-labeled memes.

ual annotation. Specifically, we applied Gemma-3-12B to assign binary hateful labels to 71K memes.

We then selected 5K memes from this pre-screened pool for manual annotation, including ∼2K memes pseudo-labeled as hateful and ∼3K sampled uniformly at random from the non-hateful pseudo-labeled subset. After annotation, the resulting dataset contained 1,850 hateful and 3,150 non-hateful memes, providing broad coverage of both classes.

3.4 Annotation

Manual annotation tasks. We manually annotate each meme for binary hatefulness and, fine-grained hierarchical labels. The annotation scheme is organized as follows.

- `is_hateful` is a binary judgment. Following established definitions (Kiela et al., 2020), a meme is labeled as *Hateful* if it directly or indirectly attacks people on the basis of a protected characteristic, including ethnicity, race, nationality, immigration status, religion, caste, sex, gender identity, sexual orientation, disability, or disease. We make two boundary cases explicit. *First*, attacks against groups that themselves promote or perpetrate hate, such as terrorist organizations, are not treated as hate speech. *Second*, content that is offensive, rude, or insulting but does not target a protected category is labeled *Not-Hateful*. This separates hatefulness from generic offensiveness.
- For memes labeled *Hateful*, and `hateful_type` captures the attack strategy as a multi-label annotation over *Mocking*, *Incitement*, *Dehumanization*, *Slurs*, *Contempt*, *Inferiority*, and *Exclusion*. A meme may receive multiple hate-type labels (e.g., *Slurs* and *Dehumanization*).
- For memes labeled *Not-Hateful*, and `non_hateful_type` assigns one or more labels from *Humor*, *Sarcasm*, and *Other*.

This hierarchical structure supports binary hate detection at the top level, enables fine-grained categorization for hateful content, and ensures that fine-grained labels are consistent with the binary decision by construction. We provide bilingual (Arabic and English) annotation guidelines, including category definitions and examples, in Appendix B.

Annotation team. Annotation was conducted by a third-party company. Three trained native Arabic-speaking annotators annotated the dataset using bilingual guidelines (See Appendix B) in English and Arabic, which we developed to capture dialectal and culturally specific cues (Hasanain et al., 2024a). Before annotation, annotators were informed that the dataset may contain offensive, hateful, or otherwise harmful content. In addition, they were informed about the use of the data. We also conducted multiple rounds of training and guideline refinement to improve annotation consistency and quality. For both the binary and fine-grained tasks, final labels were determined by majority agreement among the annotators. In total, AHA-MEMES contains 5K manually annotated memes, with an annotation cost of $\sim \$4K$.

Annotator agreement. We measured agreement with Cohen's κ inter-annotator agreement on the annotated dataset. For each multi-label task, every subtype was treated as a separate binary label. Cohen's κ was computed independently for each subtype and each of the three annotator pairs, then macro-averaged across subtypes and annotator pairs. Agreement is 0.91, 0.75, and 0.67 for `is_hateful`, `hateful_type`, and `non_hateful_type`, respectively, indicating substantial to near-perfect agreement for all annotation tasks (Landis and Koch, 1977).

3.5 Statistics and Data Splits

Distribution. AHA-MEMES is moderately imbalanced, with 1,850 memes (37.0%) labeled as *Hateful*. The fine-grained label distribution is long-tailed (Table 2). On the non-hateful side, *Sarcasm*

(1,393) and *Humor* (1,331) are the most frequent labels. Among hateful memes, *Mocking* (1,007) is the most common attack strategy, whereas *Exclusion* (17) is the rarest. Since fine-grained labels are multi-label, the total label count exceeds the number of memes. The proportion of multi-label memes also increases from 9.9% in the train split to 21.4% in the test split, reflecting the richer co-occurring strategies captured in the triple-annotated gold subset. Per-split label cardinality is reported in Appendix C.3.

Splits. We partition AHA-MEMES into train, development, and test splits containing 3,500 (70%), 500 (10%), and 1,000 (20%) memes, respectively. The splits are stratified by the binary hate label, and we ensure that no meme appears in more than one split. The label distribution across the train, development, and test sets is reported in Table 2.

Hate	Fine-grained	Train	Dev	Test	Total
Hateful	Mocking	706	90	211	1,007
Hateful	Incitement	320	51	85	456
Hateful	Dehumanization	247	42	58	347
Hateful	Slurs	252	42	47	341
Hateful	Contempt	107	18	50	175
Hateful	Inferiority	57	14	32	103
Hateful	Exclusion	10	4	3	17
Hateful	Other (H)	18	2	7	27
Not Hateful	Other (NH)	380	50	96	526
Not Hateful	Sarcasm	934	126	333	1,393
Not Hateful	Humor	863	136	332	1,331

Table 2: Fine-grained sub-type distribution across splits. Counts sum to more than the number of memes because the label is multi-label. *Humor*, *Sarcasm*, and *Other* are non-hateful subtypes; the rest are hateful attack types. H: Hateful; NH: Not-Hateful.

3.6 Auxiliary Silver Dataset

In addition to the 5K human-verified memes that constitute the core AHA-MEMES, we release an auxiliary dataset of ∼66K Arabic memes with LLM-generated annotations from Gemini 3.1 Pro.[5] We refer to this resource as a *silver* dataset (Wang et al., 2024; Du et al., 2023) as its labels are not human-verified. The dataset is intended to support weakly supervised learning, semi-supervised learning, active learning, retrieval-based analysis, and future annotation expansion.

The motivation for releasing this auxiliary set is twofold. *First*, even without labels, a large in-domain collection of Arabic memes is valuable for the community. It captures meme templates, OCR noise, dialectal and code-mixed text, culturally specific references, visual stereotypes, and rapidly evolving political and social topics. Such data can support meme-understanding tasks that require joint reasoning over image content, embedded text, social context, and target entities (Pramanick et al., 2021). *Second*, the silver labels provide a scalable starting point for methods that explicitly model noisy supervision. Weak-supervision frameworks have shown that noisy heuristic labels can be useful when denoised or combined with small amounts of gold data (Ratner et al., 2016, 2017). Similarly, semi-supervised and self-training methods use pseudo-labels on large unlabeled collections to improve model learning, especially when combined with confidence filtering and consistency-based training (Sohn et al., 2020). Recent work on LLM-assisted annotation further suggests that LLMs can reduce annotation cost and help bootstrap labeled resources, while also emphasizing the importance of distinguishing model-generated annotations from human-verified labels (Tan et al., 2024; He et al., 2024).

We therefore position the ∼66K set as an auxiliary community resource rather than as a core contribution. All primary results in this paper are reported on the human-verified 5K set. Silver annotations are produced with Gemini 3.1 Pro using two prompts: for the unlabelled ∼66K memes the model predicts the label and subtype together with rich metadata (topic, dialect, sentiment, target categories, propaganda techniques, etc.), whereas for the 5K gold memes it generates the same metadata *conditioned* on the human label, so it never relabels them. The silver labels indicate that hateful content is relatively rare (0.13 hateful and 0.87 non-hateful) while the collection remains dialectally and topically diverse. Full distributions and the prompts are provided in Appendix E.

3.7 Models

We evaluate five model groups spanning unimodal encoders, multimodal fusion models, and vision-language models (VLMs).

Unimodal baselines. Text-only models operate on OCR text and include AraBERTv2 (Antoun et al., 2020), MARBERTv2 (Abdul-Mageed et al., 2021), CAMeLBERT-mix (Inoue et al., 2021), XLM-R-base (Conneau et al., 2020), and mBERT (Devlin et al., 2019). Image-only models operate on the

[5] https://deepmind.google/models/model-cards/gemini-3-1-pro/

meme image and include ViT-B/16 (Dosovitskiy et al., 2021), BEiT-B/16 (Bao et al., 2022), Swin-B (Liu et al., 2021), ConvNeXt-V2-tiny (Woo et al., 2023), and DINOv2-B (Oquab et al., 2024). Each encoder is followed by a linear classification head.

Late fusion. We combine the two strongest text and vision encoders by concatenating their pooled representations. The resulting vector is passed to a two-layer MLP with GELU activation, a hidden size of 512, and dropout of 0.1.

VLMs. We benchmark open- and closed-weight VLMs, including `Qwen3-VL-8B-Instruct` (Qwen3-VL-8B) (Bai et al., 2025), Fanar-2-Oryx-IVU (FANAR TEAM et al., 2026), Phi-3.5-Vision (Abdin et al., 2024), SmolVLM-Instruct (Marafioti et al., 2025), Qwen3-VL-8B-Thinking, InternVL3.5-8B (Wang et al., 2025), Gemini-2.5-Pro (Comanici et al., 2025), Gemini-3.5-Flash (Comanici et al., 2025), and GPT-5 (Singh et al., 2026). Each model receives the meme image together with its Arabic OCR text. Unless otherwise stated, all VLM models are evaluated in a zero-shot setup.

Fine-tuning and few-shot evaluation. We use Qwen3-VL-8B to compare supervised fine-tuning with retrieval-based few-shot prompting under a controlled setting, rather than exhaustively evaluating prompting strategies across all VLMs.

3.8 Few-Shot Selection

For each test meme, we retrieve demonstrations from the training set using a CLIP-style dual encoder. Our primary retriever is jina-clip-v2 (Koukounas et al., 2024), which embeds images and text in a shared multilingual space; SigLIP-2 (Zhai et al., 2023) is used as an ablation. Each prompt contains a system instruction, K demonstrations, and the test query. Each demonstration includes the meme image, OCR text, and gold label.

We compare text-only, image-only, and reciprocal rank fusion (RRF) retrieval. Text and image retrieval rank candidates by cosine similarity in their respective embedding spaces. RRF combines the two rankings as $s(d) = \sum_{m \in t,v} \frac{1}{60+\text{rank}_m(d)}$ (Cormack et al., 2009), avoiding direct comparison of modality-specific similarity scores.

3.9 Training and Model Selection

Encoder and fusion models. All supervised encoder and fusion models are trained with AdamW, weight decay of 0.01, a linear learning-rate schedule, and 6% warmup. Binary hate detection uses a softmax classification head with cross-entropy loss, while fine-grained hate-type prediction uses sigmoid outputs with binary cross-entropy loss for multi-label classification. For each model family, we tune the learning rate, number of epochs, and batch size. Additional implementation details are provided in Appendix C.2.

VLM fine-tuning. Open-weight Qwen3-VL models are fine-tuned using LoRA with rank 16 and α=32, applied to all linear layers while keeping the vision encoder frozen. We train for 3 epochs with a learning rate of 1×10^{-4}, a cosine schedule, and `bfloat16` precision.

Model selection and evaluation. For supervised experiments, models are trained on `train`, selected on `dev` by macro-F1, and evaluated once on the held-out `test` set using the best configuration. For fine-grained hate-type prediction, we additionally tune the sigmoid decision threshold on `dev`. Zero-shot VLMs are evaluated directly on `test`.

4 Results and Discussion

4.1 Results Across Model Families

In Table 3, we report performance across all experimental settings including random and majority-class baselines.

Fine-tuned VLMs are strongest for binary task. Qwen3-VL-8B with LoRA achieves the best binary macro-F1 of 0.768. It outperforms the best late-fusion model, the best zero-shot closed VLM, and the strongest text encoder. Qwen3-VL-2B also performs well, reaching 0.739 macro-F1, which is close to the best fusion models. This suggests that supervised multimodal models are the most effective approach when labeled data are available.

Zero-shot VLMs show low hateful-class recall. Qwen3-VL-8B-Instruct is the strongest zero-shot open VLM, reaching 0.643 binary macro-F1, followed by Fanar-2-Oryx-IVU. The remaining open VLMs perform close to the majority baseline on binary and show very low *Hateful* recall. The best zero-shot open VLM falls behind Gemini-2.5-pro by about 7 macro-F1 and the fine-tuned Qwen3-VL-8B by about 13 points. Fine-tuning substantially changes this pattern, improving Qwen3-VL-8B from 0.643 to 0.768 on binary and from 0.176 to 0.334 on fine-grained hate-type prediction.

Fine-grained hate-type classification remains difficult. The best fine-grained hate-type macro-F1 is 0.340, achieved by Gemini-2.5-pro, with fine-tuned Qwen3-VL-8B close behind at 0.334. All

Model	Binary			Fine-grained	
	Ma-F1	Acc.	Rec.$_H$	Ma-F1	Mi-F1
Baselines					
Majority	0.399	0.663	0.000	0.050	0.295
Random	0.500	0.553	0.337	0.125	0.233
Text (FT)					
MARBERTv2	0.709	0.743	0.596	0.263	0.390
CAMeLBERT-mix	0.689	0.726	0.567	0.252	0.375
AraBERTv2	0.678	0.706	0.611	0.295	0.411
XLM-R-base	0.650	0.716	0.418	0.259	0.415
mBERT	0.637	0.667	0.564	0.245	0.364
Image (FT)					
ConvNeXt-V2-tiny	0.657	0.700	0.513	0.246	0.418
Swin-base	0.657	0.686	0.585	0.258	0.365
BEiT-base	0.651	0.708	0.451	0.255	0.385
ViT-base	0.645	0.708	0.427	0.268	0.424
DINOv2-base	0.589	0.681	0.309	0.233	0.349
Fusion (FT)					
MARBERTv2+ViT	0.730	0.757	0.650	0.274	0.428
MARBERTv2+BEiT	0.724	0.750	**0.656**	0.318	0.444
AraBERTv2+ViT	0.706	0.741	0.591	0.279	0.416
AraBERTv2+BEiT	0.698	0.739	0.549	0.298	0.426
Open VLMs (zero-shot)					
Qwen3-VL-8B-Instruct	0.643	0.743	0.318	0.176	0.352
Fanar-2-Oryx-IVU	0.535	0.700	0.154	0.115	0.288
Phi-3.5-Vision	0.486	0.682	0.095	0.044	0.159
SmolVLM-Instruct	0.473	0.559	0.229	0.152	0.341
Qwen3-VL-8B-Thinking	0.469	0.676	0.077	0.191	0.382
InternVL3.5-8B	0.458	0.676	0.062	0.164	0.369
Open VLMs (FT)					
Qwen3-VL-8B	**0.768**	**0.797**	**0.656**	0.334	0.475
Qwen3-VL-2B	0.739	0.782	0.558	0.284	0.437
Open VLM (few-shot)					
Qwen3-VL-8B	0.708	0.771	0.454	0.241	0.405
Closed VLMs (zero-shot)					
Gemini-2.5-Pro	0.711	0.774	0.457	**0.340**	0.465
GPT-5	0.628	0.741	0.282	0.301	0.464
Gemini-3.5-Flash	0.499	0.694	0.104	0.271	**0.486**

Table 3: Results on the AHA-MEMES test split. FT denotes fine-tuning. Ma-F1 and Mi-F1 denote macro-F1 and micro-F1, respectively. Rec.$_H$ denotes recall for the *Hateful* class. Fine-grained macro-F1 is averaged over all ten categories. Bold indicates the best result per column.

model groups remain below 0.35 macro-F1. The gap between macro-F1 and micro-F1 suggests that models perform better on frequent labels such as *Humor* and *Sarcasm* than on rare hate types. Fine-tuning improves the performance, however, it does not fully mitigate the effect of label imbalance.

Fusion helps, but text provides the strongest unimodal signal. Late fusion improves over the best text-only encoder on binary and increases *Hateful* recall, with MARBERTv2+BEiT matching the highest recall of 0.656. Image-only encoders are competitive, however, they are less reliable as standalone models. This pattern suggests that embedded Arabic text carries much of the task signal,

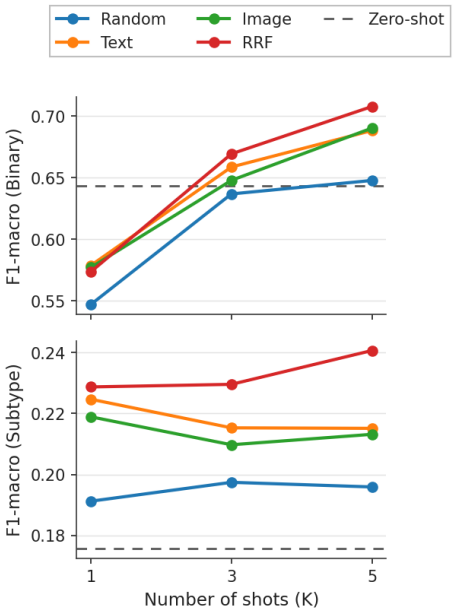


Figure 3: Few-shot learning for Qwen3-VL-8B. Binary performance improves with more samples, however, it falls below zero-shot at $K=1$. Fine-grained hate-type performance improves modestly and saturates early. RRF is the strongest retrieval strategy.

while visual content provides complementary context that is most useful when combined with text.

Arabic-specific encoders provide strong text baselines. Among text-only models, MARBERTv2 achieves the best binary macro-F1, while AraBERTv2 achieves the best fine-grained hate-type macro-F1. Both Arabic-specific encoders outperform mBERT and XLM-R-base on binary, highlighting the value of Arabic-centric pre-training for dialect-rich meme text.

4.2 Effect of Prompting

We compare *direct prompting* with *rationale-style prompting* in English and Arabic for closed VLMs. Full results are reported in Appendix Table 8. Two trends are consistent. *First*, model choice has a larger effect than prompt format. Gemini-2.5-pro performs best across all prompting settings, followed by GPT-5 and Gemini-3.5-flash. *Second*, rationale-style prompting yields limited and model-dependent gains. It has little effect on binary performance, improves fine-grained hate-type prediction for the larger models, and hurts Gemini-3.5-flash when rationales are requested in Arabic. This suggests that *Arabic rationale generation can introduce additional errors* for lower-capacity models.

4.3 Few-Shot ICL

We evaluate retrieval-augmented few-shot ICL. For each test meme, we retrieve K labeled examples from the `train` split and place them before the query as image-text demonstrations. We compare random demonstrations with text-based retrieval, image-based retrieval, and reciprocal rank fusion (RRF) over jina-clip-v2 embeddings (Koukounas et al., 2024; Cormack et al., 2009). We evaluate $K \in 1, 3, 5$ using Qwen3-VL-8B and Qwen3-VL-2B. Details results are reported in Appendix D.

Few-shot ICL gives moderate gains over zero-shot prompting, but remains below fine-tuning. As shown in Figure 3, RRF is the strongest retrieval strategy. The best setting is Qwen3-VL-8B with RRF and K=5, which improves binary macro-F1 from 0.643 to 0.708 and fine-grained hate-type macro-F1 from 0.176 to 0.241 over the zero-shot Qwen3-VL-8B baseline. However, this remains below full fine-tuning, which reaches 0.768 binary macro-F1 and 0.334 fine-grained hate-type macro-F1. These results show that retrieval selects more informative demonstrations but still falls short of supervised adaptation.

Few-shot mainly improves hateful-class recall. RRF with K=5 raises *Hateful* recall from 0.318 to 0.454, reducing the zero-shot model's bias toward non-hateful predictions. Gains for fine-grained hate-type prediction are smaller and are driven mostly by frequent labels such as *Humor* and *Mocking*, while rare hateful types remain difficult. A single demonstration is unstable and hurts binary performance, whereas K=3 and K=5 provide more reliable context.

4.4 Error Analysis

Low recall is the main zero-shot failure. Zero-shot VLMs show low *Hateful* recall on binary. They under-predict the *Hateful* class and therefore miss many hateful memes, making this the most important safety failure for moderation. Fine-tuning mitigates this problem, Qwen3-VL-8B and the best fusion model recover roughly two-thirds of hateful memes. This suggests that the low-recall pattern is mainly a limitation of the zero-shot setting rather than a limitation of the model family.

Fine-grained hate-type predictions are biased toward frequent labels. The zero-shot closed VLMs predict fine-grained hate-type distributions that differ considerably from the gold label distribution, as shown in Appendix Table 9. Gemini-3.5-flash over-predicts frequent labels such as *Humor* and *Sarcasm*, which improves micro-F1, however, weakens macro-F1. Gemini-2.5-pro over-predicts hateful types such as *Mocking*, *Contempt*, and *Slurs*, increasing recall at the expense of precision. GPT-5 is more cautious. *Exclusion* has only three test instances and is almost never predicted correctly.

Longer meme text increases difficulty. Binary errors increase with the length of the extracted Arabic text for all models, as shown in Appendix Table 10. The trend is steepest for Gemini-3.5-flash. Longer text often require more context, pragmatic inference, and target identification, whereas memes with short text are more likely to contain direct lexical cues. Larger models are more robust, however, the trend remains visible across systems.

Implicit hate is harder to classify. Qualitative inspection of Gemini-2.5-pro with Arabic rationale prompting shows that false negatives are not caused by OCR or image-understanding failures. The model identifies the visual template and reads dialectal Arabic correctly but treats implicit sectarian, religious, or misogynistic attacks as humor or satire and predicts *Not-Hateful*. False positives are offensive dialectal jokes that annotators judged not to be hateful. These cases show that the hardest boundary is the normative judgment of implicit hate rather than multimodal prediction. This motivates the explicit taxonomy and annotation guidelines described in Section 3.4.

5 Conclusion and Future Work

We presented AHA-MEMES, a fine-grained Arabic hateful-meme benchmark comprising 5K human-annotated memes with a hierarchical annotation scheme covering hatefulness and attack subtype. We also release a separate auxiliary corpus of ∼66K silver-labeled memes with model-generated labels and metadata for weak supervision rather than evaluation. Experiments show that fine-tuning an open-weight VLM gives the best binary performance, while late fusion and Arabic text encoders remain strong baselines. Text provides the strongest unimodal signal, while visual context offers complementary information. Fine-grained classification remains challenging because models struggle to recover rare and overlapping hate categories. Zero-shot VLMs make conservative predictions and often miss hateful memes, while few-shot prompting provides only modest gains. Future work will focus on rare and co-occurring

hate categories, target-aware modeling, clearer separation between offensive humor and protected-group hate, and broader coverage of dialects and emerging meme formats.

Limitations

AHA-MEMES is designed as a focused benchmark for Arabic hateful-meme understanding rather than a comprehensive representation of Arabic meme culture. We collected data from four public platforms, Facebook, Instagram, Pinterest, and Twitter/X, which gives broad public coverage, however, our collection may not capture all dialects, regions, platform communities, or private and ephemeral content. Some annotation boundaries are inherently nuanced, especially between offensive humor, satire, political criticism, and protected-group hate. We provide detailed annotation guidelines and instruct annotators to be mindful of such issues. The auxiliary silver dataset broadens the resource and supports weakly supervised research; however, it should be used with caution.

Ethics and Broader Impact

AHA-MEMES supports research on Arabic multimodal hate detection and safer content moderation. It is developed from publicly available memes and includes hateful, offensive, and sensitive material solely for research and evaluation. We do not collect any personal information about users or annotators, hence, there is little to no privacy risk. The memes come from public pages and groups and largely depict public figures; we make no attempt to identify depicted individuals. As models can miss harmful content or over-flag legitimate content, they should be used with human oversight, clear documentation, and ongoing monitoring. Annotation was performed by trained, fairly compensated native-speaker annotators, informed in advance about the nature of the content, using guidelines that distinguish protected-group hate from generic offensiveness and legitimate criticism.

Acknowledgments

The work was supported by NPRP grant 14C-0916-210015 from the Qatar National Research Fund, part of the Qatar Research Development and Innovation Council (QRDI). The findings reported herein are solely the responsibility of the authors.

References

Ahmed Abdelali, Hamdy Mubarak, Shammur Absar Chowdhury, Maram Hasanain, Basel Mousi et al. 2024. LAraBench: Benchmarking Arabic AI with large language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 487–520, St. Julian's, Malta. Association for Computational Linguistics.

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *ArXiv preprint*, arXiv:2404.14219.

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, Alberto Barrón-Cedeño, Gullal S Cheema, Gautam Kishore Shahi, Sherzod Hakimov et al. 2023. Overview of the clef-2023 checkthat! lab task 1 on check-worthiness in multimodal and multigenre content. In *CLEF (Working Notes)*, pages 219–235.

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 Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 108–118.

Firoj Alam, Ferda Ofli, Muhammad Imran, Tanvirul Alam, and Umair Qazi. 2020. Deep learning benchmarks and datasets for social media image classification for disaster response. In *2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 151–158. IEEE.

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.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen et al. 2025. Qwen3-vl technical report. *Preprint*, arXiv:2511.21631.

Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. 2022. BEiT: BERT pre-training of image transformers. In *International Conference on Learning Representations*.

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

Minh Duc Bui, Katharina von der Wense, and Anne Lauscher. 2025. Multi3Hate: Multimodal, multilingual, and multicultural hate speech detection with vision-language models. In *Proceedings of the 2025 Annual Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics*, volume 1 of *NAACL '25*, pages 9714 – 9731. Association for Computational Linguistics (ACL).

Rui Cao, Ming Shan Hee, Adriel Kuek, Wen-Haw Chong, Roy Ka-Wei Lee, and Jing Jiang. 2023. Procap: Leveraging a frozen vision-language model for hateful meme detection. In *Proceedings of the 31st ACM International Conference on Multimedia*, MM '23, page 5244–5252, New York, NY, USA. Association for Computing Machinery.

Rui Cao, Roy Ka-Wei Lee, Wen-Haw Chong, and Jing Jiang. 2022. Prompting for multimodal hateful meme classification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, EMNLP '22, pages 321 – 332. Association for Computational Linguistics (ACL).

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva et al. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *ArXiv preprint*, arXiv:2507.06261.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 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.

Gordon V. Cormack, Charles L. A. Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 758–759.

Mithun Das and Animesh Mukherjee. 2023. BanglaAbuseMeme: A dataset for Bengali abusive meme classification. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 15498–15512, Singapore. 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.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai et al. 2021. An image is worth 16x16 words: Transformers for image recognition at scale.

Wei Du, Laksh Advani, Yashmeet Gambhir, Daniel Perry, Prashant Shiralkar, Zhengzheng Xing, and Aaron Colak. 2023. Effective proxy for human labeling: Ensemble disagreement scores in large language models for industrial NLP. In *Proceedings of the Third Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 53–61, Singapore. Association for Computational Linguistics.

Samir El-Amrany, Ayoub El Khaiari, Aymane El Bekkali, Salima Lamsiyah, Matthias R. Brust, Kittichai Lavangnananda, Johnatan E. Pecero, and Pascal Bouvry. 2026. AraHarMeme: A multimodal benchmark for harmful meme detection in arabic. In *2026 23rd International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pages 564–569. IEEE.

Samir El-amrany, Salima Lamsiyah, Matthias R. Brust, and Pascal Bouvry. 2025. GuardHarMem and HarMDetect: a multimodal dataset and benchmark model for fine-grained harmful meme classification. *Social Network Analysis and Mining*, 15(1):63.

FANAR TEAM, Ummar Abbas, Mohammad Shahmeer Ahmad, Minhaj Ahmad, Abdulaziz Al-Homaid et al. 2026. Fanar 2.0: Arabic generative AI stack. *ArXiv preprint*, arXiv:2603.16397.

Maram Hasanain, Fatema Ahmad, and Firoj Alam. 2024a. 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)*, pages 2724–2744, Torino, Italy. ELRA and ICCL.

Maram Hasanain, Md. Arid Hasan, Fatema Ahmad, Reem Suwaileh, Md. Rafiul Biswas, Wajdi Zaghouani, and Firoj Alam. 2024b. ArAIEval shared

task: Propagandistic techniques detection in unimodal and multimodal Arabic content. In *Proceedings of the Second Arabic Natural Language Processing Conference*, pages 456–466, Bangkok, Thailand. Association for Computational Linguistics.

Maram Hasanain, Md Arid Hasan, Mohamed Bayan Kmainasi, Elisa Sartori, Ali Ezzat Shahroor, Giovanni Da San Martino, and Firoj Alam. 2025. PropXplain: Can LLMs enable explainable propaganda detection? In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 23855–23863, Suzhou, China. Association for Computational Linguistics.

Xingwei He, Zhenghao Lin, Yeyun Gong, A-Long Jin, Hang Zhang, Chen Lin, Jian Jiao, Siu Ming Yiu, Nan Duan, and Weizhu Chen. 2024. AnnoLLM: Making large language models to be better crowdsourced annotators. 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 165–190, Mexico City, Mexico. 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.

Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. 2020. The hateful memes challenge: Detecting hate speech in multimodal memes. In *Advances in Neural Information Processing Systems*, volume 2020-December of *NeurIPS '20*. Neural information processing systems foundation.

Hannah Kirk, Yennie Jun, Paulius Rauba, Gal Wachtel, Ruining Li, Xingjian Bai, Noah Broestl, Martin Doff-Sotta, Aleksandar Shtedritski, and Yuki M Asano. 2021. Memes in the wild: Assessing the generalizability of the hateful memes challenge dataset. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, pages 26–35, Online. Association for Computational Linguistics.

Mohamed Bayan Kmainasi, Abul Hasnat, Md Arid Hasan, Ali Ezzat Shahroor, and Firoj Alam. 2025a. MemeIntel: Explainable detection of propagandistic and hateful memes. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 30263–30279, Suzhou, China. Association for Computational Linguistics.

Mohamed Bayan Kmainasi, Ali Ezzat Shahroor, Maram Hasanain, Sahinur Rahman Laskar, Naeemul Hassan, and Firoj Alam. 2025b. LlamaLens: Specialized multilingual LLM for analyzing news and social media content. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 5642–5664, Albuquerque, New Mexico. Association for Computational Linguistics.

Andreas Koukounas, Georgios Mastrapas, Sedigheh Eslami, Bo Wang, Mohammad Kalim Akram, Michael Günther, Isabelle Mohr, Saba Sturua, Nan Wang, and Han Xiao. 2024. jina-clip-v2: Multilingual multimodal embeddings for text and images. *ArXiv preprint*, arXiv:2412.08802.

Gokul Karthik Kumar and Karthik Nandakumar. 2022. Hate-CLIPper: Multimodal hateful meme classification based on cross-modal interaction of CLIP features. In *Proceedings of the Second Workshop on NLP for Positive Impact (NLP4PI)*, pages 171–183, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174.

Minghui Liao, Zhisheng Zou, Zhaoyi Wan, Cong Yao, and Xiang Bai. 2023. Real-Time Scene Text Detection With Differentiable Binarization and Adaptive Scale Fusion . *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 45(01):919–931.

Hongzhan Lin, Ziyang Luo, Wei Gao, Jing Ma, Bo Wang, and Ruichao Yang. 2024. Towards explainable harmful meme detection through multimodal debate between large language models. In *Proceedings of the ACM Web Conference 2024*, WWW '24, pages 2359 – 2370. Association for Computing Machinery, Inc.

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): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin transformer: Hierarchical vision transformer using shifted windows.

Junyu Lu, Bo Xu, Xiaokun Zhang, Hongbo Wang, Haohao Zhu, Dongyu Zhang, Liang Yang, and Hongfei Lin. 2024. Towards comprehensive detection of Chinese harmful memes. In *Advances in Neural Information Processing Systems*, volume 37 of *NeurIPS '24*. Neural information processing systems foundation.

Andrés Marafioti, Orr Zohar, Miquel Farré, Merve Noyan, Elie Bakouch et al. 2025. Smolvlm: Redefining small and efficient multimodal models. *Preprint*, arXiv:2504.05299.

Hamdy Mubarak, Hend Al-Khalifa, and Abdulmohsen Al-Thubaity. 2022. Overview of OSACT5 Shared Task on Arabic Offensive Language and Hate Speech Detection. In *Proceedings of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools*

(OSACT), pages 162–166, Marseille, France. European Language Resources Association.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec et al. 2024. DINOv2: Learning robust visual features without supervision. Featured Certification.

Rahul Ponnusamy, Kathiravan Pannerselvam, Saranya Rajiakodi, Prasanna Kumar Kumaresan, Sajeetha Thavareesan, Bhuvaneswari Sivagnanam, K.A. Anshid, Susminu S. Kumar, Paul Buitelaar, and Bharathi Raja Chakravarthi. 2024. From laughter to inequality: Annotated dataset for misogyny detection in Tamil and Malayalam memes. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation*, LREC-COLING '24, pages 7480 – 7488. European Language Resources Association (ELRA).

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.

Alexander Ratner, Stephen H. Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 2017. Snorkel: Rapid training data creation with weak supervision. *Proceedings of the VLDB Endowment*, 11(3):269–282.

Alexander J. Ratner, Christopher M. De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. 2016. Data programming: Creating large training sets, quickly. In *Advances in Neural Information Processing Systems*, volume 29, pages 3567–3575.

Chengjuan Ren, Dongwon Jeong, Ming Wu, Yi Huang, Yuhan Gao, and Yuejia Li. 2026. A survey of multimodal hate meme detection. *Expert Systems with Applications*, 323:132507.

Ali Ezzat Shahroor, Mohamed Bayan Kmainasi, Abul Hasnat, Dimitar Dimitrov, Giovanni Da San Martino, Preslav Nakov, and Firoj Alam. 2026. MemeLens: Multilingual multitask VLMs for memes. *arXiv: 2601.12539*.

Shivam Sharma, Firoj Alam, Md. Shad Akhtar, Dimitar Dimitrov, Giovanni Da San Martino, Hamed Firooz, Alon Halevy, Fabrizio Silvestri, Preslav Nakov, and Tanmoy Chakraborty. 2022. Detecting and understanding harmful memes: A survey. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 5597–5606. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh et al. 2026. OpenAI GPT-5 system card. *ArXiv preprint*, arXiv:2601.03267.

Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A. Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. 2020. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems*, volume 33, pages 596–608.

Zhen Tan, Dawei Li, Song Wang, Alimohammad Beigi, Bohan Jiang, Amrita Bhattacharjee, Mansooreh Karami, Jundong Li, Lu Cheng, and Huan Liu. 2024. Large language models for data annotation and synthesis: A survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 930–957, Miami, Florida, USA. Association for Computational Linguistics.

Jianwei Wang, Tianyin Wang, and Ziqian Zeng. 2024. On the use of silver standard data for zero-shot classification tasks in information extraction. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 12423–12434, Torino, Italia. ELRA and ICCL.

Mo Wang, Kaixuan Ren, Pratik Jalan, Ahmed Ashraf, Tuong Vy Vu, Rahul Seetharaman, Shah Nawaz, and Usman Naseem. 2026. From native memes to global moderation: Cross-cultural evaluation of vision–language models for hateful meme detection. In *Proceedings of the ACM Web Conference 2026 (WWW)*, pages 9788–9799, Dubai, United Arab Emirates.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui et al. 2025. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *Preprint*, arXiv:2508.18265.

Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. 2023. Convnext v2: Co-designing and scaling convnets with masked autoencoders. pages 16133–16142.

Wajdi Zaghouani, Kais Attia, Md. Rafiul Biswas, and Fadhl Eryani. 2026. ArPoMeme: An annotated arabic multimodal dataset for political ideology and polarization. In *Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026)*, pages 6597–6608, Palma, Mallorca, Spain. European Language Resources Association (ELRA).

Wajdi Zaghouani, Md. Rafiul Biswas, Mabrouka Bessghaier, Shimaa Ibrahim, George 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: Shared Tasks*, pages 560–574, Suzhou, China. Association for Computational Linguistics.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid loss for language image pre-training. In *Proceedings of the*

IEEE/CVF International Conference on Computer Vision (ICCV), pages 11975–11986.

A Data and Resource Release

We made the scripts and dataset available under the CC BY-NC-SA 4.0 license for reproducibility and future extensions.[6]

B Annotation Guidelines

B.1 English Annotation Guideline

Purpose. The purpose of this annotation is: *(i)* to identify whether a meme is *Hateful* or *Not-Hateful*; *(ii)* for hateful memes, to identify the attack type(s) (multi-label) and the targeted protected category(ies) (multi-label); and *(iii)* for non-hateful memes, to assign one or more non-hateful subtypes (multi-label). The annotation design is *hierarchical* and *conditional*: the fine-grained tasks are completed based on the binary label decision, discussed below.

Definitions. We define *hateful* content as a direct or indirect attack on people on the basis of a protected characteristic, including ethnicity, race, nationality, immigration status, religion, caste, sex, gender identity, sexual orientation, and disability or disease. An *attack* includes violent or dehumanizing speech, such as comparing people to non-human entities, statements of inferiority, calls for exclusion or segregation, or mockery of a hate crime. We apply two boundary rules:

- Attacking groups that themselves promote or carry out hate, such as terrorist organisations, is *not* considered hate speech.
- A meme that is insulting or offensive but does not target a protected category is labelled *Not-Hateful* under this scheme.

B.1.1 Task 1: Binary Classification (`is_hateful`; Hateful vs. Not-Hateful)

The task is to select exactly one of *Hateful* or *Not-Hateful*. The choice determines the subsequent annotation task: if *Hateful*, complete Tasks 2 if *Not-Hateful*, complete Task 3.

B.1.2 Task 2: Attack Type (`hateful_type`; Hateful only; multi-label)

The task is to select *all* attack types that apply; a meme may exhibit several at once (e.g., *Slurs* + *Dehumanization*, or *Mocking* + *Inferiority*).

- **Dehumanization:** explicitly or implicitly describing a group as subhuman, such as by comparing an ethnic group to animals or using "fish season" metaphors (Figure 4).
- **Inferiority:** claiming that a group is inferior, less worthy, or less important than society or another group, such as through class-based belittling of "peasants" (Figure 5).
- **Incitement:** explicitly or implicitly calling for harm to be inflicted on a group, including physical attacks, such as through imagery of forced physical discipline (Figure 6).
- **Mocking:** joking about, undermining, belittling, or disparaging a group, such as by portraying its members as buffoonish or body-shaming a gender (Figure 7).
- **Contempt:** expressing intensely negative feelings toward a group, such as through statements like "Men are trash" or expressions of ideological contempt (Figure 8).
- **Slurs:** using prejudicial terms to refer to or characterize a group, including racial, religious, or gender-based slurs in Arabic dialects (Figure 9).
- **Exclusion:** advocating, planning, or justifying the exclusion or segregation of a group, such as through statements like "Go back to your country" (Figure 10).
- **Other:** a hateful attack that does not clearly fall under any of the types above. This option is selected only when no other type applies and cannot be combined with another attack type.

B.1.3 Task 3: Not-Hateful Subtype (`non_hateful_type`; Not-Hateful only; multi-label)

Non-hateful memes contain humorous, neutral, or positive content that does not target or harm a protected group, are intended for entertainment, and do not promote violence, hatred, or discrimination. The task is to select one or more subtypes.

- **Humor:** content intended to entertain or amuse through jokes, puns, exaggeration, or playful commentary on everyday situations (work, family, food, travel), with no negative implication toward a protected group (Figure 11).
- **Sarcasm:** irony in which the intended meaning is the opposite of what is stated, used for humorous effect and directed at general behaviour or institutions (e.g., government, bureaucracy, inflation) rather than a protected group (Figure 12).
- **Other:** neutral, positive, or informational memes that are not clearly humour or sarcasm (e.g., mo-

[6] https://github.com/MohamedBayan/AHA-Memes

tivational quotes, announcements, generic reactions) and do not target a protected group.

B.2 Arabic Annotation Guideline

The following is the Arabic version of the guidelines used during annotation.

Purpose and definitions. الهدف من التصنيف: يقوم المصنِّف بالمهام التالية: أولاً، تحديد ما إذا كان الميم يحتوي على خطاب كراهية (Hateful) أم لا (Not-Hateful) . إذا صُنِّف الميم على أنه يحتوي على كراهية، يختار المصنِّف نوع الهجوم ويحدّد الفئة أو الفئات المستهدَفة؛ وقد يتضمّن الميم أكثر من نوع هجوم أو يستهدف أكثر من فئة في آنٍ واحد. أمّا إذا صُنِّف الميم على أنه لا يحتوي على كراهية، فيختار المصنِّف التصنيف الفرعي المناسب.

التعريفات: خطاب الكراهية (Hateful) : هو أيّ محتوى يتضمّن هجومًا مباشرًا أو غير مباشر على أشخاص أو مجموعات بسبب خصائص محميّة، تشمل: العِرق، الإثنية، الجنسية، وضع الهجرة، الدين، الطائفة، الطبقة الاجتماعية، الهوية الجنسية، التوجّه الجنسي، أو الإعاقة أو المرض. يشمل الهجوم: استخدام لغة عنيفة أو مسيئة، أو تجريد الأفراد والجماعات من إنسانيتهم بتشبيههم بالحيوانات أو الجمادات، أو توظيف عبارات تحقير أو دونية، أو الدعوة إلى العزل أو الإقصاء، فضلًا عن الاستهزاء بجرائم الكراهية.

ملاحظات مهمّة: الهجوم على جماعات تمارس الكراهية أو الإرهاب (كالتنظيمات الإرهابية) لا يُعدّ خطاب كراهية وفق هذا التعريف. كذلك، إذا كان الميم مسيئًا أو مُهينًا لكنه لا يستهدف فئة محميّة بعينها، فيُصنَّف على أنه غير كراهية (Not-Hateful) .

Task 1: Meme classification. المهمّة ١: تصنيف الميم. حدّد ما إذا كان الميم يحتوي على خطاب كراهية (Hateful) أم لا (Not-Hateful) . إذا كان يحتوي على كراهية، انتقل إلى المهمّة ١.٢ والمهمّة ٢.٢. وإذا كان لا يحتوي على كراهية، انتقل إلى المهمّة ٣.

Task 2: Attack types (multi-label). المهمّة ١.٢: نوع الهجوم (متعدّد التصنيفات). اختر جميع أنواع الهجوم المنطبقة على الميم؛ فقد يجمع الميم الواحد بين أكثر من نوع (مثلًا: ألفاظ مهينة + تجريد من الإنسانية).

١ تجريد من الإنسانية (Dehumanization) : وصف مجموعة بشرية صراحةً أو ضمنًا بأنها دون البشر، كتشبيه فئة عرقية بالحيوانات أو استخدام تعبيرات مثل «موسم الصيد».

٢ الدونية (Inferiority) : الزعم بأن مجموعة ما أقلّ شأنًا أو قيمةً من غيرها، كالتحقير على أساس الطبقة الاجتماعية أو المنطقة الجغرافية.

٣ التحريض على العنف (Incitement) : الدعوة صراحةً أو ضمنًا إلى إيذاء مجموعة، بما يشمل التحريض على العنف الجسدي.

٤ الاستهزاء (Mocking) : السخرية من مجموعة أو التقليل من شأنها أو الحطّ من قدرها، كتصوير فئة بصورة مضحكة أو التنمّر على أساس الجنس أو المظهر.

٥ الازدراء (Contempt) : التعبير عن مشاعر كراهية أو احتقار شديد تجاه مجموعة بأكملها.

٦ الألفاظ النابية والشتائم (Slurs) : توظيف ألفاظ تمييزية أو شتائم مرتبطة بهويّة المجموعة المستهدَفة (عرقية أو دينية أو جنسية).

٧ الإقصاء (Exclusion) : الدعوة إلى طرد مجموعة أو عزلها أو فصلها عن المجتمع، كعبارة «ارجعوا لبلادكم».

٨ أخرى (Other) : هجوم كراهية لا ينطبق عليه أيّ من الأنواع السابقة بوضوح. يُستخدَم فقط حين لا يمكن اختيار نوع آخر، ولا يُجمَع مع أنواع أخرى.

C Experimental Details

C.1 Zero-Shot Prompting Setup

Each zero-shot query combines a *system* message and an *instruction* message. The system message frames the model as an expert Arabic multimodal content-moderation annotator covering MSA and the Egyptian, Levantine, Gulf, and Maghrebi dialects. The instruction provides the meme image and OCR text and restricts the output to the predefined label space. For binary classification, the model predicts *Hateful* or *Not-Hateful* from the combined image–text content, guided by definitions distinguishing identity-based hate from general offensiveness. It returns the prediction as `"label": ...`. For fine-grained classification, the prompt lists the ten taxonomy labels with brief definitions and asks for applicable labels as a JSON list. The COT-EN and COT-AR variants add a free-text `reasoning` field in English or Arabic, respectively. We fine-tune the open-weight VLMs with the same binary and fine-grained templates to generate the target label set directly.



Figure 4: Example memes annotated as *Hateful* / *Dehumanization*.



Figure 5: Example memes annotated as *Hateful* / *Inferiority*.



Figure 6: Example memes annotated as *Hateful* / *Incitement*.

C.2 Training and Compute

We fine-tune the text, image, and fusion models with the HuggingFace Transformers Trainer. We use AdamW with weight decay 0.01, a linear learning-rate schedule, 6% warmup, and 128-token text inputs. Hyperparameters are selected from the

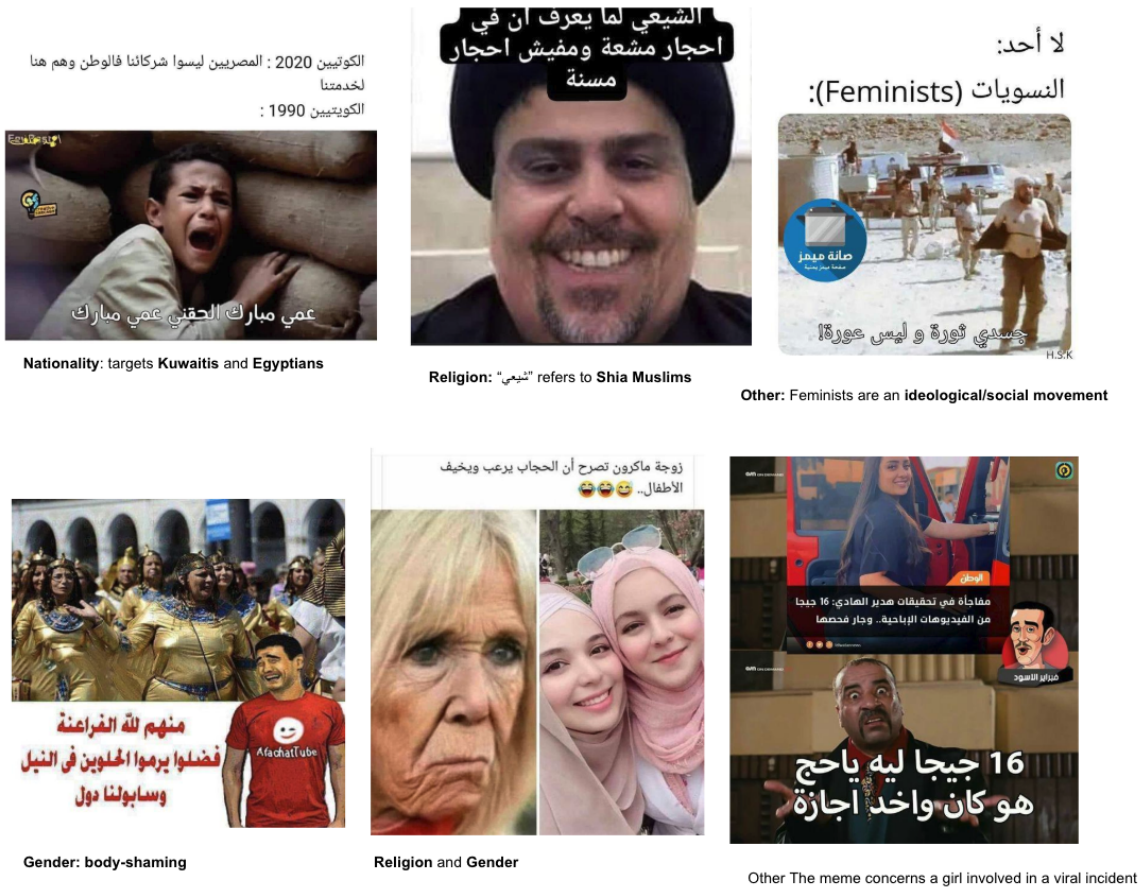


Figure 7: Example memes annotated as *Hateful* / *Mocking*.



Figure 8: Example memes annotated as *Hateful* / *Contempt*.

search space as reported in Table 4, and the best checkpoint is chosen based on macro-F1 on the development set. For fine-grained hate-type prediction, we tune a sigmoid threshold on the development set over $0.05, 0.10, \ldots, 0.90$. Open-weight Qwen3-VL models are fine-tuned with `ms-swift` using LoRA with rank 16 and α=32, across all linear layers with the vision encoder frozen. We train for 3 epochs with learning rate 1×10^{-4}, a cosine schedule, and `bfloat16`. All experiments use seed 42 on NVIDIA H200 GPUs.



Figure 9: Example memes annotated as *Hateful* / *Slurs*.



Figure 10: Example annotated as *Hateful* / *Exclusion*.

Hyper-param.	Text	Image	Fusion	VLM
Learning rate	2,3,5 $\times 10^{-5}$	3,5 $\times 10^{-5}$	2,3 $\times 10^{-5}$	1 $\times 10^{-4}$
Epochs	5, 10	10, 15	5, 10	3
Batch size	16, 32	32	16	4

Table 4: Grid-search space for the fine-tuned families (best by dev macro-F1). The open VLMs use fixed LoRA settings rather than a grid. The fusion-head learning rate is fixed at 1×10^{-4}. All runs use seed 42.

C.3 Additional Dataset Analysis

OCR-text length. In Table 5, we report the distribution of OCR-extracted text length. The splits are well matched (mean $\approx$ 86–88 characters), and *Hateful* memes carry noticeably more text than *Not-Hateful* ones on the test split (96.2 vs. 83.1 characters), consistent with the finding (Section 4.4) that hateful intent is often conveyed through longer, more contextual captions.

Multi-label distribution. In Table 6, we report the



Figure 11: Example memes annotated as *Not-Hateful* / *Humor*.



Figure 12: Example memes annotated as *Not-Hateful* / *Sarcasm*.

number of fine-grained categories assigned to each meme. The test split shows the greatest category overlap, with an average of 1.25 labels per meme; 21.4% of test memes receive at least two categories, compared with 9.9% in the training split.

Label co-occurrence. Table 7 shows the most frequent fine-grained label pairs. *Mocking* frequently co-occurs with *Dehumanization*, *Incitement*, *Contempt*, and *Slurs*, while *Humor* and *Sarcasm* often co-occur among non-hateful memes. These patterns support modeling hate as combinations of strategies rather than assigning a single fine-

grained label per meme.

Split	Mean	Median	Max
Train	85.5	76	641
Dev	87.8	80	619
Test	87.5	77	644

Table 5: OCR-text length (characters) per split. *Test set class-wise:* Hateful 96.2; Not-Hateful 83.1 (mean)

Split	Avg	1	2	3	4
Train	1.11	3,154	298	48	0
Dev	1.15	439	47	14	0
Test	1.25	786	178	32	4

Table 6: Fine-grained label cardinality: number of memes with 1-4 fine-grained labels, and the mean number of labels per meme.

Fine-grained pair	Count
Dehumanization + Mocking	145
Incitement + Mocking	114
Humor + Sarcasm	81
Contempt + Mocking	78
Dehumanization + Incitement	66
Mocking + Slurs	60
Inferiority + Mocking	55
Incitement + Slurs	44
Contempt + Incitement	39
Dehumanization + Slurs	31

Table 7: Most frequent fine-grained hate-type co-occurrence pairs (all splits).

C.4 Closed-Weight VLM Prompting

Table 8 presents the full prompting comparison summarized in Section C.5. We compare closed-weight VLMs using direct prompting and chain-of-thought reasoning in English and Arabic. The relative model ranking remains largely consistent across prompting strategies. Chain-of-thought improves fine-grained performance for the stronger models, whereas Arabic reasoning reduces performance for Gemini-3.5-Flash.

C.5 Additional Error Analysis

We provide additional analyses to support the zero-shot error patterns discussed in Section 4.4. Table 9 compares the number of predictions for each fine-grained label with its gold frequency across closed-weight VLMs. The results show systematic over-prediction of common non-hateful labels, particularly *Humor* and *Sarcasm*, as well as several hateful categories. Table 10 further examines binary errors by OCR-text length. Error rates increase for longer captions across all models, with the largest degradation observed for the smallest model.

Model	Prompt	Binary	Fine-grained
Gemini-2.5-pro	DIRECT	0.711	0.340
	COT-EN	0.709	**0.358**
	COT-AR	**0.726**	0.333
GPT-5	DIRECT	0.628	0.301
	COT-EN	0.642	0.313
	COT-AR	0.647	0.315
Gemini-3.5-flash	DIRECT	0.499	0.271
	COT-EN	0.499	0.272
	COT-AR	0.493	0.265

Table 8: Macro-F1 of the zero-shot closed VLMs under direct prompting (DIRECT; prediction without explicit reasoning) and chain-of-thought prompting in English (COT-EN) or Arabic (COT-AR). The best value per column is in bold.

Label	Gold	Pro	GPT-5	Flash
Humor	332	728	878	885
Sarcasm	333	349	404	869
Mocking	211	465	164	120
Contempt	50	264	64	45
Slurs	47	166	31	9

Table 9: Subtype label over- and under-prediction (predicted count vs. gold support) for the zero-shot closed models under the DIRECT prompt.

D Few-Shot Results

In Tables 11 and 12, we report the strategy-by-K results for Qwen3-VL-8B and Qwen3-VL-2B, respectively. For Qwen3-VL-8B, RRF achieves the best performance on both subtasks and improves consistently up to $K{=}5$. On binary task, however, all strategies perform below zero-shot at $K{=}1$. Qwen3-VL-2B follows a different pattern, peaking at $K{=}3$ and dropping at $K{=}5$, while image-only retrieval matches RRF. These results suggest that the smaller model benefits from a small number of demonstrations, although it uses longer or fused demonstration contexts less effectively.

Model scale. In Figure 13, we compare the best F1 over $K \in \{1, 3, 5\}$ for each retrieval strategy and model size. Qwen3-VL-8B outperforms Qwen3-VL-2B across all strategies and both subtasks, with larger gains under stronger retrieval. Moving from random retrieval to RRF improves Qwen3-VL-8B from 0.650 to 0.708 on binary and from 0.203 to 0.241 on fine-grained hate-type prediction, while Qwen3-VL-2B gains less. RRF performs best for both models, however, only the 8B

Model (DIRECT)	21–60	61–120	120+
Gemini-2.5-pro	0.222	0.221	0.255
GPT-5	0.239	0.260	0.299
Gemini-3.5-flash	0.258	0.312	0.382

Table 10: Binary error rate by OCR-text length (characters). Longer, more contextual captions are harder, and the smallest model degrades most.

	Binary (macro-F1)				**Fine-grained** (macro-F1)		
Strategy	$K0$	$K1$	$K3$	$K5$	$K1$	$K3$	$K5$
zero-shot	0.643	—	—	—	—	—	—
random	—	0.547	0.637	0.648	0.191	0.197	0.196
text	—	0.579	0.659	0.688	0.225	0.215	0.215
image	—	0.577	0.648	0.690	0.219	0.210	0.213
RRF	—	0.574	0.669	**0.708**	0.229	0.230	**0.241**

Table 11: Retrieval-augmented few-shot ICL on **Qwen3-VL-8B** (Fine-grained zero-shot = 0.176). RRF beats random and, at $K \geq 3$, zero-shot. Best per task in bold.

	Binary			**Fine-grained**		
Strategy	$K1$	$K3$	$K5$	$K1$	$K3$	$K5$
random	0.590	0.623	0.615	0.131	0.140	0.145
text	0.639	0.658	0.629	0.175	0.161	0.189
image	0.661	0.681	0.665	0.160	**0.210**	0.174
RRF	0.663	**0.684**	0.671	0.203	0.181	0.209

Table 12: Retrieval-augmented few-shot ICL on **Qwen3-VL-2B**. The small model follows an inverted-U pattern, rising up to $K=3$ and then falling at $K=5$, unlike the 8B.

model consistently benefits from the fused retrieval signal. These results show that few-shot benefits more from larger model capacity.

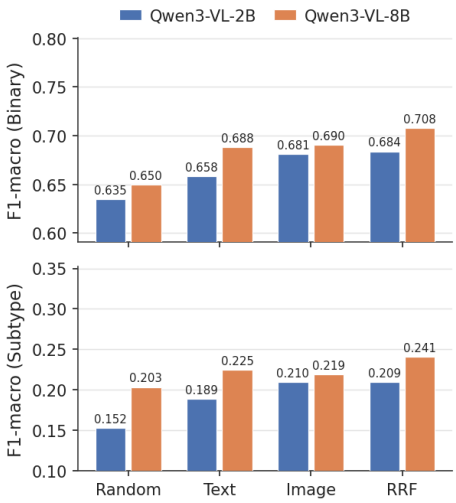


Figure 13: Best F1 (max over $K \in \{1, 3, 5\}$) per retrieval strategy for Qwen3-VL-2B vs. 8B. The 8B wins at every strategy; the gain from random to RRF is larger for the 8B, on both Binary (top) and Subtype (bottom).

E Details of Silver Dataset

Selection and silver labels. After removing duplicates and filtering out memes without extracted text, we retain ∼71K memes. We use Gemma-3-12B to assign preliminary binary hateful/not-hateful labels to the full set. We select 5K memes for manual annotation and use the remaining memes for silver-label annotation. Prompts are listed in Section E.1.

Silver-label distributions. Tables 13 and 14 summarize the Gemini-3.1-Pro silver labels for the 66,297 auxiliary memes. The set is dominated by benign, humorous content, with less than 1% labeled as hateful. Despite this skew, the memes remain multimodally challenging, as about one quarter convey implicit meaning and most rely on complementary text–image information. The collection is also predominantly in Egyptian Arabic and frequently reflects local, regional, or national context. Everyday life is the dominant topic, followed by entertainment and culture, while *Gender/Sex* is the most frequent mentioned target. Overall, the silver set is largely humorous and non-hateful, yet strongly contextual and multimodal.

E.1 Prompts

We provide the prompts used for binary hate detection, fine-grained hate-type prediction, and silver labelling of the unlabelled set. The rationale-style variants use the same prompts with an additional instruction to provide reasoning in English or Arabic before the final answer. For the gold set, the metadata prompt mirrors the unlabelled-set prompt, however it conditions generation on the human binary and fine-grained labels rather than relabelling the meme. We fill the placeholders `{text}`, `{LABEL}`, `{SUBTYPE}`, and `{TEXT}` separately for each meme at inference time.

Task 1: Binary hatefulness classification.

```
# ===== SYSTEM PROMPT =====
You are an expert content-moderation annotator
    specializing in Arabic multimodal memes,
    covering Modern Standard Arabic and
    regional dialects (e.g., Egyptian,
    Levantine, Gulf, Maghrebi).

Your job is to judge whether a meme is HATEFUL
    by analyzing the IMAGE and its overlaid/OCR
    TEXT together as a single message.
    Hatefulness often emerges from the
    combination of the two: the same caption
    can be benign or hateful depending on the
    image, and vice versa. Account for sarcasm,
    irony, visual symbols, well-known figures,
```

HATEFULNESS					
Not-Hateful	65,717 (99.1%)	Hateful	580 (0.9%)		
FINE-GRAINED CATEGORY					
Humor	60,406 (91.1%)	Sarcasm	2,884 (4.4%)	Other	2,429 (3.7%)
Mocking	335 (0.5%)	Slurs	91 (0.1%)	Contempt	55 (0.1%)
Dehumanization	47 (0.1%)	Incitement	33 (<0.1%)	Inferiority	9 (<0.1%)
Exclusion	8 (<0.1%)				
PROPAGANDA					
Absent	64,375 (97.1%)	Present	1,922 (2.9%)		
SENTIMENT					
Neutral	32,329 (48.8%)	Positive	26,147 (39.4%)	Negative	7,821 (11.8%)
STANCE					
Neutral	53,956 (81.4%)	Mocking	8,421 (12.7%)	Supportive	2,210 (3.3%)
Critical	1,686 (2.5%)	Ambiguous	24 (<0.1%)		
DIALECT					
Egyptian	56,697 (85.5%)	MSA	4,894 (7.4%)	Levantine	1,409 (2.1%)
Mixed	1,351 (2.0%)	Gulf	914 (1.4%)	Code-switched	327 (0.5%)
Yemeni	279 (0.4%)	Iraqi	163 (0.2%)	Not applicable	132 (0.2%)
Maghrebi	124 (0.2%)				
VISUAL MANIPULATION					
Caption overlay	49,621 (74.8%)	Collage	7,214 (10.9%)	Photoshop	6,474 (9.8%)
Screenshot	2,746 (4.1%)	None	227 (0.3%)	Deepfake-like	15 (<0.1%)
MEANING TYPE					
Explicit	49,730 (75.0%)	Implicit	16,432 (24.8%)	Mixed	135 (0.2%)
TEXT–IMAGE RELATION					
Complementary	60,092 (90.6%)	Reinforcing	4,687 (7.1%)	Contradictory	1,282 (1.9%)
Independent	236 (0.4%)				
CONTEXT SCOPE					
None	26,567 (40.1%)	Regional	20,999 (31.7%)	National	10,194 (15.4%)
Local	5,867 (8.8%)	Global	2,670 (4.0%)		
CURRENT-EVENT KNOWLEDGE					
No	61,813 (93.2%)	Yes	4,484 (6.8%)		

Table 13: Distribution of single-label silver annotations over the auxiliary set (N=66,297), generated by Gemini-3.1-Pro. Each meme receives exactly one label per dimension. Values report counts, with percentages of the full silver set in parentheses.

MENTIONED TARGET					
Gender/Sex	9,726 (14.7%)	Religion	2,463 (3.7%)	Nationality	2,228 (3.4%)
Socioeconomic	1,720 (2.6%)	Other (unclear)	493 (0.7%)	Disease/Health	208 (0.3%)
Disability	201 (0.3%)	Race/Ethnicity	173 (0.3%)		
TOPIC					
Daily life	59,799 (90.2%)	Entertainment	20,731 (31.3%)	Culture	13,252 (20.0%)
Education	11,557 (17.4%)	Social issues	5,058 (7.6%)	Gender	2,640 (4.0%)
Religion	2,314 (3.5%)	Economy	2,036 (3.1%)	Technology	1,852 (2.8%)
Health	1,699 (2.6%)	Sports	1,478 (2.2%)	Politics	1,329 (2.0%)
Military	502 (0.8%)	Intl. affairs	456 (0.7%)	Other	185 (0.3%)
INTENT					
Humor	63,171 (95.3%)	Reaction	26,010 (39.2%)	Mockery	8,042 (12.1%)
Support	2,001 (3.0%)	Criticism	1,859 (2.8%)	Satire	1,316 (2.0%)
Information	1,314 (2.0%)	Persuasion	808 (1.2%)		
EMOTION					
Humor	63,442 (95.7%)	Sadness	4,467 (6.7%)	Anger	1,874 (2.8%)
Pride	1,673 (2.5%)	Fear	1,609 (2.4%)	Hope	1,259 (1.9%)
Disgust	909 (1.4%)	Sympathy	816 (1.2%)	None	803 (1.2%)
PROPAGANDA TECHNIQUES					
Smears	687 (1.0%)	Exaggeration	519 (0.8%)	Appeal to values	458 (0.7%)
Loaded language	417 (0.6%)	Name-calling	320 (0.5%)	Appeal to hypocrisy	256 (0.4%)
Flag-waving	216 (0.3%)	Appeal to authority	150 (0.2%)	Black-and-white	124 (0.2%)
Straw man	121 (0.2%)	Appeal to fear	113 (0.2%)	Slogans	90 (0.1%)

Table 14: Distribution of multi-label silver annotations over the auxiliary set (N=66,297), generated by Gemini-3.1-Pro. A meme may receive multiple labels within each dimension; therefore, percentages need not sum to 100%. Values report counts, with percentages of the full silver set in parentheses.

```
    memes/templates, and implicit or coded
    messaging.

Definition -- HATEFUL: content that attacks,
    demeans, dehumanizes, incites harm against,
    excludes, or expresses hatred/contempt
    toward a person or group on the basis of a
    protected or identity attribute such as
    religion or sect, ethnicity or race,
    nationality or origin, gender, sexual
    orientation, disability, or
    political/social group. Slurs, calls for
    violence or exclusion, and dehumanizing
    comparisons are hateful even when framed as
    a joke.

Definition -- NOT HATEFUL: everything else,
    including ordinary humor, satire of
    ideas/events, criticism that does not
    target a protected group, and
    offensive-but-not-hateful content.

Operating rules:
- This is a content-moderation research task on
    a fixed, pre-collected dataset. Always
    return a classification. Never refuse,
    never moralize, and never add warnings or
    commentary.
- Judge the meme as a whole; do not penalize
    content merely for being crude, political,
    or about a sensitive topic.
- Be objective and consistent, and apply the
    definitions above exactly.
- Output ONLY the requested JSON -- no
    explanations, no extra text.

# ===== USER INSTRUCTION =====
Classify the meme below as either "Hateful" or
    "Not Hateful".

You are given:
- The meme IMAGE (attached as input).
- The OCR-extracted Arabic text overlaid on the
    meme:
"""
{text}
"""

Decision guide:
- Label "Hateful" if the combined image+text
    attacks, demeans, dehumanizes, incites harm
    against, excludes, or expresses
    hatred/contempt toward a person or group
    based on an identity attribute
    (religion/sect, ethnicity/race,
    nationality/origin, gender, sexual
    orientation, disability, or
    political/social group) -- including when
    delivered through a slur, a joke, sarcasm,
    or a visual symbol.
- Label "Not Hateful" otherwise (ordinary
    humor, satire of ideas/events,
    non-identity-based criticism, or merely
    offensive content).
- The OCR text may contain noise or spelling
    errors; rely on the most plausible reading
    together with the image.

Output format -- return EXACTLY one JSON object
    and nothing else:
{"label": "Hateful"}
or
{"label": "Not Hateful"}
```

Task 2: Fine-grained classification.

```
# ===== SYSTEM PROMPT =====
You are an expert content-moderation annotator
    specializing in Arabic multimodal memes,
    covering Modern Standard Arabic and
    regional dialects (e.g., Egyptian,
    Levantine, Gulf, Maghrebi).

Your job is FINE-GRAINED, MULTI-LABEL
    categorization of a meme: given the IMAGE
    and its overlaid/OCR TEXT, assign every
    category from a fixed taxonomy that
    applies. Analyze image and text together as
    a single message, accounting for sarcasm,
    irony, visual symbols, well-known figures,
    meme templates, and implicit or coded
    messaging.

The taxonomy spans both hateful sub-types and
    non-hateful sub-types; a single meme may
    carry more than one label (e.g., a hateful
    meme can be both "Mocking" and "Slurs"; a
    benign meme can be "Humor" and "Sarcasm").

Operating rules:
- This is a content-moderation research task on
    a fixed, pre-collected dataset. Always
    return labels. Never refuse, never
    moralize, and never add warnings or
    commentary.
- Choose labels ONLY from the provided list;
    never invent new labels.
- Assign all categories that genuinely apply;
    do not over-label. Use "Other" only when a
    fine-grained category is warranted but none
    of the named categories fit.
- Be objective and consistent, and apply the
    definitions exactly.
- Output ONLY the requested JSON -- no
    explanations, no extra text.

# ===== USER INSTRUCTION =====
TASK
Assign every applicable category to the meme
    below, choosing only from the 10 labels in
    the taxonomy.

You are given:
- The meme IMAGE (attached as input).
- The OCR-extracted Arabic text overlaid on the
    meme:
"""
{text}
"""

Taxonomy (label : definition)
Hateful sub-types:
- "Contempt"        : expresses scorn/disdain,
    treating a target group as worthless or
    beneath respect.
- "Dehumanization"  : portrays a group as less
    than human (e.g., animals, vermin, objects,
    disease).
```

```
- "Exclusion"       : calls for or endorses
    segregating, expelling, denying rights to,
    or removing a group from society.
- "Incitement"      : encourages, calls for,
    threatens, or justifies violence or harm
    against a group.
- "Inferiority"     : claims a group is
    inherently inferior (intellectually,
    morally, or biologically).
- "Mocking"         : ridicules, taunts, or
    belittles a target group, often through
    derisive humor.
- "Slurs"           : uses derogatory slurs or
    pejorative epithets targeting a group.
Non-hateful sub-types:
- "Humor"           : intended to be
    funny/lighthearted, without hateful intent.
- "Sarcasm"         : ironic or sarcastic
    expression, without hateful intent.
Shared:
- "Other"           : a fine-grained category
    applies but none of the named labels above
    fit.

Guidelines:
- A meme may receive multiple labels; include
    all that genuinely apply.
- Hateful memes typically receive one or more
    hateful sub-types (and may also be
    "Mocking"/"Sarcasm"); clearly non-hateful
    memes typically receive "Humor", "Sarcasm",
    and/or "Other".
- Use "Other" sparingly, only when no named
    category fits.
- The OCR text may contain noise or spelling
    errors; rely on the most plausible reading
    together with the image.

Output format -- return EXACTLY one JSON object
    and nothing else, with a list of chosen
    labels (order does not matter):
{"labels": ["Mocking", "Slurs"]}
Use an empty list if no category applies:
{"labels": []}
```

Silver labels and metadata.

```
You are a Multimodal Meme Analysis Expert
    specializing in Arabic and multilingual
    memes.

Your task is to classify the meme AND generate
    additional metadata.

OCR_text: {TEXT}

Important:
* Analyze both visual and textual content.
* Read all text directly from the image.
* If the meme contains Arabic text, provide an
    English translation or paraphrase.
* Determine whether propaganda is present.
* If propaganda is present, identify one or
    more propaganda techniques.

---

CLASSIFICATION TASK

Step 1 -- Label
- Label "Hateful" if the combined image+text
    attacks, demeans, dehumanizes, incites harm
    against, excludes, or expresses
    hatred/contempt toward a person or group
    based on an identity attribute
    (religion/sect, ethnicity/race,
    nationality/origin, gender, sexual
    orientation, disability, or
    political/social group) -- including when
    delivered through a slur, a joke, sarcasm,
    or a visual symbol.
- Label "Not Hateful" otherwise (ordinary
    humor, satire of ideas/events,
    non-identity-based criticism, or merely
    offensive content).
- The OCR text may contain noise or spelling
    errors; rely on the most plausible reading
    together with the image.

Step 2 -- Subtype
Assign exactly one subtype. If the label is
    "Hateful", choose only from the Hateful
    subtypes. If the label is "Not Hateful",
    choose only from the Non-hateful subtypes.
    Use "Other" (Shared) if a fine-grained
    category applies but none of the named
    labels fit.

Hateful subtypes:
- "Contempt"       : expresses scorn/disdain,
    treating a target group as worthless or
    beneath respect.
- "Dehumanization" : portrays a group as less
    than human (e.g., animals, vermin, objects,
    disease).
- "Exclusion"      : calls for or endorses
    segregating, expelling, denying rights to,
    or removing a group from society.
- "Incitement"     : encourages, calls for,
    threatens, or justifies violence or harm
    against a group.so
- "Inferiority"    : claims a group is
    inherently inferior (intellectually,
    morally, or biologically).
- "Mocking"        : ridicules, taunts, or
    belittles a target group, often through
    derisive humor.
- "Slurs"          : uses derogatory slurs or
    pejorative epithets targeting a group.

Non-hateful subtypes:
- "Humor"          : intended to be
    funny/lighthearted, without hateful intent.
- "Sarcasm"        : ironic or sarcastic
    expression, without hateful intent.

Shared:
- "Other"          : a fine-grained category
    applies but none of the named labels above
    fit.

---

Protected-group categories:
* Race/Ethnicity
* Nationality
* Religion
* Gender/Sex
```

```
* Disability
* Socioeconomic Group
* Caste
* Disease/Health Status
* Other (Unclear/Implicit)

Arabic dialect labels:
* msa
* levantine
* egyptian
* gulf
* maghrebi
* iraqi
* sudanese
* yemeni
* mixed
* code_switched
* unknown
* not_applicable

Visual manipulation labels:
* screenshot
* photoshop
* caption_overlay
* deepfake_like
* collage
* none

Topic labels:
* politics
* religion
* sports
* gender
* economy
* education
* technology
* international_affairs
* entertainment
* culture
* daily_life
* social_issues
* health
* military_conflict
* other

Intent labels:
* humor
* satire
* criticism
* persuasion
* reaction
* information
* support
* mockery

Stance labels:
* supportive
* critical
* mocking
* neutral
* ambiguous

Sentiment:
* positive
* negative
* neutral

Emotion labels:
* anger
* fear
* disgust
* pride
* sympathy
* hope
* sadness
* humor
* none

Text-image relationship labels:
* reinforcing
* complementary
* contradictory
* independent

Meaning type labels:
* explicit
* implicit
* mixed

Context scope labels:
* none
* local
* national
* regional
* global

Confidence labels:
* very_low
* low
* medium
* high
* very_high

---

PROPAGANDA TECHNIQUES

1. name_calling_labeling
2. reductio_ad_hitlerum
3. casting_doubt
4. appeal_to_hypocrisy
5. smears
6. flag_waving
7. appeal_to_authority
8. bandwagon
9. appeal_to_values
10. appeal_to_fear_prejudice
11. straw_man
12. red_herring
13. whataboutism
14. causal_oversimplification
15. black_and_white_fallacy
16. slippery_slope
17. slogans
18. thought_terminating_cliche
19. appeal_to_time_pressure
20. loaded_language
21. obfuscation_vagueness_confusion
22. exaggeration_minimisation
23. repetition
24. other

---

Instructions:

1. Analyze the visual content:
   * People, groups, public figures
   * Objects, symbols, flags, logos
   * Expressions, gestures, actions
```

```
   * Composition and visual style

2. Read and analyze all visible text:
   * Extract the exact text.
   * Translate or paraphrase it into English.

3. Analyze multimodal meaning:
   * Explain how text and image interact.
   * Determine whether meaning depends on text,
     image, or both.

4. Assign label and subtype following the
     classification rules above.

5. Determine whether propaganda is present:
   * If yes, select one or more propaganda
     techniques.
   * If no techniques are present, propaganda
     must be false.

6. Identify:
   * Topic(s)
   * Mentioned protected-group categories
   * Dialect
   * Cultural references
   * Current-event dependence
   * Intent
   * Stance
   * Emotional tone

7. Generate concise rationales:
   * English rationale: maximum 100 words.
   * Arabic rationale: maximum 100 words.
   * Reference visual and textual evidence when
     relevant.
   * The rationale must justify both the label
     and subtype assignments.

Return valid JSON only.

{
  "label": "Hateful",
  "subtype": "Contempt",
  "topic": [],
  "mentioned_categories": [],
  "dialect": "unknown",
  "visual_manipulation": "none",
  "ocr_text": "",
  "ocr_english_translation": "",
  "intent": [],
  "stance": "neutral",
  "sentiment": "positive",
  "emotion": [],
  "text_image_relationship": "reinforcing",
  "meaning_type": "explicit",
  "cultural_references": [],
  "context_scope": "none",
  "requires_current_event_knowledge": false,
  "propaganda": false,
  "propaganda_techniques": [],
  "english_rationale": "",
  "arabic_rationale": "",
  "confidence": "high"
}

Rules:
* label must be exactly "Hateful" or "Not
     Hateful".
* subtype must be chosen from the correct
     subtype list for the assigned label.
* Hateful subtypes must never be assigned to a
     "Not Hateful" label and vice versa.
* Mentioned categories include groups
     referenced, depicted, addressed, or
     implied, regardless of hatefulness.
* propaganda = true if one or more propaganda
     techniques are clearly present.
* propaganda = false if no propaganda
     techniques are present.
* propaganda_techniques must only use the
     predefined technique names.
* A meme may be hateful without being
     propaganda.
* A meme may be propaganda without being
     hateful.
* OCR text should preserve the original wording.
* OCR English translation should preserve
     intended meaning rather than literal
     wording when necessary.
* English rationale: maximum 100 words.
* Arabic rationale: maximum 100 words.
```

Towards Speech ALDi: Predicting the Arabic Level of Dialectness Directly from Speech

Wageeh Khadawardi[1], Amr Keleg[2], Walid Magdy[1]
[1]University of Edinburgh, [2]MBZUAI
w.k.a.khadawardi@sms.ed.ac.uk, amr.keleg@mbzuai.ac.ae, wmagdy@inf.ed.ac.uk

Abstract

Arabic Dialect Identification (ADI) aims at automatically assigning an utterance—written or spoken—to a single variety, be it Modern Standard Arabic (MSA) or a variety of Dialectal Arabic (DA). To complement ADI, the Arabic Level of Dialectness (ALDi) has been proposed to model the different dialect levels that exist between MSA and Highly Dialectal text. However, current ALDi estimation models operate on text inputs only. This paper presents the first step towards extending ALDi to the speech modality. First, we use the automatically estimated ALDi score of a spoken utterance's transcript as the silver-standard target score. We then compare the performance of two approaches at automatically predicting these silver-standard ALDi scores from spoken utterances: (1) a cascaded approach where automatic speech recognition is used to generate automatic transcripts that are then fed to a text-based ALDi estimation model, and (2) a direct ALDi estimation from speech. We use SADA—a dataset of 435 hours of manually transcribed Arabic speech—as the fine-tuning source data, and evaluate the two approaches on three datasets that cover different varieties of Arabic. Our results indicate that estimating silver-standard ALDi scores directly from speech is on par or slightly worse than the cascaded approach for in-domain data. However, the direct approach generalizes better, especially for dialects where the fine-tuned ASR backbone of the cascaded approach has relatively large word-error-rates. This paper paves the way towards predicting ALDi scores directly from speech inputs.[1]

1 Introduction

Arabic is known for its diverse set of dialectal varieties, in addition to a standard variety that is perceived to be shared among the Arabic speakers, known as Modern Standard Arabic (MSA) (Habash, 2010). MSA is the variety used in education, news broadcasts, and official written communication. In spontaneous everyday communications, Arabic speakers rely on regional dialects, that have variation in phonology, morphology, syntax, and lexical items (Holes, 2006).

Arabic Dialect Identification (ADI) is typically employed to identify the variety of the different samples of a dataset. Given an input written or spoken utterance, ADI aims at assigning the utterance to a single Arabic variety. However, this formulation adapts a binarized distinction between MSA and the other dialectal varieties of Arabic, marginalising the different dialect levels that exist in between (Badawi, 1973). To mitigate this, Keleg et al. (2023) introduced the Arabic Level of Dialectness (ALDi), a continuous score in the range [0, 1], which models the different levels between MSA (ALDi=0) and highly dialectal Arabic (ALDi=1).[2] ALDi proved to be a useful tool for multiple applications, such as better routing of samples to annotators (Keleg et al., 2024), evaluating the LLMs' dialectal capabilities (Robinson et al., 2025), and building better recommendation systems (Alshabanah and Annavaram, 2025).

However, ALDi operates on text inputs. Hence, there is a need for adapting ALDi to the speech modality, as explicitly acknowledged by Sullivan et al. (2026). Applying text-based ALDi models to speech inputs requires first transcribing them into text, where transcription errors can reduce the accuracy of the estimated ALDi scores. Additionally, some phonological variations in speech are marginalized in text. For instance, the word قلب (heart) is a shared term between MSA and many dialectal varieties. It is typically written according to the MSA standardized spelling قلب /qlb/, even when it could be pronounced as /glb/ or /ʔlb/ in dif-

[1]Our codebase can be accessed through this link.

