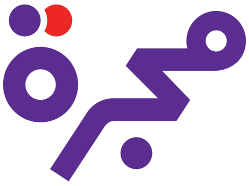
ArabicNLP 2023

The First Arabic Natural Language Processing Conference

Porceedings of the Conference

December 7, 2023

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



Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
209 N. Eighth Street
Stroudsburg, PA 18360
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
acl@aclweb.org

ISBN 978-1-959429-27-2

Preface

Welcome to The first Arabic Natural Language Processing Conference (ArabicNLP 2023) graduating from the Workshop for Arabic Natural Language Processing Workshop (WANLP) which had its seventh, and last, instance last year, in December 2022 within EMNLP 2022. Over the years, WANLP 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, then finally a hybrid event in Abu Dhabi, UAE (EMNLP 2022).

For this year's edition of ArabicNLP, we received a total of 80 main conference submissions and accepted 38 papers (32 long and 6 short), which brings us to an acceptance rate of 47.5%. All papers submitted to the conference were reviewed by at least three reviewers each. Out of the 80 submitted papers, there were 2 desk rejects.

ArabicNLP 2023 included five shared tasks with 48 submissions in totals: (i) The Nuanced Arabic Dialect Identification (NADI) with 13 submissions, (ii) ArAIEval (Persuasion Techniques and Disinformation Detection in Arabic Text) with 17 submissions, (iii) Qur'an QA with 6 submissions, (iv) WojoodNER with 8 submissions, and (v) Arabic Reverse Dictionary with 4 submissions. The shared task overview papers are included in the proceedings. The overview papers and the papers of the shared task winning systems are presented as talks during the conference. None of the shared task papers are counted toward the acceptance rate presented above.

ArabicNLP 2023 also includes a panel discussing the hot topic "Arabic LLMs: Challenges and Opportunities" by leaders in the field, like Areeb Alowisheq, Tom Baldwin, and Kareem Darwish, moderated by Mona Diab.

We were able to secure sponsorship funding from different institutions: King Salman Global Academy for Arabic Language, aiXplain, Lisan.ai, SCAI, Majarra, and Big IR, which we used to support student registrations. We thank all our sponsors for their generous support and their help in building up the Arabic NLP community.

We would like to thank everyone who submitted a paper to the conference, as well as all the members of the Program Committee, who worked hard to provide reviews on a very tight schedule.

Finally, on behalf of everyone involved, organizing committee as well as conference attendees, I would like to thank Nizar Habash and Houda Bouamor for supporting, mentoring and helping this conference be a success, being available for any request and filling any gaps that are overlooked.

Hassan Sawaf, General Chair, on behalf of the conference organizers.

Website of the conference: https://arabicnlp2023.sigarab.org

Organizing Committee

General Chair

Hassan Sawaf, aiXplain Inc., USA

Program Chairs

Samhaa El-Beltagy, Newgiza University, Egypt
Wajdi Zaghouani, Hamad Bin Khalifa University, Qatar
Walid Magdy, University of Edinburgh, UK

Publication Chairs

Ahmed Abdelali, National Center for AI, Riyadh, KSA
Nadi Tomeh, Université Sorbonne Paris Nord, France

Shared Tasks Chair

Ibrahim Abu Farha, The University of Sheffield, UK

Scholarships and Awards Chair

Nizar Habash, New York University Abu Dhabi, UAE

Publicity Chairs

Salam Khalifa, Stony Brook University, USA
Amr Keleg, University of Edinburgh, UK

Sponsorship Chairs

Hatem Haddad, University Manouba, Tunisia
Imed Zitouni, Google

Social Chairs

Khalil Mrini, TikTok, USA
Rawan Almatham, ITMSC, King Saud University, KSA

Table of Contents

Program

Thursday, December 7, 2023

09:00 - 09:20 *Welcome Session*

09:20 - 10:30 *Arabic Downstream NLP Tasks*

In-Context Meta-Learning vs. Semantic Score-Based Similarity: A Comparative Study in Arabic Short Answer Grading
Menna Fateen and Tsunenori Mina

Arabic Dialect Identification under Scrutiny: Limitations of Single-label Classification
Amr Keleg and Walid Magdy

Nâbra: Syrian Arabic Dialects with Morphological Annotations
Amal Nayouf, Tymaa Hasanain Hammouda, Mustafa Jarrar, Fadi A. Zaraket and Mohamad-Bassam Kurdy

Arabic Fine-Grained Entity Recognition
Haneen Abdallatif Liqreina, Mustafa Jarrar, Mohammed Khalilia, Ahmed Oumar El-Shangiti and Muhammad Abdul-Mageed

Cross-Dialectal Named Entity Recognition in Arabic
Niama Elkhbir, Urchade Zaratiana, Nadi Tomeh and Thierry Charnois

Leveraging Domain Adaptation and Data Augmentation to Improve Qur'anic IR in English and Arabic
Vera Pavlova

ArabIcros: AI-Powered Arabic Crossword Puzzle Generation for Educational Applications
Kamyar Zeinalipour, Mohamed Zaky Saad, Marco Maggini and Marco Gori

10:30 - 11:00 *Coffee Break*

11:00 - 12:30 *LLMs and Applications*

Evaluating ChatGPT and Bard AI on Arabic Sentiment Analysis
Abdulmohsen Al-Thubaity, Sakhar Alkhereyf, Hanan Murayshid, Nouf Alshalawi, Maha Bin Omirah, Raghad Alateeq, Rawabi Almutairi, Razan Alsuwailem, Manal Alhassoun and Imaan Alkhanen

TARJAMAT: Evaluation of Bard and ChatGPT on Machine Translation of Ten Arabic Varieties
Karima Kadaoui, Samar M. Magdy, Abdul Waheed, Md Tawkat Islam Khondaker, Ahmed Oumar El-Shangiti, El Moatez Billah Nagoudi and Muhammad Abdul-Mageed

Analyzing Multilingual Competency of LLMs in Multi-Turn Instruction Following: A Case Study of Arabic
Sabri Boughorbel and Majd Hawasly

Beyond English: Evaluating LLMs for Arabic Grammatical Error Correction
Sang Yun Kwon, Gagan Bhatia, El Moatez Billah Nagoudi and Muhammad Abdul-Mageed

Octopus: A Multitask Model and Toolkit for Arabic Natural Language Generation
AbdelRahim Elmadany, El Moatez Billah Nagoudi and Muhammad Abdul-Mageed

AlGhafa Evaluation Benchmark for Arabic Language Models
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

GARI: Graph Attention for Relative Isomorphism of Arabic Word Embeddings
Muhammad Asif Ali, Maha Alshmrani, Jianbin Qin, Yan Hu and Di Wang

Violet: A Vision-Language Model for Arabic Image Captioning with Gemini Decoder
Abdelrahman Mohamed, Fakhraddin Alwajih, El Moatez Billah Nagoudi, Alcides Alcoba Inciarte and Muhammad Abdul-Mageed

ArTST: Arabic Text and Speech Transformer
Hawau Olamide Toyin, Amirbek Djanibekov, Ajinkya Kulkarni and Hanan Aldarmaki

12:30 - 14:00 *Lunch Break*

14:00 - 14:20 *Arabic Core NLP*

SALMA: Arabic Sense-Annotated Corpus and WSD Benchmarks
Mustafa Jarrar, Sanad Malaysha, Tymaa Hasanain Hammouda and Mohammed Khalilia

CamelParser2.0: A State-of-the-Art Dependency Parser for Arabic
Ahmed Elshabrawy, Muhammed AbuOdeh, Go Inoue and Nizar Habash

Thursday, December 7, 2023 (continued)

14:20 - 14:45 *Shared Tasks Overview*

ArAIEval Shared Task: Persuasion Techniques and Disinformation Detection in Arabic Text
Maram Hasanain, Firoj Alam, Hamdy Mubarak, Samir Abdaljalil, Wajdi Zaghouani, Preslav Nakov, Giovanni Da San Martino and Abed Alhakim Freihat

NADI 2023: The Fourth Nuanced Arabic Dialect Identification Shared Task
Muhammad Abdul-Mageed, AbdelRahim Elmadany, Chiyu Zhang, El Moatez Billah Nagoudi, Houda Bouamor and Nizar Habash

WojoodNER 2023: The First Arabic Named Entity Recognition Shared Task
Mustafa Jarrar, Muhammad Abdul-Mageed, Mohammed Khalilia, Bashar Talafha, AbdelRahim Elmadany, Nagham Hamad and Alaa' Omar

KSAA-RD Shared Task: Arabic Reverse Dictionary
Rawan Al-Matham, Waad Alshammari, Abdulrahman AlOsaimy, Sarah Alhumoud, Asma Al Wazrah, Afrah Altamimi, Halah Alharbi and Abdullah Alaifi

Qur'an QA 2023 Shared Task: Overview of Passage Retrieval and Reading Comprehension Tasks over the Holy Qur'an
Rana Malhas, Watheq Mansour and Tamer Elsayed

14:45 - 15:30 *Panel on Arabic LLMs: Challenges and Opportunities (Areeb Alowisheq, Kareem Darwish and Perslav Nakov, moderated by Mona Diab)*

15:30 - 16:00 *Coffee Break*

16:00 - 17:30 *Main Conference Posters (in-Person)*

Enhancing Arabic Machine Translation for E-commerce Product Information: Data Quality Challenges and Innovative Selection Approaches
Bryan Zhang, Salah Danial and Stephan Walter

Multi-Parallel Corpus of North Levantine Arabic
Mateusz Krubiński, Hashem Sellat, Shadi Saleh, Adam Pospíšil, Petr Zemánek and Pavel Pecina

VoxArabica: A Robust Dialect-Aware Arabic Speech Recognition System
Abdul Waheed, Bashar Talafha, Peter Sullivan, AbdelRahim Elmadany and Muhammad Abdul-Mageed

Arabic Topic Classification in the Generative and AutoML Era
Doha Albared, Hadi Hamoud and Fadi A. Zaraket

Yet Another Model for Arabic Dialect Identification
Ajinkya Kulkarni and Hanan Aldarmaki

16:00 - 17:30 *EMNLP Findings Posters (in-Person)*

Filtered Semi-Markov CRF
Urchade Zaratiana, Nadi Tomeh, Niama El Khbir, Pierre Holat and Thierry Charnois

Automatic Pronunciation Assessment - A Review
Yassine El Kheir, Ahmed Ali and Shammur Absar Chowdhury

Data Augmentation Techniques for Machine Translation of Code-Switched Texts: A Comparative Study
Injy Hamed, Nizar Habash and Thang Vu

16:00 - 17:30 *Shared Task Posters (in-Person)*

LIPN at WojoodNER shared task: A Span-Based Approach for Flat and Nested Arabic Named Entity Recognition
Niama Elkhbir, Urchade Zaratiana, Nadi Tomeh and Thierry Charnois

DetectiveRedasers at ArAIEval Shared Task: Leveraging Transformer Ensembles for Arabic Deception Detection
Bryan E. Tuck, Fatima Zahra Qachfar, Dainis Boumber and Rakesh M. Verma

Nexus at ArAIEval Shared Task: Fine-Tuning Arabic Language Models for Propaganda and Disinformation Detection
Yunze Xiao and Firoj Alam

PTUK-HULAT at ArAIEval Shared Task Fine-tuned Distilbert to Predict Disinformative Tweets
Areej Jaber and Paloma Martinez

ReDASPersuasion at ArAIEval Shared Task: Multilingual and Monolingual Models For Arabic Persuasion Detection
Fatima Zahra Qachfar and Rakesh M. Verma

UL & UM6P at ArAIEval Shared Task: Transformer-based model for Persuasion Techniques and Disinformation detection in Arabic
Salima Lamsiyah, Abdelkader El Mahdaouy, Hamza Alami, Ismail Berrada and Christoph Schommer

UWB at Arabic Reverse Dictionary shared task: Computing the meaning of a gloss
Stephen Taylor

NLPeople at NADI 2023 Shared Task: Arabic Dialect Identification with Augmented Context and Multi-Stage Tuning
Mohab Elkaref, Movina Moses, Shinnosuke Tanaka, James Barry and Geeth De Mel

SANA at NADI 2023 shared task: Ensemble of Layer-Wise BERT-based models for Dialectal Arabic Identification
Nada Almarwani and Samah Aloufi

The Helsinki-NLP Submissions at NADI 2023 Shared Task: Walking the Baseline
Yves Scherrer, Aleksandra Miletić and Olli Kuparinen

LKAU23 at Qur'an QA 2023: Using Transformer Models for Retrieving Passages and Finding Answers to Questions from the Qur'an
Sarah Alnefaie, Abdullah N. Alsaleh, Eric Atwell, Mohammad Alsalka and Abdulrahman Altahhan

El-Kawaref at WojoodNER shared task: StagedNER for Arabic Named Entity Recognition
Nehal Elkaref and Mohab Elkaref

UM6P & UL at WojoodNER shared task: Improving Multi-Task Learning for Flat and Nested Arabic Named Entity Recognition
Abdelkader El Mahdaouy, Salima Lamsiyah, Hamza Alami, Christoph Schommer and Ismail Berrada

16:00 - 17:30 *Main Conference Posters (Virtual)*

IDRISI-D: Arabic and English Datasets and Benchmarks for Location Mention Disambiguation over Disaster Microblogs
Reem Suwaileh, Tamer Elsayed and Muhammad Imran

HICMA: The Handwriting Identification for Calligraphy and Manuscripts in Arabic Dataset
Anis Ismail, Zena Kamel and Reem Mahmoud

Thursday, December 7, 2023 (continued)

Simplify: Automatic Arabic Sentence Simplification using Word Embeddings
Yousef SalahEldin and Caroline Sabty

ArBanking77: Intent Detection Neural Model and a New Dataset in Modern and Dialectical Arabic
Mustafa Jarrar, Ahmet Birim, Mohammed Khalilia, Mustafa Erden and Sana Ghanem

ArSarcasMoji Dataset: The Emoji Sentiment Roles in Arabic Ironic Contexts
Shatha Ali A. Hakami, Robert Hendley and Phillip Smith

Machine Translation of Omani Arabic Dialect from Social Media
Khoula Al-Kharusi and Abdurahman Khalifa AAlAbdulsalam

Automated De-Identification of Arabic Medical Records
Veysel Kocaman, Youssef MELLAH, Hasham Ul Haq and David Talby

Arabic dialect identification: An in-depth error analysis on the MADAR parallel corpus
Helene Bøsei Olsen, Samia Touileb and Erik Velldal

Investigating Zero-shot Cross-lingual Language Understanding for Arabic
Zaid Alyafeai and Moataz Ahmed

On Enhancing Fine-Tuning for Pre-trained Language Models
Abir Betka, Zeyd Ferhat, Riyadh Barka, Selma Boutiba, Zineddine S. Kahhoul, Tiar M. Lakhdar, Ahmed Abdelali and Habiba Dahmani

LANS: Large-scale Arabic News Summarization Corpus
Abdulaziz Alhamadani, Xuchao Zhang, Jianfeng He, Aadyant Khatri and Chang-Tien Lu

Performance Implications of Using Unrepresentative Corpora in Arabic Natural Language Processing
Saied Alshahrani, Norah Alshahrani, Soumyabrata Dey and Jeanna Matthews

ArTrivia: Harvesting Arabic Wikipedia to Build A New Arabic Question Answering Dataset
Sultan Alrowili and K Vijay-Shanker

Aswat: Arabic Audio Dataset for Automatic Speech Recognition Using Speech-Representation Learning
Lamya Alkanhal, Abeer Alessa, Elaf Almahmoud and Rana Alaqil

Offensive Language Detection in Arabizi
Imene Bensalem, Meryem Ait Mout and Paolo Rosso

16:00 - 17:30 *EMNLP Findings Posters (Virtual)*

Arabic Mini-ClimateGPT : A Climate Change and Sustainability Tailored Arabic LLM
Sahal Shaji Mullappilly, Abdelrahman M Shaker, Omkar Chakradhar Thawakar, Hisham Cholakkal, Rao Muhammad Anwer, Salman Khan and Fahad Khan

Dolphin: A Challenging and Diverse Benchmark for Arabic NLG
El Moatez Billah Nagoudi, AbdelRahim A. Elmadany, Ahmed Oumar El-Shangiti and Muhammad Abdul-Mageed

16:00 - 17:30 *Shared Task Posters (Virtual)*

AAST-NLP at ArAIEval Shared Task: Tackling Persuasion technique and Disinformation Detection using Pre-Trained Language Models On Imbalanced Datasets
Ahmed El-Sayed, Omar Nasr and Noureldin Elmadany

AraDetector at ArAIEval Shared Task: An Ensemble of Arabic-specific pre-trained BERT and GPT-4 for Arabic Disinformation Detection
Ahmed Bahaaulddin, Vian Sabeeh, Hanan M. Belhaj, Serry Sibaee, Samar Ahmad, Ibrahim Khurfan and Abdullah I. Alharbi

Frank at ArAIEval Shared Task: Arabic Persuasion and Disinformation: The Power of Pretrained Models
Dilshod Azizov, Jiyong Li and Shangsong Liang

HTE at ArAIEval Shared Task: Integrating Content Type Information in Binary Persuasive Technique Detection
Khaldi Hadjer and Taqiy Eddine Bouklouha

Itri Amigos at ArAIEval Shared Task: Transformer vs. Compression-Based Models for Persuasion Techniques and Disinformation Detection
Jehad Oumer, Nouman Ahmed and Natalia Flechas Manrique

KnowTellConvince at ArAIEval Shared Task: Disinformation and Persuasion Detection in Arabic using Similar and Contrastive Representation Alignment
Hariram Veeramani, Surendrabikram Thapa and Usman Naseem

Thursday, December 7, 2023 (continued)

Legend at ArAIEval Shared Task: Persuasion Technique Detection using a Language-Agnostic Text Representation Model
Olumide E. Ojo, Olaronke O. Adebanji, Hiram Calvo, Damian O. Dieke, Olumuyiwa E. OJO, Seye E. Akinsanya, Tolulope O. Abiola and Anna Feldman

Mavericks at ArAIEval Shared Task: Towards a Safer Digital Space - Transformer Ensemble Models Tackling Deception and Persuasion
Sudeep Mangalvedhekar, Kshitij Deshpande, Yash Patwardhan, Vedant Deshpande and Ravindra Murumkar

PD-AR at ArAIEval Shared Task: A BERT-Centric Approach to Tackle Arabic Disinformation
Pritam Deka and Ashwathy T. Revi

Raphael at ArAIEval Shared Task: Understanding Persuasive Language and Tone, an LLM Approach
Utsav Shukla, Manan Vyas and Shailendra Tiwari

USTHB at NADI 2023 shared task: Exploring Preprocessing and Feature Engineering Strategies for Arabic Dialect Identification
Mohamed Lichouri, Khaled Lounnas, Aicha Zitouni, Houda Latrache and Rachida Djeradi

rematchka at ArAIEval Shared Task: Prefix-Tuning & Prompt-tuning for Improved Detection of Propaganda and Disinformation in Arabic Social Media Content
Reem Abdel-Salam

Abed at KSAA-RD Shared Task: Enhancing Arabic Word Embedding with Modified BERT Multilingual
Abdelrahim Qaddoumi

Qamosy at Arabic Reverse Dictionary shared task: Semi Decoder Architecture for Reverse Dictionary with SBERT Encoder
Serry Sibaee, Samar Ahmad, Ibrahim Khurfan, Vian Sabeeh, Ahmed Bahaaulddin, Hanan M. Belhaj and Abdullah I. Alharbi

Rosetta Stone at KSAA-RD Shared Task: A Hop From Language Modeling To Word–Definition Alignment
Ahmed Elbakry, Mohamed Gabr, Muhammad ElNokrashy and Badr AlKhamissi

ANLP-RG at NADI 2023 shared task: Machine Translation of Arabic Dialects: A Comparative Study of Transformer Models
Wiem Derouich, Sameh Kchaou and Rahma Boujelbane

DialectNLU at NADI 2023 Shared Task: Transformer Based Multitask Approach Jointly Integrating Dialect and Machine Translation Tasks in Arabic
Hariram Veeramani, Surendrabikram Thapa and Usman Naseem

Thursday, December 7, 2023 (continued)

Frank at NADI 2023 Shared Task: Trio-Based Ensemble Approach for Arabic Dialect Identification
Dilshod Azizov, Jiyong Li and Shangsong Liang

ISL-AAST at NADI 2023 shared task: Enhancing Arabic Dialect Identification in the Era of Globalization and Technological Progress
Shorouk Adel and Noureldin Elmadany

IUNADI at NADI 2023 shared task: Country-level Arabic Dialect Classification in Tweets for the Shared Task NADI 2023
Yash A. Hatekar and Muhammad S. Abdo

Mavericks at NADI 2023 Shared Task: Unravelling Regional Nuances through Dialect Identification using Transformer-based Approach
Vedant Deshpande, Yash Patwardhan, Kshitij Deshpande, Sudeep Mangalvedhekar and Ravindra Murumkar

USTHB at NADI 2023 shared task: Exploring Preprocessing and Feature Engineering Strategies for Arabic Dialect Identification
Mohamed Lichouri, Khaled Lounnas, Aicha Zitouni, Houda Latrache and Rachida Djeradi

UniManc at NADI 2023 Shared Task: A Comparison of Various T5-based Models for Translating Arabic Dialectical Text to Modern Standard Arabic
Abdullah Khered, Ingy Yasser Abdelhalim, Nadine Abdelhalim, Ahmed Soliman and Riza Batista-Navarro

UoT at NADI 2023 shared task: Automatic Arabic Dialect Identification is Made Possible
Abduslam F A Nwesri, Nabila A S Shinbir and Hassan A H Ebrahem

rematchka at NADI 2023 shared task: Parameter Efficient tuning for Dialect Identification and Dialect Machine Translation
Reem Abdel-Salam

AHJL at Qur'an QA 2023 Shared Task: Enhancing Passage Retrieval using Sentence Transformer and Translation
Hessa A. Alawwad, Lujain A. Alawwad, Jamilah Alharbi and Abdullah I. Alharbi

Al-Jawaab at Qur'an QA 2023 Shared Task: Exploring Embeddings and GPT Models for Passage Retrieval and Reading Comprehension
Abdulrezzak Zekiye and Fadi Amroush

GYM at Qur'an QA 2023 Shared Task: Multi-Task Transfer Learning for Quranic Passage Retrieval and Question Answering with Large Language Models
Ghazaleh Mahmoudi, Yeganeh Morshedzadeh and Sauleh Eetemadi

LowResContextQA at Qur'an QA 2023 Shared Task: Temporal and Sequential Representation Augmented Question Answering Span Detection in Arabic
Hariram Veeramani, Surendrabikram Thapa and Usman Naseem

Alex-U 2023 NLP at WojoodNER shared task: AraBINDER (Bi-encoder for Arabic Named Entity Recognition)
Mariam Hussein, Sarah Khaled, Marwan Torki and Nagwa El-Makky

AlexU-AIC at WojoodNER shared task: Sequence Labeling vs MRC and SWA for Arabic Named Entity Recognition
Shereen Elkordi, Noha Adly and Marwan Torki

AlphaBrains at WojoodNER shared task: Arabic Named Entity Recognition by Using Character-based Context-Sensitive Word Representations
Toqeer Ehsan, Amjad Ali and Ala Al-Fuqaha

ELYADATA at WojoodNER Shared Task: Data and Model-centric Approaches for Arabic Flat and Nested NER
Imen Laouirine, Haroun Elleuch and Fethi Bougares

Lotus at WojoodNER Shared Task: Multilingual Transformers: Unveiling Flat and Nested Entity Recognition
Jiyong Li, Dilshod Azizov, Hilal AlQuabeh and Shangsong Liang

TCE at Qur'an QA 2023 Shared Task: Low Resource Enhanced Transformer-based Ensemble Approach for Qur'anic QA
Mohammed Alaa Elkomy and Amany Sarhan

Violet: A Vision-Language Model for Arabic Image Captioning with Gemini Decoder

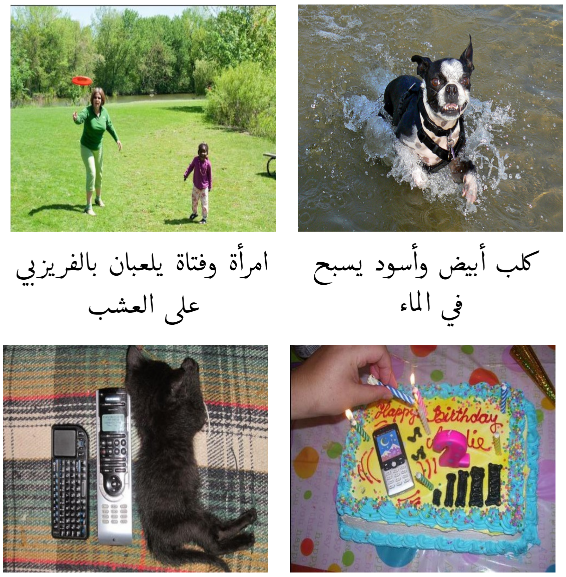
Abdelrahman Mohamed[ξ] **Fakhraddin Alwajih**[λ] **El Moatez Billah Nagoudi**[λ]
Alcides Alcoba Inciarte[λ] **Muhammad Abdul-Mageed**[λ,ξ]
[λ] Deep Learning & Natural Language Processing Group, The University of British Columbia
[ξ]Department of Natural Language Processing & Department of Machine Learning, MBZUAI
{fakhr.alwajih,moatez.nagoudi,muhammad.mageed}@ubc.ca

Abstract

Although image captioning has a vast array of applications, it has not reached its full potential in languages other than English. Arabic, for instance, although the native language of more than 400 million people, remains largely underrepresented in this area. This is due to the lack of labeled data and powerful Arabic generative models. We alleviate this issue by presenting a novel vision-language model dedicated to Arabic, dubbed *Violet*. Our model is based on a vision encoder and a Gemini text decoder that maintains generation fluency while allowing fusion between the vision and language components. To train our model, we introduce a new method for automatically acquiring data from available English datasets. We also manually prepare a new dataset for evaluation. *Violet* performs sizeably better than our baselines on all of our evaluation datasets. For example, it reaches a CIDEr score of 61.2 on our manually annotated dataset and achieves an improvement of 13 points on Flickr8k.

1 Introduction

Captioning images involves describing the visual elements of a picture using natural language. This requires a system that combines the strengths of two models: one that can represent the visual elements of an image, and another that can translate this representation into natural language. The latter employs a language model to produce *fluent* (i.e., grammatically accurate) and *adequate* (i.e., capturing sufficient semantic information) descriptions. In recent years, research on vision language models (VLMs) and their applications has boomed (Alayrac et al., 2022; Wang et al., 2022; Huang et al., 2023). Owing to the rapid advancements in large language models (LLMs), the performance of VLMs has improved dramatically. More concretely, VLMs have progressed from merely providing descriptions that vaguely resemble a given image (Vinyals et al., 2015) to accurately describing complex visual cues within the image. The *pretraining-then-finetuning* paradigm also plays a significant role in achieving such impressive results, as it allows models to first grasp general language structures and then specialize in the specific task of image captioning (Gan et al., 2022).


امرأة وفتاة يلعبان بالفريزبي على العشب

كلب أبيض وأسود يسبح في الماء

قطة مستلقية بجانب جهاز تحكم عن بعد

كعكة عيد ميلاد مع هاتف محمول عليها

Figure 1. Examples of captions generated by our model.

Progress in VLMs, however, has been witnessed thus far primarily on English Awais et al. (2023). This leaves behind a large number of other languages for which no sufficient image captioning data or language models exist. Arabic is a case in point where image captioning lags far behind (Elbedwehy and Medhat, 2023). Similar to other low-resource languages, progress in Arabic image captioning has been hampered by the lack of publicly available datasets and limited efforts in creating any such data. Manual creation of image datasets, after all, requires a huge amount of time and labor. Again, the unavailability of powerful Arabic language models that understands the struc-

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 1–11
December 7, 2023 ©2023 Association for Computational Linguistics

ture of the language and can capture its rich morphology has also caused a delay in the development of VLMs. Given the rapid progress in vision language technologies and their wide applications in society, limited progress in this area can have negative consequences for the Arabic-speaking world.

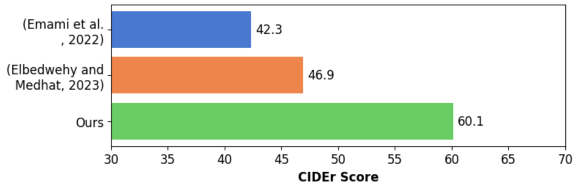


Figure 2. Performance of our model compared to previous works on Flickr8k using CIDEr metric.

To address this important issue, we introduce a novel Arabic image captioning model dubbed *Violet*. Our new model is comprised of two main components: a vision encoder and a text decoder. For the vision encoder, we employ an object detector network based on FasterRCNN (Ren et al., 2015) to extract visual features that are then passed to a compact transformer encoder. At the decoder side, we leverage the recently developed generative pretrained model JASMINE (Nagoudi et al., 2022). Taking inspiration from (Yu et al., 2022), we split our text decoder into two halves: the first half functions as a text decoder, whereas the second incorporates cross-attention layers, effectively serving as a fusion decoder. Given the dual nature of our decoder, we refer to it as *Gemini*. Drawing parallels with VisualGPT (Chen et al., 2022) and the meshed transformer (Cornia et al., 2020), we also adopt a meshed connection between the transformer vision encoder and the text decoder to foster enhanced communication between the encoder and decoder layers.