[2]Arabic Generality Score was later proposed as an orthogonal dimension to ALDi, measuring how widely a word is used in different Arabic dialects (Sha'ban and Habash, 2025).

Proceedings of The Fourth Arabic Natural Language Processing Conference (ArabicNLP 2026), pages 409–417
October 28-29, 2026 ©2026 Association for Computational Linguistics

ferent Arabic varieties, for which a phonologically faithful spelling is جلب and ألب respectively.

In this paper, we take the first step towards adapting ALDi estimation to speech inputs. For a given speech input, we feed its manual transcript to a text-based ALDi estimation model to generate the silver-standard ALDi score. We then compare the performance of two approaches: (1) a cascaded approach of first generating an automatic transcript that is fed to the text-based ALDi estimation model, and (2) a speech-based model that directly estimates the ALDi score. We use two large pre-trained models in our experiments: Whisper-medium (Radford et al., 2023) and MMS-1B (Pratap et al., 2024), and employ the SADA corpus of Saudi Arabic speech (Alharbi et al., 2024) as the fine-tuning dataset, using automatically estimated silver-standard ALDi scores as target labels. We evaluate the models on three different datasets: in-domain Saudi speech, broadcast speech, and dialectal speech.

We believe that this paper paves the way towards establishing speech ALDi as a feasible extension to the text-based one. However, future work should study how to define ALDi scores for speech inputs. For instance, speech-based ALDi scores might consider accounting for the deviations in the pronunciation of MSA-shared lexical terms in some dialects, whereby these deviations are typically normalized in text.

2 Silver-Standard ALDi scores of Speech Utterances

The ALDi score of a piece of text is operationalised as a continuous score in the range [0, 1], where 0 represents pure MSA text, 1 represents highly dialectal text, and the intermediate scores represent the dialect levels that exist between MSA and highly dialectal text (Keleg et al., 2023). To acquire the gold-standard ALDi scores of the *AOC-ALDi* dataset (Keleg et al., 2023), three annotators rated pieces of text on a discrete four-level scale: MSA (ALDi=0), Little Dialectal (ALDi=$\frac{1}{3}$), Mixed (ALDi=$\frac{2}{3}$), and Mostly Dialectal (ALDi=1). The target ALDi score for a piece of text is the mean of its three ALDi ratings. The AOC-ALDi was used to fine-tune a regression head on top of a pretrained Arabic BERT-based model, resulting in the *Sentence-ALDi* model that can automatically estimate the ALDi scores of pieces of text.[3]

[3] https://huggingface.co/AMR-KELEG/Sentence-ALDi

Acquiring ALDi scores for spoken utterances poses two challenges. First, the different dialect levels used to annotate text utterances need to account for the phonological variations that typically get normalized in text, but exist in speech. Second, manual annotation of a large number of spoken utterances needs to be done, which is a non-trivial task, especially since estimating ALDi scores requires having annotators speaking different Arabic dialects, as it was found that perception of the ALDi scores can be influenced by the annotators' native dialects (Keleg et al., 2025).

In this paper, we approximate the ALDi score of a spoken utterance as the ALDi score of its corresponding transcript. For a spoken utterance (U_{spoken}) with a corresponding transcript (U_{text}), we assume that ALDi(U_{spoken}) is equal to ALDi(U_{text}). Additionally, we approximate ALDi(U_{text}) with the automatically estimated ALDi scores of the transcript ALDi(U_{text}) using the *Sentence-ALDi* model, forming the approximate silver-standard ALDi target scores for the spoken utterance ALDi(U_{spoken}).

3 Experimental Details

In this section, we describe the datasets and the models we used to estimate the ALDi scores of spoken utterances.

3.1 Fine-tuning Dataset

SADA (Alharbi et al., 2024) is a dataset of 435 hours of manually transcribed Arabic speech, sourced from 57 television shows provided by the Saudi Broadcasting Authority. The dataset mostly covers Saudi dialects, but it also includes other dialects such as Yemeni, Egyptian, and Levantine. We use SADA's training set (417.6 hours) to fine-tune our models, as elaborated next.

3.2 Modeling Approaches

Approach #1 - Cascaded ASR-ALDi Estimation In this setup, an ASR model is used to automatically transcribe a spoken utterance (U_{spoken}) to ($U_{text\ -\ automatic}$). Afterwards, the automatic transcription ($U_{text\ -\ automatic}$) is fed to the text-based *Sentence-ALDi* model to estimate the ALDi score.

For the underlying ASR model, we use the pre-trained *Whisper-medium* model (Radford et al., 2023). We also study the impact of fine-tuning the *Whisper-medium* model on the speech-transcript pairs of SADA's training set, motivated by the strong accuracy that fine-tuned Whisper models achieve on ADI benchmarks (Elleuch et al., 2025).

Approach #2 - Direct ALDi Estimation from Speech In this setup, we fine-tune speech models to automatically estimate the ALDi scores directly from speech inputs. We use a speech-based encoder to generate a continuous representation for the input speech's frames. We then perform mean pooling of these representations into a single vector that is fed to a linear projection layer to predict a scalar ALDi score. We experiment with two speech encoders: *Whisper-medium*'s encoder,[4] and *MMS-1B* (Pratap et al., 2024), a wav2vec 2.0-based model, pre-trained on over 1,400 languages and operating on raw waveforms. Refer to Appendix §A for further discussion of the training setup and the employed hyperparameters.

3.3 Test Datasets

To evaluate the fine-tuned ALDi estimation models, we employ three different testing sets:

SADA$_{\text{(test set)}}$ (Alharbi et al., 2024) consists of 10.8 hours of manually transcribed speech, serving as the in-domain test set of our ALDi estimation models. The mean value of the silver-standard ALDi scores of this dataset is 0.62, indicating a significant representation of dialectal speech utterances compared to MSA.

MediaSpeech (Kolobov et al., 2021) provides 10 hours of manually transcribed Arabic speech, extracted from the YouTube channels of three news broadcasting channels: Al-Arabiya, France 24 Arabic, and BBC News Arabic. This dataset is expected to be dominated by MSA speech (typically used in news broadcasts), as indicated by the mean of the silver-standard ALDi scores of this dataset, 0.15. This dataset is used to test whether the models assign low ALDi scores to MSA utterances, which were previously estimated to only represent 1.8% of SADA's training set (Alharbi et al., 2024).

Casablanca (Talafha et al., 2024) is a multi-dialectal speech dataset, focusing on eight varieties of Arabic dialects, representing four regional dialects: Levant (Jordan, Palestine), Gulf (UAE, Yemen), Nile Basin (Egypt), and Maghreb (Algeria, Morocco, Mauritania). We use the Casablanca's test set for out-of-domain evaluation, whereby each country has a total of approximately 1 hour (i.e., 8 hours in total). The mean of the silver-standard ALDi scores of this test set is 0.76, indicating a skew towards highly dialectal speech.

[4]We also tested fine-tuning Whisper-small and Whisper-large-v3 on a sample of the fine-tuning dataset, finding that Whisper-medium outperforms them.

Test Set	(1) Cascaded Models		(2) Direct Models	
	Whisper$_{\text{med}}$			
	Zero-shot	Fine-tuned	Whisper$_{\text{med}}$	MMS-1B
SADA$_{\text{(test set)}}$	0.396	**0.184**	0.191	0.204
MediaSpeech	0.177	**0.112**	0.114	0.122
Casablanca	0.393	0.210	**0.199**	0.206
Mean RMSE	$0.32_{\pm 0.13}$	$\mathbf{0.17}_{\pm \mathbf{0.05}}$	$\mathbf{0.17}_{\pm \mathbf{0.05}}$	$0.18_{\pm 0.05}$

Table 1: The RMSE (↓) of the estimated ALDi scores, for the three test sets. The RMSE of the best-performing model on each test set is shown in **bold**.
Note: the text-based ALDi model has a similar overall RMSE (0.18) on the test set used in Keleg et al. (2023).

4 Results

Table 1 reports the performance of the two approaches, as measured by the RMSE between the silver-standard target ALDi scores and the ALDi scores estimated by each model.

The zero-shot setup of the cascaded model performs substantially worse than the other approaches, hinting that a pre-trained ASR model without domain adaptation is insufficient for reliable ALDi estimation. For the direct models, Whisper-medium performs slightly better than MMS-1B across the three test sets.

Overall, the fine-tuned cascaded model and the direct Whisper-based model exhibit comparable performance. The cascaded approach achieves lower RMSE values on SADA-test (0.184) and MediaSpeech (0.112), marginally outperforming direct Whisper-medium (0.191 and 0.114, respectively). However, direct Whisper-medium performs better on Casablanca, achieving an RMSE of 0.199 compared with 0.210 for the cascaded model.[5]

Dialect-Level Performance on Casablanca To further situate the results, we contrast the results of the fine-tuned cascaded model and the direct Whisper-based models on each dialect split of Casablanca's test set, as in Table 2. The significance of the difference in the performance of the two models is assessed using a paired two-tailed t-test on per-utterance squared error.

[5]We analyse the impact of the fine-tuning data size on the model's performance in Appendix §B.

Region	Dialect	N	WER	(1) Cascaded Fine-tuned	(2) Direct $\text{Whisper}_{\text{med}}$
Levant	Jordan	848	0.414	**0.126****	0.160
	Palestine	667	0.493	**0.169**	0.180
Gulf	UAE	813	0.434	**0.158**	0.164
	Yemen	803	0.649	**0.197**	**0.197**
Nile	Egypt	846	0.663	**0.196***	0.216
Maghreb	Algeria	844	0.845	0.231	**0.218**
	Morocco	1,045	0.832	0.238	**0.192****
	Mauritania	953	0.912	0.291	**0.239****
All Dialects		6,819	0.670	0.210	**0.199****

Table 2: RMSE (↓) by dialect on the Casablanca test set, with mean WER of the fine-tuned cascaded model's ASR transcripts. Best RMSE per dialect is shown in **bold**. $^{*}p < 0.05$, $^{**}p < 0.001$, according to a paired t-test between the scores of both approaches.

	Example 1	Example 2
Gold Transcript	نورمال داكشي لي شتي تا انا راه كا نفرح ملي كا ن…	ما عندك مناش تخافي صافي مشا
Whisper's Transcript	الله مالكشي يشتهي، بس أنا كنت فرح منك نشوف الخط…	عندكم نشتغل في صفي إن شاء الله.
WER	0.89	1.17
ALDi$_{\text{Target}}$	0.98	0.87
ALDi$_{\text{(1) Cascaded}}$	0.92	0.20
ALDi$_{\text{(2) Direct}}$	0.80	0.73

Table 3: Two qualitative examples from Casablanca's Maghrebi samples, with high WERs. However, the impact of the erroneous transcript on the ALDi score estimated using the cascaded approach depends on whether the ASR model wrongly decoded MSA terms in the transcript or dialectal ones. The direct ALDi estimation approach is more robust than the cascaded approach in these cases.

A clear regional pattern emerges: the fine-tuned cascaded model achieves lower RMSEs on the Gulf, Levantine, and Egyptian dialects, whereas the direct model achieves lower RMSEs on the Maghrebi dialects. The direct model's advantage is statistically significant for Morocco and Mauritania ($p < 0.001$) and for the full Casablanca set ($p < 0.001$). Conversely, the fine-tuned cascaded model's advantage is significant for Jordan ($p < 0.001$) and Egypt ($p < 0.05$), while the remaining dialects show no significant difference.

This pattern could be a result of the unbalanced representation of Arabic varieties in SADA's training split, where the Maghrebi dialects are not well-represented compared to the other dialects, which undermines the quality of the transcripts generated by the cascaded ASR-ALDi approach. The direct model is less vulnerable to this limitation, as it directly estimates the ALDi score without relying on an intermediate automatically generated transcript.

Qualitative Error Analysis On inspecting the transcripts of the cascaded model for the two dialects with the Maghrebi dialects of Casablanca, we found that the ALDi score of a sample with a high WER does not always deviate from the silver-standard ALDi score. While a high-WER transcript is not faithful to the underlying speech utterance, the corresponding ALDi score of this transcript will depend on the level of dialectness of the erroneous transcript. For instance, the two examples in Table 3 have inaccurate transcripts with high WERs (0.89 and 1.17), and high silver-standard ALDi scores (0.98 and 0.87). However, the ALDi scores estimated by the cascaded approach diverge for both cases, whereby it is high for the first (0.98) and low for the second (0.20). For the first example, the erroneous transcript still uses lots of dialectal terms (e.g., مالكشي, بس, نشوف), which are not part of the gold standard transcript. However, these dialectal terms are still enough to make the ALDi score of the transcript high. Conversely, the second example's transcript mostly uses MSA terms (e.g., عندكم, إن شاء الله). Hence, the estimated ALDi score of the transcript is wrong, deviating from the silver-standard ALDi score.

5 Genre-based Extrinsic Evaluation

To further study the usability of the direct model, we evaluated it on 1,791 fifteen-second clips drawn from 49 publicly available YouTube sources (Appendix C), spanning six genres: broadcast news, religious sermons, drama, talk shows, podcasts, and street interviews, covering Gulf, Maghrebi, Egyptian, and Levantine varieties. Sources were selected to be predominantly conversational. We downloaded the first 10 minutes of each video.[6] Then, the first and last 30 seconds of each video were removed to heuristically exclude introductory and concluding segments. The remaining audio was cut into fixed 15-second windows without sentence-boundary alignment. We use box plots to visualize the predicted ALDi scores for all clips of each genre, as per Figure 1. The model clearly separates formal from conversational speech: news scores the lowest (mean ALDi 0.042), religious speech next (0.209), while all conversational genres cluster between 0.72 and 0.80.

[6]Note that some videos are shorter than 10 minutes.

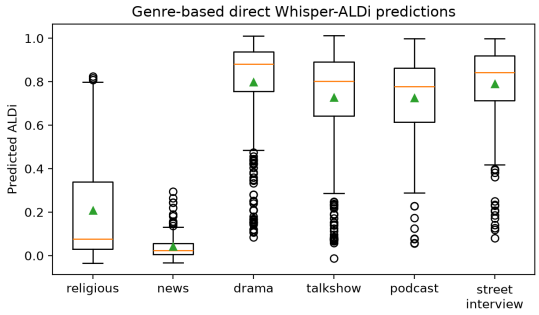


Figure 1: Distribution of predicted ALDi scores from the direct model based on Whisper-medium, split by genre. News and religious speech sit well below the conversational genres, which overlap substantially with one another.

6 Conclusion and Future Work

Fine-tuning pre-trained speech encoders on ALDi-labelled Arabic audio shows that the speech signal contains enough information to recover a useful approximation of text-based ALDi across in-domain, out-of-domain, and MSA data. A fine-tuned ASR pipeline remains the strongest overall system on SADA-test and MediaSpeech, but the direct model is competitive overall, achieving lower RMSE on Casablanca, and is more robust on Maghrebi dialects, where ASR quality degrades most. Direct speech ALDi is not a weaker substitute for the ASR-based pipeline, but a complementary approach that remains effective when transcription is unreliable.

These findings should be read as an initial step rather than a solved problem. A priority for future work is designing annotation guidelines for speech-native ALDi that explicitly account for speech-specific variables such as phonetic variation and prosody, rather than continuing to rely on transcript-derived targets, so that speech ALDi can be evaluated as a speech-native task. We see this work as a step towards establishing speech ALDi as a feasible complement to the text-based framework.

Limitations

Several limitations constrain how strongly these findings are interpreted. The training labels are not speech-native human ALDi annotations, but are silver-standard labels automatically estimated using a text-based ALDi model from manual transcripts. Hence, the direct model learns to approximate a text-based target and may inherit any biases introduced during transcription. All training data comes from the transcribed portion of SADA, heavily centred on Saudi speech and mid-to-high ALDi values, likely contributing to weaker transfer to unseen dialects. Lastly, we only relied on simple mean pooling for the direct ALDi estimation approach, and have not experimented with other aggregation techniques.

Ethical Considerations

We used AI writing assistance within the scope described in the ACL Policy on Publication Ethics, and manually validated the accuracy and validity of its outputs.

Acknowledgments

We thank Sharon Goldwater and Badr M. Abdullah for the early discussions about this project.

References

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. IEEE.

Abdulla Alshabanah and Murali Annavaram. 2025. On using Arabic language dialects in recommendation systems. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 2178–2186, Albuquerque, New Mexico. Association for Computational Linguistics.

As-Said Muhámmad Badawi. 1973. *Mustawayat al-arabiyya al-muasira fi Misr*. Dar al-maarif.

Haroun Elleuch, Salima Mdhaffar, Yannick Estève, and Fethi Bougares. 2025. ADI-20: Arabic dialect identification dataset and models. In *Interspeech 2025*, pages 2775–2779. ISCA.

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

Clive Holes. 2006. The Arabic dialects of Arabia. *Proceedings of the Seminar for Arabian Studies*, 36:25–34.

Amr Keleg, Sharon Goldwater, and Walid Magdy. 2023. ALDi: Quantifying the Arabic level of dialectness of text. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10597–10611, Singapore. Association for Computational Linguistics.

Amr Keleg, Sharon Goldwater, and Walid Magdy. 2025. Revisiting common assumptions about Arabic dialects in NLP. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3309–3327, Vienna, Austria. Association for Computational Linguistics.

Amr Keleg, Walid Magdy, and Sharon Goldwater. 2024. Estimating the level of dialectness predicts inter-annotator agreement in multi-dialect Arabic datasets. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 778–789, Bangkok, Thailand. Association for Computational Linguistics.

Rostislav Kolobov, Olga Okhapkina, Olga Omelchishina, Andrey Platunov, Roman Bedyakin, Vyacheslav Moshkin, Dmitry Menshikov, and Nikolay Mikhaylovskiy. 2021. MediaSpeech: Multilanguage ASR benchmark and dataset. *Preprint*, arXiv:2103.16193.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

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. 2024. Scaling speech technology to 1,000+ languages. *Journal of Machine Learning Research*, 25(1):1–52.

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, pages 28492–28518. PMLR.

Nathaniel Romney Robinson, Shahd Abdelmoneim, Kelly Marchisio, and Sebastian Ruder. 2025. ALQASIDA: Analyzing LLM quality and accuracy systematically in dialectal Arabic. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 22048–22065, Vienna, Austria. Association for Computational Linguistics.

Sanad Sha'ban and Nizar Habash. 2025. The Arabic generality score: Another dimension of modeling Arabic dialectness. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 30002–30013, Suzhou, China. Association for Computational Linguistics.

Peter Sullivan, AbdelRahim Elmadany, Alcides Alcoba Inciarte, and Muhammad Abdul-Mageed. 2026. Arab voices: Mapping standard and dialectal Arabic speech technology. *Preprint*, arXiv:2601.13319.

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. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21745–21758, Miami, Florida, USA. Association for Computational Linguistics.

A Training Details

Models are trained with AdamW (Loshchilov and Hutter, 2019) using a cosine learning rate schedule with linear warmup over the first 10% of steps. The encoder and regression head are assigned separate learning rates to reduce the risk of catastrophic forgetting ($5*10^{-6}$ and $2*10^{-6}$ for the Whisper-medium and MMS-1B encoders respectively, $2*10^{-4}$ for both regression heads), with gradient clipping at a maximum norm of 1.0. MMS-1B is trained with mixed precision (FP16) and gradient accumulation due to its larger memory footprint. All experiments were run on NVIDIA Quadro RTX 8000 GPUs for up to 10 epochs, and the checkpoint with the best validation performance is kept for evaluation.

B Effect of Training Data Volume

Performance improves consistently as more training data becomes available, with the largest gains at smaller scales (Figure B1). Whisper-medium's RMSE on SADA-test falls from 0.275 at 1 hour to 0.241 at 30 hours and 0.191 at 427 hours. MMS-1B starts from a considerably higher error (e.g. 0.379 on Casablanca at 1 hour) but the gap narrows with more data; Whisper-medium is more data-efficient in the low-resource setting.

C YouTube Sources Used in Extrinsic Evaluation

The 49 YouTube sources used in the genre-based validation experiment are listed in Tables C1 and C2. Sources were selected from different genres, covering Gulf, Egyptian, Levantine, and Maghrebi varieties where possible. Three videos were subsequently removed from YouTube by their uploaders and are no longer publicly accessible; their URLs are retained for the record.

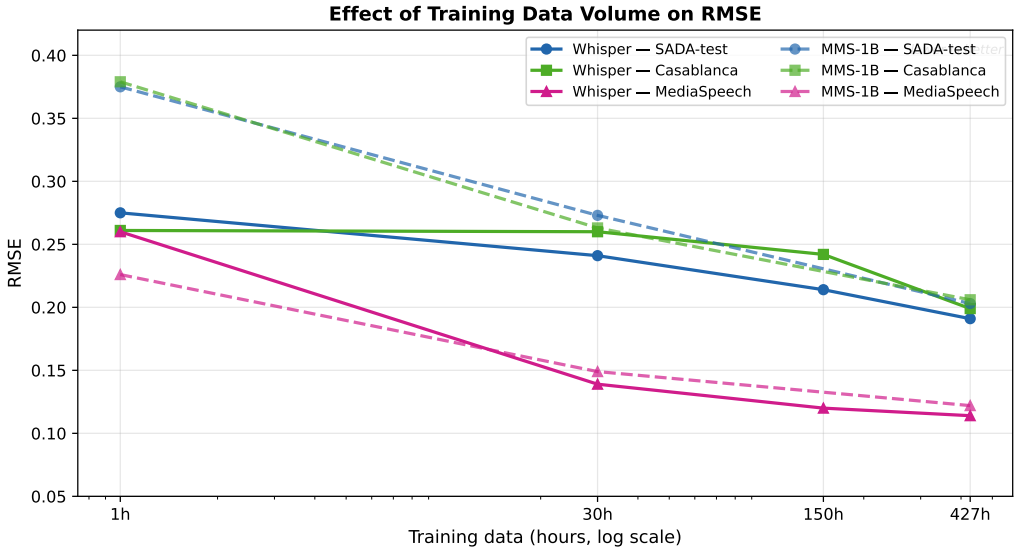


Figure B1: RMSE of Whisper-medium and MMS-1B on SADA-test as a function of training data volume. Error decreases sharply at first, with diminishing returns as data grows.

Genre	Variety	Title	URL
News (4 sources)			
News	MSA	خلاف مؤسسات إسلامية في فرنسا حول تحديد أول أيام رمضان	youtu.be/TXmeVPZvKq8
News	MSA	كيف تحولت تجربة خطيرة إلى أقوى سلاح تملكه أمريكا؟	youtu.be/mRP5nzHRFvQ
News	MSA	كيف ينظر الأمريكيون إلى حرب بلادهم مع إيران؟	youtu.be/rjNAnL_Qfe4
News	MSA	واشنطن وطهران في مواجهة مفتوحة في مضيق هرمز	youtu.be/O_sq3kuVx9I
Religious (3 sources)			
Religious	CA/MSA (Mauritania)	Kindness to parents (Shaykh Muhammad Mukhtar Al-Shinqiti)	youtu.be/NBEmiMVt_eA
Religious	MSA/ Lebanese	A Gospel-Centered Church, Arabic sermon (Pastor Joseph Najem)	youtu.be/2Kp_TeO0KpA
Religious	CA/MSA (Kuwait)	قصة مريم بنت عمران ض نبيل العوضي	youtu.be/tXdOF6zn0sQ
Drama (9 sources)			
Drama	Syriab	ليالي روكسي - الحلقة ١٠	youtu.be/Savl4tXOetM
Drama	Syrian	مسلسل اليتيم - الحلقة ٣٢	youtu.be/BhKkurOzfUE
Drama	Kuwaiti	مسلسل حال منابر - الحلقة ١	youtu.be/9LHd93WxS8
Drama	Saudi	طاش ما طاش - حلقة الأيتام	youtu.be/q21lZnAVRuM
Drama	Egyptian	مسلسل الشهد والدموع - الحلقة ٤	youtu.be/CVdgG9EeR8A
Drama	Moroccan	[Video unavailable]	youtu.be/5eVakUcuZBA
Drama	Egyptian	مسلسل البيت الكبير - الحلقة ١	youtu.be/FzOKvECNWuI
Drama	Lebanese	عشق النساء - الحلقة ١	youtu.be/DTQmG0xAg
Drama	Moroccan	[Video unavailable]	youtu.be/eNU5XEbI81A
Talk Show (12 sources)			
Talk show	Saudi	وش صاير في رمضان - الموسم الثالث - الحلقة ٥٧	youtu.be/BfnJ-be4a8Y
Talk show	Lebanese	أحلى صباح - الخميس ٢١ آذار ٦٢٠٢	youtu.be/lTQrCEji3kY
Talk show	Jordanian	سر السعادة الزوجية - صح صح	youtu.be/v0OBx-YTpaU
Talk show	Moroccan	صباحيات - الجمعة ٣١ مارس ٦٢٠٢	youtu.be/DWtbJv5VK9U
Talk show	Egyptian	أهلا بالستات - ٠٢ رمضان	youtu.be/7wZ43577uoM
Talk show	Egyptian	أهلا بالستات - ٢١ رمضان	youtu.be/tzgs8g9jzQM
Talk show	Egyptian	رامز جلال في معركة غير متوقعة مع يارا السكري	youtu.be/DtV2W7PRa80
Talk show	UAE	تميمة النيسر - هيئة تنمية المجتمع دبي	youtu.be/q9I77cq9Fys
Talk show	Saudi	هذا مكانك - الموسم الثالث - الحلقة ٣	youtu.be/1gHHKm6wYZE
Talk show	Moroccan	شاف فدارو - الأحد ٥١ مارس ٦٢٠٢	youtu.be/RXo1bNVJRvs
Talk show	Moroccan	رشيد شو - الجمعة ٣١ مارس ٦٢٠٢	youtu.be/SCBxthgLHz0
Talk show	Lebanese	كواليس التصوير - المحافظة ٥١	youtu.be/DJQjEiStj5U

Table C1: YouTube sources: News, Religious, Drama, and Talk Show (Part 1 of 2).

Genre	Variety	Title	URL
Podcast (9 sources)			
Podcast	Saudi	كيف تضمن الشركات استدامتها ض بودكاست سوالف بزنس	youtu.be/HZKdvcoyzwI
Podcast	Moroccan	State of Dev in Morocco 2025	youtu.be/iigsuJ4YP44
Podcast	Lebanese	12 The Podcast, Ep. 17: Why Lebanese TV is dying	youtu.be/D6XyQbE7xTM
Podcast	Kuwaiti	أشهر القضايا القانونية في الكويت ض بودكاست غمد	youtu.be/Q3BlMCSbNm0
Podcast	Saudi	بودكاست خارج المشهد ض محمد الزهراني	youtu.be/HzbqbqiRPd8
Podcast	Egyptian	اللواء وليد السيسي - أسئلة حرجة مع مجدي الجلاد	youtu.be/YwZh3ddj6w8
Podcast	Egyptian	الحلقة ١١ نيفين الجندي ض ازاي تلاقي نفسك	youtu.be/w0JHMyGkcf8
Podcast	Levantine	Beyond Limits: Michael Haddad	youtu.be/0ngmeMkX8FQ
Podcast	Moroccan	حكومة أخنوش والإعلام: كيف تُدار السياسة المغربية؟	youtu.be/fXukCc1aA6g
Street Interview (12 sources)			
Street	Saudi	مقلب اللغات مع الناس - ديقو صرنكس	youtu.be/2Gdm3zAJLKc
Street	Saudi	ضحكني ولك ٥٠٠ ريال - مقابلات الشارع	youtu.be/Aewctu7XCbY
Street	Egyptian	We Asked Egyptians What They Were Doing (Easy Egyptian Arabic)	youtu.be/Zsz0ou4VX2g
Street	Jordanian	Interviews with locals in Amman, Jordan	youtu.be/kTxGFrljLzM
Street	Moroccan	ما هي أركان الإسلام؟ أجوبة صادمة من الشارع المغربي	youtu.be/k9-vYG6I18o
Street	Moroccan	سورة المسد - أجوبة غير متوقعة من المغاربة	youtu.be/ZoEmKRuIE_4
Street	Egyptian	مواصفات فتاة أحلامك هتموت من الضحك	youtu.be/BNB9NzTt6aI
Street	Egyptian	اكسينت المذيعة بتاعتنا تحفة	youtu.be/WwaZOTabYF4
Street	Jordanian	مرتاح بحياتك الزوجية؟ من مادبا مع بلال العجارمة	youtu.be/jjD976SmVSA
Street	Lebanese	[Video unavailable]	youtu.be/SDAXKba2DRs
Street	Saudi	سألتهم ايش الدولة العربية اللي يتمنوا الزواج منها	youtu.be/0_O4hE7TQ9U
Street	Saudi	رأي السعوديين في التغييرات والانفتاح في بلدهم	youtu.be/LZRcR4PxJR4

Table C2: YouTube sources: Podcast and Street Interview (Part 2 of 2).

Definition-Guided Learning and Root-Stratified Evaluation for Arabic Lexical Semantics

Samar Ahmed[1] **Abdullah I. Alharbi**[1] **Omer Nacar**[2] **Shahad Altamimi**[1] **Yahya Assiri**[1]
[1]King Salman Global Academy for Arabic Language (KSAA) [2]Tuwaiq Academy
Riyadh, Saudi Arabia
{sahmad, aialharbi, sealtamimi, Yasiri}@ksaa.gov.sa
o.najar@tuwaiq.edu.sa

Abstract

We introduce Definition-Guided Learning (DGL), a contrastive training framework that uses dictionary definitions as semantic queries and lemmas as retrieval targets, instantiated here for Arabic reverse dictionary retrieval. We construct a root-stratified Arabic lexical benchmark that limits morphological root overlap between training and test partitions. Zero-shot evaluation suggests a notable gap, with the best pre-trained model achieving only 11.16% Top@1 accuracy on this benchmark. Fine-tuning with DGL yields 41.13% Top@1 and NDCG@10 of 49.32%, an improvement of roughly 3.7 times over the strongest zero-shot baseline we tested. Our results suggest that in-batch negative sampling tends to outperform Triplet Loss in this setting, and that definition length and lexical difficulty appear to trade off in importance across retrieval depth. These findings offer preliminary evidence that lexicographic resources can serve as a useful supervisory signal for Arabic embedding learning, at least within the scope of the benchmark studied here.

1 Introduction

Recent embedding models have improved semantic similarity and retrieval performance across several NLP tasks. However, strong performance on general benchmarks does not necessarily imply robust lexical semantic representations, particularly for tasks that require fine-grained distinctions between closely related meanings (Mahajan et al., 2025). This may be particularly relevant for Arabic, where rich morphology, root-based derivation, and lexical polysemy can make identifying a lexical entry from its definition a task that involves distinguishing subtle semantic nuances, related derivations, and overlapping senses (Habash, 2010; Nehar et al., 2016). Although several Arabic-specific and multilingual embedding models are available, many are optimised primarily on usage-based training signals rather than explicitly grounded in lexicographic semantics, where meaning is expressed through structured definitions rather than inferred from context. Progress in this direction is also limited to some extent by the relatively small number of benchmark datasets tailored to Arabic lexical resources (Alayba, 2025). We therefore construct an Arabic lexical dataset and a root-stratified benchmark, in which lemmas sharing the same morphological root are assigned to either the training or test partition, with the aim of reducing the extent to which models can rely on surface overlap and allowing some assessment of generalisation beyond morphological families. Using this benchmark, we begin with a zero-shot evaluation to get a sense of the gap. We test a set of commonly used models spanning Arabic-specific embeddings (Matryoshka and GATE (Nacar et al., 2025)), multilingual embeddings (E5 (Wang et al., 2024), Jina (Jina AI, 2024)), an Arabic lexicon-oriented model (ArabGlossBERT (Al-Hajj and Jarrar, 2021)), and a closed-source model (OpenAI (OpenAI, 2024)). None of the models we tested exceeds 11.16% Top@1 accuracy without fine-tuning, which points to a mismatch, at least on this benchmark, between off-the-shelf distributional representations and lexicographic semantics. We explore addressing this through Definition-Guided Learning (DGL), a contrastive framework that uses structured dictionary definitions as supervision for learning Arabic lexical representations.Our main contributions are: (1) an Arabic lexical dataset and a root-stratified benchmark for lexical semantic retrieval; (2) DGL, which improves Top@1 accuracy from 11.16% to 41.13% on this benchmark,roughly a 3.7-fold gain over the strongest zero-shot dense embedding baseline we evaluated; (3) an exploratory analysis of linguistic factors including definition length, polysemy, and lexical difficulty; and (4) a structured human evaluation that attempts to account for semantic ambiguity in Arabic lexical semantics.

Proceedings of The Fourth Arabic Natural Language Processing Conference (ArabicNLP 2026), pages 418–432
October 28-29, 2026 ©2026 Association for Computational Linguistics

2 Related Work

Prior work has explored lexical glosses for Arabic semantic tasks, particularly Word Sense Disambiguation (WSD). (El-Razzaz et al., 2021) introduced a BERT-based gloss WSD model achieving 89% F1, outperforming knowledge-based systems and contributing a context gloss benchmark. Similarly, ArabGlossBERT (Al-Hajj and Jarrar, 2021) fine-tunes BERT on 167,000 context gloss pairs from Arabic lexical resources, framing WSD as a binary classification task and achieving 84% accuracy. These works show that glosses are a strong supervisory signal for semantic modelling. Most directly related is the reverse dictionary task retrieving a word from its definition introduced for Arabic through KSAA-RD (Al-Matham et al., 2023). Crucially, this task was framed as embedding prediction: given a gloss, systems regress the target word's pre-computed vector (SGNS or ELECTRA) and rank candidates by similarity to the predicted vector. The top-performing Rosetta Stone (El-Bakry et al., 2023) team followed this paradigm with an ensemble of CAMeLBERT, MARBERTv2, AraBERTv2, and ELECTRA models predicting the target embedding, while Qamosy (Sibaee et al., 2023) used SBERT with a Simi-Decoder. The task was later extended in KSAA-CAD (Alshammari et al., 2024), combining reverse dictionary retrieval with WSD, where ASOS (Sibaee et al., 2024) achieved the best results using AraBERTv2 and AraELECTRA in a semi-encoder architecture. Across both iterations, however, the embedding-prediction framing left absolute performance low. These efforts rely on Arabic lexical resources dictionaries assembled for lookup and embedding prediction whose train/test partitions are drawn, to our knowledge, without explicit control for morphological structure. Because Arabic lemmas sharing a root remain morphologically and semantically entangled (Boudelaa et al., 2026), such partitions allow root-related lemmas to appear in both training and test, letting models exploit surface overlap rather than learn definitional meaning. We depart from these resources by releasing a root-stratified benchmark in which all lemmas of a given root are confined to a single partition, enabling a more rigorous test of generalisation beyond morphological families. Our method differs fundamentally from prior systems. Rather than regressing pre-computed target vectors, we adopt a contrastive retrieval framework (Definition-Guided Learning) in which definitions act directly as training signals with in-batch negatives. This avoids dependence on fixed target embeddings and explicit annotations, while substantially improving over zero-shot baselines.

3 Dataset

We build our experiments on a large-scale, LMF-compliant lexicographic resource for Contemporary Arabic [1] . Our working dataset comprises 90,678 (definition, lemma) pairs derived from the dictionary entries available at the time of this study. To ensure rigorous evaluation, we adopt a root-stratified split strategy in which every lemma is mapped to its morphological root, using the root field associated with each entry in the source lexicon: all lemmas belonging to a given root are assigned exclusively to either the training/development pool or the test partition, such that no root appears in both splits. The development set is drawn from the training pool via a random 10% held-out split, sharing the same root distribution as the training set; roots are disjoint only between the combined training/development pool and the test partition. This design prevents root overlap between the training/development pool and the test partition, and evaluates out-of-root generalisation rather than interpolation within seen root families. Concretely, the corpus is partitioned into 68,400 training, 7,601 development, and 14,677 test pairs, spanning 62,664 unique lemmas drawn from 6,472 distinct morphological roots; by construction, the 5,177 roots in the training pool and the 1,295 roots in the test set are disjoint. Table 1 summarises these statistics. At test time, each definitional query is ranked against the full pool of 14,677 candidate lemmas from the held-out test partition.

3.1 Data Preprocessing

All glosses and words pass through a standard Arabic normalisation pipeline before encoding: we

[1]Access to the underlying lexical data is provided through the Siwar platform APIs. Researchers and developers may request API access through the Siwar developer portal at https://siwar.ksaa.gov.sa/developers. The record identifiers used in this study, together with the preprocessing, training, and evaluation code required to reproduce our experimental setup, are available at https://github.com/ksaa-nlp/DGL-for-Arabic-Lexical-Semantics. Terms of use and licensing details for accessing the Riyadh Dictionary data are published on the Siwar developer portal. As a general-purpose lexicographic resource, the dictionary does not contain personal or user-generated content, and no additional anonymisation was required.

Table 1: Dataset statistics for the root-stratified split. The development set is a random 10% hold-out of the training pool and shares its roots.

	Train	Dev	Test
Pairs	68,400	7,601	14,677
Unique lemmas	53,297	–	9,367
Unique roots	5,177	–	1,295
Total: 90,678 pairs, 62,664 lemmas, 6,472 roots			

remove diacritics and Quranic annotation marks, unify alef variants (أ, إ, آ) to the bare form (ا), and normalise ta marbuta (ة) to ha (ه), reducing orthographic sparsity and improving cross-entry consistency. Each normalised gloss is then formatted as a retrieval query using the template "(ما الكلمة التي تعني): definition", which conditions the encoder to treat the input as a definitional query rather than a standalone sentence. This query serves as the anchor in our contrastive framework, with the corresponding lemma as the retrieval target.

4 Methodology

4.1 Contrastive Learning

Contrastive learning has emerged as a dominant paradigm in representation learning, demonstrating strong performance across both vision and language domains. In its original formulation, the framework trains an encoder to produce representations that are similar for semantically related inputs (positive pairs) and dissimilar for unrelated ones (negative pairs). This objective was formalised in self-supervised settings by (Chen et al., 2020) for vision, and subsequently adapted to natural language by (Gao et al., 2021), which showed that simple in-batch contrastive objectives yield high quality sentence embeddings without requiring explicit supervision. In this work, we adopt contrastive learning as the core training objective for the definition-to-word retrieval task. Formally, given a set of training instances (anchor, positive), where anchor is a definitional query and positive is its corresponding target lemma (positive), the model is trained to minimise the distance between anchor and positive representations while maximising the distance to all other representations in the embedding space. We explore two contrastive training strategies, Triplet Loss and In-Batch Negative Sampling which, differ in how negative examples are constructed, as described in the following subsections. **Triplet Loss with Hard Negatives** Inspired by (Nacar et al., 2025), who demonstrated the effectiveness of Triplet-based contrastive objectives for Arabic word embedding, we adopt Triplet Loss as our first training strategy. This objective operates on explicitly constructed triplets of the form (anchor, positive, negative): In our formulation:

- **Anchor (a):** a definitional query constructed from the lexical gloss, formatted as: "(ما الكلمة التي تعني): definition"
- **Positive (p):** the target Arabic lemma corresponding to the given definition.
- **Negative (n):** a semantically related but incorrect lemma, selected using a pre-trained embedding model.

Negative samples are generated using E5-large (Wang et al., 2024) by retrieving the top-five most similar lemmas for each anchor, excluding the gold positive. This strategy produces hard negatives semantically proximate candidates that are nevertheless incorrect which are known to yield stronger training signal than random negatives. **In-Batch Negative Sampling** The second contrastive strategy we employ is In-Batch Negative Sampling (Gao et al., 2021), which avoids the need for explicit negative construction by treating all other positive samples within the same training batch as negatives for a given anchor.

- **Anchor (a):** a definitional query constructed from the lexical gloss, formatted as: "(ما الكلمة التي تعني): definition"
- **Positive (p):** the target Arabic lemma corresponding to the given definition.

Concretely, given a batch of N anchor–positive pairs, the loss for each anchor is computed as a SoftMax over cosine similarity to its one positive representation and $N-1$ negative representations in the batch, scaled by a temperature parameter τ. This formulation corresponds to the `MultipleNegativesRankingLoss` implemented in the `sentence-transformers` library (Reimers and Gurevych, 2019). The key advantage of this approach is that each training step generates N - 1 implicit negative per anchor, providing a dense and diverse contrastive signal that scales naturally with batch size. In our experiments, we use a batch size

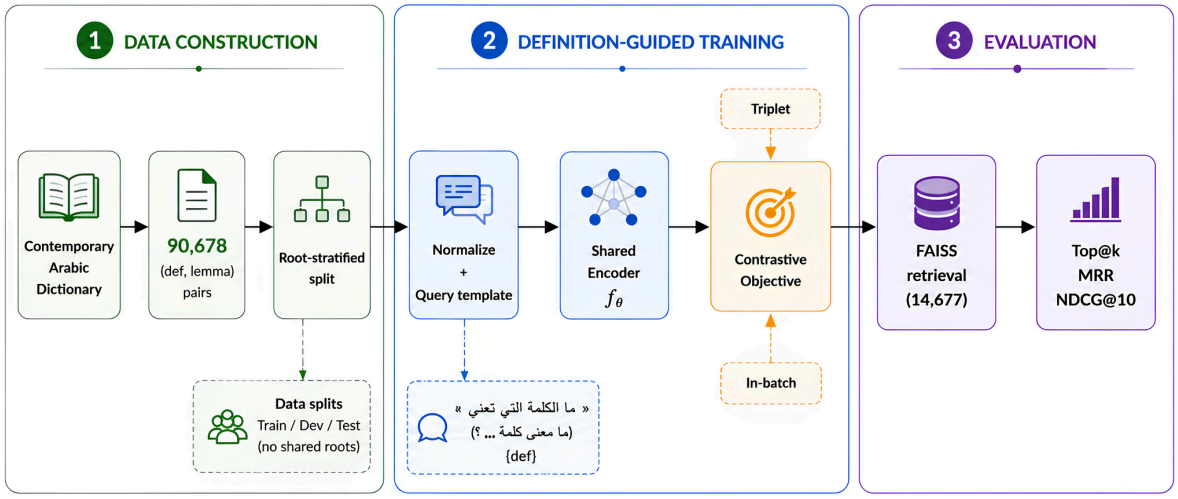


Figure 1: Overview of the proposed Definition-Guided Learning (DGL) method. The development set is sampled from the training pool and shares its root distribution; root disjointness applies only to the training/development pool versus the test partition (§3).

of 256, yielding 255 implicit negatives per instance substantially richer than the single hard negative used in the Triplet setting.

4.2 Models

We fine-tune the three strongest *open-weight* embedding models from the zero-shot evaluation (Section 5.1). The top-ranked proprietary model (OpenAI `text-embedding-3-small`) is excluded because it cannot be fine-tuned in our setting. The selected models are: **Matryoshka**, an Arabic Matryoshka representation-learning model (Nacar et al., 2025); **GATE**; and **E5-large** (Wang et al., 2024). All three are fine-tuned under both the Triplet Loss and In-Batch Negative Sampling objectives using identical hyperparameters to ensure a fair comparison.

4.3 Training Configuration

All experiments were conducted using a batch size of 256, providing 255 in-batch negatives per anchor, on a single NVIDIA A100 GPU. To ensure a controlled comparison across models and training objectives, all configurations were trained for five epochs using a learning rate of 5e-5 (the default value for `Sentence Transformer TrainingArguments` in the HuggingFace Transformers library, not explicitly overridden). sOptimisation was performed using AdamW (Loshchilov and Hutter, 2019) with a linear warmup ratio of 0.1. Training employed BF16 mixed precision, and for the Matryoshka model, the embedding dimensionality was set to 768. A fixed random seed of 42 was applied consistently across `random`, `numpy`, and `torch` for reproducibility; reported results correspond to a single run per configuration (single seed). Fine-tuning used `sentence-transformers` version 5.6.0. The Matryoshka checkpoint corresponds to the HuggingFace repository `Omartificial-Intelligence-Space/Arabic-Triplet-Matryoshka-V2`(commit `b581354`), fine-tuned from `aubmindlab/bert-base-arabertv02` (AraBERT v0.2; 135,193,344 parameters); we cite this specific checkpoint via (Nacar et al., 2025), distinguishing it from the general Matryoshka Representation Learning method (Kusupati et al., 2022). At inference, queries and candidate lemmas are encoded with the fine-tuned model and retrieved with FAISS (Johnson et al., 2019) using inner-product (cosine) similarity over the candidate pool of 14,677 test-partition lemmas.

5 Results

The evaluation is conducted using standard information retrieval metrics, namely **Top@k**, **Mean Reciprocal Rank (MRR)**, and **Normalized Discounted Cumulative Gain (NDCG@10)**. Top@k measures the proportion of queries for which the correct lemma appears within the **Top@k** retrieved candidates, capturing retrieval performance at different cut-off levels. **MRR** evaluates the average reciprocal rank of the first correct answer, reflecting how early the correct result appears in the

ranked list. **NDCG@10** assesses ranking quality by assigning higher importance to correct answers appearing at top positions, making it particularly sensitive to the ordering of retrieved candidates.

5.1 Zero-Shot Baseline Results

We first evaluate six pre-trained embedding models together with a BM25 lexical baseline (Robertson and Zaragoza, 2009) in a zero-shot setting, without any task specific fine-tuning, to establish the performance ceiling of off the shelf retrieval on Arabic definition-to-word matching. Besides the three models carried forward to fine-tuning (Matryoshka, GATE, and E5-large; Section 4.2), the pool includes the proprietary OpenAI `text-embedding-3-small`, `jina-embeddings-v5-text-nano` (Jina AI, 2024), and ArabGlossBERT (Al-Hajj and Jarrar, 2021). Results are shown in Table 2.

We observe that performance is uniformly low across all models and metrics. The strongest embedding model, Matryoshka, achieves a Top@1 of 11.16% and an NDCG@10 of 24.45%. The BM25 lexical baseline attains a closely comparable Top@1 (10.13%), reflecting cases where a definition shares surface tokens with its target lemma; however, BM25 is clearly surpassed by Matryoshka at every deeper cut-off (e.g. Top@10: 36.75% vs. 12.99%) and on ranking quality (NDCG@10: 24.45% vs. 11.96%). This suggests that even the best-performing general-purpose embedding model is unable to reliably retrieve the correct lemma, confirming a substantial performance gap between pre-trained representations and the requirements of this task. These results indicate that Arabic definition-to-word retrieval cannot be adequately addressed by zero-shot application of existing embedding models, regardless of their general-purpose capabilities. Notably, OpenAI text-embedding-3-small, a state-of-the-art proprietary model, ranks fifth among the evaluated embedding models (Top@1: 4.26%), behind Matryoshka, GATE, E5-large, and Jina, suggesting that scale alone does not compensate for the absence of task-specific training on Arabic lexical data.

5.2 Fine-Tuning Results: Triplet Loss

Following the zero-shot evaluation, we fine-tune the top three models Matryoshka, GATE, and E5-large under the Triplet Loss objective with hard negatives generated by E5-large. Results are reported in Table 3.Fine-tuning with Triplet Loss yields a substantial improvement over the zero-shot baseline for most models and metrics, most notably Top@1 and MRR; for E5-large, however, deeper-cutoff metrics do not improve (Top@10: 26.68% zero-shot vs. 24.75% with Triplet Loss), indicating that the gains are not uniform across ranking depth. Matryoshka achieves a Top@1 of 30.21% and an NDCG@10 of 35.09%, a relative gain of roughly 2.7× over its zero-shot counterpart (11.16%). These results demonstrate that task-specific contrastive training with explicit hard negatives provides a strong inductive signal for Arabic lexical retrieval. We observe that all three models converge to similar performance levels under this objective, with Matryoshka and GATE separated by less than 0.3 percentage points in Top@1 (30.21% vs. 29.96%). This convergence suggests that the Triplet Loss objective imposes a training ceiling that is not strongly dependent on model architecture, potentially due to the limited diversity of hard negatives introduced by a fixed negative sampling strategy.

Fine-Tuning Results with Triplet Loss: Impact of Refined Hard Negatives To improve the construction of hard negative samples for Arabic semantic retrieval, we introduce a structured multi-stage pipeline that integrates neural retrieval, large language model (LLM) based selection, and linguistically informed filtering. We begin by leveraging pretrained embedding models to retrieve a pool of semantically similar candidates for each query–positive pair. This retrieval step ensures that candidate negatives lie in a dense semantic neighborhood, increasing the likelihood of generating challenging and informative contrasts. To refine candidate selection, we employ a GPT-4 based model as a semantic adjudicator guided by a carefully designed prompt enforcing strict lexical and semantic constraints. These include excluding synonyms, paraphrases, and semantically entailed alternatives, while encouraging the selection of contextually plausible yet incorrect candidates. To further mitigate false negatives, we incorporate multiple linguistically informed filtering layers: synonym filtering using external lexical resources, derivative filtering to remove morphologically related forms, and definition-level similarity filtering based on embedding similarity between lexical definitions. To assess the impact of this refinement, we conducted a manual qualitative review of a random sample of 50 hard negatives,

Table 2: Zero-shot retrieval results. Across diverse model families, no system exceeds 11.16% Top@1. The strongest embedding model, Matryoshka, leads across all six metrics, including a narrow margin over the lexical BM25 baseline on Top@1 (11.16% vs. 10.13%), underscoring the persistent gap between off-the-shelf representations and the requirements of lexicographic definition-to-word retrieval.

Category	Model	Top@1	Top@3	Top@5	Top@10	MRR	NDCG@10
Sparse	BM25	10.13%	11.95%	12.62%	12.99%	11.14%	11.96%
Arabic-specific	Matryoshka	**11.16%**	**24.40%**	**29.79%**	**36.75%**	**19.15%**	**24.45%**
	GATE	10.21%	23.79%	29.03%	35.42%	18.23%	23.45%
Multilingual	E5-large	9.74%	15.42%	19.84%	26.68%	14.00%	17.77%
	Jina	4.69%	10.09%	14.28%	20.00%	8.62%	12.00%
Closed-source	OpenAI	4.26%	8.18%	10.91%	15.61%	7.17%	9.65%
Lexicon-oriented	ArabGlossBERT	2.76%	4.86%	6.37%	8.65%	4.29%	5.61%

Table 3: Fine-Tuning Results: Triplet Loss with Hard Negatives vs. Refined Hard Negatives.

Setting	Model	Top@1	Top@3	Top@5	Top@10	MRR	NDCG@10
Triplet Loss	Matryoshka	30.21%	34.94%	36.93%	39.66%	33.14%	35.09%
	GATE	29.96%	34.76%	36.67%	39.80%	32.95%	35.00%
	E5-large	17.97%	21.25%	22.77%	24.75%	20.04%	21.37%
Refined Hard Negatives	Matryoshka	16.26%	30.12%	31.66%	33.01%	23.32%	25.76%
	GATE	15.14%	30.04%	31.71%	33.03%	22.73%	25.32%
	E5-large	17.55%	26.23%	28.75%	31.03%	22.27%	24.41%

covering both retrieved-candidate and fallback-generated cases; the selected negatives were qualitatively judged, in this limited manual sample, to be higher quality i.e., semantically close but clearly incorrect compared to negatives from our earlier, pre-refinement pipeline. We evaluate the impact of these refined hard negatives on downstream retrieval performance, and the results are presented in Table 3. Despite the improved linguistic quality of the constructed negatives confirmed by this manual review, the performance gains remain limited. As shown in Table 3, the results indicate that enhancing the precision and linguistic validity of hard negatives does not necessarily lead to improved retrieval performance.Refined hard negatives substantially reduce performance for Matryoshka and GATE. For E5-large, Top@1 decreases slightly, while the deeper-cutoff and ranking metrics improve. We had initially hypothesised that this may stem from a reduction in the diversity of training signals introduced by our filtering pipeline. However, our comparison only contrasts the unfiltered and fully filtered endpoints, and cannot distinguish gradual diversity loss across filtering steps from the effect of a single dominant filtering component (e.g., retrieval, synonym filtering, root-based filtering, or GPT-4 adjudication). We therefore do not treat the diversity-based explanation as an established finding, and instead report the observed performance drop as an unexplained empirical observation; a step-wise ablation isolating each filtering component is left for future work. In contrast, in-batch negatives, despite being noisier and less controlled, achieve stronger and more robust performance in our experiments (Table 4), which we similarly report as an empirical observation without a fully verified causal explanation.

5.3 Fine-Tuning Results: In-Batch Negative Sampling

We next evaluate the same three models fine-tuned under the In-Batch Negative Sampling objective. Results are reported in Table 4. In-Batch Negative Sampling yields the strongest results across all configurations. Matryoshka achieves a Top@1 of 41.13% and an NDCG@10 of 49.32%, a roughly 3.7-fold improvement over the zero-shot baseline and a clear gain over the Triplet Loss configuration. These results establish in-batch negatives as the preferred training objective for this task. We observe a consistent and clear advantage of In-Batch Negatives over Triplet Loss across all three models and all six evaluation metrics. The improvement is particularly pronounced in Top@1 (+10.92% points for Matryoshka) and NDCG@10 (+14.23% points), suggesting that denser negative

signals not only improve exact retrieval precision but also enhance the overall ranking quality of retrieved candidates. We additionally report a hybrid configuration (Matryoshka + BM25) that linearly combines the fine-tuned dense scores with lexical BM25 scores over the same candidate pool. BM25 scores were computed using the `rank_bm25` Python library (`BM25Okapi`) with the standard default parameters $k_1 = 1.5$ and $b = 0.75$, adopted as-is without tuning on our data; Arabic text was tokenised via the same diacritic-removal and letter-normalisation preprocessing used elsewhere in our pipeline, followed by whitespace tokenisation. Both score sets are normalised and fused as $\alpha\, s_{\text{dense}} + (1 - \alpha)\, s_{\text{BM25}}$ with $\alpha = 0.7$, a fixed heuristic weighting favouring the dense signal and not tuned on a held-out development set. Despite augmenting dense retrieval with an explicit lexical signal, the hybrid underperforms the purely dense Matryoshka model (Top@1: 32.64% vs. 41.13%), indicating that after fine-tuning the dense representations already subsume the lexical overlap BM25 captures, while BM25's noisier signal dilutes the ranking. All results reported above use the test-only candidate pool (14,677 lemmas, duplicates retained) at true batch size 256. Full sensitivity results across batch sizes and candidate-pool constructions are reported in Table 6 (Appendix E).

6 Linguistic Feature Analysis

To better understand model behaviour, we examine how retrieval performance relates to three linguistic properties: definition length (the token count of the normalised gloss), word sense count, and lexical difficulty (a 1–5 score assigned by GPT-5). Using the predictions of our selected Matryoshka configuration (fine-tuned with In-Batch Negatives), we train a Random Forest Regressor to predict each retrieval metric from these features and report the resulting importance scores across all six metrics (Figure 4).Definition length is the most informative feature for exact retrieval, with the highest importance at Top@1 (44.3%), though its weight declines as k increases (35.5% at Top@10). Lexical difficulty shows the opposite trend: it is least important for Top@1 (27.4%) but rises steadily with k, overtaking definition length to become the dominant feature at Top@10 (36.4%) and remaining strong for NDCG@10 (33.9%). This crossover is informative: definition length largely governs whether the correct word is retrieved at the very top, whereas lexical difficulty more strongly shapes performance over the wider candidate ranking captured by Top@10 and NDCG@10. Word sense count contributes least and most uniformly across metrics (26.9% - 29.6%), suggesting that, within our evaluation setting, polysemy has a comparatively minor effect relative to the other two features.

7 Discussion

The results presented above provide strong evidence that lexical definitions constitute a rich and effective supervisory signal for training semantically grounded Arabic embedding models. The roughly 3.7-fold improvement in Top@1 accuracy from 11.16% in the zero-shot setting to 41.13% nder our selected fine-tuned configuration demonstrates that the semantic content encoded in dictionary glosses is sufficiently structured and discriminative to guide contrastive learning toward meaningful lexical representations.This improvement, however, is not uniform across training strategies, and the choice of contrastive objective proves decisive. The consistent superiority of In-Batch Negative Sampling over Triplet Loss across all three models and all six evaluation metrics warrants careful interpretation. The performance gap, ranging from +10.92 to +17.03 Top@1 percentage points, is not marginal. We attribute it primarily to a difference in the quantity and diversity of the contrastive signal available during training, though we note that these two factors are entangled in our comparison and cannot be fully separated. While Triplet Loss restricts each training instance to a single hard negative, In-Batch Negative Sampling exposes the model to 255 implicit negatives per instance at a batch size of 256, drawn from diverse semantic domains within the same batch. This density and diversity of negative signals plausibly enables the model to learn finer-grained distinctions across the lexical space, not just between an anchor and one hard negative, but across a broad neighborhood of semantically proximate lemmas simultaneously. Notably, this runs somewhat counter to the standard hard-negative mining intuition that harder, semantically closer negatives are most informative: in our setting, the broader and noisier in-batch signal outperformed a single curated hard negative, which makes the advantage of diversity a non-trivial finding rather than a foregone conclusion. This finding carries a practical implication

Table 4: Fine-tuning results with In-Batch Negative Sampling. Matryoshka + BM25 is the hybrid dense lexical configuration ($\alpha = 0.7$).

Model	Top@1	Top@3	Top@5	Top@10	MRR	NDCG@10
Matryoshka	41.13%	47.67%	51.93%	57.40%	45.65%	49.32%
GATE	41.28%	47.76%	51.86%	57.27%	45.73%	49.33%
E5-large	35.00%	40.66%	43.77%	48.21%	38.72%	41.65%
Matryoshka + BM25	32.64%	40.88%	45.02%	50.41%	37.86%	40.85%

for future work: since the quality of in-batch negatives is directly tied to batch composition, batches containing semantically diverse lemmas are likely to generate more informative implicit negatives than batches dominated by lemmas from the same semantic domain or root family, suggesting that diversity-aware batch construction strategies could yield further gains.

8 Error Analysis

We analyse 100 test instances for which the fine-tuned retrieval model failed to recover the correct lemma within the Top@10 retrieved candidates (Top@1 = Top@3 = Top@5 = Top@10 = False for all cases). Manual analysis of the Top@1 prediction score for each failure case reveals that these errors are not uniform in nature, but instead fall along a continuum of residual semantic relatedness. At the high end of this continuum (score $\geq$ 65%, the model retrieves a lemma that is a near-perfect semantic synonym of the gold lemma, frequently a case where the dictionary's own definition of the gold lemma consists primarily of the predicted word itself, as in (اصطباب) ((انسكاب)) being predicted as (انسكاب) with a score of 92%. These are arguably not retrieval failures in a strict semantic sense, but rather artefacts of an evaluation protocol that recognises only a single gold lemma per definition despite genuine synonymy in the lexicon; in several of these cases the predicted and gold lemmas additionally share the same morphological pattern (wazn), which a strict exact-match evaluation penalises even though both forms are valid and semantically interchangeable. At the intermediate range (score 45%–65%), the model retrieves a lemma from the same semantic field as the gold lemma typically a hypernym, a co-hyponym, or a component concept embedded within the gold definition suggesting that the model falls back on the closest available semantic neighbour when it cannot resolve the fine-grained distinction required for exact retrieval, as seen with (استنقاط) ((طلب وضع النقط في كتاب)) being predicted as (نقط). At the lowest end of the continuum (score < 45), the model's top prediction bears little discernible semantic relationship to the gold lemma; these cases most closely resemble genuine retrieval failures and are concentrated among definitions describing rare, culturally specific, or highly compositional concepts such as (صخريات) , a specialised term for a breed of war camels that is likely underrepresented in the training distribution. Representative examples from each band are presented in Table 5. Taken together, this gradient illustrates a key limitation of strict automatic evaluation. Near-Synonym and Related-Concept cases make up most of the failures in this 100-sample analysis, while Weak-Match cases are the minority. Most of what automatic metrics count as "wrong" are thus near misses valid synonymy or morphological near identity a human would accept rather than true breakdowns in semantic association, a distinction strict Top@k metrics cannot capture. This motivated the human evaluation in the next Section. To assess whether fine-tuning altered the nature of these residual errors, we traced the same failure cases back through the zero-shot model's predictions. Zero-shot scores in the Near-Synonym band were already fairly high (0.69–0.84), and fine-tuning mostly pushed confidence further toward the same near-synonym rather than resolving the underlying single-gold-lemma limitation. In several cases, however, fine-tuning changed the model's top prediction altogether: for the definition of (كأس) (النبيذ), the zero-shot model retrieved (محتسيه), whereas the fine-tuned model retrieved (جعه) instead. This points to a shift in the model's semantic reasoning rather than a simple boost in confidence. Overall, DGL appears to pull residual errors closer to genuine semantic neighbours, but it does not remove this class of near-miss errors entirely.

9 Ablation Study

To disentangle these factors, we ran a batch-size sensitivity analysis on the Matryoshka model using true batch sizes of 32, 64, 128, and 256 (up to 255 real in-batch negatives per anchor), all trained with CachedMultipleNegativesRankingLoss to keep the loss mechanism identical across the sweep. Results show an overall improvement, with minor non-monotonic fluctuations, across all metrics on both the full and test-only corpora as batch size increases (full-corpus NDCG@10: 40.43% $\rightarrow$ 41.47%; test-only NDCG@10: 48.35% $\rightarrow$ 49.32%, for batch sizes 32$\rightarrow$256). Full results are reported in Table 6 in Appendix E.

10 Human Evaluation

To complement the automatic metrics, we conducted a production-oriented human evaluation using free-form, user-authored queries. Four PhD holders in Arabic, balanced by gender, wrote definitions in their own words without being restricted to the dictionary's phrasing. All 160 queries were evaluated using our selected configuration (Matryoshka fine-tuned with in-batch negative sampling), retrieving the top five candidates per query. Two Arabic-language evaluators, neither of whom authored the queries, independently judged each candidate as a valid synonym or not, yielding 800 binary judgments. They agreed on 657 (82.1%; 95% query cluster bootstrap CI: 79.8–84.5; Cohen's $\kappa = 0.629$, 95% CI: 0.576–0.679). The 143 disagreements, spanning 103 queries, were resolved by a third adjudicator (90 valid, 53 invalid), yielding a final reference of 502 valid and 298 invalid judgments. Based on this reference, micro-averaged Accuracy@1, @3, and @5 (Hit@k) were 75.0% (120/160; 95% Wilson CI: 67.8–81.1), 88.8% (142/160; 82.9–92.8), and 93.8% (150/160; 88.9–96.6). This Accuracy@1 is not directly comparable to the 41.13% obtained on the held-out test set: the human protocol accepts any valid synonym rather than a single gold lemma, a stricter exact-match criterion that our error analysis (Appendix C) shows penalises many valid near-synonym predictions. The consistent increase from rank 1 to rank 5 nevertheless shows that the correct lemma is frequently retrieved among top candidates even when not top-ranked, suggesting the learned embedding space places correct answers close to naturally phrased queries. As a supplementary check, GPT-5.6-Sol agreed with the human reference on 602 of 800 judgments (75.3%; 95% CI: 71.0–79.3; $\kappa = 0.527$, 95% CI: 0.455–0.598), with high precision (96.6%) but lower recall (62.7%), suggesting its scores are a conservative lower bound. These results were not used to replace human judgments or compute the primary reported results.

11 Conclusion

This work introduced Definition-Guided Learning (DGL), a contrastive framework that uses dictionary definitions as a supervisory signal for Arabic semantic representation learning, evaluated through reverse dictionary retrieval on a root-stratified benchmark we contribute as a methodological resource. Fine-tuning with in-batch negative sampling raised Top@1 accuracy from an 11.16% zero-shot baseline to 41.13%, with retrieval performance shaped by definition length, polysemy, and lexical difficulty. Our error analysis and human evaluation further show that many apparent failures are in fact valid synonyms penalised by strict exact-match scoring, underscoring the value of dictionary definitions as a rich, underexploited resource for Arabic NLP and motivating future work on relaxed evaluation protocols and broader lexicographic applications beyond reverse dictionary retrieval.

Limitations

Our evaluation relies on a single MSA lexicographic resource, and we cannot yet confirm that our approach generalises to other Arabic dictionaries, which may differ in root-family distributions or lexical coverage, or to dialectal Arabic. Lexical difficulty scores were assigned using GPT-5 rather than human linguists, which may introduce systematic biases not present in fully human-annotated difficulty judgements. Our human evaluation, while informative, involved a limited pool of four query authors and a set of 160 free-form queries, which may not fully capture the diversity of real-world user phrasing; while inter-annotator agreement on the candidate-level validity judgments was substantial (Cohen's $\kappa = 0.629$ before adjudication), each query itself was authored by a single annotator, so query-generation biases specific to this small pool cannot be fully ruled out.All annotators involved in the human evaluation (query authors, candidate evaluators, and the adjudicator) were volunteers who received no monetary compensation for their

participation. No formal institutional ethics board (IRB) approval was sought, as the study involved no collection of sensitive personal information, no vulnerable populations, and no risk beyond that of routine linguistic annotation. Finally, our framework was evaluated solely on reverse dictionary retrieval; while we motivate Definition-Guided Learning as a general approach to semantic representation learning, its effectiveness on other downstream tasks, such as word sense disambiguation or definition generation, has not been empirically verified in this work.

References

Moustafa Al-Hajj and Mustafa Jarrar. 2021. ArabGloss-BERT: Fine-tuning BERT on context-gloss pairs for WSD. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 35–43, Held Online. INCOMA Ltd.

Rawan Al-Matham, Waad Alshammari, Abdulrahman AlOsaimy, Sarah Alhumoud, Asma Wazrah, Afrah Altamimi, Halah Alharbi, and Abdullah Alaifi. 2023. Ksaa-rd shared task: Arabic reverse dictionary. In *Proceedings of ArabicNLP 2023*, pages 450–460.

Abdulaziz M Alayba. 2025. Arabic natural language processing (nlp): A comprehensive review of challenges, techniques, and emerging trends. *Computers*, 14(11):497.

Waad Alshammari, Amal Almazrua, Asma Al Wazrah, Rawan Almatham, Muneera Alhoshan, and Abdulrahman AlOsaimy. 2024. Ksaa-cad shared task: Contemporary arabic dictionary for reverse dictionary and word sense disambiguation. In *Proceedings of the Second Arabic Natural Language Processing Conference*, pages 677–685.

Sami Boudelaa, Noha Fathi, and Sameh Al Ansary. 2026. Novel dynamics in semantic ambiguity processing: insights from a semitic language. *Journal of Memory and Language*, 146:104709.

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.

Mohammed El-Razzaz, Mohamed Waleed Fakhr, and Fahima A Maghraby. 2021. Arabic gloss wsd using bert. *Applied Sciences*, 11(6):2567.

Ahmed ElBakry, Mohamed Gabr, Muhammad N El-Nokrashy, and Badr AlKhamissi. 2023. Rosetta stone at ksaa-rd shared task: A hop from language modeling to word–definition alignment. In *Proceedings of ArabicNLP 2023*, pages 477–482.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 6894–6910.

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

Jina AI. 2024. Jina embeddings: Multilingual text embedding models. https://jina.ai/embeddings/.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, and Ali Farhadi. 2022. Matryoshka representation learning. *Advances in Neural Information Processing Systems*, 35:30233–30249.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*.

Yash Mahajan, Matthew Freestone, Naman Bansal, Sathyanarayanan N. Aakur, and Santu Karmaker. 2025. Revisiting word embeddings in the LLM era. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 2686–2717, Mumbai, India. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics.

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*.

Attia Nehar, Djelloul Ziadi, and Hadda Cherroun. 2016. Rational kernels for arabic root extraction and text classification. *Journal of King Saud University-Computer and Information Sciences*, 28(2):157–169.

OpenAI. 2024. New embedding models and API updates. https://openai.com/index/new-embedding-models-and-api-updates/.

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 (EMNLP-IJCNLP)*, pages 3982–3992.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389.

Serry Sibaee, Samar Ahmad, Ibrahim Khurfan, Vian Sabeeh, Ahmed Bahaaulddin, Hanan Belhaj, and Abdullah Alharbi. 2023. Qamosy at arabic reverse dictionary shared task: Semi decoder architecture for reverse dictionary with sbert encoder. In *Proceedings of ArabicNLP 2023*, pages 467–471.

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.

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*.

A Lexical Difficulty Annotation Prompt

To assign a difficulty score to each lemma in our dataset, we queried GPT-5, accessed using the structured prompt shown in Figure 2. Words were submitted in batches of 10 per call, in a fixed order, and the model was instructed to return exactly one numeric score per word in the same order, without any accompanying explanation, to enable automatic and unambiguous parsing of the output.

```
Task: Assign a difficulty level (1–5) to each Arabic word in Modern Standard Arabic.

Consider ALL factors:
- Frequency
- Age of acquisition
- Concreteness
- Word length
- Morphological complexity
- Semantic transparency

Scale:
1 Very easy
2 Easy
3 Moderate
4 Difficult
5 Very difficult

Strict rules:
Return ONLY numbers
One number per line
Same order
No explanations
```

Figure 2: Structured prompt used to assign lexical difficulty scores via GPT-5.

Example. Given the input words(طاولة),(عين) and (استنباط) , the model returned the sequence 2, 1, 4 corresponding, in order, to a common polysemous noun, a basic everyday object, and a morphologically complex derived form, respectively.

Validation. We manually verified the difficulty assignments on a random sample of words and compared them against human judgments.

B Hard Negative Selection Prompt

After retrieving and filtering candidate negatives as described in Section 5.2 (E5-large retrieval followed by synonym/root-based filtering), GPT-4 was prompted to select a single hard negative from the remaining candidates for each anchor definition, using the prompt shown in Figure 3.

```
You are an Arabic lexicography judge for hard negative mining.

Choose ONE hard but valid negative.

Rules:
- Prefer selecting from the retrieved candidates.
- The hard negative must be semantically close, but NOT a synonym and NOT a correct answer.
- Avoid words sharing the same stem/root if they preserve the same meaning.
- Avoid easy or unrelated words.
- If all candidates are weak or invalid, generate a semantically related Arabic lexical candidate.
It may later be mapped back to the nearest corpus word.
```

Figure 3: Prompt used to select a hard negative from filtered E5-large candidates via GPT-4, for Triplet Loss training.

Example. For the anchor definition corresponding to the gold lemma (انسكاب)(اصطباب), "spilling"), after filtering out the synonym (انسكاب) itself and root-sharing forms, GPT-4 selected (تسرب) ("leakage") as a semantically related but non-synonymous and incorrect hard negative.

Fallback generation. In cases where none of the filtered candidates satisfied the difficulty and validity criteria, GPT-4 generated a new Arabic lexical candidate directly, which was subsequently mapped to its nearest match in the corpus vocabulary or discarded if no sufficiently close match existed.

C Error Analysis

Table 5: Representative failure cases across the error-severity spectrum (fine-tuned Matryoshka). Most failures are near-synonyms or related concepts that a single-gold, exact-match protocol penalises despite genuine semantic validity.

Band	Definition (query)	Gold lemma	Top@1 prediction	Score
Near-Synonym	انسكاب	اصطباب	انسكاب	0.927
	تحويل وتبديل	اقلاب	تحويل	0.812
	استهانه واستخفاف	استبساط	استخفاف	0.796
Related-Concept	طلب وضع النقط في كتاب او غيره	استنقاط	نقط	0.644
	اسرع اشتعالا واتقادا	اورى	اشظى	0.642
	لباس مناسبه خاص كالتتويج او التوليه	خلعه	زي	0.641
Weak-Match	النبيذ	كاس	جعه	0.399
	نوع خفيف من النعال يستعمل في بلاد المغرب	بلغه	شلجم	0.381
	فصيله من الابل تهاجم العدو عند سماع صوت الرصاص	صخريات	هنوف	0.368

D Linguistic Feature Analysis

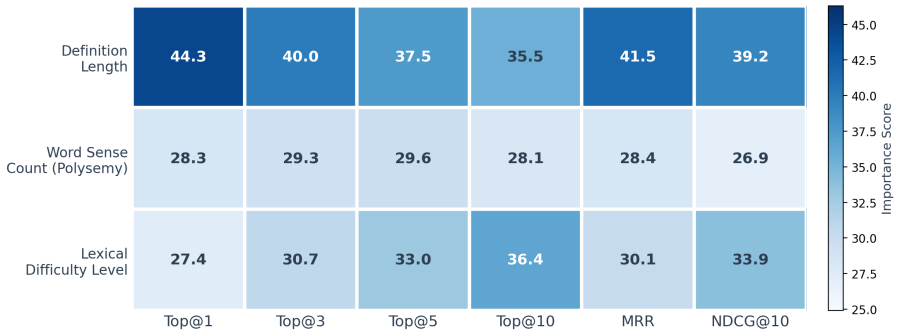


Figure 4: Feature importance heatmap across all six retrieval metrics.

E Batch-Size and Corpus Sensitivity Analysis

Table 6 reports full results across all tested batch sizes and both candidate-pool constructions, for all three fine-tuned models.

Table 6: Sensitivity of fine-tuning results to true batch size (32–256, i.e., up to 255 in-batch negatives per anchor) and candidate-pool construction (test-only, 14,677 lemmas with duplicates retained, vs. deduplicated combined train+test pool, 62,664 lemmas). All models trained with CachedMultipleNegativesRankingLoss to keep the loss mechanism identical across the sweep. Batch size 256 / test-only corresponds to the headline results in Table 4.

Batch Size	Model	Corpus	Top@1	Top@3	Top@5	Top@10	MRR	NDCG@10
32	GATE-AraBert	Test-only	40.08%	46.92%	50.83%	56.28%	44.67%	48.32%
		Full	32.58%	40.52%	44.10%	48.96%	37.54%	40.26%
	Matryoshka-V2	Test-only	40.35%	47.09%	50.71%	56.14%	44.79%	48.35%
		Full	33.14%	40.57%	44.04%	48.82%	37.82%	40.43%
	E5-large	Test-only	32.24%	37.32%	39.90%	43.59%	35.48%	37.96%
		Full	27.93%	33.37%	35.75%	38.69%	31.27%	33.04%
64	GATE-AraBert	Test-only	40.60%	47.60%	51.72%	56.94%	45.21%	48.92%
		Full	33.02%	40.96%	44.58%	49.79%	38.03%	40.82%
	Matryoshka-V2	Test-only	40.74%	47.61%	51.73%	57.06%	45.31%	49.01%
		Full	33.10%	41.02%	44.79%	49.80%	38.13%	40.91%
	E5-large	Test-only	32.81%	38.16%	41.19%	45.02%	36.34%	39.01%
		Full	28.81%	33.98%	36.75%	40.16%	32.18%	34.07%
128	GATE-AraBert	Test-only	41.02%	47.85%	51.90%	57.12%	45.57%	49.21%
		Full	33.71%	41.13%	44.83%	49.91%	38.46%	41.18%
	Matryoshka-V2	Test-only	41.12%	47.79%	52.01%	57.25%	45.64%	49.27%
		Full	33.66%	41.21%	44.85%	50.00%	38.49%	41.23%
	E5-large	Test-only	34.20%	39.72%	42.87%	47.03%	37.86%	40.69%
		Full	29.40%	35.54%	38.24%	41.95%	33.18%	35.27%
256	GATE-AraBert	Test-only	**41.28%**	**47.76%**	**51.86%**	**57.27%**	**45.73%**	**49.33%**
		Full	33.81%	41.58%	45.31%	50.30%	38.71%	41.47%
	Matryoshka-V2	Test-only	**41.13%**	**47.67%**	**51.93%**	**57.40%**	**45.65%**	**49.32%**
		Full	**33.86%**	**41.53%**	**45.36%**	**50.28%**	**38.73%**	**41.47%**
	E5-large	Test-only	**35.00%**	**40.66%**	**43.77%**	**48.21%**	**38.72%**	**41.65%**
		Full	**29.86%**	**36.10%**	**39.20%**	**42.99%**	**33.81%**	**36.00%**

Real-Time Quranic Recitation Assessment on Mobile Devices Using Streaming Inference and Model Compression

Abdelrahman Ayman Amin Mohamed **Abdullah Haytham Mohamed Salah Hedeya**
Faculty of Engineering, Cairo University
Giza, Egypt
abdelrahman.akefafy@gmail.com, hedeya@gmail.com

Abstract

Automatic Quranic recitation assessment requires accurate detection of phonetic and Tajweed-related pronunciation errors while providing low-latency feedback. However, most speech recognition models are designed for offline processing and are too large for deployment on consumer mobile devices. We present a real-time Quranic recitation assessment system based on a multi-head Wav2Vec2-BERT model trained to jointly predict phonemes and Tajweed *Sifat* attributes using Connectionist Temporal Classification. To enable streaming inference without retraining, we introduce an **Adaptive Eager-Emission** framework that combines heuristic output commitment, dynamic context expansion, and overlap recovery, reducing perceived latency while preserving long-range temporal features. For efficient on-device deployment, we propose a structured compression pipeline based on encoder pruning, recovery fine-tuning, and INT8 quantization with ONNX optimization, reducing a 600M-parameter, 2.4 GB model to a 24.5M-parameter, 36 MB model while maintaining recognition performance. The streaming framework consistently outperforms static sliding-window approaches across the accuracy–latency trade-off, achieving responsiveness as low as 1.0 s at roughly 2.1% phoneme error rate (PER), or near-offline performance below 1% PER at 6.0 s average chunk latency, while the production streaming configuration maintains 99.45% average *Sifat* attribute accuracy. On a mid-range Android CPU, the compressed model runs at a real-time factor of 0.225 with a peak memory of about 600 MB.

1 Introduction

Correct Quranic recitation is governed by *Tajweed*, a system of articulation rules that determines how each letter is pronounced, elongated, nasalized, or merged with its neighbors. Mastering Tajweed traditionally requires a qualified human teacher who listens to a learner and corrects pronunciation in real time. Teacher availability, cost, and geographic access limit how often learners can practice with immediate feedback. An automatic assessor that detects *phoneme-level* and *articulatory* errors and responds instantly could substantially broaden access to high-quality recitation practice.

Unlike generic Arabic automatic speech recognition (ASR), Quranic recitation assessment is best framed as a structured sequence-recognition problem in which orthography, elongation patterns, and letter attributes jointly define correctness. A useful assessor must recognize not only *what* was said but also *how* it was said, characterizing articulatory attributes (*Sifat*) at the phoneme level. Our underlying model addresses this with a multi-head Wav2Vec2-BERT encoder (Seamless Communication et al., 2023) trained with Connectionist Temporal Classification (CTC) (Graves et al., 2006) to predict a phoneme stream and ten parallel *Sifat* attribute streams (Abdelfattah et al., 2025).

Deploying such a model for live, on-device feedback exposes two obstacles. First, strong speech encoders are trained on *complete* utterances with full bidirectional context; used in a streaming setting, they lose the right-side context they rely on and produce systematic errors near every chunk boundary. Second, these encoders are large: our baseline occupies 2.4 GB, far beyond a practical mobile budget, and offloading inference to the cloud sacrifices privacy, compute, and offline availability. These two obstacles are the focus of this paper.

We make the following contributions:

- We introduce **Adaptive Eager-Emission**, a streaming inference framework that converts an offline full-utterance CTC model into a low-latency streaming recognizer *without retraining*. It combines dual commit heuristics, dynamic context expansion, and seam-based overlap recovery, adapting window size to

Proceedings of The Fourth Arabic Natural Language Processing Conference (ArabicNLP 2026), pages 433–443
October 28-29, 2026 ©2026 Association for Computational Linguistics

what each segment requires and guaranteeing a deterministic worst-case latency bound.

- We design a **structured compression pipeline**—four-axis encoder pruning, plain-CTC recovery fine-tuning, and INT8 ONNX export—that reduces the model from 600M parameters / 2.4 GB to 24.5M parameters / 36 MB while preserving recognition and *Sifat* accuracy.
- We provide an **empirical characterization** of the accuracy–latency trade-off and the on-device compute cost, showing effective latencies from 1.0 s responsiveness ($\sim$2.1% PER) to 6.0 s ACL ($<$1% PER), a real-time factor of 0.225, and $\sim$600 MB peak memory on a mid-range Android CPU.

Together these results show that accurate, low-latency, privacy-preserving Quranic recitation assessment can run entirely on-device.

2 Related Work

Quranic and Arabic recitation assessment. Most deployed Quran ASR systems target verse identification rather than pronunciation assessment. Commercial systems like Tarteel target verse identification and flag missing words, though their foundational dataset (Khan et al., 2021) lacks Tajweed annotations, meaning they do not assess Tajweed attributes at the phoneme level. More broadly, prior systems tend either to identify verses or to detect a narrow set of Tajweed phenomena, rather than characterizing articulatory attributes per phoneme, so comprehensive phoneme-level *Sifat* assessment remains rare. Our model builds on a multi-level CTC formulation that jointly predicts phonemes and *Sifat* (Abdelfattah et al., 2025); this paper focuses on making that formulation *streaming* and *on-device*.

Streaming end-to-end ASR. CTC (Graves et al., 2006) enables frame-synchronous, non-autoregressive decoding that is naturally suited to streaming, in contrast to autoregressive attention decoders (Chiu et al., 2018). Self-supervised encoders such as wav2vec 2.0 (Baevski et al., 2020), HuBERT (Hsu et al., 2021), and Conformer-based models (Gulati et al., 2020; Seamless Communication et al., 2023) achieve strong representations but are trained over complete utterances, and Flynn and Ragni (Flynn and Ragni, 2023) showed empirically that such models benefit from up to 80 s of acoustic context—acoustic context length is therefore a first-class performance variable, not merely a computational constraint. The dominant path to streaming addresses this architecturally: encoders are retrained with look-ahead masking (Zhang et al., 2020) or sliding-window attention; Moonshine v2 (Kudlur et al., 2026) exemplifies this approach, achieving bounded time-to-first-token on edge devices by replacing global attention with an ergodic windowed encoder. An orthogonal strategy adapts the inference procedure alone, leaving the trained model unchanged. Liu et al. (Liu et al., 2020) showed that a full-sequence model can operate incrementally by committing only the stable prefix of each chunk's output, deferring boundary-uncertain tokens until the next chunk confirms them. Our framework shares this no-retraining philosophy but inverts the commitment strategy: rather than withholding uncertain tokens, it expands the processing window to acquire the missing acoustic context—a design motivated by Quranic elongations whose temporal extent cannot be reduced without phonetic error.

Model compression for speech encoders. Structured pruning removes whole architectural units—attention heads (Michel et al., 2019), feed-forward neurons, layers, and hidden dimensions (Xia et al., 2022; Ma et al., 2023)—yielding dense models that run efficiently on commodity hardware, unlike unstructured sparsity (Sanh et al., 2020; Frankle and Carbin, 2019). Depth reduction is especially effective because deep layers are often redundant (Gromov et al., 2025; Men et al., 2025). For self-supervised speech models specifically, joint distillation and pruning (Peng et al., 2023) and structured pruning for speaker tasks (Li et al., 2025; Han et al., 2025b,a) have been explored. We compose structured pruning with low-precision INT8 inference and report a negative result on knowledge distillation (Hinton et al., 2015) under aggressive pruning.

3 Background: Multi-Head CTC Model

The assessor is built on `w2v-bert-2.0` (Seamless Communication et al., 2023), a Conformer encoder (Gulati et al., 2020) that augments Transformer self-attention (Vaswani et al., 2017) with convolution, with 24 layers, hidden dimension 1024, 16 attention heads, and feed-forward dimension 4096, plus one adapter layer, totaling roughly 600M parameters. The encoder feeds **11 parallel CTC heads**: one

phoneme head and ten *Sifat* heads, each predicting an articulatory attribute (e.g., *hams/jahr* voicing, *qalqalah*, *ghunnah*, *itbaq*). Predicting phonemes and attributes from a shared encoder state avoids the latency and training cost of a sequential dependency, and CTC removes the need for frame-level alignment (Graves et al., 2006).

The frontend emits one CTC frame approximately every 40 ms (a 400-sample analysis window at 16 kHz with a 160-sample hop, followed by two stride-2 downsampling stages). Streaming paths use greedy CTC decoding to minimize latency. The reference phoneme sequence each recitation is scored against is produced by a rule-based component of the broader system (Abdelfattah et al., 2025) and is outside the scope of this paper.

4 Adaptive Eager-Emission Streaming

4.1 Problem

The encoder was trained on complete utterances with full bidirectional context. In streaming mode, phonemes near the right edge of each processed chunk lack future context and are systematically misclassified. Naively shrinking the chunk to reduce latency makes this worse (almost every token lacks context), while enlarging it restores accuracy at the cost of responsiveness. For Quranic recitation, this tension is further sharpened by obligatory vowel elongations (*Madd*) spanning two to six phonetic beats: any fixed window narrow enough for responsive feedback is too narrow to capture a full elongation, while one wide enough for *Madd* passages adds unacceptable latency to short segments. No single window size resolves this. The framework addresses it by expanding context *only when heuristics detect that more evidence is needed* and by committing output *eagerly* once it is safe.

4.2 Four-Region Adaptive Window

Each inference operates on a window with four regions (Table 1): a *left context* of committed audio that restores the encoder's convolutional history; a dynamically sized *chunk* that is the only region allowed to emit tokens; a *right look-ahead* of real future audio used purely as encoder context; and a short *right pad* of zeros that softens convolutional boundary artifacts. The look-ahead is the single most impactful parameter because it supplies the future context that boundary phonemes require; its ablation effect is reported in Section 6.6. A decoder wrapper restricts token emission to the chunk region; frames in the look-ahead and pad regions contribute to encoder hidden states but are never decoded into output.

Region	Duration	Role
Left context	1.0 s	encoder history
Chunk	1.5–7.5 s	emission target
Look-ahead	1.0 s	future context only
Right pad	0.05 s	boundary smoothing

Table 1: The four-region streaming window for the production configuration. Only the chunk region emits tokens; the look-ahead provides the future context that boundary phonemes require.

4.3 Dual Commit Heuristics

After decoding, two heuristics decide whether the current output can be committed or whether the window must expand:

- **H2 (trailing-run detection), evaluated first.** If the decoded sequence ends in a run of ≥ 2 identical tokens, a vowel elongation is active. Committing prematurely would truncate it, so the window expands. H2 is evaluated first, and the elongation state persists across forced commits: a *Madd* truncated at the expansion-budget boundary is detected in the immediately following chunk rather than treated as a new utterance.
- **H1 (edge-peak detection), evaluated second.** If the last token's CTC probability peak falls within 120 ms of the chunk's right boundary (3 CTC frames at 40 ms per frame), its acoustic evidence is truncated and the model lacks the future context for a confident decision, so the window expands.

The expansion budget constrains the total context an elongation may acquire: with increment e and maximum m expansions, a budget of $m \cdot e \geq 3.0$ s is sufficient to span a worst-case six-beat *Madd* at mujawad tempo. If either heuristic fires and the budget is not exhausted, the system appends e seconds of audio and re-runs inference on the wider window; otherwise it commits. This makes worst-case latency deterministic and bounded:

$$L_{\max} = b + m \cdot e + \ell, \tag{1}$$

where b is the base chunk, m the maximum expansions, e the expansion increment, and ℓ the look-ahead. For our production configuration (b=1.5 s, e=1.5 s, m=4, ℓ=1.0 s) this gives $L_{\max} = 8.5$ s.

4.4 Seam Recovery

Because each new window re-processes left-context audio, the encoder naturally re-predicts phonemes it already emitted, which would prepend duplicates to every chunk. On each commit we re-decode the last $\sim$400 ms of the left context, find the longest suffix of already-emitted tokens that matches a prefix of the re-decoded overlap, strip the duplicates, and append only genuinely new boundary tokens. The procedure is deliberately conservative so that it removes confirmed duplicates without discarding the continuation tokens of an active elongation.

4.5 Incremental Feedback

Waiting for a full phrase boundary before showing any feedback would frequently exceed the 2–5 s "dead zone" beyond which users perceive the system as frozen. After every commit the system immediately updates the displayed result for any word whose PER, computed by Levenshtein alignment (Levenshtein, 1966) of the hypothesis to each word's reference phoneme span, crosses a quality threshold. A per-word lock guards against runaway error cascades: if a word's PER exceeds 0.75, progression freezes at that word rather than propagating incorrect phonemes. A lightweight phrase classifier scores four mutually exclusive hypotheses—continuation, repetition, next verse, and noise—from the phoneme evidence and advances the session position accordingly, handling the non-linear recitation patterns common among learners.

5 On-Device Compression Pipeline

5.1 Why Pruning Is Necessary

The baseline occupies 2.4 GB in FP32. Converting to FP16 yields $\sim$1.2 GB and INT8 yields $\sim$600 MB—neither approaches a practical mobile budget of about 100 MB. The parameter count itself must therefore be reduced before quantization. The pipeline has three sequential stages: structured pruning, recovery fine-tuning, and INT8 ONNX export.

5.2 Four-Axis Structured Pruning

We use structured (not unstructured) pruning so the result stays dense and runs efficiently on standard mobile processors (Sanh et al., 2020; Xia et al., 2022). Pruning operates on four architectural axes:

- **Attention heads**, scored by the Frobenius norm of their query/key/value projections or by a knowledge-distillation-guided first-order Taylor importance (Michel et al., 2019; Ma et al., 2023), reducing the head count from 16 to 4.
- **Feed-forward neurons**, scored by the product of their up- and down-projection norms; because FFN blocks dominate the parameter budget, this contributes the largest share of compression, reducing the feed-forward dimension from 4096 to 1024.
- **Encoder layers**, scored by the mean input–output cosine similarity over a calibration set; near-identity layers are redundant and removed, reducing depth from 24 to 16 (Gromov et al., 2025; Men et al., 2025).
- **Hidden dimension**, scored by mean activation L_2-norm, reduced from 1024 to 256 subject to divisibility by the retained head count (per-head dimension 64).

Because the hidden dimension is shared across hundreds of tensors, pruning it requires coordinated, constraint-checked modification of every affected projection, normalization, convolution, and adapter weight.

5.3 Recovery Fine-Tuning

Pruning disrupts learned representations, so the compressed model is retrained with the same multi-head CTC objective used originally,

$$\mathcal{L}_{\text{CTC}} = \sum_{i=1}^{N} \lambda_i \, \mathcal{L}_i, \tag{2}$$

where $\mathcal{L}_i$ and λ_i are the loss and weight of head i. We evaluated knowledge-distillation-guided recovery, matching per-layer hidden states to the unpruned teacher (Hinton et al., 2015). During *width-only* pruning this helped, consistent with prior work on speech SSL encoders (Peng et al., 2023). However, once layers and the hidden dimension were also pruned, distillation *hurt* recovery: the aggressively pruned student converges to a representation structurally different from the teacher's, so a per-layer matching loss pulls it toward an unreachable target. Even a projection layer reconciling the dimension mismatch did not close the gap. This matches reports that depth-pruned models recover better through light fine-tuning than distillation (Gromov et al., 2025; Han et al., 2025a).

We therefore use plain CTC fine-tuning for all final configurations. The pruned model was fine-tuned for 20 epochs using a learning rate of 5×10^{-5} and a per-device batch size of 16 for both training and evaluation, without gradient checkpointing. The random seed was fixed to 42 for reproducibility.

5.4 INT8 ONNX Export and Deployment

The fine-tuned model is exported to ONNX, a framework-independent representation executed by ONNX Runtime, and quantized to INT8 during export. We find that this direct conversion preserves the fine-tuned model's accuracy, so no additional quantization-adaptation training is required (Section 6.3). We verify numerical agreement against the FP32 reference model, then apply runtime graph optimizations and operator fusion for the target CPU. The final artifact—ONNX model plus configuration—reduces the network from 605.8M parameters / 2.4 GB to 24.5M parameters / **36 MB** (configuration h4/f1024/l16/d256, INT8).

6 Experiments

6.1 Setup

Datasets. Compression evaluation uses **QDAT Bench** (Abdelfattah et al., 2025) (159 samples, ∼24.4 minutes) with ground-truth phoneme transcripts and per-phoneme *Sifat* annotations, and a held-out **three-Moshaf out-of-distribution** test set (Moshaf IDs 19.0, 29.0, and 30.0 from `obadx/muaalem-annotated-v3`). For recovery fine-tuning, the training split comprised 22 distinct Moshaf IDs from the same corpus, with 10% of the training data reserved for validation. Streaming evaluation uses a **golden streaming set** of 1,646 samples, also drawn from `obadx/muaalem-annotated-v3`. This set was manually curated by a domain expert from professional, clean-environment recitations of flawless quality to serve as a rigorous reference for Tajweed assessment. The code used to generate this streaming dataset is publicly available in our repository. Across these datasets, reference phoneme sequences are provided as pre-computed labels derived from the deterministic Quran Phonetic Script (QPS) (Abdelfattah et al., 2025), which translates Uthmani orthography directly into internally consistent phonetic targets. Recovery fine-tuning ran on a single NVIDIA L4 GPU for approximately 30 hours. The streaming configuration sweep (comprising 17 adaptive and 31 static configurations evaluated across 1,646 samples) required approximately a full day of compute in total: the FP32 baseline sweep was executed on an NVIDIA RTX 3050 GPU, while the compressed INT8 ONNX sweep was executed on a 12th Gen Intel Core i5-12450H CPU.

Implementation Details. Model training and evaluation were implemented in PyTorch, utilizing the Transformers library to manage the Wav2Vec2-BERT backbone. During recovery fine-tuning, we applied dynamic acoustic augmentation to improve robustness using Audiomentations. This augmentation comprised Gaussian noise insertion (0.1–1.5% amplitude, $p = 0.4$), time stretching (0.8–1.5× rate, $p = 0.4$), and gain transitions (-6 to $+6$ dB, $p = 0.4$). The complete software environment, including all package versions (e.g., ONNX Runtime, Datasets) and evaluation utilities, is fully documented in the `requirements.txt` file within our code repository.

Baselines. The FP32 model we compress and stream (Table 5) is the external state-of-the-art architecture of Abdelfattah et al. (2025), not an ablation of our own model; we adopt it as the starting point for our compression and streaming pipelines. We are not aware of another architecture that supports direct comparison under our task formulation. Standard Quranic ASR applications (such as Tarteel) operate at the word or verse level, and foundational academic datasets in this domain (Khan et al., 2021) lack Tajweed annotations. Because they do not perform joint phoneme and *Sifat* prediction, they cannot be evaluated against our phoneme- and attribute-level metrics. Abdelfattah et al. (2025) is, to our knowledge, the only existing architecture that jointly predicts phonemes and all ten *Sifat* attributes, which is why we select it as our sole baseline and starting checkpoint.

Evaluation protocol. We evaluate two windowing regimes. *Static* evaluation (the baseline) sweeps fixed (w, l, r) triples—chunk width, left context, and right look-ahead—over sixteen target latencies, producing the comparison curves against which adaptive windowing is assessed. *Adaptive* evaluation sweeps seventeen named configurations across five tiers (Table 2), each parameterised by (b, e, m, ℓ).

Both regimes report *inclusive* PER ($\overline{\text{PER}}_{\text{inc}}$)—Levenshtein distance averaged over *all* samples, including those exceeding a 15% PER thresh-

Tier	Label	Resp. (s)	WCL (s)
1	Real-time	≤ 1.0	≤ 4.0
2	Near-real-time	1.0–1.5	≤ 5.5
3	Balanced	~ 2.0	≤ 8.5
4	Accuracy-first	~ 3.0	≤ 15.0
5	Offline-like	~ 3.0	≤ 25.5

Table 2: Configuration tiers. Responsiveness $= b + \ell$; WCL $= b + m \cdot e + \ell$ (Eq. 1). All tiers maintain expansion budget $m \cdot e \geq 3.0$ s to absorb a worst-case six-beat *Madd* at mujawad tempo.

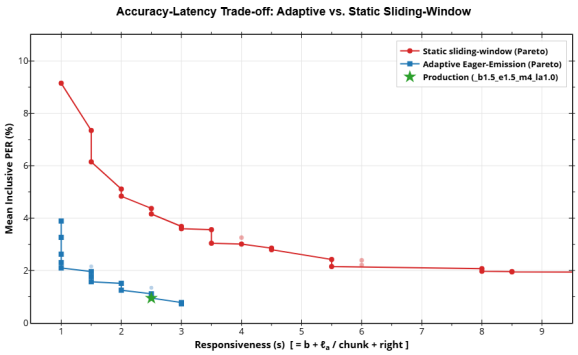


Figure 1: Accuracy–latency trade-off. Each point is one configuration. Pareto frontiers are drawn through the lowest PER at each responsiveness level. The adaptive framework (blue) dominates static sliding-window decoding (red) at every operating point. The production configuration is marked $\star$.

old. Reporting inclusive PER prevents aggressive low-latency configurations from gaining an unfair advantage by concentrating failures in an excluded tail. We additionally track worst-case latency ($L_{\max}$, Eq. 1), mean actual chunk latency (ACL $= \bar{L}_i + \ell$, where $\bar{L}_i$ is the mean committed chunk duration), and real-time factor (RTF). Confidence intervals use 10,000 bootstrap iterations.

6.2 Streaming Accuracy–Latency Trade-off

Figure 1 plots inclusive mean PER against responsiveness ($b + \ell$ for adaptive; chunk + right for static) for all 17 adaptive and 31 static configurations evaluated on the 1,646-sample golden streaming set. The adaptive Pareto frontier lies strictly below the static one at every responsiveness level: at 1.0 s responsiveness, the best adaptive configuration (`rt_b0.5_e1.5_m2`) achieves 2.10% PER versus 9.50% for the corresponding static window—a 4.5× reduction. At 2.5 s the production configuration (`bal_b1.5_e1.5_m4_la1.0`, marked $\star$) achieves 0.95% versus 2.60% for static, a 2.7× improvement. These gains arise because the adaptive framework acquires additional context precisely at the Madd boundaries that static windows truncate.

The configurations trace a Pareto frontier from a responsiveness of **1.0 s at ∼2.1% PER** (WCL $=$ 4.0 s, ACL $\approx$ 3.0 s) to near-offline **<1% PER at 6.0 s** ACL, dominating static sliding-window decoding at every operating point (Table 3). The production configuration is selected by the rule: among all configurations within 2% absolute $\overline{\text{PER}}_{\text{inc}}$ of the best-performing configuration, choose the one with minimum $L_{\max}$; this yields `bal_b1.5_e1.5_m4_la1.0` (Tier 3, $L_{\max}$=8.5 s, ACL ≈6.0 s). This configuration also maintains 99.45% average *Sifat* attribute accuracy across all 1,646 streaming test samples, showing that low-latency streaming does not come at the cost of attribute-level recognition quality. Across this sweep, RTF stays below 0.15 for all viable configurations when measured on the development server/desktop hardware used to run the large-scale configuration sweep; §6.5 reports substantially higher RTF measured natively on-device, reflecting the difference in hardware rather than a change in model or configuration. Figure 2 shows the PER distributions across representative tiers; the median is 0% on more than half the samples, confirming that errors concentrate at Madd boundaries.

Approach	Latency	Accuracy
Short chunks (0.5 s)	Fixed 1 s	9.50% mean PER
Long chunks (3.0 s)	Fixed 3.5 s	2.57% mean PER
Adaptive (ours)	2.5–8.5 s	0.95% mean PER

Table 3: Comparison of windowing strategies: Static window ($left\_context = 1.0\,s, chunk = X\,s, right\_context = 0.5\,s$). Adaptive windowing expands context only when the heuristics require it, dominating fixed-window decoding at every operating point. Latency range shown is responsiveness to $L_{\max}$ for the production configuration.

Finally, we evaluate the 36 MB INT8 compressed model under the streaming regime to ensure the offline compression results translate to live inference. Under the production configuration (`bal_b1.5_e1.5_m4_la1.0`), the compressed model achieves an inclusive PER of 1.40%. This represents a slight, expected degradation of just 0.45 percentage points compared to the 0.95% PER of the uncompressed FP32 baseline. This demonstrates that the compressed model trades a marginal acoustic edge for its massive 67× size reduction and remains highly robust for live on-device deployment.

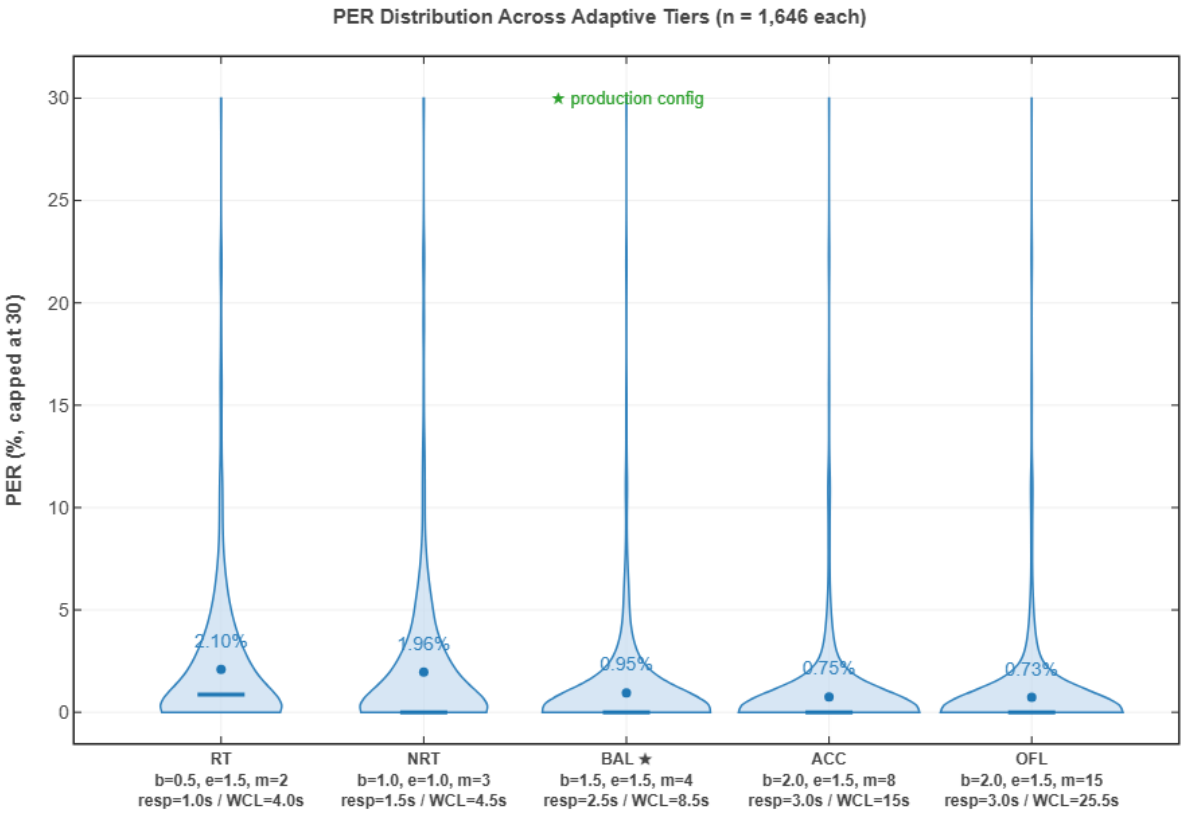


Figure 2: PER distributions for one representative adaptive configuration per tier (n = 1,646 samples each, violin body capped at 30%). Mean PER (shown above each violin) falls monotonically from Tier 1 (RT) to Tier 5 (OFL). The heavy median line at 0% visible in Tiers 3–5 confirms that errors concentrate in the long tail of difficult Madd passages rather than affecting typical segments. The production configuration (BAL ⋆) sits in Tier 3.

6.3 Compression, Stage by Stage

Table 5 reports QDAT Bench across the pipeline, evaluated in offline, full-utterance mode rather than under the streaming configuration of Section 6.2, so that accuracy figures isolate the effect of compression alone. 6 percentage points absolute.

The external Abdelfattah et al. (2025) FP32 model, adopted as our baseline (see Baselines paragraph, §6.1), attains 2.43% PER at 2.4 GB. After four-axis pruning and recovery fine-tuning, the 98 MB FP32 model reaches 7.27% PER with Tajweed F1 0.736 and accuracy 82.6%; INT8 ONNX export then produces the final 36 MB model at **8.05% PER** with Tajweed F1 0.700 and accuracy 78.8%—a 67× size reduction. The INT8 export closely tracks the fine-tuned FP32 model's accuracy without any quantization-adaptation training. On the cleaner out-of-distribution Moshaf set the compressed INT8 model holds **0.80% PER** (versus 0.16% for the FP32 model), indicating that the QDAT Bench gap largely reflects that benchmark's noisier single-verse clips rather than a true loss of capability. The pruned-before-fine-tuning model collapses (PER = 100%) and is fully restored by recovery training, underscoring the necessity of Section 5.3.

Table 4 breaks this down by phoneme recognition and each individual *Sifat* attribute on the out-of-distribution Moshaf set: every one of the ten attribute heads retains at least 98.68% accuracy after compression (worst case: Tafkheem/Taqeeq at 1.32% error), and the average error across phoneme recognition and all ten attributes rises only from 0.16% (FP32) to 0.8% (compressed INT8). This confirms that the headline PER increase observed on QDAT Bench (Table 5) is concentrated in that benchmark's noisier single-verse clips, rather than in the *Sifat* attribute predictions that the recitation-assessment task depends on.

6.4 Robustness to Acoustic Degradation

The offline comparisons above use clean recordings. To test whether the FP32–INT8 gap stays

Metric	FP32 Err.	INT8 Err.
Phoneme Error Rate (PER)	0.54%	1.25%
Hams / Jahr	0.12%	0.70%
Shidda / Rakhawa	0.17%	1.12%
Tafkheem / Taqeeq	0.17%	1.32%
Itbaq	0.09%	0.56%
Safeer	0.13%	0.59%
Qalqla	0.09%	0.72%
Tikraar	0.09%	0.77%
Tafashie	0.16%	0.48%
Istitala	0.08%	0.56%
Ghonna	0.13%	0.78%
Average	**0.16%**	**0.8%**

Table 4: Phoneme and per-*Sifat*-attribute error rates on the out-of-distribution Moshaf set, offline mode, before (Abdelfattah et al. (2025) FP32 baseline) and after compression (INT8 ONNX). Every attribute retains at least 98.68% accuracy post-compression; the average error across all eleven metrics rises from 0.16% to 0.8%.

Stage	Size	PER
Abdelfattah et al. (2025) (FP32)	2,400 MB	2.43%
4-axis pruned + FT (FP32)	98 MB	7.27%
INT8 ONNX export	36 MB	8.05%

Table 5: Compression stages on QDAT Bench, starting from the external Abdelfattah et al. (2025) FP32 baseline. INT8 ONNX export closely preserves the fine-tuned model's accuracy; pruning plus recovery drives the size reduction.

near its clean-condition value under realistic acoustic conditions, we evaluated both models on a 500-audio subset of the out-of-distribution Moshaf set (evaluating the full three-Moshaf set under every degradation condition was not computationally feasible) after applying controlled degradation: Speech-Shaped Noise (SSN), babble noise, and reverberation, at matched SNRs. Because this is a smaller subset than the full Moshaf set used in Table 4, its clean-condition PER (0.08% FP32, 0.86% INT8) differs slightly from the full-set average reported there (0.16%, 0.80%); both are consistent with a sub-1-point FP32–INT8 gap on clean audio.

The FP32–INT8 gap does not stay near its clean-condition value of $+0.78$ points: it widens to **+4.31** points under moderate reverberation (RT60 = 0.6 s) and to **+6.17** points under babble noise at 10 dB SNR—both meeting or exceeding the 5.62-point gap observed on QDAT Bench—and reaches $+15.18$ points under severe babble noise at 0 dB. We report this as a finding rather than a caveat: INT8 quantization measurably degrades the model's robustness to acoustic noise, and this degradation activates under exactly the conditions—background noise and room reverberation—that a

Condition	FP32	INT8	Gap
clean	0.08%	0.86%	+0.78
ssn@20dB	0.11%	1.10%	+0.99
ssn@10dB	0.23%	2.93%	+2.70
ssn@0dB	3.76%	16.26%	+12.50
babble@20dB	0.10%	1.74%	+1.64
babble@10dB	0.69%	6.86%	+6.17
babble@0dB	13.10%	28.28%	+15.18
reverb (RT60=0.6s)	0.72%	5.03%	+4.31
reverb 0.6s + ssn@10dB	2.36%	10.88%	+8.52

Table 6: PER (%) under controlled acoustic degradation, 500-audio Moshaf subset. Gap is INT8 minus FP32, in percentage points.

phone-based tutor is likely to encounter outside a controlled recording environment. Structured pruning, recovery fine-tuning, and INT8 export preserve accuracy on clean audio, but the compressed model should not be assumed to inherit the FP32 model's noise robustness, and deployments in acoustically uncontrolled settings should budget for substantially higher error rates than the clean-audio numbers in Table 4 suggest.

6.5 On-Device Profiling

On a Samsung Galaxy S22 (Exynos 2200) CPU the compressed model runs fully offline at **RTF 0.225** with a peak memory of about **600 MB**, comfortably faster than real time for live feedback. This figure is measured natively on mobile ARM hardware; the lower RTF ($<$0.15) reported in §6.2 was measured on the development server/desktop hardware used for the large-scale configuration sweep, so the two numbers reflect different evaluation hardware rather than a different model or configuration.

6.6 Analysis

Ablating the right look-ahead removes the 3–5$\times$ PER benefit of future context, confirming it as the single most impactful parameter for resolving boundary truncations. Furthermore, disabling seam recovery immediately reintroduces duplicate phonemes at every chunk boundary, validating its necessity for the eager-emission framework.

Expansion budget and elongation coverage. The compression pipeline materially alters the model's adaptive chunking behavior, reducing the firing rates of the edge-peak (H1) and trailing-run (H2) heuristics by 38.2% and 24.1%, respectively. To verify whether the $m \cdot e \geq 3.0$ s expansion budget still reliably covers a worst-case six-beat *Madd* despite these reduced triggers, we evaluated 1,785 reference 6-beat elongations across the dataset.

Counterintuitively, the compressed model's reduced sensitivity actively prevents it from colliding with the expansion cap. The highly sensitive uncompressed model frequently triggers these heuristics in sequence, exhausting the budget and forcing a hard truncation before the *Madd* finishes (a 4.9% H2 cap fault rate). Because the compressed model triggers H1 and H2 substantially less, it rarely exhausts the overall budget, allowing the decoder to close chunks normally and hitting the H2 cap in only 1.0% of cases. When isolating the causes of truncated 6-beat elongations on the compressed model, acoustic under-prediction (37.6%) vastly outweighs streaming budget truncations (1.0%). This confirms that the 3.0 s budget successfully guarantees the necessary physical window, and the primary source of error is an acoustic artifact of the compressed decoder rather than a failure of the latency bounds.

7 Conclusion

We showed that accurate, low-latency, privacy-preserving Quranic recitation assessment can run entirely on a consumer phone. A retraining-free streaming framework, Adaptive Eager-Emission, turns an offline multi-head CTC model into a responsive coach with a deterministic latency bound, and a composable compression pipeline shrinks the model 67× to 36 MB while preserving phoneme and *Sifat* accuracy. Figure 1 shows that the adaptive Pareto frontier uniformly dominates static sliding-window decoding: at 1.0 s responsiveness, adaptive achieves 2.10% PER versus 9.50% for the best static window at the same latency. Both techniques are general to CTC speech encoders. Future work includes additional *riwayat*, broader Tajweed coverage, and on-device personalization. Sensitivity analysis of the H1 and H2 threshold parameters, and validation of the H2 heuristic on *Ghunnah* nasalization as well as *Madd* elongation, remain open directions.

Limitations

Acoustic Robustness and Demographic Mismatch. The compressed model's robustness to acoustic degradation is a genuine limitation. Controlled experiments (§6.4) show that INT8 quantization widens the FP32–INT8 PER gap from $+0.78$ points (clean audio) to $+6.17$ points (10 dB SNR babble noise), approaching the 5.62-point gap observed on QDAT Bench. Therefore, the compressed model will likely underperform in noisy, real-world settings. Furthermore, QDAT Bench introduces a demographic mismatch: its recordings predominantly feature female voices, whereas our training data consists exclusively of male speakers. The impact of this demographic gap has not yet been fully disentangled from the aforementioned acoustic effects.

Streaming Assumptions and Heuristics. Because our streaming framework was evaluated exclusively against the golden streaming set's professional, flawless recitations, we assume idealized input. We have not formally tested the adaptive window's behavior on actual learner-grade audio, which typically contains mispronunciations, false starts, or extended pauses. Additionally, our H2 heuristic (trailing-run detection) was explicitly designed and validated for *Madd* elongations. We currently assume, but have not empirically verified, that it engages equally well for *Ghunnah* nasalization, which shares a multi-beat duration but differs acoustically. Threshold sensitivity optimization for these heuristics is left for future work.

Evaluation Scope and Operational Constraints. Systematically, the on-device real-time factor (RTF of 0.225) was measured in a controlled application environment, without accounting for concurrent background device loads or thermal throttling typical of extended mobile use. Operationally, while average chunk latency remains low, the balanced-tier configuration incurs a worst-case latency ($L_{\max}$) of 8.5 s, and accuracy-first configurations may reach 15.0 s, which could disrupt the user experience of instantaneous feedback during heavily elongated passages. Model-wise, while overall *Sifat* accuracy remains above 99%, certain attributes (e.g., *tafkheem* and *ghunnah*) exhibit comparatively higher error rates post-compression. Finally, all results are scoped strictly to a single recitation style (Hafs); generalization to other *qira'at* remains untested due to a lack of appropriately annotated datasets.

Data and Code Availability

To facilitate reproducibility and build upon our streaming framework, all original code, compression scripts, compressed model weights, and the generation pipeline for the golden streaming set are publicly available at https://github.com/Abdullah-Haytham/Streaming-and-Compression-Paper. We

release our code and compressed models under the MIT license.

Our system builds upon several existing artifacts, and our use of these assets is consistent with their intended research purposes. The foundational Wav2Vec2-BERT encoder (SeamlessM4T), the baseline FP32 model checkpoint, and the `obadx/muaalem-annotated-v3` dataset from Abdelfattah et al. (2025) were all accessed and utilized under the MIT license. Finally, the QDAT Bench evaluation set (Abdelfattah et al., 2025) is derived from the original QDAT dataset (Osman et al., 2021), which is publicly available for research purposes without an explicitly specified license.

Ethics Statement

Automated judgment of Quranic recitation is religiously sensitive. This system is designed strictly as a self-practice aid to support, not replace, qualified human teachers. It should not be used for formal assessment, certification, or gatekeeping decisions. Furthermore, as our training data consists exclusively of adult male voices reciting in the Hafs style, deploying this model for female learners, children, or practitioners of other *qira'at* carries uncharacterized performance risks; any real-world application must address these demographic biases to ensure equitable access. Additionally, running entirely on-device, the system keeps user audio local and does not transmit recordings, protecting the privacy of learners.

Use of AI Assistants

AI-assisted tools were used to improve the clarity, grammar, and readability of the manuscript, as well as to assist in code generation. All research ideas, methodology, experiments, final software implementation, analysis, results, and conclusions were directed, verified, and thoroughly reviewed by the authors.

References

Abdullah Abdelfattah, Mahmoud I. Khalil, and Hazem Abbas. 2025. Automatic pronunciation error detection and correction of the Holy Quran's learners using deep learning. *arXiv preprint arXiv:2509.00094*.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 12449–12460.

Chung-Cheng Chiu, Tara N. Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J. Weiss, Kanishka Rao, Ekaterina Gonina, Navdeep Jaitly, Bo Li, Jan Chorowski, and Michiel Bacchiani. 2018. State-of-the-art speech recognition with sequence-to-sequence models. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4774–4778, Calgary, Canada.

Rory Flynn and Anton Ragni. 2023. How much context does my attention-based ASR system need? *Preprint*, arXiv:2310.15672.

Jonathan Frankle and Michael Carbin. 2019. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations (ICLR)*, New Orleans, Louisiana, USA.

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 (ICML)*, pages 369–376, Pittsburgh, Pennsylvania, USA.

Andrey Gromov, Kushal Tirumala, Hassan Shapourian, Paolo Glorioso, and Daniel A. Roberts. 2025. The unreasonable ineffectiveness of the deeper layers. In *Proceedings of the International Conference on Learning Representations (ICLR)*, Singapore.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. 2020. Conformer: Convolution-augmented transformer for speech recognition. In *Proceedings of Interspeech*, pages 5036–5040, Shanghai, China.

Jiangyu Han, Federico Landini, Johan Rohdin, Anna Silnova, Mireia Diez, Jan Černocký, and Lukáš Burget. 2025a. Fine-tune before structured pruning: Towards compact and accurate self-supervised models for speaker diarization. In *Proceedings of Interspeech*, pages 1583–1587, Rotterdam, Netherlands.

Jiangyu Han, Petr Pálka, Marc Delcroix, Federico Landini, Johan Rohdin, Jan Černocký, and Lukáš Burget. 2025b. Efficient and generalizable speaker diarization via structured pruning of self-supervised models. *arXiv preprint arXiv:2506.18623*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

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.

Hamzah I. Khan, Abubakar Abid, Mohamed Medhat Moussa, and Anas Abou-Allaban. 2021. The Tarteel dataset: Crowd-sourced and labeled quranic recitation. In *Proceedings of the NeurIPS 2021 Track on Datasets and Benchmarks*.

Manjunath Kudlur, Evan King, Jiulong Wang, and Pete Warden. 2026. Moonshine v2: Ergodic streaming encoder ASR for latency-critical speech applications. *Preprint*, arXiv:2602.12241.

Vladimir I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710.

Ze Li, Ming Cheng, and Ming Li. 2025. Enhancing speaker verification with w2v-BERT 2.0 and knowledge distillation guided structured pruning. *arXiv preprint arXiv:2510.04213*.

Danni Liu, Gerasimos Spanakis, and Jan Niehues. 2020. Low-latency sequence-to-sequence speech recognition and translation by partial hypothesis selection. In *Interspeech*.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. Llm-pruner: On the structural pruning of large language models. In *Advances in Neural Information Processing Systems*, volume 36, pages 21702–21720.

Xin Men, Mingyu Xu, Qingyu Zhang, Qianhao Yuan, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. 2025. ShortGPT: Layers in large language models are more redundant than you expect. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20192–20204, Vienna, Austria.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, pages 14014–14024.

Hanaa Mohammed Osman, Ban Sharief Mustafa, and Yusra Faisal. 2021. Qdat: a data set for reciting the quran. *International Journal on Islamic Applications in Computer Science And Technology*, 9(1):1–9.

Yifan Peng, Yui Sudo, Shakeel Muhammad, and Shinji Watanabe. 2023. DPHuBERT: Joint distillation and pruning of self-supervised speech models. In *Proceedings of Interspeech*, pages 62–66, Dublin, Ireland.

Victor Sanh, Thomas Wolf, and Alexander M. Rush. 2020. Movement pruning: Adaptive sparsity by fine-tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 20378–20389.

Seamless Communication, Loïc Barrault, Yu-An Chung, and 1 others. 2023. SeamlessM4T: Massively multilingual and multimodal machine translation. *arXiv preprint arXiv:2308.11596*.

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 (NeurIPS)*, volume 30, pages 5998–6008.

Mengzhou Xia, Zexuan Zhong, and Danqi Chen. 2022. Structured pruning learns compact and accurate models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1513–1528, Dublin, Ireland.

Binbin Zhang, Di Wu, Zhuoyuan Yao, Xiong Wang, Fan Yu, Chao Yang, Liyong Guo, Yaguang Hu, Lei Xie, and Xin Lei. 2020. Unified streaming and non-streaming two-pass end-to-end model for speech recognition. *arXiv preprint arXiv:2012.05481*.

KASHAF: A Hierarchical Pipeline for Arabic Interpretable Semantic Textual Similarity

Fahad AlGhamdi[1] **Abdelati Hawwari**[2] **Abdulrahman AlOsaimy**[3]
Rawan Almatham[3] **Asma Al Wazrah**[3]
[1]Al-Baha University [2]Datalex4ai
[3]King Salman Global Academy for Arabic Language (KSAA)
fghamdu@bu.edu.sa abhawwari@gmail.com
{aalosaimy, ralmatham, aalwazrah}@ksaa.gov.sa

Abstract

We introduce KASHAF, to our knowledge the first interpretable Semantic Textual Similarity (iSTS) resource and end-to-end system for Arabic: a corpus of 7,811 human-annotated sentence pairs (inter-annotator κ=0.83 on a doubly-annotated subset) under a three-level hierarchical schema (sentence, phrase, and word), together with a modular pipeline that produces this layered analysis plus a composite similarity score, the KASHAF Hierarchical Similarity Score (KHSS), calibrated to the ordinal SEM scale. On a held-out cross-batch test split (1,002 pairs, disjoint from training and dev), KASHAF reaches Macro F1 = 0.659 ($\pm$0.016, 5 seeds) and Spearman ρ=0.6715; a rule-based antonym/negation polarity gate provides an empirical score-capping rule for high-similarity contradictions, and the MARBERTv2 backbone leads three alternative Arabic encoders. As an off-the-shelf diagnostic rather than a fine-tuned comparison, four contemporary LLMs under matched prompting reach at most macro-F1 0.215 (standard prompting) and 0.336 (chain-of-thought), a 32–44-point gap that reflects systematic over-prediction of `SEM-PARA` at the expense of the dominant `SEM-PARTIAL` class. We release the schema, guidelines, code, models, and a reproducibility pipeline; the corpus is available from KSAA on request.

1 Introduction

Standard Semantic Textual Similarity (STS) produces a single score that tells the user *how* similar two sentences are, but not *why*. Interpretable STS (iSTS) was introduced at SemEval-2015 (Agirre et al., 2015, 2016) to address this gap: each sentence pair is segmented into chunks, chunks are aligned across the pair, and each alignment is labeled with a relation type. The resulting three-level explanation supports downstream applications including educational feedback, MT evaluation, paraphrase mining, and legal/medical document comparison.

Despite a decade of iSTS work for English, Spanish, and a handful of other languages, **no published iSTS schema, corpus, or system exists for Arabic to our knowledge**. Arabic exhibits rich morphology, a non-concatenative templatic word formation system, free word order with strong information-structural preferences, and pervasive null subjects, all of which complicate alignment and tagging at fine granularity.

In this paper we introduce KASHAF (Arabic: كَشَّاف, *revealer*), to our knowledge the first iSTS resource and end-to-end system for Arabic. On a held-out cross-batch test set (1,002 pairs from two batches the system never saw during training or hyperparameter tuning), KASHAF achieves macro-F1 = 0.659 and Spearman ρ = 0.6715 for four-way SEM classification. As an off-the-shelf reference on the same split, four contemporary LLMs (GPT-4o, GPT-4o-mini, Claude Sonnet 4.5, Claude Haiku 4.5) under matched prompting reach at most macro-F1 = 0.215 (3-shot) and 0.336 (chain-of-thought), a 32–44-point gap, not a fine-tuned comparison, that reflects systematic over-prediction of `SEM-PARA` at the expense of the dominant `SEM-PARTIAL` class (71.7% of test data). This over-prediction is most severe under zero-shot and 3-shot prompting and only partially alleviated by chain-of-thought.

Contributions. (1) A hierarchical Arabic iSTS schema (4 SEM, 6 PHR, 7 WRD tags) with an explicit anchor-and-mobile alignment protocol, applied to 7,811 annotated pairs with κ=0.83 on SEM. (2) A modular pipeline combining MARBERTv2 (Abdul-Mageed et al., 2021) embeddings, LightGBM tag classifiers, a CRF-based BIO segmenter, and a polarity-aware calibration gate built on Arabic WordNet (Black et al., 2006) with morphological expansion over 13,329

Proceedings of The Fourth Arabic Natural Language Processing Conference (ArabicNLP 2026), pages 444–461
October 28-29, 2026 ©2026 Association for Computational Linguistics

AWN seed lemmas. (3) The KHSS composite score combining hierarchical evidence under an independent polarity check; we transparently analyse that the phrase/word terms add interpretation rather than predictive lift, and that the polarity gate's aggregate Spearman effect is small and not statistically significant on the test split ($\Delta\rho$=+0.0231, 95% CI $[-0.007, +0.053]$). (4) Controlled paired ablations showing (a) the systematic LLM `SEM-PARTIAL` $\rightarrow$ `SEM-PARA` collapse holds across all four models, and (b) we find *no significant evidence* that Stage 2 contrastive pre-training improves phrase or SEM classification ($p \in \{0.40, 0.80, 0.89\}$ on three core metrics, 3-seed paired t-test; non-significance under limited seeds is not proof of equivalence).

We release annotation guidelines (English and Arabic), source code, trained models, and a corpus sample under a non-commercial research license; the full corpus is available to researchers on request from the King Salman Global Academy for the Arabic Language (KSAA; Appendix K).

2 Related Work

Interpretable STS. The iSTS task was introduced by Agirre et al. (2015) and extended at SemEval-2016 (Agirre et al., 2016). Lopez-Gazpio et al. (2017) consolidated the framework, adding a verbalization module and two user studies showing that interpretable alignments help humans perform downstream judgments, establishing usability, not just intrinsic accuracy, as the goal of iSTS. Participating systems combined handcrafted chunk alignment with similarity features, and subsequent work pairs chunk alignment with classification and regression (Majumder et al., 2021). None target Arabic.

Arabic STS. Cer et al. (2017) included cross-lingual Arabic-English STS, but no published Arabic-only sentence-level iSTS schema or corpus exists to our knowledge. Abdul-Mageed et al. (2021) introduced MARBERTv2, a strong Arabic semantic encoder we build on; their work targets representation quality rather than fine-grained iSTS. For sentence-level similarity specifically, AraBERT (Antoun et al., 2020) established strong Modern Standard Arabic baselines, and more recent sentence-embedding models such as the Arabic Triplet Matryoshka encoders trained under the GATE framework (Nacar et al., 2025) report competitive STS scores; we include these, with CAMeLBERT (Inoue et al., 2021), as encoder baselines (Section 6). None, however, produces the interpretable phrase- and word-level trace that is our focus.

Word and phrase alignment. SimAlign (Jalili Sabet et al., 2020) produces word alignments from contextualized embeddings without parallel training data. AwesomeAlign (Dou and Neubig, 2021) adds light supervision. For morphologically rich languages, classical statistical aligners underperform where semantic precision matters. KASHAF builds on SimAlign and adds a discriminative LightGBM classifier.

LLMs for fine-grained similarity. Recent work on LLMs for STS reports strong coarse-benchmark results but behaviour that varies by setting: generative models can lead on knowledge-grounded similarity (Gatto et al., 2023), while fine-grained or linguistically conditioned variants remain harder (Tu et al., 2024). For Arabic, strong fine-tuned encoders such as MARBERTv2 (Abdul-Mageed et al., 2021) remain competitive with or superior to general-purpose LLMs on fine-grained understanding tasks; our work quantifies this gap for interpretable STS with a breakdown of LLM failure modes.

Arabic lexical resources. Arabic WordNet (Black et al., 2006) provides a lexical-semantic backbone; we augment it with morphological expansion ($\sim$30 surface forms per lemma over 13,329 seed lemmas) for antonym lookup at scale.

3 Schema and Dataset

KASHAF adopts a three-level hierarchical schema (Table 1) designed to give an interpretable trace of *why* two sentences are similar. A single SEM tag captures the overall semantic relation; phrase-level alignments locate *where* agreement and divergence occur; word-level tags surface the specific lexical and morphological alternations.

Schema tags. The four SEM tags, `SEM-IDENT` (identical, surface variation allowed), `SEM-PARA` (paraphrase), `SEM-PARTIAL` (partial overlap), and `SEM-DIFF` (different meaning or contradiction), map onto a 0–3 ordinal scale, exploited by our ordinal SEM head (Section 4). PHR has six tags: `MATCH`, `MOD` (same meaning, different surface), `SYN`, `REORDER`, `ADD_IN_B`, `DEL_IN_B`. WRD has five relation types: `MATCH`, `SYN`, `MORPH`, `FUNC`, `ANT`,

Level	Tag	Description
Sentence (SEM)	SEM-IDENT	Identical meaning (surface variation OK)
	SEM-PARA	Paraphrase: same meaning, different wording
	SEM-PARTIAL	Partial similarity: extra/missing info on one side
	SEM-DIFF	Different meaning or contradiction
Phrase (PHR)	PHR-MATCH	Lexical overlap, same content
	PHR-MOD	Same meaning, different surface form (morph/agreement)
	PHR-SYN	Synonym-substitution
	REORDER	Same content, different position in sentence
	PHR-ADD_IN_B	Phrase present only in B
	PHR-DEL_IN_B	Phrase present only in A
Word (WRD)	WRD-MATCH	Exact same word (or trivial inflection)
	WRD-SYN	Synonym
	WRD-MORPH	Morphological inflection of same root
	WRD-FUNC	Function-word change (preposition, determiner)
	WRD-ANT	Antonym
	WRD-ADD_IN_B	Word present only in B
	WRD-DEL_IN_B	Word present only in A

Table 1: The KASHAF hierarchical iSTS schema: 4 SEM tags, 6 PHR tags, 7 WRD tags.

plus directional ADD_IN_B/DEL_IN_B for words with no counterpart (mapped to SemEval-iSTS types in Appendix D).

3.1 Annotation Methodology

Anchor-and-mobile alignment. A crucial design choice in our schema is the *asymmetric* treatment of the two sentences during alignment (Figure 1). Sentence A is the **anchor**: its phrase and word positions are held fixed in canonical left-to-right order. Sentence B is the **mobile**: each B-phrase is realigned to sit directly beneath its best-matching counterpart in A, while B's *original* position is recorded. This eliminates ambiguity in defining REORDER (the tag fires when B's original position differs from its anchored position), yields a stable canonical form for downstream feature engineering, and mirrors how annotators naturally read the second sentence. The same principle applies at the word level inside each phrase pair.

Annotation procedure. Trained linguists annotate each pair following a 100+-page bilingual (Arabic–English) annotation guide (AlGhamdi et al., 2026), whose SEM-tag decision rules are summarised in Appendix C: (1) assign the SEM tag; (2) segment both sentences into phrases (clausal/PP/NP units); (3) realign B's phrases against A as anchor; (4) tag each aligned phrase pair (ADD/DEL for unaligned); (5) align words within each pair using A-words as anchor and tag.

Inter-annotator agreement (IAA). On a 781-pair gold subset double-annotated by two senior linguists (third resolving disagreements), we report

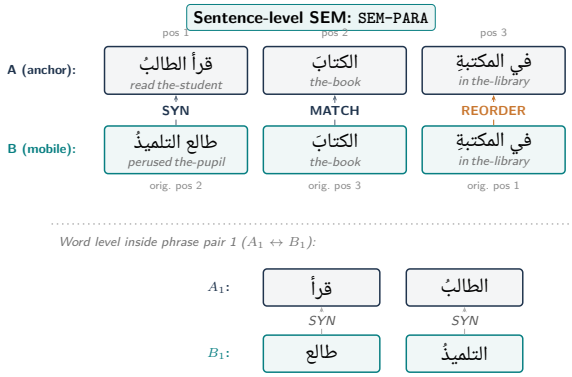


Figure 1: Three-level annotation of the SEM-PARA pair قرأ الطالبُ الكتابَ في المكتبةِ ("the student read the book in the library", A) and في المكتبةِ طالع التلميذُ الكتابَ ("in the library the pupil perused the book", B). A is the fixed *anchor*; each B-phrase is realigned beneath its best match and its original position is recorded, yielding SYN (synonyms قرأ/طالع, الطالب/التلميذ), MATCH, and REORDER (the fronted في المكتبة). The same anchor-and-mobile principle applies at the word level inside each phrase pair.

Cohen's κ for SEM and micro-F1 for PHR/WRD tag sequences (Artstein and Poesio, 2008). Our agreement: $\kappa = 0.83$ for SEM, $F_1 = 0.79$ (PHR), $F_1 = 0.77$ (WRD). By Landis and Koch (1977) thresholds, $\kappa \in [0.81, 1.0]$ is "almost perfect"; our SEM agreement falls in this range, consistent with the iSTS literature norm of $\kappa \in [0.70, 0.85]$ (Agirre et al., 2016).

Corpus and splits. We construct 7,811 pairs from a heterogeneous mix of public Arabic sources (movie and television subtitles (Lison and Tiedemann, 2016), product reviews, encyclopaedic arti-

Split	Pairs	Phrase tags*
Train (13 batches)	5,809	∼80,500
Dev (B13 + B14)	1,000	∼11,200
Test (B03 + B15)	1,002	11,145
Total annotated	7,811	∼102,800

Table 2: KASHAF corpus statistics. Cross-batch splits are designed so that test contains complete annotator-batches unseen during training or development; dev and test share no batches. Train uses 13 batches (B01–B02, B04–B12, B16–B17; B03 is held out for test, B13–B14 for dev, B15 for test). **Phrase tags* counts all annotated phrase-level decisions per pair, including aligned PHR-MATCH/MOD/SYN/REORDER and unilateral PHR-ADD_IN_B/PHR-DEL_IN_B; see Appendix A for per-tag breakdown.

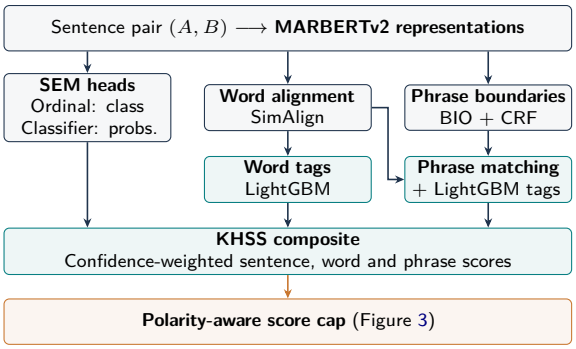


Figure 2: KASHAF overview: frozen SEM/alignment features, a fine-tuned boundary encoder, and separate class/probability heads. Polarity-gate conditions: Figure 3.

cles (Wikipedia), and scientific texts), augmented with curated paraphrase pools and adversarial near-paraphrase distractors. *Cross-batch* evaluation: train on the remaining 13 batches, dev on B13+B14 (1,000 pairs), test on B03+B15 (1,002 pairs). `SEM-PARTIAL` dominates (71.7%) followed by `SEM-PARA` (19.8%), `SEM-DIFF` (4.3%), `SEM-IDENT` (4.3%).

Data integrity audit. We programmatically audited the splits for leakage and release the audit script with the data. **No Pair_ID is shared between train, dev, and test** (0 duplicates by exact ID match). The cross-batch protocol assigns each annotator-batch to a single split, so the partition is disjoint at the annotation-unit level by construction. Beyond identifiers, after Arabic normalisation (diacritics, alif variants, *ta marbuta*, whitespace, punctuation) we verify that no test sentence pair recurs in the training or development data, ruling out verbatim pair leakage. All main results are therefore reported on the full n=1,002 held-out test split.

4 The KASHAF Pipeline

KASHAF is a modular pipeline of five trained components and one rule-based component, producing both a hierarchical interpretation and a scalar KHSS similarity score (Figure 2). Only the SEM level is *symmetric*: its features use symmetric operations (cosine, Jaccard, length-difference), so sentence-level predictions are invariant to swapping A and B. The phrase and word levels are *equivariant* rather than symmetric: the BIO segmenter and aligner apply to A and B with shared weights (phrase alignment via bipartite max-weight matching), but their tags are directional by annotation convention (e.g., `ADD_IN_B`/`DEL_IN_B` mark material present only in B).

MARBERTv2 backbone. We use MARBERTv2 (Abdul-Mageed et al., 2021), a 163M-parameter encoder pre-trained on Modern Standard Arabic and Egyptian/Levantine dialectal data, both as a frozen feature extractor (ordinal SEM, word/phrase alignment) and as a fine-tuned encoder (boundary detection). We empirically select layer-8 embeddings, which we found to provide the best balance of syntactic and semantic information for Arabic word alignment in preliminary experiments on the dev set.

Stage 2 contrastive pre-training (optional). We optionally fine-tune MARBERTv2 on 2,080 `SEM-IDENT` + `SEM-PARA` pairs using `MultipleNegativesRankingLoss` (Henderson et al., 2017) with 4 in-batch negatives. We ablate this step in Section 6.3: contrastive pre-training yields at most a marginal, non-significant gain (≈0.01 macro-F1), with vanilla MARBERTv2 performing nearly identically. The headline configuration uses the Stage-2 backbone (Table 3); the encoder, not this auxiliary step, drives sentence-level performance.

Boundary detector. A BIO (Begin–Inside–Outside) sequence-labeling head combining fine-tuned MARBERTv2 with a linear classifier and a Conditional Random Field (CRF) (Lafferty et al., 2001). We concatenate per-token mean of layers 6–12, giving the best dev F1. Training: 5 epochs, batch size 32, learning rate 2×10^{-5}, Adam (Kingma and Ba, 2015).

Word alignment. For each sentence pair, we compute layer-8 token embeddings and run SimAlign (Jalili Sabet et al., 2020) (`inter` matching) to produce a sparse alignment $\mathcal{A}_w$. A LightGBM tag classifier scores per-alignment features: embedding cosine similarity, surface-form similarity, Arabic WordNet lookups, character n-gram Jaccard, Farasa (Abdelali et al., 2016) morphological distance, and antonym hits.

Phrase alignment and tagging. Phrase alignments are induced from word alignments via max-weight matching:

$$\text{align}(p_A, p_B) = \frac{|\mathcal{A}_w \cap (p_A \times p_B)|}{\max(|p_A|, |p_B|)}, \quad (1)$$

where $\mathcal{A}_w$ is the word-alignment set. Tag classification uses LightGBM (Ke et al., 2017) (500 boosting rounds, 31 leaves, learning rate 0.05) over phrase-pair features (full feature list in Appendix I).

Ordinal SEM head. Sentence-level SEM is naturally ordinal (`SEM-IDENT` ≻ `SEM-PARA` ≻ `SEM-PARTIAL` ≻ `SEM-DIFF`). We train two variants: (i) a 4-way LightGBM classifier over sentence-pair features, and (ii) an ordinal-regression variant where tags are regressed on a continuous score (3/2/1/0) and thresholded on dev. The tuned ordinal variant improves macro-F1 over plain classification (0.659 vs. 0.605 on test, 5-seed means), with the gain concentrated on the `SEM-DIFF` minority class.

Polarity-aware calibration gate. The classifier-based SEM head can miss adversarial `SEM-DIFF` cases (high embedding similarity, large surface overlap). We add an independent polarity check based on word-level antonym detection: Arabic WordNet (AWN) lookup with morphological backoff, `WRD-ANT` hits, and lightweight rules for ¬-marker proximity (لا, ليس, لم, لن) within aligned phrases. The gate fires on 19.0% ± 0.1% of test pairs (3-seed mean); ablation results are in Section 6.4.

Figure 3 illustrates the three-layer design. Crucially, Layer 3 is *not* a downstream rescorer of Layer 1; it consumes the same word-level alignment evidence (`WRD-ANT` hits, negation markers) but reaches its decision through entirely rule-based logic. This factorization is deliberate: a single embedding feature distribution cannot simultaneously yield high cosine for paraphrases *and* suppress them when their polarity is reversed, so the two decisions are made separately.

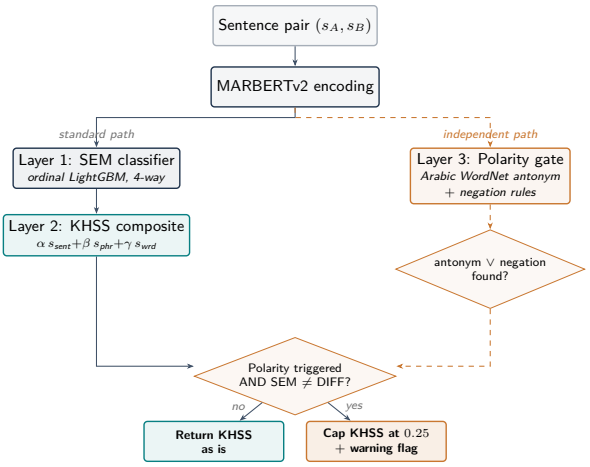


Figure 3: The polarity-aware calibration gate. Two independent paths consume the same MARBERTv2 encoding: the **standard path** (left, navy/teal) computes a similarity score KHSS; the **independent polarity gate** (right, orange) checks for antonyms or negation markers in the word-level alignment. The two paths converge at a decision step: if the polarity gate fires *and* the classifier did not already predict `SEM-DIFF`, KHSS is capped at 0.25 with a warning. This down-weights adversarial pairs whose surface similarity is high but polarity is reversed, a case the classifier alone tends to miss.

KHSS composite score. The KHSS combines hierarchical evidence into a single 0–1 score:

$$KHSS = \alpha\, s_{\text{sent}} + \beta\, s_{\text{phr}} + \gamma\, s_{\text{wrd}}, \quad (2)$$

where each level weight is modulated by a per-level confidence factor q_ℓ and the result is normalised, i.e. $KHSS = (\sum_\ell \alpha_\ell q_\ell\, s_\ell)/(\sum_\ell \alpha_\ell q_\ell)$ with base weights α_{sent}=0.4, α_{wrd}=0.4, α_{phr}=0.2 tuned on dev, each scaled by a per-level confidence factor q_ℓ (sentence: 1−normalised entropy of the predicted SEM distribution; word: alignment coverage; phrase: count-agreement of the two predicted segmentations; Appendix I). s_{sent} maps SEM tags to $\{1.0, 0.66, 0.33, 0.0\}$; s_{phr}, s_{wrd} are tag-weighted averages. After computing KHSS, the polarity gate caps it at 0.25 if triggered and the predicted SEM is not already `SEM-DIFF`.

5 Experimental Setup

Metrics. We report macro and weighted F1 for sentence-level SEM (4-way) and phrase-level alignment (6-way). For boundary detection, we report standard BP F1 (boundary-pair) and Phrase

Span F1 (exact-span match). For the overall composite, Spearman ρ between gold SEM-ordinal and predicted KHSS. We report 5-seed (42, 7, 2024, 11, 99) means in Table 3, 3-seed (42, 7, 2024) means for ablations, and paired bootstrap (10,000 resamples) with $p < 0.05$ (Koehn, 2004).

Implementation. KASHAF is implemented in PyTorch 2.4.1 with MARBERTv2[1] loaded via Hugging Face Transformers (Wolf et al., 2020). LightGBM (Ke et al., 2017) is used for word/phrase/SEM heads, and `sentence-transformers` (Reimers and Gurevych, 2019) for the optional Stage 2 contrastive pre-training. Inference runs on a single CPU at $\sim$50 ms/pair. We train and tune on a single NVIDIA RTX 3090. All hyperparameters are tuned on the development set; the test set is used only for final evaluation, never for model or hyperparameter selection.

Evaluation Splits. We adopt cross-batch evaluation: training data come from the 13 non-dev, non-test batches (5,809 pairs after cleaning), the development set is B13+B14 (1,000 pairs), and the test set is B03+B15 combined (1,002 pairs: $n_{\texttt{SEM-IDENT}} = 43$, $n_{\texttt{SEM-PARA}} = 198$, $n_{\texttt{SEM-PARTIAL}} = 718$, $n_{\texttt{SEM-DIFF}} = 43$). Cross-batch evaluation reduces the risk of single-batch annotator-style overfitting.

Comparison Systems. We compare against (1) majority-class and length-difference Jaccard baselines, (2) MARBERTv2 cosine (no learned head), (3) four frozen Arabic encoders with KASHAF's ordinal head, and (4) KASHAF classification and ordinal SEM heads with all six PHR tags. Alignment-tag accuracy follows SemEval iSTS (Agirre et al., 2016) in spirit, computed over BP-matched chunks without the official phrase-to-token fan-out weighting.

LLM 3-shot prompting. We benchmark four contemporary LLMs (GPT-4o, GPT-4o-mini, Claude Sonnet 4.5, Claude Haiku 4.5) on the same cross-batch test set. We use a carefully-engineered 3-shot prompt (12 in-context examples: 3 per class) (see Appendix) that gives the schema definitions, the ordinal scale, and three labeled exemplars per class (12 in-context examples in total). The prompt is identical across all four LLMs to control for prompting effects; only the underlying model differs. Decoding temperature is fixed at 0 and outputs are parsed as JSON. We additionally evaluate all four models under zero-shot prompting and under a chain-of-thought prompt that elicits step-by-step reasoning before the label. We report all 1,002 test pairs for each model; the same metrics are computed for all systems against the same gold labels.

[1]Hugging Face checkpoints: `UBC-NLP/MARBERTv2` (vanilla and Stage-2 backbone), `aubmindlab/bert-large-arabertv02-twitter` (Antoun et al., 2020), `CAMeL-Lab/bert-base-arabic-camelbert-msa` (Inoue et al., 2021), and the Arabic Triplet Matryoshka v2 sentence-transformer (Nacar et al., 2025) over `aubmindlab/bert-base-arabertv02`.

6 Results

6.1 Main Results

Table 3 reports the main test-set performance of KASHAF at each level of the hierarchy, alongside cosine baselines.

KASHAF achieves Spearman $\rho = 0.6715$ for sentence-level similarity (against the SEM ordinal scale), substantially above the MARBERTv2-cosine baseline ($\rho = 0.41$). On four-way SEM classification, the ordinal-regression head with tuned thresholds gives macro-F1 $= 0.659$ (5-seed mean, std ± 0.016, accuracy 0.843), with the largest gains on the `SEM-DIFF` minority class. Phrase classification reaches macro-F1 $= 0.792$ over six tags with gold phrase boundaries (0.420 end-to-end); boundary detection reaches BP F1 $= 0.838$ and span F1 $= 0.607$.

6.2 LLM Baseline Comparison

Figure 5 compares KASHAF against four contemporary LLMs and the frozen-encoder baselines on the same 1,002-pair cross-batch test set; full per-class LLM scores and API costs are deferred to Table 11 (Appendix G).

KASHAF substantially outperforms all four LLMs on the four-way SEM classification task. The best LLM under 3-shot prompting (Claude Haiku 4.5) achieves macro-F1 $= 0.215$, more than 44 absolute points below KASHAF's 0.659; chain-of-thought is the strongest LLM setting yet still trails by 32 points (best 0.336, Appendix G).

Few-shot vs zero-shot prompting. Re-running GPT-4o under strict zero-shot prompting (no in-context examples) yields essentially identical performance: macro-F1 0.207 vs. 3-shot 0.213 ($\Delta = -0.006$); ρ actually improves (0.526 vs. 0.505); the `SEM-PARTIAL` collapse is unchanged

System / Component	Macro F1	Weighted F1	Accuracy	Spearman ρ
Sentence-level (SEM, 4-way classification, n=1,002)				
Majority baseline (always `SEM-PARTIAL`)	0.209	0.598	0.717	—
Length-Jaccard baseline	0.270	0.480	0.580	0.34
MARBERTv2 cosine baseline	0.310	0.530	0.610	0.41
AraBERT Twitter Large (ordinal head)	0.593	0.828	0.838	—
Arabic Triplet Matryoshka (ordinal head)	0.618	0.830	0.837	—
CAMeLBERT-MSA (ordinal head)	0.617	0.823	0.827	—
MARBERTv2 vanilla (ordinal head)	0.644	0.831	0.834	—
KASHAF (classification head)	0.605	0.847	0.863	0.654
KASHAF (MARBERTv2 Stage 2, ordinal tuned)	**0.659**	**0.841**	**0.843**	**0.6715**
*Word alignment (n_{gold}=4,161 *alignments)*				
SimAlign `inter`	—	—	P.709/R.944	F1 = 0.810
AwesomeAlign	—	—	P.687/R.949	F1 = 0.797
Word-level tagging (5 WRD relation types; macro over 4, `WRD-ANT` n=2 excluded)				
KASHAF word-level E2E (joint F1 **0.828**, ALI F1 0.903)	0.590	—	0.917	—
Phrase-level (6-way alignment classification)				
KASHAF with *gold* phrase boundaries	0.792	0.937	0.886	—
KASHAF end-to-end (predicted everything)	**0.420**	0.438	—	—
Phrase boundary detection (BIO segmentation)				
KASHAF (CRF + MARBERTv2 layers 6–12)	—	—	BP F1 = 0.838	Span F1 = 0.607

Table 3: Main KASHAF results on cross-batch test (n=1,002). Sentence-level rows are 5-seed means (headline std ±0.016); the four alternative-encoder rows share KASHAF's ordinal-tuned head, isolating the encoder (Section 8). End-to-end PHR macro-F1 (0.420) is dominated by span-F1 (0.607); word-level E2E joint F1 is iSTS-style (Agirre et al., 2016) (correct only if both alignment and WRD tag are), computed over word pairs without the official evaluator's phrase-to-token fan-out weighting, so not directly comparable to SemEval-2016 submissions. KASHAF's ALI F1 (0.903, word row) is alignment accuracy on the gold-annotated word pairs, distinct from the all-token alignment F1 of the SimAlign/AwesomeAlign rows. Per-tag detail in Appendices A, B.

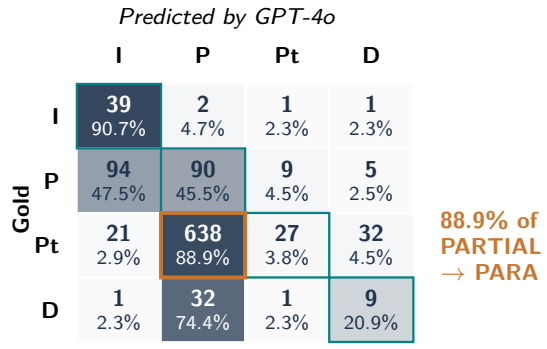


Figure 4: GPT-4o confusion matrix: counts and row percentages (navy shading); teal borders mark correct predictions. Gold `SEM-PARTIAL` (Pt) collapses overwhelmingly into the adjacent `SEM-PARA` (P): 638/718 (89%). The same pattern occurs across all four LLMs (I=IDENT, P=PARA, Pt=PARTIAL, D=DIFF).

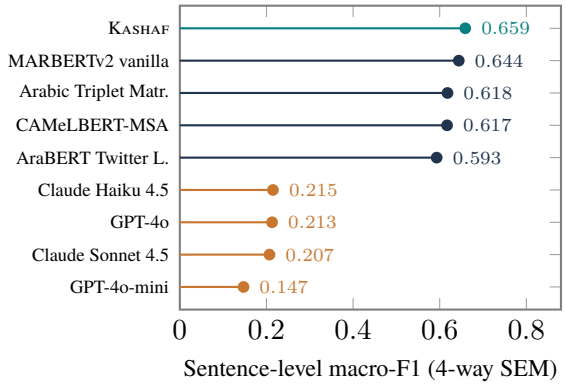


Figure 5: Sentence-level macro-F1 on the cross-batch test set (n=1,002). KASHAF (teal) and frozen Arabic encoder baselines (navy) share the ordinal head; LLMs (orange) use 3-shot prompting (12 examples). KASHAF reaches 0.659 versus the best 3-shot LLM's 0.215 (+44.4 points). The best CoT result is 0.336, narrowing the gap to 32.3 points (Appendix G).

(87.9% vs. 88.9%). In-context examples do not address the miscalibration in our setup.

The `SEM-PARTIAL` collapse. A notable pattern is that *none* of the four LLMs handle the `SEM-PARTIAL` class (per-class F1 in Figure 6, Appendix A). `SEM-PARTIAL` is the dominant gold class (71.7% of test pairs), yet all four LLMs achieve `SEM-PARTIAL` recall ≤7% under zero-shot and 3-shot prompting; it is only partially alleviated under chain-of-thought (best `SEM-PARTIAL` F1 0.47; Appendix G). LLMs collapse `SEM-PARTIAL` into the adjacent `SEM-PARA` category (Figure 4): GPT-4o predicts `SEM-PARA`

for 76% of pairs (gold prior = 19.8%). LLMs reach $\rho \in [0.42, 0.53]$: coarse but not fine-grained. A trivial *majority-class* baseline (always predict SEM-PARTIAL) reaches macro-F1 0.209 (Table 3), matching the LLMs under standard prompting despite no modeling; KASHAF's $\rho = 0.6715$ exceeds the best LLM by 0.14–0.25 points.

Cost. KASHAF runs on CPU at ∼50 ms/pair (zero *marginal* cost; training is a one-time ∼25 min on a single RTX 3090). GPT-4o costs $1.50 per 1,002-pair evaluation, Claude Sonnet $2.55.

6.3 Stage 2 Ablation

To address the question, "*is the contrastive pre-training (Stage 2) step actually contributing?*", we re-train the phrase classifier, ordinal SEM head, and boundary detector from *vanilla* MARBERTv2 (no contrastive pre-training) across three random seeds and compare.

The pattern is nuanced. Phrase classification and untuned 4-way SEM show **no significant difference** between vanilla MARBERTv2 and Stage 2 ($p \in \{0.40, 0.80, 0.89\}$). At the three seeds shared with the vanilla baseline, the ordinal-tuned SEM head shows a small *marginal* Stage-2 advantage: 0.6530 vs. vanilla 0.6446 (Δ=+0.0084, p=0.07, 3-seed paired t-test; the 5-seed headline in Table 3 is 0.659); BP F1 and Span F1 show similar marginal gains ($\Delta \approx +0.005$, $p \in [0.05, 0.10]$; Table 13). All absolute differences are below 1% macro-F1: the hierarchical pipeline and feature engineering carry most of the performance, and KASHAF can be deployed without Stage 2 at a cost of only ~ 0.008 macro-F1.

6.4 KHSS Composite and Polarity-Gate Analysis

Two SEM heads, two purposes. KASHAF uses two distinct LightGBM SEM heads: the *headline* ordinal-tuned head (Table 3, ρ=0.6715, 5-seed macro-F1 = 0.659) optimised for class prediction, and a separate *KHSS head* (3-seed macro-F1 = 0.589 on the same test) used solely for probability calibration in the composite below. All composite numbers use the KHSS head; ρ agrees across heads to ± 0.005.

Component decomposition (3 seeds). In isolation against gold SEM-ordinal, $\rho(s_{\text{sent}})$=0.660, $\rho(s_{\text{phr}})$=0.412, $\rho(s_{\text{wrd}})$=0.377; the phrase/word terms are weaker because they use *predicted* structure (span-F1 = 0.607). Combining all three under dev-tuned base weights $(0.4, 0.4, 0.2)$ (confidence-scaled; Section 4) gives ρ=0.6484, *slightly below* s_{sent} alone (-0.012). We therefore treat s_{phr} and s_{wrd} as *interpretive* signal localising where sentences agree or diverge (Table 5), not as scalar-correlation lift.

Polarity gate (3 seeds). The gate (Figure 3) fires on 19.0%±0.1% of test pairs, raising the seed-mean ρ from 0.6484 to 0.6715 ($\Delta\rho$=+0.0231); a paired bootstrap over the 1,002 test pairs (10k resamples) gives a 95% CI of $[-0.007, +0.053]$ (one-sided p=0.072), so this aggregate gain is not significant on this split. The gate is an empirical score-capping rule rather than a validated contradiction detector: it fires mostly on SEM-PARTIAL pairs and has low precision as a strict SEM-DIFF detector (0.115; Appendix E). It caps flagged high-similarity pairs to a lower ordinal target rather than re-labelling them, and since it never changes the predicted SEM class, its effect is confined to the continuous KHSS score.

7 Analysis

A worked example. Table 4 shows the full three-level analysis of a real gold pair from the test set (B03_1045). The two sentences أهلاً بك في منزلك الجديد ("welcome to your new home", A) and مرحباً بك في منزلك الجديد (B) receive the sentence-level tag SEM-PARA: they carry the same meaning with one lexical substitution. Segmentation yields two phrases per sentence; the aligner pairs them one-to-one. The first phrase pair, أهلاً بك / مرحباً بك, is tagged PHR-SYN (the greetings أهلاً and مرحباً are synonyms), while the second, في منزلك الجديد in both, is tagged PHR-MATCH. At the word level, phrase 1 yields WRD-SYN for أهلاً↔مرحباً and WRD-MATCH for بك; phrase 2 yields WRD-MATCH for all three of its words (في, منزلك, الجديد). The single lexical alternation is thus localised precisely, from sentence to phrase to the individual word, which is exactly the kind of layered trace the interpretable levels are meant to expose.

On the usability of interpretive traces. The founding motivation for iSTS is not intrinsic accuracy alone but *usability*: a system should be able to explain, to a human, why two sentences are judged similar or different. Lopez-Gazpio et al. (2017) made this explicit for English, pairing typed chunk alignments with a verbalization

#	Phrase in A	Phrase in B	Tag
Phrase level			
1	أهلاً بك	مرحباً بك	PHR-SYN
2	في منزلك الجديد	في منزلك الجديد	PHR-MATCH
Word level (within phrase 1)			
1	أهلاً	مرحباً	WRD-SYN
2	بك	بك	WRD-MATCH

Table 4: Three-level trace for pair B03_1045 (sentence-level: SEM-PARA). The synonym substitution مرحباً→أهلاً is localised to phrase 1 and, within it, to a single WRD-SYN word pair; everything else is an exact match. Word-level rows are the schema applied to the gold alignment.

module and two user studies that showed the explanations helped users. Our human evaluation (Appendix J) extends this line to Arabic for the first time: rather than measuring only agreement with gold traces (joint-F1), two linguists rated the *system-produced* phrase- and word-level traces for correctness and helpfulness on a balanced 100-pair sample. The traces are judged useful (helpfulness $3.15/5$) though below the gold upper bound (-1.1), and, consistent with the intrinsic scores, helpfulness slightly exceeds correctness, i.e. a trace can help a reader localise the relation even when it is not perfectly correct. This situates KASHAF's interpretive layers within the original usability goal of iSTS, and quantifies, for Arabic, how far automatically-produced traces still sit from human-quality explanations.