The other major challenge we face in our work is the unavailability of native Arabic captioning data. We alleviate this challenge by introducing a method for automatically acquiring captions that is based on first employing a powerful machine translation model followed by a quality assurance mechanism for removing poor captions. For evaluation, in addition to reporting on Arabic translated dataset, we task five human annotators to manually caption an image dataset. Compared to previous works and baselines, our novel model excels in captioning images in fluent Arabic. Figure 1 offers four examples of fluent Arabic captions generated by our novel model. Figure 2 shows a comparison of our model performance with prior research on Flickr8k in CIDEr score.

In summary, our contributions are as follows:

- We present a novel image captioning model that employs an effective pretrained Arabic decoder capable of outputting rich captions.
- Our model achieves competitive performance for Arabic image captioning on both the MSCOCO (Lin et al., 2014) and Flicker8k (Jia et al., 2014) datasets, establishing a new state-of-the-art in this area.
- In the process of developing our new model, we release a translated version of MSCOCO dataset that has gone through our quality assurance pipeline. Our released dataset can help further advance research in Arabic VLMs.
- We also release our manually captioned dataset, a subset of MSCOCO test set, that we dub *AraCOCO*.

2 Related Work

Image captioning. Early methods for image captioning involve either retrieving descriptions (Karpathy et al., 2014) or using template filling combined with manually designed natural language generation techniques (Yang et al., 2011; Li et al., 2011). However, modern image captioning primarily relies on deep learning models. In early work, image captioning is framed as an image-to-sequence task using encoder-decoder models, with Convolutional Neural Networks (CNNs) as encoders and Recurrent Neural Networks (RNNs) as decoders while incorporating attention mechanisms (Xu et al., 2015; You et al., 2016; Huang et al., 2019). Soon after, using a transformer architecture of a vision encoder with a text decoder became the defacto direction towards solving the problem of image captioning (Stefanini et al., 2022). Some approaches use a detection model to extract visual features and then pass it to a transformer text decoder as in Oscar (Li et al., 2020; Chen et al., 2022), while others like CoCa (Yu et al., 2022) train a transformer vision encoder with a text decoder from scratch on a large-scale dataset.

More recently, there has been a shift towards using pre-trained LLMs and vision models. Generative Image-to-text Transformer (GIT) (Wang et al., 2022) is a decoder-only transformer that utilizes a CLIP (Radford et al., 2021) visual encoder to incorporate both visual and textual inputs. Another method to consider is VisualGPT (Chen et al.,

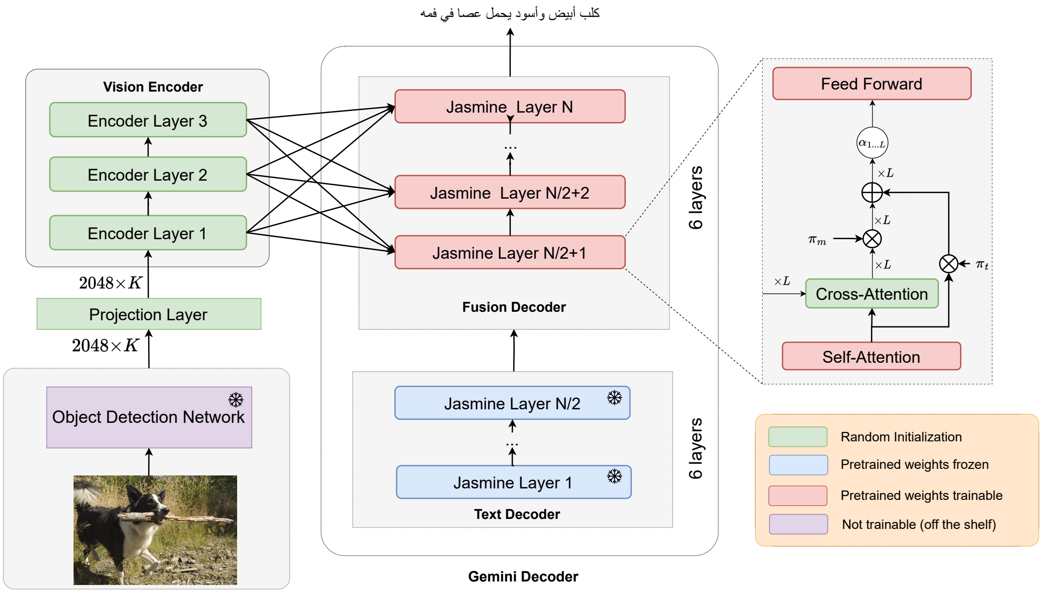


Figure 3. The architecture and output generated by our model. We use an object detection network to extract K object features (K equal 50 in our case) from an image. After projecting to a lower dimension, the features are fed to an L-layer (three-layer in our architecture) transformer encoder. Meshed connection is employed between the encoder and decoder layers, where each encoder layer contributes to the cross-attention output. Our text decoder is split into two halves, the first half is the standard frozen pretrained text decoder layers, while the second half has cross-attention layers inserted after each self-attention layer. We call this design a *Gemini decoder*. We employ a gating mechanism through π_t and π_m that controls the flow of information from the vision and language sides. The final input to the feed forward network in each cross-attention layer is the weighted sum of each encoder-decoder attention controlled by the α parameters.

2022) which uses a pretrained FasterCNN to extract visual features that it passes to a small vision encoder. For the decoder side, it uses the text-pretrained model GPT2 (Radford et al., 2019).

Arabic image captioning. Arabic poses significant challenges to image captioning. This is due to the lack of native Arabic captioning datasets in the public domain, the morphological complexity of Arabic, and the large number of diverse dialects (Attai and Elnagar, 2020). However, a number of Arabic image captioning works exist. For instance, approaches such as root-word based RNNs and deep neural networks are used for direct Arabic caption generation (Jindal, 2017). Al-Muzaini et al. (2018) employ a generative merge model with three components: an LSTM-based language model, a CNN-based image feature extraction model, and a decoder that processes outputs from the first two models. ElJundi et al. (2020) introduce an Arabic captioning model trained on a translated Flickr8K dataset, discussing issues related to translation. Afyouni et al. (2021) present AraCap, a hybrid design that combines a CNN with object detection using attention mechanisms and produces captions through an LSTM. They train their model on MSCOCO and Flickr30k (Plummer et al., 2015) datasets and test on an Arabic translated subset of MSCOCO. Lasheen and Barakat (2022) propose an encoder-decoder structure, incorporating attention mechanisms with CNN encoding and LSTM decoding. In another study (Emami et al., 2022), various Arabic image captioning models are formulated and assessed using standard metrics. The authors use transformers pretrained on diverse Arabic datasets following the architecture and training method introduced in OSCAR (Li et al., 2020). Elbedwehy and Medhat (2023) present a model employing transformers for both encoding and decoding. It uses feature extraction from images in the encoding stage and a pretrained word embedding model in the decoding stage, all tested on the Arabic-translated Flickr8k dataset in ElJundi et al. (2020). This work is closest to ours in that we also utilize transformer encoders and decoders. However, we use a GPT-styled decoder that endows our approach with high Arabic fluency.

3 Approach

3.1 Model Architecture

Our model is a vision-encoder-decoder architecture. For the vision encoder part, we employ an object detection network (Anderson et al., 2018) and a three-layer transformer. For the text decoder, we use the pretrained transformer decoder JASMINE (Nagoudi et al., 2022). To align visual and textual features, we utilize cross-attention. In standard attention, also known as self-attention, the attention output is computed using three matrices derived from the same input: the query matrix Q, the key matrix K, and the value matrix V. More concretely, given an input sequence represented as a matrix S_t, where each row corresponds to a vector in the sequence, the attention is calculated as:

$$Attn(S_t) = \text{softmax}\left(\frac{S_t W_q (S_t W_k)^T}{\sqrt{d_k}}\right) S_t W_v \quad (1)$$

Where W_q, W_k, and W_v are the learnable weight matrices for the query Q, key K, and value V respectively. d_k is the dimensionality of the query/key vectors. The division by $\sqrt{d_k}$ is a scaling factor to ensure the dot products don't grow too large as the dimensionality increases.

In the case of cross-attention, the query is derived from the output of the text decoder's self-attention, while the key and value are sourced from the vision encoder. Mathematically, given image visual features output S_m, and the textual features S_t, the formula becomes:

$$XAttn(S_t, S_m) = \text{softmax}\left(\frac{(S_t W_q)(S_m W_k)^T}{\sqrt{d_k}}\right) \times S_m W_v \quad (2)$$

Now that the attention mechanism foundations are laid out, we describe our vision encoder and text decoder in detail.

3.1.1 Vision Encoder

Our vision encoder consists of two components: a pretrained object detection network, and a three-layer transformer encoder. For the object detection network, we employ bottom-up attention network (Anderson et al., 2018). In our initial experiments, it results in superior visual features compared to using the vanilla FasterRCNN model (Ren et al., 2015). Previous works (Li et al., 2020; Cornia et al., 2020; Chen et al., 2022) also show the effectiveness of this network in feature extraction. The transformer encoder, on the other hand, is a three-layer standard transformer architecture that takes the output of the detection network to further refine the visual features. For each image, the detection network detects the potential objects and extracts the visual features from their bounding boxes.[1] These visual features are passed through a projection layer and then fed to the three-layer transformer encoder as input. We adapt meshed connection (Cornia et al., 2020) in our architecture between the encoder layers and the text decoder. This allows all the encoder layers to contribute to the input of the cross-attention rather than using only the output of the last encoder layer. The contribution of each encoder layer is determined by the learnable parameters matrix α. For each layer i, α_i is calculated as:

$$\alpha_i = \sigma(W_i[S_t \parallel XAttn(S_{m_i}, S_t) + b_i]) \quad (3)$$

Where S_t is the input sequence of each decoder layer, σ is the sigmoid activation function, W_i is a learnable weight matrix, b_i is a bias term and $\parallel$ indicates concatenation. This measures the relevance between the input for each decoder layer S_t, and the output of each encoder layer.

3.1.2 Gemini Decoder

We employ the pretrained Arabic decoder JASMINE (Nagoudi et al., 2022) as our text decoder. JASMINE is a decoder-based transformer that follows GPTNeo architecture (Black et al., 2021). JASMINE models range in complexity from 300 million to 13 billion parameters and are trained on a text dataset of approximately 400GB, covering diverse Arabic varieties from multiple domains. We utilize the JASMINE base variant in our architecture, which is a 12-layer transformer decoder with a 768-dimensional embedding.

Although the meshed connection introduced in Cornia et al. (2020) proved to have positive improvements on performance due to the richer visual features, calculating the cross-attention of each encoder layer with each decoder layer is computationally expensive. Inspired by Yu et al. (2022), we split our pretrained text decoder into two parts. The first part acts as a vanilla text decoder, while the second part acts as a fusion decoder that aligns visual and textual features. This design choice serves two purposes. First, it reduces the computations and the number of parameters by removing cross-attention layers and the mesh connections in the first half of the decoder. Second, having its first half intact acting as a vanilla text decoder, allows our decoder to keep its innate generative capabilities, while also enabling smoother convergence.

[1] A bounding box is a region in the image that contains the object.

As shown in Figure 3, the first half has only the pretrained self-attention layers of JASMINE. While the second half got cross-attention blocks inserted in-between each layer, acting as a fusion decoder. To ensure maintaining the functionality of our pretrained decoder, we freeze the first part that acts as the text decoder. This modification not only decreases computational cost but also positively impacts overall performance. In order to further enhance the quality of the features generated by both the vision encoders and the text decoder, we employ self-resurrecting activation unit (SRAU) introduced in Chen et al. (2022). The process of generating a caption relies on visual cues to convey the image's content and textual cues to provide relationships between words for a coherent and fluent output. To allow the important information to flow without distortion, SRAU selectively permits the activation above a certain threshold through a gating mechanism. This effectively filters out any weak signal produced by either the vision or language part.

Concretely, as shown in Figure 3, for each encoder-decoder connection, the output Z_i to the feedforward layer is calculated as:

$$Z_i = \pi_m \otimes XAttn(S_t, S_{m_i}) + \pi_t \otimes Attn(S_t), \quad (4)$$

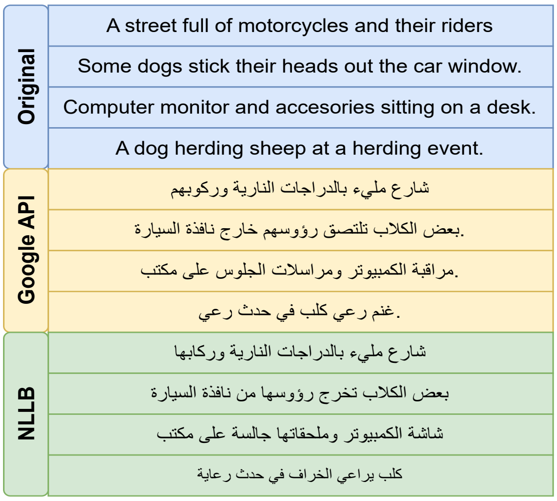
in which π_m is the gating parameter for the vision part and π_t for the text part, calculated as:

$$\begin{aligned} \pi_m &= \sigma(A_n)\mathbb{1}(\sigma(A_n) > \tau), & \forall n \in Attn(S_t) \\ \pi_t &= (1 - \sigma(A_n))\mathbb{1}(1 - \sigma(A_n) > \tau) & \forall n \in Attn(S_t) \end{aligned}$$

where σ is the sigmoid function, A_n is an element in the attention matrix, $\mathbb{1}$ is an indicator function that equals one if the condition is true and zero otherwise, and τ is a hyperparameter. This negates any disturbance caused by weak activations below the threshold τ by zeroing them out. The final output Z to the feedforward layer will be the sum of each encoder-decoder connection weighted by the learned parameter α introduced earlier, mathematically:

$$Z = \frac{1}{\sqrt{L}} \sum_{i=1}^{L} \alpha_i Z_i \quad (5)$$

Where L is the number of encoder layers, set to three in our architecture.



Original	A street full of motorcycles and their riders
	Some dogs stick their heads out the car window.
	Computer monitor and accesories sitting on a desk.
	A dog herding sheep at a herding event.
Google API	شارع مليء بالدراجات النارية وركوبهم
	بعض الكلاب تلتصق رؤوسهم خارج نافذة السيارة.
	مراقبة الكمبيوتر ومراسلات الجلوس على مكتب.
	غنم رعي كلب في حدث رعي.
NLLB	شارع مليء بالدراجات النارية وركابها
	بعض الكلاب تخرج رؤوسها من نافذة السيارة
	شاشة الكمبيوتر وملحقاتها جالسة على مكتب
	كلب يراعي الخراف في حدث رعاية

Figure 4. A comparison between the translations produced by Google translate API and NLLB for MSCOCO dataset. Unlike NLLB, Google API tends to give literal translations without incorporating the context.

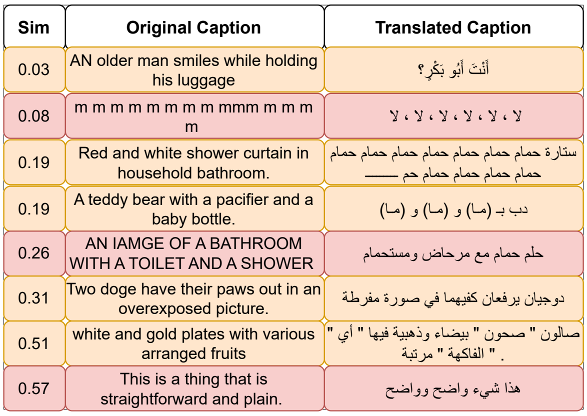
3.2 Data Collection

Owing to the unavailability of high-quality Arabic captioning training data, we first start by creating a training dataset for our model. Manually labeling and creating a new dataset would be both time-consuming and expensive; therefore, we opt for translating the commonly used captioning dataset Microsoft Common Objects in Context (MSCOCO) (Lin et al., 2014). There are two famous training/validation splits for this dataset, the 2014 Karpathy's split, and the 2017 split. Both splits contain the same images and only differ in the split ratio. The dataset covers around 80 different objects in a total of 123k images with 5 captions per image. The dataset is annotated manually, which makes it suitable for evaluation. We create our dataset in two steps, (i) translating the English MSCOCO, followed by (ii) a quality assurance step to filter poor translations.

3.2.1 Machine Translation

In all of the previous attempts at Arabic image captioning pretraining (ElJundi et al., 2020; Sabri, 2021; Emami et al., 2022), Google translate API (Google, 2023) was used for translating the datasets. However, the quality of the translations produced by it is not satisfactory. In Sabri (2021) it is reported that from a random sample of 150 examples, a whooping 46% of the translations obtained by Google API are unintelligible. Motivated by that, we investigate Meta's *No Language Left Behind model (NLLB)* model (Costa-jussà et al., 2022) for translation. Figure 4 illustrates a comparison between the translations produced by the Google Translate API and NLLB for four sentences sampled from MSCOCO dataset.

We conduct our comparison between the two translation models, Google Translate API[2] and NLLB, on two aspects. First, we manually check the quality of 200 sentences translated by both models. Second, we calculate the perplexity of the translations of both models using our JASMINE decoder. Perplexity calculates the probability of a given sequence, providing insight into the fluency of the output translations. Lower perplexity scores indicate better fluency, while higher scores indicate poor fluency. This metric helps us to quantitatively gauge how good the translations are, supplementing our manual evaluation to offer a comprehensive understanding of the models' performance. Subsequently, our observations reveal that the Google API tends to provide a more literal translation in comparison to NLLB. Empirically speaking, we find that 42% of Google's translations are unintelligible, a stark contrast to the mere 15% from NLLB. Interestingly, this observation is consistent with findings presented in Sabri (2021). Furthermore, when pitted against ChatGPT (Ouyang et al., 2022), the latter displays an impressive error rate of only 7% in its translations. However, we opted for NLLB due to its open-source nature.



Sim	Original Caption	Translated Caption
0.03	AN older man smiles while holding his luggage	أَنْتَ أَبُو بَكْرٍ؟
0.08	m m m m m m m m mmm m m m m	لا ، لا ، لا ، لا ، لا ، لا
0.19	Red and white shower curtain in household bathroom.	ستارة حمام حمام حمام حمام حمام حمام حمام حمام حمام حمام حم ـــــــ
0.19	A teddy bear with a pacifier and a baby bottle.	دب بـ (مـا) و (مـا) و (مـا)
0.26	AN IAMGE OF A BATHROOM WITH A TOILET AND A SHOWER	حلم حمام مع مرحاض ومستحمام
0.31	Two doge have their paws out in an overexposed picture.	دوجيان يرفعان كفيهما في صورة مفرطة
0.51	white and gold plates with various arranged fruits	صالون " صحون " بيضاء وذهبية فيها " أي " الفاكهة " مرتبة " .
0.57	This is a thing that is straightforward and plain.	هذا شيء واضح وواضح

Figure 5. Examples of the rejected translations from the dataset and their semantic similarity to the English caption. Where orange highlighting refers to poor translation, and red highlighting refers to poor original caption.

3.2.2 Data Quality Assurance

Although NLLB in general provides better translations compared to that of Google API, it can still output 'hallucinations' and ultimately poor translations. This can be seen in the orange highlighted instances in Figure 5. Moreover, our manual inspection reveals that some English captions in the original dataset are indeed incorrect. The MSCOCO training set can have incomprehensible samples, typos, and even unrelated captions. Examples highlighted in red in Figure 5 illustrate these poor cases. To mitigate this issue, we employ a simple method based on semantic similarity that allows us to identify and reject any such examples.

The *semantic similarity* of two sentences, as the term suggests, is an indicator of the extent to which these two sentences align. A simple comparison between the embeddings of the two sentences can be obtained by passing each of them through a model and a metric such as cosine similarity can be calculated to determine how alike the two embeddings are. The smaller the angle between the two vectors, the higher the similarity score, indicating that the sentences are closer in meaning. When the sentences are in different languages, it is crucial to employ a multilingual model to generate accurate embeddings, ensuring the semantic comparison remains valid across languages. In our experiments, we employ sentence-BERT (Reimers and Gurevych, 2019) to calculate the semantic similarity between each original caption and its translation. We empirically chose a similarity score threshold of 0.6, rejecting all captions below that threshold. This results in removing a total of 60K samples from the whole dataset, which amounts to approximately 10% of the data.

3.2.3 AraCOCO Evaluation Dataset

Evaluating the performance of an Arabic captioning model presents a significant challenge due to the limited availability of human captioned data. To tackle this issue, we manually annotate a subset of 500 images from the MSCOCO test set, dubbing our resulting dataset *AraCOCO*. For each of the 500 images, we acquire five distinct captions. To ensure diversity of image descriptions, we acquire captions from five native Arabic-speaking annotators. The human labeling process is carried out using *Label Studio*, a platform designed for such tasks. Each annotator is presented with the same set of images and is asked to write an Arabic caption describing the image given a unique English caption as a reference. We encourage annotators to provide an Arabic caption that is more descriptive whenever possible. That is, in cases where the

[2] The Google Translate API, integrated into Google Sheets, was used to translate the subset of data utilized in the comparison.

English Caption	NLLB Translation	AraCOCO
An airport with large jetliners and a bus traveling on a tarmac.	مطار مع طائرات كبيرة وحافلة تسافر على المدرج	أشجار النخيل أمام مطار به طائرتان كبيرتان للركاب وحافلات مكوكية.
a group of buses driving around at the airport	مجموعة من الحافلات تسير في المطار	مجموعة من الحافلات تتجول في المطار
Airplanes sit at the gate as transportation vehicles move about.	الجيران تجلس عند البوابة بينما تتحرك مركبات النقل.	طائرات متوقفة عند بوابة المطار وهناك مركبات نقل
A busy runway with buses and luggage carts driving around	مدرج مزدحم مع الحافلات وعربات الأمتعة التي تقود حولها	مدرج مزدحم مع حافلات وعربات أمتعة تتجول
An airplane and busses are lined up at the airport.	طائرة وحافلات منتظمة في المطار	طائرة وحافلات منتظمة في المطار

Table 1: A comparison between original MSCOCO captions (first column), their NLLB translations (second column), and AraCOCO captions (third column) for the image in Figure 6.

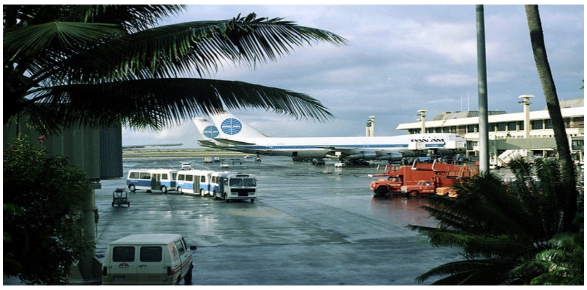


Figure 6. A sample from MSCOCO included in our AraCOCO.

English caption is not capturing all details in the image, annotators are encouraged to capture these lacking details in their Arabic captions. Each annotator gets to provide only one caption per image, This approach ensures having multiple perspectives to the captions on the same image. We provide an example image from AraCOCO in Figure 6, along with five different captions each acquired from one annotator in Table 1.

4 Experiments

We analyze the performance of three variations of our architecture: (i) using the normal decoder with cross-attention in each layer, (ii) using Gemini decoder without freezing the text part, and (iii) using Gemini decoder while freezing the text part. As a baseline, we train a VisualGPT model (Chen et al., 2022) on the English MSCOCO training set then translate output into Arabic using NLLB. Our trained VisualGPT achieves a 117.8 CIDEr score on the English MSCOCO validation set. We conduct our experiments on three datasets, as follows:
(i) Our translated MSCOCO: Following the Karpathy split, our translated and filtered MSCOCO contains 543,817 samples for training (Train), 22,845 samples for validation (Dev), and 22,912 samples for testing (Test). We refer to this dataset as MSCOCO.
(ii) Translated Flickr8K: Similar to the original Flickr8k, the translated dataset introduced in ElJundi et al. (2020) consists of 6,000 images for Train, 1,000 images for Dev, and 1,000 for Test. Each Image has three captions, all translated using Google translate API. We refer to this dataset simply as Flickr8K.
(iii) AraCOCO: As described in Section 3.2.3, AraCOCO consists of 500 images from Karpathy test split. Each image has five captions, all obtained from human annotators.

4.1 Implementation Details

We use JASMINE base (300m) as our text decoder. While for the detection network, following previous works (Li et al., 2020; Cornia et al., 2020; Chen et al., 2022), we employ bottom-up attention network (Anderson et al., 2018) based on Resnet-101 backbone (He et al., 2016) with 2,048 output features. We also limit the maximum number of detections per image to 50 bounding boxes. The three-layer transformer encoder contains 12 attention heads per layer with 768 embeddings dimension.

As we utilize the JASMINE decoder (Nagoudi et al., 2022), we adopt its byte-pair encoding (BPE) vocabulary where frequent character pairs are merged to form subwords. This vocabulary encompasses 63,999 tokens. For data preprocessing, we employ a custom normalizer that removes punctuation and repeated characters.

For the optimization part, in all experiments, we use AdamW Loshchilov and Hutter (2019) with a

Model	BLEU-1 ↑	BLEU-4 ↑	Rouge ↑	CIDEr ↑
VisualGPT	56.2	21.4	44.1	82.1
Violet (w/o Gemini)	45.1	11.3	34.1	41.2
Violet (w/ Gemini)	59.2	21.5	46.3	83.2
Violet (w/ Gemini) *	**60.3**	**24.8**	**47.2**	**84.9**

Table 2: Results on the translated MSCoco test set. VisualGPT is trained by us on the MSCOCO dataset, and the outputs were translated using NLLB (Costa-jussà et al., 2022). (w/o Gemini) means using a normal text decoder with meshed cross-attention in each layer. * indicates freezing the first part of the text decoder.

learning rate of $1e^{-4}$, and empirically set τ to 0.3. The model is trained using a batch size of 60 for 20 epochs while employing early stopping with a patience of 5 on the validation loss. For Flickr8k, we use our MSCOCO-pretrained model and only finetune it for one epoch on Flickr8k's training data. We employ a cross-entropy loss and train the model in an auto-regressive manner, where the decoder predicts the next token given the visual features and the previously generated textual tokens.

4.2 Results and Discussion

We evaluate the performance of our models against previous methods on the popular evaluation metrics BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and CIDEr (Vedantam et al., 2015). The results of our models on our MSCOCO dataset are displayed in Table 2. Our Gemini decoder with six frozen layers (last row in Table 2) achieves better performance while having fewer computations than the unfrozen counterpart. Furthermore, it achieves around three points higher CIDEr score compared to the translated VisualGPT outputs (first row in Table 2). The poor performance observed using the full decoder with cross-attention layers (second row in Table 2), compared to other variants may be due to sensitivity of the decoder parameters which end up being changed significantly with full cross-attention across all its layers.

Model	BLEU-1	BLEU-4	Rouge	CIDEr
Elbedwehy and Medhat (2023)	**58.7**	**16.5**	38.0	46.9
Emami et al. (2022)	39.0	09.0	33.4	42.3
Violet	44.2	13.0	**38.4**	**60.1**

Table 3: Results on Flickr8k test set from (ElJundi et al., 2020). The results are taken from the respective papers.

To compare our Arabic captioning model with previously published Arabic models, we evaluate our model on the Flickr8k test set from (ElJundi et al., 2020). As shown in Table 3, our model achieves 2 points better score on the ROUGE metric, while having a substantial improvement over previous published results in the CIDEr metric. Our model scores 13 points higher than the best model of the two previous models. On the other hand, our model falls behind in BLEU score against Elbedwehy and Medhat (2023). It is worth noting, however, that we are only comparing to published results of Elbedwehy and Medhat (2023) since their model is not available (i.e., not released). They have also used the validation set of Flickr8k in their training, and applied self-critical (Rennie et al., 2017) with no mention of the target data, thus giving their model an advantage over our own model. Regardless, for image captioning, it is known that the CIDEr (where our model excels) is a more relevant evaluation metric than BLEU. Finally, we score our model on our manually annotated dataset, AraCOCO. As shown in Table 4, our model again exhibits sizeable gains compared to our baseline model (i.e., the translated output of VisualGPT). This means that we cannot expect a satisfactory performance by simply taking output from a VLM trained on English data and translating it into Arabic, further corroborating our previous findings and motivating future work on developing VLM models that natively tailored to Arabic language.

Model	BLEU-1	BLEU-4	Rouge	CIDEr
VisualGPT	52.7	17.6	40.2	58.5
Violet	**54.5**	**19.0**	**41.8**	**61.2**

Table 4: Results of our model against translated outputs of VisaulGPT on AraCOCO.

5 Conclusion

In this paper, we introduced *Violet*, an Arabic image captioning model leveraging the pretrained text decoder JASMINE. Our results demonstrated the efficacy of our Gemini decoder in enhancing performance while simultaneously reducing the number of model parameters and computations. We also presented a new method that is effective for acquiring Arabic captioning data from available English data. In addition, we manually annotated a new dataset for evaluating Arabic image captioning models. Our model outperforms all of our baselines and promises to enable benchmarking in this area. We will release our model and datasets to advance Arabic vison-language research.