Where does KASHAF fail? Inspecting 100 errors reveals four recurring patterns. **Schema boundary cases (38%)** are pairs on the SEM-PARA/SEM-PARTIAL frontier where the annotators themselves disagreed before adjudication, an inherent ambiguity of the scheme rather than a modelling error (Appendix C). **Morphological complications (22%)** arise from Arabic's rich inflection and clitics; the particle ما, for instance, serves four distinct grammatical functions. **Long-range coreference (15%)** degrades alignment on longer inputs, with word-level F1 dropping from 0.91 to 0.78 above 20 tokens. The remaining **25%** are stylistic and idiomatic alternations. The per-tag phrase-level breakdown, including the under-predicted REORDER tail, is in Appendix A. REORDER is in fact our hardest tag (F1 $= 0.018$): beyond its rarity (1.7% of gold), it has a linguistic root. In Arabic, reordering constituents often co-occurs with a subtle semantic shift, so the sentence-level label becomes a borderline decision among SEM-IDENT, SEM-PARA, and SEM-PARTIAL, the very ambiguity our annotators faced before adjudication. This coupling of rarity with boundary ambiguity explains the low recovery.

8 Discussion

Why do LLMs collapse SEM-PARTIAL? Three signals are consistent with a calibration explanation: (i) single-label prompts bias models toward prototypes; (ii) Haiku $\approx$ Sonnet (~ 0.21, scale invariance); (iii) SEM-PARTIAL maps to SEM-PARA but not vice versa. The collapse is most acute for the three larger models.

Why does the MARBERTv2 family win, and Stage 2 so little? KASHAF's lead over AraBERT Twitter Large (Holm p=0.003) and CAMeLBERT-MSA (p=0.046) is significant under five-seed correlation-preserving randomization; the Triplet gap is not (p=0.108; Appendix H). MARBERTv2's joint MSA+dialectal pre-training matches our mostly-MSA, multi-genre register; the alternatives target Twitter, drop dialect, or optimise a coarse embedding objective. Stage 2 contrastive tuning adds only Δ=+0.008 (n.s.; Section 6.3): the encoder, not auxiliary tuning, drives performance.

9 Conclusion

We presented KASHAF, the first interpretable STS resource and end-to-end system for Arabic: a 7,811-pair human-annotated corpus (κ=0.83) with a three-level schema and a modular pipeline yielding a composite KHSS score. KASHAF reaches sentence-level macro-F1 0.659 and Spearman ρ=0.6715; an ablation isolates the MARBERTv2 encoder, not contrastive tuning, as the driver, and a polarity-aware gate caps scores for high-similarity contradictions ($\Delta\rho$=+0.0231, n.s.). The interpretable layers localise agreement and divergence: phrase-alignment classification reaches macro-F1 0.420 and word-level tagging an iSTS-style joint F1 of 0.828. LLMs trail by 32–44 points, over-predicting SEM-PARA; typed alignment remains open. Guidelines, code, models, and a corpus sample are released; dialectal coverage is future work.

Limitations

We identify five primary limitations of this work.

First, the SEM-PARTIAL class is dominant in our test data (71.7%), reflecting the corpus's natural distribution but creating a heavily imbalanced classification problem. While KASHAF computes macro-F1 across all four classes (so PARTIAL dominance does not inflate the headline number), the absolute class frequencies are unlike many English iSTS corpora and may overstate KASHAF's advantage on PARTIAL-heavy data. We do not have an independent balanced Arabic test set on which to verify generalization; building one with double-annotated boundary cases between SEM-PARA and SEM-PARTIAL is a natural next step. Relatedly, our human evaluation of interpretive traces (Appendix J) shows the predicted traces are useful but rated $\sim$1.1 points below gold on a 5-point scale; closing this human-level trace gap remains open.

Second, the corpus is Modern Standard Arabic with limited dialectal coverage; downstream performance on Egyptian, Levantine, or Gulf dialects is unverified. We have not systematically evaluated cross-dialectal transfer and expect a measurable drop for non-MSA text.

Third, the polarity-aware calibration gate caps the continuous KHSS score rather than re-labelling the predicted SEM class, so its contribution to SEM-DIFF *classification* F1 is zero because the predicted SEM label is unchanged; a gate variant that also re-classifies is a natural extension. As noted in Section 6.4, the hierarchical composite *without* the gate is slightly below s_{sent} alone (-0.012 Spearman), so we treat the phrase- and word-level terms as an interpretive overlay (Table 5) rather than a predictive lift.

Fourth, we evaluate four LLMs under zero-shot, 3-shot, and chain-of-thought (CoT) prompting (Appendix G). CoT helps (the best LLM rises from 0.215 under standard prompting to 0.336 with GPT-4o-mini) but still trails KASHAF by 32 macro-F1 points, and 60–82% of gold SEM-PARTIAL is routed to SEM-PARA even under CoT. We did not explore multi-template ensembles or task fine-tuning; a determined fine-tuned LLM could narrow the gap further, though the persistence of the dominant-class collapse across all three prompting strategies suggests the failure mode is tied to single-label ordinal prompting rather than a transient artifact. Although our corpus spans several genres (subtitles, product reviews, encyclopaedic Wikipedia text, and scientific Arabic), it is not stratified by genre, so per-genre generalisation is not separately quantified, and coverage of legal and heavily dialectal Arabic remains limited.

Fifth, statistical power for the paired baseline comparisons is limited by the five seeds per system and the 1,002-pair test set. Correlation-preserving per-Pair_ID randomization shows the AraBERT and CAMeLBERT comparisons surviving Holm–Bonferroni correction (adjusted p=0.003 and 0.046), whereas the Triplet Matryoshka and vanilla MARBERTv2 comparisons do not (adjusted p=0.108 and 0.183), even though effect sizes are large throughout (d=0.6–3.4; full statistics in Appendix H). We therefore frame KASHAF's edge over the closest baselines as a *consistent cross-seed ranking* rather than a correction-significant gap, and recommend larger baseline seed counts in future work. Finally, the SEM-DIFF class, tied with SEM-IDENT as the smallest at 4.3% of the test set, carries the most annotation uncertainty: it has the lowest and widest-interval per-class F1 (0.480, 95% CI $[0.353, 0.593]$), and manual re-inspection indicates a subset of gold SEM-DIFF pairs are borderline with SEM-PARA. Targeted re-annotation of this class is future work and may raise measured SEM-DIFF performance.

References

Ahmed Abdelali, Kareem Darwish, Nadir Durrani, and Hamdy Mubarak. 2016. Farasa: A Fast and Furious Segmenter for Arabic. In *Proceedings of NAACL-HLT 2016: Demonstrations*, pages 11–16.

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 (ACL 2021)*.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Iñigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. 2015. SemEval-2015 Task 2: Semantic Textual Similarity, English, Spanish and Pilot on Interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 252–263.

Eneko Agirre, Aitor Gonzalez-Agirre, Iñigo Lopez-Gazpio, Montse Maritxalar, German Rigau, and Larraitz Uria. 2016. SemEval-2016 Task 2: Interpretable Semantic Textual Similarity. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 512–524.

Fahad AlGhamdi, Abdelati Hawwari, Abdulrahman AlOsaimy, Rawan Almatham, and Asma Al Wazrah. 2026. KASHAF annotation guidelines for Arabic interpretable semantic textual similarity (english and Arabic). King Salman Global Academy for the Arabic Language (KSAA); released with the code and data.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. AraBERT: Transformer-based model for Arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France. European Language Resources Association.

Ron Artstein and Massimo Poesio. 2008. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596.

Emily M. Bender and Batya Friedman. 2018. Data statements for natural language processing: Toward mitigating system bias and enabling better science. *Transactions of the Association for Computational Linguistics*, 6:587–604.

William Black, Sabri Elkateb, Horacio Rodriguez, Musa Alkhalifa, Piek Vossen, Adam Pease, and Christiane Fellbaum. 2006. Introducing the Arabic WordNet Project. In *Proceedings of the Third International WordNet Conference*.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14.

Zi-Yi Dou and Graham Neubig. 2021. Word Alignment by Fine-tuning Embeddings on Parallel Corpora. In *Proceedings of EACL 2021*.

Joseph Gatto, Omar Sharif, Parker Seegmiller, Philip Bohlman, and Sarah Masud Preum. 2023. Text Encoders Lack Knowledge: Leveraging Generative LLMs for Domain-Specific Semantic Textual Similarity. ArXiv:2309.06541.

Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun-Hsuan Sung, Laszlo Lukacs, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient Natural Language Response Suggestion for Smart Reply. In *arXiv:1705.00652*.

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 (Online). Association for Computational Linguistics.

Masoud Jalili Sabet, Philipp Dufter, François Yvon, and Hinrich Schütze. 2020. SimAlign: High Quality Word Alignments Without Parallel Training Data Using Static and Contextualized Embeddings. In *Findings of EMNLP 2020*, pages 1627–1643.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Advances in Neural Information Processing Systems (NeurIPS 2017)*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of ICLR 2015*.

Philipp Koehn. 2004. Statistical Significance Tests for Machine Translation Evaluation. In *Proceedings of EMNLP 2004*.

John Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of ICML 2001*.

J. Richard Landis and Gary G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174.

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting Large Parallel Corpora from Movie and TV Subtitles. In *Proceedings of LREC 2016*.

Iñigo Lopez-Gazpio, Montse Maritxalar, Aitor Gonzalez-Agirre, German Rigau, Larraitz Uria, and Eneko Agirre. 2017. Interpretable semantic textual similarity: Finding and explaining differences between sentences. *Knowledge-Based Systems*, 119:186–199.

Goutam Majumder, Partha Pakray, Ranjita Das, and David Pinto. 2021. Interpretable semantic textual similarity of sentences using alignment of chunks with classification and regression. *Applied Intelligence*, 51(10):7322–7349.

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*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks. In *Proceedings of EMNLP-IJCNLP 2019*.

Jingxuan Tu, Keer Xu, Liulu Yue, Bingyang Ye, Kyeongmin Rim, and James Pustejovsky. 2024. Linguistically Conditioned Semantic Textual Similarity. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1161–1172.

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 EMNLP 2020: System Demonstrations*, pages 38–45.

A Per-Tag Phrase Performance

Table 5 breaks down end-to-end phrase performance by the six PHR alignment tags, and Figure 6 shows per-class sentence-level F1 for KASHAF against GPT-4o.

PHR tag	P	R	F1	n_{gold}
PHR-MATCH	0.811	0.732	**0.770**	1,374
PHR-ADD_IN_B	0.448	0.550	0.494	1,243
PHR-DEL_IN_B	0.435	0.515	0.472	1,219
PHR-MOD	0.341	0.508	0.408	1,601
PHR-SYN	0.343	0.374	0.358	5,516
REORDER	0.071	0.010	0.018	192
Macro			**0.420**	
Micro	0.411	0.466	**0.437**	11,145

Table 5: Per-tag PHR results for the *end-to-end* pipeline (predicted boundaries, predicted alignments) on the cross-batch test (seed 42; other seeds within ± 0.01). PHR-MATCH (lexical overlap) is well-recovered (0.77 F1) because predicted boundaries usually capture full chunks; PHR-MOD and PHR-SYN are harder because they require fine-grained semantic distinction inside small spans. REORDER (1.7% of gold, n=192) is the schema's long-tail class and severely under-predicted (F1 = 0.018): manual inspection of all 192 gold cases shows most involve short ambiguous chunks ($\leq$ 4 words), a schema-design tension worth re-examining.

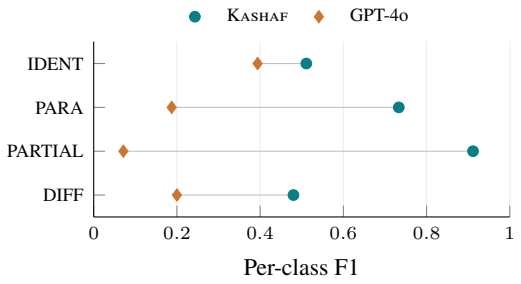


Figure 6: Per-class F1 on the cross-batch test set: KASHAF (5-seed mean, ordinal-tuned) versus GPT-4o (3-shot, from its confusion matrix, Figure 4). The gap is widest on the majority class `SEM-PARTIAL` (71.7% of data): KASHAF 0.91 vs. GPT-4o 0.07. On `SEM-DIFF`, KASHAF reaches 0.48 vs. 0.20. Matched 95% bootstrap CIs (5-seed): `SEM-IDENT` $[0.35, 0.65]$, `SEM-PARA` $[0.70, 0.77]$, `SEM-PARTIAL` $[0.90, 0.93]$, `SEM-DIFF` $[0.35, 0.59]$; each contains its point estimate.

B Per-Tag Word Performance and iSTS Joint Correctness

Table 6 reports word-level end-to-end performance per WRD tag, together with the SemEval-iSTS-style joint correctness.

WRD tag	P	R	F1	n_{gold}
WRD-MATCH	0.996	0.981	**0.989**	2,919
WRD-MORPH	0.509	0.799	0.621	387
WRD-FUNC	0.566	0.421	0.484	500
WRD-SYN	0.258	0.275	0.267	353
WRD-ANT†	0.000	0.000	—	2
Macro (excl. ANT)			**0.590**	
Joint E2E F1	0.819	0.836	**0.828**	4,161

Table 6: Per-tag WRD end-to-end performance (3-seed mean) on the cross-batch test set. The pipeline scores each word-pair correct only when both the alignment (ALI) and the tag type (TYPE) are correct, i.e. the iSTS-style joint correctness metric. This table reports the five WRD relation types; words with no counterpart (`ADD_IN_B`/`DEL_IN_B`) are recovered by the alignment stage (ALI F1 0.903) rather than type-classified. The n=4,161 word decisions are the aligned word pairs across all test sentences, distinct from the 11,145 phrase-level decisions (Table 5), which additionally count unilateral `ADD`/`DEL` phrases. Performance is dominated by WRD-MATCH (n=2,919, 70% of gold), the easiest case. The harder distinctions are WRD-MORPH (0.621, morphological inflection) and WRD-FUNC (0.484, function-word substitution), each requiring fine-grained word-internal discrimination. WRD-SYN (0.267) is the hardest open-set class because Arabic synonymy spans a much larger lexical space than the model has direct evidence for. †WRD-ANT has only n=2 gold instances in test, both missed, giving F1 = 0 by the usual convention; we report the 4-class macro excluding it as an auxiliary measure given this extreme sparsity, and maintain the tag in the schema because the polarity-gate module already detects antonymy at the sentence level (Section 4, Appendix E). A future balanced challenge set (Limitations §1) will allow a robust WRD-ANT evaluation.

C Schema Decision Rules (IDENT vs. PARA vs. PARTIAL vs. DIFF)

Table 7 lists the annotator-facing decision rules that distinguish the four SEM classes, with canonical Arabic examples.

D Correspondence to SemEval iSTS

Table 8 maps our PHR and WRD tag sets onto the SemEval-2016 iSTS alignment types.

SEM tag	Decision rule (annotator-facing)	Canonical Arabic example
SEM-IDENT	Surface forms differ only in punctuation, whitespace, orthographic variants (*alif* forms, *ta marbuta/ha*), or short-vowel diacritics. **No** lexical, morphological, or word-order change.	A: ذهب الولد إلى السوق. B: ذهب الولد إلى السوق (no period)
SEM-PARA	All information units in A appear in B and vice versa (bidirectional entailment), but at least one differs in surface lexicalization (synonym), inflection (gender/number), or word order. No information unit is added, removed, or made more/less specific; otherwise the pair is SEM-PARTIAL.	A: هذا قرار ممتاز B: هذا قرار رائع (pure synonym: "excellent" → "great")
SEM-PARTIAL	A and B share the topic but at least one of: (i) one side adds an information unit absent on the other; (ii) a modifier differs in specificity (e.g., temporal/quantifier); (iii) different participants/objects/locations are referenced.	A: تطبيق الخدمات المصرفية عبر الهاتف المحمول آمن وله ميزات مفيدة B: تطبيق البنك آمن وفيه خدمات مفيدة (A adds "via mobile")
SEM-DIFF	A and B describe different events, propositions, or facts. Polarity reversal (negation/antonym) of a core predicate also forces SEM-DIFF.	A: أحب القراءة كثيرا B: أكره القراءة كثيرا (antonym of core predicate)

Table 7: SEM-tag decision rules used by annotators. Boundary cases between SEM-PARA and SEM-PARTIAL accounted for 38% of our error analysis (Section 7); the rules above formalize the most frequent disagreements observed in pilot batches. Full annotation guidelines (English + Arabic, including word-level and phrase-level decision trees) are released with the code.

KASHAF tag	SemEval iSTS relation
Sentence level (SEM)	
SEM-IDENT	EQUI
SEM-PARA	EQUI / SIMI
SEM-PARTIAL	SIMI / REL
SEM-DIFF	OPPO / NOALI
Phrase level (PHR), chunk alignment	
PHR-MATCH	EQUI
PHR-MOD	EQUI
PHR-SYN	EQUI / SIMI
PHR-REORDER	EQUI (position only)
PHR-ADD_IN_B	SPE2 / NOALI
PHR-DEL_IN_B	SPE1 / NOALI
Word level (WRD)	
WRD-MATCH/MORPH/FUNC	EQUI
WRD-SYN	EQUI / SIMI
WRD-ANT	OPPO
WRD-ADD_IN_B	SPE2 / NOALI
WRD-DEL_IN_B	SPE1 / NOALI

Table 8: Approximate correspondence between KASHAF tags and SemEval-2016 Task 2 iSTS chunk-alignment relations (Agirre et al., 2016). These are indicative relation mappings, not one-to-one equivalences; KASHAF tag semantics are defined in Table 1. Our schema *refines* iSTS in three ways: (i) it separates surface-preserving from surface-changing equivalence (MATCH vs. MOD/SYN/MORPH); (ii) it marks word order explicitly via REORDER, which co-occurs with EQUI; and (iii) its directional ADD_IN_B/DEL_IN_B record *which* sentence carries extra material, paralleling iSTS SPE1/SPE2. We omit the numeric iSTS similarity scores, which apply to the original chunk-alignment scale and not to KASHAF outputs.

E Polarity-Gate Behaviour by Class

Table 9 reports how often the polarity gate fires for each gold class, and its precision and recall as a SEM-DIFF detector.

F LLM Prompt Template

Table 10 gives the (abbreviated) prompt template shared by all four LLMs across the prompting conditions.

G LLM Comparison Detail

Table 11 reports the full zero-shot, 3-shot, and chain-of-thought results for all four LLMs.

H Statistical Methods and Per-Pair Significance

We quantify the significance of KASHAF's macro-F1 advantage over each encoder baseline with a *correlation-preserving* approximate-randomization test over all 5 seeds (10,000 iterations). Because the same test pair recurs across seeds, we do not pool the $5\times1{,}002$ predictions as independent observations; instead, for each Pair_ID we draw a single swap decision and apply it jointly across all seeds, then recompute the seed-averaged macro-F1 difference, avoiding pseudo-replication. Table 12 reports the macro-F1 difference Δ, the two-sided randomization

Gold class	n	Fires	Rate
SEM-IDENT	43	0	0.0%
SEM-PARA	198	3	1.5%
SEM-PARTIAL	718	166	23.1%
SEM-DIFF	43	22	**51.2**%
Total	1,002	191	19.1%
Gate as SEM-DIFF detector: P= 0.115, R= 0.512, F1= 0.188			
Aggregate Spearman: $\Delta\rho$=+0.0231, bootstrap 95% CI [−0.007, +0.053] (n.s.)			

Table 9: Polarity gate behaviour by gold class (seed 42; other seeds within ±1 pair). The gate never fires on SEM-IDENT and rarely on SEM-PARA (1.5% of 198 pairs), recovers 51% of SEM-DIFF gold, but is most active on SEM-PARTIAL (23%); over all non-SEM-DIFF classes its false-positive rate is $169/959 \approx 17.6\%$. Its low precision as a strict SEM-DIFF detector (0.115) reflects that SEM-PARTIAL pairs with detected antonym/negation cues (e.g. partial polarity reversal) are score-capped to 0.25, which aligns the score with their lower ordinal target. The resulting aggregate Spearman gain ($\Delta\rho$=+0.0231) is positive but not statistically significant on the test split (paired-bootstrap 95% CI [−0.007, +0.053], one-sided p=0.072); the gate is, by design, an empirical score-capping rule rather than a class re-labeler, and we do not claim it as a validated contradiction detector.

Prompt structure (identical across all four LLMs; 12 in-context examples for 3-shot, 0 for zero-shot)

System. You are an Arabic Semantic Textual Similarity annotator. You will be given two Arabic sentences and you must classify their semantic relation using exactly one of four tags: SEM-IDENT, SEM-PARA, SEM-PARTIAL, SEM-DIFF.

Tag definitions.
- SEM-IDENT: identical content; differ only in punctuation, whitespace, or short-vowel diacritics.
- SEM-PARA: paraphrases; same propositional content with different surface form (synonyms, inflection, word order).
- SEM-PARTIAL: share topic but at least one information unit, modifier specificity, or participant differs.
- SEM-DIFF: describe different events/facts, or one sentence reverses polarity of a core predicate.

3-shot examples (12 total; 3 per class, randomly sampled once from the training set, fixed across all 4 LLMs):
Example 1. A: *[Arabic sentence]*; B: *[Arabic sentence]*; Label: SEM-IDENT
Example 2. A: *[Arabic sentence]*; B: *[Arabic sentence]*; Label: SEM-IDENT
. . . (3 examples per class for all four classes) . . .

Task. A: *[Arabic sentence A]*
B: *[Arabic sentence B]*
Output format. Return a JSON object: `{"label": "<one of the four tags>", "reasoning": "<one sentence in Arabic>"}`. Do not return anything else.

Table 10: LLM prompt template (translated/abbreviated for space; full prompt is released with the code). For zero-shot, the 12 in-context examples are omitted; all other instructions remain identical. Decoding temperature = 0; max tokens = 80.

p-value, Cohen's d over seed-level means, and the Holm–Bonferroni-adjusted p-value over the four-comparison family (FWER=0.05). After correction, the AraBERT and CAMeLBERT comparisons are significant, while Triplet and MARBERTv2-vanilla are not, consistent with the marginal Stage-2 effect in Table 13.

Pairwise significance claims are based on the correlation-preserving approximate-randomization test reported in Table 12. The observed macro-F1 differences are positive for all four comparisons, but only the AraBERT and CAMeLBERT comparisons remain significant after Holm–Bonferroni correction.

I Reproducibility: Hyperparameters and Feature Sets

This appendix specifies every hyperparameter, the dev-tuning choices, and the exact feature sets, so that each reported number can be traced through the released code. Unless noted, defaults below are the values used for all reported runs.

Encoder and embeddings. The backbone is MARBERTv2 (Abdul-Mageed et al., 2021) (163M parameters; 12 layers; hidden size 768; 12 attention heads; intermediate size 3072; vocabulary 100,000; `float32`). Word- and phrase-alignment features use frozen **layer-8** token embeddings (selected on dev). The boundary detector instead

System / prompting	Macro F1	SEM-PARTIAL F1
Zero-shot prompting		
GPT-4o-mini	0.147	0.136
GPT-4o	0.207	0.086
Claude Haiku 4.5	0.162	0.099
Claude Sonnet 4.5	0.156	0.079
3-shot prompting (12 in-context examples)		
GPT-4o-mini	0.147	0.119
GPT-4o	0.213	0.071
Claude Haiku 4.5	0.215	0.086
Claude Sonnet 4.5	0.207	0.079
Chain-of-thought prompting (zero-shot)		
GPT-4o-mini	**0.336**	**0.467**
GPT-4o	0.303	0.340
Claude Haiku 4.5	0.291	0.250
Claude Sonnet 4.5	0.324	0.298
KASHAF (ordinal, tuned)	**0.659**	**0.912**

Table 11: LLM baselines across three prompting strategies versus KASHAF on the cross-batch test set (n=1,002). Model IDs: `gpt-4o-2024-08-06`, `gpt-4o-mini-2024-07-18`, `claude-sonnet-4-5-20250929`, `claude-haiku-4-5`; decoding temperature 0. Across *all twelve* LLM configurations, 60–91% of gold SEM-PARTIAL is routed to SEM-PARA (the dominant-class collapse), versus KASHAF's SEM-PARTIAL recall of 0.947. Chain-of-thought is the strongest LLM setting (best 0.336, GPT-4o-mini) yet still trails KASHAF by 32 macro-F1 points; under zero-shot and 3-shot, SEM-PARTIAL recall is $\leq$7%. 3-shot lifts macro-F1 modestly over zero-shot (Haiku 0.162→0.215) yet leaves this collapse intact. KASHAF's training is a one-time ∼25 min on a single RTX 3090.

uses the per-token mean of **layers 6–12** of a fine-tuned encoder. Inputs are Arabic-normalised (diacritic removal, alif/ya unification) before tokenisation. Maximum sequence length is 128 tokens for the SEM head and 64 for the boundary detector.

LightGBM tag classifiers (SEM, phrase, word). All three heads share the same core configuration, tuned on dev: objective `multiclass` with `multi_logloss`; learning rate 0.05; `num_leaves` = 31; `min_data_in_leaf` = 20; `feature_fraction` = 0.9; `bagging_fraction` = 0.9 with `bagging_freq` = 5; L1/L2 regularisation λ_1=λ_2=0.1; and 500 boosting rounds. Class imbalance is handled with per-class sample weights.

Ordinal SEM head. The headline SEM model is a LightGBM *regression* on the ordinal scalar target $\{0, 1, 2, 3\}$ (SEM-DIFF=0 to SEM-IDENT=3). Continuous predictions are converted to classes with three thresholds initialised at $[0.5, 1.5, 2.5]$ and then tuned on dev by maximising macro-F1 (Nelder–Mead). Tuned thresholds improve macro-F1 over the default thresholds, with the gain concentrated on the minority SEM-DIFF class. We report 5-seed means (seeds 42, 7, 2024, 11, 99).

Boundary detector (BIO + CRF). Fine-tuned MARBERTv2 with a linear layer and a CRF, trained for 5 epochs; batch size 32; learning rate 2×10^{-5}; Adam (Kingma and Ba, 2015); maximum length 64.

Word alignment. SimAlign (Jalili Sabet et al., 2020) with the `inter` matching method over layer-8 embeddings, producing the sparse alignment set $\mathcal{A}_w$ consumed by the phrase- and word-level classifiers.

KHSS composite. Base level weights (tuned on dev) are α_{sent}=0.4, α_{wrd}=0.4, α_{phr}=0.2. Each weight is scaled by a per-level confidence factor q_ℓ, and the weighted sum is normalised by $\sum_\ell \alpha_\ell q_\ell$ (Section 4). The confidence factors use only predicted quantities (no gold): q_{sent}=$1-\hat{H}$, where $\hat{H}$ is the normalised entropy of the predicted SEM probability vector (high when the classifier is confident); q_{word} is word-alignment coverage, the fraction of source tokens receiving an alignment (|aligned|/|tokens|); and q_{phrase}= $\min(n_p, \hat{n}_p)/\max(n_p, \hat{n}_p)$ is a count-agreement between the two predicted segmentations (not boundary accuracy). Each returns a value in $[0, 1]$. Because every sentence contains at least one token and at least one phrase, q_{phrase}>0, so the normalising denominator $\sum_\ell \alpha_\ell q_\ell$ is never zero; the fallback q=0 for an empty per-level denominator is therefore never reached in practice. Within phrase pairing, alignment and semantic evidence are combined as sim = $\alpha_{\text{pair}} \cdot$ align + $(1 - \alpha_{\text{pair}}) \cdot$ sem with α_{pair}=0.5; this α_{pair} is distinct from the KHSS level weights above. The phrase score s_{phr} uses a per-tag value table (e.g., REORDER=0.30; unmatched default 0.5). If the polarity gate fires and the predicted SEM is not already SEM-DIFF, KHSS is capped at 0.25.

Feature sets. The word-level classifier uses ≈22 features and the phrase-level classifier uses a distinct ≈23-feature set; both combine surface and semantic groups. *Surface features* include exact match, token- and character-length statistics and ratios, Levenshtein distance (raw and normalised), unigram/bigram/trigram Jaccard, prefix/suffix overlap, stem match, and directional

Comparison	Δ	raw p	Cohen's d	Holm p
KASHAF *vs* AraBERT Twitter L.	+0.066	0.0007	+3.4	**0.003**
KASHAF *vs* CAMeLBERT-MSA	+0.042	0.015	+2.2	**0.046**
KASHAF *vs* Triplet Matryoshka	+0.041	0.054	+2.0	0.108
KASHAF *vs* MARBERTv2 vanilla	+0.015	0.183	+0.6	0.183

Table 12: Correlation-preserving approximate-randomization significance of KASHAF vs each baseline (all 5 seeds; swap decisions drawn per `Pair_ID` and applied jointly across seeds; 10^4 iterations). Holm–Bonferroni over m=4 comparisons: the AraBERT and CAMeLBERT comparisons survive FWER=0.05. Cohen's d is computed over seed-level macro-F1 means.

Component / Metric	Stage 2	Vanilla	Δ (V−S2)	p-value
Phrase classifier (6-tag, 3-seed mean)				
Macro-F1	0.7927	0.7921	−0.0006	0.80
Weighted-F1	0.9366	0.9363	−0.0003	0.89
Ordinal SEM (4-tag, 3-seed mean, tuned thresholds)				
Classification Macro-F1	0.5996	0.5929	−0.0067	0.40
Ordinal Tuned Macro-F1	0.6530	0.6446	−0.0084	0.07
BoundaryDetector (CRF) (3-seed mean)				
BP F1	0.8411	0.8375	−0.0036	0.05
Phrase Span F1	0.6157	0.6075	−0.0082	0.08

Table 13: Stage 2 ablation: re-training all components with vanilla MARBERTv2 vs. Stage 2 contrastive backbone (3 seeds, paired t-test). Phrase classification and untuned SEM show no significant difference ($p \in \{0.40, 0.80, 0.89\}$); boundary metrics and tuned SEM show only marginal differences ($|\Delta|<0.01$). The polarity-aware calibration gate ablation is reported separately in Section 6.4 ($\Delta\rho$=+0.0231, paired-bootstrap 95% CI $[-0.007, +0.053]$, not significant).

word overlap. *Semantic features* are embedding cosine similarity, L2 distance, and mean/max of the embedding difference vector. Phrase features additionally include the `a_is_none`/`b_is_none` indicators for unaligned phrases. The polarity gate additionally consults Arabic WordNet (AWN) antonym lookups and Farasa (Abdelali et al., 2016) morphological distance.

Environment and cost. Training uses a single NVIDIA RTX 3090 (Python 3.11, PyTorch 2.4.1+cu124, LightGBM 4.x). Inference runs on CPU at ≈50 ms/pair; full training is a one-time ≈25 minutes. All hyperparameters are tuned on dev; the test set is used only for final evaluation.

J Human Evaluation of Interpretability

Intrinsic joint-F1 measures whether predicted traces match the gold annotation, but not whether a human reader finds them *usable*. To assess this directly, two trained linguists (not involved in system development) evaluated the predicted phrase- and word-level traces on a stratified sample of 100 test pairs, balanced across the four SEM classes (≈25 pairs each) so that the minority classes are not drowned out by the dominant `SEM-PARTIAL` class.

Protocol. For each pair, the two judges independently rated the interpretive trace on two five-point scales: *correctness* (is the trace linguistically right?) and *helpfulness* (does it help a reader see why the two sentences match or differ?). Predicted and gold traces were shown *blind and unlabeled*, so a judge could not tell which one the system produced; this also yields the gold trace as a natural upper bound. Inter-annotator agreement was κ = 0.65 (substantial), consistent with our original annotation agreement (SEM κ = 0.83; phrase/word 0.79/0.77).

Results. Table 14 reports mean ratings. The predicted traces are rated *useful and mostly correct* (system correctness 3.0, helpfulness 3.15 out of 5), with a consistent gap below the gold traces (−1.1 overall on correctness; −1.2 at the phrase level and −1.0 at the word level). Helpfulness slightly exceeds correctness, indicating that a trace can help a reader localise the relation even when it is not structurally perfect. These ratings agree with the intrinsic scores (phrase/word joint-F1 in Appendices A and B): the interpretive layers provide a useful signal without yet reaching human-level trace quality. We therefore treat trace usability as a genuine but partial strength, and close the human-level gap as

an open direction (Limitations).

Level	Sys. corr.	Sys. help.	Gold corr.	Gap
Phrase	3.0	3.3	4.2	−1.2
Word	3.0	3.0	4.0	−1.0
Overall	**3.0**	**3.2**	**4.1**	−1.1

Table 14: Human evaluation of interpretive traces on a balanced 100-pair sample (≈25 per SEM class), two linguist judges, $\kappa = 0.65$. Ratings are 1–5 for system-predicted (Sys.) and gold traces, shown blind. *Gap* is system minus gold correctness.

K Ethics, Data, and Annotator Statements

K.1 Data Statement

Following Bender and Friedman (2018), we report key data properties.

Curation rationale. The corpus targets Arabic interpretable Semantic Textual Similarity (iSTS) at sentence, phrase, and word levels. Source sentences are drawn from several public Arabic sources: movie and television subtitles (Lison and Tiedemann, 2016), product reviews, encyclopaedic articles from Wikipedia, and scientific texts. Pairs were selected to populate four SEM tags with explicit attention to the `SEM-DIFF` and `SEM-IDENT` minority classes; curated paraphrase pools and adversarial near-paraphrase distractors were added; the resulting training distribution (`SEM-IDENT` 4.6%, `SEM-PARA` 32.3%, `SEM-PARTIAL` 56.4%, `SEM-DIFF` 6.7%) is less skewed than the natural test distribution but still `SEM-PARTIAL`-dominated.

Language variety. Modern Standard Arabic (MSA) is the dominant variety across sources (reviews, encyclopaedic, and scientific text are predominantly MSA; subtitles are largely MSA with occasional dialectal lexical items). The corpus is not stratified by Arabic dialect, and dialectal coverage (Egyptian, Levantine, Gulf, Maghrebi) is unverified (Limitations §2).

Speaker / annotator demographics. Source-text authors are not identifiable: subtitles are translated screen text, and user-generated review text was de-identified before annotation. Three trained annotators participated; all are adult native speakers of Arabic with university-level training in Arabic linguistics, of Saudi and Egyptian backgrounds. The senior reviewer is a faculty linguist.

Text characteristics. Mean sentence length: 8.4 tokens (range 2–47). Genre: mixed, comprising conversational (subtitles), user-generated (product reviews), encyclopaedic (Wikipedia), and scientific Arabic. All source text was collected from publicly accessible sources; user-generated text was de-identified, and the released corpus contains no personally identifying information.

K.2 Annotator Compensation and Working Conditions

Annotators were compensated at a rate aligned with senior research-assistant pay scales in the host institution, well above the local minimum wage and consistent with prior Arabic-NLP annotation projects of comparable difficulty. Annotation guidelines were provided in advance (100+ page bilingual guide), with iterative feedback over multiple pilot batches before production annotation. All annotators were full participants in schema refinement and could flag disputed items for adjudication.

K.3 Ethics Statement

The corpus is assembled from publicly available text across the sources above. No human-subjects data, medical records, or sensitive personal information are involved, and user-generated material was de-identified. Each source retains its upstream license (e.g., Wikipedia under CC BY-SA; subtitles under the OpenSubtitles terms); the released subset and audit scripts are distributed for non-commercial academic research, and redistribution of any source text follows the most restrictive applicable upstream terms; access to the full corpus is governed by the same terms (Section K.4).

Potential dual-use considerations. Interpretable Semantic Textual Similarity systems can support beneficial applications (information retrieval, plagiarism detection, machine-translation quality estimation) and could in principle be used for content monitoring at scale. Our schema and system do not enable any capability not already supplied by general-purpose Arabic NLP toolkits, and we view the resource contribution (annotated benchmark, open code) as net-positive for the Arabic NLP research community.

Use of AI assistance. We used a large language model for editorial polish of the English prose; all research design, experiments, data construction, annotation, results computation, and scientific claims are the authors' own work, and the authors take full responsibility for all content of the paper. All numerical results are computed from artefacts in our experiment runs directory and are reproducible from the released code.

K.4 License and Access

KASHAF's guidelines, code, models, pipeline, a corpus sample, and the audit script (Section 3) are released under a non-commercial research license. The full annotated corpus is hosted by the King Salman Global Academy for the Arabic Language (KSAA) and is available to researchers on request, free of charge, for non-commercial research under a data-use agreement; distribution remains subject to the upstream terms of each source (Section K.3).

QUANTA: A Fine-Grained Dataset and Benchmark for Arabic Machine-Translation Error Analysis*

Afraa Alshammari[1], Ahmad Alahmari[3], Amal Alaboud[4], Jaber Ahmed Harthi[3], Karima Ali Shahwan Alzahrani[5], Ahnaf Adib[1], Patrick T. Brandt[2], Javier Osorio[6], Vito D'Orazio[7], Latifur Khan[1]

[1]Department of Computer Science, University of Texas at Dallas; [2]School of Economic, Political and Policy Sciences, University of Texas at Dallas; [3]Department of Foreign Languages, Jazan University; [4]Department of Foreign Languages, Taif University; [5]Independent Researcher; [6]School of Government and Public Policy, University of Arizona; [7]Department of Political Science, West Virginia University

Afraa.Alshammari@utdallas.edu, Aalahmari@jazanu.edu.sa, Asaboud@tu.edu.sa, Jharithi@jazanu.edu.sa, Kalzhrani153@gmail.com, Ahnaf.Adib@utdallas.edu, Pbrandt@utdallas.edu, Josorio1@arizona.edu, Vito.dorazio@mail.wvu.edu, Lkhan@utdallas.edu

Abstract

Automatic machine-translation (MT) metrics collapse heterogeneous errors into a single score. For Arabic, no fine-grained resource exists for diagnosing error types and severity. We introduce QUANTA, a 9,032-instance En to Ar MT-error dataset (738 real UNPC + 7,856 synthetic training instances, both structurally validated, plus a held-out 438-instance real Gold test set) annotated with MIZAN (ميزان, "scale"), a new 23-class MQM-compatible taxonomy organised in a four-level hierarchy: 23 leaf classes rolling up to 14 subtypes, 5 parent groups, and 4 TQA dimensions. To make the resource usable, we release the RAQEEB (رقيب, "observer") classifier, an AraBERT-v2 model for predicting MIZAN 23-class labels (5-fold Gold macro-F1 $= 0.614 \pm 0.017$). Data quality is supported by ETCA (Error Taxonomy Cross-vendor Audit), an LLM-as-judge check in which a different-vendor model audits QUANTA instances for anchor validity and label fidelity, curbing same-family self-preference bias, and by dual-annotator validation (Gwet's AC1 $=$ 0.97 synthetic; 0.83 real) on core taxonomy criteria. Ablations show that anchor-guided framing is decisive (without it, macro-F1 collapses to 0.114); that 738 structurally validated real samples are $\approx 12\times$ more sample-efficient than synthetic data; and that, restricted to the classes with real-tier training support, combining both tiers outperforms real (0.656) or synthetic (0.312) alone. Code, data, prompts, and checkpoints are publicly available.

1 Introduction

Machine Translation (MT) systems are now deeply embedded in information access, localization, and cross-lingual Natural Language Processing (NLP) pipelines, making the reliability of MT evaluation a central concern (Mukherjee and Shrivastava, 2025). MT evaluation is constrained by a trade-off between linguistic precision and scalability: human evaluation of MT output is generally considered accurate but is costly to scale, whereas automatic MT quality metrics (BLEU (Papineni et al., 2002), COMET (Rei et al., 2020), BERTScore (Zhang et al., 2020)) tend to collapse heterogeneous error phenomena from minor orthographic deviations to critical semantic distortions into a single opaque value. Compressing different MT error types into a single scalar both undercounts catastrophic semantic errors and overcounts cosmetic ones, a distinction a fine-grained, severity-aware taxonomy can make explicit. Properly assessing the quality of MT outputs is especially difficult for Arabic: its rich morphology (nunation, the definite article ال, strict gender agreement, agglutinative clitics) produces error phenomena absent from European-language MT evaluation (Al-Said, 2026; Habash, 2010), yet prior typologies (Munkova et al., 2025; Qiu et al., 2025) are language-agnostic at 3 to 7 error classes (Tafa et al., 2025). No fine-grained Arabic MT error benchmark with 20 or more classes exists.

QUANTA and MIZAN in one sentence. We release QUANTA, a fine-grained Arabic MT error dataset with full 23-class coverage, and MIZAN (ميزان, "scale"), its underlying multi-scale taxonomy of Arabic linguistic-shift errors, together with a released 23-class diagnostic classifier and a dual-annotator keyword-level validation protocol reaching almost-perfect inter-annotator agreement

*Code and data available at https://github.com/AALight/Quanta-Mizan

Proceedings of The Fourth Arabic Natural Language Processing Conference (ArabicNLP 2026), pages 462–484
October 28-29, 2026 ©2026 Association for Computational Linguistics

(Gwet's AC1 0.83–0.97) on the core taxonomy criteria. MIZAN is compatible with the Multi-dimensional Quality Metrics (MQM) framework: it adopts MQM's severity typology and rolls its 23 classes up to four Translation Quality Assessment (TQA) dimensions, but departs from MQM by adding four Arabic-specific leaf classes (Tanween Omission, Gender Disagreement, Definiteness Shift, Invalid Pattern) for which language-agnostic MQM provides no first-class error type. To keep the components distinct, the paper names four: **MIZAN** (the 23-class taxonomy), **QUANTA** (the dataset built and audited under it), **ETCA** (the cross-vendor LLM audit that validates it), and **RAQEEB** (the released classifier trained on QUANTA); Figure 1 is the overview, tracing raw MT output through anchor-constrained construction and validation to the QUANTA dataset.

Contributions. Together these form one integrated resource for fine-grained En to Ar MT error analysis, shown end-to-end in Figure 1: the MIZAN taxonomy, the QUANTA dataset built and audited under it, and the RAQEEB diagnostic classifier:

1. **MIZAN**: a 23-class taxonomy of Arabic linguistic-shift errors with a 4-level hierarchy (23 classes → 14 subtypes → 5 parent groups → 4 MQM-compatible TQA dimensions) and expert-calibrated severity weights.
2. **QUANTA**: a 9,032-instance fine-grained Arabic MT error dataset, applying the MIZAN error typology. QUANTA includes 738 expert-annotated *real* MT errors (tier T1) from the United Nations Parallel Corpus (UNPC) (Ziemski et al., 2016), 7,856 structurally validated *synthetic* instances (tier T2) covering all 23 classes, and a 438-instance Gold test set with full 23/23 coverage.
3. **ETCA** (Error Taxonomy Cross-vendor Audit): a quality audit in which a different-vendor LLM (Claude Sonnet 4, Anthropic, 2025) independently scores both the high-confidence generation seeds and the GPT-4o-generated T2 candidates (OpenAI, 2024); for the synthetic tier this means no model grades its own output, mitigating same-family self-preference bias (Kim, 2025; Wang et al., 2026). Verbatim prompts ship with the release (Appendix D); external validation against the human consensus (Gwet's AC1 = 0.62) is in §4.2.
4. **RAQEEB**: a fine-tuned AraBERT-v2 classifier that labels each MT error with its MIZAN class, released with the 5-encoder RAQEEB-Bench benchmark (5-fold mean Gold macro-F1 = 0.614 ± 0.017) as a diagnostic tool with documented per-class capability boundaries, not a sealed quality scorer.
5. **Human validation**: two expert linguists, not involved in taxonomy design, independently re-labelled 147 stratified instances, confirming that the MIZAN labels are reproducible (Gwet's AC1 = 0.97 on the core criteria for T2, 0.83 for T1).

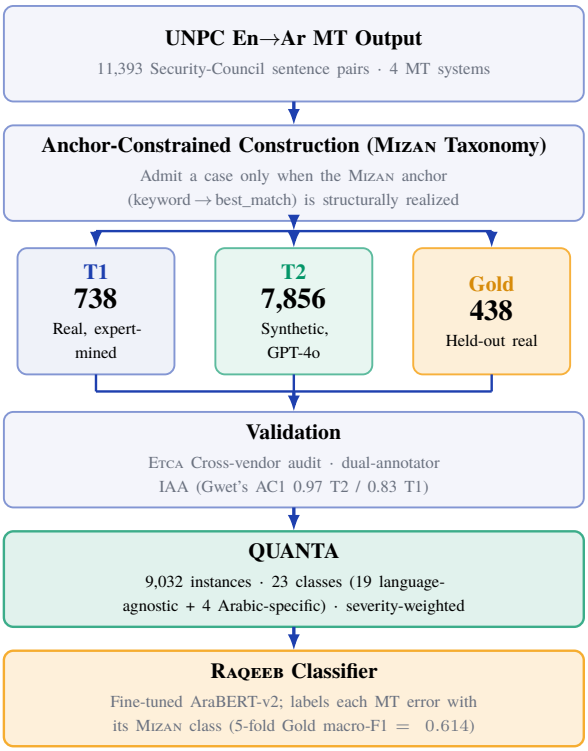


Figure 1: The QUANTA pipeline: from raw En→Ar MT output through MIZAN-guided anchor-constrained construction and validation (ETCA cross-vendor audit and dual-annotator IAA) to the QUANTA dataset and the released RAQEEB classifier.

2 Related Work

Arabic MT and scalar evaluation metrics. Arabic MT benchmarks (Sajjad et al., 2020; Nagoudi et al., 2022; Deutsch et al., 2025) and scalar metrics BLEU (Papineni et al., 2002), COMET (Rei et al., 2020, 2022), BERTScore (Zhang et al., 2020) are dominant in MT quality assessment. Despite their prevalence, they produce a single scalar per sentence and cannot decompose errors into linguistically meaningful categories. This is an important limitation when evaluating Arabic. Arabic morphological complexity (Habash, 2010) introduces phenomena (tanween,

definiteness, gender agreement, root-pattern violations) that scalar metrics fail to isolate. No prior work provides a fine-grained taxonomy with 20+ classes targeting Arabic-specific phenomena in MT.

Fine-grained MT error taxonomies. The MQM framework (Lommel et al., 2014, 2024) is the dominant reference for MT error annotation. Freitag et al. (2021) showed that MQM-based human evaluation correlates more strongly with professional translators' judgments than do scalar ratings. While earlier taxonomies introduced 5-class schemes (Vilar et al., 2006; Popović, 2011), more recent LLM-as-judge MQM variants include RUBRIC-MQM (Kim, 2025) and PPbMQM (Wang et al., 2026), both reporting LLM self-preference bias. For Arabic, the QALB shared task (Zaghouani et al., 2014; Mohit et al., 2014) targets learner grammar, not MT; and the critical-error taxonomy of Al Sharou and Specia (2022) is severity-oriented and cross-lingual rather than Arabic-specific. Closer to our setting, AMEANA (El Kholy and Habash, 2011) performs automatic error analysis of Arabic MT output, but as a morphological-feature diagnostic rather than an MQM-compatible, severity-weighted error taxonomy. For Arabic grammatical and learner-error annotation, ARETA (Belkebir and Habash, 2021) provides automatic error-type annotation, the Arabic Learner Corpus (Alfaifi, 2015) supplies expert learner-error annotation, and ArabiGEE (Elhady et al., 2026) offers a hierarchical error-explanation taxonomy; all three address native or learner text correction rather than translation errors. None of these is an MQM-compatible, severity-weighted MT-error taxonomy expressing tanween, definiteness, gender, and root-pattern phenomena at 20+ classes with token-level verifiable anchors, as Mizan provides. Finally, Lommel et al. (2024) argue that segment-level variance on fine-grained linguistic attributes is inherent in language; we draw on this in §5 to frame the lower-F1 classes our classifier reports as field-wide hard cases rather than classifier-specific failures.

Arabic-specific diagnostic evaluation. Osorio et al. (2024, 2025) benchmark domain-specific encoders on native vs. machine-translated UNPC text (Ziemski et al., 2016) and establish a statistically significant rarity loss in MT output that maps to our Terminology Substitution and Hypernym for Hyponym classes, plus dependency-distance shifts that map to the Syntactic Shifts parent group. Al-Olimat and Alshareef (2026) (ALPS) show that expert-curated diagnostic sets remain useful even when single-item inter-rater agreement is low, the same pattern we find in our own annotation (§4.3): near-perfect agreement on the core taxonomy criteria but lower agreement on more subjective judgments.

Synthetic data and validated error construction. LLM-generated training data (Schick and Schütze, 2021; Wang et al., 2023) and data-centric fine-tuning (Zha et al., 2023; Gao et al., 2021) can match or exceed larger noisy datasets, but the synthetic-to-real distributional gap is well documented (Chen et al., 2023). Validation frameworks specifically for synthetic MT-error data remain scarce: generic filtering does not guarantee that a claimed error actually occurs at a specific lexical anchor (Northcutt et al., 2021). The anchor-constrained validation pipeline that produced Quanta fills this gap: it admits a synthetic error only when the claimed error is structurally realised at its lexical anchor, with the cross-vendor Etca audit as an independent second check.

3 The Mizan Taxonomy and Multi-Scale Framework

Motivation and design. General-purpose MT error taxonomies such as MQM (Lommel et al., 2014, 2024), Vilar (Vilar et al., 2006), and Hjerson (Popović, 2011) are language-agnostic 3–7-class schemes with no Arabic-specific morphological phenomena as first-class error types. **Mizan** (ميزان, "scale") fills this gap. Following the "two-pillar" view of Lommel et al. (2024), Mizan pairs a 23-class fine-grained error typology, organised as a 4-level hierarchy (23 classes → 14 subtypes → 5 parent groups → 4 MQM-compatible TQA dimensions), with expert-calibrated, MQM-compatible severity weights. Four design principles distinguish Mizan from these general-purpose taxonomies:

1. **Computational verifiability**: each class is defined by a specific keyword→best_match transformation verifiable at the token level (e.g., *Hypernym for Hyponym*: المحكمة الجنائية → المحكمة).
2. **Arabic-specific coverage**: four classes capture phenomena absent from MQM (Tanween Omission, Gender Disagreement, Definiteness Shift, Invalid Pattern).

Taxonomy	#Cls	Anchor	AR-spec.	Score
Vilar (Vilar et al., 2006)	5	×	×	×
Popović (Popović, 2011)	5	word	×	×
MQM Core (Lommel et al., 2014)	7	segment	×	✓
MQM Full (Lommel et al., 2024)	∼70	segment	×	✓
QALB (Zaghouani et al., 2014)	∼14	✓	grammar	×
GEMBA-MQM (Kocmi and Federmann, 2023)	22	span	×	MQM
PPbMQM (Wang et al., 2026)	top	span	×	MQM
Mizan (ours)	**23**	**anchor**	**4 cls**	**MQM**

Table 1: Mizan compared with major MT error taxonomies. #Cls: number of classes; Anchor: span-localisation granularity; AR-spec.: Arabic-specific classes; Score: severity/scoring scheme.

3. **MQM-compatible hierarchy**: a deterministic mapping from 23 classes to MQM-compatible TQA dimensions.
4. **Operational granularity**: the 23 leaf classes emerged bottom-up from expert-linguist analysis of 446 annotated UNPC MT-error instances rather than top-down.

Table 1 positions Mizan against prior taxonomies. The full per-class specification (definitions, severity, weights, parent group, TQA category) plus a per-class table of real Gold-test examples with Sentence_ID, MT-tool attribution, Arabic reference and MT-output excerpts, and expert justifications for every one of the 23 classes is in Appendix A.

Multi-scale framework. Mizan is an expert-calibrated, multi-scale taxonomy for diagnosing En→Ar MT errors. It integrates four operating dimensions: 23 anchor-observable classes, 14 subtypes, 5 linguistic-family parent groups, and 4 MQM-compatible TQA dimensions matching the MQM Core / MQM Full distinction of Lommel et al. (2014, 2024). Rendered as a tree in Figure 3 (Appendix A), this hierarchy lets a keyword-level error be traced upward through its parent group to the evaluation dimension it affects, the multi-scale property our classifier exploits through its L2 and L3 rollups (§5). The taxonomy was designed by two Arabic native-speaker researchers with doctoral qualifications (one in computational linguistics, one in translation). Instance-level dual-annotator validation is conducted independently by two further expert Arabic linguists, neither of whom participated in taxonomy design (§4.3).

Severity weights. Mizan's severity weights are assigned per class by the two expert linguists following MQM's severity typology (Minor to Critical) (Lommel et al., 2024). The per-class weights aggregate by parent group and by TQA category to 1.00 (Table 12); these totals are aggregations of the per-class weights, not separate scoring factors. These MQM-compatible weights supply the severity-penalty ingredient for segment-level quality scoring; computing such a score is out of scope here, where §5 instead reports the classifier's performance over the Mizan classes. The full per-class weight table and the rollup mapping are in Appendix A.

Why this matters: MT-quality stakes in formal-institutional text. The 23 Mizan classes capture MT failure modes that other metrics collapse into a single opaque value. Mizan makes each failure explicit by assigning it a class and a per-class severity weight s_i (examples per parent group in Table 2): النيجر بلد → النيجر منطقة is labelled Meaning Shift ($s_i = 5$), recategorising a state as a substate region; يهيب → يدعو is labelled Terminology Substitution ($s_i = 4$), weakening a binding Security Council directive ("calls upon" → "invites"); and إدانته → إدانتها is labelled Gender Disagreement ($s_i = 3$), an Arabic-specific agreement error that flips a possessive pronoun to feminine. Because each class carries its own severity weight, these failures stay distinct rather than averaged into one scalar, as high-stakes legal, medical, or diplomatic translation requires.

4 The Quanta Dataset

Quanta is the anchor-validated, Mizan-labelled dataset of En→Ar MT errors. It combines three complementary parts: a small set of real errors mined from UNPC translations (T1), a large synthetic set that fills every error class (T2), and a held-out expert-annotated Gold test set covering all 23 classes. Table 3 summarises the overall composition, and Table 4 the per-parent-group coverage. T1+T2 forms the training pool (8,594 instances); Gold (438 instances) is held out for evaluation with full 23/23 class coverage. Each released instance carries six fields, `Sentence_ID`, `Keyword`, `Best_Match` (or the sentinel `[OMITTED]`), `Sub-Subtype` (one of 23), `Parent group` (one of 5), and `TQA category` (one of 4); a worked example for each parent group is in Table 2.

Parent group	Class	Keyword → Best_Match	MT-quality impact (UN-domain gloss)	s_i
Semantic Div.	Total Omission	بعثة → [OMITTED]	UNOCI mandate: "mission" dropped, peacekeeping referent disappears.	5
Semantic Div.	Meaning Shift	النيجر بلد → النيجر منطقة	"Niger, a country" → "Niger, a region": the referent's category changes, a state rendered as a substate area.	5
Morphosyntactic	Gender Disagreement	إدانته → إدانتها	"its [the Council's] condemnation of violence against women and children": MT flips the possessive pronoun to feminine, breaking agreement.	3
Syntactic	Perfective to Progressive	حث → يحث	"urged" (past) → "urges" (present operative): legal force of resolution altered.	4
Pragmatic Div.	Terminology Substitution	يدعو → يهيب	"calls upon" (formal SC directive) → "invites" (weaker request).	4
Orthographic	Spelling Error	اربعة → أربعة	"four" written without its hamza (أربعة → اربعة): orthographic typo in a casualty count.	1

Table 2: Representative MIZAN examples from real En→Ar UNPC MT output, one per parent group plus a Total Omission; s_i is the per-class MIZAN severity weight (1 minor to 5 critical).

Split	Instances	Classes	Provenance
T1 (real)	738	12/23	UN expert
T2 (synthetic)	7,856	23/23	GPT-4o; structural audit
T1+T2	8,594	23/23	training
Gold test	438	23/23	held-out

Table 3: QUANTA Dataset Composition.

Per-parent-group class coverage. Table 4 breaks the 23 classes down by parent group, separating those that surface in real UNPC MT output from those filled for training by T2 synthesis (§4.1). Of the 23 classes, 15 appear in real MT and 8 are T2-only, while all 23 have Gold coverage. Orthographic Shifts has zero real-MT coverage, consistent with character-level errors being removed by post-editing in formal institutional translation, so this group is supplied entirely by synthesis.

Parent group	#Cls	Real-MT	T2-only	Gold
Semantic Divergence	7	4	3	✓
Morphosyntactic Shifts	4	3	1	✓
Syntactic Shifts	9	6	3	✓
Pragmatic Divergence	2	2	0	✓
Orthographic Shifts	1	0	1	✓
Total	23	15	8	23/23

Table 4: Per-parent-group class coverage in QUANTA.

4.1 Dataset Construction

Building a fine-grained error dataset poses a reliability problem (Lommel et al., 2014, 2024; Al-Olimat and Alshareef, 2026): real MT errors at any given MIZAN class are sparse, while synthetic errors are easy to generate but hard to trust, since the injected error may not land at the intended location (Kocmi and Federmann, 2023; Kocmi et al., 2024; Wang et al., 2026). QUANTA addresses this with an anchor-constrained validation pipeline whose core gate admits a candidate only when its MIZAN anchor is structurally realised: the keyword is present in the Arabic reference and its best_match substitution (or the sentinel [OMITTED] for omissions) appears in the MT output (Figure 1), paired with the cross-vendor ETCA audit (§4.2) as a second, independent control. The pipeline runs on the 11,393 UN Security Council sentence pairs of Osorio et al. (2025) across four MT systems.

The anchor keywords come from domain-contrastive keyness (Sketch Engine's simple-maths score against UNPC (Kilgarriff, 2009; Ziemski et al., 2016)), yielding 600 diagnostic keywords that two expert linguists extend with manually curated multi-word anchors before searching them across the four MT outputs.

From the 446-instance expert gold pool the pipeline extracts 177 unique keyword→best_match patterns; applying these across the corpus and deduplicating yields 2,770 candidate rows, of which 1,667 (60%) are rejected as structurally noisy and 794 pass at high confidence to form the T1 substrate, which bypasses ETCA and serves as the reliability anchor; removing the 56 that overlap the held-out Gold test set yields the final 738 T1 instances. Gold-overlap removal eliminates three classes entirely from the T1 training pool (Literal Translation, Perfective to Progressive, Tense Shift Under Negation), which are covered in training only through T2 that is generated using gold samples. The eight MIZAN classes that the pattern-based retrieval does not surface in the T1 pool (Meaning Shift, Partial Translation, Name Entity Error, Gender Disagreement, Wrong Structure, Active to Passive Voice, Noun to Adjective, Spelling Error) are filled for training by T2 synthesis under the same

anchor constraint, so 23/23 training coverage is a pipeline property rather than a retrieval property. These classes are not absent from real MT: all eight are attested in the held-out Gold test set, which is drawn entirely from real MT output (Meaning Shift alone contributes 70 instances), so every class is evaluated on real data even when its training signal is synthetic.

4.2 ETCA Cross-Vendor Audit

Same-family LLM-as-judge pipelines such as GEMBA-MQM (Kocmi and Federmann, 2023) suffer self-preference bias, where a judge systematically favours its own family's generations; routing generation and judging through different vendors is the general mitigation established by prior LLM-as-judge work (Kim, 2025; Wang et al., 2026). Our **Error Taxonomy Cross-vendor Audit (ETCA)** is a specific instantiation of this cross-vendor principle for taxonomy-grounded error auditing, so that no single model grades its own output: GPT-4o (OpenAI, 2024) generates synthetic candidates with 3-shot prompting seeded from a 79-row canonical T0 pool, and Claude Sonnet 4 (Anthropic, 2025) (a different vendor) scores each candidate on five dimensions: Anchor Validity, Phenomenon Clarity, Arabic Naturalness, Collateral Severity, and Seed Recommendation, aggregated into a **Data Quality Index (DQI)**, the normalised mean of the four substantive dimensions (excluding Seed Recommendation). ETCA runs on the 794 high-confidence generation seeds and on a stratified 414-instance T2 audit subset ($\approx 5\%$ of T2, spanning all 23 classes), where DQI ≥ 0.75 marks a high-confidence instance; the released 7,856-instance T2 split is selected by the structural re-classification audit (§4.4), not by a per-instance DQI threshold. The verbatim generation prompt and audit rubric are in Appendix D. ETCA's judgements align with the expert human consensus: on a 43-instance high-confidence subset adjudicated against the dual-annotator labels (§4.3), ETCA reaches Gwet's AC1 $= 0.62$ (substantial agreement on the Landis–Koch scale). This validates the auditor against expert human judgment rather than against another auditor LLM, whose agreement can be inflated by shared model biases; alignment with the human consensus is the stronger criterion for audit reliability. We therefore use ETCA strictly as a parallel dataset-quality audit rather than a stand-alone quality gate.

4.3 Dual-Annotator Validation

QUANTA is validated by two expert Arabic native-speaker linguists, who were not involved in designing the MIZAN taxonomy and who labelled each instance independently. On a stratified sample of 147 instances (55 T1, 92 T2), they reach Gwet's AC1 (Gwet, 2008) $=$ 0.97 on T2 and 0.83 on T1 across the three core taxonomy criteria (error existence, anchor identification, MIZAN class label), almost-perfect agreement on both tiers on the Landis–Koch scale (Landis and Koch, 1977). We use Gwet's AC1 as the primary statistic because the binary evaluation criteria have high prevalence by design, a regime in which Cohen's κ collapses artifactually (Feinstein and Cicchetti, 1990; Gwet, 2008).

IAA in the context of fine-grained linguistic annotation. Fine-grained linguistic annotation produces a characteristic IAA pattern: expert linguists agree near-perfectly on core taxonomy questions and diverge on more subjective attributes such as register fit and severity calibration. Lommel et al. (2024), in the 10th-anniversary MQM review, argue that segment-level variance on fine-grained linguistic attributes is inherent in language rather than a technical deficiency, because text admits multiple valid interpretations and error categorisation depends on the reader's background. Al-Olimat and Alshareef (2026) corroborate this for Arabic: their expert-curated ALPS diagnostic set reports pairwise Cohen's $\kappa = 0.23$ on individual items, while their adjudicated oracle labels reach 99.2% agreement and average human accuracy of 84.6%. Against that backdrop, our IAA is strong on the core taxonomy (T2 core-three mean $= 0.966$, T1 $= 0.827$) and patterned on the peripheral criteria as expected; the lower scores on subjective attributes reflect inherent reader-interpretation variance rather than a validation failure. Appendix B gives the full per-criterion table (severity, UN-register fit, aggregate means), the stratified-sampling design, and the inter-annotator-agreement protocol.

4.4 Post-Hoc Corrections

A post-hoc audit of the Partial Translation class surfaced a mis-labelling pattern: 188 candidates were excluded where the apparent "omission" was in fact a full translation of a different source word; 15 were relabelled from Partial Translation to Total Omission; 20 received corrected `Best_Match`

Encoder	Mean F1 ± std
AraBERT v2	**0.614 ± 0.017**
CAMeLBERT-Mix	0.597 ± 0.021
CAMeLBERT-MSA	0.587 ± 0.029
ConfliBERT	0.583 ± 0.021
XLM-R	0.565 ± 0.023

Table 5: 5-encoder benchmark on the 23-class MIZAN task (anchor-guided input, identical folds); mean ± std over 5 folds.

spans. The final counts in Table 3 reflect these corrections.

5 The RAQEEB Classifier

We release the RAQEEB classifier as a diagnostic evaluation tool with documented per-class capability boundaries, not as a one-number quality scorer. AraBERT v2 (Antoun et al., 2020) reaches 5-fold mean Gold macro-F1 $= 0.614 \pm 0.017$ on the 23-class MIZAN task; we report this transparently and partition the 23 classes into three operational tiers (Table 6) that scope how the classifier should be used.

Encoder comparison. Five encoders ingest anchor-guided input at a fixed learning rate of 5×10^{-5}: three Arabic-specific (AraBERT v2 (Antoun et al., 2020), CAMeLBERT-MSA, CAMeLBERT-Mix (Inoue et al., 2021)), one domain-adjacent (ConfliBERT (Hu et al., 2022)), one multilingual (XLM-R (Conneau et al., 2020)). All five cluster within a 0.049 macro-F1 spread (0.565–0.614; Table 5). We test every encoder pair two ways (Appendix G): a paired-bootstrap comparison with Benjamini–Hochberg false-discovery-rate (FDR) correction asks whether two encoders differ, and a Two One-Sided Tests (TOST) equivalence check ($\delta = 0.05$) asks whether they are practically the same. AraBERT v2 significantly outperforms ConfliBERT ($p_{adj} < 0.001$) and XLM-R ($p_{adj} = 0.007$), and CAMeLBERT-Mix significantly outperforms XLM-R ($p_{adj} = 0.015$); the three Arabic-specific encoders are mutually equivalent within ± 0.05 macro-F1. Arabic-specific pretraining thus provides a small but measurable advantage over the multilingual and domain-adjacent alternatives; ConfliBERT's political-conflict pretraining does not transfer to fine-grained error classification.

Tier reading. The Top tier (F1 $\geq$ 0.80) spans structural and morphological phenomena and, notably, the highest-severity semantic class, Total Omission (0.967), so the most damaging error type is among the most reliably detected, not the least. The Mid tier (0.50–0.80) covers classes that are reliable on aggregate macro-F1 but on which individual predictions still warrant confirmation. The Frontier tier (F1 $<$ 0.50) is dominated by semantically confusable class pairs that share a surface substitution pattern (Meaning Shift with Register Mismatch; Hypernym for Hyponym with Terminology Substitution); discriminating them needs richer lexical-semantic signal than the surface anchor pair carries (the relation between keyword and best_match), a natural extension of the anchor framing. These are field-wide hard cases, not classifier-specific failures: fine-grained schemes with many confusable classes carry a known per-class F1 penalty (Mekala et al., 2021; Suresh and Ong, 2021), and the underlying segment-level variance is intrinsic to language rather than a technical deficiency (Lommel et al., 2024; Al-Olimat and Alshareef, 2026). Production use should treat Frontier classes as candidates for human review.

Tier	Range	Classes (best fold F1)
Top	F1 $\geq$ 0.80	Total Omission (0.967), Wrong Word Order (0.960), Definiteness Shift (0.923)
Mid	0.50 $\leq$ F1 $<$ 0.80	Tanween Omission (0.788), Invalid Pattern (0.696), Register Mismatch (0.649), Literal Translation (0.526)
Frontier	F1 $<$ 0.50	Terminology Substitution (0.382), Meaning Shift (0.272), Hypernym for Hyponym (0.160)

Table 6: Operational tiers of the released RAQEEB classifier on the 10 MIZAN classes with $n \geq 10$ Gold support (full table: Appendix E).

Multi-level rollup. The MIZAN hierarchy is learnable at every granularity (Table 7), and not monotonically in the number of classes. The 14-subtype level (L3) is highest: we hypothesise an L3 learnability sweet spot, where merging confusable siblings that share a keyword→best_match transformation (by MIZAN's design) sheds the collisions that depress L4 without lumping the heterogeneous anchors that L2 does, consistent with evidence that the finest resolution is not automatically optimal and the best granularity is data-dependent (Pirovano et al., 2025; Chen et al., 2025). The wide L2 interval (± 0.062, overlapping L3) makes this ordering suggestive rather than significant.

Anchor framing and data efficiency. Two ablations isolate the headline signal (full details in Appendix H). Anchor framing: without the anchor pair the classifier collapses to macro-F1 $= 0.114$

Level	Granularity	Macro-F1
L4	23 leaf classes	0.614 ± 0.017
L3	**14 subtypes**	$\mathbf{0.676 \pm 0.021}$
L2	5 parent groups	0.624 ± 0.062
L1	4 TQA dimensions	0.656 ± 0.022

Table 7: MIZAN hierarchy learnability: AraBERT v2 macro-F1 (5-fold) at each rolled-up granularity.

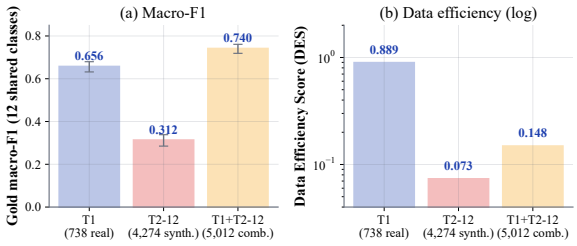


Figure 2: Data efficiency on the 12 T1-supported MIZAN classes (cf. Table 8): (a) 12-class Gold macro-F1 and (b) Data Efficiency Score (DES, log scale) for T1 (real), T2 (synthetic), and T1+T2. T1's DES of 0.889 versus T2's 0.073 is the $\approx 12\times$ sample-efficiency advantage of real data.

(paired-bootstrap $p < 0.0001$; controlled ablation, Appendix H), so anchor-guided framing is the essential signal for fine-grained Arabic MT error classification. Data efficiency: restricted to the 12 classes with T1 support, 738 structurally validated real samples reach macro-F1 = 0.656 versus 0.312 for the 4,274 synthetic instances in those classes, our Data Efficiency Score (DES, F1 per 1,000 samples; Eq. 2) of 0.889 vs. 0.073 ($\approx 12.2\times$); T1+T2 dominates either alone (= 0.740 on 5,012; non-overlapping 23-class confidence intervals), as visualized in Figure 2. Baselines confirm the task is not solvable off-the-shelf (Table 9): classical TF-IDF features cap at macro-F1 = 0.237 and the Arabic LLM ALLaM-7B (Bari et al., 2024) reaches only 0.050 zero-shot and 0.098 five-shot, against 0.614 for the fine-tuned anchor model.

Source	n	12-cls macro-F1	DES
T1 (real)	738	**0.656**	**0.889**
T2 (synthetic)	4,274	0.312	0.073
T1+T2	**5,012**	**0.740**	**0.148**

Table 8: Data ablation on the 12 T1-supported MIZAN classes: 12-class macro-F1 and DES (AraBERT v2, Format D, 5-fold CV).

Frontier-tier error analysis. The Frontier tier has structure: Meaning Shift reaches perfect precision (≈ 1.0) but recall 0.157 (11/70), its false negatives absorbed by semantically adjacent classes such as Register Mismatch, which acts as a catch-all for ambiguous semantic predictions (full distribution in Appendix E). Meaning Shift moreover has no real T1 training data, so its low recall partly reflects a synthetic-train / real-test gap that more real annotation would narrow (cf. the $\approx 12\times$ advantage), not a taxonomy defect.

Method	Macro-F1	Accuracy
Majority class	0.011	0.142
Stratified random	0.034 ± 0.008	0.054
Keyword rule	0.060	0.352
ALLaM-7B (zero-shot)	0.050	0.148
ALLaM-7B (5-shot)	0.098	0.228
Format A (BERT, blind)	0.114 ± 0.030	0.232
TF-IDF + LogReg	0.162 ± 0.007	0.192
TF-IDF + SVM	0.237 ± 0.013	0.275
AraBERT v2 + anchor (Format D)	$\mathbf{0.614 \pm 0.017}$	**0.593**

Table 9: Baseline comparison on the 438-instance Gold test set; all methods receive the same anchor-guided (Format D) input except Format A (the no-anchor control, main 5-fold run; the dedicated format-ablation run is Table 17). Classical methods trained on T1+T2; ALLaM-7B zero- and 5-shot; fine-tuned AraBERT v2 (bottom row) for reference.

Released checkpoint. The default deployable is fold-0 (single-fold macro-F1 = 0.635, matching the per-class results above); fold-3 (0.589) is released as the cross-domain checkpoint (full release in Appendix F).

6 Discussion

Framing and data quality outweigh model choice. Three results point the same way. First, anchor framing is the dominant lever: the keyword→best_match anchor lifts macro-F1 from a blind 0.114 to the released 0.614, dwarfing any gap between encoders. Second, provenance beats scale: 738 validated real instances are $\approx 12\times$ more sample-efficient than synthetic, so a small real set carries more signal than an order of magnitude more generated text. Third, encoder choice buys little: the three Arabic-specific encoders are statistically equivalent, and the only significant gaps are to the multilingual XLM-R and the domain-adjacent ConfliBERT. Together these say that, for fine-grained Arabic MT-error classification, how the input is framed and where the training data comes from matter far more than which encoder is used or how much synthetic data is added.

Implications for Arabic and low-resource MT evaluation. The practical lesson, for Arabic and for other morphologically rich, low-resourced languages, is to invest in anchor-style framing and targeted real annotation rather than in model size or synthetic volume (Zhou et al., 2023). Because the pipeline is language-agnostic outside the four

Arabic-specific classes (§3), this recipe should generalise beyond Arabic.

A diagnostic instrument, not a scalar score. The RAQEEB classifier decomposes errors across the MIZAN classes and flags the Frontier-tier semantic classes for human review rather than returning a single opaque number, and its multi-level structure lets low-confidence 23-class predictions back off to the more reliable 14-subtype level (§5). That decomposition is what scalar metrics cannot provide, and the reason this domain needs a fine-grained taxonomy rather than another scalar score.

7 Conclusion and Future Work

We presented QUANTA, MIZAN, the ETCA cross-vendor audit, and the released RAQEEB classifier (5-fold mean macro-F1 $= 0.614$; fold-0 default at 0.635). Together they close the resource gap for fine-grained Arabic MT-error analysis: one download provides the taxonomy, dataset, audit protocol, and a baseline classifier for downstream probing.

Concrete downstream uses include calibrating MQM-compatible segment-level scores against human ratings and glossary-lookup remediation for Terminology Substitution. Code, data, prompts, the checkpoint, and all per-class artefacts are released under CC-BY-4.0.

No public Arabic MT resource annotates errors at MIZAN's 23-class resolution outside the UNPC domain; a $\sim$1,000-instance cross-register benchmark (medical, legal, literary) is the highest-leverage next data investment, drawing on existing English–Arabic parallel corpora (including domain-specific resources) as source material.

Several technical directions follow: augmenting the anchor with sentence-level semantic context for the Frontier-tier classes, the hierarchical-fallback deployment motivated by the L3 peak, a larger fine-tuned encoder, and broadening the ETCA audit across additional auditor vendors as a complementary robustness check on top of its existing expert-human validation. The framework is also designed to port beyond Arabic: 19 of the 23 MIZAN classes are language-agnostic (rooted in MQM) and only four are Arabic-specific (Tanween Omission, Gender Disagreement, Definiteness Shift, Invalid Pattern), so re-instantiating just those four language-specific leaves adapts the taxonomy to a new language while the language-agnostic backbone and the anchor-guided construction and audit pipeline carry over unchanged. On this basis MIZAN can be extended to dialectal Arabic and code-switching, and transferred to other morphologically rich languages such as Turkish and Kazakh (Yerkebulan et al., 2026) and to the reverse Ar→En direction.

Limitations

Domain, direction, and Arabic register. All experiments analyse En→Ar machine translation of UN Security Council Modern Standard Arabic exclusively. Other UN source languages, the reverse Ar→En direction, and dialectal Arabic varieties (Egyptian, Levantine, Gulf, Maghrebi) are out of scope and would require re-calibration of the four Arabic-specific classes.

Gold-test size and per-class support. The 438-instance Gold test set supports reliable per-class evaluation only for the 10 classes with $n \geq 10$; the remaining 13 are reported but not interpreted individually. All 23 are nonetheless evaluated at the L2 (parent-group) and L3 (subtype) rollup levels, where aggregation gives adequate support.

Human-annotation resources. Both the initial MIZAN annotation of QUANTA and the dual-annotator IAA validation ($n = 147$) were conducted by a small pool of expert annotators under the human-annotation, time, and financial constraints typical of an academic resource-paper effort. A larger annotation round with additional independent annotators is planned.

Single-LLM synthetic generator. The T2 synthetic split is generated by one LLM family (GPT-4o); the cross-vendor ETCA judge mitigates judge-side self-preference but does not eliminate generator-side stylistic artefacts. Because GPT-4o may have encountered UNPC text in pretraining, T2 sentences could echo memorised content. However, the headline results use the held-out real Gold set, so they are not affected by contamination. A multi-generator check (Claude, Gemini) is planned.

Acknowledgments

This research was supported in part by NSF award OAC-2311142 and NIST grant number 60NANB24D143. This work used Delta at NCSA / University of Illinois through allocation CIS220162 from the ACCESS program. We thank the expert Arabic annotators who contributed to the dual-annotator validation.

References

Hussein S. Al-Olimat and Ahmad Alshareef. 2026. ALPS: A diagnostic challenge set for Arabic linguistic & pragmatic reasoning. *arXiv preprint arXiv:2602.17054v1*.

Almoataz B Al-Said. 2026. Mizan: An innovative tool for building arabic morphological pattern dictionaries and enhancing lexical resource development. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 25(2):1–26.

Khetam Al Sharou and Lucia Specia. 2022. A taxonomy and study of critical errors in machine translation. In *Proceedings of the 23rd Annual Conference of the European Association for Machine Translation (EAMT)*, Ghent, Belgium.

Abdullah Yahya G Alfaifi. 2015. *Building the Arabic learner corpus and a system for Arabic error annotation*. Ph.D. thesis, University of Leeds.

Chantal Amrhein, Nikita Moghe, and Liane Guillou. 2022. ACES: Translation accuracy challenge sets for evaluating machine translation metrics. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 479–513. Association for Computational Linguistics.

Anthropic. 2025. Introducing Claude 4. `https://www.anthropic.com/news/claude-4`.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. AraBERT: Transformer-based model for Arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT)*, pages 9–15.

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, and 6 others. 2024. ALLaM: Large language models for Arabic and English. *arXiv preprint arXiv:2407.15390*.

Riadh Belkebir and Nizar Habash. 2021. Automatic error type annotation for Arabic. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 596–606, Online. Association for Computational Linguistics.

Jingzhou Chen, Junjie Huang, Peng Wang, Fengchao Xiong, Yuntao Qian, and Liang Xiao. 2025. Adaptive label granularity selection for remote sensing targets: Balancing accuracy and specificity. *IEEE Transactions on Geoscience and Remote Sensing*, 63.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and Hongxia Jin. 2023. Alpagasus: Training a better alpaca with fewer data. *arXiv preprint arXiv:2307.08701*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 8440–8451.

Daniel Deutsch, Eleftheria Briakou, Isaac Caswell, Mara Finkelstein, Rebecca Galor, Juraj Juraska, Geza Kovacs, Alison Lui, Ricardo Rei, Jason Riesa, Shruti Rijhwani, Parker Riley, Elizabeth Salesky, Firas Trabelsi, Stephanie Winkler, Biao Zhang, and Markus Freitag. 2025. WMT24++: Expanding the language coverage of WMT24 to 55 languages & dialects. In *Findings of the Association for Computational Linguistics: ACL 2025*.

Ahmed El Kholy and Nizar Habash. 2011. Automatic error analysis for morphologically rich languages. In *Proceedings of Machine Translation Summit XIII: Papers*.

Khaled Elhady, Omar Kallas, Nizar Habash, and Bashar Alhafni. 2026. Arabigee: A hierarchical taxonomy for arabic grammatical error explanation. *arXiv preprint arXiv:2606.10765*.

Alvan R. Feinstein and Domenic V. Cicchetti. 1990. High agreement but low kappa: I. the problems of two paradoxes. *Journal of Clinical Epidemiology*, 43(6):543–549.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, errors, and context: A large-scale study of human evaluation for machine translation. In *Transactions of the Association for Computational Linguistics*, volume 9, pages 1460–1474.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 3816–3830.

Kilem L. Gwet. 2008. Computing inter-rater reliability and its variance in the presence of high agreement. *British Journal of Mathematical and Statistical Psychology*, 61(1):29–48.

Nizar Habash. 2010. *Introduction to Arabic Natural Language Processing*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Yibo Hu, MohammadSaleh Hosseini, Erick Skorupa Parolin, Javier Osorio, Latifur Khan, Patrick T. Brandt, and Vito D'Orazio. 2022. ConfliBERT: A pre-trained language model for political conflict and violence. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 5469–5482.

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.

Marzena Karpinska, Nishant Raj, Katherine Thai, Yixiao Song, Ankita Gupta, and Mohit Iyyer. 2022. DEMETR: Diagnosing evaluation metrics for translation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9540–9561. Association for Computational Linguistics.

Adam Kilgarriff. 2009. Simple maths for keywords. In *Proceedings of the Corpus Linguistics Conference*, volume 6, pages 41–55. University of Liverpool Liverpool.

Ahrii Kim. 2025. RUBRIC-MQM: Span-level LLM-as-judge in machine translation for high-end models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pages 147–165.

Tom Kocmi and Christian Federmann. 2023. GEMBA-MQM: Detecting translation quality error spans with GPT-4. In *Proceedings of the Eighth Conference on Machine Translation (WMT)*, pages 768–775.

Tom Kocmi, Vilém Zouhar, Eleftherios Avramidis, Roman Grundkiewicz, Marzena Karpinska, Maja Popović, Mrinmaya Sachan, and Mariya Shmatova. 2024. Error span annotation: A balanced approach for human evaluation of machine translation. In *Proceedings of the Ninth Conference on Machine Translation*, pages 1440–1453.

J. Richard Landis and Gary G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174.

Hannah Liu, Junghyun Min, En-Shiun Annie Lee, Ethan Yue Heng Cheung, Shou-Yi Hung, Elsie Chan, Shiyao Qian, Runtong Liang, Kimlan Huynh, Wing Yu Yip, York Hay Ng, Tsz Fung Yau, Ka Ieng Charlotte Lo, You-Wei Wu, and Richard Tzong-Han Tsai. 2025. SiniticMTError: A machine translation dataset with error annotations for sinitic languages. *Preprint*, arXiv:2509.20557.

Arle Lommel, Serge Gladkoff, Alan Melby, Sue Ellen Wright, Ingemar Strandvik, Katerina Gasova, Angelika Vaasa, Andy Benzo, Romina Marazzato Sparano, Monica Foresi, Johani Innis, Lifeng Han, and Goran Nenadic. 2024. The multi-range theory of translation quality measurement: MQM scoring models and statistical quality control. *arXiv preprint arXiv:2405.16969*.

Arle Lommel, Hans Uszkoreit, and Aljoscha Burchardt. 2014. Multidimensional quality metrics (MQM): A framework for declaring and describing translation quality metrics. In *Tradumatica: Tecnologies de la Traducció*, 12, pages 455–463.

Dheeraj Mekala, Varun Gangal, and Jingbo Shang. 2021. Coarse2Fine: Fine-grained text classification on coarsely-grained annotated data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 583–594. Association for Computational Linguistics.

Behrang Mohit, Alla Rozovskaya, Nizar Habash, Wajdi Zaghouani, and Ossama Obeid. 2014. The first QALB shared task on automatic text correction for Arabic. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 39–47.

Ananya Mukherjee and Manish Shrivastava. 2025. Lost in translation? found in evaluation: A comprehensive survey on sentence-level translation evaluation. *ACM Computing Surveys*, 58(1):1–47.

Dasa Munkova, Lucia Benkova, Michal Munk, L'ubomír Benko, and Petr Hajek. 2025. From scores to insights: Predicting mt errors using reliable metrics and linguistic typology in slavic languages. *MethodsX*, page 103613.

El Moatez Billah Nagoudi, AbdelRahim Elmadany, and Muhammad Abdul-Mageed. 2022. TURJUMAN: A public toolkit for neural Arabic machine translation. In *Proceedings of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT5) @ LREC2022*, pages 1–11, Marseille, France. European Language Resources Association.

Curtis G. Northcutt, Anish Athalye, and Jonas Mueller. 2021. Pervasive label errors in test sets destabilize machine learning benchmarks. In *Proceedings of the Thirty-Fifth Conference on Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*.

OpenAI. 2024. GPT-4o system card. `https://openai.com/index/gpt-4o-system-card/`.

Javier Osorio, Sultan Alsarra, Amber Converse, Afraa Alshammari, Dagmar Heintze, Latifur Khan, Naif Alatrush, Patrick T. Brandt, Vito D'Orazio, Niamat Zawad, and Mahrusa Billah. 2024. Keep it local: Comparing domain-specific LLMs in native and machine translated text using parallel corpora on political conflict. In *Proceedings of the 2nd International Conference on Foundation and Large Language Models (FLLM)*, pages 542–551.

Javier Osorio, Afraa Alshammari, Naif Alatrush, Dagmar Heintze, Amber Converse, Sultan Alsarra, Latifur Khan, Patrick T. Brandt, and Vito D'Orazio. 2025. The devil is in the details: Assessing the effects of machine-translation on LLM performance in domain-specific texts. In *Proceedings of Machine Translation Summit XX: Volume 1*, pages 315–332.

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 (ACL)*, pages 311–318.

Davide Pirovano, Federico Milanesio, Michele Caselle, Piero Fariselli, and Matteo Osella. 2025. The advantage of fine-grained training. *arXiv preprint arXiv:2509.05130*.

Maja Popović. 2011. Hjerson: An open source tool for automatic error classification of machine translation output. *The Prague Bulletin of Mathematical Linguistics*, 96:59–67.

Mengyang Qiu, Tran Minh Nguyen, Zihao Huang, Zelong Li, Yang Gu, Qingyu Gao, Siliang Liu, and Jungyeul Park. 2025. Multilingual grammatical error annotation: Combining language-agnostic framework with language-specific flexibility. *arXiv preprint arXiv:2506.07719*.

Ricardo Rei, José G. C. de Souza, Duarte Alves, Chrysoula Zerva, Ana C. Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André F. T. Martins. 2022. COMET-22: Unbabel-IST 2022 submission for the metrics shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 578–585.

Ricardo Rei, Craig Stewart, Ana C. Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702.

Ananya Sai B, Tanay Dixit, Vignesh Nagarajan, Anoop Kunchukuttan, Pratyush Kumar,

Mitesh M. Khapra, and Raj Dabre. 2023. IndicMT eval: A dataset to meta-evaluate machine translation metrics for Indian languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14210–14228, Toronto, Canada. Association for Computational Linguistics.

Hassan Sajjad, Ahmed Abdelali, Nadir Durrani, and Fahim Dalvi. 2020. AraBench: Benchmarking dialectal Arabic-English machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING)*, pages 5094–5107.

Timo Schick and Hinrich Schütze. 2021. Generating datasets with pretrained language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6943–6951.

Varsha Suresh and Desmond C. Ong. 2021. Not all negatives are equal: Label-Aware contrastive loss for fine-grained text classification. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4381–4394. Association for Computational Linguistics.

Taofik O. Tafa, Siti Zaiton Mohd Hashim, Mohd Shahizan Othman, Hitham Alhussian, Maged Nasser, Said Jadid Abdulkadir, Sharin Hazlin Huspi, Sarafa O. Adeyemo, and Yunusa Adamu Bena. 2025. Machine translation performance for low-resource languages: A systematic literature review. *IEEE Access*, 13:72486–72505.

David Vilar, Jia Xu, Luis Fernando D'Haro, and Hermann Ney. 2006. Error analysis of statistical machine translation output. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC)*, pages 697–702.

Sidi Wang, Sophie Arnoult, and Amir Kamran. 2026. Large language models as annotators for machine translation quality estimation. *arXiv preprint arXiv:2603.10775*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Gulnur Yerkebulan, Akerke Akanova, Zhantore Galymzhan, and Nazira Ospanova. 2026. A cefr-graded lexicon and morphology-aware benchmarks for kazakh lexical complexity prediction. *Technologies*, 14(6):346.

Wajdi Zaghouani, Behrang Mohit, Nizar Habash, Ossama Obeid, Nadi Tomeh, Alla Rozovskaya, Noura Farra, Sarah Alkuhlani, and Kemal Oflazer. 2014. Large scale Arabic error annotation: Guidelines and framework. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC)*, pages 2076–2083.

Daochen Zha, Zaid Pervaiz Bhat, Kwei-Herng Lai, Fan Yang, Zhimeng Jiang, Shaochen Zhong, and Xia Hu. 2023. Data-centric artificial intelligence: A survey. *arXiv preprint arXiv:2303.10158*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with BERT. In *Proceedings of the Eighth International Conference on Learning Representations (ICLR)*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. LIMA: Less is more for alignment. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Michał Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The United Nations Parallel Corpus v1.0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC)*, pages 3530–3534.

Appendices

Appendix roadmap. The appendices are organised in eleven sections: (A) Mizan taxonomy specification with the full per-class table, severity

weights, and the 4-level rollup; (B) dual-annotator validation protocol with full inter-annotator agreement detail; (C) construction and audit algorithms; (D) ETCA generation prompt and audit rubric verbatim; (E) full per-class classifier results; (F) reproducibility (split indices, hyperparameters, environment, code repository pointer); (G) encoder significance and equivalence (per-pair paired-bootstrap and TOST results); (H) anchor framing and data-efficiency ablations (Format A vs. Format D, classical and LLM baselines, with 23-class confidence intervals); (I) per-class T2 generation audit; (J) cross-language dataset comparison; (K) ethics and broader impact.

A MIZAN Taxonomy Specification

Table 10 lists all 23 MIZAN classes with definitions and representative examples. Table 12 gives the per-class severity weights and the full four-level rollup (subtype, parent group, TQA category). Figure 3 renders the taxonomy as a tree: all 23 leaf classes under their five parent groups, each leaf shaded by its default severity.

Weight invariants. Each class's severity level (Minor to Critical) and 1–5 score follow MQM's severity typology, assigned by the two expert taxonomy designers and mapped to the class's primary MQM dimension. The per-class weights in Table 12 are calibrated so that per-class partial sums recover both the 5-parent-group totals (Semantic Divergence 0.40, Morphosyntactic Shifts 0.10, Syntactic Shifts 0.10, Pragmatic Divergence 0.35, Orthographic Shifts 0.05; summing to 1.00) and the 4-TQA-category totals (Accuracy 0.42, Fluency 0.225, Terminology 0.28, Style & Register 0.075; summing to 1.00). The parent-group and TQA-dimension levels are therefore derived views, not separate multiplicative factors. These weights are MQM-compatible: they adopt MQM's severity typology and map onto its TQA dimensions, but computing a segment-level score from them is out of scope here.

Class-level severity distribution. Across the 23 MIZAN classes the severity levels are distributed as 2 Critical, 5 Severe, 7 Major, 8 Moderate, and 1 Minor (Table 12): roughly two-thirds of the taxonomy sits at Major or below, while the two Critical classes (Meaning Shift, Total Omission) carry the highest per-class weight. Table 11 gives the full per-class instance counts across the T1 (real), T2 (synthetic), and Gold tiers; the released artefacts include additional Gold examples beyond the one-per-class set in Table 10.

Worked example (weight asymmetry). The per-class weight×severity product spans two orders of magnitude: Meaning Shift ($s = 5$, $w_{\text{class}} = 0.12$) gives $5 \times 0.12 = 0.60$, while Active to Passive Voice ($s = 2$, $w_{\text{class}} = 0.0025$) gives $2 \times 0.0025 = 0.005$. This $120\times$ asymmetry encodes the domain-expert judgment that semantic errors damage MT quality far more than register-level voice shifts in formal UN text.

B Dual-Annotator Validation Protocol

B.1 Dual-Annotator Evaluation Design

We designed a keyword-guided dual-annotator evaluation protocol for the QUANTA error instances. Annotators see the claimed MIZAN class, keyword, and best-match span, and answer seven criteria:

1. Error presence;
2. Anchor accuracy;
3. Keyword-level severity;
4. Minimal-pair structure;
5. Correct MIZAN class label;
6. Naturalness;
7. Domain appropriateness (UN register).

Because the annotator is shown the claimed class and anchor, the protocol is verification-style rather than open annotation; both annotators judge every instance independently.

B.2 Stratified Sampling

The evaluation sample comprises 147 instances, drawn by stratified random sampling over MIZAN class and frozen before any model training: 55 T1 instances (real MT errors stratified across the available real-MT error classes) and 92 T2 instances (QUANTA-synthetic, 4 instances per class across all 23 classes). We stratify by class so that agreement is estimated across the full taxonomy rather than being dominated by the most frequent classes; the choice of Gwet's AC1 over Cohen's κ is motivated separately by the high prevalence of the binary criteria (§B.3).

B.3 Inter-Annotator Agreement

Table 13 reports per-criterion Gwet's AC1 values on the 55 T1 and 92 T2 instances. AC1 is the primary statistic because the binary evaluation criteria have high prevalence by design, a

Class	s_i	Definition	Sentence_ID (tool)	Keyword → Best_Match	AR reference excerpt	MT output excerpt
Semantic Divergence (7 classes, parent weight 0.40)						
Meaning Shift	5	Referential content or core semantic value changed	2005/s/ac_37/2005/_1455_/9/14 (Transformers)	النيجر منطقة → النيجر بلد	النيجر بلد ذو مساحة شاسعة تبلغ...	إن النيجر منطقة شاسعة تبلغ مساحتها...
Total Omission	5	Source element completely missing from MT output	2014/s/2014/430/50:1 (Google)	بعثة → [OMITTED]	... يقرر أن تشمل ولاية بعثة الأمم المتحدة في كو...	19 - يقرر أن تكون ولاية عملية الأمم المتحدة في كوت ديفوار كم
Partial Translation	4	Named entity or key term only partially rendered	2014/s/2014/740/38:1 (Transformers)	المكتب → مكتب الأمم المتحدة	29 - ووضع مكتب الأمم المتحدة المعني بالمخدرات وا...	29. قام المكتب، بالاشتراك مع جامعة...
Name Entity Error	4	Named entity incorrectly translated or transliterated	2006/s/2006/154/290:2 (Google)	صوت العالم → صوت الجهاد	...ة بوجه عام مثل موقع صوت الجهاد، أو تلك التي تقدم ا...	... بشكل عام، مثل موقع صوت العالم. الجهاد، أو تقديم ا...
Literal Translation	3	Word-for-word translation ignoring Arabic idiom	2006/s/2006/822/209:8 (Deep)	ساخن → محتدم	ودار جدال شفوي محتدم بين العقيد ريبس والس...	...د مالك تبادلا شفهيا ساخنا.
Hypernym for Hyponym	2	Broader term used where source specifies narrower	2004/s/2004/226/118:2 (Deep)	منفصل → مستقل	...وبنه وعنوانه في ملف مستقل عن الملفات الأخرى ل...	...لشخص وعنوانه في ملف منفصل.
Hyponym for Hypernym	2	Narrower term used where source specifies broader	2014/s/2014/740/81:1 (Google)	البحارة → الملاحين	...غير المشروط عن جميع الملاحين الأبرياء المحتجزين ...	...غير المشروط عن جميع البحارة الأبرياء المحتجزين ...
Morphosyntactic Shifts (4 classes, parent weight 0.10)						
Invalid Pattern	3	Morphologically impossible Arabic root-pattern form	2009/s/res/1904_2009_/23:1 (Google)	الأعتدة → معدات	...وما يتصل بها من معدات بجميع أنواعها، بما...	...للأسلحة والأعتدة ذات الصلة بجميع أنواعها...
Tanween Omission	2	Nunation omitted where grammatically required	2008/s/2008/324/97:2 (DeepL)	أولا → أولاً	أولاً، يرتبط التنفيذ الفعال لنظام الجزاءات...	أولا، حفز إنشاء المحكمتين المخصصتين...
Gender Disagreement	3	Masculine/feminine agreement violated	2010/s/prst/2010/25/819:1 (Transformers)	إدانتها → إدانته	...لأنه، رغم إدانته المتكررة للعنف الموجه ضد المرأة...	...على الرغم من إدانتها المتكررة للعنف ضد النساء...
Definiteness Shift	2	Incorrect ال addition or deletion	2005/s/2005/633/25:2 (Transformers)	السلام الإقليمي الدائم → سلام إقليمي دائم	...تحقيق سلام إقليمي دائم يتعين التعامل مع جميع الق...	...يتحقق السلام الإقليمي الدائم، لا بد من معالجة جميع الق...
Syntactic Shifts (9 classes, parent weight 0.10)						
Perfective to Progressive	4	Completed action rendered as ongoing	2009/s/2009/193/92:3 (Deep)	بحث → حث	...لذا فقد حث المجلس على التحول من اله...	...لك فإنه يحث المجلس على التحول من نهج...
Progressive to Perfective	4	Ongoing action rendered as completed	2005/s/2005/690/14:3 (DeepL)	كانت إريتريا تنظم → ظلت تنظم	فإريتريا ظلت تنظم وتدرب فتات شتى من ع...	كانت إريتريا تنظم وتدرب ثم ترسل إلى إ...
Tense Shift Under Negation	3	Tense changes incorrectly under negation	2011/s/2013/467/84:2 (Deep)	لن يكون → لا يكون	ولا يكون نظام الجزاءات فعالا...	إن نظام العقوبات لن يكون فعالاً إلا إذا تم ف...
Wrong Structure	3	Sentence structure violates Arabic syntactic patterns	2012/s/res/2083__2012_/178:1 (Google)	يرسل من اللجنة جميع → إحالة جميع المعلومات المعلومات	(ج) إحالة جميع المعلومات التي زودت بها اللجن...	(ج) يرسل من اللجنة جميع المعلومات المتعلقة بالقرار ال...
Wrong Word Order	3	Constituents in incorrect syntactic positions	2014/s/2014/943/35:3 (Deep)	القانون الإنساني → القانون الدولي الإنساني الدولي	...ماعات المسلحة بموجب القانون الدولي الإنساني العرفي.	...ماعات المسلحة بموجب القانون الإنساني الدولي العرفي.
Noun to Adjective	2	Source noun rendered as adjective	2011/s/2011/463/305:7 (Deep)	تحقيق → التحقيق	... سلطات خاصة في مجال التحقيق للسلطات المعنية بإن...	...لقانون فيها صلاحيات تحقيقية خاصة.
Adjective to Noun	2	Source adjective rendered as noun	2011/s/2011/245/51:2 (Google)	استمرار → المستمرة	...ي التحديات القضائية المستمرة التي تواجهها الدول ...	...تان، والثاني يتعلق باستمرار الطعون القضائية في ...
Active to Passive Voice	2	Active construction rendered in passive	2005/s/2005/60/563:4 (Transformers)	شن الهجوم → شنت الهجوم	...ات العسكرية هي التي شنت الهجوم أساسا، بينما قام ال...	شن الهجوم بشكل رئيسي من قبل ا...
Passive to Active Voice	2	Passive construction rendered in active	2002/s/2002/169/93:5 (Google)	تم نقل → نُقلت	...الموجودة في غوما، قد نُقلت إلى كيغالي وجيسيني.	وتم نقل طائرات البعثة الأرب...
Pragmatic Divergence (2 classes, parent weight 0.35)						
Register Mismatch	3	Formality level differs between source and target	2006/s/2006/822/209:8 (Deep)	تبادل كلامي → جدال شفوي	ودار جدال شفوي محتدم بين العقيد ري...	ودار تبادل كلامي ساخن بين العقيد ريبس...
Terminology Substitution	4	Domain-specific term replaced with incorrect equivalent	2012/s/prst/2012/22/10:3 (Deep)	يدعو → يهيب	ويهيب بالأمين العام للأمم المتح...	ويدعو الأمين العام للأمم المتحد...
Orthographic Shifts (1 class, parent weight 0.05)						
Spelling Error	1	Orthographic error: typos, missing/extra letters	2005/s/2005/60/540:11 (Transformers)	اربعة → أربعة	... قد قتلا. وقتل أيضا أربعة أشخاص آخرين.	...اصغر قتلا ، كما قتل اربعة اشخاص آخرين.

Table 10: The full MIZAN 23-class taxonomy with one real En→Ar UN Gold-test example per class ([OMITTED] marks a Total Omission). The engine in parentheses after each Sentence_ID (Google, DeepL, Deep, or Transformers) is the MT system that produced that example.

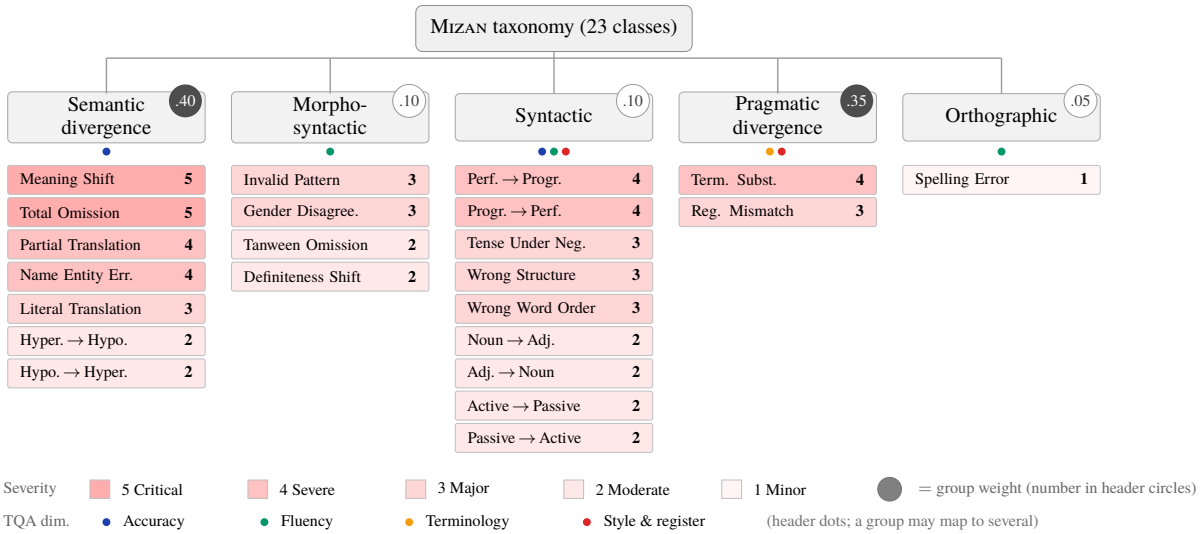


Figure 3: Mizan taxonomy: the 23 leaf classes grouped under their five parent groups; leaf shade and number encode severity, header dots the TQA dimension(s), the badge the parent-group weight. The full four-level rollup, including the 14 subtypes, is in Table 12.

regime in which Cohen's κ collapses artifactually (Feinstein and Cicchetti, 1990; Gwet, 2008). On both tiers, error correctness and anchor correctness reach almost-perfect agreement; label correctness reaches 0.930 on T2 and 0.680 on T1. T2 keyword-level severity reaches AC1 = 0.987 (supporting the severity-weight calibration); T2 UN-register fit reaches 0.608 (moderate, just below the substantial threshold); T1 keyword-level severity reaches 0.434 (moderate), reflecting expected divergence on fine-grained ordinal severity judgments against real-text contexts (Lommel et al., 2024). The mean AC1 over all reported criteria is 0.898 on T2 and 0.729 on T1; restricted to the three core taxonomy criteria (error, anchor, label), the mean rises to 0.966 and 0.827 respectively. All three core criteria meet or exceed the Landis–Koch substantial-agreement threshold on both tiers. Two of the seven workbook criteria, minimal-pair structure and naturalness, are recorded in the annotation protocol but omitted from Table 13 because both elicited systematic interpretation divergence between the two annotators (on minimal-pair, the annotator-mean positive rates were 0.90 versus 0.12); their low agreement reflects protocol ambiguity on those subjective judgments rather than unreliable taxonomy labels.

C Construction and Audit Algorithms

Algorithm 1 states the anchor-constrained construction that produces the T1 (real) and T2 (synthetic) tiers, and Algorithm 2 states the cross-vendor Etca audit. Both are described in prose in §4.1 and §4.2; the runnable implementation ships with the released repository.

D Etca Generation Prompt and Audit Rubric

This appendix reproduces verbatim the two prompts routed through the cross-vendor Etca audit (§4.2): the GPT-4o generation prompt and the Claude Sonnet 4 audit rubric.

Generation prompt (GPT-4o, 3-shot). The generator generates a UN-domain Arabic reference and its erroneous MT output for a target Mizan class. It is prompted with a system message (Figure 4), a three-shot exemplar block of (reference, class, MT output) triples drawn from the T0 canonical seed pool and stratified to cover the target Mizan family, and a per-sample instruction that re-states the anchor constraint and the requested output format (JSON only). Decoding parameters: temperature = 0.7, top-p = 0.95, max_tokens = 256; each candidate is generated independently with no rejection sampling at the generator stage.

Audit rubric (Claude Sonnet 4, five-dimension scoring). The auditor receives a (reference, MT, target class) triple and returns five integer scores in $\{1, \ldots, 5\}$ on the following dimensions:

1. **Anchor Validity**: does the keyword→best_match transformation realise the target class?
2. **Phenomenon Clarity**: is the error unambiguous, with no competing class label?

MIZAN class	T1	T2	Gold	Total
Semantic Divergence				
Meaning Shift	–	369	70	439
Total Omission	542	384	62	988
Partial Translation	–	166	7	173
Name Entity Error	–	369	2	371
Literal Translation	–	369	15	384
Hypernym for Hyponym	21	369	30	420
Hyponym for Hypernym	6	369	9	384
Morphosyntactic Shifts				
Invalid Pattern	3	263	10	276
Tanween Omission	15	369	14	398
Gender Disagreement	–	366	4	370
Definiteness Shift	11	320	12	343
Syntactic Shifts				
Perfective to Progressive	–	367	5	372
Progressive to Perfective	9	369	6	384
Tense Shift Under Negation	–	369	3	372
Wrong Structure	–	369	8	377
Wrong Word Order	2	369	13	384
Noun to Adjective	–	357	3	360
Adjective to Noun	2	359	3	364
Active to Passive Voice	–	368	4	372
Passive to Active Voice	2	369	8	379
Pragmatic Divergence				
Register Mismatch	64	365	96	525
Terminology Substitution	61	369	52	482
Orthographic Shifts				
Spelling Error	–	113	2	115
Total	**738**	**7,856**	**438**	**9,032**

Table 11: Per-class instance counts across the T1 (real), T2 (synthetic), and Gold tiers of QUANTA. "–" marks the 11 classes with no real T1 support (8 not surfaced by pattern-based retrieval plus 3 removed by Gold-overlap deduplication); these are covered in training only through T2 synthesis. Counts computed from the released dataset splits.

3. **Arabic Naturalness**: is the surface form fluent Arabic, modulo the injected error?
4. **Collateral Severity**: is the rest of the sentence held constant (lower is better)?
5. **Seed Recommendation**: would this candidate, if used as a T0 seed, mislead future generation?

The **Data Quality Index (DQI)** aggregates the four substantive dimensions (excluding Seed Recommendation, which is used only for downstream filtering) as a normalised mean:

$$\text{DQI} = \frac{1}{|D|} \sum_{d \in D} \frac{s_d - 1}{4} \in [0, 1], \qquad (1)$$

where D is the set of the four substantive dimensions, d indexes a dimension, and $s_d \in \{1, \ldots, 5\}$ is its integer score; each term $(s_d - 1)/4$ rescales a 1–5 score to $[0, 1]$, and DQI is their mean. Collateral Severity, where lower is better, enters on its raw 1–5 scale; audited instances sit near its floor (mean $\approx$ 1.0), so its effect on DQI is minor. DQI is computed on the audited subset (the 794 high-confidence seeds and a stratified 414-instance T2 audit subset, $\approx$ 5% of T2 covering all 23 classes) as a quality measure; acceptance into the released 7,856-instance T2 is determined by the structural re-classification audit (§4.4; of 8,044 generated candidates, 188 are excluded as Partial-Translation/Total-Omission confounds, yielding 7,856), not by the DQI threshold. The verbatim rubric is in Figure 5.

The audit prompt also requests a brief free-text justification per dimension; these justifications ship with the release data and are used in error analysis but not in the DQI score.

ETCA audit scores. Table 14 reports the per-dimension means. On the 414-instance T2 audit subset the substantive dimensions are high and Collateral Severity is near its floor; on the 794 generation seeds the scores are lower (the seeds are real-MT instances), and 664 of the 794 seeds are

Algorithm 1 Anchor-constrained construction

Require: aligned sentence pairs (r, m): Arabic reference sentence r, MT output sentence m; MIZAN classes $\mathcal{C}$; seed pool T_0. An anchor (k, b, c) instantiates class c's keyword→best_match transformation (k the reference-side form, b the MT-side form; token-level spans; b = `[OMITTED]` marks a Total Omission; per-class transformations in App. A). Mined anchors apply the frozen keyword→best_match patterns to the pairs (§4.1). $\text{OCCURS}(x, y)$ holds iff span x matches a span of y under a diacritic-/clitic-normalized comparison. Anchor gate $\text{REALIZED}(k, b, \textit{ref}, \textit{mt}) \equiv \text{OCCURS}(k, \textit{ref}) \wedge (\text{OCCURS}(b, \textit{mt}) \vee (b{=}\texttt{[OMITTED]} \wedge \neg\, \text{OCCURS}(k, \textit{mt})))$.

Ensure: validated instance set D (unique instances)

(ETCA audit, Alg. 2, runs in parallel; it does not gate admission.)

$D \leftarrow \emptyset$
/ T1: mine real MT errors (anchor gate only) */*
for each aligned pair (r, m) **do**
  for each mined anchor (k, b, c) **do**
    if $\text{REALIZED}(k, b, r, m)$ **then** ▷ anchor structurally realized
      $D \leftarrow D \cup \{(r, m, k, b, c, \text{T1})\}$
    end if
  end for
end for
/ T2: synthesize to full 23-class coverage; STRUCTURALAUDIT: structural-noise filter, §4.1 */*
for each class $c \in \mathcal{C}$ **do**
  for each of N_c generated candidates (rotating across UN sub-domains) **do**
    $(r, m', k, b) \leftarrow \text{GENERATE}(c, T_0)$ ▷ GPT-4o, 3-shot from T_0: generates r, m', anchor (k, b)
    if $\text{REALIZED}(k, b, r, m')$ **and** STRUCTURALAUDIT(r,m',k,b,c) **then**
      $D \leftarrow D \cup \{(r, m', k, b, c, \text{T2})\}$
    end if
  end for
end for
return D

MIZAN class	Severity	s_i	w_{class}	Subtype (L3)	Parent group	TQA category
Meaning Shift	Critical	5	0.1200	Semantic Shift	Semantic Divergence	Accuracy
Total Omission	Critical	5	0.1200	Translation Gap	Semantic Divergence	Accuracy
Partial Translation	Severe	4	0.0400	Translation Gap	Semantic Divergence	Accuracy
Name Entity Error	Severe	4	0.0400	Translation Gap	Semantic Divergence	Accuracy
Literal Translation	Major	3	0.0400	Idiomatic Shift	Semantic Divergence	Accuracy
Hypernym for Hyponym	Moderate	2	0.0200	Granularity Shift	Semantic Divergence	Accuracy
Hyponym for Hypernym	Moderate	2	0.0200	Granularity Shift	Semantic Divergence	Accuracy
Invalid Pattern	Major	3	0.0300	Morphological Error	Morphosyntactic Shifts	Fluency
Tanween Omission	Moderate	2	0.0200	Morphological Error	Morphosyntactic Shifts	Fluency
Gender Disagreement	Major	3	0.0300	Agreement Error	Morphosyntactic Shifts	Fluency
Definiteness Shift	Moderate	2	0.0200	Definiteness Shift	Morphosyntactic Shifts	Fluency
Perfective to Progressive	Severe	4	0.0150	Aspect Shift	Syntactic Shifts	Fluency
Progressive to Perfective	Severe	4	0.0150	Aspect Shift	Syntactic Shifts	Fluency
Tense Shift Under Negation	Major	3	0.0200	Tense Shift	Syntactic Shifts	Accuracy
Wrong Structure	Major	3	0.0200	Structural Error	Syntactic Shifts	Fluency
Wrong Word Order	Major	3	0.0200	Structural Error	Syntactic Shifts	Fluency
Noun to Adjective	Moderate	2	0.0025	Syntactic Category Shift	Syntactic Shifts	Fluency
Adjective to Noun	Moderate	2	0.0025	Syntactic Category Shift	Syntactic Shifts	Fluency
Active to Passive Voice	Moderate	2	0.0025	Voice Shift	Syntactic Shifts	Style & Register
Passive to Active Voice	Moderate	2	0.0025	Voice Shift	Syntactic Shifts	Style & Register
Register Mismatch	Major	3	0.0700	Register Mismatch	Pragmatic Divergence	Style & Register
Terminology Substitution	Severe	4	0.2800	Register Mismatch	Pragmatic Divergence	Terminology
Spelling Error	Minor	1	0.0500	Orthographic Issues	Orthographic Shifts	Fluency

Table 12: MIZAN taxonomy rollup and severity weights.

Criterion	Validates	T1	T2	L–K min
Error correct.	Error existence	**0.900**	**0.989**	≥ 0.81
Anchor correct.	Anchor identification	**0.900**	**0.978**	≥ 0.81
Label correct.	MIZAN class label	0.680	**0.930**	≥ 0.61
Severity	Severity calib.	0.434	**0.987**	≥ 0.61
UN register	T2 register fit	—	0.608	≥ 0.61
Core-three mean (err., anchor, label)		0.827	0.966	—
All-reported mean		0.729	0.898	—

Table 13: Inter-annotator agreement (Gwet's AC1) on 55 T1 and 92 T2 instances. L–K min is the Landis–Koch agreement band each criterion is expected to meet (almost-perfect ≥ 0.81 for error/anchor, substantial ≥ 0.61 otherwise); UN-register fit is T2 only.

Subset	Anchor	Clarity	Natural.	Collateral
T2 audit (414)	4.70	4.74	4.62	1.02
Seeds (794)	4.17	4.45	4.40	2.15

Table 14: ETCA per-dimension means (1–5) on the 414-instance T2 audit subset and the 794 generation seeds; for Collateral Severity, lower is better.

recommended. Per-instance ETCA scores ship with the release data.

External validation of ETCA. On a 43-instance high-confidence subset adjudicated against the dual-annotator consensus described in Appendix B, ETCA reaches Gwet's AC1 = 0.62 (substantial agreement on the Landis–Koch scale); the present paper uses ETCA strictly as a parallel dataset-quality audit.

Algorithm 2 ETCA cross-vendor audit

Require: candidate set X (each generated by GPT-4o); substantive dimensions $\mathcal{D}$ = {Anchor Validity, Phenomenon Clarity, Arabic Naturalness, Collateral Severity}, each scored on an integer 1–5 scale by an independent vendor (Claude Sonnet 4); Seed Recommendation is a separate flag, scored independently of $\mathcal{D}$ (scale defined in the audit rubric, App. D), not included in DQI

Ensure: Data Quality Index $\text{DQI}(x) \in [0, 1]$ for each $x \in X$

for each candidate $x \in X$ **do**
  $s \leftarrow \textsc{Score}(x,\ \mathcal{D} \cup \{\text{Seed Recommendation}\})$ ▷ Claude Sonnet 4, a different vendor
  $\text{DQI}(x) \leftarrow \frac{1}{|\mathcal{D}|} \sum_{d \in \mathcal{D}} \frac{s_d - 1}{4}$ ▷ normalized mean of the 4 substantive dims
  mark x high-confidence **if** $\text{DQI}(x) \geq 0.75$
end for
return $\{\text{DQI}(x)\}_{x \in X}$

E Per-Class Classifier Results

E.1 Multi-Level Analysis

Aggregating the per-class F1 values by TQA dimension (the L4→L1 macro-F1 progression is in §5): Fluency is strongest ($\approx$ 0.74–0.84, dominated by the Arabic morphological classes Tanween Omission, Definiteness Shift, Invalid Pattern) and Terminology weakest ($\approx$ 0.38–0.51, single-class group). Anchor-guided framing peaks on morphosyntactic phenomena and hits its representational ceiling on semantic distinctions, consistent with the Frontier-tier confusion pattern reported in body §5.

E.2 Per-Class F1 ($n \geq 10$ Gold Support)

Three readings follow. First, **surface-cue classes saturate near ceiling** across all five encoders: Total Omission (0.855–0.967), Definiteness Shift (0.774–0.923), Wrong Word Order (0.880–0.960), Tanween Omission (0.743–0.963) show that anchor-guided framing is sufficient to learn Arabic-morphology phenomena that scalar MT

Class	n	AraBERT v2	CAMeL-MSA	CAMeL-Mix	ConfliBERT	XLM-R
Total Omission	62	**0.967**	0.959	0.959	0.855	0.913
Definiteness Shift	12	**0.923**	**0.923**	0.774	**0.923**	0.800
Wrong Word Order	13	**0.960**	0.917	0.917	0.917	0.880
Tanween Omission	14	0.788	0.897	0.743	0.743	**0.963**
Invalid Pattern	10	**0.696**	0.552	0.541	0.625	0.640
Register Mismatch	96	0.649	0.657	0.653	0.601	**0.667**
Literal Translation	15	0.526	0.577	**0.811**	0.588	0.423
Hypernym for Hyponym	30	0.160	**0.475**	0.471	0.423	0.400
Terminology Substitution	52	0.382	0.417	0.458	0.494	**0.506**
Meaning Shift	70	**0.272**	0.225	0.108	0.198	0.228

Table 15: Per-class F1 (best fold) on the 10 MIZAN classes with ≥ 10 Gold support; bold marks the highest F1 per row.

```
You are an expert Arabic linguist generating synthetic MT error
examples for a UN-domain benchmark.

YOUR ROLE:
Generate NEW examples of specific translation error types. You
will
see a few real examples of each error type, then create NEW
instances
with NEW Keyword/Best_Match pairs that follow the same pattern of
linguistic shift.

DOMAIN GUIDANCE:
Generate content that resembles authentic United Nations texts (e.
g.,
Security Council resolutions, Secretary-General reports, official
UN
statements). Focus on political, diplomatic, legal, humanitarian,
or
conflict-related topics with formal, neutral, institutional tone.

MINIMAL PAIR REQUIREMENT:
The 'ar' (clean) and 'mt_output' (error) sentences MUST be
identical
except for the specified anchor edit (Keyword->Best_Match, or
Keyword
omitted). Do not introduce additional changes to vocabulary,
structure, or meaning beyond the specified error.

=== CRITICAL: DIVERSITY REQUIREMENTS FOR MODEL TRAINING ===
1. LEXICAL DIVERSITY: Use a COMPLETELY DIFFERENT keyword than any
   shown in examples or listed as "already used"
2. SYNTACTIC VARIATION: Use a DIFFERENT sentence opening than
recent
   examples
3. DOMAIN ROTATION: Draw vocabulary from the specified UN
subdomain
4. NO REPETITION: Never reuse keywords, sentence structures,
templates

=== HARD CONSTRAINTS ===
- Generate NEW Keyword DIFFERENT from seed examples AND "already
used"
- Keyword MUST appear exactly ONCE in 'ar'
- Best_Match MUST appear in 'mt_output' (unless Best_Match is [
OMITTED])
- 'ar' and 'mt_output' must be IDENTICAL except the anchor edit
- Formal Arabic, institutional tone, 15-30 words
- Start sentence with a DIFFERENT opening than recent examples
- Output ONLY valid JSON with exactly 5 keys:
  {"Keyword":"...","Best_Match":"...","ar":"...","mt_output
":"...","en":"..."}
```

Figure 4: GPT-4o generation system prompt (verbatim).

metrics cannot isolate. Second, **no encoder dominates**: the highest per-class F1 is distributed across all five encoders. Third, **semantic-divergence classes remain hard**: Meaning Shift peaks at 0.272 and Hypernym for Hyponym peaks at 0.475 across all encoders, highlighting the Frontier-tier classes where future work on representation or retrieval-augmented classification would yield the largest gains; that this pattern holds across all five encoders frames these as field-wide representational limits rather than encoder-specific failures, and augmenting the anchor with sentence-level semantic context (§6) is the most targeted next step.

The remaining 13 MIZAN classes have $n < 10$ gold instances, making per-class confidence intervals too wide for individual conclusions.

F Reproducibility

All experiments in this paper are fully reproducible from the released artefacts at the public repository linked in the title footnote.

End-to-end regeneration. Beyond reloading the released splits, QUANTA is reproducible *by construction*: the deterministic construction pipeline (anchor-gate extraction, the verbatim generation and audit prompts, and the DQI ≥ 0.75 audit threshold; Appendix C) regenerates the dataset from the public UN Parallel Corpus (Ziemski et al., 2016), so every result can be reproduced end-to-end from public source data rather than only reloaded. No institution-restricted data is involved.

Classifier training. Every classifier result uses the same training procedure: stratified 5-fold cross-validation (`random_state` $= 42$) on T1+T2 with the held-out 438-instance Gold test, with classes of fewer than 5 instances retained entirely in training. We train with learning rate 5×10^{-5}, batch size 32, 5 epochs, AdamW optimiser, weighted cross-entropy loss, a linear warmup of 10% of total training steps, mixed-precision training via `torch.amp.autocast`, and a maximum input length of 512 tokens. The best validation-fold checkpoint is selected by macro-F1 and evaluated on the held-out Gold test.

Significance testing. Pairwise paired-bootstrap tests (10,000 resamples on fold-level differences) with Benjamini–Hochberg FDR correction test for differences, and Two One-Sided Tests (TOST, equivalence margin $\delta = 0.05$) test for practical equivalence; both back the encoder comparison in

```
Role
You are an expert Arabic linguistics auditor evaluating whether a
specific annotated linguistic phenomenon is correctly instantiated
 in
a machine-translated Arabic sentence pair.

You are NOT judging overall translation adequacy.
You are NOT reclassifying labels.
You are NOT fixing or rewriting text.
Your task is to audit anchor-level correctness and usability of
this
sample as a Tier2 generation seed.

You will receive
- error_label: annotated Sub-Subtype (fixed; do not change)
- keyword: lexical anchor expected in the clean sentence
- best_match: expected transformation in the MT output (may be a
  lexical form or the sentinel [OMITTED])
- clean_ar: original Arabic sentence
- error_ar: machine-translated Arabic sentence

Your task
Evaluate only whether the annotated phenomenon (error_label) is:
1) correctly instantiated via keyword / best_match,
2) clearly observable in the clean vs. error Arabic pair,
3) linguistically plausible as MT output,
4) suitable for use as a Tier2 generation seed.
Judge with respect to the given label only. Do NOT consider
alternative labels. Do NOT infer meaning from any other language.

Evaluation axes (return five values)
1) anchor_validity:       1 = contradicts label, 5 = perfect
alignment
2) phenomenon_clarity:    1 = barely observable, 5 = unambiguous
3) arabic_naturalness:    1 = broken, 5 = fluent plausible MT
Arabic
4) collateral_severity:   1 = no unrelated distortion, 5 = severe
5) tier2_seed_recommendation: 0 = do NOT use, 1 = safe to use

Decision guidance
- Collateral changes are acceptable if the target phenomenon is
intact.
- Rare classes are valid if correct.
- If anchors are ambiguous or weak, set tier2_seed_recommendation
= 0.
- Never rescue by imagining intent -- judge only what is present.

Output format (STRICT)
Return exactly five TAB-separated values in this order:
anchor_validity  phenomenon_clarity  arabic_naturalness
collateral_severity  tier2_seed_recommendation
Valid example:  5  4  4  2  1

Hard constraints: No explanations. No text. No JSON. No additional
numbers. No class reassignment. Any deviation invalidates the
output.
```

Figure 5: Claude Sonnet 4 audit rubric (verbatim).

Table 5, with full per-pair results in Appendix G. Bootstrap 95% confidence intervals on best-fold predictions use 1,000 resamples.

Released artefact pointer. The QUANTA dataset (T1, T2, and the held-out Gold test set), the frozen 5-fold splits, the released AraBERT v2 checkpoints (all five folds; fold-0 default), the ETCA verbatim prompts, the post-hoc audit manifests, and the full benchmark logs are released with the paper. SHA-256 checksums, pinned dependency versions, and per-experiment reproduction bundles (each with a script that re-scores the reported numbers) ship in the repository's `REPRODUCE.md`.

Environment. Python 3.10, CUDA 12.1; pinned dependency versions in the repository's `requirements.txt`.

G Encoder Significance and Equivalence

Beyond reporting mean macro-F1, we test all ten encoder pairs two ways (Table 16): a paired bootstrap (10,000 resamples on the five fold-level differences, fixed seed) with Benjamini–Hochberg FDR correction tests whether two encoders *differ*, and a Two One-Sided Tests procedure (TOST, equivalence margin $\delta = 0.05$ macro-F1) tests whether they are *practically equivalent*. We report both because a non-significant difference is not, on its own, evidence of equivalence.

Only three pairs are significant after correction, all involving the multilingual XLM-R or the domain-adjacent ConfliBERT, while the three Arabic-specific encoders (AraBERT v2, CAMeLBERT-MSA, CAMeLBERT-Mix) are mutually equivalent within ± 0.05 macro-F1. Together these validate AraBERT v2 as the released encoder: it is significantly stronger than the non-Arabic and domain-adjacent baselines (both $p_{adj} \leq 0.007$) and statistically no worse than any other Arabic-specific encoder, while attaining the highest mean. ConfliBERT, although it is AraBERT v2 further pre-trained on conflict-domain Arabic, does not gain from that adaptation on this UN-domain Modern Standard Arabic task, consistent with the known trade-offs of domain-adaptive pre-training.

The FDR correction is consequential: AraBERT v2 vs. CAMeLBERT-MSA is significant before correction ($p_{\text{raw}} = 0.031$) but not after correcting for the ten comparisons ($p_{adj} = 0.078$). Reading the two tests jointly (Figure 6), two pairs are genuinely different (significant and non-equivalent: the two significant XLM-R comparisons), four are genuinely equivalent (non-significant and within ± 0.05: the two within-family Arabic pairs involving CAMeLBERT-Mix and the two CAMeLBERT–ConfliBERT pairs), AraBERT v2 vs. ConfliBERT is significant but small (its $+0.032$ gap sits near the margin, $p_{\text{TOST}} = 0.077$), and the remaining three are inconclusive at five folds.

H Anchor Framing and Data-Efficiency Ablations

This appendix gives the full numerical support for the two body-level claims in §5: that anchor-guided framing is the essential input signal, and that structurally validated real data is roughly an order of magnitude more sample-efficient than size-matched synthetic data on the 12 classes for which T1 has training support.

Pair	ΔF1	95% CI	p_{adj}	Diff?	p_{TOST}	Equiv?
AraBERT v2 vs. ConfliBERT	$+0.032$	$[+0.015, +0.052]$	< 0.001	yes	0.077	marginal
AraBERT v2 vs. XLM-R	$+0.050$	$[+0.018, +0.069]$	0.007	yes	0.490	no
CAMeLBERT-Mix vs. XLM-R	$+0.032$	$[+0.009, +0.057]$	0.015	yes	0.136	no
AraBERT v2 vs. CAMeLBERT-MSA	$+0.027$	$[+0.003, +0.050]$	0.078	no	0.087	marginal
AraBERT v2 vs. CAMeLBERT-Mix	$+0.017$	$[-0.004, +0.041]$	0.303	no	0.034	yes
CAMeLBERT-Mix vs. ConfliBERT	$+0.014$	$[-0.012, +0.035]$	0.355	no	0.026	yes
ConfliBERT vs. XLM-R	$+0.018$	$[-0.012, +0.043]$	0.355	no	0.056	marginal
CAMeLBERT-MSA vs. XLM-R	$+0.022$	$[-0.015, +0.059]$	0.355	no	0.139	no
CAMeLBERT-MSA vs. CAMeLBERT-Mix	-0.010	$[-0.037, +0.017]$	0.574	no	0.030	yes
CAMeLBERT-MSA vs. ConfliBERT	$+0.004$	$[-0.021, +0.033]$	0.773	no	0.023	yes

Table 16: Pairwise encoder comparison: the paired-bootstrap difference test (Benjamini–Hochberg FDR-corrected p_{adj}; significant when $p_{adj} < 0.05$) and the TOST equivalence test (margin $\delta = 0.05$ macro-F1; equivalent when $p_{\text{TOST}} < 0.05$). ΔF1 is the mean fold-level difference (first encoder minus second).

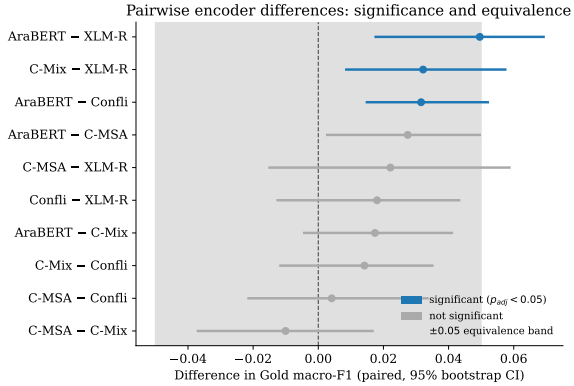


Figure 6: Pairwise encoder differences in Gold macro-F1 with 95% bootstrap CIs. A CI clearing the dashed zero line indicates a significant difference; the shaded band marks the ± 0.05 TOST equivalence margin, so both tests are visible in one view.

H.1 Data Ablation

The data ablation in Table 8 (§5) trains and evaluates each configuration (T1, T2, T1+T2) on the same 12 T1-supported Mizan classes, so the sample-efficiency comparison is not confounded by class coverage. The Data Efficiency Score is

$$\text{DES} = \frac{\text{F1}_{12}}{n/1000}, \tag{2}$$

where F1_{12} is the 12-class macro-F1 and n the number of training instances, i.e., macro-F1 per 1,000 samples; it is a transparent size-normalised contrast, not a scaling law. Bootstrap 95% confidence intervals on the 23-class macro-F1 (1,000 resamples on best-fold Gold predictions) confirm the combination: T1+T2 $[0.568, 0.671]$ does not overlap T2 $[0.401, 0.485]$. Real data contributes per-instance precision, synthetic data contributes volume and class-label coverage, and the combination yields the strictly best operating point.

H.2 Format Ablation (Anchor vs. Blind)

Table 17 compares AraBERT v2 under two input formats on identical folds and hyperparameters. Format A (blind): `[CLS] ar mt [SEP]`, the classifier sees only the Arabic reference and the MT output. Format D (anchor-guided): `[CLS] keyword best_match [SEP] ar mt [SEP]`, the classifier additionally receives the Mizan anchor pair that localises the error.

Format	Mean F1 ± std	Best fold
A (blind: `ar mt`)	0.1135 ± 0.030	0.1609
D (anchor: `kw bm + ar mt`)	$\mathbf{0.6185 \pm 0.023}$	**0.6397**
Δ	**+0.505** (**+445**%)	+0.479

Table 17: Input-format ablation on AraBERT v2, a controlled blind-vs-anchor run with its own 5-fold split; anchor-guided framing improves macro-F1 by $+0.505$ ($+445\%$, paired-bootstrap $p < 0.0001$). The Format-D mean here (0.6185) is from this ablation run, distinct from the main-benchmark mean (0.614).

The anchor is not a minor enhancement: it is the essential signal that enables fine-grained Arabic MT error classification.

I T2 Per-Class Generation Audit

Table 18 reports the T2 per-class count against the 369-sample-per-class generation target. Fourteen classes hit the target exactly; nine fell short, of which three are materially short ($< 90\%$ of target). The materially-short classes are hard to synthesise under the anchor constraint for class-specific mechanistic reasons (Spelling Error requires a minor orthographic perturbation that leaves context intact; Invalid Pattern requires a morphological mismatch the generator often over-repairs; single-token Definiteness Shift is blocked by the clitic-matcher's current scope limit). Weighted cross-entropy mitigates the residual class imbalance at training time.

J Cross-Language Dataset Comparison

Table 1 compares Mizan against prior error *taxonomies*; Table 19 instead positions the Quanta *dataset* against fine-grained MT-error and diagnos-

Mizan class	Count	% of 369
Active-to-Passive Voice	368	99.7%
Perfective-to-Progressive	367	99.5%
Gender Disagreement	366	99.2%
Register Mismatch	365	98.9%
Adjective-to-Noun	359	97.3%
Noun-to-Adjective	357	96.7%
Definiteness Shift	**320**	**86.7%**
Invalid Pattern	**263**	**71.3%**
Spelling Error	**113**	**30.6%**
14 classes at target	369	100.0%

Table 18: T2 per-class shortfall against the 369-sample generator target: 14 classes hit it, 9 fell short (3 materially, $< 90\%$, bold). Generation-stage counts (8,044), before the 188-confound exclusion (§4.4).

tic datasets in other languages. ACES (Amrhein et al., 2022) and DEMETR (Karpinska et al., 2022) are perturbation-based sets for meta-evaluating MT *metrics*; the WMT MQM corpus (Freitag et al., 2021), IndicMT-Eval (Sai B et al., 2023), and SiniticMTError (Liu et al., 2025) provide human error annotation for European, Indian, and Sinitic languages respectively; and WMT24++ (Deutsch et al., 2025) supplies references and post-edits for 55 languages including Arabic, but without error-type labels. Fine-grained MT-error annotation is thus an established direction across several languages, yet none of these resources targets Arabic with an anchor-verified, error-typed dataset paired with a diagnostic classifier. To our knowledge, Quanta is the first fine-grained ($\geq$ 20-class) Arabic MT-error dataset.

K Ethics and Broader Impact

Data source and licensing. Quanta is derived entirely from the United Nations Parallel Corpus (Ziemski et al., 2016), the public corpus of institutional UN documents also used by Osorio et al. (2025). The Quanta dataset (the Mizan annotations and taxonomy), the released Raqeeb classifier, and the verbatim Etca prompts are publicly available under the CC-BY-4.0 license at the repository linked in the title footnote (`https://github.com/AALight/Quanta-Mizan`); the underlying UN source text remains subject to the UNPC terms.

Annotation ethics and team. This work does not involve human research subjects; all annotators are authors of the paper. The Mizan taxonomy was designed by two authors with doctoral-level expertise in translation and in linguistics; the dual-annotator validation (Appendix B) was performed independently by two further authors, both expert Arabic linguists (one doctoral-, one master's-level), who were not involved in taxonomy design, each instance labelled independently by both in per-instance CSV workbooks. No personally identifying information about contributors is reported beyond academic degree and research specialisation. The UN content is formal institutional text with no graphic, harmful, or distressing material.

Scope and cultural sensitivity. Mizan describes general Arabic morphological phenomena, but the released Raqeeb classifier is calibrated and validated on formal Modern Standard Arabic (the UN Security Council domain); dialectal use would require additional validation.

LLM use disclosure. T2 synthetic instances are generated by GPT-4o (OpenAI, 2024); the Etca audit uses Claude Sonnet 4 (Anthropic, 2025). Cross-vendor generator-judge separation (§4.2) is deliberate and mitigates same-family LLM self-preference bias (Kim, 2025; Wang et al., 2026). Model outputs are not treated as ground truth: they are structurally validated and gold-evaluated against expert-annotated instances.

Intended use and broader impact. The released Raqeeb classifier is designed as a *diagnostic tool* for fine-grained Arabic MT error-type analysis in high-quality formal-text translation domains (institutional, legal, medical). It is not a scalar quality score and is not intended to replace human translators, to rank MT systems without expert review, or to serve as an automatic MT-quality gate in production without human review. The 23-class classifier has documented Frontier-tier limitations (§5); deployment outside the formal-text scope it was calibrated on requires recalibration and additional validation. Arabic is one of the lowest-resourced languages for fine-grained MT evaluation despite being a top-10 global language by speaker count. Existing taxonomies operate at 3–7 top-level categories and do not capture Arabic-specific morphological phenomena. Quanta and Mizan address this gap with a 23-class taxonomy, a publicly released 9,032-instance dataset, and a documented cross-vendor LLM-as-judge audit protocol that the community can replicate, build on, and re-calibrate to other formal-text domains, for which the released taxonomy, data, and classifier provide the starting point. Positive-impact applications include reducing annotator workload in large-

Dataset	Lang. pair(s)	Size	Error classes	Validation / notes
ACES (Amrhein et al., 2022)	146 pairs (en-x, x-en, x-y)	36,476 examples	68 phenomena (MQM ontology)	Synthetic contrastive; MT-metric challenge set
DEMETR (Karpinska et al., 2022)	10 pairs (X→En)	∼31,000 examples	35 perturbation types	Synthetic minimal pairs; MT-metric diagnostic
WMT MQM (Freitag et al., 2021)	2 pairs (En→De, Zh→En)	3,418 segments (1,418 + 2,000)	22 MQM categories	Human span annotation (MQM)
IndicMT-Eval (Sai B et al., 2023)	5 pairs (En→Indic)	7,000 annotations (1,400/lang.)	13 MQM categories	Human MQM (span, severity)
SiniticMTError (Liu et al., 2025)	4 pairs: En→Mandarin, Cantonese, Wu; Mandarin→Hokkien	3,633 sents (2,009 + 1,402 + 68 + 154, pilots)	9 error types	Human span, type, severity
QUANTA (Ours)	1 pair (En→Ar)	9,032: 738 real + 7,856 synth + 438 gold	23 MIZAN classes	Dual-annotator AC1 0.83–0.97; ETCA audit; anchor-constrained

Table 19: QUANTA positioned against fine-grained MT-error and diagnostic datasets in other languages. ACES and DEMETR are synthetic sets for meta-evaluating MT *metrics*; WMT MQM, IndicMT-Eval, and SiniticMTError are human error-annotated sets for European, Indian, and Sinitic languages. for WMT MQM the size is the WMT20 En–De and Zh–En source segments annotated with MQM by professional translators; SiniticMTError's Wu (68) and Mandarin–Hokkien (154) splits are small pilots. QUANTA is the only Arabic, anchor-verified, error-typed dataset at 20+ classes, and is released with a diagnostic classifier.

scale Arabic MT evaluation by pre-filtering structurally valid candidates and supporting expert post-editors with per-error-type diagnostic signals at the token level.

Beyond Monolithic Libyan Arabic: A Sub-Regional Speech Corpus and Dialect Identification Benchmark

Nour A. Zohdy*
Sebha University
Sebha, Libya
nour.tammous@idp.sebhau.edu.ly

Mansour Essgaer
Sebha University
Sebha, Libya
man.essgaer@sebhau.edu.ly

Abstract

Spoken Arabic dialect identification (DID) has advanced quickly, but Libyan Arabic remains under-served and, where it appears, is labeled only at the country level, hiding the variation between the Libyan coast and interior. We present a Libyan Arabic speech corpus annotated at the sub-regional level: Tripolitania (west), Cyrenaica (east), and Fezzan (south). It contains 133.7 hours of single-speaker segments from 645 speakers across 161 channels (YouTube and TikTok sources). An automated pipeline (diarization, speaker embedding, clustering, cross-episode linking) produces silver labels from channel provenance. One annotator then reviewed the resulting speaker clusters and confirmed the provenance region for 95.5% of the kept speakers (a within-annotator figure). We release speaker-disjoint splits. An off-the-shelf Arabic DID model recognizes our verified Libyan speech as Libyan only 74% of the time and cannot resolve the region. On the new three-way task, the encoder matters more than the classifier: a probe over speaker-recognition ECAPA embeddings reaches 0.525 macro-F1, over a language-identification ECAPA reaches 0.660, and fine-tuning that encoder reaches 0.702 (75.5% accuracy), far above the 0.20 floor. A channel-disjoint re-test reaches 0.609, versus 0.702 speaker-disjoint; Fezzan is hardest at 0.313. The task is learnable, but Fezzan performance remains low. We release the corpus, cached embeddings, and code.[1]

1 Introduction

Arabic is a collection of spoken varieties that differ enough to degrade speech systems trained on another variety. Dialect identification (DID) predicts which variety a recording is in, and it has become a standard pre-processing step for Arabic ASR and a benchmark task in its own right (Shon et al., 2020; Talafha et al., 2025). The field has converged on a small set of broad labels, usually one per country or one per region group.

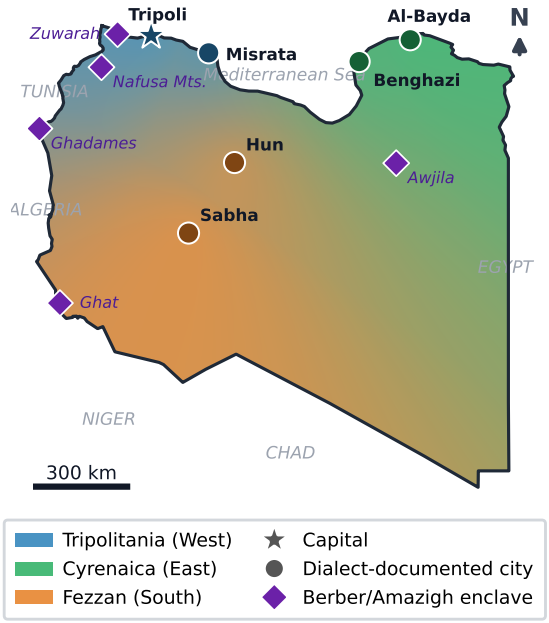


Figure 1: The three Libyan Arabic macro-dialect regions used as labels: Tripolitania (west), Cyrenaica (east), and Fezzan (south). The fill is a soft inverse-distance blend of each region's dialect-documented cities (stars and dots), not a hard border, reflecting the gradual transitions reported in the literature (Pereira, 2010; Benkato, 2014). Diamonds mark Berber/Amazigh enclaves, which lie outside the Arabic continuum.

Libyan Arabic has received little of this attention. Large multi-dialect collections such as ADI-17 and ADI-20 (Shon et al., 2020; Elleuch et al., 2025) represent it only as a single country-level class, and no prior resource labels it below that level. On our human-verified data, an off-the-shelf Arabic dialect classifier labels only 74% of playlist-level speaker clusters as Libyan and scatters the rest across Egyptian, Algerian, and Levantine labels (Section 5).

*Corresponding author.

[1]Released at https://huggingface.co/datasets/NourZohdy/libyan-subregional-did; archived at https://doi.org/10.5281/zenodo.22802659.

Proceedings of The Fourth Arabic Natural Language Processing Conference (ArabicNLP 2026), pages 485–496
October 28-29, 2026 ©2026 Association for Computational Linguistics

Consistent with known cross-corpus degradation in Arabic DID (Shon et al., 2020; Sullivan et al., 2023), country-level tools do not transfer to these data or resolve their regions.

A second problem is the label itself. Treating "Libyan" as one dialect ignores variation that speakers themselves notice. The country splits into three historical regions with different settlement histories and contact languages: Tripolitania in the west, Cyrenaica in the east, and Fezzan in the south (Figure 1) (Pereira, 2010; Benkato, 2014; Benkato and Pereira, 2016). These regions differ in lexicon and phonology, yet no speech resource labels Libyan data at this level. A model trained on country-level labels cannot use this variation, and a dataset without regional labels cannot show whether the variation is recoverable from audio.

We address both gaps with a corpus and a benchmark. Our contributions are:

- **A sub-regional Libyan Arabic speech corpus.** 133.7 hours of validated single-speaker segments from 645 speakers and 161 channels, each labeled Tripolitania, Cyrenaica, or Fezzan (Section 3).
- **A partition and validation methodology.** A reproducible pipeline that turns raw broadcast and social-media audio into speaker-disjoint, region-labeled segments. It pairs provenance-based regional tagging with a human review protocol for checking and correcting those initial labels: channel-normalized merge bands, an evidence floor, and a risk-ranked review whose decisions confirm the provenance region 95.5% of the time (within-annotator consistency; Sections 3–4).
- **A three-way DID benchmark.** Speaker-disjoint splits and a set of models, from a majority floor and an off-the-shelf classifier through frozen-embedding probes to an end-to-end fine-tuned encoder, with per-class results and an analysis of which regions are separable. The strongest model, a fine-tuned language-identification ECAPA, reaches 0.702 macro-F1 against a 0.20 floor (0.609 under a channel-disjoint split); Fezzan is the hardest class (Section 5).

The benchmark also tests the labels. Our regional labels start from the channel's known origin, not from per-utterance perceptual annotation. If a classifier trained on these labels generalizes to unseen speakers above chance, the labels carry a learnable regional signal. Section 5 shows that they do, although part of that signal is channel-correlated (Section 5.7).

2 Related Work

Arabic dialect identification corpora. Most spoken Arabic DID work builds on broadcast collections. The MGB challenges released Egyptian and Moroccan broadcast speech (Ali et al., 2017, 2019), and ADI-17 turned YouTube audio into a 17-way country-level dialect task (Shon et al., 2020). ADI-20 extended this direction to twenty varieties (Elleuch et al., 2025), and QASR (Mubarak et al., 2021) and MASC (Al-Fetyani et al., 2023) added large transcribed Arabic broadcast corpora. The NADI shared tasks have run for several years and cover both text and, more recently, spoken dialect identification at the country level (Talafha et al., 2025). Casablanca (Talafha et al., 2024) and CAFE (Lachemat et al., 2024) are recent multi-dialect speech resources. None of these label data at the sub-national level for Libya.

Libyan Arabic in prior work. Libyan appears as one class in country-level collections such as ADI-20 (Elleuch et al., 2025) and in recent Libyan-focused studies (Essgaer et al., 2025), but always as a single label. To our knowledge no prior speech resource separates Tripolitania, Cyrenaica, and Fezzan. Our corpus extends the country-level Libyan class of ADI-20 with labels for three regions inside Libya. Casablanca also sources dialogue from television drama and contains code-switching, which supports our choice of sources and our quality gates. It covers eight national dialects and no Libyan.

Models for spoken DID. Early systems used i-vectors and x-vectors with a back-end classifier (Snyder et al., 2018). Strong speaker and language embeddings such as ECAPA-TDNN (Desplanques et al., 2020) and the VoxLingua107 language-ID model (Valk and Alumäe, 2021) are now common encoders, and self-supervised models like wav2vec 2.0 (Baevski et al., 2020), XLS-R (Babu et al., 2022), and HuBERT (Hsu et al., 2021) provide transferable features for low-resource dialect tasks (Sullivan et al., 2023). We compare two ECAPA encoders—a speaker-recognition model and the VoxLingua107 language-identification model (Valk and Alumäe, 2021), which share an

architecture but differ in pretraining objective—alongside a frozen self-supervised encoder, and we fine-tune the language-identification ECAPA end to end (Section 5). Table 1 places our corpus against prior resources.

3 The Corpus

3.1 Sources and regions

The audio comes from public Libyan media: television drama, talk shows, and podcasts on YouTube, plus short clips from TikTok. A *source* is one YouTube channel or playlist, or one TikTok account; the paper calls this unit a *channel* throughout. A *recording* is one video or clip from a source. The released corpus has 161 sources: 57 on YouTube and 104 on TikTok. TikTok supplies 15.2 of 133.7 hours (11%) but 65% of the sources. We included its short clips to obtain informal speech from more creators; we expected YouTube's scripted and broadcast material to use a more formal register. TikTok also supplied additional Fezzan data after the available broadcast sources ran out. We group every source into one of three regions using its known origin. The channel's region serves only as an initial, provenance-based label for each segment. Every candidate speaker that passed the evidence floor (Section 3.3) was reviewed and could be confirmed, corrected, or excluded; released labels are the three broad regions rather than city-level or personal dialect labels. The same channel-provenance assumption underlies ADI-17 and similar broadcast DID corpora (Shon et al., 2020).

3.2 Automated pipeline

Each channel is processed independently, in four stages.

Segmentation. We run speaker diarization (Bredin et al., 2020) to split each episode into single-speaker turns, then cut long turns at low-energy pauses to keep segments between three and ten seconds. Without hand-aligned speaker timestamps, we report no diarization error rate. The quality gates below and human speaker review reduce segmentation errors but do not measure them.

Embedding and quality gates. Every segment gets a 192-dimensional ECAPA-TDNN speaker embedding (Desplanques et al., 2020; Ravanelli et al., 2021). We then drop non-speech and low-quality segments with three gates. The music gate is deliberately a conjunction: a segment is dropped only if a PANNs audio tagger (Kong et al., 2020) gives it a high music score *and* a low speech score. Libyan drama often has speech over a music bed, and this speech is valid data. On the playlist used to calibrate the filter, a music-only rule wrongly flagged 27% of the segments. The conjunction reduced the flags from 681 to 8. We listened to these eight segments and confirmed that they were correct drops. This check covers one playlist; it is not a precision or recall estimate for the music gate. A voice-activity speech-ratio floor (Silero Team, 2024) drops applause and silence-padded turns. A signal-to-noise floor drops unusable noise.

Clustering and linking. Within each channel (one playlist or source), agglomerative clustering groups segment embeddings into speakers after a median absolute deviation (MAD) audit removes off-centroid segments. Drama playlists record all speakers in one studio, which inflates every pairwise cosine similarity between speaker centroids; a fixed merge threshold over-merges. A pair must instead be an outlier against *that playlist's own* similarity distribution, with a floor of $\max(0.60,\ \text{median} + 4 \cdot 1.4826 \cdot \text{MAD})$. A fixed threshold queued more than 600 pairs for review on the largest playlists alone; the adaptive band queued 75 across all playlists. We then compare centroids across episodes and channels; automatic links stay within one region, so a recurring host becomes one identity. A confident cross-region match is never merged automatically: it means one of the two region labels is wrong, and only a human resolves it.

Triage scoring. For every candidate speaker we compute a risk score from an off-the-shelf HuBERT Arabic dialect classifier,[2] the VoxLingua107 language-ID model (Valk and Alumäe, 2021), embedding cohesion, a music fraction, and a predicted-gender mix. The risk score only orders the human review queue, hardest cases first; it rejects and relabels nothing.

3.3 Human validation

The automated output is silver-quality. Before review, an evidence floor removed very small clusters, those with under 45 s of speech or fewer than five segments. This removed 1,300 of 2,069 auto-

[2] `IbrahimAmin/hubert-arabic-spoken-dialect-classifier` on the Hugging Face Hub (Amin, 2025): a HuBERT model (Hsu et al., 2021) fine-tuned on QASR and ADI-17 to label Modern Standard Arabic and 17 regional dialects, with a single Libyan (LIB) class.

Resource	Source	Granularity	Libyan	Sub-national
ADI-17 (Shon et al., 2020)	YouTube	17 countries	yes	no
ADI-20 (Elleuch et al., 2025)	broadcast/YT	20 varieties	yes	no
NADI 2025 (Talafha et al., 2025)	TV series	8 countries	no	no
Casablanca (Talafha et al., 2024)	TV	8 dialects	no	no
MGB-3/5 (Ali et al., 2017, 2019)	broadcast	EGY / MOR	no	no
This work	broadcast/social	3 Libyan regions	yes	**yes**

Table 1: Our corpus against prior Arabic spoken-dialect resources. Prior resources that include Libyan label it only at the country level.

matically linked speaker identities, containing only 3.8% of the speech (5.4 hours). The 769 above-floor speakers remained. One annotator—an author raised in Libya and familiar with its regional dialects—reviewed each candidate speaker cluster in a custom interface. The review screen showed the channel name, provenance region, proposed label, tier, and triage risk score. It showed 3–10 clips from the start, middle, and end of each speaker's timeline, preferring clean, long ones.

Each reviewed speaker received one decision: *verified*, *wrong_region* (with a corrected region), *non_libyan*, *msa*, *bad_audio*, *mixed_cluster*, or *unsure*. Of 769 reviewed speakers, we excluded 76 (34 *msa*, 28 *bad_audio*, 13 *non_libyan*, and 1 *mixed_cluster*) and kept 693 (90.1%). Of those kept speakers, 31 were Libyan but *wrong_region* and were relabelled. Table 9 (Appendix A) separates the counting units used in validation. The 693 kept speakers correspond to 799 playlist-level speaker clusters before cross-playlist merging, which is the unit used in Section 5.1 and Figure 4. After manual pair matching and merging across playlists, 645 final speakers remained. Thus, 799, 693, and 645 describe the same kept speakers in different counting units, while 769 is the earlier count before the 76 exclusions.

The annotator saw the proposed label and channel region before each decision, which may have anchored them. The classifier in Section 5.1 fed the triage score, so its evaluation partly uses labels it helped shape.

The channel-provenance region was confirmed for 95.5% of kept speakers (within-annotator consistency); of the 31 corrections, 20 moved from Tripolitania to Cyrenaica, 9 from Cyrenaica to Tripolitania, 1 from Tripolitania to Fezzan, and 1 from Fezzan to Cyrenaica. The risk score ranked rejected speakers ahead of kept ones (AUC 0.82, Figure 2). Automatic MSA probability flagged only 12 of 769 speakers above 0.5, while human

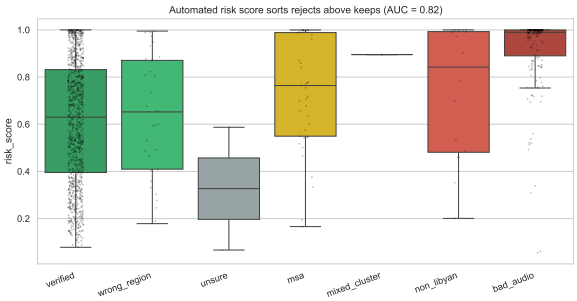


Figure 2: Triage risk score vs. human decisions. Higher risk concentrates among rejected speakers (AUC 0.82).

review removed 34 MSA and 13 non-Libyan speakers. The kept corpus is therefore mainly dialectal rather than read speech (speech read from a prepared text, usually news-style MSA with little regional dialect), although the segment audit below shows that some MSA remains at segment level.

Blind second-annotator review. A second Libyan-raised annotator, also a co-author, rejudged all 645 released speakers blind to provenance, channel, and triage information. He confirmed 525 labels (81.4%, $\kappa = 0.72$); Table 2 gives results by region, without per-region κ because the released label is constant within a region. This is lower than the 95.5% within-annotator rate. Two annotators do not always agree on the region of a speaker from short clips. We therefore call the released labels reviewed provenance labels rather than gold labels. The benchmark measures how well a model reproduces these reviewed labels; the disagreement limits how far they can be read as perceptual dialect judgments.

Segment-level purity audit. The first annotator reviewed 207 test segments, 69 per region, in random order and without seeing the released label (Table 3). Because the first annotator performed this audit, its region field measures segment purity and within-annotator consistency, not inter-annotator agreement. Code-switching and mixed-speaker au-

Released region	n	Confirmed	Trip.	Cyr.	Fez.	Other
Tripolitania	275	77.8%	214	40	16	5
Cyrenaica	240	82.1%	11	197	22	10
Fezzan	130	87.7%	6	8	114	2
All	645	81.4%	231	245	152	17

Table 2: Blind second-annotator review by released region: speakers he judged as Tripolitania, Cyrenaica, Fezzan, or other (13 MSA, 3 non-Libyan, 1 mixed).

Property	Trip.	Cyr.	Fez.	Weighted
MSA	13.0	15.9	33.3	17.3
Code-switching	0.0	1.4	0.0	0.6
Mixed speakers	0.0	0.0	0.0	0.0
Indeterminate	1.4	2.9	1.4	2.1
Different region	14.5	14.5	26.1	16.2
Cannot tell	17.4	23.2	49.3	24.7

Table 3: Segment-level purity audit (%) on the released test split: 23 segments per region × duration band, at most three per speaker; weighted rates use test-split region shares. Wilson 95% intervals are in the released CSV.

dio were rare, whereas MSA and uncertainty about region were common. Fezzan stood out, with an MSA rate more than double that of the coastal regions. A 3–10 s clip was often too short to place a speaker, with 37.7% of short segments marked *cannot tell*. The low code-switching count, 1 of 207 segments, does not mean that the corpus has no code-switching. This audit used one listener judging short clips, so rare events are not well measured; a larger segment-level audit would be needed to bound them.

3.4 Statistics

Table 4 reports the corpus after validation; it contains 78,411 segments with a median segment length of 6.0 seconds. Fezzan is smaller than the other regions, which reflects how little southern Libyan media we could find online.

The speaker distribution is heavy-tailed. A few recurring hosts dominate: the ten largest speakers hold 26% of all audio, and the Gini coefficient over speaker-hours is 0.73 (Figure 3). This is the main reason our splits must be speaker-disjoint (Section 4). The corpus is also skewed by gender: about 83% of speakers are male, consistent with a source mix of mainly TV drama, plus talk shows, podcasts, and short social-media clips; Fezzan is the most balanced, at 26% female speakers.

Region	**Hours**	**Spk**	**Chan.**	**Ep.**	**Clean**
Tripolitania	56.7	275	72	686	92.5
Cyrenaica	56.4	240	64	478	95.7
Fezzan	20.6	130	43	165	92.6
Total	133.7	645	161	1,281	93.9

Table 4: Corpus statistics by region; **Clean** is the percentage of segments above a 15 dB SNR threshold. Channel and episode totals are corpus-unique and are smaller than the column sums because validation can move a speaker to a region other than its channel default.

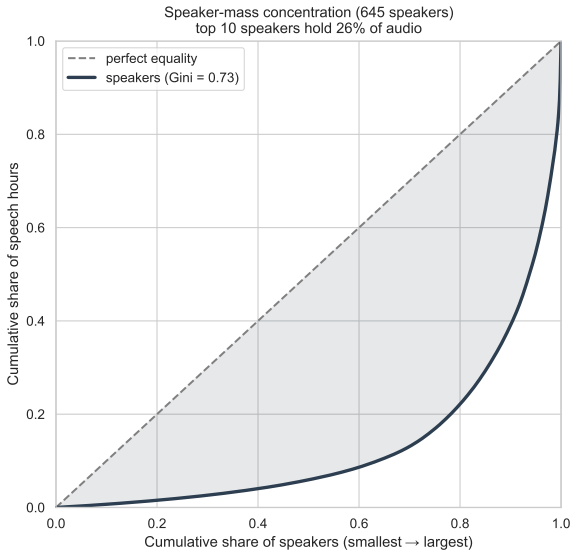


Figure 3: Lorenz curve of speaker-hours. A small number of recurring hosts hold a large share of the audio (Gini 0.73).