6 Limitations

Similar to other image detection-based captioning models, the dependence on an external network to provide the visual features introduces an additional layer of complexity to the model. Since the model is not trained end to end, during inference, the visual features must first be obtained from the detection network before passing it to the vision encoder. Another limitation arises from the constraints of the training data. Since MSCOCO focuses solely on 80 class objects, the model's applicability in real-world scenarios is restricted. In our future work, we aim to address both of these limitations to enhance Arabic models' efficiency and broaden their practical usage.

7 Ethics Statement and Broad Impact

Bridging the Gap in Multilingual Image Captioning. Image captioning serves as a crucial bridge between vision and language, with its applications touching numerous domains such as accessibility, education, and search engines. For a long time, the privilege of these advancements has been constrained to a handful of languages, primarily due to the lack of necessary datasets and dedicated research in other languages. Arabic, with its vast speakers and rich history, has unfortunately been left behind in this domain. Our work with *Violet* seeks to rectify this disparity, providing a robust foundation for Arabic image captioning. By releasing *Violet* and the datasets, we aim to invigorate research in this direction, promoting inclusivity and equal opportunity in NLP and computer vision advancements across languages.

Automated Data Acquisition and Transparency. To overcome the challenge of limited labeled data for Arabic image captioning, we employed a novel method for data acquisition using available English datasets. While this approach provides a solution, it also warrants a discussion on the accuracy, bias, and quality of the automatically acquired data. We emphasize that while our method provides a foundational dataset, manual annotations and human evaluations remain paramount for ensuring data quality and avoiding propagation of errors.

Acknowledgment of Data Sources and Fair Credit. Similar to ensuring proper credit assignment for benchmarking tasks, we emphasize the importance of acknowledging the original data sources we leveraged, especially in the context of automated data acquisition. Users and researchers utilizing our datasets and model are encouraged to cite and acknowledge the original datasets and sources. This practice ensures that original creators receive the recognition they deserve and promotes a culture of transparency and fairness in the research community.

Acknowledgments

We acknowledge support from Canada Research Chairs (CRC), the Natural Sciences and Engineering Research Council of Canada (NSERC; RGPIN-2018-04267), the Social Sciences and Humanities Research Council of Canada (SSHRC; 435-2018-0576; 895-2020-1004; 895-2021-1008), Canadian Foundation for Innovation (CFI; 37771), Digital Research Alliance of Canada,[3] and UBC ARC-Sockeye.[4] We also thank Samar Magdy, Ahmed Omar, and Karima Kadaoui for their help in creating the manually captioned subset of MSCOCO, which we refer to as AraCOCO.

References

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

Huda A Al-Muzaini, Tasniem N Al-Yahya, and Hafida Benhidour. 2018. Automatic arabic image captioning using rnn-lstm-based language model and cnn. *International Journal of Advanced Computer Science and Applications*, 9(6).

[3] https://alliancecan.ca

[4] https://arc.ubc.ca/ubc-arc-sockeye

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. 2022. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736.

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086.

Anfal Attai and Ashraf Elnagar. 2020. A survey on arabic image captioning systems using deep learning models. In *2020 14th International Conference on Innovations in Information Technology (IIT)*, pages 114–119. IEEE.

Muhammad Awais, Muzammal Naseer, Salman Khan, Rao Muhammad Anwer, Hisham Cholakkal, Mubarak Shah, Ming-Hsuan Yang, and Fahad Shahbaz Khan. 2023. Foundational models defining a new era in vision: A survey and outlook. *arXiv preprint arXiv:2307.13721*.

Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. 2021. GPT-Neo: Large scale autoregressive language modeling with meshtensorflow.

Jun Chen, Han Guo, Kai Yi, Boyang Li, and Mohamed Elhoseiny. 2022. Visualgpt: Data-efficient adaptation of pretrained language models for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18030–18040.

Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara. 2020. Meshed-memory transformer for image captioning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10578–10587.

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Samar Elbedwehy and T Medhat. 2023. Improved arabic image captioning model using feature concatenation with pre-trained word embedding. *Neural Computing and Applications*, pages 1–17.

Obeida ElJundi, Mohamad Dhaybi, Kotaiba Mokadam, Hazem M Hajj, and Daniel C Asmar. 2020. Resources and end-to-end neural network models for arabic image captioning. In *VISIGRAPP (5: VISAPP)*, pages 233–241.

Jonathan Emami, Pierre Nugues, Ashraf Elnagar, and Imad Afyouni. 2022. Arabic image captioning using pre-training of deep bidirectional transformers. In *Proceedings of the 15th International Conference on Natural Language Generation*, pages 40–51.

Zhe Gan, Linjie Li, Chunyuan Li, Lijuan Wang, Zicheng Liu, Jianfeng Gao, et al. 2022. Vision-language pretraining: Basics, recent advances, and future trends. *Foundations and Trends® in Computer Graphics and Vision*, 14(3–4):163–352.

Google. 2023. Google translate api. Accessed: 15/07/2023.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Lun Huang, Wenmin Wang, Jie Chen, and Xiao-Yong Wei. 2019. Attention on attention for image captioning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4634–4643.

Shaohan Huang, Li Dong, Wenhui Wang, Yaru Hao, Saksham Singhal, Shuming Ma, Tengchao Lv, Lei Cui, Owais Khan Mohammed, Qiang Liu, et al. 2023. Language is not all you need: Aligning perception with language models. *arXiv preprint arXiv:2302.14045*.

Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. 2014. Caffe: Convolutional architecture for fast feature embeding. *arXiv preprint*, arXiv:1408.5093.

Vasu Jindal. 2017. A deep learning approach for arabic caption generation using roots-words. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Andrej Karpathy, Armand Joulin, and Li F Fei-Fei. 2014. Deep fragment embeddings for bidirectional image sentence mapping. *Advances in neural information processing systems*, 27.

Moaz T Lasheen and Nahla H Barakat. 2022. Arabic image captioning: the effect of text pre-processing on the attention weights and the bleu-n scores. *Int J Adv Comput Sci Appl*, 13(7):11.

Siming Li, Girish Kulkarni, Tamara Berg, Alexander Berg, and Yejin Choi. 2011. Composing simple image descriptions using web-scale n-grams. In *Proceedings of the fifteenth conference on computational natural language learning*, pages 220–228.

Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. 2020. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX 16*, pages 121–137. Springer.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. *In ICLR*.

El Moatez Billah Nagoudi, Muhammad Abdul-Mageed, AbdelRahim Elmadany, Alcides Alcoba Inciarte, and Md Tawkat Islam Khondaker. 2022. Jasmine: Arabic gpt models for few-shot learning. *arXiv preprint arXiv:2212.10755*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

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.

Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. 2015. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7008–7024.

Sabri Monaf Sabri. 2021. *Arabic image captioning using deep learning with attention*. Ph.D. thesis, University of Georgia.

Matteo Stefanini, Marcella Cornia, Lorenzo Baraldi, Silvia Cascianelli, Giuseppe Fiameni, and Rita Cucchiara. 2022. From show to tell: A survey on deep learning-based image captioning. *IEEE transactions on pattern analysis and machine intelligence*, 45(1):539–559.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164.

Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. 2022. Git: A generative image-to-text transformer for vision and language. *arXiv preprint arXiv:2205.14100*.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR.

Yezhou Yang, Ching Teo, Hal Daumé III, and Yiannis Aloimonos. 2011. Corpus-guided sentence generation of natural images. In *Proceedings of the 2011 conference on empirical methods in natural language processing*, pages 444–454.

Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. 2016. Image captioning with semantic attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4651–4659.

Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. 2022. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*.

Nabra: Syrian Arabic Dialects with Morphological Annotations

Amal Nayouf
Syrian Virtual University, Syria
amal_124724@svuonline.org

Tymaa Hasanain Hammouda
Birzeit University, Palestine
thammouda@birzeit.edu

Mustafa Jarrar
Birzeit University, Palestine
mjarrar@birzeit.edu

Fadi A. Zaraket, zfadi@utexas.edu
Doha Institute for Graduate Studies, Doha
American University of Beirut, Beirut

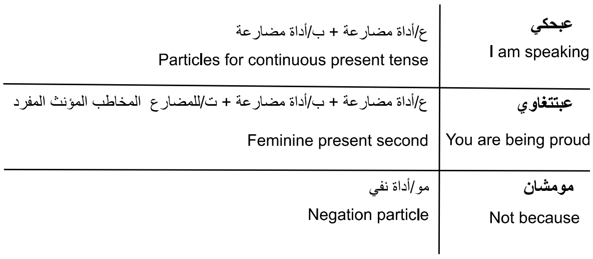
Mohamad-Bassam Kurdy
Syrian Virtual University, Syria
t_bkurdy@svuonline.org

Abstract

This paper presents Nâbr̄a (نَبْرَة), a corpora of Syrian Arabic dialects with morphological annotations. A team of Syrian natives collected more than $6K$ sentences containing about $60K$ words from several sources including social media posts, scripts of movies and series, lyrics of songs and local proverbs to build Nâbr̄a. Nâbr̄a covers several local Syrian dialects including those of Aleppo, Damascus, Deir-ezzur, Hama, Homs, Huran, Latakia, Mardin, Raqqah, and Suwayda. A team of nine annotators annotated the $60K$ tokens with full morphological annotations across sentence contexts. We trained the annotators to follow methodological annotation guidelines to ensure unique morpheme annotations, and normalized the annotations. F1 and κ agreement scores ranged between 74% and 98% across features, showing the excellent quality of Nâbr̄a annotations. Our corpora are open-source and publicly available as part of the Currasat portal https://sina.birzeit.edu/currasat.

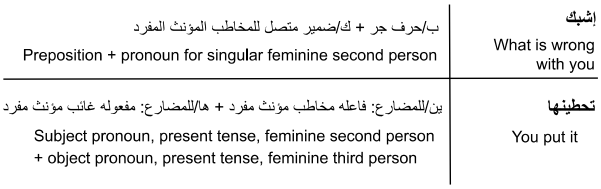
1 Introduction

Dialectal Arabic (DA) content dominates informal writings in emails, social media, blogs, and social messaging. Interest in building computational resources for Arabic dialects has been in the rise to provide both (i) annotated corpora (Jarrar et al., 2022b; Alshargi et al., 2019; Khalifa et al., 2018; Bouamor et al., 2018; Jarrar et al., 2017; Al-Shargi et al., 2016; Zribi et al., 2015; Jarrar et al., 2014) and (ii) morphological dialect analyzers (Obeid et al., 2020; Khalifa et al., 2020; Pasha et al., 2014; Zribi et al., 2017; Abdul-Mageed et al., 2021).



عبحكي I am speaking	ع/أداة مضارعة + ب/أداة مضارعة Particles for continuous present tense
عبتتغاوي You are being proud	ع/أداة مضارعة + ب/أداة مضارعة + ت/للمضارع المخاطب المؤنث المفرد Feminine present second
مومشان Not because	مو/أداة نفي Negation particle

Figure 1: Examples of typical prefixes in Syrian dialects



إشبك What is wrong with you	ب/حرف جر + ك/ضمير متصل للمخاطب المؤنث المفرد Preposition + pronoun for singular feminine second person
تحطينها You put it	ين/للمضارع: فاعله مخاطب مؤنث مفرد + ها/للمضارع: مفعوله غائب مؤنث مفرد Subject pronoun, present tense, feminine second person + object pronoun, present tense, feminine third person

Figure 2: Examples of typical suffixes in Syrian dialects.

In this paper, we present Nâbr̄a نَبْرَة, a set of corpora that complement existing Arabic dialect corpora by covering several dialect variants of Syrian Arabic. Nâbr̄a covers dialects from 10 Syrian localities including Aleppo, Damascus (a.k.a. Shami) , Deir-ezzur, Hama, Homs, Huran, Latakia, Mardin, Raqqah, and Suwayda. Nâbr̄a was collected from several sources including social media posts, scripts of movies and series, lyrics of songs, and local proverbs. Nine annotators worked on annotating 6K sentences with 60,021 tokens with full morphological annotations. Each word was annotated using: prefix(s), stem, and suffix(s), part of speech (POS), dialect lemma, MSA lemma, person, number, gender, gloss, and synonyms; in addition to the sub-dialect it belongs to.

We adopted the same annotation methodology used to annotate the Palestinian Curras2 and the Lebanese Baladi corpora (Haff et al., 2022), which we also used with the four corpora of Lisan (Jarrar et al., 2023b). As we will discuss later, we adopted the SAMA tagsets (Maamouri et al., 2010), but we introduced new prefixes and suffixes that are commonly used in Syrian dialects (Figures 1 and 2).

1.1 Arabic and its Dialects

Over 300 million people speak Arabic, including Classical Arabic (CA), Modern Standard Arabic (MSA), and dialectal forms of Arabic (DA), in

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 12–23
December 7, 2023 ©2023 Association for Computational Linguistics

more than 23 countries. Natural language processing (NLP) research has traditionally focused on MSA because it is the most widely used form of Arabic in formal communication, newspapers, education, and media. CA dominates historical and cultural texts, whereas most colloquial and real-life communication uses local DA variants. DA content is lately gaining massive growth especially through blogs, social media, and local entertainment outlets in songs, movies, and series.

NLP pipelines often struggle with tasks involving DA content due to the inherent morphological richness of DA variants, their relative lack of resources compared to MSA, and the absence of a standardized orthography (Darwish et al., 2021). DA is classified regionally into Egyptian, Gulf, Levantine, North African, and Yemeni (Diab et al., 2010) with Syrian and Lebanese dialects considered as Northern Levantine, and Palestinian and Jordanian as Southern Levantine.

Syrian Arabic is well-understood across the Arab world due to its popularity in historical dramas, TV series, and soap operas. Twenty million Syrians speak it for daily life. Expatriates from the Levant (Jordan, Lebanon, Palestine, and Syria) helped spread the dialect throughout the world.

The rest of this paper is organized as follows. Section 2 reviews related work. We introduce Syrian as a Levantine dialect in Section 3 and discuss variant Syrian dialects in Section 4. Nâbra data collection and annotation methodology follow in Sections 5 and 6, respectively. We discuss the evaluation of Nâbra in Section 7, then we conclude in 8 and discuss limitations and ethics considerations.

2 Related work

There are several annotated corpora and lexicographic resources for MSA.

The LDC's Penn Arabic Treebank PATB (Maamouri et al., 2005) consists of about consists of 791,210 tokens collected from several news sources. PATB annotations include: tokenization, segmentation, POS tagging, lemmatization, diacritization, English gloss and syntactic structure. The LDC Ontonotes 5 (Weischedel et al., 2013) is another MSA corpus collected from news sources, consisting of about 330K tokens, which are annotated in the same way as the PATB. Ontonotes 5 also contains multiple layers of annotation, including the PATB annotation layer.

The Prague Arabic Dependency Treebank (Ar-PADT) (Hajič et al., 2004) is a treebank that contains morphological annotations for a corpus of MSA text. These annotations include lemmas, part-of-speech tags, and other morphological features. Ar-PADT contains about 224K words.

The LDC's SAMA is a stem database (Maamouri et al., 2010), which is an extension of BAMA (Buckwalter, 2004), designed only for morphological modeling. It contains stems and their lemmas and compatible affixes. It contains about 40K lemmas.

The lexicographic database at Birzeit University (Jarrar and Amayreh, 2019) provides a large set of MSA lemmas, word forms, and morphological features, which are linked with the Arabic Ontology (Jarrar, 2021) using the W3C LEMON model (Jarrar et al., 2019).

2.1 Dialectal Arabic Resources

There are several Arabic dialectal corpora with diverse morphological annotations.

An early pilot to build a Levantine Arabic Tree bank is presented in (Maamouri et al., 2006). The Palestinian dialect corpus Curras (Haff et al., 2022; Jarrar et al., 2017, 2014) comprises about $56K$ tokens. Each word in the Curras was annotated with different morphological features, including Prefixes, Stem, Suffixes, MSA lemma, Dialect Lemma, Gloss, POS, Gender, Number, and Aspect. The Lebanese Baladi corpus ($9.6K$ tokens) was developed in the same manner as Curras in order to form a more Levantine corpus (Haff et al., 2022).

CALLHOME (Canavan et al., 1997) is an Egyptian Arabic corpus with transcripts of telephone conversations in Egyptian. CALIMA (Maamouri et al., 2006) extended ECAL (Kilany et al., 2002) which built on CALLHOME to provide morphological analysis of Egyptian. The COLABA project (Diab et al., 2010) collected Egyptian and Levantine resources from online blogs leading to the construction of Egyptian Tree Bank (ARZATB) (Maamouri et al., 2014).

The Lisan (Jarrar et al., 2022b) consists of 1.2 million tokens, covering Iraqi, Yemeni, Sudanese, and Libyan dialects. The Yemeni corpus (about 1.05M tokens) was collected automatically from Twitter, while the other three dialects (about 50K tokens each) were manually collected from Facebook and YouTube. Each word in the four corpora was annotated with different morphological features, such as POS, stem, prefixes, suffixes, lemma, and a gloss in English.

A corpus of $200K$ tokens was morphologically annotated covering seven different Arabic dialects including Taizi, Sanaani, Najdi, Jordanian, Syrian, Iraqi, and Moroccan (Alshargi et al., 2019). The GUMAR Emirati corpus (Khalifa et al., 2018) consists of 200K tokens collected from novels. MADAR (Bouamor et al., 2018) is an ongoing multi-dialect corpus covering 26 cities and their corresponding dialects. The Arabizi Tunisian corpus has 42K tokens (Gugliotta and Dinarelli, 2022).

The NADI (nuanced Arabic dialect identification) SharedTask (Abdul-Mageed et al., 2021, 2020) provided researchers with 10-million/21K unlabeled/labeled tweets and challenged researchers to identify the province-level dialects across 21 countries.

3 Syrian as a Levantine Dialect

The Levantine family of dialects can be linguistically split across the north including Lebanon and Syria, and the south including Palestine and Jordan. During the seventh century, Arabic spread across the area, which spoke Western Aramaic before then (Skaf, 2015).

Aramaic is a Semitic language continuum spoken during antiquity throughout the Levant where It served as the *lingua-franca*. Aramaic survives today through modern dialects such as Turoyo Syriac and Western Neo-Aramaic spoken in parts of Syria. It also survives more subtly in the noticeable substratum underlying Levantine dialects that differ from MSA on several linguistic characteristics such as phonology, syntax, morphology, and lexicon. This additionally motivates the development of morphologically annotated resources for Levantine dialects. In the sequel, we briefly review the differentiating factors between Levantine dialects, Syrian dialects, and MSA.

3.1 Levantine Phonology

Aramaic variants use the Abjad alphabet composed of 22 letters. When Arabic spread, the population of the region transcribed Arabic with its 28 letters using the 22-letter Abjad resulting in “Garshouni”, a Syriac writing tradition (Briquel Chatonnet, 2005). Adaptations to fit the additional letters led some Syriac graphemes to represent multiple phonemes of Arabic, especially some of the emphatic letters.

3.2 Syrian Phonology and Orthography

The Syrian Dialect has a glottal stop phoneme /ʔ/ that is cognate with either Hamza ء إ أ ؤ ئ /ʾ or Qaf ق /*q* . In spontaneous Syrian orthography, the two forms are distinguished in a manner similar to Lisan guidelines (Jarrar et al., 2023b). Exceptions include هلأ /*hlʾa* (now) written هلق /*hlq* in $Token$ with normalization rules to highlight its etymology link to هالوقت /*hālwqt* (this time). Less common spelling variations include devoicing ج /*ǧ* /ʒ/ to /ʃ/,which sometimes reflects in spontaneous orthography, e.g., نجتمع /*nǧtmʿ* /niʒtmiʔ/ (we meet) may appear as نشتمع /*nštmʿ* /niʃtimʕ/.

3.3 Levantine Morphology

Levantine inherits templatic morphology from Semitic languages where affixes play important roles. Several morphological differences exist when compared to MSA.

- Diacritic marking for syntax roles is less required in Levantine. They are marked with suffixes resulting in similar phonetic effects. For example, there is no need for writing Dhamma ـُ /*u* to distinguish the subject from the object. The MSA sentence غلب البطلُ الأسدَ /*ġlb ālbṭlu alʾasda* (The hero conquered the lion) may switch the subject and object as in غلب الأسدَ البطلُ /*ġlb ālʾasda ālbṭlu* and the diacritics distinguish the roles. The Levantine variants are البطل غلب الأسد /*ālbṭl ġlb ālʾasd* and الأسد غلبو البطل /*ālʾasd ġlbw ālbṭl* (also written as الأسد غلبه البطل /*ālʾasd ġlbh ālbṭl*) with no need for diacritics.
- Some Levantine-specific morphemes do not exist in MSA such as عم /*ʿm* which denotes present continuous tense when it precedes imperfect verbs أنا عم باكل /*ʾanā ʿm bākl* (I am eating). Without it أنا باكل /*ʾanā bākl* means the general truth (I eat). MSA lacks such an indicator and the tense is inferred from context: أنا آكل /*ʾanā ʾākl* can mean both "I am eating” or "I eat”.
- Other morphemes include رح /*rḥ* and ح /*ḥ* that are Levantine future indicators compared to MSA’s س /*s* and سوف /*swf* . (iv) The progressive Levantine particle بـ /*b* (as in باكل /*bākl*) indicates imperfective verbs and no counterpart exists in MSA.

Syrian dialects lack the negation enclitic ش /*š* in a distinction from southern Levantine dialects. Syrian dialects make use of a number of future particles in free distribution. The progressive particle عم /*ʿm* strictly indicates active momentarily progression, while the progressive proclitic +ب /*b* indicates a wider habitual to the progressive range.

3.4 Levantine Dialect Lexicon

The Levantine lexicon is rich with loan words from other languages due to its cross-civilization frequent passage location.

Some Syrian words are originally Syriac, e.g., شوب /*šwb* (hot), or براني /*brāny* (outer). Other words are originally Turkish, e.g., دغري /*dġry* (straightforward). Some words encountered major semantic shifts, e.g., طز /*ṭz* comes from Turkish tuz for 'salt', then semantically shifted to mean 'something unimportant', and eventually 'good riddance'. Other words were borrowed from French, e.g., ديكور /*dykwr* (decor) and جاتو /*ğātw* (gateaux), and from Persian, e.g., سرسري /*srsry* (badman). Military terms كورنيت /*kwrnyt* are used to specify accuracy and sharpness.

4 Variant Syrian Dialects

Syrian Arabic dialects are used in daily communication among most Syrians. Some of them are closer to Iraqi dialects, and the rest are closer to the Levantine southern Levantine dialects. Here, we review the most famous dialects spoken in Syria.

The Shami dialect is the dominant dialect in the Damascus area and is the most widespread and used Syrian dialect. As the dialect of the capital, it dominates Syrian series and films which are widely accepted, appreciated, and spread in the Arab world. It is used in dubbing and translation of foreign series (Turkish and Hindi).

Table 1 shows Shami dialect features:

- Sculpture: abbreviate two or more words.
- Substitution: an example is the replacement of ق /*q* with ء /ʾ hamza.
- Spatial inversion: the introduction or delay of letters to simplify pronunciation.
- Inclination: vowel exchange where ا /*ā* is pronounced ي /*y* .

The Aleppo dialect is dominant in Aleppo in northern Syria. It is distinctive in pronunciation and has a unique vocabulary used in Aleppo alone. The distinct vocabulary comes from ancient Syriac or Turkish. Examples of Syriac and Turkish vocabulary used in Aleppo follow. Syriac إيمت /*ʾiymt* replaces MSA متى /*mtā* (when), and Syriac دعك /*dʿk* replaces MSA عجن /*ʿğn* (knead). Turkish فرتيكة /*frtykh* and سكرتون /*skrtwn* replace MSA شوكة /*šwkh* (fork), خزانة /*ẖzānh* (closet), respectively.

Shami	MSA	Gloss	Rule
شو بدّك *šw bdk*	أي شيء بودّك *ʾay šyʾ bwdk*	what do you want?	النحت Sculpture
بالمشرمحي *bālmšrmḥy*	بكلام عربي واضح وفصيح *bklām ʿrby wāḍḥ wfṣyḥ*	In clear words	النحت Sculpture
أديش *ʾadyš*	كم يساوي *km ysāwy*	how much	ابدال Substitution
جوز *ğwz*	زَوْج *zawğ*	husband	قلب المكاني spatial inversion
هنيك *hnyk*	هناك *hnāk*	There	إمالة inclination

Table 1: Examples of Shami Dialect

With non-Arabic Syriac vowels (e, o), Aleppo words and verbs do not need the Dammah ـُ (nourishing) and fatha ـَ (accusative) diacritics. Verbs may require more than one object denoting the concept of تعدي /*tʿdy* (exceeds). Verbs connect to ن to denote the masculine plural instead of the MSA suffix م /*m* Turkish influence on Aleppo dialects morphs the pronunciation of fixed letters such as ج /*ğ* and ق /*q* to a majestic Turkish tone, and also reduces the pronunciation of vowels.

The Latakia dialect is spoken across the coast in Latakia and Tartous. It is a mixture of Arabic, Syriac, and Phoenician. It is characterized by the strong pronunciation of the letter ق /*q* , and also features the letter م /*m* before verbs to denote the present tense in all its forms, e.g.منكتب /*mnktb* (we write/are writing), ميدرس /*mydrs* (he studies/is studying).

The Raqqa dialect is one of the closest dialects to classical Arabic in terms of vocabulary. Raqqa enjoys a distinguished location on the shores of the Euphrates River. It is home (ديار /*dyār*) Mudar, who are Arabs from the north. Mudar were displaced to the Euphrates island several centuries before Islam. The Raqqa syllables sound commensurate to the corresponding classical Arabic syllables. For example, the pronunciation of ک /*k* results in a thirsty ج /*ğ* as in کانت /*kānt* pronounced as جانت /*ğānt* . The letter ق /*q* is pronounced ک /*k* similar to Yemeni dialects as in قاع /*qāʿ* (earth) pronounced as کاع /*kāʿ*.

The Deir-ezzur dialect aka. as الديرية /*āldyryh* is in proximity to the Euphrates as well, and preserves most of the phonetic aspects of standard Arabic. The significantly different phonemes are ق /*q* , ك /*k* and ء /ʾ, while there is no different in the gingival sounds.

The Homs dialect varies slightly across several rural and urban areas in the Homs district. This is mainly due to the habitual diversity of the countryside including a sizeable Turkman population.

This paper covers the dominant variant in the city of Homs. The Homs dialect is characterized by pronouncing the first letter in a word as if it has a Dammah ـُ /*u* diacritic (inclusion). This includes the name of the city حِمص /*ḥimṣ* , pronounced with a Kasra إِ /*i* dialect everywhere else. It also flips gender when it comes to masculine second-person إنتِ /*ʾinti* (you-male in Homsi) and feminine second person إنتَ /*ʾinta* (you-female in Homsi). It also differs in the pronunciation of the letter ج /*ǧ* as they phonetically annex a silent د /*d* resulting in a دج /*dǧ* sound.

The Hama dialect is spoken in the central Syrian governorates. It is a good representative of the Syrian Levantine dialects and close to the Shami one, as it tends to be soft and long in speech. It is distinguished by its eloquence and stretch in speech. Al-Hader (city in Hama) variant of the Hama dialect is the most prominent variant.

The Hauran dialect is spoken south of the Damascus countryside down to the Ajloun mountains in Jordan including Daraa. It is an ancient Arabic dialect spoken by multiple Arab tribes, where each of them has some distinguishing phonetic characteristics.

The Al-Suwayda dialect is spoken in Jabal al-Arab. The harshness of the mountain environment is reflected in the dialect's tone. It is taut, clear, and possesses a fast rhythm. Syllable notes exit soundly and eloquently. The concept of المضافة /*ālmḍāfh* played a major role in preserving the strength of the dialect. Therein, prominent, cultured, and experienced speakers exchange arguments. This highly contributed to the rigor of the dialect and brought it closer to standard and classical Arabic.

The Mardini dialect takes its name from the city of Mardin in الحسكة /*ālḥskh* . It is also called الجزراوية /*ālǧzrāwyh* in relevance to the الفراتية /*ālfrātyh* island. The dialect contains many Turkish, Persian, and Aramaic words.

5 Nâbīa Corpora Collection

We manually collected about 6,000 sentences with 60K tokens from Facebook, blogs, popular proverbs, Syrian films and series, local poetry, and lyrics of popular local songs in several Syrian dialects to build Nâbīa. Table 2 provides statistics on tokens, unique tokens, sentences, lemmas, nouns, verbs, and functional words in each of the 10 dialects Nâbīa covers.

The distribution relatively follows the order of dialect demographics. The Shami dialect is the richest with 17.3K tokens, used as primary dialect in Damascus, the capital, and in various Syrian TV series and films. Nâbīa contains 9.2K Aleppo tokens collected from popular stories on Facebook and from vocal poetry. Coastal Latakia features 7.9K tokens collected from film dialogues such as رسايل شفهية /*rsāyl šfhyh* - قمران وزيتونة /*qmrān wzytwnh* (Voice letters, Qumran and Zeitouna) and series such as ضيعة ضايعة /*ḍyʿh ḍāyʿh* (lost town). We also added common proverbs. Suwayda dialect features 3.2K tokens from the الخربة /*ālẖrbh* series. For Homs and Hama we collected jokes, and food discussions from social media blogs.

The Raqqa, Huran, and Mardin dialects feature the remaining 6.3K, 3.8K, and 1.6K tokens, respectively. We manually collected texts from social media for Raqqa and Huran. We found blogs documenting Raqqa. We used blogs and traditional stories for Raqqa, vocal poetry and lyrics of popular folklore songs for Mardini, and scenes from the Bedouin series for Huran dialects. We noticed that the collected data reflected spontaneous dialect documentation all across, contrary to what one would expect. Films and series were no less spontaneous than blogs and social media.

As Arabic is diacritic-sensitive (Jarrar et al., 2018), we did not remove any diacritics We tokenized the text of Nâbīa so that each token has a tuple with the following information.

⟨SentenceID, TokenID, TokenText, Local-DialectName, Governate⟩

6 Annotation Methodology and Features

We followed a semi-automated methodology, with an integrated productivity tool, friendly to non-programmers, to annotate Nâbīa.

6.1 Methodology

We developed the *Tawseem* annotation portal to help automate and validate the annotation process. The portal leverages spreadsheets, familiar to common users, and is powered by smart functionalities to improve annotation productivity. Figure 3 shows a snapshot of *Tawseem* annotation portal with the sentence شلون دا تدخلي تسلمي عالنفسا /*šlwn dā tdẖly tslmy ʿālnfsā* (how would you enter to greet someone in childbed).

For each token in the sentence, the portal saves 17 data elements. The $SentenceID$ and $TokenID$ columns identify the sentence and token.

Dialect لهجة	Damascus (Shami) شامية	Aleppo حلبية	Latakia ساحلية	Raqqa رقاوية	Deir-Ezzur ديرية	Homs حمصية	Huran حوران	Suwayda سويداء	Hama حموية	Mardin ماردلية
Tokens	17,274	9,255	7,893	6,284	4,322	4,139	3,807	3,150	2,322	1,575
Unique Tokens	7,123	4,452	3,829	3,389	2,453	2,047	2,094	1,681	1,355	949
Sentences	1,181	787	829	679	519	518	457	381	340	243
Unique MSA Lemma	4,230	2,825	2,548	2,367	1,909	1,543	1,580	1,312	1,051	686
Unique DA lemma	4,351	2,969	2,681	2,490	1,954	1,591	1,646	1,354	1,095	710
Nouns	7,700	4,251	3,771	3,316	2,384	2,064	2,090	1,527	1,135	694
Verbs	3,524	1,897	1,557	985	714	709	518	554	369	339
Functional Words	6,027	3,090	2,560	1,960	1,213	1,359	1,194	1,069	815	534

Table 2: Counts of tokens, unique tokens, sentences, unique MSA lemmas, unique dialectal lemmas, Nouns, Verbs, and functional words for each of the Syrian dialects

The rest of the columns specify the $rowToken$, $Token$, $prefix(s)$, $stem$, $suffix(s)$, POS, $gender$, $number$, $person$, $aspect$, $MSAlemma$, $dialectlemma$, $synonym(s)$, $gloss$, as well as the $sub-dialect$.

To simplify and accelerate the annotation process we leverage existing annotations in the following manner. First, we uploaded existing annotated corpora for dialects and MSA (Haff et al., 2022; Jarrar et al., 2023b) into the $Tawseem$ tools.

The tool allows the annotators to search and look up previous annotations. The lookup services search the database and return the top matching results ranked. Annotators can then select one of the results, and correct the corresponding features if needed.

Second, annotators can search the *Tawseem* portal annotations in other sentences whether made by themselves or by other annotators. This helps leverage previous annotations and improves the correction process. Additionally, annotators can look for existing annotations of a specific token in the *Tawseem* portal results.

6.2 Annotation Guidelines

Training annotators to use the *Tawseem* portal was straightforward as they were all familiar with the interface of a productivity spreadsheet. We also trained them with annotation guidelines for each of the features in Nâbra as follows:

rowToken: $rawToken$ is the raw word as it appears in the corpus, without any modification.

Token : $Token$ is the normalized version of the $rawToken$. This entry corrects spelling errors if needed. The idea is to unify different forms of spelling the same word with one specification to mitigate the lack of spelling rules for Arabic dialects. It is necessary to unify the different ways one word can be written by multiple users to reflect the same pronunciation. We adopted the $Token$ guidelines used in the Lisan corpora (Jarrar et al., 2023b) as well as the Palestinian Curras2 and Lebanese Baladi corpora (Haff et al., 2022) so that Nâbra can be included smoothly in a larger family of Arabic dialects for further research and applications if needed.

Dialect lemma (المدخلة المعجمية العامية) determines the dialect's original source of the token. Thus, if the word is a verb, we choose the past masculine 3rd person singular form as its colloquial origin. For nouns, we select the singular masculine, if not attained we select the singular feminine form. When introducing a new lemma, we specify the following: (i) definitions of senses in Arabic, which is important for word sense disambiguation tasks (Al-Hajj and Jarrar, 2021a; Jarrar et al., 2023a) and Word-in-Context WiC disambiguation tasks (Al-Hajj and Jarrar, 2021b). (ii) Equivalent lemmas in MSA (Jarrar et al., 2019, 2021).

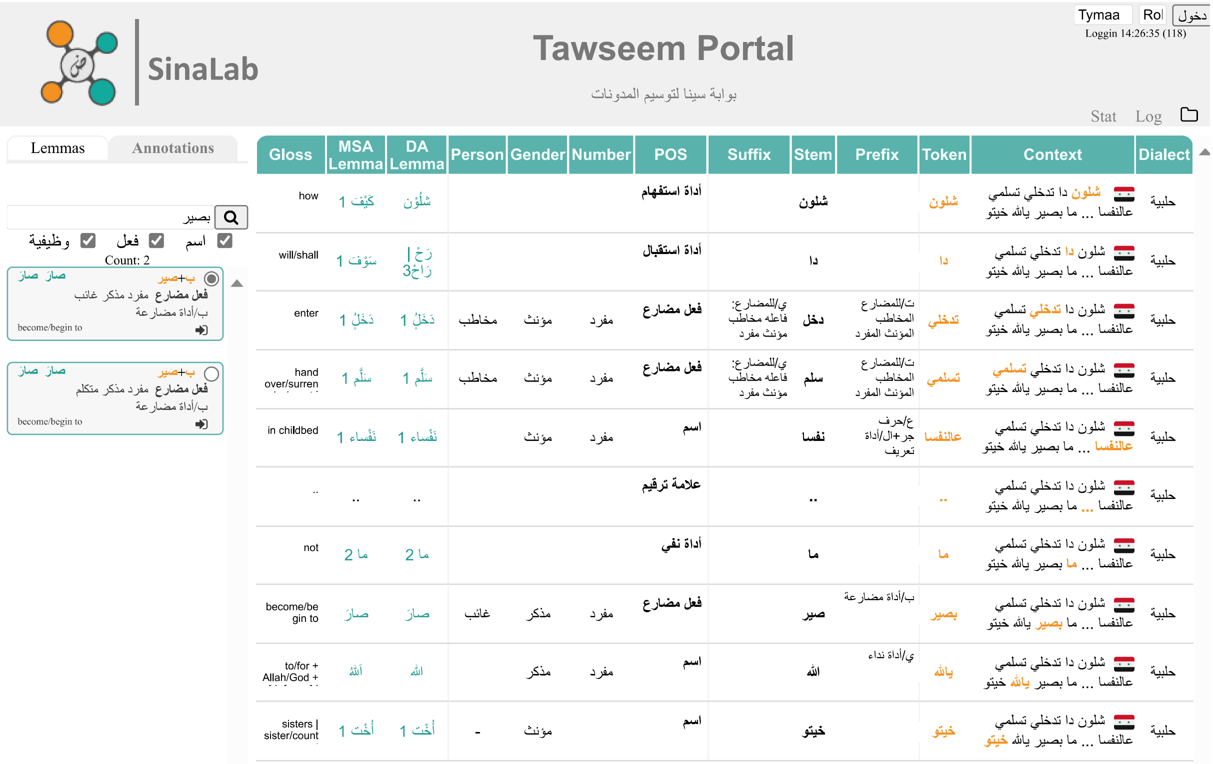
MSA Lemma (المدخلة المعجمية الفصحى) determines the MSA original source of the token. Table 3 shows examples of some tokens with their $Token$, and dialect and MSA lemmas.

The $Tawseem$ portal allows to search for lemmas in the Birzeit's Lexicographic database (Jarrar and Amayreh, 2019; Alhafi et al., 2019) and Arabic Ontology (Jarrar, 2021, 2011); otherwise, we introduced a new lemma.

The Synonym (المرادف) feature provides synonyms for the token and sometimes explains the token semantics. We used an online tool for automatic synonym discovery (Ghanem et al., 2023; Khallaf et al., 2023).

Gloss (المعنى بالانجليزية) specifies the meaning of the token in English. It typically specifies a short definition of lemma semantics. See an elaboration on the gloss formulation guidelines in (Jarrar, 2006).

POS (قسم الكلام) specifies the part of speech of the token. This concerns the grammatical category



Gloss	MSA Lemma	DA Lemma	Person	Gender	Number	POS	Suffix	Stem	Prefix	Token	Context	Dialect
how	كَيْفَ 1	شْلُوْن				أداة استفهام		شلون		شلون	شلون دا تدخلي تسلمي عالنفسا ... ما بصير يالله خيتو	حلبية
will/shall	سَوْفَ 1	رَحْ | رَاحْ3				أداة استقبال		دا		دا	شلون دا تدخلي تسلمي عالنفسا ... ما بصير يالله خيتو	حلبية
enter	دَخَلَ 1	دَخَلَ 1	مخاطب	مؤنث	مفرد	فعل مضارع	ي/للمضارع: فاعله مخاطب مؤنث مفرد	دخل	ت/للمضارع المخاطب المؤنث المفرد	تدخلي	شلون دا تدخلي تسلمي عالنفسا ... ما بصير يالله خيتو	حلبية
hand over/surren	سَلَّمَ 1	سَلَّمَ 1	مخاطب	مؤنث	مفرد	فعل مضارع	ي/للمضارع: فاعله مخاطب مؤنث مفرد	سلم	ت/للمضارع المخاطب المؤنث المفرد	تسلمي	شلون دا تدخلي تسلمي عالنفسا ... ما بصير يالله خيتو	حلبية
in childbed	نَفْساء 1	نَفْساء 1		مؤنث	مفرد	اسم		نفسا	ع/حرف جر+ال/أداة تعريف	عالنفسا	شلون دا تدخلي تسلمي عالنفسا ... ما بصير يالله خيتو	حلبية
..	..	..				علامة ترقيم		..		..	شلون دا تدخلي تسلمي عالنفسا ... ما بصير يالله خيتو	حلبية
not	ما 2	ما 2				أداة نفي		ما		ما	شلون دا تدخلي تسلمي عالنفسا ... ما بصير يالله خيتو	حلبية
become/be gin to	صارَ	صارَ	غائب	مذكر	مفرد	فعل مضارع		صير	ب/أداة مضارعة	بصير	شلون دا تدخلي تسلمي عالنفسا ... ما بصير يالله خيتو	حلبية
to/for + Allah/God +	اَللّه	الله		مذكر	مفرد	اسم		الله	ي/أداة نداء	يالله	شلون دا تدخلي تسلمي عالنفسا ... ما بصير يالله خيتو	حلبية
sisters | sister/count	أُخْت 1	أُخْت 1	-	مؤنث		اسم		خيتو		خيتو	شلون دا تدخلي تسلمي عالنفسا ... ما بصير يالله خيتو	حلبية

Figure 3: Screenshot of the *Tawseem* annotation portal, our web-based annotation tool

of the token. We follow the SAMA tagset for compatibility reasons (Maamouri et al., 2010).

Stem (الجذر) specifies the segment of the token after removing suffixes and prefixes. It helps in the morphological analysis of the tokens. We follow the (Stem/POS) tagging schema used in (Maamouri et al., 2010) where the stem and POS are specified separated by '/'.

Affixes: prefixes and suffixes. We follow the prefixes السوابق and suffixes اللواحق tagging schema used in SAMA.

⟨Prefix1/POS⟩ + ⟨Prefix2/POS⟩ . . .

⟨Suffix1/POS⟩ + ⟨Suffix2/POS⟩ . . .

The schema specifies a sequence of affix and affix POS pairs separated by '+'. Each pair is an affix and affix POS separated by '/'.

Affixes and stems are morphemes where the concept of morpheme denotes the smallest morphological unit of text. Prefixes specify morphemes that connect to the beginning of a stem or to other prefixes to form a word. Suffixes specify morphemes that connect to the end of a stem or to other morphemes to form a word. Dialect affixes and their POS tags differ from MSA affixes and augment them due to the extended morpho-syntactic and semantic roles of dialect affixes.

Note here, for example, the synergy of using the future and progressive particles ع استقبال (FUT_PART) + ب مضارعة(PROG_PART) as prefixes to indicate present continuous tense for verbs in Aleppo as in بشتغل /*bštġl* (I am working).

While most of the Syrian dialects precede present tense verbs with the IV1P POS with م مضارعة (PROG_PART), the Latakia coastal dialect applies it to almost all present tense verbs as with مأدرس /*mʔadrs* (I am studying). Latakia dialect also uses the prefix أ /*ʔa* for negation (and thus it corresponds to a NEG_PART POS tag) before present tense verbs as in أبعرف /*ʔabʕrf* (I don't know).

Person (الإسناد) specifies whether the subject of the token is a متكلم /*mtklm* (first), (مخاطب /*mḫāṭb*) (second) or غائب /*ġāʔyb* (absent) person when applicable.

Aspect (صيغة الفعل) concerns verbs and specifies whether they are in (مضارع /*mḍārʕ*) present for imperfective verbs (ماضي /*māḍy*) past for perfective verbs and (أمر /*ʔamr*) imperative tense.

Gender (الجنس) specifies whether a word is of مذكر /*mḏkr* male for masculine, مؤنث /*mʔwnṯ* female for feminine, or لا ينطبق /*lā ynṭbq* not applicable association when applicable.

Number (العدد) denotes مفرد /*mfrd* for singular, جمع /*ǧmʕ* for plural, مثنى /*mṯnā* for dual (to count two units), or لا ينطبق for uncountable words when

$rowToken$		$Token$	Dialect lemma	MSA lemma
ألت /*ʾalt*	I said	قلت /*qlt*	قال /*qāl*	قَالَ /*qaāla*
تختك /*tẖtk*	your bed	تختك /*tẖtk*	تخت /*tẖt*	سَرِيْر /*saryr*
مهندز /*mhndz*	engineer	مهندس /*mhnds*	مهندس /*mhnds*	مُهَنْدِس /*muhandis*
طريئ /*ṭryʾy*	street	طريق /*ṭryq*	طريق /*ṭryq*	طَرِيق /*ṭariyq*

Table 3: Example annotations for Nâbīa tokens

applicable.

7 Evaluation and Agreement

Before evaluating Nâbīa, we normalized the annotations to unify variant annotations that are equivalent. These variants occur due to human mistakes such as typos (ماصي /*māṣy* instead of ماضي /*māḍy*), ordering of tags in sequences of tags, and inconsistent use of separators and spacing.

Another source of variants is tokens with no feature values in the existing annotated dialects. Annotators have to come up with novel values. We detected these tag values, ranked them based on their frequencies, and clustered them based on their edit distance from each other. Then we reviewed them and unified them across Nâbīa and its features.

We developed a small suite of VBA scripts empowered with regular expressions to check for these variants and correct them automatically where possible. If automatic correction is not possible and human attention is required, then our reference annotators interfere to correct it.

7.1 Inter-annotation agreement

After the automatic corrections, six linguists visited the annotations to approve or correct them. This created a significant overlap of annotations as shown in Table 5. The overlap column shows the number of annotations per feature that had more than one annotation. Some of the second annotations were performed by the original annotator, so the reviewed column shows the number of annotations that were reviewed by two or more annotators. The unique column shows the number of unique values for the tokens with overlapping annotations.

The correction approach secured a significant overlap. We report the performance of the annotators in terms of precision, recall, and F1-score taking the correcting annotator as a reference in Table 4. A true positive (TP) for a feature value fv, denotes that the original annotation matched the reference annotation. A false positive (FP) for fv reflects an original annotator selecting fv for the token in conflict with the selection of the reference annotator. A false negative (FN) is when the original annotator fails to select fv for a token when the reference annotator selected it. Precision (P) and recall (R) are given by the ratios $TP/(TP+FP)$, and $TP/(TP+FN)$, respectively. The F1-score is given by $2PR/(P+R)$.

We also computed the Kappa-Cohen metric (McHugh, 2015) as implemented in the Scientific Kit Learn package (scikit learn, 2022). Table 4 shows the results where we compared the feature values of the reference annotators versus those of the original annotators.

The results show performance and agreement across all features. The κ scores are lower than the F-scores as the the κ metric accommodates for agreement by chance. The difference shows more with prefixes and suffixes as a significant part of the tokens had empty prefix and suffix, allowing more agreement by chance.

7.2 Qualitative Evaluation

To conduct a qualitative evaluation, we randomly selected about $7K$ annotations and reviewed them manually. We found a high agreement between the annotators who followed the specific guidelines and used our annotation tool. In what follows, we discuss some of the common mistakes:

(i) In rare cases, tokens specific to small local communities were hard to understand, Such as the token زنطر /*znṭr* (become cold) in the Latakia dialect. Although the annotators did their best to search external resources to understand such words, some mistakes still existed.

(ii) Tokens with no clear MSA equivalent led to difficulty in selecting MSA lemmas; thus, different annotators might not agree on selecting the same lemma. For example, the token عَمنوَّل /*ʿamnwal* may have several MSA lemmas, such as عام /*ʿām* (year), or ماضي /*māḍy* (past).

(iii) Semantic ambiguities in contexts led to disagreements on selecting lemmas. For instance, the token بقى /*bqā* has three possible meanings (was), (therefore) and (also). And sometimes all three fit the context.

8 Conclusion

This paper presents Nâbīa, a morphologically annotated corpora of Syrian Arabic dialects. The corpora contain about $60K$ tokens from 10 Syrian dialects, collected from social media platforms, movies and series, common proverbs, and song lyrics and poetry. To be compatible with SAMA

Feature	TP	FP	FN	P	R	F	κ
Stem	21,506	4,933	5,461	0.813	0.797	0.805	0.796
POS	20,727	2,979	3,316	0.874	0.862	0.868	0.843
Prefix	22,886	448	496	0.981	0.979	0.980	0.939
Suffix	22,096	1,247	1,380	0.947	0.941	0.944	0.837
DA Lemma	18,600	5,765	6,451	0.763	0.742	0.753	0.739
MSA Lemma	19,300	5,161	5,749	0.789	0.770	0.780	0.767

Table 4: Precision and recall results due to annotation correction with F and κ scores

Feature	Overlap	Reviewed	Unique
Stem	44,687	26,967	3,102
POS	39,007	24,043	56
Prefix	39,007	23382	163
Suffix	39,007	23,476	358
DALemma	41,579	25,052	3,586
MSALemma	41,579	25,050	3,352

Table 5: Reviewed overlap and unique feature values across Nâbr̄a

and other Arabic corpora, we chose to annotate the corpora using SAMA tagsets. To evaluate the quality of the corpora, we used the $F1$ and $kappa$ scores which show high agreement.

We plan to use Nâbr̄a to extend Wojood (Jarrar et al., 2022a; Liqreina et al., 2023) by annotating the corpora for Named Entity Recognition, similar to what we did with Curras and Baladi.

Limitations

The work in Nâbr̄a has the following limitations.

- Nâbr̄a covers 10 Syrian dialects. variants of these dialects and other smaller dialects confined in less urban localities exist. Future work should extend Nâbr̄a to better cover the Syrian dialect.
- Nâbr̄a addressed the Syrian dialects and their relation to the Arabic language and touched in prose on the relations to languages of origin such as Aramaic and Cyrillic. More data-oriented work is needed to relate Nâbr̄a to languages of origin that were spoken in Syria as well as to the geo-linguistic features of these languages.
- The annotation and evaluation process leveraged linguists who may be better at some of the dialects than others. We will make Nâbr̄a available online with correction suggestion capacities to accommodate for possible potential corrections.

Ethics Statement

The collection of texts used in Nâbr̄a respects intellectual property of the material. The annotation process employed annotators who were paid a fair rate per hour based on their living locality. Results from Nâbr̄a will be shared online for the research community to use and improve upon.

Acknowledgements

We would like to thank all the students who helped in the annotation process, especially Salam Eideh, Nagham Idrees, Shahd Khader, Shimaa Hamayel, Rwaa Assi, Hiba Zaid, Lama Saqqour, Ali Maihoob, and Eman Hariri.

References

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, Barcelona, Spain (Online). Association for Computational Linguistics.

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, Houda Bouamor, and Nizar Habash. 2021. NADI 2021: The second nuanced Arabic dialect identification shared task. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 244–259, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Moustafa Al-Hajj and Mustafa Jarrar. 2021a. Arabglossbert: 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 40–48, Online. INCOMA Ltd.

Moustafa Al-Hajj and Mustafa Jarrar. 2021b. Lu-bzu at semeval-2021 task 2: Word2vec and lemma2vec performance in arabic word-in-context disambiguation. In *Proceedings of the 15th International Workshop* *on Semantic Evaluation (SemEval-2021)*, pages 748–755, Online. Association for Computational Linguistics.

Faisal Al-Shargi, Aidan Kaplan, Ramy Eskander, Nizar Habash, and Owen Rambow. 2016. Morphologically annotated corpora and morphological analyzers for Moroccan and sanaani yemeni Arabic. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1300–1306, Portorož, Slovenia. European Language Resources Association (ELRA).

Diana Alhafi, Anton Deik, and Mustafa Jarrar. 2019. Usability evaluation of lexicographic e-services. In *The 2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA)*, pages 1–7. IEE.

Faisal Alshargi, Shahd Dibas, Sakhar Alkhereyf, Reem Faraj, Basmah Abdulkareem, Sane Yagi, Ouafaa Kacha, Nizar Habash, and Owen Rambow. 2019. Morphologically annotated corpora for seven Arabic dialects: Taizi, sanaani, najdi, jordanian, syrian, iraqi and Moroccan. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 137–147, Florence, Italy. 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 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).

Françoise Briquel Chatonnet. 2005. De l'intérêt de l'étude du garshouni et des manuscrits écrits selon ce système. In *L'Orient Chrétien dans l'Empire musulman, en hommage au Professeur Gérard Troupeau*, Studia Arabica III, pages 463–475. Editions de Paris.

Tim Buckwalter. 2004. Buckwalter arabic morphological analyzer version 2.0. *LDC2004L02*.

Alexandra Canavan, George Zipperlen, and David Graff. 1997. Callhome egyptian arabic speech. *LDC97S45*.

Kareem Darwish, Nizar Habash, Mourad Abbas, Hend Al-Khalifa, Huseein T. Al-Natsheh, Houda Bouamor, Karim Bouzoubaa, Violetta Cavalli-Sforza, Samhaa R. El-Beltagy, Wassim El-Hajj, Mustafa Jarrar, and Hamdy Mubarak. 2021. A panoramic survey of natural language processing in the arab worlds. *Commun. ACM*, 64(4):72–81.

Mona Diab, Nizar Habash, Owen Rambow, Mohamed Altantawy, and Yassine Benajiba. 2010. Colaba: Arabic dialect annotation and processing. *LREC Workshop on Semitic Language Processing*, pages 66–74.

Sana Ghanem, Mustafa Jarrar, Radi Jarrar, and Ibrahim Bounhas. 2023. A benchmark and scoring algorithm for enriching arabic synonyms. In *Proceedings of the 12th International Global Wordnet Conference (GWC2023)*, pages 215–222. Global Wordnet Association.

Elisa Gugliotta and Marco Dinarelli. 2022. TArC: Tunisian Arabish corpus, first complete release. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1125–1136, Marseille, France. European Language Resources Association.

Karim El Haff, Mustafa Jarrar, Tymaa Hammouda, and Fadi Zaraket. 2022. Curras + baladi: Towards a levantine corpus. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2022)*, Marseille, France.

Jan Hajič, Otakar Smrž, Zemanek Petr, Jaň Snaidauf, and Emanuel Beška. 2004. Prague arabic dependency treebank: development in data and tools. *Proc. of the NEMLAR Intern. Conf. on Arabic Language Resources and Tools.*

Mustafa Jarrar. 2006. Towards the notion of gloss, and the adoption of linguistic resources in formal ontology engineering. In *Proceedings of the 15th international conference on World Wide Web (WWW2006)*, pages 497–503. ACM Press, New York, NY.

Mustafa Jarrar. 2011. Building a formal arabic ontology (invited paper). In *Proceedings of the Experts Meeting on Arabic Ontologies and Semantic Networks*. ALECSO, Arab League.

Mustafa Jarrar. 2021. The arabic ontology - an arabic wordnet with ontologically clean content. *Applied Ontology Journal*, 16(1):1–26.

Mustafa Jarrar and Hamzeh Amayreh. 2019. An arabic-multilingual database with a lexicographic search engine. In *The 24th International Conference on Applications of Natural Language to Information Systems (NLDB 2019)*, volume 11608 of *LNCS*, pages 234–246. Springer.

Mustafa Jarrar, Hamzeh Amayreh, and John P. McCrae. 2019. Representing arabic lexicons in lemon - a preliminary study. In *The 2nd Conference on Language, Data and Knowledge (LDK 2019)*, volume 2402, pages 29–33. CEUR Workshop Proceedings.

Mustafa Jarrar, Nizar Habash, Diyam Akra, and Nasser Zalmout. 2014. Building a corpus for palestinian arabic: a preliminary study. In *Proceedings of the EMNLP 2014, Workshop on Arabic Natural Language*, pages 18–27. Association For Computational Linguistics.

Mustafa Jarrar, Nizar Habash, Faeq Alrimawi, Diyam Akra, and Nasser Zalmout. 2017. Curras: An annotated corpus for the palestinian arabic dialect. *Journal Language Resources and Evaluation*, 51(3):745–775.

Mustafa Jarrar, Eman Karajah, Muhammad Khalifa, and Khaled Shaalan. 2021. Extracting synonyms from bilingual dictionaries. In *Proceedings of the 11th International Global Wordnet Conference (GWC2021)*, pages 215–222. Global Wordnet Association.

Mustafa Jarrar, Mohammed Khalilia, and Sana Ghanem. 2022a. Wojood: Nested arabic named entity corpus and recognition using bert. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2022)*, Marseille, France.

Mustafa Jarrar, Sanad Malaysha, Tymaa Hammouda, and Mohammad Khalilia. 2023a. Salma: Arabic sense-annotated corpus and wsd benchmarks. In *Proceedings of the 1st Arabic Natural Language Processing Conference (ArabicNLP), Part of the EMNLP 2023*. ACL.

Mustafa Jarrar, Fadi Zaraket, Rami Asia, and Hamzeh Amayreh. 2018. Diacritic-based matching of arabic words. *ACM Asian and Low-Resource Language Information Processing*, 18(2):10:1–10:21.

Mustafa Jarrar, Fadi Zaraket, Tymaa Hammouda, Daanish Masood Alavi, and Martin Waahlisch. 2023b. Lisan: Yemeni, irqi, libyan, and sudanese arabic dialect copora with morphological annotations. In *The 20th IEEE/ACS International Conference on Computer Systems and Applications (AICCSA)*. IEEE.

Mustafa Jarrar, Fadi A Zaraket, Tymaa Hammouda, Daanish Masood Alavi, and Martin Waahlisch. 2022b. Lisan: Yemeni, iraqi, libyan, and sudanese arabic dialect copora with morphological annotations.

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, 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.

Nouran Khallaf, Elin Arfon, Mo El-Haj, Jon Morris, Dawn Knight, Paul Rayson, and Tymaa Hammoudaand Mustafa Jarrar. 2023. Open-source thesaurus development for under-resourced languages: a welsh case study.

Hanaa Kilany, H Gadalla, Howaida Arram, A Yacoub, Alaa El-Habashi, and C McLemore. 2002. Egyptian colloquial arabic lexicon. *LDC99L22*.

Haneen Liqreina, Mustafa Jarrar, Mohammed Khalilia, Ahmed Oumar El-Shangiti, and Muhammad Abdul-Mageed. 2023. Arabic fine-grained entity recognition. In *Proceedings of the 1st Arabic Natural Language Processing Conference (ArabicNLP), Part of the EMNLP 2023*. ACL.

Mohamed Maamouri, Ann Bies, Tim Buckwalter, Mona Diab, Nizar Habash, Owen Rambow, and Dalila Tabessi. 2006. Developing and using a pilot dialectal Arabic treebank. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

Mohamed Maamouri, Ann Bies, Tim Buckwalter, Hubert Jin, and Wigdan Mekki. 2005. Arabic treebank: Part 3 (full corpus) v 2.0. *LDC2005T20*.