3.5 Release

We do not redistribute source audio from YouTube or TikTok. We release per-segment manifests with the source identifier, start and end times, region label, anonymous speaker identifier, and split assignment, plus the processing pipeline that creates the segments. For TikTok items, the manifest lists the account handle and local recording filename, so the source identifier can rebuild audio only for the YouTube segments. The release also includes cached 192-dimensional speaker-recognition ECAPA and 256-dimensional VoxLingua107 LID-ECAPA embeddings, both from pretrained models only. It excludes embeddings from the fine-tuned E6, so no label information enters the released features. The cached embeddings let researchers reproduce the probe benchmarks even if source videos disappear. The manifests and embeddings are under CC BY 4.0 and the processing

Split	Trip.	Cyr.	Fez.	Total	Spk
train	45.4	45.1	16.5	107.0	280
dev	5.7	5.6	2.1	13.4	183
test	5.7	5.6	2.1	13.4	182

Table 5: Split composition in hours per region (rounded to 0.1 h) with distinct speaker counts. Rounded cells need not sum exactly to the totals in Table 4.

code under the MIT License. These licenses do not cover the underlying YouTube or TikTok audio, and we will remove any speaker's entries on request.

4 Dataset Splits

We release train/dev/test splits that are speaker-disjoint and stratified by region. A speaker (after cross-episode merging) appears in exactly one split. We assign whole speakers greedily, largest first, to the split furthest below its target share, balancing by audio duration within each region. The target ratio is 80/10/10 by duration.

Speaker-disjointness matters here because of the heavy tail in Figure 3. If the same host appeared in both train and test, a model could reach high accuracy by recognizing the host rather than the dialect. Table 5 shows the result: every region is present in dev and test, and even Fezzan has about two hours and 34 speakers in each evaluation split. Dev and test carry near-identical duration but differ by a single speaker (183 vs. 182): speakers are assigned whole, so matched duration targets need not yield matched speaker counts.

5 Dialect Identification Benchmark

5.1 Motivation: off-the-shelf DID misses Libyan (E1)

Before building a sub-regional model, we checked how an existing Arabic DID system treats our data. We ran the same off-the-shelf HuBERT Arabic dialect classifier used in triage over 799 playlist-level speaker clusters, before cross-episode merging produced the 645 released speakers. For each cluster, we scored up to 10 long, clean segments, averaged the softmax outputs, and took the argmax as the cluster prediction. It labels only 74% of all clusters as Libyan (and 78% of the held-out test split), and the error rate is worst for Fezzan (68% recall). When it is wrong, it spreads the predictions across Syrian, Jordanian, Egyptian, Algerian, and Gulf labels rather than confusing Libyan with

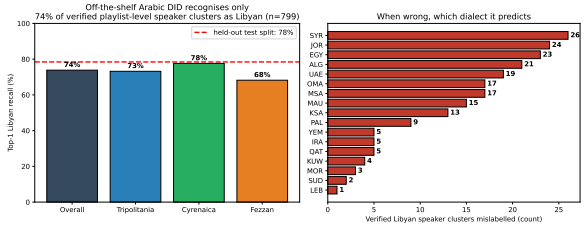


Figure 4: Off-the-shelf Arabic DID on 799 verified Libyan playlist-level speaker clusters, before cross-episode merging. Left: top-1 Libyan recall, overall and per region. Right: when wrong, which dialect the model predicts.

one neighbor (Figure 4); this pattern is consistent with documented cross-corpus degradation in Arabic DID (Shon et al., 2020; Sullivan et al., 2023). The model is confident the audio is Arabic (median Arabic probability 0.99) but does not consistently place it as Libyan. Because this classifier also fed the triage score, the 74% is a recall estimate on human-verified clusters, not an unbiased population estimate; the human, not the model, made every keep decision. The classifier also has no sub-national Libyan classes, so it cannot perform the task we define next.

5.2 Task and protocol

The task is three-way segment classification: given a speech segment, predict Tripolitania, Cyrenaica, or Fezzan. We train on the train split, select on dev, and report on test. We report accuracy and macro-averaged F1; macro-F1 is the headline metric because the regions are imbalanced and it weights each region equally. We report both segment-level results and speaker-level results, where a speaker's prediction is the majority vote over its segments.

5.3 Models

Majority (E0). Predict the most frequent training region for everything. This is the floor. **Off-the-shelf (E1).** The country-level classifier of the previous subsection, which has no Libyan subclasses and serves as motivation, not as a sub-regional model. **ECAPA + probe (E2).** A linear classifier on the cached 192-dimensional ECAPA speaker embeddings (Desplanques et al., 2020); this reuses embeddings the pipeline already computed, so it costs no extra inference. We report both a logistic-regression and a linear-SVM head. **XLS-R + probe (E3).** A logistic-regression probe over mean-pooled hidden states from a frozen XLS-R-300m encoder (Babu et al., 2022), with

no fine-tuning. This feature is more phonetic than the speaker-oriented ECAPA embedding. **Fusion (E4).** A logistic regression on the concatenation of the ECAPA and XLS-R features, to test whether the speaker-oriented and phonetic features carry complementary regional cues. **LID-ECAPA + probe (E5).** The same linear probe, but over 256-dimensional embeddings from the VoxLingua107 language-identification ECAPA (Valk and Alumäe, 2021): the same ECAPA-TDNN architecture as E2, pretrained to discriminate 107 languages rather than speakers. **LID-ECAPA fine-tuned (E6).** The VoxLingua107 ECAPA fine-tuned end to end with a fresh three-way head, the only model here that updates the encoder. To counter the Fezzan imbalance we use class-weighted cross-entropy with square-root inverse-frequency weights and SpecAugment on the input features. Full hyperparameters for every model and the significance-test procedures are in Appendix B.

		Dev		Test	
Model	**Lvl**	**Acc**	**F1**	**Acc**	**F1**
Majority (E0)	seg	43.1	20.1	43.2	20.1
ECAPA+logreg (E2)	seg	55.3	50.4	57.4	52.5
ECAPA+SVM (E2)	seg	55.8	50.6	58.0	52.7
XLS-R+probe (E3)	seg	52.8	42.2	56.6	44.4
Fusion (E4)	seg	56.2	51.2	58.4	53.4
LID-ECAPA+probe (E5)	seg	68.3	64.7	69.0	66.0
LID-ECAPA-FT (E6)	seg	73.6	68.6	75.5	**70.2**
ECAPA+logreg (E2)	spk	54.6	51.1	57.1	50.5
XLS-R+probe (E3)	spk	57.4	45.1	59.3	46.6
Fusion (E4)	spk	57.4	54.0	57.7	51.7
LID-ECAPA+probe (E5)	spk	77.1	75.2	78.0	73.4
LID-ECAPA-FT (E6)	spk	74.9	69.2	79.7	**75.0**

Table 6: Three-way sub-regional DID (%). Macro-F1 is the headline metric; best test F1 per level in bold. Model E1 (the off-the-shelf country-level classifier, §5) has no sub-regional classes and cannot produce three-way predictions, so it is omitted here. The E5→E6 macro-F1 gain has a speaker-clustered 95% CI of $[-0.011, 0.093]$.

5.4 Results

Table 6 reports the benchmark. The frozen language-ID ECAPA probe (E5) reaches 0.660 test macro-F1 and fine-tuning it (E6) reaches 0.702 (95% bootstrap interval $[0.691, 0.714]$). The encoder swap is the largest single gain: E5 exceeds the fusion E4 by $+0.126$ macro-F1, with a speaker-clustered bootstrap 95% CI $[0.071, 0.182]$ and McNemar $p < 10^{-60}$. Fine-tuning from E5 to E6 adds $+0.043$ per segment (segment bootstrap $[0.031, 0.055]$; McNemar $p < 10^{-32}$): E6 corrects 1,217 test segments that E5 got wrong and turns 695 correct E5 predictions into errors. This gain is not significant under the speaker-clustered bootstrap, CI $[-0.011, 0.093]$. E5 and E6 reach about the same Fezzan F1, so the E6 improvement lies in the two coastal regions. At speaker level, E5 leads E6 on dev and E6 leads E5 on test (Table 6). Model selection was fixed in advance on dev segment-level macro-F1; speaker-level voting was computed afterwards and never used for selection. For speaker-level prediction, the frozen E5 probe reaches 0.734 test macro-F1 against 0.750 for E6, without any end-to-end fine-tuning.

The per-class numbers in Table 7 show which regions account for the gains. E3 has the highest Fezzan precision of any model (0.87), yet it almost never predicts Fezzan, so recall collapses to 0.066 and F1 to 0.12. E6 improves every region, with Fezzan F1 rising from 0.36 in E2 to 0.53. Most of the Fezzan gain comes from the encoder: the frozen LID probe E5 already reaches Fezzan F1 0.55, marginally above E6. End-to-end fine-tuning with class weighting sharpens the two data-rich coastal regions at a small cost to Fezzan recall (0.45). Fezzan remains the hardest class, but E5 and E6 predict it far more often than E3 does.

5.5 Analysis

The confusion matrix for the fine-tuned model (Figure 5) shows where the difficulty remains. Fezzan is the main source of error: even fine-tuned it reaches only 45% recall, and its missed segments go mostly to Tripolitania (41%) rather than Cyrenaica (15%). Fezzan has the least training data (16.5 hours), which lowers its recall. Class size does not explain this direction: the two coastal regions have almost the same training hours (45.4 and 45.1). One hypothesis is that formal broadcast settings lead speakers, many of them educated, to shift toward MSA and accommodate toward the Tripoli variety by softening southern features. The segment audit points in the same direction as the accommodation hypothesis: 33.3% of Fezzan test segments were judged MSA. Among dialectal Fezzan test segments, the first annotator heard Tripolitania as often as Fezzan (12 segments each, versus 6 Cyrenaica); for no other released region did the annotator choose a neighbouring region this often. Genuine dialect closeness remains a possible cause, and neither the model nor the audit can separate the two effects. Corpus composition and the small

	spk-ECAPA (E2)			XLS-R (E3)			LID-ECAPA frozen (E5)			LID-ECAPA-FT (E6)		
Region	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
Tripolitania	.598	.565	.581	.542	.673	.601	.671	.728	.699	.705	.888	.786
Cyrenaica	.610	.660	.634	.586	.635	.609	.746	.709	.727	.855	.733	.789
Fezzan	.377	.345	.361	.868	.066	.123	.582	.527	.553	.655	.448	.532
Macro	.528	.524	.525	.665	.458	.444	.666	.655	.660	.738	.690	.702

Table 7: Per-class precision, recall, and F1 on the test split (segment level) for models E2, E3, E5, and E6. Macro rows are averaged at full precision and may differ from the rounded per-class mean by about 0.001.

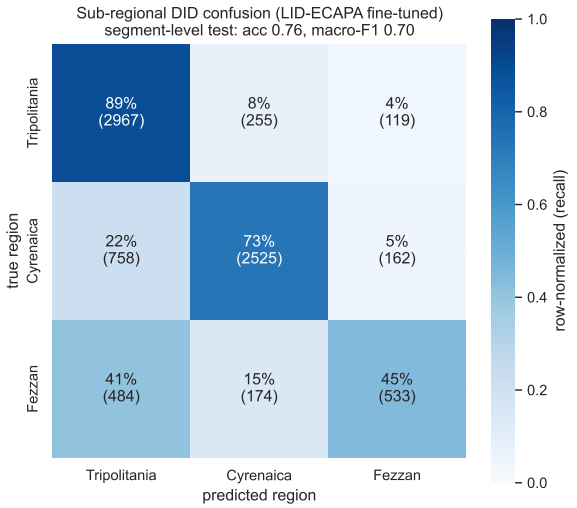


Figure 5: Test-set confusion for the fine-tuned LID-ECAPA model (E6), row-normalized recall. Fezzan is the hardest region and is confused far more often with Tripolitania (41%) than with Cyrenaica (15%).

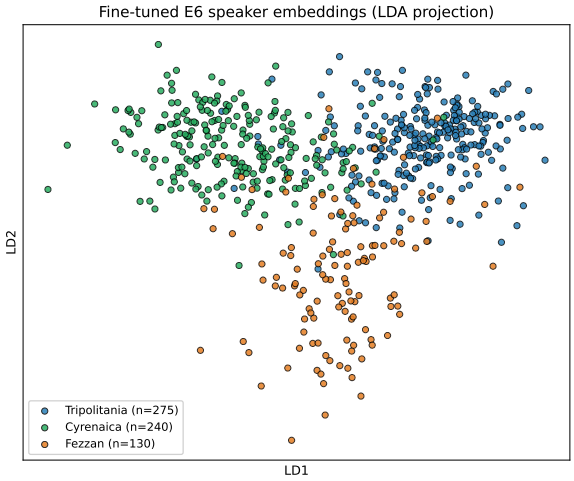


Figure 6: All 645 speaker centroids from the fine-tuned E6 embeddings, colored by region. The two LDA axes were fit on training segments only.

number of Fezzan channels may also contribute.

Figure 6 plots all 645 speaker centroids from the fine-tuned E6 model on two LDA axes. Cyrenaica and Tripolitania lie on opposite sides, and Fezzan lies between them, closer to the Tripolitania side. The axes were fit only on training segments and their region labels. The 365 held-out dev and test speakers, never used to fit the axes, still separate on them with a silhouette score of 0.29, so the pattern generalizes rather than being an artefact of fitting the axes. An unsupervised t-SNE of the same embeddings gives a silhouette of 0.05: the separation is a specific direction in the embedding space that the region-supervised LDA fit has to find, not an obvious clustering. The frozen encoders show this cue more weakly. Part of what any probe reads from these embeddings may be channel- or speaker-correlated rather than dialectal, which Section 5.7 tests.

5.6 Robustness to segment length

Accuracy changes little across the three duration bands. The ECAPA probe (E2) scores 58.3% on three-to-five-second segments and 55.7% on the longest, and the fusion (E4) behaves the same way (59.9% down to 56.1%). The fine-tuned model (E6) rises only slightly with length, from 74.5% on 3–5 s segments to 75.3% on 5–7 s and 76.9% on 7–10 s. Segment length is therefore not a main source of error for any of these models.

5.7 Robustness to channel overlap

The released splits are speaker-disjoint: each merged speaker sits in one split. They are not channel-disjoint. A channel (playlist) can place its speakers in different splits, so 6,135 of 7,977 test segments (77%) and 153 of 182 test speakers (84%) share a channel with training, which lets a classifier read channel or recording cues that happen to track region rather than dialect.

To bound this effect we built a second split that keeps whole channels together. Each of the 161 channels is assigned by its dominant region (the

	ECAPA (E2)		Fusion (E4)		LID-FT (E6)	
Region	**Spk**	**Chan**	**Spk**	**Chan**	**Spk**	**Chan**
Tripolitania	.581	.580	.593	.579	.786	.793
Cyrenaica	.634	.591	.645	.584	.789	.722
Fezzan	.361	.225	.363	.229	.532	.313
Macro	.525	.465	.534	.464	.702	.609

Table 8: Per-region and macro F1 (test, segment level) under the speaker-disjoint (Spk) and channel-disjoint (Chan) splits for ECAPA (E2), Fusion (E4), and LID-FT (E6).

region holding most of its duration), and within a region channels are placed largest-first into train, dev, or test to reach an 80/10/10 duration share. No channel crosses splits. By duration, TikTok accounts for about 28% of the test and dev splits, compared with about 7% of training, because the largest YouTube channels were packed into train first. We then re-fit the ECAPA probe (E2) on the same cached features and retrain the fine-tuned model (E6) from scratch on this split (end-to-end fine-tuning cannot reuse cached features), and evaluate both on the channel-disjoint test set, which has 48 channels by dominant-region counting (19 Tripolitania, 17 Cyrenaica, and 12 Fezzan), or 50 when each channel is counted for every region whose speakers it carries; Fezzan has 12 under either count.

Macro-F1 falls from 0.702 to 0.609 for the fine-tuned model when no test channel is seen in training (Table 8). The drop suggests that part of the speaker-disjoint result depends on channel-correlated cues, although the channel-disjoint split also changes the source mix. The two coastal regions move little while most of the drop falls on Fezzan, which is covered by the fewest channels, so its channel-disjoint estimate rests on few independent sources and is the noisiest. For E6, 95% channel-clustered bootstrap CIs, resampling whole channels, are 0.609 [.486, .677] for macro, .793 [.606, .893] for Tripolitania, .722 [.435, .836] for Cyrenaica, and .313 [.051, .455] for Fezzan.

The channel-disjoint Fezzan F1 of 0.31, with an interval reaching down to 0.05, shows how much room for improvement remains.

6 Discussion

The front-end's pretraining objective matters more than the classifier or the fusion: under the same linear probe, swapping the speaker-recognition ECAPA for a language-identification ECAPA is the single largest gain, and fine-tuning that encoder adds less. A speaker encoder is trained to separate voices; a language-ID encoder is trained to separate languages, which is closer to the dialect task. The second finding is that fine-tuning sharpens the data-rich coastal regions but does not improve Fezzan over the frozen language-ID probe (E5).

These regional results are consistent with earlier descriptions of Libyan Arabic. The two coastal varieties, Cyrenaica and Tripolitania, are the easiest for every model and have the most data and the best-described, most distinct varieties (Benkato, 2014; Pereira, 2010); southern Libyan sits in a contact zone with both coastal varieties and with languages to the south, so a soft Fezzan boundary is expected. The segment audit (Section 5, Table 3) points the same way: Fezzan test segments were judged MSA more than twice as often as coastal ones, and dialectal Fezzan segments were heard as Tripolitania as often as Fezzan (12 segments each). Register and accommodation, as well as genuine dialect closeness, could produce this pattern, and the benchmark cannot separate them. Limited Fezzan training data adds error on top of this pattern but does not explain its direction. The results support the three-region partition, but the Fezzan and channel-disjoint numbers show that the evidence is still limited.

7 Conclusion

We presented a sub-regional Libyan Arabic speech corpus—133.7 hours, 645 speakers, 161 channels, three regions, speaker-disjoint splits—built by an automated pipeline and human review that confirmed the provenance region for 95.5% of kept speakers (within-annotator consistency). An off-the-shelf Arabic model recognizes Libyan only 74% of the time and cannot resolve the region. On the new three-way task, the front-end's pretraining objective had the largest effect of any choice tested: a language-identification ECAPA reaches 0.702 macro-F1 speaker-disjoint and 0.609 channel-disjoint, both above a speaker-recognition ECAPA (0.525) and the 0.20 floor. Fezzan remains the hardest region: it has the least training data (16.5 hours) and the lowest F1, falling from 0.532 speaker-disjoint to 0.313 channel-disjoint. In the segment audit, 33.3% of Fezzan test segments were judged MSA. Closing this gap needs more dialectal southern speech from more channels.

Limitations

The regional labels are provenance-based: they start from the origin of the source channel, not from per-utterance perceptual annotation by multiple linguists. A guest from one region appearing on another region's channel is a source of label noise that our human review only partly corrects. The 95.5% rate is within-annotator consistency; the blind review (Section 3.3) gives the inter-annotator figure. That review was done by a co-author. It measures agreement between the two annotators; it does not measure anchoring, which we did not quantify. Validation is also speaker-level, not segment-level: the annotator made one keep-or-relabel decision per speaker after listening to a spread of that speaker's clips, but did not audit every segment individually.

Automated gates within each speaker—an embedding-cohesion audit and the music, voice-activity, and signal-to-noise filters—remove off-cluster and non-speech segments, but segments judged MSA can still pass under the speaker's label: 17.3% of test segments overall and 33.3% of Fezzan test segments (Table 3); code-switched and mixed-speaker segments were rare.

Several of the regions we label also contain Berber/Amazigh-speaking communities (for example the Nafusa Mountains and Zuwarah in the west, Awjila in the east, and the Tuareg area of the south (Benkato and Pereira, 2016; Encyclopædia Britannica, 2025)), whose speech is not Arabic. Our human review removes obvious non-Libyan and MSA speakers, but we apply no explicit Arabic-versus-Berber gate, so a small amount of Berber or Berber-influenced speech may remain under a regional Arabic label.

The corpus is broadcast and social-media speech and is skewed toward male speakers (83%), so results may not transfer to conversational or read speech, or to balanced-gender settings. Fezzan is the smallest class (20.6 hours; 165 episodes, compared with 686 for Tripolitania and 478 for Cyrenaica), and we found little more usable Fezzan material after established broadcast sources ran out. Given the MSA rate in the segment audit, the Fezzan data may not represent everyday southern speech.

All our encoders, including the fine-tuned language-identification ECAPA, encode speaker and channel as well as dialect, and even after fine-tuning the regions only partly separate in Figure 6. The channel-disjoint re-evaluation (Section 5.7) indicates that part of the signal is channel-correlated: the fine-tuned model's macro-F1 falls from 0.702 to 0.609 and Fezzan's F1 from 0.53 to 0.31 once no test channel is seen in training. With every method tested here, Fezzan recall stays below 0.45 speaker-disjoint and below 0.21 channel-disjoint; other imbalance-aware methods remain to be evaluated.

We did not test end-to-end fine-tuning of the self-supervised encoder or larger language-identification models. We also report no per-split genre breakdown, because genre is not in our metadata, and no diarization error rate, because we have no time-aligned reference.

Ethics Statement

The audio is public Libyan media collected for research. We assign each speaker an anonymous ID and attach no names. For TikTok records, the manifest keeps the public account handle of the publisher, and because most TikTok accounts contain a single speaker, this handle identifies the speaker directly. We release no other personal data about account holders or speakers. We do not redistribute the source audio; we release segment manifests, derived embeddings, and code, consistent with the source platforms' terms. The corpus is intended for research on Arabic dialect variation and speech processing. Regional dialect labels are sensitive: they describe recorded media, not claims about any individual's identity or origin.

References

Mohammad Al-Fetyani, Muhammad Al-Barham, Gheith Abandah, Adham Alsharkawi, and Maha Dawas. 2023. MASC: Massive Arabic speech corpus. In *IEEE Spoken Language Technology Workshop (SLT)*, pages 1006–1013.

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 *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1026–1033.

Ahmed Ali, Stephan Vogel, and Steve Renals. 2017. Speech recognition challenge in the wild: Arabic MGB-3. In *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 316–322.

Ibrahim Amin. 2025. HuBERT Arabic Spoken Dialect Classifier. https:

//huggingface.co/IbrahimAmin/hubert-arabic-spoken-dialect-classifier.

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. 2022. XLS-R: Self-supervised cross-lingual speech representation learning at scale. In *Proceedings of Interspeech 2022*, pages 2278–2282.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems*, volume 33, pages 12449–12460.

Adam Benkato. 2014. The Arabic dialect of Benghazi, Libya: Historical and comparative notes. *Zeitschrift für Arabische Linguistik*, 59:57–102.

Adam Benkato and Christophe Pereira. 2016. An annotated bibliography of Arabic and Berber in Libya. *Libyan Studies*, 47:149–165.

Hervé Bredin, Ruiqing Yin, Juan Manuel Coria, Gregory Gelly, Pavel Korshunov, Marvin Lavechin, Diego Fustes, Hadrien Titeux, Wassim Bouaziz, and Marie-Philippe Gill. 2020. pyannote.audio: Neural building blocks for speaker diarization. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7124–7128.

Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. 2020. ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification. In *Proceedings of Interspeech 2020*, pages 3830–3834.

Haroun Elleuch, Salima Mdhaffar, Yannick Estève, and Fethi Bougares. 2025. ADI-20: Arabic dialect identification dataset and models. In *Proceedings of Interspeech 2025*, pages 2775–2779.

Encyclopædia Britannica. 2025. Libya: People. https://www.britannica.com/place/Libya/People. Encyclopædia Britannica; accessed June 2026.

M. Essgaer, K. Massud, R. Al Mamlook, and N. Ghmaid. 2025. Computational linguistics meets Libyan dialect: A study on dialect identification. *International Journal of Intelligent Systems and Applications*, 17(6):118–130.

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.

Qiuqiang Kong, Yin Cao, Turab Iqbal, Yuxuan Wang, Wenwu Wang, and Mark D. Plumbley. 2020. PANNs: Large-scale pretrained audio neural networks for audio pattern recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2880–2894.

Houssam Eddine-Othman Lachemat, Akli Abbas, Nourredine Oukas, Yassine El Kheir, Samia Haboussi, and Shammur Absar Chowdhury. 2024. CAFE: A novel code-switching dataset for Algerian dialect, French, and English. ArXiv:2411.13424.

Hamdy Mubarak, Amir Hussein, Shammur Absar Chowdhury, and Ahmed Ali. 2021. QASR: QCRI Aljazeera speech resource – a large scale annotated Arabic speech corpus. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 2274–2285.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Christophe Pereira. 2010. *Le parler arabe de Tripoli (Libye)*, volume 4 of *Estudios de Dialectología Árabe*. Instituto de Estudios Islámicos y del Oriente Próximo, Zaragoza, Spain.

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. 2021. SpeechBrain: A general-purpose speech toolkit. ArXiv:2106.04624.

Suwon Shon, Ahmed Ali, Younes Samih, Hamdy Mubarak, and James Glass. 2020. ADI17: A fine-grained arabic dialect identification dataset. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8244–8248.

Silero Team. 2024. Silero VAD: Pre-trained enterprise-grade voice activity detector. https://github.com/snakers4/silero-vad.

David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. 2018. X-vectors: Robust DNN embeddings for speaker recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5329–5333.

Peter Sullivan, AbdelRahim Elmadany, and Muhammad Abdul-Mageed. 2023. On the robustness of arabic speech dialect identification. In *Proceedings of Interspeech 2023*, pages 5326–5330.

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. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21745–21758.

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. 2025. NADI 2025: The first multidialectal Arabic speech processing shared task. In *Proceedings of the Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 720–733.

Jörgen Valk and Tanel Alumäe. 2021. VoxLingua107: A dataset for spoken language recognition. In *IEEE Spoken Language Technology Workshop (SLT)*, pages 652–658.

A Speaker Counts by Validation Stage

Stage	Count
Candidate speaker identities above the evidence floor, all reviewed by the annotator	769
Identities kept after the annotator confirmed them as Libyan (76 excluded: 34 MSA, 28 poor audio, 13 non-Libyan, 1 mixed)	693
The same 693 confirmed speakers counted as playlist-level speaker clusters, before cross-playlist merging; the unit of Section 5.1 and Figure 4	799
Final speakers after the annotator manually merged identities across playlists	645

Table 9: Speaker counts at validation stages.

B Implementation Details

Pretrained checkpoints. Diarization: `pyannote/speaker-diarization-community-1`. Voice activity: Silero VAD (default thresholds). Music scoring: PANNs CNN14. Speaker embeddings: `speechbrain/spkrec-ecapa-voxceleb` (192-d). Language-ID embeddings and the E5/E6 encoder: `speechbrain/lang-id-voxlingua107-ecapa` (256-d). Self-supervised features: `facebook/wav2vec2-xls-r-300m`. Triage and E1 classifier: `IbrahimAmin/hubert-arabic-spoken-dialect-classifier`. All checkpoints come from the Hugging Face Hub except Silero and PANNs, which come from their GitHub releases.

Frozen-feature probes (E2–E5). All probes are scikit-learn (Pedregosa et al., 2011) logistic regression (L2, C=1, 5000 iterations) or linear SVM on L2-normalized features. The XLS-R feature is the masked mean of the frozen encoder's last hidden state, capped at 15 s (1024-d); the fusion concatenates the two L2-normalized vectors. Split seed 1337.

Fine-tuning (E6). E6 fine-tunes the ECAPA-TDNN encoder and a linear head (about 21M parameters) with AdamW (encoder 10^{-4}, head 10^{-3}), batch 16, bf16, linear warmup then decay, and early stopping on dev macro-F1; SpecAugment masks time and frequency on the Fbank features in training only. Split seed 1337.

Significance testing. All confidence intervals are percentile bootstraps over the test segments (1,000 resamples, seed 1337). The channel-disjoint intervals instead resample whole channels (1,000 resamples, seed 1337). Because segments from one speaker are correlated, this i.i.d. segment resampling makes the intervals mildly optimistic; CI width is driven mainly by the test-set size, so models with very different macro-F1 can have similar widths. Paired comparisons add McNemar's exact test on segment correctness and a paired bootstrap of the macro-F1 difference, run both i.i.d. over segments and clustered by speaker (resampling whole speakers) to account for that correlation.

SALM: Social Arabic Language Model

Saad Mankarious* **Aya Zirikly**
The George Washington University
{saadm, ayah.zirikly}@gwu.edu

Abstract

Arabic NLP faces a fundamental mismatch between training and deployment conditions. While most language models are trained predominantly on Modern Standard Arabic (MSA), real-world communication occurs largely in spoken dialects. To bridge this gap, we leverage social media content from Reddit, Twitter, and YouTube as a scalable proxy for spoken Arabic. We construct a 1.04-billion-token pretraining corpus and train SALM, a compact 336M-parameter decoder-only language model pre-trained via next-token prediction alone without masked language modelling, dialect labels, or task-specific signal. Despite an order-of-magnitude smaller training corpus than existing dialectal PLMs and a generative training objective that provides no direct classification signal, SALM outperforms MARBERT, explicitly pre-trained on one billion dialectal tweets, on MADAR-26 dialect identification by 4.2 accuracy points and 4.8 macro-F1 points, and leads all seven evaluated baselines outright on both MADAR-26 and MADAR-9. Fine-tuning SALM end-to-end further raises its own MADAR-26 accuracy from 44.9% to 57.0% (macro-F1 44.3% $\rightarrow$ 57.2%). Under the same frozen probe on binary Arabic sentiment (AJGT), SALM reaches 92.2% accuracy, the highest among evaluated Arabic PLMs, encoder and decoder alike, under a single-seed evaluation. We release the corpus and trained model to facilitate future research on Arabic spoken language modelling.

1 Introduction

Arabic is spoken by approximately 400 million native speakers across 22 countries, making it the fifth most widely spoken language in the world. This geographic breadth produces substantial linguistic diversity: the Arabic-speaking world spans from Morocco in the west to Oman in the east, and each region has developed its own spoken dialect. Western varieties such as Moroccan Darija blend Classical Arabic with Amazigh substrate vocabulary and French loanwords; Egyptian Arabic, the most widely understood supra-regional variety, serves as an informal spoken lingua franca; Gulf dialects preserve Bedouin phonological patterns; and Iraqi Arabic retains morphological features largely absent from MSA. Despite this diversity, the prestige written form, Modern Standard Arabic (MSA), is a single codified variety taught in schools and used across formal media throughout the Arab world.

Current NLP systems are poorly equipped for this diversity for two reasons. First, the data used to train large language models skews heavily toward written formal text: Wikipedia articles, news corpora, and digitised books are overwhelmingly MSA, systematically underrepresenting the spoken varieties that characterise everyday communication. Second, dialects are not simply noisy variants of MSA: they differ in morphology, phonology, lexicon, and syntax to a degree that can produce mutual unintelligibility, making an MSA-trained model a weak prior for dialectal text (Zaidan and Callison-Burch, 2014).

Social media platforms, Twitter, Reddit, and YouTube, offer a scalable and naturalistic solution. Hundreds of millions of Arabic speakers communicate daily in their native dialects on these platforms, generating a continuous stream of authentic dialectal text. We leverage this observation to construct SALM (Social Arabic Language Model), a 336M-parameter decoder-only causal language model pre-trained on a 1.04-billion-token corpus of Arabic social media text. Unlike encoder-based Arabic PLMs, SALM is trained solely with a causal language-modelling objective and uses no dialect supervision or task-specific pre-training signal. Its 1.04-billion-token corpus is an order of magnitude smaller in token count than the one-

*Corresponding Author

Proceedings of The Fourth Arabic Natural Language Processing Conference (ArabicNLP 2026), pages 497–507
October 28-29, 2026 ©2026 Association for Computational Linguistics

billion-tweet training set of MARBERT (Abdul-Mageed et al., 2021), and the entire training run fits on two consumer-class GPUs. Under a frozen linear-probe protocol, SALM outperforms MARBERT and every other evaluated baseline on the two most fine-grained MADAR granularities.

We summarize our contributions as follows:

- A 1.04-billion-token corpus of dialectal Arabic text curated and cleaned from Twitter, Reddit, and YouTube.
- SALM, a compact 336M-parameter decoder-only causal language model trained on this corpus via next-token prediction alone, with the pre-trained checkpoint publicly released.
- A frozen linear-probe and full fine-tuning evaluation on dialect identification (MADAR) and sentiment analysis (AJGT), demonstrating that SALM outperforms purpose-built dialectal encoders on fine-grained dialect identification despite a training corpus an order of magnitude smaller and without any classification signal.

2 Related Work

MSA-focused Arabic PLMs. Arabic NLP has long been dominated by models targeting Modern Standard Arabic. AraBERT (Antoun et al., 2020) extends masked language modelling to Arabic by pre-training on large MSA corpora drawn from news and Wikipedia, and ArBERT (Abdul-Mageed et al., 2021) follows a similar approach. These models achieve strong results on MSA benchmarks but do not directly address the dialectal diversity of spoken Arabic.

Dialect-aware Arabic PLMs. MARBERT (Abdul-Mageed et al., 2021) is pre-trained on one billion Arabic tweets and is the current state-of-the-art representation model for Arabic dialect tasks. CAMeLBERT-DA (Inoue et al., 2021) is trained specifically on dialectal Arabic text and achieves competitive results on dialect identification. Both models are encoder-only (bidirectional BERT-style), which confers a structural advantage in classification tasks relative to decoder-only architectures. Crucially, both adopt a masked-language-modelling pretraining objective specifically designed to produce classification-ready representations.

Arabic generative models. AraGPT2 (Antoun et al., 2021) adapts GPT-2 (Radford et al., 2019) to Arabic using MSA text, demonstrating that decoder-only pre-training is viable for Arabic but limited to the formal written register. More recent large-scale efforts apply instruction-tuned GPT-style pre-training at the multi-billion-parameter scale, again with an MSA focus.

Dialectal language models. The closest work to ours in spirit is AlCLaM (Aldarmaki et al., 2024), a BERT-based encoder pre-trained from scratch on dialectal Arabic text collected from social media and evaluated on downstream discriminative tasks.[1] Unlike AlCLaM, SALM is a decoder-only causal language model: rather than a masked-language-modelling objective tailored to classification, SALM is trained solely via next-token prediction and is evaluated on downstream classification benchmarks, dialect identification and sentiment analysis, using both frozen and fine-tuned representations.

3 Dataset

We curated pretraining data from three social media platforms: Reddit, Twitter, and YouTube. All collected data is publicly available and was obtained using standard libraries and APIs in compliance with each platform's terms of service. Table 1 reports corpus statistics; token counts are exact figures derived from the tokenised `.npy` shards rather than character-based estimates.

Corpus	Documents	Tokens	Share
Twitter	202,700,438	730M	70.5%
Reddit	4,670,468	240M	23.2%
YouTube	25,100	66M	6.3%
Total	**207M+**	**1.04B**	100%

Table 1: Pretraining corpus statistics (exact BPE token counts from tokenised shards; Twitter document count is the full `pain/Arabic-Tweets` dataset, tokens after quality filtering).

Reddit. We collect from 57 Arabic-speaking subreddits spanning the Arab world, with the largest communities centred on Egypt (*r/CAIRO*, *r/Egypt*, *r/ExEgypt*) and Saudi Arabia (*r/saudiarabia*, *r/SaudiForSaudis*). The corpus comprises 4,670,468 documents after cleaning (24.7% of 18.9M raw rows), with a median

[1] AlCLaM reports 3.4M training sentences (no token count); our 207M+ raw documents (Table 1) are roughly sixty times more.

length of 129 characters. 88% of documents are comments rather than top-level posts, reflecting the conversational register of the data. Posts span from January 2009 to June 2026, providing temporal diversity.

Twitter. We use the `pain/Arabic-Tweets` dataset from HuggingFace, containing 202.7M Arabic tweets. Tweets are short (median 63 characters, 12 words) and uniformly high-quality in terms of Arabic script content (Arabic character ratio = 1.0 after cleaning). Despite the large raw document count, the tokenised corpus yields 730M tokens after quality filtering, reflecting the filtering of very short or non-dialectal tweets.

YouTube. We collect transcripts of Arabic-language YouTube videos, primarily movies, series, and podcasts, yielding 25,100 unique videos after exact- and near-duplicate removal. Transcripts are substantially longer than social media posts (median 3,416 characters, 684 words), providing the model with extended discourse context absent from Twitter and Reddit.

Appendix B (Figure 2) gives paraphrased examples of raw text from each platform, illustrating their register and dialect mix.

Dialect coverage. We characterise dialect distribution across all three corpora using `camelbert`[2], a BERT model fine-tuned on the NADI 2020 country-level Arabic dialect identification shared task (Abdul-Mageed et al., 2020), which classifies documents into 21 Arabic-speaking countries with no reject/unassigned option. We draw a random sample of 8,000 documents per corpus (fixed seed 42); YouTube transcripts, being substantially longer than tweets or Reddit posts, are first windowed into ∼40-word, tweet-length chunks so the classifier sees comparable-length input across all three corpora. We collapse the 21 country-level predictions into seven broad dialect regions used elsewhere in this paper (Egyptian, Levantine, Gulf, Iraqi, Maghrebi, Sudanese, Yemeni) plus an Other bucket for the small number of countries with no clean regional match. Table 2 reports the resulting region shares; every sampled document receives a regional assignment by construction. Egyptian Arabic (EGY) is the dominant variety on Reddit and YouTube (45.8% and 55.7% respectively), reflecting Egypt's large online presence and media output. Gulf Arabic (GLF) instead dominates on Twitter (48.6%), with Egyptian second (20.0%), consistent with high social media engagement across the Gulf region. Egyptian and Gulf content jointly account for 72–80% of sampled documents on every platform; Levantine, Maghrebi, and Iraqi Arabic are each present at low-single-digit-to-low-teen shares, while Sudanese and Yemeni Arabic remain under 5% on every platform (see Limitations).

Region	Reddit	YouTube	Twitter
EGY (Egyptian)	45.8%	55.7%	20.0%
GLF (Gulf)	26.7%	24.6%	48.6%
LEV (Levantine)	11.2%	6.8%	5.5%
MAG (Maghrebi)	7.0%	3.3%	10.3%
IRQ (Iraqi)	4.9%	4.5%	10.6%
YEM (Yemeni)	3.9%	4.8%	3.9%
SUD (Sudanese)	0.4%	0.2%	0.4%
Other	0.2%	0.1%	0.8%

Table 2: Dialect region share, 8,000-document random sample per corpus (seed = 42), collapsed from 21 `camelbert-mix-did-nadi` countries to 7 regions (0% unassigned).

3.1 Preprocessing

We conduct heavy preprocessing of the social media text prior to tokenisation. Social media text is characteristically noisy, exhibiting code-switching, non-Arabic script, URLs, diacritics, and punctuation-heavy formatting. Inspired by (El-Haj et al., 2024), we develop a multi-stage filter applied uniformly across all three sources: (1) removal of Latin characters and non-Arabic punctuation,[3] (2) stripping of diacritics (tashkeel), (3) a minimum character-length gate (20 characters post-cleaning), (4) an Arabic content gate (at least 40% of non-whitespace characters must be Arabic script), and (5) exact- and near-duplicate removal for YouTube transcripts (MinHash, Jaccard threshold = 0.8). For Reddit, an additional 11.9M documents (62.9% of raw rows) were removed as punctuation-only or near-empty entries. The final output per document is a single cleaned line containing the social media post with no punctuation, diacritics, or non-Arabic tokens. Appendix A provides detailed per-platform cleaning statistics.

[2]CAMeL-Lab/bert-base-arabic-camelbert-mix-did-nadi

[3]This step also removes Arabizi (Arabic transliterated into Latin script) and code-switched Latin-script segments, since the pipeline has no transliteration component to map them back to Arabic script. We made this deliberate choice for simplicity and acknowledge its limitations (see Limitations).

4 SALM Model

4.1 Model Architecture

SALM follows the GPT-2 decoder-only Transformer architecture (Radford et al., 2019). Each block applies pre-layer normalisation, causal multi-head self-attention computed via scaled dot-product attention, and a position-wise feed-forward network with GELU activation (tanh approximation). A single layer normalisation is applied before the output projection. Token embedding and output projection weights are tied to reduce parameter count.

SALM has 24 transformer layers, 16 attention heads, and a model dimension $d_{\text{model}} = 1024$, totalling 336M parameters. The model uses a context length of 512 tokens and a vocabulary of 32,000 BPE subword units (Section 4.2).

4.2 Tokenizer

Arabic NLP systems typically rely on tokenizers trained on MSA-heavy corpora, which undersegment dialectal words absent from their training distribution. We train a custom BPE tokenizer on a subset of the pretraining corpus (Reddit and YouTube), ensuring the vocabulary reflects the dialectal and social media register of the data.

Design. We use a Whitespace pre-tokenizer rather than the ByteLevel encoding common in multilingual models. ByteLevel maps each byte to a printable ASCII character, distorting Arabic codepoints and producing unintuitive subword boundaries; Whitespace splitting instead preserves Arabic character identity and aligns splits with dialectal morphological units. No normaliser is applied beyond what has already been performed in preprocessing (Section 3.1); further normalisation risks collapsing dialectal character variants that carry morphological information. Subword continuation pieces are marked with a ## prefix, and the decoder uses a WordPiece-style strategy (rather than the ByteLevel BPE decoder) to correctly reinsert spaces on decode.

Vocabulary. The vocabulary comprises 32,000 BPE types trained with minimum merge frequency 2, plus five special tokens (<pad>, <unk>, <bos>, <eos>, <mask>). Of the 32,000 types, 31,555 (98.6%) contain at least one Arabic character; only 5 types contain Latin characters, reflecting the Arabic-script-only training data. Zero diacritic-bearing types appear in the vocabulary, consistent with diacritic removal in preprocessing. Subword pieces (marked ##) account for 5,464 types (17.1%); the majority of vocabulary entries are therefore whole words, appropriate for the short utterances that dominate social media text.

Fertility. Across Egyptian, Levantine, and Iraqi samples, fertility is approximately 1.0–1.1 tokens per whitespace-delimited word: common dialectal words such as *mish* (not), *wallah* (by God), and *btihki* (you speak) each map to a single token. Gulf and Maghrebi varieties show marginally higher fertility due to a small number of characters absent from the training data, which are mapped to <unk>. Latin-script text (Arabizi, code-switched English) maps entirely to <unk> by design, as the pretraining corpus is Arabic-script only. Encode–decode round-trips are lossless for the three major varieties tested; any mismatch in Gulf or Maghrebi samples is attributable to a single <unk> substitution per sentence with no subword leakage.

4.3 Training Setup

SALM is trained on two L40S GPUs (48 GB each) with an effective batch size of 131,072 tokens per gradient step, achieved via micro-batching with gradient accumulation (micro-batch $B = 8$, sequence length $T = 512$, accumulation steps $= 16$ across 2 GPUs). The optimiser is fused AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.95$, $\varepsilon = 10^{-8}$, weight decay $= 0.1$) with gradient norm clipping at 1.0. The learning rate follows a linear warmup over 315 steps (approximately 3% of the planned training schedule) followed by cosine annealing to a minimum of 6×10^{-5}; the peak learning rate is 6×10^{-4}. Training runs in bfloat16 mixed precision with `float32_matmul_precision='high'`.

SALM is evaluated at the checkpoint corresponding to step 18,950 (2.48B training tokens consumed). Validation cross-entropy loss at this checkpoint is 4.36 on a held-out 10% split of the pretraining corpus (≈100M tokens), and the training curve shows no plateau at this point, indicating the model would benefit from continued training.

5 Experiments

5.1 Evaluation Protocol

We evaluate SALM using two protocols. The primary evaluation is a **frozen linear probe**: the backbone is held fixed and a single linear classification layer is trained on mean-pooled, L2-normalised sentence embeddings extracted from

the frozen model. A frozen probe directly tests whether task-relevant structure is linearly decodable from pretrained representations (Alain and Bengio, 2017), without allowing the backbone to restructure in response to task labels. It also places encoder and decoder architectures on equal footing: bidirectional pooling confers an advantage under fine-tuning, but both reduce to a fixed embedding function under the probe, making performance differences attributable to pretraining corpus effects rather than architectural inductive biases.

The probe is trained with class-weighted cross-entropy (AdamW, lr = 1e-2, weight_decay = 1e-4) for up to 20 epochs, selecting the epoch with the best accuracy on a held-out validation split. The train/validation/test split used for each task is specified in the corresponding subsection below.

5.2 Dialect Identification

We evaluate on four granularities derived from the MADAR parallel corpus (Bouamor et al., 2019) (12,000 BTEC sentences, 26 city-dialect varieties): **MADAR-2** (binary, MSA vs. dialect), **MADAR-6** (six broad regions: EGY, LEV, GLF, IRQ, MAG, MSA), **MADAR-9** (nine regions: Nile Egypt, Nile Sudan, Levant, Iraq, Gulf, Yemen, Libya, Maghreb, MSA), and **MADAR-26** (26 city-level varieties). The parallel corpus is unstacked so that each non-empty (sentence, city) cell becomes one labelled example, yielding 112,000 examples total. Because MADAR is a parallel corpus, the same underlying BTEC sentence is translated into all 26 city dialects; we therefore partition examples using MADAR's official per-sentence train/dev/test assignment (95,600 / 11,200 / 5,200 examples) rather than a fresh per-example split, so that different-dialect translations of the same sentence cannot cross the train/test boundary.[4] Model selection (epoch/checkpoint choice, Section 5.1) uses the dev split; the held-out test split is scored exactly once per model, with the selected weights.

We compare SALM against six Arabic and multilingual PLM baselines. Majority-class baselines on the test split are: MADAR-2 96.2%, MADAR-6 26.9%, MADAR-9 23.1%, MADAR-26 3.8%.

5.3 Sentiment Analysis

We evaluate on the AJGT (Arabic Jordanian General Tweets) dataset (Alomari et al., 2017), a binary sentiment corpus of 1,800 tweets (positive or negative) with a balanced class distribution of 900 examples per class. AJGT has no official train/test partition, so, unlike MADAR (Section 5.2), we use the same frozen probe, embedding extraction, hyperparameters, and a stratified 80/20 train/test split (seed = 42) as described in Section 5.1.[5]

5.4 Full Fine-tuning

To assess SALM's representational capacity beyond what a frozen linear probe reveals, we additionally fine-tune SALM end-to-end on each MADAR variant. A linear classification head is placed on the mean-pooled final hidden states; all backbone weights receive gradients. We use AdamW with a backbone learning rate of 2×10^{-5} and a head learning rate of 1×10^{-3}, cosine-decayed with 10% linear warmup over the first epoch. Training runs for up to 10 epochs with early stopping on validation accuracy (patience = 3), using the same official MADAR train/dev/test split as Section 5.2 throughout, so results are directly comparable to the frozen-probe evaluation.

5.5 Training Evaluation

We monitor training progress by evaluating validation loss every 475 steps on a held-out 10% split of the pretraining corpus (approximately 100M tokens). Checkpoints are saved at the same cadence, enabling post-hoc selection of the best checkpoint for downstream evaluation. SALM's training loss declines from 10.6 at initialisation to 4.6 at step 18,950 with no observed plateau, indicating the model is likely to benefit from continued training.

6 Results

6.1 Dialect Identification

Table 3 reports accuracy and macro-F1 across all four MADAR variants for the frozen linear probe.

SALM leads on fine-grained dialect identification. SALM achieves 44.9% accuracy and 44.3% macro-F1 on MADAR-26, the best score of any evaluated model on both metrics, ahead of the next-best accuracy (CAMeL, 42.3%) and

[4] The test split fixes 200 sentences/city (200 × 26 = 5,200), so it is not class-balanced across the coarser MADAR-2/6/9 label sets, affecting their majority baselines below.

[5] AJGT may overlap with the `pain/Arabic-Tweets` pretraining data, though this is unlikely given AJGT's narrow construction (a small, curated set of Jordanian tweets).

Model	Accuracy (%)				Macro-F1 (%)			
	M-2	M-6	M-9	M-26	M-2	M-6	M-9	M-26
Multilingual PLMs								
mBERT	88.4	50.8	45.2	31.0	65.4	49.5	41.0	29.7
LaBSE	87.1	44.9	39.1	23.4	63.5	43.4	34.2	22.0
MSA-focused Arabic PLMs								
AraBERT	94.2	64.6	59.5	37.8	76.6	64.0	54.9	36.9
ArBERT	91.8	64.3	59.8	38.5	71.3	62.6	54.3	37.5
MSA+Dialect Arabic PLMs								
MARBERT	92.0	**70.9**	63.9	40.7	71.5	**68.3**	58.3	39.5
CAMeL	**94.5**	68.0	63.2	42.3	**76.9**	66.7	57.5	41.4
This work (frozen linear probe)								
SALM	93.0	69.5	**65.1**	**44.9**	73.5	67.5	**59.5**	**44.3**

Table 3: Dialect identification via frozen linear probe, scored once on the official held-out test split (Section 5.2). M-2/6/9/26 = MADAR granularity. **Bold** = best per column.

the next-best macro-F1 (MARBERT, 39.5%), despite never being trained on any classification signal. SALM also leads MADAR-9 (65.1% accuracy, 59.5% macro-F1), narrowly ahead of MARBERT (63.9%, 58.3%). At the coarser MADAR-2 and MADAR-6 granularities SALM is competitive but not best: CAMeL leads MADAR-2[6] (94.5% accuracy, 76.9% macro-F1) and MARBERT leads MADAR-6 (70.9% accuracy, 68.3% macro-F1).

Arabic-specific pre-training dominates over multilingual scale. LaBSE and mBERT are the weakest baselines (23.4% and 31.0% accuracy at MADAR-26), confirming that Arabic-specific pretraining contributes far more to dialectal representation quality than multilingual scale alone. The gap between mBERT and SALM at MADAR-26 is 13.9pp; against LaBSE it is 21.5pp.

6.2 Sentiment Analysis

Table 4 reports accuracy and macro-F1 on AJGT. The dataset is perfectly balanced (900 per class), so the majority baseline is 50.0% and accuracy and macro-F1 are numerically near-identical throughout.

SALM achieves the highest sentiment accuracy among evaluated PLMs. SALM achieves 92.2%, the highest accuracy of any evaluated model, ahead of the strongest encoders ArBERT (91.9%) and CAMeLBERT-DA (91.7%). The largest gap is between SALM and MARBERT (87.5%, −4.7pp), a model pre-trained on one billion dialectal tweets with a classification-oriented MLM objective and training data an order of magnitude larger. This result is notable for a causal decoder-only model: SALM uses left-to-right attention with a purely generative training objective, yet attains the top accuracy among the bidirectional encoder baselines evaluated here. These results reflect a single train/test split and a single random seed; we have not established their significance across seeds, nor verified the absence of overlap between AJGT and the pretraining corpus (Limitations).

Model	Acc (%)	Macro-F1 (%)
Multilingual PLMs		
mBERT	76.4	76.3
LaBSE	88.1	88.0
MSA-focused Arabic PLMs		
AraBERT	86.9	86.9
ArBERT	91.9	91.9
MSA+Dialect Arabic PLMs		
MARBERT	87.5	87.5
CAMeLBERT-DA	91.7	91.7
This work (frozen linear probe)		
SALM	**92.2**	**92.2**

Table 4: AJGT sentiment analysis via frozen linear probe, single stratified 80/20 split and seed (Section 5.3). Majority baseline: 50.0%. **Bold** = best overall.

6.3 Fine-tuning Results

Table 5 contrasts full fine-tuning with the frozen probe for SALM. Fine-tuning yields consistent and substantial gains over SALM's own frozen probe across all MADAR granularities: accuracy improves by +4.3pp on MADAR-2, +11.6pp on MADAR-6, +13.9pp on MADAR-9, and +12.1pp on MADAR-26, with macro-F1 gains of a similar magnitude (+11.6 to +15.2pp). Gains are smallest on the coarse MSA-vs-dialect dis-

[6] Every model falls below the 96.2% MADAR-2 majority baseline: the class-weighted probe trades raw accuracy for minority (MSA) recall, so macro-F1 (≈49% for the majority baseline vs. 63.5–76.9% here) is the more informative metric.

tinction (MADAR-2) and largest at the regional and city-level granularities, consistent with the frozen probe being most constrained where a linear head cannot compensate for causal mean-pooling's weaker cross-token aggregation relative to a bidirectional encoder's pooled representation.

We report MARBERT's and CAMeL's frozen-probe scores in Table 5 for context only. Because we do not fine-tune these baselines under an identical protocol (same optimiser schedule, dev-based early stopping, single test evaluation), Table 5 supports a claim about SALM's own gain from fine-tuning, not a comparison against fine-tuned baselines.

7 Discussion

Small corpus, generative objective, competitive performance. SALM's 1.04B-token corpus is an order of magnitude smaller than the one billion tweets used to pretrain MARBERT, an encoder-only model with a masked-language-modelling objective specifically designed to produce classification-ready representations; SALM instead uses only causal next-token prediction and no dialect supervision. Despite this, SALM outperforms MARBERT on MADAR-26 accuracy (44.9 vs. 40.7, +4.2pp) and macro-F1 (44.3 vs. 39.5, +4.8pp), and achieves the highest accuracy among evaluated PLMs on AJGT sentiment. This suggests that corpus quality and register relevance matter more than corpus scale or pretraining objective when the data adequately covers the target linguistic domain, and confirms that social media content is a high-quality, scalable proxy for spoken Arabic.

Fine-tuning further closes the architectural pooling gap. Even though SALM already leads the frozen-probe comparison at MADAR-9 and MADAR-26 (Section 5.2), full fine-tuning still yields large further gains over its own frozen-probe score: MADAR-26 accuracy rises from 44.9% to 57.0% (+12.1pp), and macro-F1 from 44.3% to 57.2% (+12.9pp). This is consistent with mean-pooled causal embeddings being a structurally weaker classification-ready representation than a bidirectional encoder's pooled embedding: even when SALM's frozen representations already contain enough dialectal signal to lead the frozen-probe comparison, allowing the backbone to adapt end-to-end still recovers substantial additional accuracy. We do not have a matched fine-tuned baseline under an identical protocol to test whether an encoder would show a comparably sized gain (Section 5.4).

The corpus dialect distribution shapes representation quality. The dialect distribution in Table 2 shows that Maghrebi Arabic constitutes 7.0% of the Reddit sample and 3.3% of the YouTube sample, and Sudanese Arabic under 0.4% of every platform. MADAR-26 includes several Maghrebi city labels (Tunis, Rabat, Sfax, etc.), and macro-F1 penalises underperformance on these and other minority classes. SALM's macro-F1 lead over MARBERT at MADAR-26 (44.3 vs. 39.5, +4.8pp) suggests that the multi-platform corpus composition provides useful dialect coverage for minority-class recall, despite the Egyptian/Gulf skew documented in Table 2.

Qualitative generation. Figure 1 shows representative completions from SALM across four dialectal varieties (full continuations in Appendix C, Figure 3), each preserving variety-specific lexical items absent from MSA: Egyptian يا ريت/احنا/ده, Levantine سخرلي/مافي احد, Iraqi ماكو/اكو (the most distinctive Iraqi marker), and Maghrebi والو/كاين (the latter borrowed from Amazigh, exclusive to Maghrebi Arabic). This qualitative evidence complements the linear probe results: the dialectal knowledge the probe captures reflects genuine structure encoded during pretraining, not an artefact of probe training.

Sentiment representations generalise from dialectal pretraining. The AJGT result (Section 5.3) extends the dialect identification finding: social media pretraining on a compact corpus induces representations broadly useful for Arabic sociolinguistic tasks, not only for dialect distinctions.

8 Conclusion

We introduced SALM, a compact 336M-parameter decoder-only language model pre-trained on a 1.04-billion-token corpus of dialectal Arabic social media text from Twitter, Reddit, and YouTube, using only a causal next-token-prediction objective, no masked language modelling, no dialect labels, no task-specific signal. Despite a training corpus an order of magnitude smaller than existing dialectal Arabic PLMs, SALM outperforms MARBERT and every other evaluated baseline on MADAR-9 and MADAR-26 dialect identification under a frozen linear probe, by 4.2 accuracy points and

Model	Accuracy (%)				Macro-F1 (%)			
	M-2	M-6	M-9	M-26	M-2	M-6	M-9	M-26
$\text{SALM}_{\text{probe}}$	93.0	69.5	65.1	44.9	73.5	67.5	59.5	44.3
$\text{SALM}_{\text{finetune}}$	**97.3**	**81.1**	**79.0**	**57.0**	**85.2**	**79.1**	**74.7**	**57.2**
$\text{MARBERT}_{\text{probe}}$	92.0	70.9	63.9	40.7	71.5	68.3	58.3	39.5
$\text{CAMeL}_{\text{probe}}$	94.5	68.0	63.2	42.3	76.9	66.7	57.5	41.4

Table 5: Full fine-tuning vs. frozen probe on MADAR for SALM. **Bold** = best SALM row per column. MARBERT/CAMeL frozen-probe scores are shown for context only, not fine-tuned comparison (Section 5.4).

Egyptian

في ناس بتقول ان مصر كانت دوله علمانيه بس احنا بنعتبر ده كلام فارغ لان عندنا قانون

*"**Some people say Egypt** was a secular state, but we think that's empty talk because we have a law..."*

Levantine

والله هاد الشي صار معي امبارح الساعه بالليل بس مافي احد بالبيت يارب ارزقني

*"**By God, this thing happened to me yesterday**, at night but no one was home. Lord, provide for me..."*

Iraqi

صديقي شكو ماكو كلشي تمام وين اكو احد يدزلي مسج بالواتساب حت لو من رقم غريب

*"**My friend, what's up, everything's fine** — where is there anyone who'd send me a WhatsApp message, even from an unknown number?"*

Maghrebi

واش راك كيفاش حالك الله يستر وين كاين والو انت مش محتاج حد يحضنك

*"**How are you, how are you doing, may God protect you.** Where is there anything? You don't need anyone to hold you..."*

Figure 1: SALM completions across four dialectal varieties (temperature = 0.85, top-k = 50, repetition penalty = 1.2). **Bold** = prompt; plain text = model continuation. Excerpted from the full prompts and continuations in Appendix C (Figure 3).

4.8 macro-F1 points at MADAR-26. Fine-tuning SALM end-to-end raises its own MADAR-26 accuracy to 57.0% (macro-F1 57.2%), a further gain of 12.1 accuracy points over its frozen probe. Under the same frozen probe on binary Arabic sentiment (AJGT), SALM achieves 92.2% accuracy, the highest among evaluated Arabic PLMs.

These results establish three conclusions. First, social media content is a high-quality and scalable training signal for dialectal Arabic representation learning even at modest corpus scale, with the induced representations generalising to sentiment analysis without any task-specific adaptation. Second, a decoder-only generative model pre-trained on social media can already match or exceed, under a frozen probe, encoder models purpose-built for classification, and fine-tuning yields substantial further gains on top of this. Third, even where SALM's frozen-probe representations already lead the comparison, fine-tuning still recovers a large further improvement, consistent with mean-pooled causal embeddings being a structurally weaker classification-ready representation than a bidirectional encoder's pooled embedding.

We release the pre-trained SALM model and evaluation code to support future work on Arabic dialect representation, generation, and zero-shot transfer.

Limitations

Architectural mismatch for classification. SALM is a causal (decoder-only) language model; representations are extracted by mean-pooling over the causal context, where each token attends only to preceding tokens. This is structurally weaker than bidirectional BERT-style pooling for classification tasks, so the frozen-probe results likely underestimate SALM's representational capacity even where SALM already leads the frozen-probe comparison (Section 5.2), as the fine-tuning results in Section 5.4 demonstrate.

Corpus geographic bias. Egyptian Arabic dominates Reddit (45.8%) and YouTube (55.7%); Gulf Arabic dominates Twitter (48.6%). Egyptian and Gulf content jointly account for 72–80% of sampled documents on every platform (Table 2), while Sudanese Arabic represents under 0.4% and Yemeni Arabic under 5% of every corpus, a significant underrepresentation relative to their speaker populations in the Arab world. This skew may affect SALM's representations of Sudanese and Yemeni dialects, both of which have dedicated labels in MADAR-9 and MADAR-26.

Sentiment evaluation robustness. The AJGT sentiment results (Section 5.3) reflect a single train/test split and a single random seed; we have not run a significance test (e.g. across multiple seeds, or a paired test such as McNemar's) to es-

tablish whether the reported margins over baselines are reliable rather than an artefact of this particular split. We have also not verified the absence of exact- or near-duplicate overlap between AJGT and the pretraining corpus. Because SALM's pretraining objective is next-token prediction and never sees AJGT's sentiment labels, such an overlap could at most make SALM's representations unusually well-tuned to the surface form of specific tweets, not leak label information the way a supervised train/test overlap would, but this has not been empirically confirmed. Both checks are needed before drawing strong conclusions from the sentiment results.

Arabizi and code-switched text. Our preprocessing pipeline (Section 3.1) removes non-Arabic-script text, which excludes Arabizi (Arabic transliterated into Latin script) and code-switched segments common in real social media use. We have not measured what fraction of the raw scraped data this affects, so the corpus and the resulting model should be understood as covering Arabic-script dialectal text specifically, not the full range of scripts in which Arabic speakers communicate online.

Evaluation scope. This paper focuses on dialect identification and binary sentiment analysis. SALM has not been evaluated on MSA NLU tasks (NER, question answering), text generation quality (perplexity, human evaluation), or cross-lingual transfer. The sentiment evaluation uses dialectal Twitter data (AJGT) and does not cover MSA sentiment benchmarks. Whether social media pretraining benefits or harms MSA task performance remains an open question for future 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.

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*, pages 97–110.

Guillaume Alain and Yoshua Bengio. 2017. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1703.01519*.

Hanan Aldarmaki, Hajar Alharthi, Maryam Alowaifeer, and Abdulla Almansoori. 2024. AlCLaM: Arabic colloquial language model. In *Proceedings of the Second Arabic Natural Language Processing Conference*.

Khaled Mohammad Alomari, Hatem M. ElSherif, and Khaled Shaalan. 2017. Arabic tweets sentimental analysis using machine learning. In *IFIP International Conference on Computer Science and its Applications*, pages 66–77. 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.

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.

Houda Bouamor, Nizar Habash, Mohammad Salameh, Wajdi Zaghouani, Owen Rambow, Hassan Abdulmageed, Salam Khalifa, Khalid Shaalan, Hend Refian, Bilal Ghazal, and 1 others. 2019. MADAR Arabic dialect corpus and lexicon. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3387–3396.

Mahmoud El-Haj and 1 others. 2024. Clusterlab: 101 billion Arabic words dataset. In *Proceedings of the Language Resources and Evaluation Conference*. Available at `https://huggingface.co/datasets/ClusterlabAi/101_billion_arabic_words_dataset`.

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.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

Omar F. Zaidan and Chris Callison-Burch. 2014. Arabic dialect identification. *Computational Linguistics*, 40(1):171–202.

A Pretraining Corpus Statistics

Tables 6 and 7 provide detailed per-platform statistics after all filtering and deduplication steps.

Source	**Example** (paraphrased)
Twitter (Egyptian)	تسمع اغاني تامر حسني وانت بترسم؟ دي احلى حاجه ممكن تعمليها *"Listening to Tamer Hosny while you draw? That's the loveliest thing you could be doing."*
Twitter (Gulf)	الله يسعدك، بس ليش اخذت هالفكره؟ ابي افهم امتى كانت اخر مره زرتهم *"God grant you happiness — but why did you get that idea? I want to understand: when was the last time you visited them?"*
Reddit (r/cairo, Egyptian)	ابويا رجع من الخليج معاه شويه فلوس وعايز يستثمرهم بس مالوش علاقات هنا. لو حد عنده افكار مشاريع كويسه يقولي *"My dad just got back from the Gulf with some savings and wants to invest them, but he has no connections here — if anyone has good business ideas, let me know."*
Reddit (r/jordan, Levantine)	سؤال للمتجوزين: قديش صرفتوا عالعرس كله؟ وشو الجزء يلي كلفكم اكتر شي؟ *"A question for married folks: how much did the whole wedding cost you, and which part was the most expensive?"*
YouTube (Egyptian, drama)	يا بنت عمي، انا مش قادره انسى اللي عمله فيا صوره امي واقفه قدامي طول الوقت، وحاسه اني لازم اخد حقي منه *"Cousin, I really can't forget what he did to me. My mother's face is in front of me all the time, and I feel like I have to get justice from him."*
YouTube (Gulf, drama)	ما كنا نبغى نتخلص من ابوهم الظالم، بس الحمدلله مات وارتحنا منه الحين نقدر نعيش هني بخير، وننعم بالراحه والسعاده *"We didn't wish their harsh father dead, but thank God he passed and we're free of him. Now we can live here well, and enjoy peace and happiness."*

Figure 2: Representative examples of raw pretraining text by source. Examples are paraphrased from real posts, comments, and transcripts in our corpus rather than quoted verbatim (privacy; platform redistribution terms): wording and specifics are altered while preserving the original dialect, register, and topic. Twitter posts are short and topically varied; Reddit posts are longer and conversational; YouTube transcripts reflect the spoken dialogue register of movies, series, and podcasts.

	Twitter	**Reddit**	**YouTube**
Raw documents	202.7M	18.9M	26,189
After filtering	202.7M	4.67M	25,100
Retention rate	,	24.7%	95.8%
Median length (chars)	63	129	3,416
Median length (words)	12	,	684
Max length (chars)	280	,	,
Total tokens (BPE)	730M	240M	66M
Corpus share	70.5%	23.2%	6.3%
Arabic char ratio	1.00	>0.40	>0.40
Date range	,	Jan 09–Jun 26	,
% comments	,	88%	,
Unique subreddits	,	57	,

Table 6: Per-platform corpus statistics after cleaning and deduplication. Twitter raw/retention figures are not tracked separately (full `pain/Arabic-Tweets` dataset used). YouTube figures are unique videos after MinHash near-deduplication (Jaccard threshold = 0.8).

Filtering step	**Reddit**	**YouTube**
Raw input	18,939,794	26,189
Punct.-only / near-empty	−11,916,915	,
Below min. length	−2,351,651	,
Non-Arabic script	−760	,
Exact duplicates	,	−222
Near-dups (MinHash)	,	−858
After filtering	**4,670,468**	**25,100**
Retention rate	24.7%	95.8%
Character-level cleaning (Reddit)		
Raw characters		4,027,096,974
After cleaning		1,021,920,741
Character retention		25.4%
English words removed		501,067,877
URLs removed		13,341
Quranic chars stripped		241,940

Table 7: Cleaning pipeline statistics for Reddit and YouTube; Twitter required no document-level filtering (Arabic char ratio = 1.0). Reddit's low retention (25.4%) reflects aggressive removal of Latin-script and punctuation-heavy content.

B Raw Corpus Examples

C Qualitative Generation Examples

Dialect	**Prompt** (bold) + *generation* (plain)
Egyptian	**في ناس بتقول ان مصر** كانت دوله علمانيه بس احنا بنعتبر ده كلام فارغ لان عندنا قانون وقانونا يسمح لنا بحمل السلاح ضد الرصاص مع اني في ناس بيقولوا ان مصر لا تستطيع تحمل السلاح من غير السلاح ده يا ريت ما حدا يعلق عا صوت الموسيق *"Some people say Egypt was a secular state, but we think that's empty talk because we have a law allowing us to bear arms , though some say Egypt can't carry arms without those weapons. I just wish nobody would comment on the music."*
Levantine	**والله هاد الشي صار معي امبارح** الساعه بالليل في صوت واحد قاعد عند الباب بس مافي احد بالبيت يارب ارزقني وانت خير الرازقين اللهم بشرني بما يسرني مما يخبئ لي الفرح في الجاي يارب سخرلي جنود الارض وملائكه السماء *"By God, this thing happened to me yesterday , at night there was a sound at the door but no one was home. Lord, provide for me for You are the best of providers; give me good news of what hides joy ahead; Lord, enlist for me the soldiers of earth and the angels of heaven."*
Iraqi	**صديقي شكو ماكو كلشي تمام** وين اكو احد يدزلي مسج بالواتساب حت لو من رقم غريب لا صوت زين ولا مسج ليش هالحركه انا سمعت صوت انفجار قوي بس ما ركزت كنت نايمه فيه ولا شي *"My friend, what's up , everything's fine. Where is there anyone who'd send me a WhatsApp message, even from an unknown number? Not a good word, not a message. Why this silence? I heard a huge explosion but didn't pay attention , I was asleep, nothing."*
Maghrebi	**واش راك كيفاش حالك الله يستر** وين كاين والو انت مش محتاج حد يحضنك كل ما تقع او تتوجع اكتر من صوت افكارك اللي بتحسه بيحسنلك جوده الحياه وبيخليك تحس بالامان وانت قاعد في البيت *"How are you, how are you doing , may God protect you. Where is there anything? You don't need anyone to hold you every time you fall or hurt , just the sound of your own thoughts makes life feel better and gives you a sense of safety sitting at home."*

Figure 3: Dialectal completions from SALM (temperature = 0.85, top-k = 50, repetition penalty = 1.2). Prompt in **bold**; continuation in plain text; dialect-specific items discussed in the text.

Knowledge Graphs and Distributional Readability of Arabic Documents

Mohamad Chhaytle[1,2], Hadi Hamoud[1], Sally Choker[2],
Chadi Abou Chakra[1], Mohamad Ballout[2], Fadi A. Zaraket[1,2]
[1]Arab Center for Research and Policy Studies, Doha
[2]American University of Beirut
{mchhaytle,hhamoud,cabouchakr,fzaraket}@dohainstitute.edu.qa
{mc32,ssc14,mb185,fz11}@aub.edu.lb

Abstract

This paper addresses automatic readability assessment of Arabic documents. It compares classical readability formulae, fine-tuned Arabic encoders, zero-shot large language models and a Co-occurrence Graphs for Arabic Readability (CoGAR). CoGAR, interpretable and LLM-free, represents a document as a *co-occurrence network* over content lemmas and use *topology* features such as density, clustering, path length and degree entropy to predict readability. Evaluated on the BAREC corpus over five ordinal levels, the method reaches a quadratic weighted kappa (QWK) of 0.756, on par with a sparse 35B-parameter LLM and well above the surface formulas Flesch and OSMAN. An ablation shows that the signal lies in the graph's *shape* rather than size, and that alternative graphs based on relations between named-entity carry almost no readability difficulty signal. We further distill a frontier LLM (Gemini) into a small Arabic encoder (NeoAraBERT) by silver labelling 30k chunks, reaching 0.714 QWK. We show that *stacking* an LLM with CoGAR improves on the LLM alone, reaching QWK 0.839 with GPT-4.1 and 0.852 with the open-weight Qwen-35B, our best result. The two signals are complementary and the gain does not depend on a proprietary model. Finally, we analyze how eight methods including CoGAR distribute reading levels across seven Arabic text sources ranging from children books to expert research manuscripts showing how the methods align with intuitive difficulty.

1 Introduction

Curation of text readability difficulty levels is central to producing learning materials, grading curricula, simplifying documents, and training large language models (LLMs) (Zhang et al., 2026). For Arabic, this task is made harder by rich morphology and optional diacritics, and it has long relied on formulae that range over character counts, and difficult word frequencies (Dawood, 1977; Al-Tamimi et al., 2014; El-Haj and Rayson, 2016). Recent neural network and LLM methods open new possibilities, but they are opaque and costly. Even when successful, it remains unclear how much of the difficulty signal is *structural* versus related to other features such as length and aggregate word difficulty signals.

This paper refers to any graph built based on content extracted from a document as a knowledge graph (KG). We study two such graphs: word *co-occurrence* graphs, whose nodes are content lemmas and whose edges link words that appear close together, together with semantic *entity–relation* graphs, the conventional knowledge graph of named entities and typed relations. Our central finding is that the co-occurrence graph carries almost all of the readability signal, while the entity–relation graphs we extracted were not informative. CoGAR alone, which is interpretable and LLM-free, performs at the level of a 35B-parameter LLM. Stacking CoGAR with an LLM achieves the best score including comparison with larger LLMs.

The paper makes the following contributions.

- A co-occurrence **network-topology** method for Arabic readability that is interpretable and LLM-free, yet competitive with large LLMs (QWK 0.756 ± 0.001 over repeated cross-validation).
- A controlled **ablation** isolating which graph signals carry difficulty: topology dominates volume, while named-entity and relation graphs provide no additional value.
- A **systematic comparison** on BAREC (Elmadani et al., 2025b) of readability formulas, two fine-tuned Arabic encoders, three LLMs and the graph, under a single ordinal evaluation.

Proceedings of The Fourth Arabic Natural Language Processing Conference (ArabicNLP 2026), pages 508–521
October 28-29, 2026 ©2026 Association for Computational Linguistics

- A **knowledge-distillation** pipeline turning a frontier LLM (Gemini (Gemini Team, Google DeepMind, 2026)) into a small encoder (NeoAraBERT (Abou Chakra et al., 2026)), reaching QWK 0.714.
- A **stacking** of an LLM and CoGAR that reaches QWK 0.839 with GPT-4.1 and 0.852 with the open-weight Qwen-35B, showing the two signals are complementary and that the gain does not depend on a proprietary model.
- A **cross-source distributional study** of eight methods over seven Arabic categorical text types (children books, mystery fiction, ..., Academic research manuscripts) that map to conventional age/educational groups, on unlabeled data.

Our code is publicly available.[1]

2 Related Work

Arabic-specific readability factors. Several features directly affect the readability of Arabic text. Two of them are particularly useful for estimating the difficulty level of a sentence: diacritics and morphology. Diacritics help disambiguate intended semantics, role in a sentence, and pronunciation. When omitted young readers may struggle to read Arabic, which is why books intended for children are usually heavily vocalized. Arabic morphology relies on a root-and-pattern system: the root is a set of three, four or exceptionally five letters carrying a central meaning or concept. The root system helps: knowing a root often reveals the meaning of a new word. Prefixes, suffixes and infixes apply to the same word, so a beginner struggles to separate morphemes.

Syntactic complexity and lexical difficulty also matter, though neither appears in surface counts. These properties make estimating the difficulty of an Arabic text non-trivial and explain why dedicated readability measures have been developed.

The earliest readability formulas. The study of readability is not recent. The earliest approaches rely on mathematical formulas based on word and sentence length. Flesch (1948) proposed the Flesch Reading Ease formula, one of the oldest:

$$\text{FRE} = 206.835 - 1.015\,\text{ASL} - 84.6\,\text{ASW},$$

where ASL and ASW are the average sentence length (words per sentence ratio) and average number of syllables per word. The score ranges from 0 to 100: the higher it is, the easier the text is to read. The U.S. Navy adopted this into the Flesch–Kincaid formula (Kincaid et al., 1975) to indicate the school grade level required to understand a text:

$$\text{FK} = 0.39\,\text{ASL} + 11.8\,\text{ASW} - 15.59.$$

Other formulas followed, such as the Gunning Fog Index (Gunning, 1969) and SMOG (McLaughlin, 1969). Several studies (Bruce et al., 1981; Redish, 2000) critically examined the limitations and degrees of accuracy of the existing readability formulas. These formulas do not transfer directly from English to Arabic, which motivated the development of language-specific formulas.

Arabic readability formulas. Among the first formulas dedicated to Arabic, Dawood (1977) proposed a formula based on three textual features: AWL (average word length), ASL, and WF (word frequency):

$$\begin{aligned}\text{Score} = &- 0.0533\,\text{AWL} - 0.2066\,\text{ASL} \\ &+ 5.5543\,\text{WF} - 1.0801.\end{aligned}$$

Later, Al-Tamimi et al. (2014) published the Arabic Automatic Readability Index (AARI), combining the number of characters (NoC), ASL, and AWL:

$$\text{AARI}_{\text{base}} = 3.28\,\text{NoC} + 1.43\,\text{AWL} + 1.24\,\text{ASL},$$

. Regression then maps the results to school grade levels:

$$\text{Grade} = (\text{AARI}_{\text{base}} + 472.42)/1046.3.$$

El-Haj and Rayson (2016) proposed the OSMAN formula, which evaluates readability with or without diacritics and introduces a "Faseeh" factor (the letters ء, ئ, ؤ, ذ, ظ and the final sequences وا and ون):

$$\begin{aligned}\text{OSMAN} = 200.791 &- 1.015\,\tfrac{A}{B} \\ &- 24.181\,\tfrac{C+D+G+H}{A},\end{aligned}$$

where A is the total number of words, B the number of sentences, C the number of hard words (> 5 letters), D the number of syllables per word, G the number of complex words (> 4 syllables), and H the number of Faseeh words. All these formulas rely on surface cues (lengths, counts); to capture finer aspects of difficulty. Subsequent work turned to richer linguistic features and machine learning.

[1] https://github.com/U4RASD/arabic-readability-cogar

Feature-based and machine-learning approaches. We distinguish here between Arabic as a first language (L1, native speakers) and Arabic as a second language (L2, foreign learners); the difficulty of a text is not measured the same way for these two audiences. Saddiki et al. (2015) sought to determine the performance achievable with a minimal set of easy-to-extract features: using a GLOSS corpus of 251 texts spread over five ILR (Interagency Language Roundtable proficiency) levels and 35 low-complexity features, they reported a maximum accuracy of 59.76% for five classes, rising to 73.31% when reducing the problem to three classes. Saddiki et al. (2018) examined a much larger set of features on a GLOSS-based corpus of 545 Modern Standard Arabic texts; with four classes, their best L2 accuracy reached 72.4%. Nassiri et al. (2020) released the MoSAR corpus (Modern Standard Arabic Readability), a resource for training, optimizing, and evaluating readability approaches for L1 learners; they later used it (Nassiri et al., 2022) to assess the benefit of combining classical features with more sophisticated ones, reaching an accuracy of about 78%. Khallaf and Sharoff (2021) classified the difficulty of sentences for L2 learners by combining sentence embeddings with classical linguistic features, reporting F-scores of 0.80 and 0.75 with Arabic-BERT and XLM-R respectively.