Mohamed Maamouri, Ann Bies, Seth Kulick, Michael Ciul, Nizar Habash, and Ramy Eskander. 2014. Developing an Egyptian Arabic treebank: Impact of dialectal morphology on annotation and tool development. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2348–2354, Reykjavik, Iceland. European Language Resources Association (ELRA).

Mohamed Maamouri, David Graff, Basma Bouziri, Sondos Krouna, Ann Bies, and Seth Kulick. 2010. Ldc standard arabic morphological analyzer (sama) version 3.1. *LDC2010L01*.

Mary L. McHugh. 2015. Interrater reliability: the kappa statistic. *Biochemia medica*, 22.

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.

Arfath Pasha, Mohamed Al-Badrashiny, Mona Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan Roth. 2014. MADAMIRA: A fast, comprehensive tool for morphological analysis and disambiguation of Arabic. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1094–1101, Reykjavik, Iceland. European Language Resources Association (ELRA).

scikit learn. 2022. sklearn.metrics.cohen_kappa_score.

Roula Skaf. 2015. *Le morphème d= en araméen-syriaque : étude d'une polyfonctionalité à plusieurs échelles syntaxiques*. Theses, Université Sorbonne Paris Cité ; Università degli studi (Torino, Italia).

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Mohammed El-Bachouti, Robert Belvin, and Ann Houston. 2013. Ontonotes release 5.0 ldc2013t19. *Philadelphia: Linguistic Data Consortium*.

Inès Zribi, Mariem Ellouze, Lamia Hadrich Belguith, and Philippe Blache. 2015. Spoken tunisian arabic corpus "stac": Transcription and annotation. *Res. Comput. Sci.*, 90:123–135.

Inès Zribi, Mariem Ellouze, Lamia Hadrich Belguith, and Philippe Blache. 2017. Morphological disambiguation of tunisian dialect. *Journal of King Saud University - Computer and Information Sciences*, 29(2):147–155. Arabic Natural Language Processing: Models, Systems and Applications.

A Appendix: Nâbr̄a Statistics

Table 6: Distribution of Gender feature. Arabic Words especially verbs and nouns and some of the functional words are annotated with “Male”“Female”. In some cases, the gender can be both, depending on the context, such as الجميع /*ālǧmyʿ* (everyone).

Gender	**Count**
Male	25,538
Female	11,790
Both	931

Table 7: Distribution of the Number feature. Arabic words especially verbs and nouns are annotated with “Singular”, “Dual”, “Plural”, and in some rare cases, the number can be "Any" like أبدى /*ʾabdā* (more important).

Number	**Count**
Singular	32,372
Dual	192
Plural	4,450
Any	163

Table 8: Distribution of the verbs' Person: 1st person (متكلم), 2nd person (مخاطب), 3rd person (غائب).

Person	**Count**
1st	2,767
2nd	2,794
3rd	6,769

Table 9: Distribution of the POS tags and categories.

Category	**POS**	**Count**
NOUN Total: 28,932	NOUN	21,250
	ADJ	4,742
	NOUN_PROP	1,540
	NOUN_QUANT	556
	NOUN_NUM	315
	ADJ_COMP	257
	ADJ_NUM	152
	ABBREV	31
	DIGIT *	89
VERB Total: 11,166	IV	5,926
	PV	3,846
	CV	1,080
	IV_PASS	289
	PV_PASS	25
FUNC_WORD Total: 19,923	PUNC *	5,010
	PREP	3,133
	CONJ	2,506
	NEG_PART	1,642
	ADV	1,485
	PRON	1,252
	SUB_CONJ	991
	REL_PRON	687
	DEM_PRON	645
	INTERROG_PART	489
	VOC_PART	357
	PART	342
	PROG_PART *	218
	VERB	171
	INTERROG_PRON	166
	FUT_PART	130
	RESTRIC_PART	117
	FOREIGN	115
	PSEUDO_VERB	101
	EMOJI *	95
	VERB_PART	44
	INTERJ	43
	DET	40
	INTERROG_ADV	38
	EXCLAM_PRON	35
	FOCUS_PART	33
	PREP + SUB_CONJ	27
	REL_ADV	11
	Total	**60,021**

HICMA: The Handwriting Identification for Calligraphy and Manuscripts in Arabic Dataset

Anis Ismail
KU Leuven
Leuven, Belgium
anis.ismail@student.kuleuven.be

Zina Kamel
Lebanese American University
Beirut, Lebanon
zina.kamel@lau.edu

Reem Mahmoud
intervu.ai
Richardson, TX, USA
reem.mahmoud@intervu.ai

Abstract

Arabic is one of the most globally spoken languages with more than 313 million speakers worldwide. Arabic handwriting is known for its cursive nature and the variety of writing styles used. Despite the increase in effort to digitize artistic and historical elements, no public dataset was released to deal with Arabic text recognition for realistic manuscripts and calligraphic text. We present the Handwriting Identification of Manuscripts and Calligraphy in Arabic (HICMA) dataset as the first publicly available dataset with real-world and diverse samples of Arabic handwritten text in manuscripts and calligraphy. With more than 5,000 images across five different styles, the HICMA dataset includes image-text pairs and style labels for all images. We further present a comparison of the current state-of-the-art optical character recognition models in Arabic and benchmark their performance on the HICMA dataset, which serves as a baseline for future works. Both the HICMA dataset and its benchmarking tool are made available to the public under the CC BY-NC 4.0 license in the hope that the presented work opens the door to further enhancements of complex Arabic text recognition.

1 Introduction

Handwriting is a method used by humans to convey information in a written medium. Every person possesses a unique style when drawing characters. This leads to a wide variation in the expression of written characters and texts. Arabic text is of particular interest as Arabic is one of the most globally spoken languages with more than 313 million speakers worldwide. In the Arabic language, the complexity of written text increases since each character inherently has different forms depending on its position in the word, that is, whether it is in the beginning, middle, or end of the word.

Historical Arabic text is abundant with more than ten centuries of rich Arabic history and is often in need of being digitized. Arabic historical manuscripts typically encompass handwritten texts, often of a significant age, characterized by cursive script, varying styles, and various artistic intricacies surrounding the written text. Arabic calligraphy is a special form of Arabic handwriting often used in manuscripts and as a prominent tool for ornating architecture. The Arabic language relies on a variety of styles in manuscripts and calligraphy, each providing a different level of aesthetic artistic views and possessing its own rules. The most popular styles of handwriting in Arabic manuscripts and calligraphy are Diwani, Thuluth, Kufic, Farsi, Naskh, and Ruqaa. Arabic calligraphy is usually hand-drawn by experienced artists with complex drawing techniques that include heavy use of diacritics and decorative symbols. Consequently, non-expert readers struggle to understand the calligraphic text.

Handwriting recognition is the task involved in converting handwritten text, which is typically captured as images, into machine-readable text. The complexity of this task is in accurately recognizing variations in the different styles of writing. Moreover, the complexity becomes more apparent in historical Arabic handwritten text due to its nature. To address the challenges in handwritten Arabic and enhance the accessibility of Arabic calligraphic content, the development of models capable of accurately recognizing this intricate handwritten text becomes essential. This, in turn, necessitates the availability of large datasets for the training and validation of such models. Many works focused on creating datasets for the task of style classification of Arabic calligraphy, such as the work of Kaoudja et al.'s (2019), while others focused on creating datasets for single character recognition (Altwaijry and Al-Turaiki, 2021), Alrehali et al.'s (2020)[1] or single-digit recognition (Abdelazeem

[1] The dataset is a combination of 3 subsets containing each 2,240, 1,000 and 2,000 characters

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 24–32
December 7, 2023 ©2023 Association for Computational Linguistics

and El-Sherif, 2017). The Calliar dataset (Alyafeai et al., 2021) is the only existing dataset today that is tailored for Arabic calligraphy recognition, on the character, word, sentence, and stroke levels. This dataset, however, contains calligraphic text drawn using digital pens on a plain white background, eliminating the realistic calligraphy style found in real-world Arabic scripts.

Despite the plethora of datasets available in the Arabic handwriting recognition space, very few represent a realistic and rich variety of styles for both historic manuscripts and calligraphy, target full-sentence handwriting recognition from unprocessed images, and are publicly accessible. We present the first publicly available dataset for Arabic handwritten text in both manuscripts and calligraphy forms called the Handwriting Identification for Calligraphy and Manuscripts in Arabic (HICMA) Dataset. With more than 5,000 images across five different Arabic writing styles, the HICMA dataset includes image-text pairs and style labels for all images. In this manuscript, we describe the collection, labeling, and processing steps of the novel HICMA dataset and present a benchmark evaluation of the latest Optical Character Recognition (OCR) models for the Arabic language on HICMA. The contributions of our work are three-fold:

1. We present the first publicly available Arabic handwriting recognition dataset targeting full sentence recognition from unprocessed images.

2. We introduce an Arabic handwriting recognition dataset that is among the most diverse collections of Arabic historic manuscripts and calligraphy with more than 5,000 images across five different writing styles.

3. We preserve the contextual details and artistic styles of the Arabic manuscripts and calligraphic text in our dataset to closely represent the occurrence of such text in real-world materials.

We make the HICMA dataset[2] and the benchmarking tool[3] presented in this manuscript publicly accessible to the research community.

2 Related Work

Several studies have dealt with collecting various types of datasets for different formats of Arabic handwriting. For regular Arabic handwriting, there are many datasets present in literature such as KHATT (Mahmoud et al., 2018), consisting of 1,000 handwritten forms collected across 1,000 different writers from different countries. It was then extended to the Online-KHATT (Mahmoud et al.) dataset consisting of 10,040 lines of handwritten text by 623 different writers. ADAB (Märgner and El Abed, 2009) is another dataset that consists of 32,492 Arabic words handwritten by more than 1,000 writers. There are also multilingual datasets that combine Arabic and English like MAYASTROUN (Njah et al., 2012), which consists of 67,825 samples written by 355 writers. The MAYASTROUN dataset consists of varying script types including words, characters, digits, mathematical expressions, and signatures.

In contrast to regular Arabic handwriting datasets, few studies in the literature have dealt with Arabic manuscript and calligraphy text. One important dataset for Arabic calligraphy is the Calliar dataset (Alyafeai et al., 2021) which records digitized versions of images as strokes and drawings using digital pens. Calliar is annotated for stroke, character, word, and sentence-level prediction. It also consists of 45,572 strokes, 7,556 words, and 2,500 sentences. However, the resulting dataset overlooks the contextual details present in real-world calligraphy such as the texture of the paper, surrounding artistic styles, noise, and interactions with other elements in the artwork. This as a result impacts an Optical Character Recognition (OCR) model's ability to recognize calligraphy in diverse and authentic settings.

Other datasets in literature targeted calligraphy style classification by focusing on the style classification alone such as the dataset by Kaoudja et al.'s (2019). Kaoudja et al. (2019) collected 1,685 images and classified them into 9 different calligraphic styles including Thuluth, Naskh, and Diwani. Each calligraphy style consists of around 180 to 195 images. Moreover, Allaf and Al-Hmouz (2016) developed a dataset and designed a system for classifying calligraphy images with artistic Arabic calligraphy types, mainly Thuluth, Reqaa, and

[2] https://hicma.net/

[3] https://github.com/anisdismail/HICMA-benchmark

Dataset	Size	Data Type	Number of Styles	Data Public
Alrehali et al.'s (2020)	5,240	characters	1 (Naskh)	×
MADbase (Abdelazeem and El-Sherif, 2017)	70,000	digits	unspecified	✓
KHATT (Mahmoud et al., 2018)	4,000	paragraphs	unspecified	✓
Calliar (Alyafeai et al., 2021)	2,500/40,000	sentences /strokes	4	✓
ADAB (Märgner and El Abed, 2009)	32,492	words	unspecified	✓
Hijja (Altwaijry and Al-Turaiki, 2021)	47,434	characters	unspecified	✓
Kaoudja et al.'s (2019)	1,685	sentences	9	×
Allaf and Al-Hmouz's (2016)	267	sentences	3	✓
KERTAS (Adam et al., 2018)	2,000	letters	unspecified	✓
Salamah and King's (2018)	1,000	letters	10	✓
Khayyat and Elrefaei's (2020)	8,638	pages	unspecified	×
MAYASTROUN (Njah et al., 2012)	67,825	varied	unspecified	×
HICMA (Ours)	5,031	sentences/styles	5	✓

Table 1: Summary of Available Datasets in Literature

Kufi. Their dataset consists of 267 images divided evenly across the three calligraphy types. Salamah and King (2018) also approached the challenge of calligraphy style classification and collected 1,000 calligraphy images scraped from public websites in various calligraphy styles. Other sophisticated datasets, such as KERTAS (Adam et al., 2018), studied images of historical manuscripts. For producing KERTAS, 2,000 images were taken from various handwritten Arabic scripts dating back to the fourteenth century and were manually annotated and segmented to extract images of the characters in the text. Furthermore, Khayyat and Elrefaei (2020) collected 8,638 images of historical Arabic manuscripts. Their dataset is categorized into fourteen classes with six handwriting styles. Adam et al. (2017) collected 330 images of isolated Arabic letters that were extracted from ancient manuscripts. This dataset consists of Ruqaa, Diwani, Kufi, Naskh, and Farsi styles and has been used to classify Arabic script styles based on segmented letters.

The aforementioned calligraphy works can be classified into two categories, (a) datasets that simplified calligraphy for recognition tasks and (b) datasets that focused only on style classification with authentic calligraphy text. The simplified calligraphy datasets removed the contextual details commonly seen in real-world calligraphy. The remaining datasets that preserved the calligraphy in its true form were focused only on style classification, making them not directly useful for handwriting recognition. To the best of our knowledge, there is no dataset in the literature that deals with Arabic handwriting recognition in both manuscript and calligraphy images. Furthermore, many of the aforementioned datasets were either not publicly available or did not allow tampering with their dataset content. This makes the majority of the datasets in the literature not readily accessible for

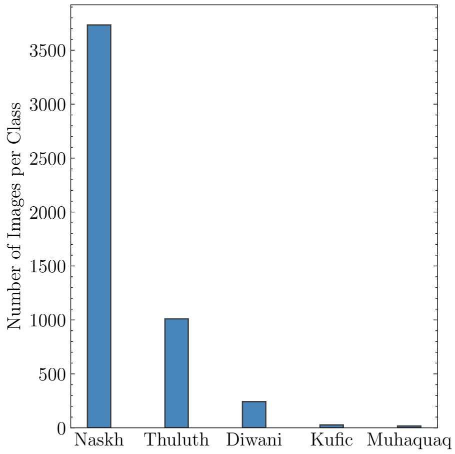


Figure 1: The style distribution of Arabic text across the HICMA dataset.

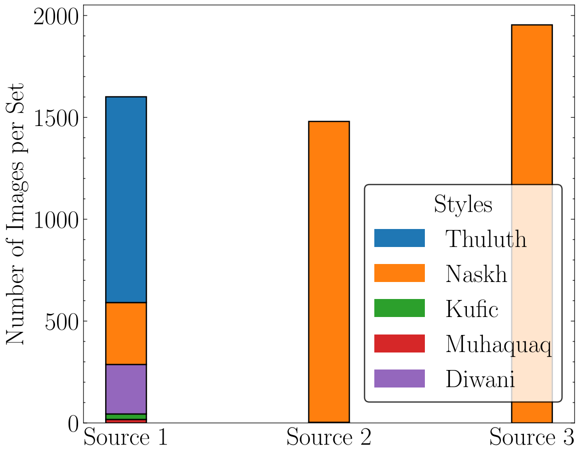


Figure 2: The style distribution of Arabic text per the 3 data sources of HICMA.

research purposes.

In Table 1, we present a comparative analysis of the existing datasets based on five criteria namely size, data type, number of styles, and whether the dataset is publicly available or not. In this paper, we introduce the HICMA dataset that targets both Arabic manuscripts and calligraphy handwriting recognition while preserving the artistic styles and contextual details of the calligraphy to closely represent real-world data.

3 HICMA Dataset

3.1 Data Collection

The first step of creating the HICMA dataset was collecting the images of the handwritten Arabic text. We collected images with various calligraphy styles including Thuluth, Diwani, Muhaquaq, Naskh, and Kufic. We relied on the following resources for building our dataset:

- **Source 1**: The Free Islamic Calligraphy website[4], which represents a Jordanian non-governmental organization (NGO) dedicated to sharing Islamic calligraphy paintings for free in a variety of styles.

- **Source 2**: The Ibn Bawab Qur'an from the Chester Beatty Library[5] located in Dublin, Ireland. This Qur'an is one of the oldest versions of the Qur'an that is written in the Naskh style by Abu'l-Hasan 'Ali ibn Hilal, who was known as Ibn al-Bawwab in the 11th century. We selected 106 pages of the Qur'an text with each page containing around 15 lines.

- **Source 3**: A private collection of manuscripts and religious writings in Naskh style dating back to the 17th century, which were made accessible by courtesy of Dr. Vahid Behmardi. We photographed and collected manuscripts of 202 available pages.

Permission was granted from all the above resources to publish all collected images in a dataset for academic research purposes.

3.2 Data Labeling

For the labeling process, 11 volunteers were recruited and trained to support in reading and recording the Arabic text in the images. The volunteers were divided into two teams who worked on labeling different images in parallel. Both teams started working on source 1, followed by source 2, and finally source 3. Every set was divided among the two teams, and once a team labeled their corresponding subset, the other team would validate the opposing team's labels. This cross-validation technique is employed to improve the quality of the produced labels and ensure accurate labels.

After the labeling process was finished, the images were processed to remove duplicate samples as well as remove diacritics and punctuation using the pyArabic[6] package. The prepared dataset was then divided into training, validation, and testing sets following an 80%-10%-10% division, respectively. To ensure that the three resulting sets have

[4] https://freeislamiccalligraphy.com

[5] https://viewer.cbl.ie/viewer/image/Is_1431/1/

[6] https://pypi.python.org/pypi/pyarabic

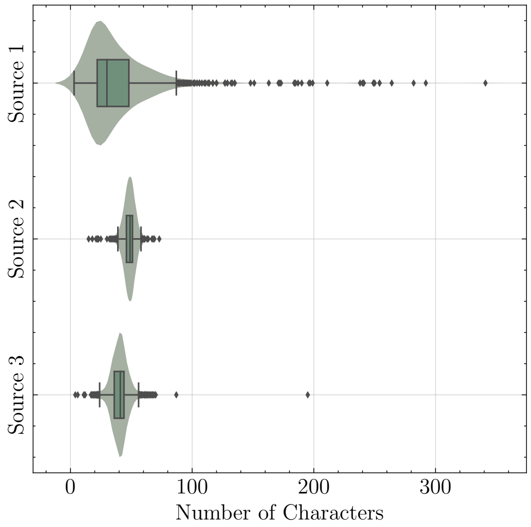


Figure 3: The distribution of label length by character count across the different dataset sources of HICMA.

the same style distributions, we relied on stratified sampling to preserve class distribution between the original set and produced subsets.

3.3 Dataset Preparation & Statistics

The data preparation process involved manually dividing the images into smaller segments. Images that originally contained multiple lines of text were further divided to create multiple images containing a single line of text. Images that only contained decorative motifs were discarded. This resulted in a total of 1,597 images from source 1, 1,480 images from source 2, and 1,954 images from source 3.

The combined HICMA dataset is thus made of exactly 5,031 images and is distributed across five styles: Kufic, Thuluth, Naskh, Diwani and Muhaquaq, with the Naskh style being the most prevalent followed by Thuluth as depicted in Figure 1. Figure 2 highlights that the most diverse set of calligraphy styles is present in source 1, whereas sources 2 and 3 predominantly consist of Naskh scriptures. This discrepancy in style diversity likely stems from the datasets' origins.

Source 1 encompasses a diverse collection of artistic Arabic calligraphy images, contributing to the wider variety of styles observed. In contrast, sources 2 and 3 comprise manuscripts only, where the Naskh style is mostly used for writing such scripts. The variation in style diversity is also evident in the sentence lengths within each set, as depicted in the violin plot in Figure 3. Although all three sets exhibit similar distributions of sentences with lengths under 100 characters and averaging around 50 characters, source 1 stands out due to the presence of numerous outliers with sentence lengths surpassing 300 characters.

The disparity in sentence lengths within source 1 can be explained by the nature of the images in this source. Calligraphy images allow for more text to be densely packed into a limited space compared to manuscript images. This aspect, combined with the challenge of segmenting intricate calligraphy words, contributes to difficulties in processing such images into smaller segments. For a visual representation refer to Table 2, which provides examples of images from all three dataset sources. The HICMA dataset is publicly available[7] for research purposes under the Attribution-NonCommercial 4.0 International (CC BY-NC 4.0) license.

4 Benchmark Experiments

4.1 Models

We investigated three state-of-the-art OCR tools that supported Arabic text recognition, namely TesseractOCR[8], Kraken (Kiessling, 2022), and EasyOCR[9], and describe them below. We ran the tools on the validation subset of the HICMA dataset (10%) for the presented benchmark evaluation.

1. **TesseractOCR**[8]: A widely-used open-source OCR engine developed by Hewelet-Packard and then by Google. It is a reliable and robust option for general text recognition tasks. The TesseractOCR engine is pre-trained for segmenting and recognizing text in images. Throughout our research, we assessed two pre-trained models for Arabic OCR from TesseractOCR[10] and ClearCypher[11].

2. **Kraken (Kiessling, 2022)**: An open-source tool specialized in recognizing historical and non-latin scripts, making it particularly suitable for the HICMA dataset. Kraken is trained on specialized datasets focusing on unique writing styles and scripts, allowing it to excel in scenarios where standard OCR engines might struggle. We evaluated the performance of three Kraken models pre-trained on Arabic manuscripts and publicly available online.

[7] https://hicma.net

[8] https://tesseract-ocr.github.io/

[9] https://www.jaided.ai/easyocr/documentation/

[10] https://github.com/tesseract-ocr/tessdata_best/blob/main/ara.traineddata

[11] https://github.com/ClearCypher/enhancing-tesseract-arabic-text-recognition

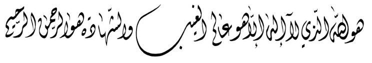
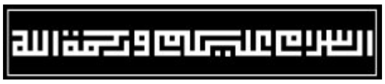
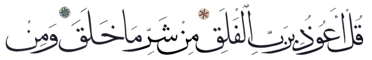
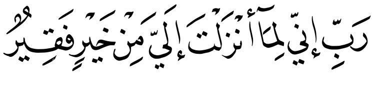
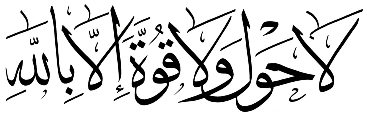
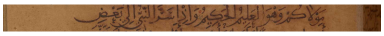
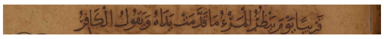
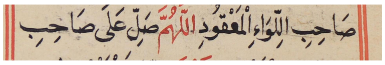
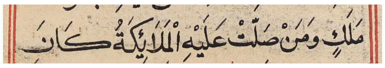
Image	Text label	Style	Source
	هو الله الذي لا اله الا هو عالم الغيب والشهادة هو الرحمن الرحيم	Diwani	Source 1
	السلام عليكم و رحمة الله	Kufic	Source 1
	قل أعوذ برب الفلق من شر ما خلق ومن	Muhaquaq	Source 1
	رب إني لما أنزلت إلي من خير فقير	Naskh	Source 1
	لا حول ولا قوة الا بالله	Thuluth	Source 1
	مولاكم وهو العليم الحكيم وإذ أسر النبي إلى بعض	Naskh	Source 2
	قريبا يوم ينظر المرء ما قدمت يداه ويقول الكافر	Naskh	Source 2
	صاحب اللواء المعقود اللهم صل على صاحب	Naskh	Source 3
	ملك و من صلت عليه الملائكة كان	Naskh	Source 3

Table 2: Sample images from HICMA along with associated labels, styles, and corresponding sources.

The three models will be referred to as Kraken-Arabic Best[12], Kraken-All Arabic Scripts[13], and Kraken-Arabic Generalized[14].

3. **EasyOCR**[9]: A user-friendly OCR library designed by Jaided AI that employs deep learning models to accurately segment and recognize text from images. It is designed to be easy to integrate into applications and supports multiple languages, including Arabic.

With the TesseractOCR and the Kraken models, the images were first transformed to grayscale and converted into binary format. In contrast, the images used for EasyOCR were not subjected to any pre-processing as no significant change in performance was observed. Moreover, as there were no available pre-trained Kraken segmentation models for Arabic, the images were resized to a smaller dimension of 200x1200 before being fed to the Kraken models. The image resizing helped decrease the inference time while also enhancing the accuracy of the Kraken models.

4.2 Evaluation Metrics

We utilized three evaluation metrics to assess the performance of the benchmark OCR models on the HICMA dataset.

1. **Levenshtein Ratio**: The Levenshtein Ratio (Sarkar et al., 2016) measures the similarity between two strings, that is, the ground

[12] https://zenodo.org/record/7050270/files/all_arabic_scripts.mlmodel

[13] https://zenodo.org/record/7050296/files/arabic_best.mlmodel

[14] https://github.com/OpenITI/OCR_GS_Data/blob/master/ara/abhath/arabic_generalized.mlmodel

	WER	CER	Levenshtein ratio
EasyOCR	**94.51%**	**58.47%**	**53.86%**
Kraken-Arabic Best	95.96%	65.84%	43.36%
Kraken-All Arabic Scripts	97.01%	67.14%	42.23%
Kraken-Arabic Generalized	100.55%	75.09%	34.82%
TesseractOCR-ClearCypher	98.99%	75.44%	31.94%
TesseractOCR	99.44%	81.96%	26.79%

Table 3: Summary of HICMA evaluation results across the three benchmark OCR models.

truth and OCR-generated text. It is derived from Levenshtein distance (Levenshtein, 1966), which calculates the minimum number of single-character edits required to convert one string into another and then computes the ratio of correct characters to the total number of characters in the ground truth text. A higher Levenshtein ratio reflects a more accurate OCR model.

2. **Character Error Rate (CER)** (Morris et al., 2004): The CER relies on the Levenshtein distance (Levenshtein, 1966) to calculate the ratio of incorrect characters recognized as compared to the ground truth text. It quantifies the accuracy of OCR models at the individual character level. The CER is associated with the portion of characters being incorrectly predicted. A lower CER reflects a more accurate OCR model with 0 being a perfect score. The CER score may exceed 1 if the value of insertions is high.

3. **Word Error Rate (WER)** (Morris et al., 2004): The WER calculates the ratio of incorrectly recognized words to the total ground truth words. Similarly to the CER, lower values of WER indicate better performance with 0 meaning the handwritten text was perfectly recognized. The WER may also exceed the value of 1.

All three metrics were developed using the python-levenshtein[15] package and are included in the benchmarking tool available on Github[16].

4.3 Model Results

Table 3 provides an overview of the models' performance on the HICMA validation set, measured using the three evaluation metrics: WER, CER, and Levenshtein ratio. Evidently, among the pretrained models, the EasyOCR pre-trained model for Arabic text stands out in terms of performance. However, even the best-performing model falls short of meeting the requirements for a practical OCR system for handwritten text, as the standard acceptable character error rate is around 20%(Tomoiaga et al., 2019), a benchmark that these models are quite far from achieving.

A deeper examination of the EasyOCR model's performance, shown in Figure 4, reveals that it excels particularly in recognizing text written in the Naskh style. This style exhibits a CER that is 53% lower than Diwani, the next style in terms of performance. Furthermore, the Naskh WER is 7% lower while the Levenshtein ratio is 2 times higher than Diwani. The gradual decline in performance as we transition from Naskh to Diwani, Thuluth, Muhaqaq, and finally Kufic can be attributed to their frequency of usage as calligraphy fonts as present in our dataset as well as the characteristics of each style, making some more difficult to recognize than others.

Given that Naskh is one of the most commonly used styles for Arabic manuscripts and everyday writing, the success of the EasyOCR model in this style is expected due to its primary training on Arabic computer-generated text, using the Amiri and Noto Sans Arabic fonts[17]. These fonts are very similar to manuscript handwriting styles like Naskh. On the other hand, the remaining styles like Diwani, Thuluth, Muhaqaq, and Kufic are more ornamental and artistic in nature. Therefore, the model's accuracy diminishes in recognizing these artistic styles.

This variation in performance across different calligraphic styles highlights the significance of

[15] https://github.com/maxbachmann/Levenshtein

[16] https://github.com/anisdismail/HICMA-benchmark

[17] https://github.com/Belval/TextRecognitionDataGenerator

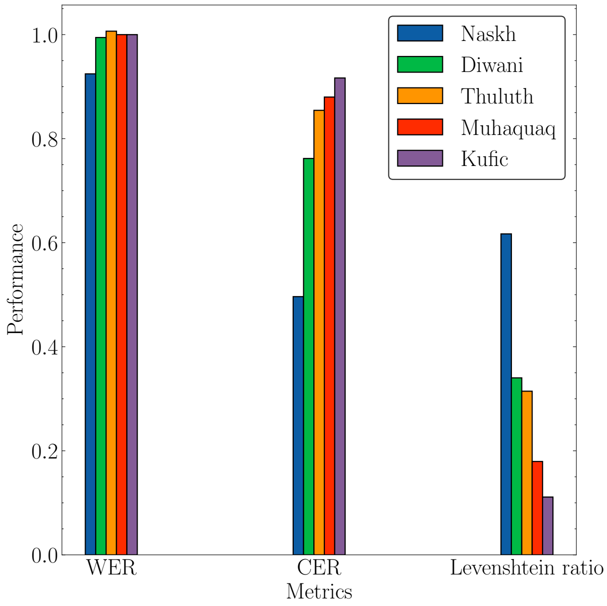


Figure 4: Performance metrics of the EasyOCR model across the different styles in HICMA.

having a diverse dataset that encompasses various styles. It also emphasizes the need to enhance OCR models' adaptability to challenging stylistic patterns within Arabic calligraphy. This endeavor would contribute to the development of more robust OCR systems capable of accurately recognizing text in images containing intricate calligraphy.