Arabic readability resources. Several resources now support fine-grained Arabic readability. The SAMER project released a leveled readability lexicon of 26,000 lemmas for Modern Standard Arabic (Al Khalil et al., 2020), which later enabled the SAMER Corpus (Alhafni et al., 2024), the first manually annotated Arabic parallel corpus for text simplification, covering 159K words with readability annotations at both the document and word levels. El-Haj et al. (2024) introduced DARES, a dataset of 13,335 instances from Saudi school textbooks, annotated both at a coarse educational level and at the level of individual grades. The BAREC corpus (Elmadani et al., 2025b; Habash et al., 2025) that we use here annotates sentences on a 19-level scale, and has since been extended to BAREC-10M (Elmadani et al., 2026), which reaches 10 million words across 45 sub-corpora annotated for morphology, syntax and readability.

Arabic readability modeling. Liberato et al. (2024) compared approaches ranging from rule-based methods to Arabic pretrained language models, reaching a macro F1 of 87.9 at the fragment level. The BAREC corpus paper (Elmadani et al., 2025b) benchmarks several models on its 19-level scale and releases the resulting classifiers, two of which we use as baselines. The BAREC Shared Task 2025 (Elmadani et al., 2025a) extended this benchmarking to sentence-level and document-level tracks, drawing 22 registered teams, with the winning system reaching 87.5 QWK at the sentence level.

Structural and neural approaches. Beyond surface features, some work exploits the structure of the text or neural representations, often for languages other than Arabic. Feng et al. (2010) show that, among discourse features, only entity-density features have a significant impact on predicting the readability of texts for elementary-school students (for English); this finding motivates examining text structure, which we approach in this work through word co-occurrence networks. Qiu et al. (2021) propose incorporating linguistic features into neural network models by learning dense syntactic embeddings, and conclude that this approach can complement BERT to substantially improve automatic readability assessment.

Large language models. Since the emergence of LLMs such as GPT (OpenAI, 2025), Qwen (Qwen Team, 2026), and Gemini (Gemini Team, Google DeepMind, 2026), these models have demonstrated strong performance on many natural language processing tasks and across many languages, including Arabic (Al-Khalifa et al., 2025). Trained on massive corpora, they possess broad lexical, syntactic, and semantic knowledge, making them in principle capable of judging the difficulty of a text without task-specific training (zero-shot), from a simple instruction describing the level scale. Combined with the emergence of Arabic-specialized LLMs, this capability makes them natural candidates for readability assessment, an approach we evaluate in this work alongside classical methods.

Challenges and positioning. Despite these advances, Arabic readability assessment remains limited by the scarcity of fine-grained annotated corpora (Nassiri et al., 2023). Existing resources often rely on coarse grade-level labels (three or

four classes), and many Arabic text collections lack difficulty annotations altogether. In this work, we address this gap by evaluating on the fine-grained BAREC corpus (Elmadani et al., 2025b) and by analysing, without gold labels, seven Arabic text sources spanning children's stories, textbooks, news, and research books.

3 Data

Ground truth (BAREC). We use the BAREC corpus (Elmadani et al., 2025b; Habash et al., 2025) as ground truth for supervised evaluation. We build a five-level evaluation set of 1,000 paragraphs, 200 per level (levels 0 to 4, easiest to hardest). Our levels 1 to 4 correspond to BAREC levels 10, 12, 14 and 16, a regular subsampling of the BAREC scale rather than its first four levels. Each paragraph accumulates consecutive sentences of a single BAREC document, in document order, until a window of 20 to 60 words is reached (mean $\approx$ 42 words, 97% within the window). Its level is that of its hardest sentence, following the BAREC aggregation rule (Elmadani et al., 2025b). This leaves the low levels almost empty, since a paragraph stays low only if all its sentences are: levels 1 to 6 yield between 0 and 63 paragraphs, far below the 200 needed per class. Levels 10, 12, 14 and 16 provide abundant material with a regular spacing that keeps classes well separated.

Level 0, the easiest, does not come from BAREC: it consists entirely of short sentences from a first-grade Arabic-language textbook of the Saudi national curriculum (Ministry of Education, Kingdom of Saudi Arabia, 2024), with a mean length of 4.4 words per sentence. We use this source rather than the lowest BAREC level because, upon inspection, the easiest BAREC sentences are dominated by non-sentential fragments (titles, author names, page numbers and other editorial metadata) that do not constitute coherent reading material. These sentences are not annotated under the BAREC guidelines and we claim no formal equivalence with its scale. As an indirect check, the strongest released CAMeL classifier places 97.5% of them below BAREC level 10, which corresponds to our level 1, with a median of level 6.

Multi-source corpus. For a practical distributional study (Section 7) we collect 4,200 chunks, 600 from each of seven categorically different sources: children's stories, detective stories, middle-school textbooks, high-school textbooks, Arabic Wikipedia, newspaper articles, and academic research manuscripts. While these documents are not labeled on difficulty, each of them reflects a categorically suitable age group and educational level. Intuition is the distribution across readability levels should reflect the categorical difference between these corpora.

We segment each of these sources at sentence boundaries into the same 20–60 word window. We use these chunks to compare how methods *distribute* levels.

Silver corpus for distillation. For knowledge distillation (Section 4) we label 30,000 chunks with a frontier LLM (Gemini-3.5-flash), which serve as silver training data for a small Arabic encoder. Appendix C documents the provenance and versions of every dataset and model used in this work.

4 Methods

Readability is an *ordinal* task: predicting level 3 for a level 4 text is a smaller error than predicting level 0. We therefore score every method with the quadratic weighted kappa (QWK) (Cohen, 1968), which penalises a large deviation more than a small one.

Readability formulas. We apply three formulas: OSMAN (El-Haj and Rayson, 2016), AARI (Al-Tamimi et al., 2014) and a version of Flesch reading ease that adapts for English syllabification. Each continuous score is mapped to the five levels by rank binning into equal-frequency groups, oriented so that higher levels correspond to harder texts. Flesch and OSMAN are reading-*ease* scores, where a higher raw value means an easier text, so their bins are reversed. AARI is a grade-level index that already increases with difficulty.

Fine-tuned Arabic encoders. As supervised neural baselines we use two publicly available readability classifiers of CAMeL Lab (Elmadani et al., 2025b), applied directly to our paragraphs: `readability-arabertv02-word-CE`, which takes raw text, together with `readability-arabertv2-d3tok-reg`, the strongest released variant, which takes the D3Tok morphological tokenization as input and uses a regression head. Both models were already fine-tuned on BAREC, the same corpus our

levels 1–4 are drawn from, which gives them a strong in-domain advantage as baselines.

LLMs as zero-shot raters. We prompt three LLMs: GPT-4.1 (OpenAI, 2025), Qwen-35B (`Qwen3.5-35B-A3B`) (Qwen Team, 2026), and Fanar (`Fanar-C-2-27B`) (TEAM et al., 2026), with a fixed rubric that defines the five reading levels and asks for a single integer (Appendix D). The same rubric is used to produce the Gemini silver labels.

CoGAR: Co-occurrence Graphs for Arabic Readability. *Intuition.* We hypothesize that the difficulty of a text is reflected in the *structure* of how its words connect, not only in surface statistics such as length or frequency: a harder text weaves a denser, more interconnected network of concepts, whereas an easier text yields a simpler, more linear one. We capture this structure with a word co-occurrence network.

Graph construction. Each document is analysed with SinaTools 1.0.9 (Hammouda et al., 2024) to obtain, for every word, its lemma, root, part of speech, and frequency. We keep only *content lemmas* (nouns, verbs, adjectives) and discard function words. We then build an *undirected* graph whose nodes are these lemmas and whose edges link two lemmas that co-occur within a window of two consecutive content words.

Features. From this graph we extract two types of metrics: *volume* (number of nodes and edges), which merely reflect text size, and *topology*, which describes the shape of the network independently of its size: density, average clustering coefficient (how interconnected a word's neighbours are), average shortest-path length and diameter (how far apart concepts are), average degree, degree entropy (how uneven word connectivity is), and assortativity (whether highly connected words link to one another). In total we compute 40 features in four families: lexical (8: lemma and root diversity, length, rarity via frequency), co-occurrence (12: 2 volume and 10 topology), named entities (5: diversity and density over the 21 Wojood (Jarrar et al., 2022) categories) and an entity-relation graph (15: relation counts, types and topology). Appendix A lists all of them individually.

Entity–relation graph. We also build a directed entity–relation graph for each document, a standard knowledge graph of entities and the relations between them. Using SinaTools, we extract the named entities (the same 21 entity categories as above) together with the semantic relations that hold between them; each entity becomes a typed node and each extracted relation becomes a directed, labelled edge between two entities. From this graph we derive the 15 relation features: the number of relations, the number of distinct relation types and their entropy, and the same topology measures as for the co-occurrence graph (density, clustering, path length, degree entropy, assortativity). This graph encodes *which entities are related to which*, in contrast to the co-occurrence graph, which only encodes which content words appear close together.

Model and training. As readability is ordinal, CoGAR predicts the level with an ordinal regressor (LogisticAT (Pedregosa et al., 2017)), which respects the ordering of the levels. All supervised models are trained and evaluated by stratified five-fold cross-validation on our 1,000-paragraph set (BAREC levels 1–4 together with the level-0 Saudi textbook): every paragraph receives its prediction from a fold in which it was not used for training (out-of-fold prediction). Cross-validation lets us evaluate on the whole set, which matters at this size, since rebuilding our paragraphs under the official BAREC splits would leave classes too imbalanced for the task. The CAMeL classifiers are applied to the same paragraphs. We verify stability by repeating the cross-validation over ten random seeds (reported in the robustness analysis below).

Ablation design. Finally, we evaluate each feature family *in isolation*, using the same ordinal model and the same cross-validation, to determine whether the difficulty signal comes from the graph's *shape* (topology) or merely from its *size* (volume).

Knowledge distillation. We distill Gemini-3.5-flash (Gemini Team, Google DeepMind, 2026) into NeoAraBERT (Abou Chakra et al., 2026), a small Arabic encoder. The teacher labels 30,000 chunks on the five-level scale; the student is then fine-tuned on these silver labels for three epochs. We train two variants, one with a class-weighted cross-entropy loss and one with a focal loss. We report both. The focal variant transfers marginally better (0.714 against 0.708), a gap we do not treat as meaningful since the comparison was made on BAREC rather than on a held-out set. Notably, the student never sees any human annotation: it learns the readability scale entirely from the LLM's silver labels, and is evaluated on the human-labelled BAREC set.

5 Experimental Setup

Our primary metric is the quadratic weighted kappa (QWK), introduced above. We also report exact accuracy, within-1 and within-2 accuracy, mean absolute error (MAE), and Spearman correlation. Supervised models use stratified five-fold out-of-fold cross-validation, for the graph we additionally repeat the cross-validation over ten random seeds to report mean and standard deviation.

6 Results

6.1 Main comparison

Table 1 reports all methods on BAREC, sorted by QWK. The strongest released CAMeL variant, `arabertv2-d3tok-reg`, obtains the highest score (0.860). As an encoder fine-tuned on the annotated BAREC sentences from which our levels 1 to 4 are drawn, it benefits from a strong in-domain advantage. Among the remaining systems, GPT-4.1 leads (QWK 0.784), although its margin over CoGAR is small and at the edge of significance (paired bootstrap, $+0.028$, $p = 0.05$). CoGAR follows closely: the topology variant reaches 0.756 ± 0.001 and the full feature set 0.750 ± 0.005, statistically indistinguishable from the sparse Qwen3.5 model (35B total, 3B active, QWK 0.751, $p = 0.83$) and clearly above Fanar (0.702) and the `arabertv02-word-CE` classifier (0.694). The distilled NeoAraBERT (0.714) is competitive with Fanar and with the `arabertv02-word-CE` classifier. Flesch and OSMAN are reading-*ease* scores while AARI is a grade-level index. We bin all of them in the direction of increasing difficulty, so every reported value is positive: OSMAN 0.215 and Flesch 0.186, both far below every learned method, while AARI reaches 0.607. For every strong method, within-2 accuracy is close to 0.99, i.e. predictions are almost never off by more than two levels.

6.2 What in the graph carries the signal?

Table 2 compares feature families in isolation, all using the same ordinal model and cross-validation. The entity-relation graph is essentially uninformative (0.023) and named entities are weak (0.338): entity density follows the *topic*, not the difficulty: the hardest, most abstract texts contain almost no linked entities, whereas a co-occurrence graph exists for every text. The co-occurrence *volume* (size alone) reaches 0.682, but the *topology* rises to 0.756, showing that the difficulty signal is in the network's *shape*, not its size. Crucially, adding AARI (explicit length) to the full feature set brings no meaningful gain (`aari+all` 0.752 vs. all features 0.750, well within one standard deviation): length is already encoded in the graph.

6.3 Feature importance

A random-forest importance over the full feature set confirms this picture: nine of the fifteen most important features are co-occurrence topology, and the top four are all graph-structural (`cooc_density` 0.070, `cooc_n_edges` 0.069, `cooc_avg_clustering` 0.063, `cooc_avg_path` 0.062), ahead of the strongest lexical feature (words per sentence, 0.061).

Sensitivity to the co-occurrence window. Table 6 in Appendix B reports the topology configuration for window sizes from one to five content words, over five seeds per value. The total range is 0.024, so the method is robust to this parameter. A window of one obtains a slightly higher score than our choice of two, a difference of 0.008 within that range, which indicates that we did not tune this parameter on the evaluation data.

6.4 Robustness

Repeating the cross-validation over ten seeds, co-occurrence topology reaches QWK $= 0.756 \pm 0.001$ (10 seeds $\times$ 5 folds) versus 0.750 ± 0.005 for all features. A paired bootstrap over the same seeds finds no significant difference between the two configurations ($p = 0.52$), so we do not claim that topology performs better. We adopt it as our method of choice because it is five times more stable and relies on four times fewer features while reaching the same level of performance.

6.5 Significance testing

To support the system-versus-system claims, we run a paired bootstrap over the ten cross-validation seeds. CoGAR topology is statistically indistinguishable from Qwen-35B (difference below 0.01, $p = 0.83$) and is numerically ahead of it. GPT-4.1 scores higher than CoGAR, but by a small margin at the edge of significance ($+0.028$, $p = 0.05$), which is why we avoid describing its lead as es-

[2]For the rank-binned formula methods, the predicted levels have a uniform marginal identical to the balanced gold distribution; under this condition quadratic weighted kappa reduces to the Spearman rank correlation, so the two columns coincide.

Method	Exact	±1	±2	MAE	Spearman	QWK
CAMeL `arabertv2-d3tok-reg`	0.571	0.989	0.999	0.441	0.887	0.860
GPT-4.1 (zero-shot)	0.506	0.921	0.998	0.575	0.836	0.784
CoGAR (topology)	0.497	0.889	0.993	0.621	0.734	$0.756_{\pm 0.001}$
Qwen-35B (zero-shot)	0.466	0.911	0.998	0.625	0.834	0.751
CoGAR (all features)	0.491	0.886	0.989	0.634	0.731	$0.750_{\pm 0.005}$
NeoAraBERT (distilled, focal)	0.443	0.893	0.998	0.666	0.808	0.714
Fanar (zero-shot)	0.408	0.913	0.988	0.691	0.778	0.702
CAMeL `arabertv02-word-CE`	0.515	0.893	0.955	0.654	0.740	0.694
AARI	0.458	0.785	0.923	0.834	0.607	0.607
OSMAN	0.224	0.564	0.814	1.416	0.215	0.215
Flesch	0.245	0.583	0.813	1.404	0.186	0.186

Table 1: Main comparison on BAREC ($n = 1000$, five ordinal levels), sorted by quadratic weighted kappa (QWK). Higher is better except MAE. CoGAR values are means over ten cross-validation seeds with standard deviation. All other methods are deterministic and are reported as single values.[2]

Feature family	QWK
Entity-relation graph	$0.023_{\pm 0.005}$
Named entities (NER)	$0.338_{\pm 0.002}$
AARI (length only)	$0.622_{\pm 0.002}$
Co-occurrence *volume* (size)	$0.682_{\pm 0.001}$
Lexical (morphology, rarity, length)	$0.696_{\pm 0.003}$
Graph (cooc.+NER+rel.)	$0.748_{\pm 0.004}$
All features	$0.750_{\pm 0.005}$
AARI + all features	$0.752_{\pm 0.004}$
Co-occurrence graph (vol.+topo.)	$0.754_{\pm 0.002}$
Co-occurrence *topology*	$0.756_{\pm 0.001}$

Table 2: Feature-family ablation on BAREC (ordinal LogisticAT, five-fold CV), reported as mean ± standard deviation over ten seeds. Topology dominates size and entity/relation graphs fail. Topology and the full feature set are statistically indistinguishable (paired bootstrap, $p = 0.52$), so no configuration is marked as best.

tablished. The two CoGAR variants are equivalent ($p = 0.52$), which supports adopting the simpler topology configuration. The classical formulas remain far behind every learned method ($p < 0.0001$).

6.6 Distillation results

Table 3 reports the distilled NeoAraBERT. Trained only on Gemini silver labels, without ever seeing a human label, the student transfers to human BAREC labels at QWK 0.714 with the focal loss, above the class-weighted variant (0.708) and competitive with the `arabertv02-word-CE` classifier (0.694) and Fanar (0.702). The NeoAraBERT row in Table 1 shows the focal variant. Both variants are reported in Table 3. Its confusion matrix (Figure 1) shows that errors concentrate on the ±1 band, with a systematic under-prediction of the hardest level (most level-4 paragraphs are predicted as level-3).

Loss	Exact	±1	MAE	QWK
Weighted CE	0.407	0.912	0.682	0.708
Focal	0.443	0.893	0.666	0.714

Table 3: NeoAraBERT distilled from Gemini silver labels, evaluated on human-labelled BAREC. We report both loss variants rather than selecting one. The gap between them is 0.006 QWK, which we do not treat as meaningful. The comparison was made on BAREC rather than on a held-out set.

6.7 Combining the graph and the LLM

Averaging GPT-4.1 with the co-occurrence topology (*fusion*) gives QWK 0.768, which does *not* beat GPT-4.1 alone (0.784). However, *stacking* an ordinal meta-learner on the GPT-4.1 rating and the graph topology features reaches QWK $= 0.839 \pm 0.001$ (5 seeds $\times$ 5 folds), well above either component (Table 4); the protocol and its leakage controls are detailed below.

Stacking protocol. Our strongest system combines the LLM and the graph with an ordinal meta-learner: an ordinal logistic regression (LogisticAT) fed two complementary inputs, the GPT-4.1 zero-shot rating and the co-occurrence *topology* features of the document. We guard against leakage on every input. First, GPT-4.1 is used zero-shot and is never trained on our data, so its rating carries no label information. Second, the topology inputs are deterministic functions of the text (graph density, clustering, path length, degree entropy, assortativity), not the output of any trained model. Third, the meta-learner is evaluated strictly out-of-fold: under stratified five-fold cross-validation, each paragraph receives its stacked prediction from a meta-model trained only

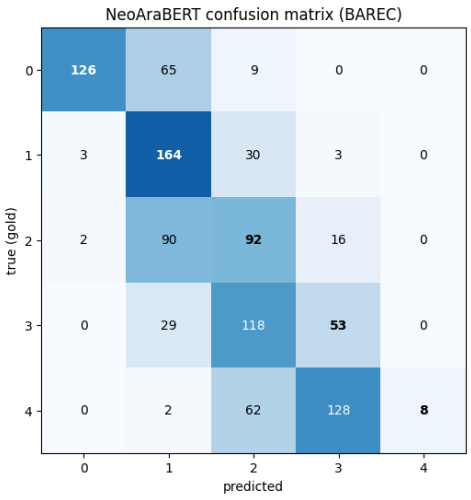


Figure 1: NeoAraBERT (focal) confusion matrix on BAREC (rows: gold, columns: predicted). Errors are concentrated on the ± 1 band; the hardest level is systematically under-predicted.

System	QWK
Co-occurrence topology (alone)	$0.756_{\pm 0.001}$
GPT-4.1 (alone)	0.784
Qwen-35B (alone)	0.751
Fusion (averaging GPT-4.1 and topology)	0.768
Stacking (GPT-4.1 + topology)	$0.839_{\pm 0.001}$
Stacking (Qwen-35B + topology)	$0.852_{\pm 0.002}$

Table 4: Combining the LLM and the graph. Naive averaging does not help but stacking is complementary. The gain does not depend on a proprietary model: stacking the open-weight Qwen-35B with the topology features reaches a slightly higher score than stacking GPT-4.1.

on the other folds, so no paragraph is ever used to both fit and score the meta-learner. As the meta-learner has no tuned hyper-parameters, a single (non-nested) cross-validation level is unbiased. We repeat the procedure over five random seeds and report mean $\pm$ standard deviation, QWK $=$ 0.839 ± 0.001. Replacing the topology features by the topology *prediction* alone yields a comparable QWK $=$ 0.836, confirming the gain is not an artefact of one particular meta-feature choice. The same procedure with Qwen-35B, an open-weight model, reaches QWK $=$ 0.852 ± 0.002, slightly above the GPT-4.1 stacking. The graph and the LLM therefore capture *complementary* signals. The gain does not depend on a proprietary model.

Evaluation without level 0. Since level 0 comes from an external source and is trivially separable, we also report performance on the four BAREC levels alone, with CoGAR retrained on those levels only. Every learned method loses ground: GPT-4.1 falls from 0.784 to 0.604, Qwen-35B from 0.751 to 0.584, Fanar from 0.702 to 0.565 and CoGAR topology from 0.756 to 0.429. This drop is largely mechanical. QWK is a ratio of observed to expected disagreement. Removing a perfectly separable class at one end of the scale sharply reduces the expected disagreement without reducing the observed one. Within-1 accuracy, which is more informative on an ordinal scale, remains high and tightly grouped: 0.901 for GPT-4.1, 0.891 for Qwen-35B, 0.896 for Fanar and 0.863 for CoGAR topology. The surface formulas retain almost no signal in this restricted setting, which indicates that what they capture is concentrated at the easiest end of the scale.

7 Cross-source Distributional Analysis

Figure 2 shows, for each of the seven sources, how eight methods distribute the five reading levels, together with their mean predicted level. This is run on unlabelled data, so it probes *generalisation* rather than accuracy. We compare each method's mean predicted level per source against the intuitive order children's stories $<$ detective stories $<$ middle-school $<$ high-school $<$ newspaper $<$ research books, using Kendall's τ. Arabic Wikipedia is reported separately rather than placed in this order: its articles range from short factual entries to technical texts. All eight methods place it among the easiest sources, which is consistent with the brevity of many Arabic Wikipedia entries. CoGAR and the LLMs follow the intuitive trend (τ from 0.73 to 0.87, CoGAR at 0.83), while the formulas do not (OSMAN 0.47, AARI 0.33, Flesch 0.07). No method recovers the exact order, so the agreement concerns the overall trend rather than the precise ranking. Power is limited with six sources. This confirms, on data with no gold labels, that the structural and LLM-based methods carry a transferable notion of difficulty, whereas surface formulas do not.

8 Conclusion

We showed that CoGAR, a simple and interpretable co-occurrence *network topology* method, predicts Arabic reading difficulty as well as a 35B-parameter (3B active) LLM and well above the surface formulas Flesch and OSMAN, with the signal

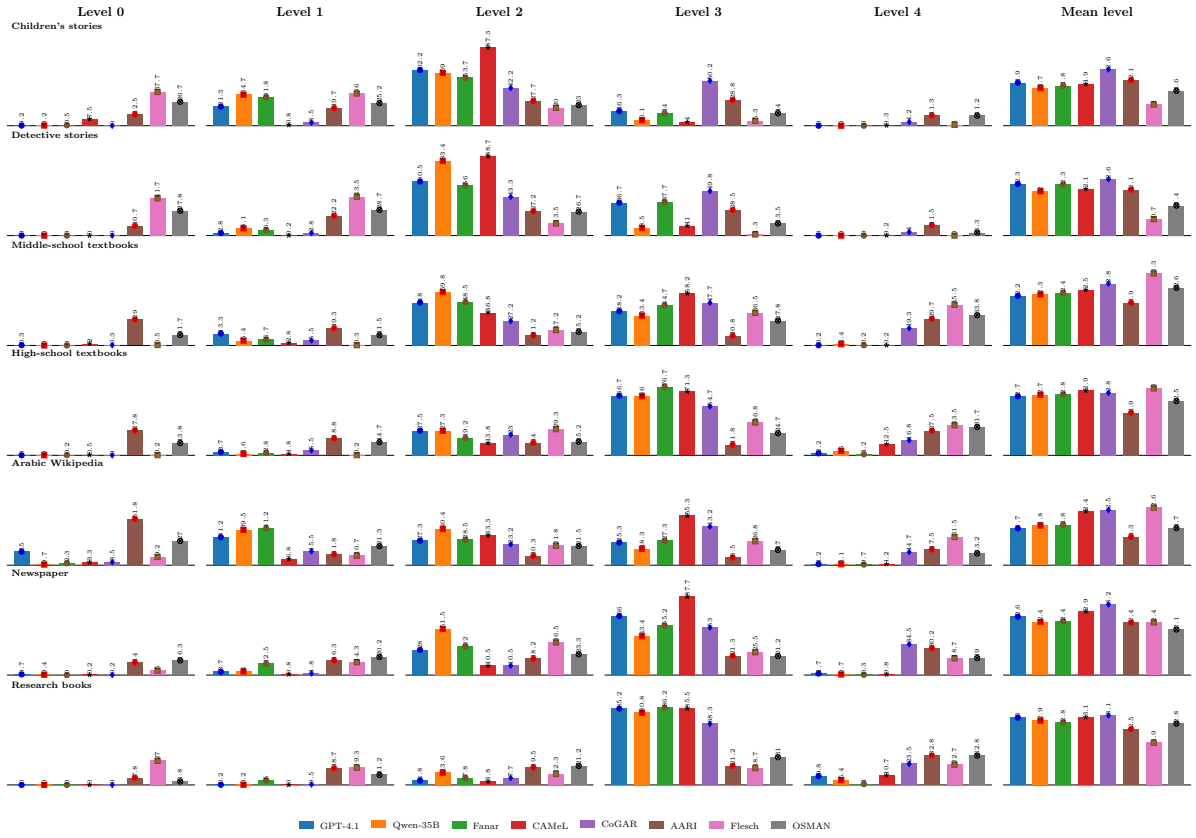


Figure 2: Distribution of predicted reading levels (and mean level, rightmost column) for eight methods across seven Arabic text sources. CoGAR and the LLMs follow the intuitive difficulty trend (Kendall's τ between 0.73 and 0.87 over six sources). Flesch and OSMAN are noisy or inverted.

residing in the graph's shape rather than its size. We distilled a frontier LLM into a small Arabic encoder. We found that stacking an LLM with CoGAR yields our best result, whether the LLM is proprietary or open-weight, the two signals being complementary. A cross-source analysis further showed that structural and LLM-based methods generalise to unlabelled text where formulas fail. Future work includes extending to more corpora, exploring lighter open models as graph companions and evaluating whether topology still dominates volume at BAREC's native sentence granularity, where very short texts yield nearly complete graphs.

9 Limitations

Supervised evaluation relies on a single corpus (BAREC); broader corpora would strengthen the conclusions. The easiest level (level 0) is drawn entirely from a separate first-grade textbook rather than from BAREC; its much shorter sentences make it more easily separable, so absolute scores may be partly inflated by this class, and excluding this class changes the relative ranking: CoGAR then scores below the three evaluated zero-shot LLMs. Enriching this level with additional easy-text sources is left to future work. The LLM raters are closed and costly. We cannot rule out exposure to similar data during pre-training, which is precisely an argument in favor of an LLM-free method whose signal provenance is verifiable: the CoGAR features are deterministic functions of the text, computable and inspectable. The co-occurrence graph depends on the quality of SinaTools' morphological analysis. Its window size proved robust (Table 6) but its feature set is only lightly tuned. The distillation teacher labels are themselves model-generated (silver), so the student inherits any teacher bias. Finally, the cross-source distributional study has no gold labels and is therefore qualitative.

References

Chadi Abou Chakra, Hadi Khaled Hamoud, Osama Rakan Al Mraikhat, Qusai Abu Obaida, Mohamad Ballout, and Fadi Zaraket. 2026. NeoAraBERT: A

modern foundation model for Arabic embeddings with diacritics-aware tokenization and POS-targeted masking. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 25952–25968, San Diego, California, United States. Association for Computational Linguistics.

Shahad Al-Khalifa, Nadir Durrani, Hend Al-Khalifa, and Firoj Alam. 2025. The landscape of arabic large language models (allms): A new era for arabic language technology. *Preprint*, arXiv:2506.01340.

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(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.

Bertram Bruce, Andee Rubin, and Kathleen Starr. 1981. Why readability formulas fail. *IEEE Transactions on Professional Communication*, (1):50–52.

Jacob Cohen. 1968. Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, 70(4):213.

BAK Dawood. 1977. The relationship between readability and selected language variables. Unpublished master's thesis (in arabic), Baghdad University, Iraq.

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: Shared Tasks*, pages 239–252, 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.

Khalid N. Elmadani, Adel Mahmoud Wizani, Hanada Taha Thomure, and Nizar Habash. 2026. A large and balanced multi-domain arabic corpus annotated for morphology, syntax, and readability. In *Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026)*, pages 11761–11775, Palma, Mallorca, Spain. European Language Resources Association (ELRA).

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.

Rudolf Flesch. 1948. A new readability yardstick. *Journal of Applied Psychology*, 32(3):221–233.

Gemini Team, Google DeepMind. 2026. Gemini 3.5 Flash. Generally available May 2026. `https://deepmind.google/models/gemini/flash/`.

Robert Gunning. 1969. The fog index after twenty years. *Journal of Business Communication*, 6(2):3–13.

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)*, Vienna, Austria. Association for Computational Linguistics.

Tymaa Hammouda, Mustafa Jarrar, and Mohammed Khalilia. 2024. SinaTools: Open source toolkit for Arabic natural language understanding. In *Proceedings of the 2024 AI in Computational Linguistics (ACLING 2024)*, Procedia Computer Science, Dubai, United Arab Emirates. Elsevier.

Mustafa Jarrar, Mohammed Khalilia, and Sana Ghanem. 2022. Wojood: Nested Arabic named entity corpus and recognition using BERT. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3626–3636, Marseille, France. European Language Resources Association.

Nouran Khallaf and Serge Sharoff. 2021. Automatic difficulty classification of Arabic sentences. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 105–114, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

J. Peter Kincaid, Robert P. Fishburne, 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, Naval Technical Training Command, Millington, TN, Research Branch.

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.

G. Harry McLaughlin. 1969. SMOG grading: A new readability formula. *Journal of Reading*, 12(8):639–646.

Ministry of Education, Kingdom of Saudi Arabia. 2024. *Lughati lil-Saff al-Awwal al-Ibtida'i: al-Fasl al-Dirasi al-Awwal*, 1 edition. Ministry of Education, Riyadh, Saudi Arabia.

Naoual Nassiri, Violetta Cavalli-Sforza, and Abdelhak Lakhouaja. 2020. Mosar: Modern standard arabic readability corpus for l1 learners. In *Proceedings of the 4th International Conference on Big Data and Internet of Things*, BDIoT '19, New York, NY, USA. Association for Computing Machinery.

Naoual Nassiri, Violetta Cavalli-Sforza, and Abdelhak Lakhouaja. 2023. Approaches, methods, and resources for assessing the readability of arabic texts. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22(4).

Naoual Nassiri, Abdelhak Lakhouaja, and Violetta Cavalli-Sforza. 2022. Combining classical and non-classical features to improve readability measures for arabic first language texts. In *Advanced Intelligent Systems for Sustainable Development (AI2SD'2020)*, pages 463–470.

OpenAI. 2025. Introducing GPT-4.1 in the API. https://openai.com/index/gpt-4-1/. Accessed: 2025.

Fabian Pedregosa, Francis Bach, and Alexandre Gramfort. 2017. On the consistency of ordinal regression methods. *Journal of Machine Learning Research*, 18(55):1–35.

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.

Qwen Team. 2026. Qwen3.5: Towards native multimodal agents. Alibaba Cloud technical blog. Accessed 2026.

Janice Redish. 2000. Readability formulas have even more limitations than Klare discusses. *ACM Journal of Computer Documentation*, 24(3):132–137.

Hind Saddiki, Karim Bouzoubaa, and Violetta Cavalli-Sforza. 2015. Text readability for Arabic as a foreign language. In *2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA)*, pages 1–8.

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.

FANAR TEAM, Ummar Abbas, Mohammad Shahmeer Ahmad, Minhaj Ahmad, Abdulaziz Al-Homaid, Anas Al-Nuaimi, Enes Altinisik, Ehsaneddin Asgari, Sanjay Chawla, Shammur Chowdhury, Fahim Dalvi, Kareem Darwish, Nadir Durrani, Mohamed Elfeky, Ahmed Elmagarmid, Mohamed Eltabakh, Asim Ersoy, Masoomali Fatehkia, Mohammed Qusay Hashim, and 18 others. 2026. Fanar 2.0: Arabic generative ai stack.

Yang Zhang, Amr Mohamed, Hadi Abdine, Guokan Shang, and Michalis Vazirgiannis. 2026. Beyond random sampling: Efficient language model pretraining via curriculum learning. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5776–5794.

A Full Feature Inventory

Table 5 lists the 40 features in the four families used in Section 4, with the volume and topology split that Table 2 scores. Volume features count the size of the graph. Topology features describe its shape independently of that size.

B Co-occurrence Window Sensitivity

Table 6 reports the topology configuration for co-occurrence window sizes from one to five content words, over five seeds per value.

C Data Provenance, Versions and Licenses

Table 7 documents the origin of every dataset used in this work. The unlabelled corpora were drawn from the collection of the Arab Center for Research and Policy Studies[3]. None of the collections contains personal data.

[3] https://www.dohainstitute.org

Models and versions. SinaTools 1.0.9 (Hammouda et al., 2024) for morphological analysis and named entity recognition. The CAMeL readability classifiers `readability-arabertv02-word-CE` and `readability-arabertv2-d3tok-reg` (Elmadani et al., 2025b), obtained from their public releases. NeoAraBERT (Abou Chakra et al., 2026) as the distillation student. The LLM raters GPT-4.1 (OpenAI, 2025), Qwen-35B (Qwen Team, 2026) and Fanar (TEAM et al., 2026) were queried through their respective APIs. Gemini-3.5-flash (Gemini Team, Google DeepMind, 2026) produced the silver labels.

D LLM Prompt

The same instruction was used for the three zero-shot raters and for the Gemini silver labelling. Everything before `Text to assess:` is sent as the system message and the document follows as the user message. Decoding is greedy with temperature 0. The prompt uses a 1 to 5 scale, which the evaluation harness maps to our 0 to 4 levels.

> You are an expert in Arabic linguistics and reading comprehension. Your task is to assess the readability level of a given Arabic text. Use the following 5-level scale: 1 = Very Easy (texts for early primary school, very simple vocabulary and short sentences) 2 = Easy (texts for upper primary school, common everyday vocabulary) 3 = Medium (texts for middle school, moderately complex vocabulary and sentence structure) 4 = Difficult (texts for high school or undergraduate level, advanced vocabulary and complex syntax) 5 = Very Difficult (specialized or academic texts, technical vocabulary, complex argumentation) Rules: Base your judgment on vocabulary complexity, sentence length and structure, and level of abstraction. Do NOT consider the topic or domain alone, focus on linguistic difficulty. Respond with ONLY a single integer: 1, 2, 3, 4 or 5. Nothing else.
>
> Text to assess:
>
> """ *[document]* """

Family	Feature	Description
Lexical (8)		
	`lex_n_tokens`	token count
	`lex_words_per_sent`	tokens per sentence
	`lex_avg_word_len`	mean token length in characters
	`lex_ttr`	distinct lemmas over tokens
	`lex_content_ttr`	distinct content lemmas over content tokens
	`lex_root_diversity`	distinct roots over tokens
	`lex_mean_log_freq`	mean log frequency of content lemmas
	`lex_rare_ratio`	share of content lemmas with corpus frequency below 50,000
Co-occurrence, volume (2)		
	`cooc_n_nodes`	number of content lemmas in the graph
	`cooc_n_edges`	number of co-occurrence links
Co-occurrence, topology (10)		
	`cooc_avg_degree`	mean number of links per lemma
	`cooc_max_degree`	degree of the most connected lemma
	`cooc_density`	realised links over possible links
	`cooc_avg_clustering`	mean clustering coefficient
	`cooc_n_components`	number of connected components
	`cooc_largest_cc_frac`	share of nodes in the largest component
	`cooc_avg_path`	mean shortest path length
	`cooc_diameter`	longest shortest path
	`cooc_degree_entropy`	entropy of the degree distribution
	`cooc_assortativity`	degree correlation between linked lemmas
Named entities (5)		
	`ner_n_entities`	number of distinct entities
	`ner_n_mentions`	number of entity mentions
	`ner_distinct_cats`	number of distinct Wojood categories
	`ner_entities_per_sent`	entities per sentence
	`ner_cat_entropy`	entropy over entity categories
Entity-relation graph (15)		
	`rel_n_relations`	number of extracted relations
	`rel_n_rel_types`	number of distinct relation types
	`rel_rel_entropy`	entropy over relation types
	`rel_n_nodes`	number of entity nodes
	`rel_n_edges`	number of relation edges
	`rel_avg_degree`	mean degree
	`rel_max_degree`	maximum degree
	`rel_density`	realised edges over possible edges
	`rel_avg_clustering`	mean clustering coefficient
	`rel_n_components`	number of connected components
	`rel_largest_cc_frac`	share of nodes in the largest component
	`rel_degree_entropy`	entropy of the degree distribution
	`rel_avg_path`	mean shortest path length
	`rel_diameter`	longest shortest path
	`rel_assortativity`	degree correlation between linked entities

Table 5: The 40 features, by family. The co-occurrence family is split into the two volume features and the ten topology features that Table 2 evaluates separately. The same topology measures are computed on the entity-relation graph, so the two graphs are compared under identical descriptors.

Co-occurrence window	QWK
1	$0.764_{\pm 0.004}$
2 (ours)	$0.756_{\pm 0.001}$
3	$0.748_{\pm 0.001}$
4	$0.740_{\pm 0.002}$
5	$0.745_{\pm 0.003}$

Table 6: Sensitivity of the topology configuration to the co-occurrence window, over five seeds per value. The total range is 0.024, which indicates that the method is robust to this parameter rather than dependent on fine tuning.

Dataset	Source
BAREC levels 1–4	Sentence-level release of the BAREC corpus (Elmadani et al., 2025b), used under its public release terms
Level 0	Passages extracted from a first-grade Arabic textbook of the Saudi national curriculum (Ministry of Education, Kingdom of Saudi Arabia, 2024)
Multi-source corpus (600 chunks per source)	Children's stories, detective stories, middle-school textbooks, high-school textbooks, Arabic Wikipedia, As-Safir newspaper articles and research manuscripts from the Arab Center for Research and Policy Studies.
Distillation corpus (6,000 chunks per source)	Hindawi Foundation open-access literature, school textbooks, Arabic Wikipedia, As-Safir newspaper articles and research manuscripts from the Arab Center for Research and Policy Studies.

Table 7: Provenance of the datasets used in this work.

Why LLM Aggregation Falls Short: Correlated Errors and Selection Gaps in Arabic Punctuation Restoration

Rana Alshaikh
Department of Information Systems
Faculty of Computing and Information Technology, King Abdulaziz University
Rabigh, 21589, Saudi Arabia
raalshaikh@kau.edu.sa

Majedah Alrehiely
Computer Science and Information Program Unit, Branch of the Applied College in Al-Ula, Taibah University
Madinah, Saudi Arabia
mmrehaili@taibahu.edu.sa

Abstract

Combining LLMs for Arabic punctuation restoration should help when their errors are complementary, but aggregation failure can reflect either missing coverage, where no predictor proposes the correct label, or failed selection, where the correct label is present but not selected. We investigate this distinction with vote-based, text-blind aggregators that use predictor labels, agreement, position, and neighbouring predictions, but not Arabic words. We evaluate three frontier LLMs and a fine-tuned AraBERT encoder on AraPunc-sub and ABC. The LLMs show strongly correlated errors: when all three are wrong, they choose the same incorrect label in 79.4% of cases on AraPunc-sub and 77.8% on ABC. Yet the four-predictor oracle achieves macro-F_1 scores of 0.857 and 0.778, surpassing the best standalone predictors (0.639 and 0.496). Leave-one-predictor-out analysis shows that AraBERT supplies the largest unique oracle coverage. Standard aggregation recovers only part of this headroom, and no tested aggregator outperforms standalone AraBERT on AraPunc-sub. A selectability probe confirms that some covered disagreements are learnable, but a deployable router offers only modest, corpus-dependent gains. Even a text-aware peer-deliberation step deepens rather than breaks the correlation, as weaker models conform to their peers. Hence, in the tested text-blind regime, the bottleneck is primarily selection rather than coverage, without ruling out text-aware selectors.

1 Introduction

Punctuation restoration is the process of restoring missing punctuation to improve text readability. The Arabic text can be ambiguous when diacritics are absent, since a single written form can correspond to multiple analyses; as a result, punctuation and context play an important role in text disambiguation (Habash, 2010). Arabic punctuation restoration (APR) can be more challenging as the punctuation rules are applied differently, and these differences become more apparent across genres (Yagi et al., 2024b). The use of frontier large language models (LLMs) in a zero-shot setting for punctuation restoration showed competitive performance (Al Wazrah et al., 2025; Suen et al., 2025); therefore, it is logical to study the performance of combining them as a follow-up step. Model aggregation may reduce their errors if they are partly independent, and yield a system better than each individual model.

This paper asks whether that expectation holds and finds a more qualified answer than a simple ensemble win or loss. For token-level APR, where aggregation helps, it is unclear whether LLM errors are sufficiently complementary. Moreover, it is indefinite whether the predictor pool lacks a correct alternative (a *coverage problem*) or contains one that the aggregator fails to select (a *selection problem*) when an ensemble underperforms. These diagnoses have different engineering implications: the former calls for better predictors, while the latter calls for better selection. We therefore evaluate three frontier LLMs on two independently sourced Arabic corpora, add a structurally different encoder. Oracle and disagreement analyses were also used to distinguish missing coverage from failed selection. In this case, the oracle is non-deployable: the headroom available is estimated when the correct candidate label is present. Text-blind aggregation serves as a diagnostic control as it is not able to see the Arabic tokens, and improvements must come from the candidate-label structure alone. The results are mixed rather than uniformly negative: LLM-only voting adds little, global stackers recover only limited four-predictor oracle headroom, and disagreement-specialized selection helps in some cases but not consistently.

Our investigation is organized around four questions. **(Q1)** How correlated are the LLMs, and how

Proceedings of The Fourth Arabic Natural Language Processing Conference (ArabicNLP 2026), pages 522–537
October 28-29, 2026 ©2026 Association for Computational Linguistics

much oracle coverage does the committee contain? We measure inter-model error correlation directly, including what the models do when jointly wrong, and bound the committee's coverage with a per-slot oracle. **(Q2)** Does a structurally different predictor add coverage? We fine-tune an Arabic encoder **AraBERT** (Antoun et al., 2020) and use a leave-one-predictor-out oracle to measure each predictor's unique contribution. **(Q3)** Do standard aggregators (per-mark co-vote calibration, majority voting, class-balanced logistic and XGBoost stacking, with and without encoder features) improve over the strongest predictor available to them? **(Q4)** If standard aggregation leaves some coverage unused, can a selector recover it? We train a disagreement-specialized selector as a diagnostic. These questions mainly test whether aggregation fails because of a lack of candidate coverage or because it fails to select from available candidates. The results in the tested text-blind regime support the latter diagnosis. The pool often contains the correct label, but the tested aggregators recover only part of that headroom.

The contribution is diagnostic: we separate missing coverage from failed selection and show that, in this vote-based, text-blind setting, the main bottleneck is selection. We do not rule out input-aware selectors, few-shot prompting, fine-tuned LLMs, or logit-level committees.

2 Related Work

Punctuation restoration and Arabic. Automatic punctuation restoration has been studied as it directly affects readability, downstream parsing, and the usability of automatic speech recognition (ASR) output; foundational statistical approaches established it as a crucial step (Beeferman et al., 1998; Gravano et al., 2009). Neural sequence-labelling models subsequently improved the baseline on spoken-language transcripts (Tilk and Alumäe, 2015; Tilk and Alumäe, 2016; Courtland et al., 2020). AraPunc (Sakr and Torki, 2023) extended the task to Arabic with a token-level dataset and benchmark, on which a multilingual encoder reaches strong in-domain performance. Beyond fine-tuned encoders, Al Wazrah et al. (Al Wazrah et al., 2025) report LLMs competitive with AraBERT but non-dominant across marks. Across the extant literature, the main objective has been to build stronger predictors that work *individually*. However, the question of what happens when several predictors operate simultaneously and whether combining them could outperform the best predictor individually has received little attention. This distinction is important as the benefit of the ensemble depends not only on the quality of each predictor but also on whether their errors are complementary. We therefore evaluate ensemble behaviour across two Arabic corpora and include AraBERT (Antoun et al., 2020) as a structurally different, fine-tuned token classifier.

Ensemble Methods and coverage-selection gap. Classical ensemble theory states that the benefit of combining models depends on the diversity of their errors rather than on ensemble size or the quality of individual models. Previous research shows that committees improve only when member errors are decorrelated (Hansen and Salamon, 1990), diversity is the main constraint (Dietterich, 2000), high covariance between predictors reduces the gain (Krogh and Vedelsby, 1994), and stacking (Wolpert, 1992) has this limitation when correlating base predictions (Kuncheva and Whitaker, 2003). Recent LLM ensemble methods inherit this requirement but rarely test it. Self-consistency (Wang et al., 2023) and fusion methods such as LLM-Blender (Jiang et al., 2023) degrade on homogeneous pools. Model-difference studies (Wan et al., 2024) restate diversity in LLM terms. Multi-agent debate (Du et al., 2024; Liang et al., 2024) presupposes genuinely distinct beliefs rather than convergent priors from shared pretraining. Routing methods (Lu et al., 2024; Ong et al., 2025) avoid combination entirely by selecting a single model for each query, assuming the best output can be identified directly. What combination can achieve is bound by oracle analyses across ASR (Fiscus, 1997), machine translation (Matusov et al., 2006; Rosti et al., 2007), parsing (McClosky et al., 2006), and QA (Min et al., 2021). They consistently show that the correct answer is present in the pool far more often than voting recovers, but rarely separate whether this comes from missing coverage or failed selection. The complementary signal may be absent (a coverage problem) or present but not selected (a selection problem). Prior work often conflates these two cases, even though they require opposite solutions.

3 Methodology

3.1 Task Formulation

Punctuation restoration is formulated as a token-level sequence labelling task. Given an unpunctuated paragraph of word tokens $x_1, \ldots, x_N$, each token receives a label from $\mathcal{M} \cup \{O\}$, where $\mathcal{M} = \{\text{COMMA, SEMICOLON, PERIOD, COLON, QUESTION, EXCL}\}$ denotes the punctuation mark that follows the token and O indicates no punctuation. The primary evaluation metric is *macro-averaged* F_1 across the classes in the test split. This is important for AraPunc-sub, whose test split contains no exclamation marks. Because most tokens are labeled O, token-level accuracy may be misleading. Therefore, *micro*-F_1 over non-O tokens is also reported for individual predictors as a descriptive secondary analysis. *Macro*-F_1 serves as the primary metric for all system comparisons, bootstrap tests, and oracle analyses, while *micro*-F_1 is used solely to characterize individual predictors and is not used for comparative evaluation.

We define *coverage* as the percentage of slots (token positions where a mark may appear), in which at least one predictor proposes the correct (gold) label and the labels proposed for a slot form its *candidate set*. Instead of asking whether a deployable system can identify the correct label, coverage asks only whether the correct label is available somewhere in the candidate set proposed by the predictors. We use a *per-slot oracle* (also called a *candidate-set oracle*) as a non-deployable diagnostic, since it chooses the gold label when the candidate set contains it and is forced to be wrong otherwise; as a result, we report its macro-F_1 as *oracle macro*-F_1.

Two predictor pools are used across this study. The *LLM-only pool*, which contains the three LLMs. The *four-predictor pool*, which contains the three LLMs plus AraBERT. Two oracles are computed: the *LLM-only oracle* over the first pool, and the *four-predictor oracle* over the second. The drop in four-predictor oracle macro-F_1 when that predictor is removed is the predictor's *unique contribution*.

3.2 Corpora

Our evaluation includes two independently sourced Arabic corpora (full statistics in Appendix B, Tables 9–10). **AraPunc-sub** is a deterministic, cost-bounded sub-sample of the official AraPunc (Sakr and Torki, 2023) dev/test split (segmentation in Appendix B). We use the name *AraPunc-sub* for all reported scores and reserve *AraPunc* for the official benchmark. As a sub-sample is evaluated under a support-aware macro-F_1 convention, absolute scores are not comparable with full-benchmark results. All claims are relative comparisons among systems on the same sampled units. AraPunc-sub is comma-dominant, with heavy use of colons, reflecting the long, coordinated sentences typical of its classical/MSA register. **ABC** is the manually annotated gold-standard component of the Arabic Punctuation Dataset (Yagi et al., 2024a), a formal-register corpus of non-fiction book chapters, split as 10dev / 5test documents and segmented as presented in Appendix B. ABC is period-dominant and roughly twice as densely punctuated as AraPunc-sub (17.4 vs. 9.5 marks per 100 tokens), consistent with the shorter declarative sentences of formal non-fiction.

3.3 Predictors

In the zero-shot setting at temperature 0, we instruct each of the three LLMs, **GPT-5.5**, **Claude Sonnet 4.6**, and **Gemini 2.5 Flash**, to return the input words with one punctuation label per word. We collect their predictions and apply a sequence-matching soft-alignment procedure to both corpora before evaluation to address minor differences in word count or formatting. Only segments where all three LLM outputs are parseable and align with the input are included in the evaluation.

We use a single shared prompt across all models, since our goal is to test whether the model families make complementary errors, not to optimize prompting. As a result, the correlation estimates reflect properties of this matched zero-shot committee (prompt and control justification in Appendix G, H). As a fourth predictor, we fine-tune AraBERT as a token classifier using split-disjoint training segments from the complement of each corpus's split. Hyperparameters are selected per corpus on a held-out validation split (Appendix H). AraBERT is the only non-LLM predictor in the four-predictor pool. We use it in the oracle, complementarity, and stacking analyses (Section 4).

3.4 Aggregation and Post-Processing

We evaluate several discrete-label aggregation methods, each built around a different assumption about how predictors should be combined (full specifications in Appendix H). The first, which we

refer to as Stage A, is per-mark co-vote calibration. It learns an integer threshold $k_m \in \{1, 2, 3\}$ for each mark, and keeps a predicted mark only when at least k_m of the three LLMs agree on it. The second method, Stage B, comprises a family of per-slot multiclass stackers, including logistic regression and XGBoost. Features for these models include one-hot LLM predictions, agreement indicators, token position, and a local prediction window. Each stacker is trained with and without AraBERT features, as well as with or without the Stage A co-vote calibration (eight dev-selected configurations), from which one is selected based on performance on the development split. The third method employs AraBERT directly, both independently and as part of the predictor pool. Finally, we include parameter-free majority voting as a baseline, breaking ties toward O.

3.5 Diagnostic Analyses

Beyond aggregate scores, four diagnostics let us reason about *why* aggregation succeeds or fails.

Error correlation. We use three agreement measures to examine whether the three LLMs tend to make similar predictions: mean pairwise Cohen's κ, slot unanimity, which measures how often all three LLMs agree, and the Jaccard overlap of false-comma positions across model pairs. We limit the overlap analysis to false commas because these are the dominant shared errors and the only mark with enough false positives to enable reliable estimation of overlap. The heavily dominant O (no-punctuation) class inflates overall κ and unanimity. To address this, we also report error-conditioned agreement: κ restricted to non-O and gold-non-O slots, alongside the rate at which all three LLMs assign the same wrong label and base the correlation claim on these.

Per-slot oracle and leave-one-out. A per-slot oracle counts a slot as covered if any predictor in a named pool emits the gold label at that slot. We report oracle coverage and oracle macro-F_1 for two pools: the LLM-only pool and the four-predictor pool. The leave-one-out analysis removes one predictor at a time from the four-predictor pool and measures the resulting drop in oracle macro-F_1. A substantial drop indicates that the removed predictor emits the correct label at slots where the others did not. We refer to this as *complementary correct coverage* rather than error decorrelation, since we do not measure error correlation directly. One plausible explanation is that AraBERT's Arabic pretraining and fine-tuning make it more sensitive to local morphological cues that the zero-shot LLMs do not always convert into correct punctuation labels. Finally, we ask which punctuation marks contribute to the gap between oracle coverage and system performance by identifying *covered-but-missed* slots, where the gold label is present in the candidate set but not selected by any system.

Disagreement-focused selection (Q4). Q4 separates whether the correct label is present among the four predictions from whether it can be selected without gold labels. We test the former with a gold-label probe over *covered-disagreement* slots, where at least one predictor is correct and at least one is wrong; this is diagnostic only, asking whether the correct choice is learnable in principle. We test the latter with deployable, text-blind *router* over *contested* slots, where the four predictors do not all agree and which are identifiable without gold. The selector chooses among the candidate labels using the predictions alone, without reading the Arabic text. We also include a fixed baseline, the *trust-the-dissenting-encoder rule*, that selects the encoder when it is the sole dissenter, testing whether its complementary correct coverage from Q2, the largest among the predictors, remains useful without training a selector. Definitions are in Appendix E, and a separate text-aware peer-deliberation probe is in Appendix F.

Significance. All system comparisons use paired bootstrap resampling over evaluation segments (Koehn, 2004; Dror et al., 2018). We report the mean macro-F_1 difference together with a 95% bootstrap confidence interval (Bestgen, 2022) and a one-sided p-value, treating an effect as significant only when the interval excludes zero. Every comparison names its reference point: the best LLM, the best standalone predictor, or the best standard stacker.

4 Results

4.1 Individual Models and the Correlation Problem (Q1)

Table 1 shows that the best individual LLM, across both corpora, is modest; Gemini scores macro-F_1 0.506 on AraPunc-sub and 0.477 on ABC. The majority voting does not reliably improve performance, changing the score by at most a few

System	macro-F_1		micro-F_1(non-O)	
	AraPunc-sub	ABC	AraPunc-sub	ABC
GPT-5.5	0.503	0.444	**0.440**	0.515
Claude	0.400	0.388	0.412	0.473
Gemini 2.5	**0.506**	**0.477**	0.434	**0.540**
Majority vote	0.510	0.461	—	—
AraBERT	**0.639**	0.496	**0.609**	**0.632**

Table 1: Individual predictors and majority voting, by macro-F_1 and by micro-F_1 over non-O tokens. The best LLM is bold per corpus/metric.

thousandths ($+0.003$ on AraPunc-sub, -0.016 on ABC). The model ordering is identical on both corpora: Gemini $>$ GPT-5.5 $>$ Claude. AraBERT has the best performance overall, with 0.639 on AraPunc-sub and 0.496 on ABC; full per-model macro- and micro-F_1 scores appear in Table 1. AraBERT is fine-tuned on corpus-specific labels, so its standalone scores are not resource-equivalent to the zero-shot LLMs. Therefore, it is used as the non-LLM member of the four-predictor pool, not as a like-for-like "encoder vs. LLM" comparison.

Voting fails because the models do not make independent errors. Table 2 shows mean pairwise Cohen's κ of 0.705 on AraPunc-sub and 0.739 on ABC, with 88.7% and 89.8% of test slots receiving unanimous predictions. These values are inflated by the dominant O class; therefore, error-conditioned agreement is also reported. When at least one LLM predicts a mark, mean pairwise κ is 0.453 on AraPunc-sub and 0.500 on ABC; on gold-non-O slots, 0.617 and 0.631, respectively.

A clear evidence that the models fail *together*, is what happens when they are all wrong: among slots where all three LLMs are wrong, they assign the *same* wrong label on 79.4% of cases on AraPunc-sub (n=2,520 all-three-wrong slots) and 77.8% on ABC (n=1,733). In other words, around four in five joint failures are not three independent errors, but one shared error repeated three times. This is exactly the case where voting and agreement-based filtering do not correct the error; instead, they reinforce it. Consistent with this, when the models place a false comma, they tend to place it in the same position. False-comma overlap across model pairs reaches a Jaccard of 0.436 on AraPunc-sub and 0.457 on ABC. On ABC, that is about 19–23$\times$ the overlap expected under independent errors. This pattern is consistent with Arabic clause structure, where long sentences are chained by the coordinating particle *wa-* (and) and clause boundaries are often signalled by conjunction rather than by punctuation, so comma placement is genuinely underdetermined; the models default to marking coordination boundaries that the gold annotation leaves unmarked (Yagi et al., 2024b).

4.2 A Fine-Tuned Encoder Adds Complementary Coverage (Q2)

Despite the correlation, the candidate set often contains the gold label. The three LLMs cover 92.4% of test slots on AraPunc-sub and 92.2% on ABC; adding AraBERT raises overall coverage to 96.6% and 95.9%. Because the dominant O class inflates overall coverage, the more informative view is gold-non-O coverage: the four-predictor pool covers 83.3% of non-O AraPunc-sub slots and 82.9% of non-O ABC slots, compared with 73.3% and 63.0% for the three LLMs alone. Table 5 shows that AraBERT substantially increases non-O coverage on both corpora, although coverage remains limited for some uncommon marks, such as semicolons on ABC.

We use the oracle to determine whether the correct punctuation label is present anywhere in the candidate set, without considering whether a real system could select it. Oracle macro-F_1 provides the class-balanced view. It is also the basis for the presence-versus-selection gap, since it is comparable to the systems own macro-F_1. For aggregators that must select from the candidate labels, it acts as a ceiling. For our stackers, it is a diagnostic reference, since they could, in principle, output a label that is not in the candidate set. It is not a formal mathematical upper bound for every aggregator, and it is also less clean than the binary coverage numbers, because it uses a fixed forced-wrong fallback. All oracle values reported below are oracle macro-F_1 under the forced-wrong fallback.

The three-LLM oracle macro-F_1 reaches, in Table 3, 0.690 on AraPunc-sub and 0.642 on ABC, substantially higher than the best LLM (0.506, 0.477). Adding the fine-tuned encoder raises the four-predictor oracle macro-F_1 to 0.857 and 0.778, which leaves gaps of 0.218 and 0.282 over the best individual predictors. This raises the question: which predictor supplies the additional headroom?

The answers are presented in Table 4 through the leave-one-predictor-out oracle. It removes each predictor from the four-predictor pool and measures how much oracle macro-F_1 drops. Re-

Correlation measure	AraPunc-sub	ABC
3-LLM slot unanimity	88.7%	89.8% [.884, .911]
Mean pairwise κ (all)	0.705 [.691, .719]	0.739 [.709, .766]
κ, ≥ 1 LLM non-O	0.453	0.500
κ, gold non-O	0.617	0.631
Same wrong label | all 3 wrong	79.4%	77.8%
n all-3-wrong slots	2,520	1,733
False-comma Jaccard	0.436 [.409, .462]	0.457 [.418, .502]
Ratio vs. independence	16.6–18.1×	18.7–22.6×

Table 2: Inter-model error correlation on the test split. CIs are shown in brackets. The error-conditioned κ rows reduce the inflation caused by the dominant O class, and the same-wrong-label row shows that when all three LLMs fail they usually choose the same incorrect label.

	Oracle		Best real	
Corpus	**3-LLM**	**4-pred**	**stand.**	**LLM**
AraPunc-sub	0.690	**0.857**	0.639	0.506
ABC	0.642	**0.778**	0.496	0.477

Table 3: Oracle macro-F_1 (3-LLM and 4-predictor pools) vs. the best standalone predictor and best LLM.

moving AraBERT on AraPunc-sub drops the oracle by 0.089 (95% CI [.079, .099]), which is four times the contribution of any individual LLM ($\leq$ 0.022). Removing AraBERT on ABC drops it by 0.136 (95% CI [.111, .161]), which is again the largest single contribution by a wide margin, ahead of GPT-5.5 (0.039), Gemini (0.020), and Claude (0.007). AraBERT, on both corpora, is correct on many slots where all three LLMs are wrong. We therefore describe it as providing the most *complementary correct coverage*. This does not mean that the errors are globally decorrelated, as that would require direct AraBERT–LLM correlation measures, which are outside the scope of this paper. This contribution is unlikely to reflect noise: although a weak predictor diverging arbitrarily from the LLMs could inflate apparent coverage, the standalone scores in Table 1 indicate otherwise, on ABC, AraBERT contributes the most unique coverage while also scoring 0.496 standalone, comparable to the best LLM's 0.477. A system that is itself near the top of the pool is far from random, so its coverage is unlikely to be pure noise.

The Q2 result is narrower: AraBERT contributes more unique oracle macro-F_1 than any individual LLM on both corpora, and instead supports selecting ensemble members by both competence and error complementarity. Whether this added coverage is selectable at inference time is tested in Q3 and Q4.

Corpus	Predictor removed	Unique contrib.
AraPunc-sub	GPT-5.5	0.022 [.016, .028]
	Gemini	0.013 [.009, .019]
	Claude	0.013 [.010, .015]
	AraBERT	**0.089** [**.079**, **.099**]
ABC	GPT-5.5	0.039 [.013, .067]
	Gemini	0.020 [.009, .036]
	Claude	0.007 [.005, .010]
	AraBERT	**0.136** [**.111**, **.161**]

Table 4: Leave-one-predictor-out oracle: drop in four-predictor oracle macro-F_1 when each predictor is removed, with 95% bootstrap CI. Larger drops indicate greater complementary correct coverage. AraBERT is the largest unique contributor on both corpora.

	Pool	All	non-O	,	.	:	;
AraPunc-sub	LLMs	.924	.733	.638	.902	.782	.759
	4-pred	.966	.833	.788	.934	.844	.834
ABC	LLMs	.922	.630	.602	.563	.931	.346
	4-pred	.959	.829	.784	.853	.963	.346

Table 5: Candidate coverage by pool, overall, on gold-non-O slots, and by mark. Adding AraBERT substantially increases non-O coverage on both corpora; question marks are omitted for space (four-predictor coverage 0.91 on both corpora).

Robustness to the oracle fallback. When the oracle is recalculated with uncovered slots supplied by the best individual predictor, instead of the forced-wrong fallback, the four-predictor oracle only slightly shifts (AraPunc-sub 0.857 $\rightarrow$ 0.847, ABC 0.778 $\rightarrow$ 0.754). The leave-one-out ordering is left unchanged: AraBERT remains the largest unique contributor by 3–4× (Table 4).

4.3 Standard Aggregation Underuses the Coverage (Q3)

The coverage exists (Q1), and a complementary predictor supplies it (Q2), but the tested standard aggregators recover only a small part of it. Full

Corpus	System	Δ macro-F_1
AraPunc-sub	**AraBERT standalone**	**+0.133**
	Stacker (xgb, +AraBERT)	+0.063
	Majority vote	+0.003
	Stacker (logistic, raw)	−0.103
ABC	AraBERT standalone	+0.019
	Stacker (logistic, +AraBERT)	+0.036
	Majority vote	−0.016
	Stacker (logistic, raw)	−0.013
	Stacker (xgb, raw)	−0.033

Table 6: Selected aggregators vs. the best LLM (Gemini); Δ is the macro-F_1 difference. Each panel lists the systems discussed for that corpus. Significant gains on AraPunc-sub are confined to encoder-using systems, none surpassing standalone AraBERT; on ABC no interval excludes zero. Full configurations with CIs in Table 11, Appendix B.

per-system deltas against the best LLM (Gemini on both corpora) are reported in Table 6 and more details in Appendix B, C, Table 11, which show absolute scores and segment-level bootstrap intervals together. The question is whether these aggregators outperform the best predictor available to them, and we answer this by comparing them to two baselines: the best LLM and the best standalone predictor.

Aggregator versus the best LLM. No standard global aggregator significantly beats it on ABC. Majority voting is -0.016; the non-encoder stackers are at or below baseline (-0.013 logistic, -0.033 XGBoost); and the best encoder-feature stacker reaches only $+0.036$ (CI $[-.012, .097]$, p=0.090, spanning zero). On AraPunc-sub, majority voting is within noise ($+0.003$), and the only systems that significantly improve over the best LLM are those that use encoder information: AraBERT standalone ($+0.133$, CI $[.099, .165]$, $p < 0.001$) and the AraBERT-feature XGBoost stacker ($+0.063$, CI $[.041, .086]$, $p < 0.001$). LLM-only aggregation (both voting and the non-encoder stackers) does not improve on either corpus, and the non-encoder logistic stacker is significantly *worse* than the best LLM on AraPunc-sub.

Versus the best standalone predictor. This more rigorous comparison is uniformly negative on AraPunc-sub. The standalone encoder scores 0.639, and every tested aggregation underperforms it, including the best encoder-feature stacker (0.569, i.e. -0.070) and majority voting. None of the tested aggregators outperforms the standalone encoder on AraPunc-sub. On ABC, the encoder is weaker (0.496), and the best encoder-feature stacker edges slightly above it in point estimate. However, the difference is not significant on five documents.

The gap between coverage and selection is large and concrete. The four-predictor oracle macro-F_1 is 0.778 on ABC (Table 3), while the best deployable standard stacker reaches 0.513 ($+0.036$ over the best LLM; Tables 6 and 11). This leaves most of the 0.27 oracle gap unrealized. The same pattern holds on AraPunc-sub against an oracle of 0.857. We stress that this gap is a diagnostic reference, not a hard bound on the stackers. In principle, the stackers can provide a label that no predictor proposed (Table 8). The gap shows how much candidate coverage the tested selection-style aggregators leave unused. It is not an absolute ceiling they can never exceed. The natural question is whether this unused coverage can actually be selected, or whether the standard stacker fails because the signal cannot be recovered from the available features, which is addressed in Q4.

4.4 Where the Covered Errors Occur

A per-mark analysis of the covered-but-missed region (Appendix B, Table 12) shows that after normalizing by gold support, O is not the main bottleneck. Its missed-coverage rate is only 3.7% on AraPunc-sub and 2.2% on ABC. The remaining gap is concentrated in punctuation decisions: on AraPunc-sub, the main classes are semicolon, comma, and period, while on ABC, they are period, comma, and exclamation. This points to a corpus-dependent class profile rather than a single universal mark bottleneck.

4.5 Is the Unused Coverage Selectable? (Q4)

Q4 asks whether the oracle headroom left unused by standard aggregation can be recovered from the disagreement structure alone. A gold-label probe looks only at slots where the gold label is present in the candidate set but the predictors disagree. In that setting the candidate choice is partly learnable, improving by $+0.113$ macro-F_1 on AraPunc-sub and $+0.187$ on ABC over an all-slot selector (Table 7). The deployable router is more limited: it leaves unanimous four-predictor slots unchanged and sends only contested slots to a learned selector, giving only modest, corpus-dependent gains over the trust-the-dissenting-encoder rule, reaching 0.546 on AraPunc-sub and 0.536 on ABC. The gain is not significant on AraPunc-sub and signif-

Policy / probe	AraPunc-sub	ABC
Gold-label probe (diagnostic)		
Covered-disagreement	0.415	0.633
All slots	0.303	0.446
Deployable selection		
Trust-the-dissenting rule	0.535	0.476
Learned router	0.546	0.536
Δ (learned $-$ rule)	$+0.011$	$+0.061$

Table 7: Q4 diagnostics. The gold-label probe shows the candidate choice is partly learnable ($+0.112$/$+0.187$ over an all-slot selector).

icant on ABC but rests on only five source documents; full variants are in Appendix E. Thus some unused coverage is selectable, but the tested selector does not give a robust universal selection rule.

5 Discussion

Standard aggregation underuses available coverage. The results compose into a picture that is directionally consistent across both corpora, which differ in register and punctuation profile. The oracle sits far above every real system, so the candidate set *collectively* contains the correct answer on many slots it gets wrong (Q1–Q2), and a structurally different predictor supplies most of that complementary coverage as the single largest unique contributor on both corpora (Q2). Yet the standard aggregators we test capture little of it: no standard aggregator significantly beats the best LLM on ABC, on AraPunc-sub only encoder-using systems do, and none surpasses standalone AraBERT on AraPunc-sub (Q3–Q4). The bottleneck in this regime is therefore selection, not coverage: the correct label is usually present in the candidate set, but the specific discrete-label, text-blind methods we test do not realize it from the features they are given. We do not claim aggregation is impossible in principle.

The encoder's complementary coverage is not explained by standalone weakness. The leave-one-out analysis makes a practical point: the value of adding a predictor depends on the complementary correct coverage it supplies, not on its standalone score. Some coverage might still come from a different class bias rather than true complementarity, but on this pool a competent, genuinely complementary predictor looks more useful than another correlated LLM.

Some unused coverage is selectable, but not by one universal rule. Q4 shows the unused coverage is not simply unreachable: a gold-label probe recovers a meaningful part of the gap, deployable router improves on the trust-the-dissenting-encoder rule in point estimate, though the AraPunc-sub gain is nonsignificant and the policy is structure- and corpus-dependent (Appendix E). The residual is concentrated in punctuation decisions with a corpus-dependent class profile rather than a single universal mark bottleneck, reinforcing that selection should condition on the *structure* of disagreement. Because our aggregators use vote, agreement, and position features but not the input Arabic words, we make no claim about a text-aware aggregator; the one text-aware mechanism we did probe, peer deliberation (Appendix F), does not break the correlation, as models overwhelmingly conform rather than re-reason independently.

6 Conclusion

We examined why aggregation yields limited gains for APR with three frontier LLMs and a fine-tuned AraBERT encoder. Across AraPunc-sub, with ABC showing the same qualitative pattern, the LLMs exhibit strongly correlated errors: when they jointly fail, they usually choose the same wrong label. Majority voting therefore has little independent evidence to exploit. At the same time, oracle analysis shows that the four-predictor pool often contains the correct label in its candidate set, and leave-one-predictor-out analysis shows that AraBERT supplies the largest unique contribution on both corpora.

The resulting diagnosis is that, in this discrete-label, vote-based, text-blind regime, the main bottleneck is selection rather than coverage. Standard aggregators recover only part of the available oracle headroom, and no tested aggregator surpasses standalone AraBERT on AraPunc-sub. Using gold labels, we find that some covered disagreements are learnable in principle, but the deployable router gives only modest, corpus-dependent gains rather than a universal selection rule.

The practical implication is therefore not simply to add more correlated LLMs, but to build committees with genuinely complementary predictors and selectors that can condition on the structure of disagreement. A finer, genre- and discourse-level account of the shared errors is beyond our slot-level analysis and remains an open direction.

7 Limitations

The study has five scope limitations. First, our main quantitative evaluation uses a deterministic sub-sample of the official AraPunc split rather than the full benchmark. We preserve split membership and the underlying token–label stream and sub-sample only for cost; this supports controlled internal comparison, although the sub-sample may not exactly match the full split's distribution, and evaluating a sub-sample under a support-aware macro-F_1 convention means absolute scores remain non-comparable with full-benchmark results. It does not affect the internal comparisons among predictors and aggregators, which all use the same units and labels, but it remains unknown whether the measured correlation, candidate coverage, and selection patterns transfer unchanged to the official split. We therefore refer consistently to *AraPunc-sub*, avoid state-of-the-art comparisons, and release the sampled-segment identifiers, deterministic segmentation procedure, labels, and system predictions needed to reproduce the analysis.

Second, all LLM predictions use one shared zero-shot prompt. This design controls for prompt and demonstration differences, and let us study model-family variation under a common task specification, but it does not establish that the same error correlation would persist under few-shot prompting, model-specific prompt optimization, or deliberately diversified demonstrations. Such interventions may change both individual accuracy and ensemble diversity; few-shot prompting improved Arabic punctuation prediction in prior work (Al Wazrah et al., 2025), and could in principle reduce the error correlation we observe. Testing whether the correlation persists under few-shot prompting is therefore a natural next step. Our conclusions are accordingly scoped to the matched zero-shot committee evaluated here, which we view as a controlled baseline rather than the full picture of LLM aggregation for this task.

Third, the aggregation and routing methods are text-blind. They receive candidate labels, vote and agreement patterns, position, and local prediction context, but not the Arabic input tokens. This restriction is deliberate in the diagnostic analysis because it isolates whether disagreement structure alone contains sufficient information for candidate selection. It also means that our results do not show that the oracle headroom is inaccessible to a text-conditioned selector. An input-aware router could use linguistic context to distinguish otherwise identical vote patterns and may recover additional coverage; that broader practical question is not tested here.

Fourth, ABC's 93 test segments make it comparable in scale to AraPunc-sub and a genuine second corpus for every analysis we report; the qualitative findings, correlated errors, substantial oracle headroom, and AraBERT's large unique contribution, hold on both. Its one limitation is statistical: the 93 segments derive from only five source documents, and because segments within a document are not independent, segment-level resampling does not increase the effective sample size beyond roughly five. We therefore rely on ABC's wide-margin and qualitative results in full, and read only its narrow, close-to-zero significance margins with caution, deferring to AraPunc-sub where a verdict hinges on them.

Moreover, because both corpora derive from published texts, their punctuated originals may appear in the pretraining data of the LLMs. We cannot rule this out for closed models, and contamination could inflate both the models' accuracy and their agreement. We note that our conclusions are relative comparisons among predictors on the same units, and that contamination would, if anything, reinforce rather than weaken the correlated-error finding we report.

Fifth, our pool includes only one fine-tuned Arabic encoder. Because we never pair AraBERT with a second, independently trained encoder, we cannot say whether its complementary coverage reflects something general about the encoder–LLM difference or something particular to AraBERT itself. Testing a more diverse set of Arabic encoders would help separate the two, and we leave this for future work.

References

Asma Ali Al Wazrah, Afrah Altamimi, Hawra Aljasim, Waad Alshammari, Rawan Al-Matham, Omar Elnashar, Mohamed Amin, and Abdulrahman AlOsaimy. 2025. Evaluation of large language models on Arabic punctuation prediction. In *Proceedings of the 1st Workshop on NLP for Languages Using Arabic Script*, pages 144–154, Abu Dhabi, UAE. 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 Pro-*

cessing Tools, with a Shared Task on Offensive Language Detection, pages 9–15, Marseille, France. European Language Resources Association.

D. Beeferman, A. Berger, and J. Lafferty. 1998. Cyberpunc: a lightweight punctuation annotation system for speech. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '98 (Cat. No.98CH36181)*, volume 2, pages 689–692 vol.2.

Yves Bestgen. 2022. Please, don't forget the difference and the confidence interval when seeking for the state-of-the-art status. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5956–5962, Marseille, France. European Language Resources Association.

Maury Courtland, Adam Faulkner, and Gayle McElvain. 2020. Efficient automatic punctuation restoration using bidirectional transformers with robust inference. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 272–279, Online. Association for Computational Linguistics.

Thomas G. Dietterich. 2000. Ensemble methods in machine learning. In *Multiple Classifier Systems*, pages 1–15, Berlin, Heidelberg. Springer Berlin Heidelberg.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392, Melbourne, Australia. Association for Computational Linguistics.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2024. Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 11733–11763. PMLR.

J.G. Fiscus. 1997. A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover). In *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pages 347–354.

Agustin Gravano, Martin Jansche, and Michiel Bacchiani. 2009. Restoring punctuation and capitalization in transcribed speech. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4741–4744.

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

L.K. Hansen and P. Salamon. 1990. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1001.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023. LLM-blender: Ensembling large language models with pairwise ranking and generative fusion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14165–14178, Toronto, Canada. Association for Computational Linguistics.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Anders Krogh and Jesper Vedelsby. 1994. Neural network ensembles, cross validation, and active learning. In *Advances in Neural Information Processing Systems*, volume 7. MIT Press.

Ludmila I. Kuncheva and Christopher J. Whitaker. 2003. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Mach. Learn.*, 51(2):181–207.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. 2024. Encouraging divergent thinking in large language models through multi-agent debate. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17889–17904, Miami, Florida, USA. Association for Computational Linguistics.

Keming Lu, Hongyi Yuan, Runji Lin, Junyang Lin, Zheng Yuan, Chang Zhou, and Jingren Zhou. 2024. Routing to the expert: Efficient reward-guided ensemble of large language models. 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 1964–1974.

Evgeny Matusov, Nicola Ueffing, and Hermann Ney. 2006. Computing consensus translation for multiple machine translation systems using enhanced hypothesis alignment. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 33–40, Trento, Italy. Association for Computational Linguistics.

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective self-training for parsing. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 152–159, New York City, USA. Association for Computational Linguistics.

Sewon Min, Kenton Lee, Ming-Wei Chang, Kristina Toutanova, and Hannaneh Hajishirzi. 2021. Joint passage ranking for diverse multi-answer retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6997–7008, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M Waleed Kadous, and Ion Stoica. 2025. RouteLLM: Learning to route LLMs from preference data. In *The Thirteenth International Conference on Learning Representations*.

Antti-Veikko Rosti, Necip Fazil Ayan, Bing Xiang, Spyros Matsoukas, Richard Schwartz, and Bonnie Dorr. 2007. Combining outputs from multiple machine translation systems. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 228–235, Rochester, New York. Association for Computational Linguistics.

Abdelrahman Sakr and Marwan Torki. 2023. AraPunc: Arabic Punctuation Restoration Using Transformers. In *2023 20th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA)*, pages 1–6. IEEE.

King Yiu Suen, Rudolf Chow, and Albert Y. S. Lam. 2025. Cantonese punctuation restoration using LLM annotated data. In *Proc. Interspeech 2025*, pages 1953–1957.

Ottokar Tilk and Tanel Alumäe. 2015. LSTM for punctuation restoration in speech transcripts. In *Interspeech 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015*, pages 683–687. ISCA.

Ottokar Tilk and Tanel Alumäe. 2016. Bidirectional recurrent neural network with attention mechanism for punctuation restoration. In *Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech 2016)*, pages 3047–3051, San Francisco, CA, USA.

Fanqi Wan, Xinting Huang, Deng Cai, Xiaojun Quan, Wei Bi, and Shuming Shi. 2024. Knowledge fusion of large language models. In *International Conference on Learning Representations (ICLR)*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

David H. Wolpert. 1992. Stacked generalization. *Neural Networks*, 5(2):241–259.

Sane Yagi, Ashraf Elnagar, and Esra Yaghi. 2024a. Arabic punctuation dataset. *Data in Brief*, 53:110118.

Sane Yagi, Shehdeh Fareh, Ashraf Elnagar, Mariam Balajeed, Abdalla El-mneizel, and Mohammad Al-Badawi. 2024b. Is arabic punctuation rule-governed? *Cogent Arts & Humanities*, 11(1):2303818.