5 Limitations

As we present the HICMA Arabic dataset and the methodologies employed in this research, it is essential to acknowledge a few limitations that remain open for enhancement in future work.

- **Dataset Size and Style Diversity**: Despite HICMA being the most diverse public Arabic manuscript and calligraphy recognition dataset to date, there remains a need for further style diversification and an increase in sample count per text style. HICMA is currently composed from three sources, which do not represent the wide range of variations in Arabic texts. More so, the dataset's size remains limited compared to the vast range of Arabic texts available and would benefit from further expansion.

- **Pre-processing Challenges**: Given the inherent complexity of Arabic scripts and the variability in textual layouts, certain images in the HICMA dataset may present challenges during pre-processing. Some documents might contain lengthy texts or intricate structures, requiring manual segmentation or cropping and making it challenging to ensure reliable pre-processing across the dataset.

- **Model Limitations**: Variability in image quality, skewed perspectives, rotated motifs, and uncommon fonts have been shown to affect the existing OCR models' accuracy. To address existing Arabic OCR performance limitations, it is crucial to investigate the development of models that are fine-tuned to be native to Arabic manuscripts and calligraphy.

By addressing these limitations, future research will lead to advancements in Arabic OCR technology.

6 Conclusion

In this work, we presented HICMA as the largest and most diverse public dataset to date for Handwriting Identification of Calligraphy and Manuscripts in Arabic. The introduced dataset includes more than 5,000 images across five diverse Arabic text styles along with image-text sentence pairs and style labels for all images. This dataset fills the existing literature gap for Arabic manuscript and calligraphy text recognition. In this work, we detailed the data collection, labeling, and pre-processing steps of the created HICMA dataset. We further presented statistics about the dataset styles and label size diversity. We finally conducted a benchmark evaluation of the top three current state-of-the-art OCR models for Arabic and reported their performance on the HICMA dataset, serving as a baseline for future works. Upon analysis of the benchmark results, we highlight remaining open challenges in the HICMA dataset and the existing OCR models that support Arabic as a language. The HICMA dataset and the accompanied benchmarking tool are made publicly available for the research community. We believe our work is the first among many making more inclusive Arabic handwriting recognition for manuscripts and calligraphy possible.

7 Acknowledgements

We express our heartfelt gratitude to the dedicated volunteers and professors Vahid Behmardi and Lina Karam for their invaluable contributions to the dataset project. Special thanks for Hawraa Hellani, Kevin Nammour, Lyn Ismail, Nour Ismail, Pia

Kattoura, and all other volunteers for their meticulousness and attention to detail which significantly enhanced the dataset's quality. Their collective efforts exemplify collaboration, curiosity, and innovation, and without them, this project would not have been possible.

References

S Abdelazeem and E El-Sherif. 2017. The arabic handwritten digits databases: Adbase & madbase.

Kalthoum Adam, Somaya Al-Maadeed, and Ahmed Bouridane. 2017. based classification of arabic scripts style in ancient arabic manuscripts: Preliminary results. In *2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR)*, pages 95–98. IEEE.

Kalthoum Adam, Asim Baig, Somaya Al-Maadeed, Ahmed Bouridane, and Sherine El-Menshawy. 2018. Kertas: dataset for automatic dating of ancient arabic manuscripts. *International Journal on Document Analysis and Recognition (IJDAR)*, 21(4):283–290.

SR Allaf and R Al-Hmouz. 2016. Automatic recognition of artistic arabic calligraphy types. *Journal of King Abdulaziz University*, 27(1):3–17.

Bodour Alrehali, Najla Alsaedi, Hanan Alahmadi, and Nahla Abid. 2020. Historical arabic manuscripts text recognition using convolutional neural network. In *2020 6th Conference on Data Science and Machine Learning Applications (CDMA)*, pages 37–42. IEEE.

Najwa Altwaijry and Isra Al-Turaiki. 2021. Arabic handwriting recognition system using convolutional neural network. *Neural Computing and Applications*, 33(7):2249–2261.

Zaid Alyafeai, Maged S Al-shaibani, Mustafa Ghaleb, and Yousif Ahmed Al-Wajih. 2021. Calliar: An online handwritten dataset for arabic calligraphy. *arXiv preprint arXiv:2106.10745*.

Zineb Kaoudja, Mohammed Lamine Kherfi, and Belal Khaldi. 2019. An efficient multiple-classifier system for arabic calligraphy style recognition. In *2019 International Conference on Networking and Advanced Systems (ICNAS)*, pages 1–5. IEEE.

Manal M Khayyat and Lamiaa A Elrefaei. 2020. A deep learning based prediction of arabic manuscripts handwriting style. *Int. Arab J. Inf. Technol.*, 17(5):702–712.

Benjamin Kiessling. 2022. The Kraken OCR system.

V. I. Levenshtein. 1966. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707. ADS Bibcode: 1966SPhD...10..707L.

Sabri A. Mahmoud, Hamzah Luqman, Baligh M. Al-Helali, Galal BinMakhashen, and Mohammad Tanvir Parvez. Online-khatt: An open-vocabulary database for arabic online-text processing.

Sabri A Mahmoud, Hamzah Luqman, Baligh M Al-Helali, Galal BinMakhashen, and Mohammad Tanvir Parvez. 2018. Online-khatt: an open-vocabulary database for arabic online-text processing. *The Open Cybernetics & Systemics Journal*, 12(1).

Volker Märgner and Haikal El Abed. 2009. Icdar 2009 arabic handwriting recognition competition. In *2009 10th International Conference on Document Analysis and Recognition*, pages 1383–1387. IEEE.

Andrew Morris, Viktoria Maier, and Phil Green. 2004. From wer and ril to mer and wil: improved evaluation measures for connected speech recognition.

Sourour Njah, Badreddine Ben Nouma, Hala Bezine, and Adel M Alimi. 2012. Mayastroun: A multilanguage handwriting database. In *2012 International Conference on Frontiers in Handwriting Recognition*, pages 308–312. IEEE.

Seetah AL Salamah and Ross King. 2018. Towards the machine reading of arabic calligraphy: a letters dataset and corresponding corpus of text. In *2018 IEEE 2nd International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR)*, pages 19–23. IEEE.

Sandip Sarkar, Dipankar Das, Partha Pakray, and Alexander Gelbukh. 2016. JUNITMZ at SemEval-2016 Task 1: Identifying Semantic Similarity Using Levenshtein Ratio. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 702–705, San Diego, California. Association for Computational Linguistics.

Ciprian Tomoiaga, Paul Feng, Mathieu Salzmann, and Patrick Jayet. 2019. Field typing for improved recognition on heterogeneous handwritten forms. ArXiv:1909.10120 [cs].

Automated De-Identification of Arabic Medical Records

Veysel Kocaman John Snow Labs Inc.

Youssef Mellah John Snow Labs Inc.

Hasham Ul Haq John Snow Labs Inc.

David Talby John Snow Labs Inc.

Abstract

As Electronic Health Records (EHR) become ubiquitous in healthcare systems worldwide, including in Arabic-speaking countries, the dual imperative of safeguarding patient privacy and leveraging data for research and quality improvement grows. This paper presents a first-of-its-kind automated de-identification pipeline for medical text specifically tailored for the Arabic language. This includes accurate medical Named Entity Recognition (NER) for identifying personal information; data obfuscation models to replace sensitive entities with fake entities; and an implementation that natively scales to large datasets on commodity clusters.

This research makes two contributions. First, we adapt two existing NER architectures—BERT For Token Classification (BFTC) and BiLSTM-CNN-Char – to accommodate the unique syntactic and morphological characteristics of the Arabic language. Comparative analysis suggests that BFTC models outperform BiLSTM models, achieving higher F1 scores for both identifying and redacting personally identifiable information (PII) from Arabic medical texts. Second, we augment the deep learning models with a contextual parser engine to handle commonly missed entities. Experiments show that the combined pipeline demonstrates superior performance with micro F1 scores ranging from 0.94 to 0.98 on the test dataset, which is a translated version of the i2b2 2014 de-identification challenge, across 17 sensitive entities. This level of accuracy is in line with that achieved with manual de-identification by domain experts, suggesting that a fully automated and scalable process is now viable.

1 Introduction

Arabic is one major language that covers a large geographic and demographic portion of the world population with a high EHR adoption rate (Abdullah Alharbi, 2023). This means there is a high volume of both structured and unstructured digital data available that can be leveraged for different use cases. However, the data needs to be de-identified before being used for any research or development purpose.

De-identification of unstructured documents poses challenges due to various types of noise. Furthermore, every language has its own lexical rules, which makes it challenging to have a single model that can perform well across multiple languages. Therefore, there is a need to have models trained for different languages to get the best results. Usually, Named Entity Recognition (NER) models are used to extract sensitive information from the text which can then be de-identified (Uzuner et al., 2007). However, training NER models require labeled datasets, which are scarce and laborious to produce. In particular, the Arabic language has an extremely limited number of public datasets that can be leveraged.

The principal aim of this study is fourfold: Firstly, we introduce the first-of-its-kind medical Named Entity Recognition (NER) and De-identification models tailored specifically for the Arabic language, addressing a critical gap in the field. Secondly, we adapt existing NER architectures—BiLSTM-CNN-Char and BERT For Token Classification (BFTC)—to meet the unique syntactic and morphological requirements of the Arabic language. Thirdly, we implement a novel approach to overcome dataset limitations by translating a standard English dataset used in the 2014 i2b2 De-Identification challenge to Arabic using an entity-preservation technique. Fourthly, we employ a contextual parser engine to supplement weak entity extractions, thereby increasing the robustness of our models.

To train, evaluate, and compare these NER models, we use the Spark NLP for Healthcare library (Kocaman and Talby, 2021b), which offers both comprehensive NER support (Kocaman and Talby, 2022) and token embedding models for the Arabic

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 33–40
December 7, 2023 ©2023 Association for Computational Linguistics

language. Importantly, this is not purely academic research; it's an applied study that has been engineered to be fully compatible and scalable with Apache Spark, making it immediately deployable in large-scale healthcare systems.

2 Related Work

The concept of automatic de-identification was first introduced into the Informatics for Integrating Biology and the Bedside (i2b2) project as explained by (Uzuner et al., 2007) and then expanded by (Stubbs et al., 2015), as an academic NLP challenge on automatically detecting PHI identifiers from medical records. These challenges have boosted research and development of Machine & Deep Learning algorithms for robust PHI identification.

Since then, there have been numerous studies to expand automatic de-identification to multiple languages. (Marimon et al., 2019) generated a dataset, and trained NER models for medical texts in Spanish language. (Catelli et al., 2020) applied similar techniques to Italian COVID-19 documents for de-identification.

Over the years, researchers have proposed multiple architectures aiming to achieve better performance. Initial approaches relied on hand-crafted features and lexical rules to extract required concepts from data. However, as token embedding models (Mikolov et al., 2013) advanced, other architectures started leveraging these embedding models. Among these, Bi-LSTM and Conditional Random Fields (CRF) based models (Huang et al., 2015) became notable for NER. More recently, attention-based models have been showing significantly better performance for sequence labeling tasks (Vaswani et al., 2023).

Regarding other efforts towards extracting medical terms from Arabic medical texts, several noteworthy studies have been conducted. (Nayel et al., 2023) explored deep learning techniques, including LSTM-CRF and BiLSTM-CRF models, for disease entity recognition in Arabic medical texts, achieving impressive precision, recall, and F1-scores. (Alanazi, 2017) introduced Bayesian Belief Networks (BBN) as an innovative approach to extracting various medical entities, demonstrating promising precision and recall for diseases and treatment methods.

In addition, (Abdelhay et al., 2023) tackled the challenges of implementing medical bots in Arabic with the introduction of the MAQA dataset, highlighting the effectiveness of Transformer models. (Hammoud et al., 2020) fine-tuned neural networks for medical entity recognition in Arabic medical texts, while (Hammoud et al., 2021) presented a novel dataset for disease classification, emphasizing the potential of pre-trained models. Finally, (Samy et al., 2012) compared strategies for medical term extraction, revealing the advantages of using Arabic equivalents of Latin prefixes and suffixes. These studies collectively advance the field of NER and medical term extraction in Arabic medical texts, offering a range of valuable approaches and insights.

Despite these advancements, it is crucial to note that there has been a notable absence of de-identification models or efforts explicitly targeting the Arabic language. This gap in the literature underscores the importance and timeliness of our study, which aims to address this void by introducing the first Arabic-specific medical NER and De-identification models.

3 Dataset Construction and Annotation

Training a named entity recognition model requires data to be annotated with named entities which is a laborious process. Instead of manually annotating an Arabic dataset, we took the standard 2014 i2b2 dataset (in English) (Stubbs et al., 2015) and translated it to Arabic using the Google translate API [1]. The i2b2 dataset is in CoNLL format, which means text is tokenized, and entities are identified using the IOB2 tagging scheme [2]. Since entities have fixed boundaries relative to the original text, translating the text naively would result in entity boundary mismatch.

For example, the name and age in the text *"Alan is a 30 year old male"* start at token 1 and 4, however, after translation, the name and age start at token 1 and 6 "آلان رجل يبلغ من العمر ٣٠ عامًا". This is because translation can change the entire structure of the text, consequently, making entity

[1] https://cloud.google.com/translate

[2] In Named Entity Recognition (NER), the IOB2 (Inside-Outside-Beginning) tagging scheme is a common way to annotate and identify entities in a text. In this scheme, each word in a sequence is tagged with one of the following prefixes: "B-" (Beginning): Indicates that the word is the start of a named entity. "I-" (Inside): Indicates that the word is inside a named entity, but is not the first word of the entity. "O" (Outside): Indicates that the word is not part of any named entity. These prefixes are then followed by the type of the entity, such as "PER" for person, "LOC" for location, "ORG" for organization, etc. This makes it easier to identify not just the entities in a sequence, but also their types and spans.

boundary mapping challenging. This problem is further exacerbated for entities spanning across multiple tokens as the number of tokens could also vary.

To solve this problem, we replace entities in the original (English) text with their types. For example, *"Alan is a 30 year old male"* would be converted to *"NAME is a AGE year old male"*. This way when the text is translated, we can search for the entity types by simple string matching, and replace them with Arabic values. For instance, "NAME" is replaced with an actual Arabic name "يوسف". In addition to solving the problem of preserving entity boundaries, this technique also helps to adapt the data to the new language, as entities, such as names, cities, addresses are native Arabic values.

The original i2b2 Deid dataset provides two types of entity sets: Generic and Granular. The granular approach provides additional context that can be crucial for specific applications. For example, in a healthcare setting, knowing that a name refers to a "PATIENT" rather than just a "NAME" could be highly useful. Similarly, distinguishing between ZIP codes, cities, and countries can be very important in applications like location-based services or logistics. The generic approach is more broad and could be useful for general-purpose NER tasks where such granular distinctions are not necessary. It may also require less computational power and resources than the more detailed granular approach. Here is a sample list of mapping between generic and granular set of entities:

- NAME (PATIENT, DOCTOR, USERNAME)
- LOCATION (ROOM, DEPARTMENT, HOSPITAL, ORGANIZATION, STREET, CITY, STATE, COUNTRY, ZIP, OTHER)
- AGE
- DATE
- CONTACT (PHONE, FAX, EMAIL, URL, IPADDRESS)
- IDs (SOCIAL SECURITY NUMBER, MEDICAL RECORD NUMBER, HEALTH PLAN NUMBER, ACCOUNT NUMBER, LICENSE NUMBER, VEHICLE ID, DEVICE ID, BIOMETRIC ID, ID NUMBER)
- PROFESSION

Table 1 illustrates the difference between the generic and granular entity datasets. The details regarding the differences between entity sets, annotation schema, and annotation guidelines can be found at (Stubbs et al., 2015).

Chunk	**Generic**	**Granular**
2000 16	DATE	DATE
ليلى حسن	NAME	PATIENT
789	LOCATION	ZIP
جدة	LOCATION	CITY
54321	LOCATION	ZIP
المملكة العربية	LOCATION	CITY
السعودية	LOCATION	COUNTRY
النور	LOCATION	HOSPITAL
اميرة احمد	NAME	DOCTOR
ليلى	NAME	PATIENT
35	AGE	AGE

Table 1: Tokenized illustration of difference between generic and granular entities. In the "Generic" column, entities are tagged with broad, high-level categories. On the other hand, the "Granular" column takes entity recognition a step further by using more specific, detailed tags.

4 Architecture

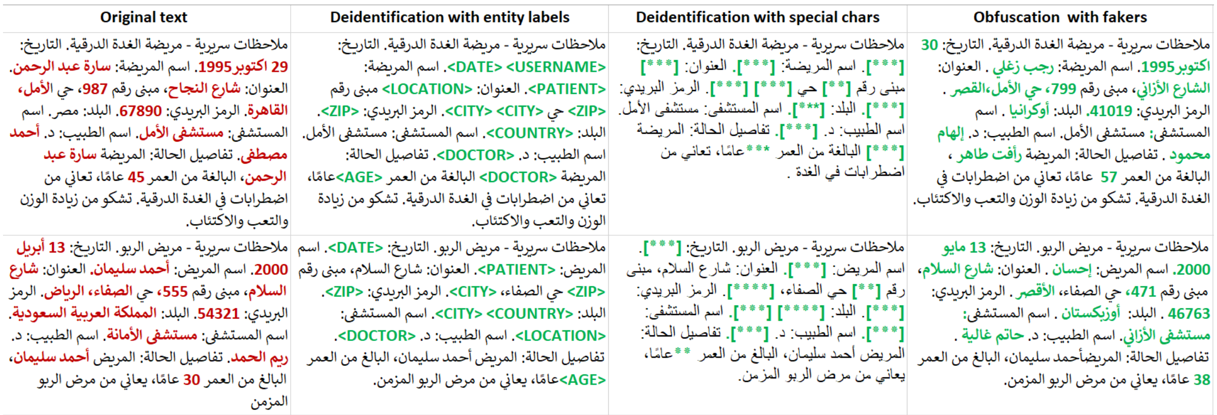
4.1 Scalable NLP Pipeline

Our system leverages the capabilities of Spark NLP (Kocaman and Talby, 2021b), a widely-used open-source NLP library that excels in scalability for both training and inference tasks on any Apache Spark setup. The architecture allows for easy deployment either on a single machine or across a Spark cluster without requiring any modification to the code base. The de-identification process for Arabic text is realized through a multi-stage NLP pipeline, consisting of text pre-processing, deep learning models, contextual guidelines, and data masking techniques. The pipeline components can be seen at Figure 2.

4.1.1 Text Pre-Processing

The pipeline's initial phase involves multiple components such as a document assembler, sentence detector, token generator, and word embedding creator. These components are designed to prepare the data for identification and subsequent anonymization of Protected Health Information (PHI) tokens in Arabic.

At the outset, a document assembler is utilized to structure raw Arabic text, generating annota-



Original text	Deidentification with entity labels	Deidentification with special chars	Obfuscation with fakers
ملاحظات سريرية - مريضة الغدة الدرقية. التاريخ: 29 اكتوبر1995. اسم المريضة: سارة عبد الرحمن. العنوان: شارع النجاح، مبنى رقم 987، حي الأمل، القاهرة. الرمز البريدي: 67890. البلد: مصر. اسم المستشفى: مستشفى الأمل. اسم الطبيب: د. أحمد مصطفى. تفاصيل الحالة: المريضة سارة عبد الرحمن، البالغة من العمر 45 عامًا، تعاني من اضطرابات في الغدة الدرقية. تشكو من زيادة الوزن والتعب والاكتئاب.	ملاحظات سريرية - مريضة الغدة الدرقية. التاريخ: <DATE> <USERNAME>. اسم المريضة: <PATIENT>. العنوان: <LOCATION> مبنى رقم <ZIP> حي <CITY> <CITY>. الرمز البريدي: <ZIP>. البلد: <COUNTRY>. اسم المستشفى: مستشفى الأمل. اسم الطبيب: د. <DOCTOR>. تفاصيل الحالة: المريضة <DOCTOR> البالغة من العمر <AGE>عامًا، تعاني من اضطرابات في الغدة الدرقية. تشكو من زيادة الوزن والتعب والاكتئاب.	ملاحظات سريرية - مريضة الغدة الدرقية. التاريخ: [***]. اسم المريضة: [***]. العنوان: [***] مبنى رقم [**] حي [***] [***]. الرمز البريدي: [***]. البلد: [***]. اسم المستشفى: مستشفى الأمل. اسم الطبيب: د. [***]. تفاصيل الحالة: المريضة [***] البالغة من العمر ***عامًا، تعاني من اضطرابات في الغدة .	ملاحظات سريرية - مريضة الغدة الدرقية. التاريخ: 30 اكتوبر1995. اسم المريضة: رجب زغلي . العنوان: الشارع الأزاني، مبنى رقم 799، حي الأمل،القصر . الرمز البريدي: 41019. البلد: أوكرانيا . اسم المستشفى: مستشفى الأمل. اسم الطبيب: د. إلهام محمود . تفاصيل الحالة: المريضة رأفت طاهر ، البالغة من العمر 57 عامًا، تعاني من اضطرابات في الغدة الدرقية. تشكو من زيادة الوزن والتعب والاكتئاب.
ملاحظات سريرية - مريض الربو. التاريخ: 13 أبريل 2000. اسم المريض: أحمد سليمان. العنوان: شارع السلام، مبنى رقم 555، حي الصفاء، الرياض. الرمز البريدي: 54321. البلد: المملكة العربية السعودية. اسم المستشفى: مستشفى الأمانة. اسم الطبيب: د. ريم الحمد. تفاصيل الحالة: المريض أحمد سليمان، البالغ من العمر 30 عامًا، يعاني من مرض الربو المزمن	ملاحظات سريرية - مريض الربو. التاريخ: <DATE>. اسم المريض: <PATIENT>. العنوان: شارع السلام، مبنى رقم <ZIP> حي الصفاء، <CITY>. الرمز البريدي: <ZIP>. البلد: <CITY> <COUNTRY>. اسم المستشفى: <LOCATION>. اسم الطبيب: د. <DOCTOR>. تفاصيل الحالة: المريض أحمد سليمان، البالغ من العمر <AGE>عامًا، يعاني من مرض الربو المزمن.	ملاحظات سريرية - مريض الربو. التاريخ: [***]. اسم المريض: [***]. العنوان: شارع السلام، مبنى رقم [**] حي الصفاء، [****]. الرمز البريدي: [***]. البلد: [****] [***]. اسم المستشفى: [***]. اسم الطبيب: د. [***]. تفاصيل الحالة: المريض أحمد سليمان، البالغ من العمر **عامًا، يعاني من مرض الربو المزمن.	ملاحظات سريرية - مريض الربو. التاريخ: 13 مايو 2000. اسم المريض: إحسان . العنوان: شارع السلام، مبنى رقم 471، حي الصفاء، الأقصر . الرمز البريدي: 46763 . البلد: أوزبكستان . اسم المستشفى: مستشفى الأزاني. اسم الطبيب: د. حاتم غالية . تفاصيل الحالة: المريضأحمد سليمان، البالغ من العمر 38 عامًا، يعاني من مرض الربو المزمن.

Figure 1: Example of de-identifying a text in Arabic using masking and obfuscation.

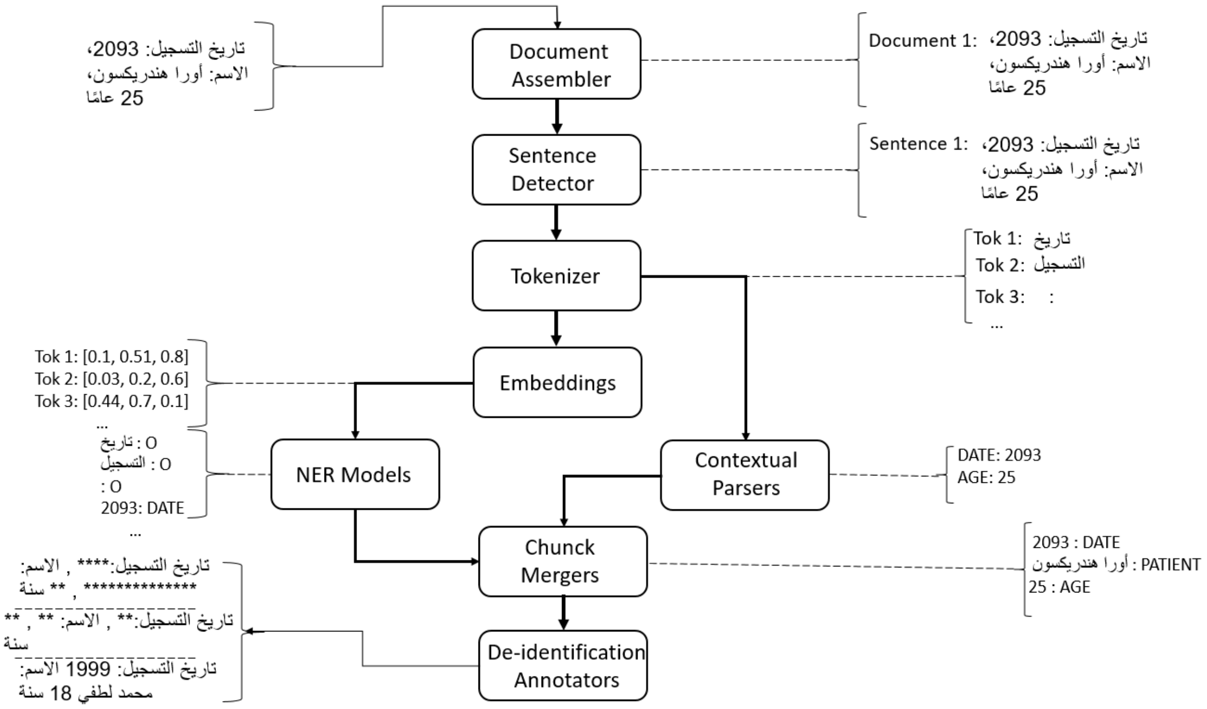


Figure 2: Full pipeline architecture

tions that can be processed further downstream. Following this, the pipeline employs a specialized deep-learning model (Schweter and Ahmed, 2019) optimized for Arabic clinical texts to perform sentence boundary detection. Rule-based techniques underperform in this context, owing to the unique grammar and punctuation in Arabic medical notes.

4.1.2 Named Entity Recognition

The core of the de-identification mechanism is the Named Entity Recognition (NER) model. It identifies PHI components like patients' names, healthcare providers, facilities, geographical locations, and specific identification numbers in Arabic text. The NER model is a crucial element as it minimizes data loss while recognizing PHI efficiently. For this, we employ a Bi-directional LSTM (BLSTM) architecture as detailed in (Kocaman and Talby, 2021a).

4.1.3 Enhancing NER with Contextual Rule Engine

While machine learning models excel in generalization, they might lack the granularity required for certain PHI identifiers. Therefore, a regular-expression-based rule engine is included in the pipeline to address this limitation. The rule engine, called Contextual Parser (CP), offers a set of adjustable parameters for prefix and suffix matching, enhancing the system's precision.

In the realm of de-identification, augmenting our NER models with CP rules offers a robust strategy for enhanced recognition and protection of Personal Health Information (PHI) elements. CP rules are linguistically tailored regulations that exploit the surrounding context of entities to optimize their detection accuracy. This is particularly useful for handling complex medical terminology, ambiguous entities, and cultural or geographical variations, especially in Arabic medical texts.

Rule Formulation: A collaborative effort between domain experts and translators allows us to design a set of CP rules that are specific to both the medical domain and the Arabic language. These rules address the unique linguistic complexities of medical texts, such as abbreviations, compound terms, and varying morphological patterns. Special attention is given to rules that target the identification of critical PHI elements like email addresses, dates, and identification numbers.

Entities Reinforced by CP Rules: The CP rules particularly bolster the NER model's ability to identify and protect a diverse array of entities. These include but are not limited to Social Security Numbers (SSN), Account Numbers (ACCOUNT), License Numbers (LICENSE), Ages (AGE), Phone Numbers (PHONE), ZIP Codes (ZIP), Medical Record Numbers (MEDICALRECORD), Emails (EMAIL), Dates (DATE), Driver's License Numbers (DLN), and Vehicle Identification Numbers (VIN).

In summary, the incorporation of CP rules into a de-identification process enhances the capabilities of our NER models, making them highly adaptable and effective in identifying a broad range of PHIs. Our model now proficiently identifies and protects the aforementioned entities, demonstrating the efficacy of our approach in safeguarding patient information in Arabic medical texts.