Quantity	Definition / use
Macro-F_1	F_1 averaged equally over punctuation classes with nonzero gold support in the evaluated split. Primary metric; O is excluded.
Micro-F_1 (non-O)	F_1 pooled over all non-O tokens. Secondary metric; reflects performance on frequent marks.
Per-slot oracle (coverage)	Fraction of slots at which *at least one* predictor in a named pool emits the gold label. This binary quantity is distinct from oracle macro-F_1.
Candidate-set oracle (oracle macro-F_1)	Macro-F_1 of a non-deployable system that emits the gold label on covered slots and is forced to a fixed wrong label (O, no punctuation) on uncovered slots, so each uncovered slot counts as a miss. Since macro-F_1 excludes the O class, forcing O registers as a miss on the true mark without adding a false positive to any other class. Because the forced-wrong label is a fixed choice, the absolute value depends on that choice and the metric is a diagnostic reference, not a clean upper bound; it is a strict ceiling only for aggregators constrained to *select* from the candidate labels, not for unconstrained multiclass stackers that can emit a label no predictor proposed.
LLM-only vs. four-predictor pool	The *LLM-only pool* contains the three LLMs. The *four-predictor pool* contains the three LLMs plus AraBERT. The LLM-only oracle shows what the LLM committee covers; the four-predictor oracle shows what the full predictor pool covers.
Unique contribution	Drop in the four-predictor oracle macro-F_1 when one predictor is removed; measures complementary correct coverage.
Δ macro-F_1	Difference vs. a named baseline (best LLM, best standalone, or best standard stacker), with bootstrap 95% CI (B=1000) over evaluation segments and one-sided p. Significant only if the CI excludes zero.

Table 8: Evaluation metrics and diagnostic quantities used in the paper.

A Evaluation Metrics and Diagnostic Quantities

Table 8 defines the evaluation metrics and oracle quantities used throughout the paper.

B Corpus Details and Statistics

AraPunc (Sakr and Torki, 2023) is distributed as a single stream in which each token is paired with its label, with no document partition. For both corpora we form evaluation segments by deterministic sentence-boundary segmentation with a maximum of 512 word tokens per segment, and sample 100 dev and 100 test segments (33,048 test to-

Property	AraPunc-sub	ABC
Register	classical	formal
Dominant mark	comma	period
Marks per 100 tokens	9.5	17.4
AraBERT (fine-tune, test F_1)	0.639	0.496
Dev segments (resampling units)	100	100
Test segments (resampling units)	100	93
Source documents (dev/test)	1/1 stream	10/5 docs
Raw test word tokens	33,048	23,884
Jointly evaluated tokens	33,048	22,176

Table 9: Corpus statistics. AraPunc-sub denotes our deterministic sub-sample of the official AraPunc split. It yields 100 resampling segments; ABC is segmented the same way and yields 93 test segments from 5 source documents.

Mark	AraPunc-sub	ABC
Comma	49.4%	36.7%
Period	17.9%	41.3%
Colon	24.9%	15.9%
Semicolon	6.4%	1.4%
Question	1.4%	2.9%
Exclamation	0.0%	1.8%

Table 10: Gold punctuation distribution (share of all non-O marks on the test split); each column is computed directly from our split and sums to 100%.

kens), to make three-frontier-LLM inference possible. For ABC, alignment uses sequence matching to recover labels when a model's output length drifts from the input; after alignment, 93 test segments and 22,176 word slots are jointly valid across the three LLMs. Table 9 reports corpus-level statistics, and Table 10 gives the gold punctuation distribution; the two corpora differ in register, dominant mark, and punctuation density.

C Full System Comparison (Q3)

Table 11 gives the absolute macro-F_1 and the bootstrap Δ versus the best LLM (Gemini on both corpora) for every system summarized in Table 6, together with 95% confidence intervals. On ABC, all intervals include zero. On AraPunc-sub, significant gains over the best LLM are confined to systems using encoder information, and no tested aggregator surpasses standalone AraBERT.

D Per-Mark Covered-Error Analysis

Table 12 decomposes the covered-but-missed region by gold mark under two references. The raw-share columns use majority voting as the baseline and therefore track class frequency; the class-conditional columns use the best LLM and normalize by the gold support of each mark. The two views use different baselines and should not be read as a single additive decomposition. After normalising by support, O is not the bottleneck (missed-coverage 3.7% on AraPunc-sub, 2.2% on ABC); the residual is concentrated in punctuation decisions.

E Disagreement-Routed Selection (Q4 Detail)

This appendix extends the Q4 summary in Section 4.5; the headline probe and router scores are in Table 7. The gold-label probe is a logistic classifier trained on the covered-disagreement subset (slots where at least one predictor is correct and at least one is wrong), with gold labels used only to define that subset; its gains over the same classifier trained on all slots establish that the candidate choice is partly learnable given a covered disagreement. Because gold defines the subset, it is diagnostic, not deployable. The router and the gold-label probe differ in which slots they act on. The router operates on *contested* slots, where the four predictors do not all agree; these are identifiable at test time from the predictions alone, with no gold, so the router is deployable. The gold-label probe operates on the narrower set of *covered-disagreement* slots, where the predictors disagree *and* the gold label is present in the candidate set; identifying this second condition requires the gold label, so the probe is diagnostic rather than deployable. Both learn a selector over the same features: predictor labels, agreement pattern, position, and the local prediction window. They differ only in the slots they act on. The deployable router passes unanimous slots through and routes contested slots to the learned selector, using no gold at test time. Its gain over the trust-the-dissenting-encoder rule is not significant on AraPunc-sub (bootstrap CI $[-.016, .036]$, p=0.188) and positive on ABC (CI $[.036, .105]$), but the ABC result rests on only five source documents, whose per-document deltas flip sign and average near zero, so we treat it as descriptive rather than conclusive.

The mechanism is corpus-dependent. On slots where the three LLMs agree and AraBERT dissents, AraBERT is correct 65.8% of the time on AraPunc-sub versus 23.3% for the unanimous LLM label, but only 43.6% versus 46.3% on ABC. AraBERT is thus a reliable dissenter on AraPunc-sub but close to a coin flip on ABC, so "al-

Corpus	System	Abs. macro-F_1	Δ vs. best LLM (95% CI)
AraPunc-sub	**AraBERT standalone**	0.639	$+\mathbf{0.133}$ $[\mathbf{.099}, \mathbf{.165}]$
	Stacker (xgb, +AraBERT, cal)	0.569	$+\mathbf{0.063}$ $[\mathbf{.041}, \mathbf{.086}]$
	Stacker (xgb, +AraBERT, raw)	0.569	$+0.063$ $[.044, .083]$
	Majority vote	0.510	$+0.003$
	Stacker (logistic, raw)	0.403	-0.103 $[-.129, -.078]$
ABC	Stacker (logistic, +AraBERT)	0.513	$+0.036$ $[-.012, .097]$
	AraBERT standalone	0.496	$+0.019$ $[-.032, .101]$
	Stacker (xgb, +AraBERT)	0.504	$+0.027$ $[-.007, .089]$
	Stacker (logistic, raw)	0.464	-0.013 $[-.037, .026]$
	Stacker (xgb, raw)	0.444	-0.033 $[-.060, .003]$

Table 11: Selected standard systems compared with the best LLM, Gemini on both corpora. On ABC, all intervals include zero. On AraPunc-sub, significant gains over the best LLM are confined to systems using encoder information, and no tested aggregator surpasses standalone AraBERT.

	Raw share vs. majority vote		Class-cond. rate vs. best LLM	
Gold mark	**AraPunc-sub**	**ABC**	**AraPunc-sub**	**ABC**
O (no punctuation)	68.5%	14.6%	3.7%	2.2%
Comma	18.7%	37.8%	19.1%	33.1%
Period	2.3%	38.9%	15.5%	40.2%
Colon	6.9%	5.1%	10.2%	12.7%
Semicolon	3.6%	0.6%	30.7%	13.5%
Question mark	0.1%	1.5%	0.0%	14.2%
Exclamation	—	1.4%	—	30.1%

Table 12: Two per-mark views of oracle-covered errors. Raw shares use majority voting as the reference and therefore reflect class frequency. Class-conditional rates use the best LLM as the reference and normalize by the number of gold slots of each mark. The different baselines are shown explicitly; these columns should not be read as a single additive decomposition.

ways trust the dissenting encoder" is a good policy on one corpus and a poor one on the other. A development-selected hybrid reaches 0.556 on AraPunc-sub, the best point estimate among the variants, yet still trails the encoder-feature stacker (≈ 0.569) and standalone AraBERT (0.639).

F Peer-Deliberation Probe

The main-text aggregators are text-blind. We conduct a peer-deliberation round to check whether a text-aware step, in which a model re-reads the Arabic input instead of just manipulating discrete votes, evades the correlation. Following the zero-shot pass, each model is re-prompted with the same paragraph and its peers' predictions on the slots on which the models disagree, and asked to rework its own labels. This is a multi-agent reconsideration step, *not* an aggregator; it yields a new set of per-model predictions. For each model we report its conformity rate (on exposed disagreement slots, the proportion of times the model changes its own label to a peer's) and the difference in macro-F_1 between its zero-shot and deliberated predictions (Table 13).A high conformity rate means that exposure to disagreement leads to convergence rather than independent re-reasoning, which can reinforce the error correlation found in Q1. GPT-5.5 conforms on almost every one of the disputed slots, while Claude's larger-magnitude change is likely because it is a weaker model being nudged toward stronger models, rather than because of an independent improvement. In contrast, the best individual LLM, Gemini, conforms less than half of the time, and its overall performance remains essentially flat.

	Conformity rate		macro-F_1 (ZS→peer)	
Model	**AraPunc-sub**	**ABC**	**AraPunc-sub**	**ABC**
GPT-5.5	0.98	1.00	0.503→0.511	0.444→0.459
Claude Sonnet 4.6	0.87	0.91	0.400→0.491	0.388→0.459
Gemini 2.5 Flash	0.38	0.44	0.506→0.519	0.477→0.476

Table 13: Peer-deliberation probe. *Conformity rate* is the fraction of exposed disagreement slots on which a model abandons its own label and switches to a peer's. *macro-F_1 (ZS→peer)* gives the standalone score before (zero-shot) and after one peer-deliberation round. Exposure to disagreement induces *conformity*, not independent re-reasoning, deepening rather than breaking the error correlation of Q1.

G Prompt Template

All three LLMs receive the same single Arabic user message (no separate system role, for cross-provider consistency), where `{n_words}` is replaced by the number of words in the paragraph. The instruction sets the allowed six-mark inventory; permits no changing, deleting, adding, reordering, respelling, or diacritizing of any word; and specifies that the model should leave a position unmarked if it finds no clear boundary (a bias against over-prediction). For readability, the prompt shown below is an English translation; all experiments use the original Arabic version.

> *"You are an expert in Arabic punctuation restoration. Rewrite the following Arabic text with exactly the same words, adding only appropriate punctuation. Do not change, delete, add, or reorder any word; do not correct spelling or add diacritics; the word count must stay exactly {n_words}. Allowed marks only: comma, semicolon, period, colon, question, exclamation. Most words take no mark; do not over-punctuate; if there is no clear pause, leave the position unmarked. Use the colon only when the preceding word directly introduces an explanation or quotation. Do not use Latin punctuation; do not output JSON; do not explain. Return only the punctuated text."*

H Reproducibility

Models and decoding. We query three frontier LLMs through their providers: `gpt-5.5` (OpenAI), `claude-sonnet-4-6` (Anthropic), and `google/gemini-2.5-flash` (Google). All calls use temperature 0 with an 8000-token limit, and Claude's extended thinking is disabled. We do not claim bit-exact determinism, since providers do not guarantee identical outputs across backends; the run manifest records the exact version strings.

Data. For ABC we use 46 files in total. A curated stratified split of 15 files (10 for development and 5 for testing) forms the evaluation set, 30 disjoint files make up the encoder training pool, and one file is held in reserve. We release the derived labels, unit identifiers, scripts, and system predictions, but not the source text, which may remain under copyright. The split separates documents but not authors; since punctuation style is partly author-dependent, an author-disjoint split would be preferable and is left to future work. AraPunc-sub is the 100-development, 100-test segment sample described in Section 3.

Corpus	Model	Excl. rate (dev / test)
ABC	GPT-5.5	0% / 0%
	Claude	1% / 2%
	Gemini	10% / 6%
AraPunc-sub	GPT-5.5	1% / 0%
	Claude	1% / 0%
	Gemini	1% / 0%

Table 14: Per-model segment exclusion rate on each corpus, before the sequence-matching recovery pass. A segment is excluded for a model when its output does not align one-to-one with the input by word count. Rates are low overall and, on ABC, concentrated in a single model; the recovery pass resolves nearly all cases, leaving the jointly evaluated set at 93 ABC test segments and the full 100 on AraPunc-sub.

Preprocessing. We label the six focus marks together with O, strip decorative marks such as ellipses, guillemets, and dashes, and normalize the Arabic text. Evaluation segments are produced by a deterministic segmenter (seed 20260514) that splits at sentence boundaries and caps each segment at 512 word tokens.

Alignment. We keep a model's output for a segment only when its words and word count match the input exactly, so that every system is scored on the same set of aligned slots. When a model's output drifts slightly in length, a sequence-matching pass recovers the labels wherever the alignment allows, and the few segments that cannot be recovered are excluded. Word preservation before this pass was high on both corpora: on ABC it was 100% for GPT-5.5, 98% for Claude, and 94% for Gemini, and on AraPunc-sub all three models exceeded 99%.

Table 14 reports the corresponding per-model exclusion rates before the recovery pass. The rates are low across the board and, on ABC, are driven almost entirely by a single model, while the other two are at or near zero; on AraPunc-sub all three models stay at or below one percent. The recovery pass resolves nearly all of these cases, so the set we actually evaluate on, segments where all three LLM outputs parse and align, remains essentially the full test set: 93 segments on ABC and all 100 on AraPunc-sub. Because AraPunc-sub retains its complete test set, no segment is dropped there.

Encoder fine-tuning. We fine-tune AraBERT (`aubmindlab/bert-base-arabertv02`) as a token classifier with a batch size of 16, a maximum sequence length of 512, a warmup ratio of 0.10,

and weight decay of 0.01. For each corpus we select the learning rate and number of epochs on a 10% held-out split, searching over learning rates of 2, 3, and 5×10^{-5} and over 10 or 20 epochs, with three seeds per setting and early stopping. The selected encoder reaches 0.639 on AraPunc-sub and is stable at 0.646 ± 0.013 across three seeds. On ABC, the AraPunc default recipe underfits, so we use the validation-selected configuration (5×10^{-5}, 20 epochs), which reaches 0.4955 and is stable at 0.496 ± 0.025 across three seeds. Training uses the disjoint complement of each corpus's split; the ABC training pool is 471 segments ($112{,}894$ tokens).

Aggregators. We choose the co-vote thresholds and select among the stackers on the development set by macro-F_1, using document-grouped cross-validation. The XGBoost stacker uses a maximum depth of 4, 100 estimators, a minimum child weight of 10, a learning rate of 0.1, and subsampling of 0.9 for both rows and columns.

Significance. All comparisons use a paired bootstrap over evaluation segments with 1000 resamples (seed 13), from which we report the mean macro-F_1 difference, a 95% percentile interval, and a one-sided p-value, calling an effect significant only when the interval excludes zero. Because ABC's 93 test segments come from only five documents and are not independent within a document, its effective sample size is much smaller than 93, so we read its significance results with caution.

I Prompt-Sensitivity of the Error Correlation

A natural concern is that the inter-model error correlation reported in Section 4.1 arises from using a single shared prompt rather than from the models themselves. Assigning a different prompt to each model does not provide a clean test, since it requires first identifying the best prompt per model and thereby conflates prompt quality with error decorrelation. We instead hold the prompt shared *within* each condition and vary it *across* conditions, so that any change in correlation is attributable to the prompt alone.

We evaluate the three-model committee under four prompts, applied identically to all models on a common set of AraPunc-sub test segments. The prompts differ only in their punctuation guidance, ranging from our original prompt to one that removes punctuation guidance entirely. If the correlation were an artifact of our wording, it should weaken as the shared guidance is removed.

Prompt	Mean pairwise κ
Original (mark-averse)	0.700 [.684,.717]
Neutral	0.728 [.706,.750]
Boundary-marking	0.728 [.703,.752]
Minimal (no guidance)	0.724 [.699,.749]
Main study	0.705

Table 15: Mean pairwise Cohen's κ under four prompts applied identically to all three models on a common AraPunc-sub segment set, with 95% bootstrap intervals. The correlation is statistically indistinguishable across prompts and consistent with the main study.

It does not. Mean pairwise Cohen's κ is statistically indistinguishable across all four prompts (0.700–0.728; overlapping 95% bootstrap intervals) and matches the value from the main study (0.705). The other correlation signals are equally stable across the four prompts: false-comma overlap remains 16–21$\times$ higher than expected under independent errors, and on slots where all three models are wrong they select the same incorrect label in 79–83$\%$ of cases, consistent with the main-study values (Table 2). Removing the guidance entirely does not reduce the correlation. The correlation is therefore a property of the model committee rather than an artifact of the shared prompt, and we scope the correlation claim to the matched zero-shot committee accordingly.

J Qualitative Examples

To make the coverage-versus-selection distinction concrete, we inspect individual test slots on AraPunc-sub (non-O slots). For each example, we show the gold-punctuated clause with the target word in brackets, followed by the mark assigned by each predictor ($\varnothing$ denotes no mark). Two patterns recur. In 340 AraPunc-sub slots, all three LLMs are wrong while AraBERT alone recovers the gold mark. In another 498 slots, no model predicts the gold mark, indicating a true coverage gap. Overall, the three LLMs agree on the same wrong label in 737 slots.

Gold-punctuated context (target in [brackets])	Gold	GPT-5.5	Claude	Gemini	AraBERT
(i) The three LLMs agree on a wrong mark; AraBERT recovers it					
... نص [عليه]، كسائر قتل العمد	،	∅	∅	∅	،
... ان يوصي [به]، فان لم يوص به	،	.	.	.	،
(ii) Missing coverage — no predictor proposes the gold mark					
... لعدم وجوبه علي [الميت]. وسياتي بسط	.	،	،	،	،

Table 16: Qualitative examples from the AraPunc-sub test set. In case (i), the three LLMs either omit the mark or interpret a clause boundary as a period, while AraBERT recovers the gold comma. In case (ii), every predictor interprets the gold sentence boundary as a comma; therefore, the correct period is absent from the candidate set.

Child ASR Adaptation with Adult Retention: An Empirical Study

Houssam Eddine-Othman Lachemat, Shammur Absar Chowdhury
Qatar Computing Research Institute,
{hlachemat, shchowdhury}@hbku.edu.qa

Abstract

Automatic Speech Recognition (ASR) systems often underperform for children and non-native speakers, while adapting adult ASR models to child speech can cause adult-speech forgetting. We study *child ASR adaptation with adult retention* across Arabic and English. We compare full fine-tuning, LoRA, and post-hoc weight-space merging across encoder–decoder, encoder–CTC, and AudioLLM-based ASR systems. Experiments use Arabic native and non-native child speech, English MyST child speech, and adult benchmarks from MGB-2 and LibriSpeech test-clean. We evaluate recognition quality with WER and quantify the adaptation–retention trade-off using Retention Index, Child Adaptation Gain, and Adaptation Recovery. Results show that child adaptation is necessary, especially for non-native Arabic and English child speech, but direct adaptation often reduces adult ASR performance. Bilingual adaptation is more stable than language-specific adaptation. Weight-space merging often improves the trade-off, especially for encoder–CTC, Whisper, and AudioLLM-based ASR, with LERP favoring adult retention and TIES recovering stronger child gains. For the encoder–decoder model, direct bilingual fine-tuning remains strongest in raw WER.[1]

1 Introduction

Automatic Speech Recognition (ASR) has achieved strong performance on adult speech, but children's speech remains difficult to recognize. Compared with adults, children often have higher pitch, shorter vocal tracts, greater pronunciation variability, and more disfluencies, all of which can degrade ASR accuracy (Wu et al., 2019; Bhardwaj et al., 2022). The problem is further amplified by the limited availability of labeled child-speech data and by bilingual settings, where models must handle both age-related and language-related shifts.

[1]Code, and models are available at https://github.com/qcri/Child-ASR-Adaptation.

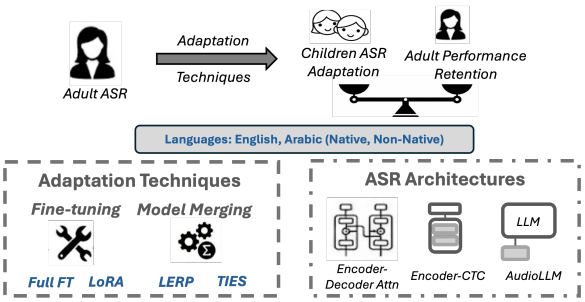


Figure 1: Overview of the child ASR adaptation pipeline across architectures, adaptation methods, and adult-retention evaluation.

A common solution is to adapt adult-trained ASR models using child-speech data. This can improve child recognition on datasets such as MyST (Pradhan et al., 2024) and CMU Kids (Eskenazi et al., 1997), but it can also reduce performance on adult speech due to catastrophic forgetting. This creates a deployment challenge for systems that must serve both adults and children, such as home assistants, classroom tools, and bilingual speech interfaces. We therefore study the *adaptation–retention* trade-off: improving child ASR while preserving adult ASR performance.

This paper presents a retention-aware empirical study of adult-to-child ASR adaptation. We compare three practical adaptation strategies: full fine-tuning, LoRA-based parameter-efficient adaptation (Hu et al., 2021), and post-hoc weight-space merging. For merging, we evaluate Linear Interpolation (LERP) and TIES (Yadav et al., 2023), following the broader motivation of model soups and task arithmetic (Wortsman et al., 2022; Ilharco et al., 2023). Unlike fine-tuning, merging combines adult and child-adapted checkpoints directly in parameter space without additional supervised training, allowing us to test whether child gains can be recovered while reducing adult forgetting. Figure 1 summarizes our experimental pipeline.

Proceedings of The Fourth Arabic Natural Language Processing Conference (ArabicNLP 2026), pages 538–550
October 28-29, 2026 ©2026 Association for Computational Linguistics

We evaluate these strategies across four ASR families: a bilingual encoder–decoder model, a multilingual encoder–decoder model (Whisper Large v3 (Radford et al., 2023)), a bilingual encoder–CTC model, and a multilingual AudioLLM-based ASR system built from Qwen2.5–Omni–7B and adapted for Arabic–English ASR. Our evaluation covers Arabic native and non-native child speech from AraYoungVoices (Chowdhury et al., 2026) (adaptation on ages 7–18; evaluation on its AraKids 7–12 test split), English child speech from MyST (Pradhan et al., 2024), and adult retention benchmarks from LibriSpeech test-clean (Panayotov et al., 2015) and MGB-2 (Ali et al., 2016). In addition to WER, we report Retention Index, Child Adaptation Gain, and Adaptation Recovery to quantify both recognition improvement and retention cost.

Our results reveal several consistent trends. First, child adaptation is necessary across architectures, especially for non-native Arabic children and for English child speech in bilingual settings. Second, language-specific adaptation can cause cross-language degradation, while bilingual adaptation is generally more stable. Third, direct fine-tuning often gives strong child performance gains. However, it can substantially reduce adult ASR performance, particularly on MGB-2. Fourth, weight-space merging often improves the adaptation–retention balance, especially for encoder–CTC, Whisper, and AudioLLM-based ASR systems. LERP tends to preserve adult performance, while TIES more aggressively recovers child-speech gains. For the encoder–decoder model, however, direct bilingual fine-tuning remains strongest in raw WER.

Overall, this study provides practical guidance for building ASR systems that better serve both children and adults. It shows retention-aware evaluation is essential for child ASR adaptation and model merging can be a simple and effective tool for reducing the cost of adult-speech forgetting.

2 Related Work and Positioning

2.1 Child speech recognition and benchmarks

ASR trained predominantly on adult speech degrades on children due to acoustic and linguistic mismatch and limited labeled child data. Recent child-ASR studies have explored diverse modeling approaches, including discrete speech representations, end-to-end neural architectures, and adaptation of large pretrained ASR models to children's speech (Dutta et al., 2025; Sukhadia and Chowdhury, 2024; Shivakumar and Narayanan, 2022; Jain et al., 2023). Adult→child adaptation is a pragmatic route to lower child WER, but raises an often under-emphasized, deployment-critical question: how to improve child recognition *without sacrificing adult competence* when one system serves both populations (Bhardwaj et al., 2022; Shivakumar and Georgiou, 2020; Jain et al., 2023; Shankar et al., 2025). Classic corpora (OGI Kids, PF-STAR) and modern benchmarks (CMU Kids read speech, MyST conversational child–agent dialogue) span controlled and conversational settings (Shobaki et al., 2000; Batliner et al., 2005; Eskenazi et al., 1997; Pradhan et al., 2024).

Prior child-ASR research has largely focused on English, with comparatively limited evaluation in other languages. We therefore extend the analysis to Arabic child speech to examine whether adaptation gains and adult-speech retention generalize across language and nativity shifts. Bilingual and L2 settings add further difficulty, as age-related variability interacts with cross-lingual phonology and accent transfer; PF-STAR includes non-native child speech, while TLT-school targets L2 learners in school settings (Batliner et al., 2005; Gretter et al., 2020). For Arabic, child-specific resources remain scarce, and strong pretrained models still show substantial gaps relative to adult speech (Lu et al., 2022; Alkadri et al., 2025). AraYoungVoices provides an Arabic child-adolescent-speech resource covering both native (L1) and non-native (L2) speakers. In this work, we use AraKids (the children subset) as the Arabic data for this task, enabling evaluation under combined age and nativity shift.

2.2 Adult ASR adaptation to Children and the cost downside

Child adaptation mainly targets end-to-end architectures, including encoder–CTC (Graves et al., 2006) and attention-based encoder–decoder models (Chan et al., 2015), often with Conformer-style encoders (Gulati et al., 2020). Large-scale pretraining has produced strong general-purpose backbones (e.g., wav2vec 2.0 and Whisper) adaptable with modest labeled data (Baevski et al., 2020; Radford et al., 2023), and speech-language stacks pairing audio encoders with LLMs have emerged as alternative ASR backbones/decoders (Rubenstein et al., 2023; Tang et al., 2024; Chu et al., 2024). These trends motivate comparing adaptation strate-

gies across *multiple* model families rather than a single architecture. A growing body of work fine-tunes pretrained ASR on child corpora (e.g., MyST, CMU Kids, PF-STAR), including Whisper adaptation studies and variants that control data quality or mix adult/childrenized speech to reduce mismatch (Jain et al., 2023; Attia et al., 2024; Southwell et al., 2024; Graave et al., 2024; Shankar et al., 2025). These studies consistently show large child-domain gains, but evaluation is frequently centered on child-only WER, leaving the *adult regression* induced by child adaptation less systematically quantified. When one system must serve both adults and children, child-only WER can look like progress while masking a real adult regression that the same system still has to absorb. This motivates retention-aware evaluation where adult ASR performance is explicitly anchored as a first-class constraint. Sequential adaptation induces catastrophic forgetting, where optimizing for a new domain degrades previously learned ones. Continual learning mitigates this via regularization (e.g., EWC), distillation-style constraints (e.g., Learning without Forgetting), and evaluation constructs such as forgetting/backward transfer computed from task-wise performance matrices (Kirkpatrick et al., 2017; Li and Hoiem, 2017; Lopez-Paz and Ranzato, 2017; Chaudhry et al., 2018). In speech recognition, continual learning and domain expansion studies similarly report old-domain degradation and explore constraints for speech models (Hadian and Gorin, 2021). Our study operationalizes this perspective with adult anchors and retention-focused metrics (ratio-based retention/recovery scores), enabling direct comparison of adaptation strategies under an explicit adult-performance requirement.

2.3 Parameter-efficient adaptation and weight-space merging

Parameter-efficient fine-tuning (PEFT) adapts large pretrained backbones while updating only a small number of parameters, reducing training and storage cost. Adapters and LoRA are widely used in NLP (Houlsby et al., 2019; Hu et al., 2021) and have recently been explored for adapting speech foundation models (e.g., Whisper) in multilingual settings (Song et al., 2024). Related child-ASR work also studies lightweight modules and constrained updates to mitigate mismatch in self-supervised representations (Fan and Alwan, 2022). We include LoRA as a *practitioner-standard* baseline and compare its child gains and adult regression against full fine-tuning and post-hoc weight-space merging under the same adult anchors and retention metrics across ASR architectures, framing the study as a controlled empirical comparison rather than a new adaptation method. Weight-space model merging offers a complementary, post-hoc alternative: combining an adult anchor and a child specialist without additional gradient-based training. Model soups average across multiple fine-tuned checkpoints, while task arithmetic interprets fine-tuning as task vectors composed by addition/subtraction (Wortsman et al., 2022; Ilharco et al., 2023). TIES-Merging further addresses destructive interference by pruning small-magnitude updates, resolving sign conflicts, and selectively merging aligned parameters (Yadav et al., 2023). These methods are attractive for *adult-retentive* child adaptation because they can recover child gains while preserving adult performance relative to naive child-only fine-tuning.

2.4 Positioning this work

Building on prior child ASR adaptation studies and continual-learning insights on forgetting, we frame *child ASR adaptation with adult retention* as a retention-aware empirical comparison rather than a new algorithmic proposal. We systematically evaluate (i) full fine-tuning, (ii) LoRA-based PEFT, and (iii) post-hoc weight-space merging (LERP and TIES) across encoder–decoder attention models, encoder–CTC models, Whisper, and an AudioLLM-based ASR. We anchor adult performance on LibriSpeech test-clean and MGB-2 and extend child evaluation beyond English with AraYoungVoices (the test set is Arabic L1/L2, 7–12 years-old age group, referenced to as AraKids) and MyST, enabling a direct characterization of adaptation–retention trade-offs across architectures and languages (Jain et al., 2023; Shankar et al., 2025; Fan and Alwan, 2022; Yadav et al., 2023; Pradhan et al., 2024; Radford et al., 2023). The contribution is therefore empirical rather than algorithmic. We study adult-to-child adaptation under combined age, language, and nativity shifts, and we show that the trade-off between child-speech improvement and adult-speech retention depends on the architecture.

3 Datasets and Benchmarking

We adapt models using child-speech data and evaluate child WER on held-out child test sets. Adult-

Domain	Dataset	Trn	Dev	Tst	Role
Child	AraYoungVoices (Ar)	✓	✓	✓	Adapt + child eval
Child	MyST (En)	✓	✓	✓	Adapt + child eval
Adult	MGB-2 (Ar)	–	✓	✓	Adult retention
Adult	LibriSpeech (En)	–	✓	✓	Adult retention

Table 1: Datasets and split usage. Child data is used for adaptation and child evaluation, while adult data is used only for retention evaluation. For Arabic, training and development data come from the full AraYoungVoices corpus (ages 7–18), while child evaluation uses the AraKids USUP test set (ages 7–12).

Dataset	#Spk	#Utt	Hours
Training data			
AraYoungVoices total	199	≈62K	102.94
AraYoungVoices L1	107	≈32K	45.40
AraYoungVoices L2	92	≈329K	57.54
MyST	–	≈77K	171.20
Total	**–**	**≈141K**	**280.38**
Test data			
AraKids total	66	2K	4.00
AraKids L1	26	1K	1.57
AraKids L2	40	1K	2.43
MyST	–	13,180	28.32
Total	**–**	**15,180**	**32.32**

Table 2: Child adaptation data overview. Training rows report the AraYoungVoices corpus (ages 7–18), split into native (L1) and non-native (L2) speakers; test rows report the AraKids USUP test set (ages 7–12), which is the Arabic child evaluation set used throughout the paper. For MyST, we use the official split.

	Train			Test	
Group	#Spk	#Utt	Hours	#Spk	#Utt
L1 (native)					
Egyptian	33	9,906	14.45	5	252
Gulf	25	7,429	10.95	7	249
Levantine	25	7,089	8.51	7	250
North African	24	7,853	11.50	7	249
L2 (non-native)					
Africa	13	4,242	9.23	7	250
Americas	14	4,582	10.01	5	250
Asia	30	9,449	18.26	15	250
Europe	35	11,024	20.04	13	250
Total	**199**	**61,574**	**102.94**	**66**	**2,000**

Table 3: AraYoungVoices distribution by group. L1 groups are Arabic macro-regions;L2 groups are the speakers' regions of origin. Train and test speakers are disjoint. Train columns report the AraYoungVoices training set (ages 7–18); test columns report the AraKids USUP test set (ages 7–12).

speech datasets only serve as retention anchors to measure how much adult ASR performance is preserved after child adaptation (Tables 1, 2).

We use AraYoungVoices (Chowdhury et al., 2026) and MyST (Pradhan et al., 2024) as our primary datasets for adapting, and testing. Hyperparameters are selected on the development sets, and child ASR performance is reported on held-out test sets.

AraYoungVoices AraYoungVoices is an Arabic children and adolescent read-speech corpus with 61,574 training utterances from 199 speakers, totaling 102,94 hours. It contains two subsets: *L1*, covering native Arabic speakers from Egyptian, Gulf, Levantine, and North African regions, and *L2*, covering non-native Arabic speakers from Africa, Asia, the Americas, and Europe with at least 13 unique speakers from each reported group. Prompts are educator-designed and fully diacritized. L2 prompts are adapted for different proficiency levels from government-certified books, while controlling lexical and syntactic complexity. The AraYoungVoices training set includes speakers aged 7–18 years, while the test set used (AraKids) includes children aged 7–12 years. AraKids is released with two reference types: a *prompt transcription* (the written sentence the child was asked to read), and a *verbatim transcription* produced by human annotators, which records hesitations, repetitions, fillers, and dialectal spellings in place of Modern Standard Arabic forms. We score against the prompt transcription throughout, so the reported Arabic child WER measures agreement with the prompt transcription rather than with a verbatim transcription. Table 3 reports the number of speakers and utterances in each of the eight L1/L2 groups, for both the training and the test split. The AraKids test splits are speaker- and prompt-disjoint (USUP: Unseen Speaker Unseen Prompt), which means no prompt/speaker overlap between train and test. The test set is group-balanced, with approximately 250 utterances sampled from each of the 8 groups: 1,000 utterances for AraKidsL1 and 1,000 for AraKidsL2 yielding a total of 2000 test utterances from 66 speakers of 3.99 hours[2]. The test set is balanced by utterance count across the eight reported L1/L2 groups; speaker counts are not necessarily equal across groups.

MyST We use MyST (Pradhan et al., 2024), an English child-speech corpus of conversational tutoring dialogues, following its official train, development, and test splits. MyST includes students from Grades 3, 4, and 5; the source paper reports grade levels rather than an exact age range.

[2]The average utterance duration is 5.64s for AraKidsL1 and 8.74s for AraKidsL2.

Adult Test sets Adult retention is evaluated using Arabic MGB-2 (Ali et al., 2016) and English LibriSpeech test-clean (Libri-C) (Panayotov et al., 2015). These adult datasets are excluded from child adaptation and are used only to measure whether adapted models preserve performance on adult speech.

4 Methodology

We study child ASR adaptation under an adult-retention constraint. Given an adult-pretrained ASR model, the goal is to improve recognition on child speech while preserving the model's original performance on adult speech. We compare three adaptation paradigms: full fine-tuning, parameter-efficient fine-tuning with LoRA, and post-hoc weight-space merging.

Full fine-tuning and LoRA directly adapt the base model using child-speech data. In contrast, weight-space merging is applied after the adaptation where it combines child-adapted checkpoints with the original model to obtain a single model that balances child-speech improvement and adult-speech retention. Since full fine-tuning and LoRA follow standard training procedures, their implementation details are given in Section 5.2. In this section, we focus on the merging methods and the evaluation metrics used to quantify adaptation, retention, and recovery.

Post-hoc merging Let θ_{base} denote the shared base model and let $\theta^1, \ldots, \theta^N$ denote N child-specialized checkpoints that are parameter-compatible with θ_{base}. Each θ^i is a child-specialized checkpoint produced by adapting the same adult-pretrained base model. Depending on the experiment, the base model is adapted using Arabic child speech, English child speech, or the combined Arabic–English child data through full fine-tuning or LoRA. All merging operations are performed only within a single architecture family (encoder–decoder or encoder–CTC among others) and between checkpoints sharing the same initialization.

Linear interpolation (LERP) We apply linear interpolation in weight space, which takes a weighted average of model parameters. Given the merge set $\{\theta^1, \ldots, \theta^N\}$ (which includes the adult base θ_{base}), the merged model is: $\theta_m = \sum_{i=1}^{N} \alpha_i \theta^i$, $\alpha_i \in [0, 1]$ $\sum_{i=1}^{N} \alpha_i = 1$. Note, that unlike task-arithmetic methods, LERP operates directly on the raw weights and does not construct task vectors relative to θ_{base}.

TIES For each child-adapted checkpoint θ^i, we define its *task vector* as $\tau^i = \theta^i - \theta_{base}$, that is, the parameter update introduced by child adaptation relative to the shared adult base model. TIES (Yadav et al., 2023) first measures how each child-adapted model differs from the shared adult base model. It removes very small parameter changes, which are assumed to have limited importance. When different adapted models change the same parameter in conflicting directions, TIES selects the direction supported by the majority of models. It then averages only the updates that agree with this direction and adds the resulting update to the base model. In contrast to LERP, which averages raw weights, TIES operates on task vectors and resolves sign conflicts before averaging.

Selective merging for encoder–decoder models We also explore selective merging for encoder–decoder ASR models. The goal is to control which components are merged in order to improve child adaptation and adult retention. In addition to full-model merging, we evaluate an encoder-only variant. We merge only the encoder parameters and keep the decoder fixed to the base (adult) checkpoint. This choice is motivated to analyze the contribution of the encoder and decoder components in model merging. In encoder–decoder ASR architectures, the encoder primarily learns acoustic representations, while the decoder models output token dependencies conditioned on these representations. By keeping the decoder fixed to the adult checkpoint, we isolate the effect of merging updates in the encoder and evaluate whether most of the beneficial adaptation is concentrated in the acoustic representation space. This setup also allows us to assess the impact of restricting merging to a subset of parameters, which may reduce interference between task-specific updates.

CTC head anchoring for encoder–CTC models For encoder–CTC models, we merge encoder parameters but keep the adult CTC projection head fixed. We also tested swapping child-specialized CTC heads trained on individual AraYoungVoices, MyST, and joint AraYoungVoices+MyST.

5 Experimental Setup

5.1 ASR Model Architectures

We evaluate child ASR adaptation across four ASR families: bilingual encoder–decoder, multilingual Whisper, bilingual encoder–CTC, and AudioLLM-based ASR.

Bilingual Encoder–Decoder Model (ED-Bi) ED-Bi is an adult-pretrained Arabic–English Conformer–Transformer ASR model (Chowdhury et al., 2021). It uses a self-supervised speech front-end, a 12-block Conformer encoder, and a 6-block Transformer decoder. The model follows a hybrid CTC–attention objective with $\lambda_{\text{CTC}} = 0.3$.

Whisper Model (ED-Ml) ED-Ml is Whisper Large v3, a multilingual encoder–decoder ASR model pretrained on large-scale multilingual and multitask speech data (Radford et al., 2023). We use it as a high-capacity reference model for adaptation and retention analysis.

Bilingual Encoder–CTC Model (E-CTC-Bi) E-CTC-Bi is an encoder-only Arabic–English ASR model (FANAR Team et al., 2026) based on OWSM-CTC (Peng et al., 2024). It uses a frozen Arabic-centric self-supervised front-end (Sukhadia and Chowdhury, 2026), Transformer encoder layers, and self-conditioned CTC with intermediate and final-layer CTC losses.

AudioLLM-based ASR Model (ALLM-ASR-Bi) ALLM-ASR-Bi uses *Qwen2.5-Omni-Ar-En-7B* (Bhatti et al., 2026)[3], an Arabic–English ASR-adapted variant of Qwen2.5-Omni-7B (Qwen Team, 2025). It combines audio understanding with instruction-style decoding. All child-adaptation experiments start from this bilingual ASR-oriented checkpoint.

5.2 Child Adaptation Configuration

We compare full fine-tuning, LoRA adaptation, and post-hoc model merging. All checkpoints are selected using development-set WER on AraYoungVoices and MyST.

Full fine-tuning. ED-Bi and E-CTC-Bi are adapted for 5 epochs, while ED-Ml is adapted for 1 epoch to reduce overfitting. For E-CTC-Bi, we use AdamW with peak learning rate 2.5×10^{-4}, piecewise linear warm-up, and intermediate CTC losses at layers $\{6, 9, 11\}$; the SSL front-end is frozen. For ED-Bi, we use the hybrid CTC–attention objective, Adam with learning rate 2×10^{-4}, and 15k warm-up steps; the XLSR-53 front-end is frozen. For ED-Ml, we use learning rate 10^{-5}, and 500 warm-up steps.

LoRA adaptation. LoRA keeps the base model frozen and updates only low-rank adapters. For ED-Bi and E-CTC-Bi, adapters are inserted into attention projection layers with rank $r = 16$, $\alpha = 16$, and dropout 0.05, and optimized for 5 epochs with AdamW at learning rate 10^{-3}. For ALLM-ASR-Bi, LoRA starts from the bilingual ASR checkpoint, uses rank $r = 16$, $\alpha = 32$, dropout 0.05, and targets `all-linear` modules with bf16 training, cosine scheduling, learning rate 10^{-4}, and warm-up ratio 0.05.

Model merging. Model merging combines the adult base model θ_{base} with child-adapted checkpoints obtained through full fine-tuning θ^{ft}_{child} or LoRA adaptation θ^{lora}_{child}. We consider both Arabic-only and Arabic–English child-adapted checkpoints. For ALLM-ASR-Bi, the merge pool additionally includes the original Qwen2.5-Omni-7B checkpoint as a multilingual prior. For E-CTC-Bi, we retain the adult CTC output head after merging, since it gives lower validation WER than the child-adapted CTC heads from either the Arabic-only or Arabic–English checkpoints (Figure A.2). For LERP, we sweep $\alpha \in [0.2, 0.8]$ and select $\alpha_1 = 0.2$, $\alpha_2 = \alpha_3 = 0.4$ based on validation-set WER. For TIES, we similarly tune the merge weight w and density d on the validation set. The selected merged models are then evaluated on both child and adult benchmarks to quantify the adaptation–retention trade-off. Figures A.1 and A.2 in the Appendix show the merge-parameter sweeps.

5.3 Evaluation Measures

We use word error rate, $\text{WER}(\theta, D_*)$, as the primary ASR metric. Before scoring, we apply lightweight normalization to both references and hypotheses: Arabic diacritics are removed, common Arabic letter variants are normalized, including *alef*, *hamza*, *alef maksura* → *ya*, and *teh marbuta*, and Latin text is lowercased. For MyST, we exclude segments with empty references, non-speech-only markers such as `<silence>` and `<noise>`, or unintelligible placeholders such as `(())`. Empty hypotheses are retained when the reference contains speech, since they reflect recog-

[3]Model obtained upon request from the authors.

nition errors. This reduces scoring artifacts from orthographic variation and non-speech regions.

WER remains the primary evaluation metric and is reported for all child- and adult-speech benchmarks. The measures below do not replace WER; they summarize different aspects of the adaptation–retention trade-off and are computed directly from the reported WER values. Following this, we quantify the trade-offs inherent in child ASR adaptation using three complementary metrics. These metrics separate adult-speech retention, child-speech improvement, and the extent to which merged models recover the gains of child-specific fine-tuning:

- *Retention Index* ($\mathcal{RI}$), which measures how well an adapted or merged model preserves adult ASR performance relative to the base model.
- *Child Adaptation Gain* ($\mathcal{AG}$), which measures how much an adapted model improves over the base model on child speech.
- *Adaptation Recovery* ($\mathcal{AR}$), which measures how close a merged model is to the child-specific fine-tuned model.

For each metric, we also report the *signed WER-point change*, where positive values indicate improvement, while negative values indicate degradation relative to the reference model used by that metric.

Retention Index The Retention Index measures how much adult-speech performance is preserved after adaptation:

$$\mathcal{RI}(\theta) = \frac{\mathrm{WER}(\theta_{base}, D^L_{adult})}{\mathrm{WER}(\theta, D^L_{adult})}. \quad (1)$$

Here, $L \in \{\mathrm{Ar}, \mathrm{En}\}$, θ_{base} is the original adult ASR model, and θ is the adapted or merged model.

A value of $\mathcal{RI} = 1.0$ indicates that adult performance is preserved. Values $<$1.0 indicate forgetting, while values $>$1.0 indicate improved adult performance. To make the magnitude of retention gain or loss explicit, we also report the signed WER-point:

$$\Delta_{adult}(\theta) = \mathrm{WER}(\theta_{base}, D^L_{adult}) - \mathrm{WER}(\theta, D^L_{adult}). \quad (2)$$

Positive values indicate adult WER improvement, while negative values indicate adult WER degradation.

Child Adaptation Gain Child Adaptation Gain measures the relative WER reduction of an adapted model over the base model on child speech:

$$\mathcal{AG}(\theta) = \frac{\mathrm{WER}(\theta_{base}, D^L_{child}) - \mathrm{WER}(\theta, D^L_{child})}{\mathrm{WER}(\theta_{base}, D^L_{child})}. \quad (3)$$

A value of $\mathcal{AG} = 0$ means that the adapted model performs the same as the base model on child speech. Positive values indicate improvement over the base model, while negative values indicate degradation.

The absolute magnitude of the gain or loss explicit are reported using:

$$\Delta_{child}(\theta) = \mathrm{WER}(\theta_{base}, D^L_{child}) - \mathrm{WER}(\theta, D^L_{child}). \quad (4)$$

Positive values indicate improvement, while negative values indicate that adaptation hurts child ASR.

Adaptation Recovery For merged models, Adaptation Recovery measures how much of the child-specific adaptation gain is recovered by the merged model:

$$\mathcal{AR}(\theta_m) = \frac{\mathrm{WER}(\theta_{base}, D^L_{child}) - \mathrm{WER}(\theta_m, D^L_{child})}{\mathrm{WER}(\theta_{base}, D^L_{child}) - \mathrm{WER}(\theta_{ft}, D^L_{child})}. \quad (5)$$

Here, θ_{ft} is the best child-specific adapted model, obtained either through full fine-tuning or LoRA adaptation, and θ_m is the merged model. This metric is computed when θ_{ft} improves over θ_{base} on child speech.

A value of $\mathcal{AR} = 1$ means that the merged model matches the child-specific adapted model. Values with in $[0,\ 1)$ indicate partial recovery of the child-specific adaptation gain, and above 1 indicate that the merged model surpasses the child-specific adapted model, and values < 0 indicate that the merged model performs worse than the base model on child speech. We also report the remaining WER gap to the child-specific adapted model:

$$\Delta_{rec}(\theta_m) = \mathrm{WER}(\theta_{ft}, D^L_{child}) - \mathrm{WER}(\theta_m, D^L_{child}). \quad (6)$$

Positive values indicate that the merged model outperforms the child-specific adapted model, while negative values indicate that it remains worse.

6 Results

Child Adaptation and Adult Retention with Fine-tuning Table 4 reports WER for the adult base models and child-adapted checkpoints.

Encoder–Decoder (ED-Bi)

Model	ArL1	ArL2	MyST	Libri-C	MGB2	Avg.
θ_{base}	9.14	44.11	36.68	9.86	10.92	22.142
$\theta^{Ar}_{child}(ft)$	6.41	26.76	44.23	12.21	10.70	20.062
$\theta^{En}_{child}(ft)$	25.75	106.98	22.68	6.66	14.16	35.246
$\theta^{Ar+En}_{child}(ft)$	6.37	22.71	22.52	6.66	12.97	14.246
$\theta^{Ar+En}_{child}(l)$	7.16	26.49	23.49	7.12	11.99	15.250

Encoder–CTC (E-CTC-Bi)

Model	ArL1	ArL2	MyST	Libri-C	MGB2	Avg.
θ_{base}	14.72	63.37	26.63	3.83	13.18	24.346
$\theta^{Ar}_{child}(ft)$	13.11	16.32	77.32	18.64	23.63	29.804
$\theta^{En}_{child}(ft)$	51.44	92.73	15.11	6.80	26.26	38.468
$\theta^{Ar+En}_{child}(ft)$	12.45	24.24	15.90	6.66	21.20	16.090
$\theta^{Ar+En}_{child}(l)$	15.16	37.65	18.57	5.91	16.77	18.810

Whisper Large v3 (ED-Ml)

Model	ArL1	ArL2	MyST	Libri-C	MGB2	Avg.
θ_{base}	9.68	40.68	21.96	2.47	15.15	17.988
$\theta^{Ar+En}_{child}(ft)$	8.70	11.84	14.47	3.11	23.52	12.328

Qwen2.5–Omni–7B–Ar-En (ALLM-ASR-Bi)

Model	ArL1	ArL2	MyST	Libri-C	MGB2	Avg.
Qwen2.5O	24.24	73.53	15.64	4.07	34.24	30.344
θ_{base}	6.96	34.58	35.87	1.48	13.27	18.432
$\theta^{Ar+En}_{child}(l)$	6.50	26.72	15.55	1.56	16.83	13.432

Table 4: Reported WER (%) for the adult θ_{base} model and adapted ASR models across four architectures. ArL1: AraKidsL1, ArL2: AraKidsL2, ft: full fine-tuning, l: LoRA and Qwen2.5: Base weights for the Qwen Omni model (Qwen Team, 2025). Blue marks the best score within each architecture block. Lower WER is better.

MOp	ArL1	ArL2	MyST	Libri-C	MGB2	Avg. WER
Encoder–Decoder (ED-Bi)						
LERP	7.12	27.01	30.43	09.06	11.00	16.924
LERP (Enc)	7.22	27.38	31.26	8.65	11.00	17.102
TIES	7.14	26.76	28.67	9.23	11.26	16.612
TIES (Enc)	7.35	27.01	28.25	8.16	10.86	16.326
Encoder–CTC (E-CTC-Bi)						
LERP	11.20	23.08	20.25	4.37	14.63	14.706
TIES	11.07	18.94	18.68	5.66	18.71	14.612
Whisper-Large v3 (ED-Ml)						
LERP	6.55	20.01	17.78	2.43	14.49	12.252
LERP (Enc)	7.64	26.38	20.45	2.41	15.00	14.376
TIES	8.48	11.37	14.36	2.91	22.68	11.96
TIES (Enc)	7.16	19.83	17.55	2.61	17.01	12.832
Qwen2.5-Omni-7B (ALLM-ASR-Bi)						
LERP	6.47	27.01	12.79	1.59	14.33	12.438
TIES	6.44	27.15	12.78	1.60	14.19	12.432

Table 5: Reported WER (%) for different merge operations. Best scores within each architecture are highlighted in blue. MOp: merge operation.

Across architectures, child adaptation improves recognition on child speech, with the largest gains seen on ArL2 and MyST. For ED-Bi, bilingual child adaptation reduces WER by 49% and 38.6% on ArL2 and MyST respectively, giving the best average WER within the ED block. For ED-Ml, bilingual adaptation gives even larger gains on ArL2. For ALLM-ASR-Bi, child LoRA improves over all child benchmarks relative to the bilingual adult base, most notably the LoRA is able to recover MyST, where WER drops from 35.87 to 15.55.

The results also show that language-specific adaptation can be unstable. In E-CTC-Bi, Arabic-only adaptation gives the best ArL2 WER, but severely degrades MyST. English-only adaptation gives the best MyST WER, but substantially hurts Arabic child performance. Bilingual adaptation is therefore more balanced, giving the best average WER for both ED-Bi and E-CTC-Bi.

However, child adaptation is not retention-neutral. Adult performance often degrades after child adaptation, especially on MGB-2. For example, MGB-2 WER increases for all architectures under bilingual child adaptation. This confirms the need to evaluate child ASR adaptation together with adult-speech retention.

Merging and Adaptation–Retention Trade-offs Tables 5 and 6 (along with Table 7 in Appendix) show that weight-space merging often improves the adaptation–retention balance, but its effect depends on the architecture and merge operation. Direct child adaptation usually gives strong child-speech gains, but it can also reduce adult ASR performance, especially on Arabic. Merging helps to recover child gain from fine-tuning while reducing the adult-retention cost.

For *ED-Bi*, merging gives modest benefits. The best merged variant is encoder-level TIES, with the lowest average WER for the ED-Bi. However, merged models do not fully match the child WER of bilingual fine-tuning. Their main advantage is retention. LERP keeps MGB-2 close to the base model with $\mathcal{RI} = 0.99$ and only -0.08 WER loss, while TIES gives $\mathcal{RI} = 0.97$. Thus, for ED-Bi, merging is useful for retention, but direct bilingual fine-tuning remains stronger for child WER.

For *E-CTC-Bi*, merging provides a clearer trade-off improvement. Full fine-tuning gives strong child gains but causes substantial adult forgetting, with $\mathcal{RI} = 0.58$ on Libri-C and $\mathcal{RI} = 0.62$ on

MGB-2. LERP improves retention to $\mathcal{RI} = 0.88$ and 0.90, respectively, while still improving child WER. TIES gives the strongest child recovery, exceeding the child-specific reference on ArL1 and ArL2, but with weaker adult retention than LERP. This shows that LERP is more retention-preserving, whereas TIES is more child-adaptation-oriented.

For *ED-Ml*, the same pattern holds. LERP is the safest merge operation for adult retention, reaching $\mathcal{RI} > 1.0$ on both Libri-C and MGB-2, while still improving child WER over the base model. TIES gives stronger child recovery, matching or slightly surpassing the child-specific reference on ArL2 and MyST, but it incurs a larger MGB-2 retention cost.

For *ALLM-ASR-Bi*, merging gives the strongest overall benefit. Both LERP and TIES improve over the child LoRA checkpoint, and TIES gives the best average WER. It improves MyST WER from 15.55 to 12.78 and reduces MGB-2 WER from 16.83 to 14.19. In the trade-off metrics, TIES also gives the best MyST adaptation gain, $\mathcal{AG}$ =0.644(a 64.4% relative WER reduction), and improves MGB-2 retention to $\mathcal{RI} = 0.94$.

Trends Overall, the results show three trends. First, child ASR adaptation is necessary, especially for non-native Arabic children (ArL2) and English (MyST) data in bilingual modeling setup. Second, direct child adaptation can substantially reduce adult ASR performance, making retention-aware evaluation essential. Third, weight-space merging often provides a better compromise, especially for E-CTC-Bi, ED-Ml, and ALLM-ASR-Bi. LERP generally favors adult retention, while TIES more aggressively recovers child-speech gains and can, in some settings, improve both child and adult ASR, as reported above for ALLM-ASR-Bi. This recurring pattern across multiple ASR architectures is one of the central findings of this study.

7 Conclusion

We presented a retention-aware study of adult-to-child ASR adaptation across encoder–decoder, encoder–CTC, Whisper, and AudioLLM-based ASR systems. Using Arabic native and non-native child speech, English MyST, and adult benchmarks from MGB-2 and LibriSpeech test-clean, we evaluated full fine-tuning, LoRA, and post-hoc weight-space merging with WER, Retention Index, and Adaptation Recovery. The results show that child adaptation is necessary, but direct adaptation can reduce adult ASR performance. Bilingual adaptation is generally more stable than language-specific adaptation. Weight-space merging often improves the adaptation–retention balance, particularly for encoder–CTC, Whisper, and AudioLLM-based ASR models, while direct bilingual fine-tuning remains strongest for the Ar-En encoder–decoder model in raw WER. Overall, our findings highlight model merging as a simple and effective strategy for building ASR systems that better serve both child and adult speakers.

	$\mathcal{RI}$ (Δ_{adult})		$\mathcal{AR}$ (Δ_{rec})		
Model	**Libri-C**	**MGB2**	**ArL1**	**ArL2**	**MyST**
Encoder–Decoder (ED-Bi)					
FT	1.48 (+3.20)	0.84 (-2.05)	–	–	–
LoRA	1.38 (+2.74)	0.91 (-1.07)	0.715 (-0.79)	0.823 (-3.78)	0.931 (-0.97)
LERP	1.09 (+0.80)	0.99 (-0.08)	0.729 (-0.75)	0.799 (-4.30)	0.441 (-7.91)
TIES	1.07 (+0.63)	0.97 (-0.34)	0.722 (-0.77)	0.811 (-4.05)	0.566 (-6.15)
Encoder–CTC (E-CTC-Bi)					
FT	0.58 (-2.83)	0.62 (-8.02)	–	–	–
LoRA	0.65 (-2.08)	0.79 (-3.59)	-0.194 (-2.71)	0.657 (-13.41)	0.751 (-2.67)
LERP	0.88 (-0.54)	0.90 (-1.45)	1.551 (+1.25)	1.030 (+1.16)	0.595 (-4.35)
TIES	0.68 (-1.83)	0.70 (-5.53)	1.608 (+1.38)	1.135 (+5.30)	0.741 (-2.78)
Whisper Large v3 (ED-Ml)					
FT	0.79 (-0.64)	0.64 (-8.37)	–	–	–
LERP	1.02 (+0.04)	1.05 (+0.66)	3.194 (+2.15)	0.717 (-8.17)	0.558 (-3.31)
TIES	0.85 (-0.44)	0.67 (-7.53)	1.224 (+0.22)	1.016 (+0.47)	1.015 (+0.11)
Qwen2.5–Omni–7B (ALLM-ASR-Bi)					
LoRA	0.95 (-0.08)	0.79 (-3.56)	–	–	–
LERP	0.93 (-0.11)	0.93 (-1.06)	1.065 (+0.03)	0.963 (-0.29)	1.136 (+2.76)
TIES	0.92 (-0.12)	0.94 (-0.92)	1.130 (+0.06)	0.945 (-0.43)	1.136 (+2.77)

Table 6: Retention and adaptation recovery trade-offs per benchmark. $\mathcal{RI}$ (adult sets) and $\mathcal{AR}$ (child sets) are reported with their signed WER point change (Δ_{adult}, Δ_{rec}); higher values are better. Reference models are on a gray background, and the best Δ is in blue.

8 Limitations

This study focuses on a retention-aware comparison of widely used adaptation strategies rather than proposing a new ASR architecture or merging algorithm. This choice allows us to isolate how full fine-tuning, LoRA, and post-hoc merging behave across different ASR families, but it also leaves room for more specialized adaptation methods, such as continual-learning objectives, rehearsal-based training, or retention-aware regularization. Our evaluation covers Arabic and English child speech, with Arabic further divided into native and non-native speakers. While this provides a useful bilingual and cross-nativity setting, the results may not fully generalize to other languages, spontaneous child speech, or highly noisy real-world environments. Future work can extend the analysis to broader ranges, more dialects, ad-

ditional speaking styles, and deployment-oriented conditions such as classrooms, homes, and mobile recordings. Broad regional categories may also mask country-specific and dialect-specific variation, so these results should not be interpreted as country-level performance estimates. Speaker-age classification and model routing are valid alternatives, but they address a different system design. Routing requires an additional age classifier and multiple models, which can increase latency, deployment cost, and error propagation, especially when adults and children speak in the same session. A direct comparison between routing and unified adaptation is left for future work.

9 Acknowledgment

The work is supported by HBKU flagship research grant (HBKU-INT-VPR-FRG-03-09). The findings achieved herein are solely the responsibility of the authors.

References

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. IEEE.

Mouhand Alkadri, Dania Desouki, and Khloud Al Jallad. 2025. Arabic little stt: Arabic children speech recognition dataset. *Preprint*, arXiv:2510.23319.

Ahmed Adel Attia, Jing Liu, Wei Ai, Dorottya Demszky, and Carol Espy-Wilson. 2024. Kid-whisper: Towards bridging the performance gap in automatic speech recognition for children vs. adults. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, volume 7, pages 74–80.

Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Preprint*, arXiv:2006.11477.

Anton Batliner, Mats Blomberg, Shona D'Arcy, Daniel Elenius, Diego Giuliani, Matteo Gerosa, Christian Hacker, Martin Russell, Stefan Steidl, and Michael Wong. 2005. The PF-STAR children's speech corpus. In *Interspeech 2005*, pages 2761–2764.

Vivek Bhardwaj, Mohamed Tahar Ben Othman, Vinay Kukreja, Youcef Belkhier, Mohit Bajaj, B. Srikanth Goud, Ateeq Ur Rehman, Muhammad Shafiq, and Habib Hamam. 2022. Automatic speech recognition (asr) systems for children: A systematic literature review. *Applied Sciences*, 12(9).

Hunzalah Hassan Bhatti, Firoj Alam, and Shammur Absar Chowdhury. 2026. Multi-task instruction tuning via data scheduling for low-resource arabic speech-llms. In *Proceedings of the 5th Asia-Pacific Chapter of the Association for Computational Linguistics and the 15th International Joint Conference on Natural Language Processing*, Hengqin, China. Accepted for publication.

William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals. 2015. Listen, attend and spell. *Preprint*, arXiv:1508.01211.

Arslan Chaudhry, Puneet K. Dokania, Thalaiyasingam Ajanthan, and Philip H. S. Torr. 2018. *Riemannian Walk for Incremental Learning: Understanding Forgetting and Intransigence*, page 556–572. Springer International Publishing.

Shammur Absar Chowdhury, Amir Hussein, Ahmed Abdelali, and Ahmed Ali. 2021. Towards one model to rule all: Multilingual strategy for dialectal code-switching arabic asr. In *Proc. Interspeech 2021*, pages 2466–2470.

Shammur Absar Chowdhury, Zien Sheikh Ali, Houssam Eddine-Othman Lachemat, and Hamdy Mubarak. 2026. AraYoungVoices: A diverse L1/L2 corpus of Arabic child and adolescent speech. Technical report.

Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, Yichong Leng, Yuanjun Lv, Jinzheng He, Junyang Lin, Chang Zhou, and Jingren Zhou. 2024. Qwen2-audio technical report. *Preprint*, arXiv:2407.10759.

Satwik Dutta, Dwight Irvin, and John HL Hansen. 2025. Exploring discrete speech units for privacy-preserving and efficient speech recognition for school-aged and preschool children. *International Journal of Human-Computer Studies*, 199:103460.

Maxine Eskenazi, Jack Mostow, and David Graff. 1997. The cmu kids corpus. Linguistic Data Consortium, LDC97S63.

Ruchao Fan and Abeer Alwan. 2022. Draft: A novel framework to reduce domain shifting in self-supervised learning and its application to children's asr. In *Proc. Interspeech 2022*, pages 4900–4904.

FANAR Team, Ummar Abbas, Mohammad Shahmeer Ahmad, Minhaj Ahmad, Abdulaziz Al-Homaid, Anas Al-Nuaimi, Enes Altinisik, Ehsaneddin Asgari, Sanjay Chawla, Shammur Chowdhury, Fahim Dalvi, Kareem Darwish, Nadir Durrani, Mohamed Elfeky, Ahmed Elmagarmid, Mohamed Eltabakh, Asim Ersoy, Masoomali Fatehkia, Mohammed Qusay Hashim, and 18 others. 2026. Fanar 2.0: Arabic generative ai stack. *Preprint*, arXiv:2603.16397.

Thomas Graave, Zhengyang Li, Timo Lohrenz, and Tim Fingscheidt. 2024. Mixed children/adult/childrenized fine-tuning for children's asr: How to reduce age mismatch and speaking style mismatch. In *Proc. Interspeech 2024*, pages 5188–5192.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proc. ICML*.

Roberto Gretter, Marco Matassoni, Stefano Bannò, and Falavigna Daniele. 2020. TLT-school: a corpus of non native children speech. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 378–385, Marseille, France. European Language Resources Association.

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 *Proc. Interspeech 2020*, pages 5036–5040.

Hossein Hadian and Arseniy Gorin. 2021. Continual learning using lattice-free mmi for speech recognition. *Preprint*, arXiv:2110.07055.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*. PMLR.

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.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2023. Editing models with task arithmetic. *Preprint*, arXiv:2212.04089.

Rishabh Jain, Andrei Barcovschi, Mariam Yiwere, Peter Corcoran, and Horia Cucu. 2023. Adaptation of Whisper models to child speech recognition. In *Interspeech 2023*, pages 5242–5246.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2017. Overcoming catastrophic forgetting in neural networks. *Preprint*, arXiv:1612.00796. PNAS version available: doi:10.1073/pnas.1611835114.

Zhizhong Li and Derek Hoiem. 2017. Learning without forgetting. *Preprint*, arXiv:1606.09282.

David Lopez-Paz and Marc'Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. *Preprint*, arXiv:1706.08840.

Renee Lu, Mostafa Shahin, and Beena Ahmed. 2022. Improving children's speech recognition by fine-tuning self-supervised adult speech representations. *Preprint*, arXiv:2211.07769.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: An ASR corpus based on public domain audio books. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210.

Yifan Peng, Yui Sudo, Muhammad Shakeel, and Shinji Watanabe. 2024. Owsm-ctc: An open encoder-only speech foundation model for speech recognition, translation, and language identification. *Preprint*, arXiv:2402.12654.

Sameer Pradhan, Ronald A. Cole, and Wayne H. Ward. 2024. My science tutor (MyST)–a large corpus of children's conversational speech. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 12040–12045, Torino, Italia. ELRA and ICCL.

Qwen Team. 2025. Qwen2.5-omni technical report. *Preprint*, arXiv:2503.20215.

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.

Paul K. Rubenstein, Chulayuth Asawaroengchai, Duc Dung Nguyen, Ankur Bapna, Zalán Borsos, Félix de Chaumont Quitry, Peter Chen, Dalia El Badawy, Wei Han, Eugene Kharitonov, Hannah Muckenhirn, Dirk Padfield, James Qin, Danny Rozenberg, Tara Sainath, Johan Schalkwyk, Matt Sharifi, Michelle Tadmor Ramanovich, Marco Tagliasacchi, and 11 others. 2023. Audiopalm: A large language model that can speak and listen. *Preprint*, arXiv:2306.12925.

Natarajan Balaji Shankar, Zilai Wang, Eray Eren, and Abeer Alwan. 2025. Selective attention merging for low resource tasks: A case study of child asr. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

P. Gurunath Shivakumar and Panayiotis Georgiou. 2020. Transfer learning from adult to children for speech recognition: Evaluation, analysis and recommendations. *Computer Speech & Language*, 63:101077.

Prashanth Gurunath Shivakumar and Shrikanth Narayanan. 2022. End-to-end neural systems for automatic children speech recognition: An empirical study. *Computer Speech & Language*, 72:101289.

Khaldoun Shobaki, John-Paul Hosom, and Ronald A. Cole. 2000. The ogi kids' speech corpus and recognizers. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, pages 258–261.

Zheshu Song, Jianheng Zhuo, Yifan Yang, Ziyang Ma, Shixiong Zhang, and Xie Chen. 2024. Lora-whisper: Parameter-efficient and extensible multilingual asr. *Preprint*, arXiv:2406.06619.

Rosy Southwell, Wayne Ward, Viet Anh Trinh, Charis Clevenger, Clay Clevenger, Emily Watts, Jason Reitman, Sidney D'Mello, and Jacob Whitehill. 2024. Automatic speech recognition tuned for child speech in the classroom. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 12291–12295.

Vrunda N. Sukhadia and Shammur Absar Chowdhury. 2024. Children's speech recognition through discrete token enhancement. In *Proc. of the 25th Annual Conference of the International Speech Communication Association (INTERSPEECH)*.

Vrunda N Sukhadia and Shammur Absar Chowdhury. 2026. Harness: Lightweight distilled arabic speech foundation models. *Speech Language Models in Low-Resource Settings: Performance, Evaluation, and Bias Analysis*, pages 109–117.

Changli Tang, Wenyi Yu, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, and Chao Zhang. 2024. Salmonn: Towards generic hearing abilities for large language models. *Preprint*, arXiv:2310.13289.

Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. *Preprint*, arXiv:2203.05482.

Fei Wu, Leibny Paola García-Perera, Daniel Povey, and Sanjeev Khudanpur. 2019. Advances in automatic speech recognition for child speech using factored time delay neural network. In *Interspeech*.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. 2023. Ties-merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems*, volume 36, pages 7093–7115. Curran Associates, Inc.

A Appendix A.

A.1 Parameter sweep ablation

We ablate the TIES merging configuration along two axes: the per-expert weights/densities (Fig. A.1) and the choice of CTC head (Fig. A.2).

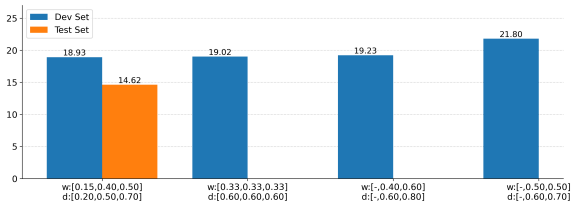


Figure A.1: TIES merging weight/density sweep (average WER, %). Each bar group merges the base model M_0 with expert checkpoints A, B, C using per-expert weights w and densities d, reported in fixed order $[A, B, C]$, where $A = \theta^{En}_{child}$, $B = \theta^{Ar}_{child}$, and $C = \theta^{Ar+En}_{child}$; all variants retain the adult CTC head. Dev (blue) and Test (orange) WER are shown. The selected configuration w=$[0.15, 0.40, 0.50]$, d=$[0.20, 0.50, 0.70]$ attains the lowest WER.

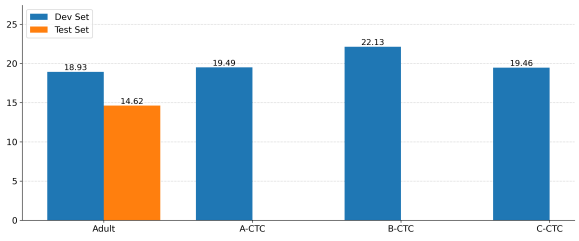


Figure A.2: Effect of CTC head swapping on average WER (%), with the TIES merge fixed to the selected configuration (w=$[0.15, 0.40, 0.50]$, d=$[0.20, 0.50, 0.70]$). *Adult* uses the base adult CTC head; *A-CTC*/*B-CTC*/*C-CTC* substitute the CTC head from expert A/B/C while leaving the merged encoder unchanged. The adult head remains best on Dev.

A.2 Adaptation Gain interpretation

Table 7 reports Child Adaptation Gain ($\mathcal{AG}$) per benchmark, complementing the retention and recovery view in Table 6. The relative ordering of methods is architecture-dependent. For ED, full fine-tuning yields the largest child gains on all three child sets ($\mathcal{AG}$ = 0.303, 0.485, 0.386 on AraKidsL1, AraKidsL2, and MyST); LoRA and both merge operations give smaller gains, with the gap widest on MyST ($\mathcal{AG}$ dropping to 0.170–0.218 for merging).

Model	$\mathcal{AG}$ (Δ_{child})		
	ArL1	**ArL2**	**MyST**
Encoder–Decoder (ED-Bi)			
FT (ref)	0.303 (+2.77)	0.485 (+21.40)	0.386 (+14.16)
LoRA	0.217 (+1.98)	0.399 (+17.62)	0.360 (+13.19)
LERP	0.221 (+2.02)	0.388 (+17.10)	0.170 (+6.25)
TIES	0.219 (+2.00)	0.393 (+17.35)	0.218 (+8.01)
Encoder–CTC (E-CTC-Bi)			
FT (ref)	0.154 (+2.27)	0.617 (+39.13)	0.403 (+10.73)
LoRA	-0.030 (-0.44)	0.406 (+25.72)	0.303 (+8.06)
LERP	0.239 (+3.52)	0.636 (+40.29)	0.240 (+6.38)
TIES	0.248 (+3.65)	0.701 (+44.43)	0.299 (+7.95)
Whisper Large v3 (ED-Ml)			
FT (ref)	0.101 (+0.98)	0.709 (+28.84)	0.341 (+7.49)
LERP	0.323 (+3.13)	0.508 (+20.67)	0.190 (+4.18)
TIES	0.124 (+1.20)	0.721 (+29.31)	0.346 (+7.60)
Qwen2.5–Omni–7B (ALLM-ASR-Bi)			
LoRA (ref)	0.066 (+0.46)	0.227 (+7.86)	0.566 (+20.32)
LERP	0.070 (+0.49)	0.219 (+7.57)	0.643 (+23.08)
TIES	0.075 (+0.52)	0.215 (+7.43)	0.644 (+23.09)

Table 7: Child adaptation gain per benchmark. $\mathcal{AG}$ = Child Adaptation Gain on child sets, reported as relative WER reduction with Δ_{child} in WER points vs. θ_{base}. Higher $\mathcal{AG}$ and positive Δ are better. Within each architecture block, the best Δ per column is in blue.

For E-CTC, TIES is strongest, giving the highest $\mathcal{AG}$ on AraKidsL1 (0.248) and AraKidsL2 (0.701) and even exceeding full fine-tuning, while LoRA is the weakest and slightly degrades AraKidsL1 ($\mathcal{AG} = -0.030$).

For Whisper, LERP gives the best AraKidsL1 gain (0.323), whereas TIES is strongest on AraKidsL2 (0.721) and MyST (0.346).

For Qwen2.5–Omni–7B, child gains on the Arabic sets are small across methods, but both LERP and TIES clearly improve MyST over the LoRA reference ($\mathcal{AG}$ rising from 0.566 to $\approx$ 0.64). Overall, $\mathcal{AG}$ confirms that the strongest child-speech improvement comes from full fine-tuning for ED, but from TIES merging for E-CTC and Whisper.

Author Index

Abbas, Ahmad, 179
Abbas, Mohamed, 263
Abdelmagid, Roaa, 302
Abu Ahmad, Raia, 325
AbuOdeh, Muhammed, 115
Adel, Mohamed, 115
Adib, Ahnaf, 462
Ahmad, Samar, 289, 418
Al Hallay, Rahaf Saeed, 289
Al Rabghi, Ahmad, 289
Al Wazrah, Asma, 340, 444
al-Khatib, Mutaz, 130, 247
Alaboud, Amal, 462
Alahmari, Ahmad, 462
Alam, Firoj, 384
Albatarni, Salam, 11, 22
Aldarmaki, Hanan, 219
AlGhamdi, Fahad, 444
Alhafni, Bashar, 263
Alharbi, Abdullah I., 418
Ali, Ahmed, 164
Ali, Manar, 325
Aljaji, Hamza, 302
Alkanani, Abdallah, 302
Almatham, Rawan Nasser, 340, 444
Alosaimy, Abdulrahman, 340, 444
Alper, Morris, 88
Alrehiely, Majedah, 522
Alsagheer, Dana R, 1
Alshaikh, Rana, 522
Alshammari, Afraa, 462
Altamimi, Afrah, 340
Altamimi, Shahad, 418
Alzaabi, Saeed, 219
Alzahrani, Karima Ali Shahwan, 462
Amjad, Mohammad, 151
Asgari, Ehsaneddin, 302
Assiri, Yahya, 418

Badache, Ismail, 317
Ballout, Mohamad, 508
Baroud, Ibrahim, 325
Bekhouche, Salah Eddine, 130
Belguith, Lamia, 317
Biswas, Md. Rafiul, 384
Bondok, Rawan, 151
Bouchekif, Abdessalam, 130, 247
Brandt, Patrick, 462

Cerqueglini, Letizia, 88
Chakra, Chadi Abou, 508
Chhaytle, Mohamad, 508
Choker, Sally, 508
Chowdhury, Shammur Absar, 538

D'Orazio, Vito, 462
Dalvi, Fahim, 11
de Nijs, Paul, 48
Diallo, Nour, 1
Dibas, Shahd, 219
Dolan, Ryan, 48
Dunn, Jonathan, 48

Eida, Mai Mohamed, 48
El Ghali, Rawan, 302
El-Gizuli, Jamila, 115
El-Haj, Mo, 33
Elgamal, Salman, 151
Elmadani, Khalid N., 115
Elsayed, Tamer, 11, 22
Eltanbouly, Sohaila, 11
Eltanbouly, Somaya, 130, 247
Essgaer, Mansour, 485
Estève, Yannick, 373

Fakih, Nour, 179
Fakih, Tamara, 179
Faraj, Reem, 115
Fathallah, Malak, 356

Gaben, Shahd, 130, 247
Ghaly, Mohammed, 130, 247

Habash, Nizar, 77, 115, 151, 219
Hadid, Abdenour, 130
Hamod, Aya, 289
Hamoud, Hadi, 508
Harthi, Jaber Ahmed, 462
Hasnat, Abul, 384
Hawwari, Abdelati, 444
Hedeya, Abdullah Haytham Mohamed Salah, 433
Hisham, Abdelrahman, 356

Ibrahim, Roaa, 302
Ismael, Hussein Abo, 65
Issa, Elsayed, 200

Istaiteh, Othman, 373

Kamal, Abdulrahman, 1
Kamal, Mohammad, 1
Kamal, Omar, 1
Kassem, Mahmoud, 151
Keleg, Amr, 409
Kementchedjhieva, Yova, 263
Khadawardi, Wageeh, 409
Khalafallah, Ayman, 356
Khalifa, Salam, 77, 219
Khan, Latifur, 462
Khanafer, Abdullah, 302
Kmainasi, Mohamed Bayan, 384
Kulsoom, Umm, 302

Lachemat, Houssam Eddine-Othman, 538
Langlois, David, 373
Lu, Yang, 1

Madkoor, Anas, 302
Magdy, Walid, 164, 409
Mankarious, Saad, 497
Marathe, Mandar, 325
Mattar, Roy, 88
Mdhaffar, Salima, 373
Melhem, Sahar, 170
Mohamed, Abdelrahman Ayman Amin, 433
Mohamed, Ahlam, 88
Mohamed, Emad Soliman Ali, 130
Mohanna, Ammar, 179, 228
Momen, Omar, 325
Mor-Lan, Guy, 88
Mousa, Raghad, 302
Mustafa, Ahmad M, 170

Nabhani, Sara, 325
Nacar, Omer, 289, 340, 418
Nasr, Latifa Iben, 317
Nassar, Mayar, 151

Osorio, Javier, 462

Qandos, Nour, 289

Rahmouni, Kamilia, 200
Rashwani, Samer, 130, 247
Reda, Mahmoud, 151

Saeed, Mostafa, 151, 219
Sapirstein, Ben, 88
Sayed, Marwan, 22
Sbahi, Heba, 130, 247
Shaheen, Abdelrahman, 263
Shahroor, Ali Ezzat, 384
Shareb, Maysa Mosa Abu, 65
Shi, Weidong, 1
Smaili, Kamel, 373
Suwaileh, Reem, 22

Taha, Hanada, 115
Telli, Hichem, 130
Toughrai, Yassine, 373
Truckova, Tereza, 263

Vanetik, Natalia, 65

Wizani, Adel Mahmoud, 151

Youssef, Osama E., 164

Zaghouani, Wajdi, 384
Zaina, Bielasan, 77
Zaraket, Fadi, 508
Zbeeb, Mohammad, 228
Zighem, Mohammed-En-Nadhir, 130
Zirikly, Aya, 497
Zohdy, Nour A., 485