This multi-dimensional approach, combining data-driven deep learning with domain-specific linguistic rules, showcases the flexibility and robustness of our NER models. It not only fortifies our system against privacy intrusion but also aligns it with data protection laws.

4.1.4 Chunk Merger

Subsequent to the identification of PHI chunks by machine learning models and rule-based methods, the pipeline consolidates these identifications to optimize overall accuracy. The system assigns priorities to each type of entity, allowing for customization depending on use-cases.

4.1.5 Masking or Obfuscation

In the final stage, the system performs the actual deidentification and obfuscation. This involves masking or substituting PHI elements with dummy data while preserving the overall structure and format of the documents.

Accurate NER is the first step towards de-identifying a text - the next step is to redact the information. This can be achieved by applying either masking or obfuscation. Masking essentially replaces the identified entities with either their entity type or asterisks. These asterisks can either be of fixed character length for all the identified entities, or of the same length as the entity chunk being replaced; we found the later option to be helpful while de-identifying pdf and image documents, as it minimizes any changes to the original document layout.

Obfuscation involves replacing PHI with surrogate values that are semantically, and linguistically correct. For example, names are replaced with random names, similarly, dates are replaced with randomized dates within an offset window. Although obfuscation appears to be the better de-identification strategy as it obfuscates the entire text, making it harder to re-identify (even when an entity is missed by the NER model), there are some inherent challenges while maintaining data integrity. For example, multiple occurrences of names, addresses, and dates should be replaced with similar values throughout the document to maintain data integrity. The Spark NLP for Healthcare library already has built-in methods to track entities for consistent obfuscation.

Figure 1 illustrates text de-identified using masking and obfuscation.

5 Experimentation & Analysis

Two different NER architectures are trained and evaluated on a standard 80-20 split, and their performance is evaluated based on the model architecture and the embeddings used while training. The first model is based on a Bi-LSTM architecture as explained in (Kocaman and Talby, 2021a). This Bi-LSTM model is versatile and can be paired with virtually any token embedding model. In our experiments, we use this architecture with GLoVe (Pennington et al., 2014) and BERT (Devlin et al., 2019) embeddings. The GLoVe embeddings are trained on the Arabic common crawl dataset [3] [4]. For Arabic BERT embeddings, we utilize models pre-trained on an Arabic dataset; AraBERT (Antoun et al., 2021), and CamelBERT (Inoue et al., 2021). The second model architecture is based solely on BERT, upon which we train end-to-end BERT For Token Classification (BFTC) models.

In terms of model architecture, the BFTC models outperformed Bi-LSTM based models on both datasets as explained in Table 2 and 3. The Bi-LSTM model trained with GLoVe, AraBERT, and CamelBERT embeddings achieved macro F1 score of **0.9378**, **0.9372**, **0.9590** on the generic entity dataset, and **0.9386**, **0.9178**, **0.9369** on the granular entity dataset. In comparison, the BFTC models achieved 1-2% higher F1 scores.

In addition to the named entities in our training dataset, most documents contain certain rule-based entities like unique organizational/national identifiers. Extracting such information does not necessarily require re-training the model, as most of these identifiers have a fixed format, and can be easily extracted using regular expressions. Therefore, we include a regular expression engine in the final pipeline that is fully customizable as explained in section 4.1.3. Figure 2 illustrates a complete end-to-end pipeline with all the components.

6 Conclusion

In conclusion, this study successfully presents a groundbreaking advancement in healthcare data privacy and research for Arabic-speaking communities by introducing the first medical Named Entity Recognition (NER) and De-identification models tailored specifically for the Arabic language. Through the adaptation of existing architectures—BiLSTM-CNN-Char and BERT For Token Classification (BFTC)—we were able to accommodate the unique linguistic features of Arabic. Furthermore, our novel entity-preservation technique was pivotal in overcoming the challenges associated with limited datasets, enabling the translation of a standard English dataset into Arabic for training and evaluation.

Our comparative analyses demonstrated that BERT For Token Classification models outperformed Bi-LSTM models, achieving higher F1 scores in both the identification and redaction of personally identifiable information (PII) in Arabic medical texts. The contextual parser engine deployed in our study further enhanced the robustness of our models.

Significantly, this work is more than just an academic endeavor; it is an applied study with tools that are ready to be deployed at scale using Apache Spark. As a seminal contribution, this research not only provides essential tools for the safe and efficient handling of Arabic medical records but also lays a foundation for future studies, opening up avenues for the adaptation of NER and De-identification techniques to other underrepresented languages.

7 Limitations

Following are some of the limitations of the solution that may affect its generalizability and reliability, and need to be studied further for improvements:

7.1 Dataset quality and Diversity

The translation of English to Arabic (achieved through the Google Translate API), may not be able to completely take into account the detailed linguistic diversity and medical terminology in this domain. This could result in inaccurate data from a translated dataset that would affect the performance of NER models. Moreover, since there are differences in grammatical structures between the languages, direct substitution of masked chunks with Arabic texts may produce syntactic and contextual ambiguities. The division of entities and their classifications may be affected by these ambiguities. In translation errors, noise, and inconsistencies in the dataset could be introduced that might affect model performance.

7.2 Limited Vocabulary and Language Nuances

Arabic, which may be difficult for the NER models to read accurately, is a diverse language with different dialects and nuances. In the field of medicine, there are further difficulties to be encountered with domain-specific jargon and terminology. The model's performance may be hindered by the fact that it does not have an effective ability to deal with uncommon and distinct domain terms which could result in erroneous negative findings or misclassification.

[3] https://commoncrawl.org/

[4] https://fasttext.cc/docs/en/crawl-vectors.html

Model	AGE	CNTC	DATE	ID	LOC	NAME	PRO	GEND	Macro	Micro
Bi-LSTM (GLoVe CC)	0.9870	0.9799	0.9870	0.8358	0.9413	**0.9648**	**0.9210**	0.8863	0.9378	0.9572
Bi-LSTM (AraBERT-base)	0.9727	0.9696	0.9734	0.8450	0.8675	0.8784	0.8071	0.8869	0.9372	0.9505
Bi-LSTM (CamelBERT)	**0.9885**	0.9666	0.9757	0.8656	0.8975	0.9111	0.8675	0.9096	0.9590	0.9712
BFTC (AraBERT-base)	0.9854	**0.9852**	**0.9901**	**0.9467**	0.9225	0.9425	0.8622	0.9507	0.9600	0.9800
BFTC (CamelBERT)	0.9830	0.9828	0.9899	0.9333	**0.9494**	0.9624	0.8601	**0.9556**	**0.9700**	**0.9800**

Table 2: F1 scores on the generic entity dataset (CNTC: Contact, LOC: Location, PRO: Profession, GEND: Gender).

Entity	1	2	3	4	5
ZIP	0.9756	0.9580	0.9566	0.9483	0.9510
USER	1.0000	1.0000	1.0000	0.9557	1.0000
STR	0.9856	0.9841	0.9836	0.9186	0.9824
GEND	0.8850	0.8895	0.8918	0.9508	0.9262
PRO	0.9113	0.8284	0.8780	0.8498	0.8676
PH	0.9268	0.9135	0.8918	0.9352	0.9558
PAT	0.8711	0.7786	0.7898	0.8054	0.8134
ORG	0.8283	0.6046	0.7469	0.7376	0.8571
MR	0.9714	0.8571	0.7441	0.9230	1.0000
ID	0.9630	0.9629	0.9629	0.9718	0.9390
HOSP	0.8319	0.8081	0.8766	0.8969	0.9363
EMAIL	0.9782	0.9955	1.0000	1.0000	1.0000
DOC	0.9392	0.8951	0.9199	0.9345	0.9314
DATE	0.9876	0.9775	0.9768	0.9903	0.9922
CNTR	0.9461	0.8650	0.8750	0.9038	0.9362
CITY	0.9756	0.8788	0.8953	0.9400	0.9641
AGE	0.9799	0.9755	0.9879	0.9854	0.9830
Macro	0.9386	0.9178	0.9369	0.9400	0.9100
Micro	0.9434	0.9419	0.9547	0.9800	0.9800

Table 3: F1 scores on the granular entity dataset. Numbers in the columns refer to the following models: 1: Bi-LSTM (GLoVe CC), 2: Bi-LSTM (AraBERT-base), 3: Bi-LSTM (CamelBERT), 4: BFTC (AraBERT-base), 5: BFTC (CamelBERT) (USER: UserName, GEND: Gender, PRO: Profession, PH: Phone, PAT: PATIENT, ORG: Organization, MR: Medical Record, HOSP: Hospital, DOC: Doctor, CNTR: Country).

7.3 Privacy and Ethical Considerations

For patients' privacy and to comply with laws and regulations, de-identification of medical data is necessary. However, limitations may exist even in the case of state-of-the-art de-identification pipelines. It should be noted that the automated de-identification process does not guarantee absolute confidentiality, and manual verification by healthcare professionals may still be needed to ensure the correct erasure of sensitive information. Careful consideration has to be given to the ethical consequences of false positives and false negatives in de-identification.

7.4 Performance Evaluation Metrics

The metrics of precision, recall, and F1 score are widely applied for evaluating NER model's performance, but they may lack a full understanding of the actual world impact of false positives and false negatives in healthcare contexts. In order to provide a more comprehensive assessment of model efficiency, it would be useful to develop domain-specific evaluation metrics that account for the criticality of different types of entities in medical documents.

References

Mohammed Abdelhay, Ammar Mohammed, and Hesham A Hefny. 2023. Deep learning for arabic healthcare: Medicalbot. *Social Network Analysis and Mining*, 13(1):71.

Raed Abdullah Alharbi. 2023. Adoption of electronic health records in saudi arabia hospitals: Knowledge and usage. *Journal of King Saud University - Science*, 35(2):102470.

Saad Alanazi. 2017. *A named entity recognition system applied to Arabic text in the medical domain*. Ph.D. thesis, Staffordshire University.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2021. Arabert: Transformer-based model for arabic language understanding.

Rosario Catelli, Francesco Gargiulo, Valentina Casola, Giuseppe De Pietro, Hamido Fujita, and Massimo Esposito. 2020. Crosslingual named entity recognition for clinical de-identification applied to a covid-19 italian data set. *Applied Soft Computing*, 97:106779.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Jaafar Hammoud, Natalia Dobrenko, and Natalia Gusarova. 2020. Named entity recognition and information extraction for arabic medical text. In *Multi Conference on Computer Science and Information Systems, MCCSIS*, pages 121–127.

Jaafar Hammoud, Aleksandra Vatian, Natalia Dobrenko, Nikolai Vedernikov, Anatoly Shalyto, and Natalia Gusarova. 2021. New arabic medical dataset for diseases classification. In *Intelligent Data Engineering and Automated Learning–IDEAL 2021: 22nd International Conference, IDEAL 2021, Manchester, UK,* *November 25–27, 2021, Proceedings 22*, pages 196–203. Springer.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging.

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.

Veysel Kocaman and David Talby. 2021a. Biomedical named entity recognition at scale. In *International Conference on Pattern Recognition*, pages 635–646. Springer.

Veysel Kocaman and David Talby. 2021b. Spark nlp: natural language understanding at scale. *Software Impacts*, 8:100058.

Veysel Kocaman and David Talby. 2022. Accurate clinical and biomedical named entity recognition at scale. *Software Impacts*, 13:100373.

Montserrat Marimon, Aitor Gonzalez-Agirre, Ander Intxaurrondo, Heidy Rodriguez, Jose Lopez Martin, Marta Villegas, and Martin Krallinger. 2019. Automatic de-identification of medical texts in spanish: the meddocan track, corpus, guidelines, methods and evaluation of results. In *IberLEF@ SEPLN*, pages 618–638.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

Hamada Nayel, Nourhan Marzouk, and Ahmed Elsawy. 2023. Named entity recognition for arabic medical texts using deep learning models. In *2023 Intelligent Methods, Systems, and Applications (IMSA)*, pages 281–285. IEEE.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Doaa Samy, Antonio Moreno-Sandoval, Conchi Bueno-Díaz, Marta Garrote Salazar, and José María Guirao. 2012. Medical term extraction in an arabic medical corpus. In *LREC*, pages 640–645.

Stefan Schweter and Sajawel Ahmed. 2019. Deep-eos: General-purpose neural networks for sentence boundary detection. In *KONVENS*.

Amber Stubbs, Christopher Kotfila, and Özlem Uzuner. 2015. Automated systems for the de-identification of longitudinal clinical narratives: Overview of 2014 i2b2/uthealth shared task track 1. *Journal of biomedical informatics*, 58:S11–S19.

Özlem Uzuner, Yuan Luo, and Peter Szolovits. 2007. Evaluating the state-of-the-art in automatic de-identification. *Journal of the American Medical Informatics Association*, 14(5):550–563.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need.

ArTST: Arabic Text and Speech Transformer

Hawau Olamide Toyin,[*] Amirbek Djanibekov,[*] Ajinkya Kulkarni, Hanan Aldarmaki
Mohamed bin Zayed University of Artificial Intelligence
Abu Dhabi, UAE
{hawau.toyin;amirbek.djanibekov;ajinkya.kulkarni;hanan.aldarmaki}@mbzuai.ac.ae

Abstract

We present ArTST, a pre-trained Arabic text and speech transformer for supporting open-source speech technologies for the Arabic language. The model architecture follows the unified-modal framework, SpeechT5, that was recently released for English, and is focused on Modern Standard Arabic (MSA), with plans to extend the model for dialectal and code-switched Arabic in future editions. We pre-trained the model from scratch on MSA speech and text data, and fine-tuned it for the following tasks: Automatic Speech Recognition (ASR), Text-To-Speech synthesis (TTS), and spoken dialect identification. In our experiments comparing ArTST with SpeechT5, as well as with previously reported results in these tasks, ArTST performs on a par with or exceeding the current state-of-the-art in all three tasks. Moreover, we find that our pre-training is conducive for generalization, which is particularly evident in the low-resource TTS task. The pre-trained model as well as the fine-tuned ASR and TTS models are released for research use.

1 Introduction

Large pre-trained transformer models are currently at the forefront of speech and text technologies, with applications in various text and speech recognition and generation tasks (Devlin et al., 2019; Raffel et al., 2020; Hsu et al., 2021; Baevski et al., 2020). These models share several aspects: (1) they are based on the transformer architecture (Vaswani et al., 2017), which enables efficient training of larger models and incorporating wider contexts, (2) they are scaled in terms of model size, which has been shown to correlate with performance (Alabdulmohsin et al., 2022; Hestness et al., 2017), and (3) they generally use a self-supervised training objectives, such as next token prediction (Brown et al., 2020), masked prediction (Devlin et al., 2019; Hsu et al., 2021), and contrastive loss (Baevski et al., 2020), which enable the utilization of large unlabeled datasets for multiple potential downstream tasks. Pre-trained self-supervised models like Wav2Vec2.0 (Baevski et al., 2020), and its multilingual variant (Babu et al., 2022), have mostly replaced traditional acoustic features like MFCCs and filter banks in the speech domain. These pre-trained models implicitly learn robust and generalizable acoustic representations that consistently improve performance in various supervised downstream tasks with acoustic inputs like Automatic Speech Recognition (ASR). This is achieved by simply adding a prediction layer and fine-tuning the model using a suitable loss function, such as CTC loss (Graves, 2012).

This *pre-train-then-finetune* framework is flexible for a variety of applications, but most pre-trained models are uni-modal and therefore are limited to tasks that share the same input modality. For instance, acoustic models like Wav2Vec2.0 are not typically used in text-to-speech synthesis applications, where the input is text, and the output is typically in the form of mel spectrograms. For this reason, self-supervised pre-training has not been as widely adopted in speech synthesis research. One exception to this trend is the SpeechT5 model (Ao et al., 2022), which accepts both text and speech as input and output using modal-specific networks in addition to the core encoder-decoder network. The model is first pre-trained using self-supervised objectives in both text and speech modalities, and then fine-tuned on a variety of supervised tasks, including speech transcription, speech synthesis, and speech classification. SpeechT5 has been trained only on English using more than 900 hours of speech and 400 million sentences of text data. While the model can technically be fine-tuned for other languages, our preliminary evaluations of Arabic fine-tuning show poor performance; the pre-training seems to have biased the model severely for recognizing and generating English speech.

[*] These authors contributed equally to this work.

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 41–51
December 7, 2023 ©2023 Association for Computational Linguistics

In this paper, we introduce **Ar**abic **T**ext and **S**peech **T**ransformer, **ArTST**[1], a project aiming to push the boundaries for Arabic open-source speech technology by providing various pre-trained speech and text transformers. The Arabic language exhibits significant dialectal variation and code-switching, which introduce a layer of complexity for speech recognition and generation tasks. We believe this can be best addressed via methodical and focused development of self-supervised models that target this linguistic landscape rather than multi-lingual models that may compromise mono-lingual performance for multi-lingual coverage. The first release, as described in this paper, is a direct adaptation of the SpeechT5 model, but pre-trained from scratch using Modern Standard Arabic data and evaluated on various downstream tasks. Future versions will include dialectal Arabic, as well as code-switched speech and text, by exploring the best architectural modifications for improving coverage without sacrificing performance for individual variants.

We demonstrate the performance of ArTST in the following tasks: Automatic Speech Recognition (ASR), Text-To-Speech synthesis (TTS), and spoken Dialect Identification (DID). The fine-tuned models on each task achieved performance on a par with or exceeding previously reported results on our test sets, establishing a new state-of-the-art for open-source models. For ASR, the model additionally outperforms the large pre-trained ASR models, `Whisper` (Radford et al., 2023), and `MMS` (Pratap et al., 2023), which further demonstrates the advantage of focusing only on Arabic. Moreover, we report some interesting findings in TTS fine-tuning, as the model learns to synthesize speech without explicit text diacritization in a way that generalizes to unseen domains, which we believe is a result of the unsupervised pre-training on large Arabic speech data. Our main contributions are:

1. Releasing a pre-trained cross-modal transformer model capable of handling diverse speech and text tasks, in addition to fine-tuned ASR and TTS models for MSA[2].

2. Demonstrating state-of-the-art performance in ASR, TTS, and DID, using standard open-domain datasets for MSA.

3. Demonstrating unique generalization capabilities, such as speech synthesis without explicit diacritization.

2 Related Works

To the best of our knowledge, there is no model pre-trained on Arabic that can perform multiple downstream speech-related tasks with different input modalities. In the text domain, AraT5 (Elmadany et al., 2022) was implemented as an Arabic version of the Text-To-Text Transfer Transformer (T5) model (Raffel et al., 2020), which uses transfer learning with a unified Transformer framework for several downstream text generation tasks. In the speech domain, multi-lingual acoustic models, such as XLSR-R (Babu et al., 2022), Whisper (Radford et al., 2023), or MMS (Pratap et al., 2023), include Arabic as one of many languages in supervised or self-supervised pre-training, but they can only handle speech as input modality, and text as output modality. ArTST is directly inspired from the SpeechT5 model (Ao et al., 2022), which is a pre-trained encoder-decoder transformer with additional modal-specific networks to handle both text and speech modalities in the input and output. The model was shown to be versatile as it can achieve superior performance when fine-tuned for ASR, TTS, and other speech related tasks. However, the model was pre-trained only on English data, and as a result, the internal representations seem to be heavily biased towards English speech. By fine-tuning the model for Arabic ASR and TTS, our experiments indicate that it may be difficult to overcome this bias without multi-lingual pre-training.

Several studies attempted to measure the effect multi-lingual pre-training in acoustic models, with mixed results (Yadav and Sitaram, 2022). Heigold et al. (2013) compared models pre-trained on English only with models trained on multi-lingual data using conventional HMM-DNN models, and showed empirically that multilingual pre-training is better than fine-tuning an English model on a different target language. Huang et al. (2013) further shows that multilingual pre-trained features can generalize to unseen languages. Tong et al. (2017) shows that multi-lingual ASR training is worse than monolingual training in the target language, but multilingual pre-training followed by target language fine-tuning is better than monolingual training. Language similarity likely plays a role in generalization: Ram and Aldarmaki (2022)

[1] Pronounced '`artist`'.

[2] `https://github.com/mbzuai-nlp/ArTST`

showed that acoustic word embeddings obtained using Wav2Vec 2.0 features that are pre-trained on English generalize to languages like French and German, but don't generalize as well for Arabic. Furthermore, several studies show that multilingual models generalize better using language vectors or language adapters (Kannan et al., 2019; Toshniwal et al., 2018; Shetty and NJ, 2020; Radford et al., 2023; Pratap et al., 2023), which indicates that some language-specificity in the model is preferable to crude multi-lingual training. Some empirical evidence also suggests that performance of some high-resource languages can potentially degrade in multi-lingual settings compared to monolingual pre-training (Watanabe et al., 2017).

The above mentioned studies all focus on acoustic models where speech is the input rather than the output. Text-to-speech synthesis models, on the other hand, are generally more fragile and highly depend on the quality and size of training data. Generally speaking, TTS models require consistent and clean recordings in order to synthesize natural and intelligible speech (Kulkarni et al., 2023). Multi-lingual TTS synthesis is an emerging topic of research, but these attempts are rare compared to multi-lingual ASR and cover only a small subset of languages due to shortage of resources suitable for speech synthesis (Li et al., 2021; Cho et al., 2022).

3 ArTST

ArTST is a text and speech transformer optimized for the Arabic language. Based on observations from previous studies on multilingual and monolingual ASR, TTS, and self-supervised pre-training, we believe that training a model from scratch with the Arabic language in mind would improve the quality of the resulting models. Our strategy is to start with a monolingual setting, and explore the optimal settings for Modern Standard Arabic (MSA) speech processing. In future iterations of the model, we will explore how best to expand it to handle various dialects as well as other languages that are often mixed with Arabic (i.e. English and French). We believe that an incremental approach of this kind is more likely to lead to optimal performance. Here, we describe the first stage of this project, which focuses only on MSA. ArTST is adapted from the transformer-based SpeechT5 model, which we briefly describe in this section. For more details, please refer to Ao et al. (2022).

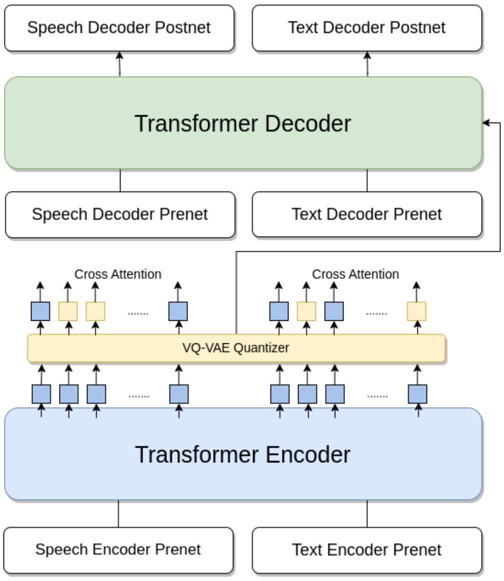


Figure 1: Model architecture.

3.1 Model Architecture

Figure 1 shows the overall architecture of the model. It consists of a main encoder-decoder transformer network, similar to the architecture employed in T5 (Raffel et al., 2020). This network is shared for both speech and text modalities. To account for the differences in pre- and post-processing, additional modal-specific pre- and post-nets are used to handle the text and speech features.

3.2 Pre-training

The model is pre-trained using various self-supervised objectives to account for both speech and text modalities in the input and output:

Speech bidirectional masked prediction: Following the framework of HuBERT (Hsu et al., 2021), discrete frame-level targets are employed for masked prediction, where random spans of 10 steps from the output of the speech encoder pre-net are masked across each utterance, and the model is trained to predict the correct discrete labels via cross-entropy. The discrete labels are obtained from a pre-trained HuBERT model (Hsu et al., 2021), where the hidden representations are clustered into 500 classes using the k-means algorithm. This training objectives can be a stepping stone towards learning speech to text transformation as the model is trained to map continuous speech features into discrete units. This objective updates the speech encoder pre-net as well as the main encoder.

Speech de-noising auto-encoder: This objective trains the speech decoder pre-net, decoder, and speech decoder post-net to reconstruct speech features in the form of 80-dimensional log mel filterbanks from the randomly masked utterances as described above.

Text de-noising auto-encoder: Using unlabeled text, the text encoder pre-net, encoder-decoder network, and text decoder pre- and post-nets, are all optimized using a denoising reconstruction loss.

Cross-modal loss: Vector-quantized embeddings are used to implicitly align speech and text representations through a shared code-book. During training, 10% of the contextual embeddings are replaced with the corresponding quantized embeddings, and the cross-attention in the main encoder-decoder transformer is calculated based on this mixed representation. A diversity loss is used to encourage sharing more codes between the text and speech inputs.

In ArTST, each of the encoder and decoder components are similar in size and configuration to SpeechT5 (Ao et al., 2022). Speech pre/post-nets and text pre/post-nets all have the same structure as in the SpeechT5 model, with the only difference being in the text tokenizer which we initialize using the characters in our training sets. We employed the official HuBERT model[3] to generate the discrete labels for the bidirectional masked prediction objective since a pre-trained Arabic HuBERT model was not available for our perusal. In future work, we will explore the potential of improving this component using a model pre-trained on Arabic speech.

3.3 Fine-Tuning

Task-specific fine-tuning is carried out by employing the encoder-decoder backbone in addition to the relevant pre- and post-nets. For example, for ASR, the speech encoder pre-net, and text decoder pre- and post-nets are used to handle speech input and text output. All relevant model parameters are updated during fine-tuning.

4 Training & Fine-Tuning Settings

4.1 Dataset

For training our MSA ArTST model, we utilize the Multi-Genre Broadcast (MGB2) dataset (Ali et al., 2016), which is collected from Aljazeera TV recordings of Arabic speech, mostly in MSA. This dataset is often used for benchmarking ASR models for MSA, which enables fair comparison with previous research. The original dataset contains 1.4K unique speakers with ∼1.2K hours of transcribed speech data. We excluded overlapping speech utterances from the set, which are tagged in the corpus. Furthermore, to avoid high amount of padding and maintain a balance between computational efficiency and effectiveness, we excluded speech samples that exceeded a duration of 40 seconds. The resulting dataset consists of roughly 1K hours of speech. We also randomly extracted a 200 hr subset of MGB2 for the purpose of performing preliminary experiments to evaluate SpeechT5 fine-tuning on ASR. Moreover, we extracted a random subset from the QASR corpus (Mubarak et al., 2021), a multi-dialectal broadcast speech corpus from Aljazeera that includes MSA speech as well as dialectal Arabic of different varieties. As we are focusing mainly on MSA in this work, we do not utilize this dataset for pre-training, but instead utilize it to test the generalization potential of the model. For TTS fine-tuning, we utilize open-source Arabic datasets curated for speech synthesis, namely: The Arabic Speech Corpus (ASC) (Halabi et al., 2016) and Classical Arabic Text-to-Speech Corpus (ClArTTS) (Kulkarni et al., 2023)[4]. We also utilize these two datasets for evaluating the ASR models. For all datasets, we use the predefined test/dev splits if applicable. We summarize all dataset statistics in Table 1.

4.2 Text & Speech Pre-Processing

All punctuation marks were removed with the exception of `@` and `%`. Additionally, all diacritics were removed, and Indo-Arabic numerals were replaced with Arabic numerals to ensure uniformity. The vocabulary is comprised of individual Arabic alphabets, numerals, and select English characters from the training dataset, in addition to some special characters like `@` and `%`. For speech data, we standardized the sampling rate to be 16 kHz across all collected datasets.

[3] https://github.com/facebookresearch/fairseq/blob/main/examples/hubert

[4] www.clartts.com

	Split	# of Hours	# of Words
MGB2	MGB2-1K (train)	1005.39	6.96M
	MGB2-200 (train)	201.32	1.39M
	test	9.57	64.38K
QASR	QASR-267 (train)	267.91	2.00M
	test	9.57	64.38K
ASC	train	3.81	20.58K
	test	0.28	1.40K
ClArTTS	train	11.16	76.27K
	test	0.24	1.69K

Table 1: Datasets used in our experiments.

	Train set / Test set	Enc	WER ↓	CER ↓
SpeechT5	ASC / ASC	Ar	78.07%	23.54%
		Bw	76.92%	22.02%
ArTST	ASC / ASC	Ar	45.8%	9.88%
SpeechT5	ClArTTS / ClArTTS	Ar	32.31%	6.88%
		Bw	24.32%	5.12%
ArTST	ClArTTS / ClArTTS	Ar	12.51%	3.60%
SpeechT5	MGB2-200 / MGB2	Ar	69.74%	26.47%
		Bw	45.09%	17.55%
ArTST	MGB2-200 / MGB2	Ar	16.56%	7.68%
SpeechT5	QASR-267 / MGB2	Ar	72.70%	26.27%
		Bw	53.19%	19.01%
ArTST	QASR-267 / MGB2	Ar	17.27%	9.99%

Table 2: Fine-tuned ASR resutls using English SpeechT5 vs. ArTST in terms of Word Error Rate (WER) and Character Error Rate (CER). Character Encoding (Enc): Arabic (Ar), BuckWalter (Bw).

4.3 ArTST Pre-training

We pre-trained ArTST using the MGB2-1K subset. Since the pre-training is unsupervised, aligned text and speech data are not required at this stage. For text pre-training, we employed the cleaned transcriptions from the MGB2 dataset as unlabeled data. We pre-trained ArTST using Adam optimizer with a learning rate of 2×10^{-4}, spanning 200K updates, and a warm-up phase of 64K updates. The maximum speech token length was set at 250K (equivalent to 15.625 seconds), and the text tokens were capped at 600 characters. The pre-training was run on four A100 GPUs for 14 days.

5 Results & Evaluation

5.1 SpeechT5 Finetuning vs. ArTST

We conducted preliminary assessments of the SpeechT5 model from Ao et al. (2022), which was pre-trained and fine-tuned on English, to assess the ability of cross-lingual transfer by directly fine-tuning the model for Arabic ASR using various Arabic speech datasets. We experimented with both the original Arabic script as input, as well as Buckwalter transliteration (Habash et al., 2007) instead of Arabic script to account for the fact that the model was pre-trained only on English characters.

For Arabic script, we augmented the original character tokenizer to incorporate symbols that correspond to Arabic letters and special symbols contained in the fine-tuning set. The original tokenizer contained approximately 80 symbols; after incorporating the Arabic letters and special symbols, the extended tokenizer vocabulary increased to 130 symbols. Furthermore, we modified the input embeddings structure to align with the dimensionality of the updated tokenizer. The embedding layer retains the weights from the earlier-trained SpeechT5 model for its initial 80 components. Meanwhile, additional elements were initialized randomly. Similarly, for Buckwalter transcriptions, we modified the tokenizer accordingly. Since the transliteration scheme contains mostly English alphabets in addition to some special ASCII characters, the extended vocabulary in this setting was increased to 90 characters. We start with the pre-trained English ASR from SpeechT5[5] and fine-tune it on the specified datasets until training and validation loss diverge.

Table 2 shows the results in terms of Word Error Rate (WER) and Character Error Rate (CER) in all different settings. The ArTST ASR model was fine-tuned using our pre-trained ArTST using the same tokenizer as the pre-trained model, which contains Arabic script.

Effect of Input Encoding

We see from these experiments that SpeechT5 fine-tuning is improved using Buckwalter rather than Arabic script. Since the transcription scheme mostly results in mapping Arabic letters to similar-sounding English letters, the learning objective does not diverge greatly from the original English model, which results in improved performance compared to using Arabic script. In our analysis, approximately 85% of Arabic characters were replaced with corresponding English characters, facilitating the continuation of fine-tuning for SpeechT5's ASR, even with limited data.

[5] huggingface.co/microsoft/speecht5_asr

Model	WER ↓	CER ↓
From (Hussein et al., 2022):		
HMM-DNN	15.80%	—
E2E, CTC + LM	16.90%	—
E2E, Attention + LM	13.40%	—
E2E, CTC , Attention + LM	**12.50%**	—
ArTST	13.42%	6.43%
ArTST + LM	**12.78%**	**6.33%**

Table 3: Comparing ArTST performance against models reported in (Hussein et al., 2022), which include best performing model previously reports on MGB2.

Effect of Pre-Training

We also observe large reductions in error rate using the same datasets for fine-tuning ArTST. The difference in performance is evident in all cases, but it's particularly large for the ASC and MGB2-200 subsets. SpeechT5 fine-tuned with Buckwalter transcriptions on the ClArTTS corpus results in relatively good performance of 24% WER compared to 12.78% WER for ArTST. For the other two datasets, the difference is roughly 30% absolute WER in favor of ArTST. This could potentially be resulting from two factors: ClArTTS is a consistent and clean dataset that was curated for TTS, compared to MGB2 which is extracted from TV shows. ASC is also curated for TTS, and therefore consists of clean and consistent recordings, but dataset size could have played a role in the high WER for ASC, which is much smaller than the ClArTTS dataset (∼3.8 hrs compared to ∼11.16 hrs). While MGB2 contains orders of magnitude more data than ClArTTS, the error rates are higher than ClArTTS for all models, including ArTST, which is further evidence that dataset quality is most likely playing a role in these results.

Finally, we also used a subset of QASR for fine-tuning ASR models as a counterpoint for the MGB2 datasets because the latter was used in pre-training and could have biased the results in favor of ArTST. However, even in this set, we clearly see that ArTST performs much better than the fine-tuned SpeechT5, with error rates on a par with the ones observed for MGB2.

5.2 Benchmarking ArTST for MSA

We fine-tuned ArTST on our MGB2-1K dataset, and compared the performance against comparable models trained and tested on MGB2. Since 2017, the lowest WER on MGB2 test set was reported in Smit et al. (2017) as 13.2%. Recently, Hussein et al. (2022) explored the potential of an end-to-end transformer model compared to conventional ASR systems, and achieved state-of-the-art performance in the MGB2 test set. The model was trained on the MGB2 dataset, so it's comparable to our model in that regard. Furthermore, they utilize a language model for rescoring using the MGB2 transcriptions as well as the additional 130M words of text data provided in the MGB2 challenge. Our model consists of the speech pre-net, encoder, and text pre/post-nets fine-tuned with CTC loss. We also experiment with LM shallow fusion using a transformer-based auto-regressive character language model trained on the same sets. We used the default LM setting from the Fairseq library[6], and we trained the model for 300K updates using the Adam optimizer, with 4K warm-up steps, a learning rate of 0.0005, and 0.1 dropout rate.

The results are shown in Table 3. Our model without LM fusion achieves 13.42% WER, which is on a par with the transformer-based end-to-end model with attention and LM rescoring reported in Hussein et al. (2022). Furthermore, ArTST outperforms the architecture most similar to it (E2E, CTC + LM) by more than 3% absolute WER, without incorporating a language model for inference. The error rates are further reduced to 12.78% by incorporating LM fusion, which is comparable to the best model reported in Hussein et al. (2022); the latter incorporates both Attention and CTC, as well as LM rescoring with beam size of 20.

5.3 Comparing ArTST With Multilingual Models

Recently, a few large multi-lingual pre-trained models have been released for ASR in multiple languages, such as `Whisper` (Radford et al., 2023), and `MMS` (Pratap et al., 2023). Both models include Arabic as one of many languages included in their supervised pre-training. Training data, model architectures, training objectives, and model sizes vary considerably between these models, so they are not directly comparable, However, the fact that these models are widely circulated and used necessitates some kind of performance comparison with our model.

Table 4 shows the WER/CER of these models in Arabic ASR using our test sets. We also report the number of parameters for each model.

[6] `github.com/facebookresearch/fairseq`

Test Set	ArTST		Whisper$_{medium}$		Whisper$_{large}$		MMS$_{medium}$		MMS$_{large}$	
	WER	CER	WER	CER	WER	CER	WER	CER	WER	CER
ASC	45.70%	9.73%	48.46%	10.74%	47.73%	10.83%	54.05%	11.71%	57.37%	11.13%
ClArTTS	13.52%	3.90%	20.49%	6.24%	19.25%	6.23%	36.18%	9.17%	31.13%	6.58%
MGB2	13.42%	6.43%	28.69%	11.72%	26.71%	10.78%	45.58%	14.86%	40.33%	13.06%
QASR(1hr)	26.08%	16.65%	36.54%	17.45%	32.32%	15.56%	52.79 %	20.86 %	47.81 %	18.80%
# params	155 M		769 M		1550 M		300 M		965 M	

Table 4: ArTST compared with large multi-lingual models: Whisper & MMS on our test sets. ArTST was fine-tuned for ASR using MGB2-1k train set. Results are shown without LM fusion.

While Whisper performs relatively well compared to MMS, ArTST outperforms both models, including the large variant of each model, in all test sets, while having a smaller number of parameters. For instance, without LM fusion, ArTST achieved 13.5% WER on MGB2 test set, while the large variants of Whisper and MMS achieved 26.7% and 40.3% WER, respectively.

5.4 Qualitative Analysis of ASR Output

In Table 5, we show some examples of ASR outputs from ArTST compared with the reference transcriptions. These examples show the drawback of the raw WER/CER metrics as they don't account for potential variations in spelling. In particular, we observed several cases where English words are transliterated or misspelled. Furthermore, numeric expressions, like 80%, are in some cases written in numeric format, and others spelled out in words. Furthermore, the large error rates reported for ASC are in a large part caused by intentional misspelling in the reference ASC transcriptions, which are intended to facilitate learning of TTS synthesis in a low-resource setting. In the shown examples, ArTST output is in fact the correct spelling. We also show a couple of examples of ArTST, which is fine-tuned on MGB2, generalizing to dialectal Arabic utterances from QASR.

Examples from MGB2 Test
أنه إذا أنت تطرح الأبرتهايد يجب أن تطرح حالة من المساواة للجميع أنه إذا أنت تطرح الـ apartheid يجب أن تطرح حالة من المساواة للجميع
وتشرفهم وتكرمهم بل في الثمانين بالمائة الذين لم ينجحوا لا هم معدون لشيء وتشرفهم وتكرمهم بل في 80% الذين لم ينجحوا لا هم معدون لشيء
Examples from ASC Test
وبالتالي تساعد على لوقاية من لإمساك وبالتالي تساعد على الويقاية من الإنساك
وذلك على خلاف نظرائه لسابقين وذلك على خلاف نظرائه السابق
Examples from QASR Test
أنه كأنه والله الجندي السوري السني الذي يقاتل هو فقط يقاتل خوفا أنه كأنه والله الجندي السوري السني الذي يقاتل هو فقط يقاتل خوفا أنا لا
وا لأولاد يمسكوا و يقعدوا هناك في الجنينة عشان يعملوا بعض مظاهر تتعلق بالثورة وا لولاد يمسكوا يقعدوا هناك في الجنينة عشان يعملوا بعض مظاهر تتعلق بالثورة

Table 5: Sample ArTST ASR transcriptions (bottom) vs. reference transcriptions (top). Highlighting differences or errors. Correct words not present in ArTST or reference. Correct words in ArTST but not present in reference.

5.5 ArTST for TTS Synthesis

We experimented with TTS fine-tuning, comparing ArTST pre-trained model with SpeechT5 TTS[7] as a starting point. We fine-tuned each model using the ClArTTS and the ASC datasets, which are two open-source datasets curated for Arabic TTS. For the SpeechT5 model, we used Buckwalter transcriptions for the text, as our experiments in ASR demonstrated it to be more suitable for this model. For both models, we fine-tuned the TTS model without using input diacritics, so no automatic diacritizer is needed for inference. This feature diverges from previous works in Arabic TTS, where efforts are taken to include diacritization in the input text. However, since this would necessitate the use of text-based diacritizers for inference, and as shown in Aldarmaki and Ghannam (2023), text-based diacritizers have high error rates when applied to the speech domain. We opted to train undiacritized TTS instead, and let the model implicitly learn the correct pronunciation.

The fine-tuning was carried out using the text encoder pre-net, encoder/decoder backbone, and speech decoder pre/post-nets. All model parameters were updated during fine-tuning. We used the pre-trained HiFi-GAN vocoder[8] to convert the output of each model to waveform.

[7] huggingface.co/microsoft/speecht5_tts

[8] huggingface.co/microsoft/speecht5_hifigan

	Fine-tuning Data	MOS ↑
Ground Truth	ASC	4.31
	ClArTTS	4.64
English SpeechT5	ASC	1.57
	ClArTTS	1.88
ArTST	ASC	2.93
	ClArTTS	4.11
ArTST*	ASC	3.44
	ClArTTS	4.31

Table 6: Subjective listening tests in terms of Mean Openion Score (MOS) for models fine-tuned using English SpeechT5 vs. ArTST, vs. ArTST* (variant of TTS model pre-trained on MGB-2 data).

TTS Pre-Training

Since both ASC and ClArTTS are relatively small datasets, we also experimented with TTS pre-training using ASR data from MGB2. Generally speaking, ASR data are not suitable for TTS training due to the high variability is speaking style and presence of noise. On the other hand, ASR data are available in abundance, and can potentially help improve the model's generalization potential. We start by fine-tuning the TTS model on MGB2-1K train set, and then fine-tune it again on the TTS train sets. We refer to this variant as ArTST*.

TTS Evaluation

We conducted subjective evaluation through listening tests to assess the naturalness and intelligibility of the synthesized speech from differnet models in a single score from 1 to 5 (higher is better). We selected random utterances from each test set, and synthesized speech based on the corresponding text transcription using the variants speechT5, ArTST and ArTST*. Fifteen native Arabic speakers participated in the evaluation. The Mean Openion Score (MOS) for each model is shown in Table 6. The audio samples used in the evaluation are available here [9]. As seen from the table, and through the provided samples, using the pre-trained SpeechT5 model as a basis for fine-tuning leads to very poor speech synthesis. On the other hand, using the pre-trained ArTST as a basis for fine-tuning results in high-quality synthesis. Furthermore, pre-training the TTS model using MGB2 ASR data further improves the quality of the transcriptions. Moreover, we observed through listening tests that the model generalizes to unseen sentences from MSA, where we synthesized speech from transcriptions obtained from QASR[10]. In particular, the model learns to produce the correct pronunciation in spite of not being provided with any diacritics.

Model	Dev	Test
E2E (softmax) (Shon et al., 2020)	83.00%	82.00%
HuBERT-17 (Sullivan et al., 2023)	92.23%	92.12%
XLS-R-300M-17 (Sullivan et al., 2023)	90.77%	90.20%
ArTST	**95.08%**	**94.18%**
MGB-5 Challenge (Ali et al., 2019) Top 2 Systems:		
UKent	93.50%	93.10%
DKU [Single best system]	94.70%	93.80%
DKU [Fusion of 4 systems]	97.40%	94.90%

Table 7: Accuracy results for dialect identification on the ADI17 set.

5.6 Dialect Identification

To fine-tune ArTST for speech classification, we recast the multi-class classification task as a speech to text generation task. The decoder is then trained to predict the dialect class at the first time step (which is equivalent to a regular softmax classifier). We fine-tuned all parameters using the Arabic Dialect Identification for 17 countries (ADI17) dataset (Shon et al., 2020). We compared our model to previously reported results in Table 7. As seen from these results, ArTST outperforms previous models, including the best single system submitted to the MGB-5 challenge (Ali et al., 2016), and is not far behind the top model which fuses 4 different system; it is worth noting that the latter also incorporates data augmentation to further improve performance, which we do not explore in this work.

6 Conclusions & Future Work

We demonstrated the potential of ArTST in speech recognition, synthesis, and classification, where we achieved results on a par with or outperforming previously reported results with relatively straightforward fine-tuning. What we have demonstrated in this paper is only a subset of potential applications of this framework. As the model can handle both text and speech modalities, it can potentially be applied for text-to-text and speech-to-speech applications, in addition to text classification and generation tasks. We will explore these avenues of application in future work. In this initial work, we focused on MSA as the main variant of Arabic

[9] https://artstts.wixsite.com/artsttts

[10] Samples are available in the same website.

for pre-training. We explored the potential of the model to generalize to dialectal Arabic using small test sets that include dialectal Arabic, as well as the dialect identification task. Future edition will focus on expanding the coverage of the pre-trained model to include various dialects, and potentially code-switched speech, without sacrificing performance on MSA. As demonstrated in this paper, our model outperforms larger multi-lingual models like Whisper and MMS, which we believe is a result of focusing on the Arabic language as a basis of our model from its inception. While multi-linguality may be desirable for some applications, and could be beneficial for low-resource languages, monolingual models have a greater potential for high-resource languages, and the Arabic language currently boasts large volumes of open-source datasets that can be utilized to develop high-quality models across various tasks.

7 Limitations

As this is a large on-going project comprising several tasks and potential variations in pre-training, there are several limitations that can be acknowledged here. First, the model's pre-training consists of mainly MSA speech from a single dataset (MGB2). While this dataset is large and comparable to the pre-training conditions in SpeechT5, there are other datasets that could be incorporated to potentially improve performance. Furthermore, we did not focus on dialectal Arabic in this edition, and only alluded to potential generalization to dialects through some experiments on ASR and dialect identification. Given small amount of code-switching in the MGB2 set, the model does have limited code-switching recognition, but it can be improved by intentionally using code-switching dataset for pre-training and fine-tuning. One more limitation is the use of pre-trained HuBERT for generating intermediate discrete labels in the pre-training stage. While our model demonstrably achieves excellent results in all tested tasks in spite of that, we did not explore the possibility of optimizing HuBERT for Arabic, mainly due to the additional computational load for training another large model. Finally, we did not probe the internal representations of the model to explore potential architectural improvements. Further analysis of these representations, and a thorough analysis of the dialect identification model could shed light on the properties of these representations.

References

Ibrahim M Alabdulmohsin, Behnam Neyshabur, and Xiaohua Zhai. 2022. Revisiting neural scaling laws in language and vision. *Advances in Neural Information Processing Systems*, 35:22300–22312.

Hanan Aldarmaki and Ahmad Ghannam. 2023. Diacritic Recognition Performance in Arabic ASR. In *Proc. INTERSPEECH 2023*, pages 361–365.

Ahmed Ali, Peter Bell, James Glass, Yacine Messaoui, Hamdy Mubarak, Steve Renals, and Yifan Zhang. 2016. The mgb-2 challenge: Arabic multi-dialect broadcast media recognition. In *2016 IEEE Spoken Language Technology Workshop (SLT)*, pages 279–284. 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.

Junyi Ao, Rui Wang, Long Zhou, Chengyi Wang, Shuo Ren, Yu Wu, Shujie Liu, Tom Ko, Qing Li, Yu Zhang, Zhihua Wei, Yao Qian, Jinyu Li, and Furu Wei. 2022. SpeechT5: Unified-modal encoder-decoder pre-training for spoken language processing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5723–5738, Dublin, Ireland. Association for Computational Linguistics.

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 *Interspeech 2022, 23rd Annual Conference of the International Speech Communication Association, Incheon, Korea, 18-22 September 2022*, pages 2278–2282. ISCA.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460.

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

Hyunjae Cho, Wonbin Jung, Junhyeok Lee, and Sang Hoon Woo. 2022. Sane-tts: Stable and natural end-to-end multilingual text-to-speech. In *23rd Annual Conference of the International Speech Communication Association, INTERSPEECH 2022*, pages 1–5. ISCA-INT SPEECH COMMUNICATION ASSOC.

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.

AbdelRahim Elmadany, Muhammad Abdul-Mageed, et al. 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.

Alex Graves. 2012. Connectionist temporal classification. *Supervised Sequence Labelling with Recurrent Neural Networks*, pages 61–93.

Nizar Habash, Abdelhadi Soudi, and Timothy Buckwalter. 2007. On arabic transliteration. *Arabic computational morphology: Knowledge-based and empirical methods*, pages 15–22.

Nawar Halabi et al. 2016. Arabic speech corpus. *Oxford Text Archive Core Collection*.

Georg Heigold, Vincent Vanhoucke, Alan Senior, Patrick Nguyen, Marc'Aurelio Ranzato, Matthieu Devin, and Jeffrey Dean. 2013. Multilingual acoustic models using distributed deep neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 8619–8623. IEEE.

Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory F. Diamos, Heewoo Jun, Hassan Kianinejad, Md. Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou. 2017. Deep learning scaling is predictable, empirically. *CoRR*, abs/1712.00409.

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.

Jui-Ting Huang, Jinyu Li, Dong Yu, Li Deng, and Yifan Gong. 2013. Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 7304–7308. IEEE.

Amir Hussein, Shinji Watanabe, and Ahmed Ali. 2022. s. *Computer Speech & Language*, 71:101272.

Anjuli Kannan, Arindrima Datta, Tara N Sainath, Eugene Weinstein, Bhuvana Ramabhadran, Yonghui Wu, Ankur Bapna, Zhifeng Chen, and Seungji Lee. 2019. Large-scale multilingual speech recognition with a streaming end-to-end model. *Proc. Interspeech 2019*, pages 2130–2134.

Ajinkya Kulkarni, Atharva Kulkarni, Sara Abedalmon'em Mohammad Shatnawi, and Hanan Aldarmaki. 2023. Clartts: An open-source classical arabic text-to-speech corpus. In *2023 INTERSPEECH*, pages 5511–5515.

Song Li, Beibei Ouyang, Lin Li, and Qingyang Hong. 2021. Light-tts: Lightweight multi-speaker multi-lingual text-to-speech. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8383–8387. IEEE.

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, Online. Association for Computational Linguistics.

Vineel Pratap, Andros Tjandra, Bowen Shi, Paden Tomasello, Arun Babu, Sayani Kundu, Ali Elkahky, Zhaoheng Ni, Apoorv Vyas, Maryam Fazel-Zarandi, et al. 2023. Scaling speech technology to 1,000+ languages. *arXiv preprint arXiv:2305.13516*.

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.

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. *J. Mach. Learn. Res.*, 21(1).

Sreepratha Ram and Hanan Aldarmaki. 2022. Supervised acoustic embeddings and their transferability across languages. In *Proceedings of the 5th International Conference on Natural Language and Speech Processing (ICNLSP 2022)*, pages 212–218.

Vishwas M Shetty and Metilda Sagaya Mary NJ. 2020. Improving the performance of transformer based low resource speech recognition for indian languages. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8279–8283. IEEE.

Suwon Shon, Ahmed Ali, Younes Samih, Hamdy Mubarak, and James Glass. 2020. Adi17: A fine-grained arabic dialect identification dataset. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8244–8248.

Peter Smit, Siva Reddy Gangireddy, Seppo Enarvi, Sami Virpioja, and Mikko Kurimo. 2017. Aalto system for the 2017 arabic multi-genre broadcast challenge. In *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 338–345. IEEE.

Peter Sullivan, AbdelRahim Elmadany, and Muhammad Abdul-Mageed. 2023. On the Robustness of Arabic Speech Dialect Identification. In *Proc. INTERSPEECH 2023*, pages 5326–5330.

Sibo Tong, Philip N Garner, and Hervé Bourlard. 2017. Multilingual training and cross-lingual adaptation on ctc-based acoustic model. *arXiv preprint arXiv:1711.10025*.

Shubham Toshniwal, Tara N Sainath, Ron J Weiss, Bo Li, Pedro Moreno, Eugene Weinstein, and Kanishka Rao. 2018. Multilingual speech recognition with a single end-to-end model. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4904–4908. IEEE.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Shinji Watanabe, Takaaki Hori, and John R Hershey. 2017. Language independent end-to-end architecture for joint language identification and speech recognition. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 265–271. IEEE.

Hemant Yadav and Sunayana Sitaram. 2022. A survey of multilingual models for automatic speech recognition. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5071–5079.

TARJAMAT: Evaluation of Bard and ChatGPT on Machine Translation of Ten Arabic Varieties

Karima Kadaoui[λ,⋆] **Samar M. Magdy**[λ,⋆] **Abdul Waheed**[λ,⋆] **Md Tawkat Islam Khondaker**[ξ,⋆]
Ahmed Oumar El-Shangiti[λ] **El Moatez Billah Nagoudi**[ξ] **Muhammad Abdul-Mageed**[ξ,λ,⋆]
[ξ] Deep Learning & Natural Language Processing Group, The University of British Columbia
[λ]Department of Natural Language Processing & Department of Machine Learning, MBZUAI
{karima.kadaoui,samar.magdy,abdul.waheed,ahmed.oumar}@mbzuai.ac.ae
{tawkat@cs,moatez.nagoudi,muhammad.mageed}@ubc.ca

Abstract

Despite the purported multilingual proficiency of instruction-finetuned large language models (LLMs) such as ChatGPT and Bard, the linguistic inclusivity of these models remains insufficiently explored. Considering this constraint, we present a thorough assessment of Bard and ChatGPT (encompassing both GPT-3.5 and GPT-4) regarding their machine translation proficiencies across ten varieties of Arabic. Our evaluation covers diverse Arabic varieties such as Classical Arabic (CA), Modern Standard Arabic (MSA), and several country-level dialectal variants. Our analysis indicates that LLMs may encounter challenges with dialects for which minimal public datasets exist, but on average are better translators of dialects than existing commercial systems. On CA and MSA, instruction-tuned LLMs, however, trail behind commercial systems such as Google Translate. Finally, we undertake a human-centric study to scrutinize the efficacy of the relatively recent model, Bard, in following human instructions during translation tasks. Our analysis reveals a circumscribed capability of Bard in aligning with human instructions in translation contexts. Collectively, our findings underscore that prevailing LLMs remain far from inclusive, with only limited ability to cater for the linguistic and cultural intricacies of diverse communities.

1 Introduction

Large language models (LLMs) finetuned to follow instructions (Wei et al., 2021; Wang et al., 2022; Ouyang et al., 2022) have recently emerged as powerful systems for handling a wide range of NLP tasks. In accordance with the scaling law (i.e., pretraining larger models will continue to result in better performance) (Kaplan et al., 2020), a number of LLMs such as GPT-3 (Brown et al., 2020), Chinchilla (Hoffmann et al., 2022), Claude (Anthropic, 2023), ChatGPT[1] (OpenAI, 2022), GPT-4 (OpenAI, 2023), and Bard (Google, 2023) have been introduced. Most of these models, however, are 'closed'. That is, little-to-no information about them is known. This includes details about model architectures, pretraining data, languages involved, and training configurations. LLMs are also expensive both to pretrain and deploy. To alleviate these concerns, 'open' LLMs such as BLOOM (Scao et al., 2022), LLaMA-1 (Touvron et al., 2023a), Falcon (Almazrouei et al., 2023), and LLaMA-2 (Touvron et al., 2023b) were introduced. These more open models can facilitate research and (non-)commercial deployment.

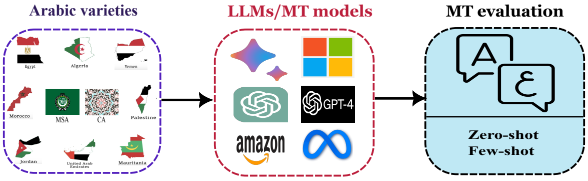


Figure 1: Experimental setup for our evaluation. We evaluate multiple language models on different Arabic varieties.

In spite of drawbacks such as their closed nature, computational costs (Dasgupta et al., 2023), and biases they exhibit (Ferrara, 2023), closed LLMs remain attractive primarily due to their remarkable performance (Bang et al., 2023a; Laskar et al., 2023a). It is thus important to fully understand the full capabilities of these closed models. Although there has been a recent flurry of works attempting to evaluate ability of LLMs to carry out NLP tasks, many of these models remain opaque. This is especially the case when it comes to understanding how LLMs fare on different varieties and dialects of several popular languages and on vital tasks such as machine translation (MT). For example, the extent to which LLMs can handle MT from Arabic varieties into other languages is unknown.

⋆Equal contribution

[1] In this work, we refer `gpt-3.5-turbo` as ChatGPT.

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 52–75
December 7, 2023 ©2023 Association for Computational Linguistics

Another challenge is how more recent models such as Google's Bard are yet to be evaluated and understood. Bard was released in 41 different languages, which makes it a particularly attractive target for MT evaluation. This is also the case given Google's strong history of investment in MT (Wu et al., 2016a). In this work, we offer a thorough evaluation of LLMs on MT from major Arabic varieties into English (Figure 1). Namely, we evaluate ChatGPT, GPT-4, and Bard on MT of ten Arabic varieties into English. Since there are usually concerns about downstream evaluation data leaking into LLM pretraining, which involves data collected from the web, we benchmark the models on new test sets that we manually prepare for this work. Our evaluation targets diverse varieties of Arabic. Namely, we evaluate on Classical Arabic (CA), Modern Standard Arabic (MSA), and several country-level Arabic dialects such as Algerian and Egyptian Arabic (Section 3).

Bard provides three different drafts for each text input we ask it to translate. Contents of the three drafts are diverse, providing us with excellent contexts to analyze the degree to which the model adheres to our prompts. We leverage these contexts to carry out a human evaluation study investigating the *helpfulness* of the model, allowing us to reveal a number of Bard's limitations. We carefully analyze these limitations against the different Arabic varieties we target, thus affording even better understanding of the model's ability to translate from Arabic.

Overall, our work offers the following contributions:

(i) We offer a detailed MT evaluation of instruction finetuned LLMs on ten diverse varieties of Arabic.

(ii) To the best of our knowledge, our work is the first to assess performance of Bard on NLP tasks in any language, and on Arabic MT in particular.

(iii) We introduce a new manually created multi-Arabic dataset for MT evaluation that has never been exposed to any existing LLM.

(iv) We extensively evaluate Bard through a human study to analyze its behavior in terms of *helpfulness*. We examine how well the model follows human instructions when tasked with translating across ten different Arabic varieties.

The rest of the paper is organized as follows: In Section 2, we review previous research evaluating LLMs on NLP tasks in general and MT in particular. In Section 3, we introduce our newly developed multi-Arabic MT dataset. In Section 4, we describe our evaluation methods. In Section 5, we present our results and the main findings obtained from comparing ChatGPT and Bard to various commercial MT products. In Section 6, we present our human study analyzing Bard's helpfulness, particularly in terms of its ability to follow human instructions in MT. We conclude in Section 7.

2 Related Work

Evaluation of ChatGPT and Other LLMs. A growing body of literature has focused on evaluating ChatGPT and other LLMs on NLP tasks. Laskar et al. (2023a) find ChatGPT effective on many tasks. Other works find it either on par with supervised models (Ziems et al., 2023) or in some cases (e.g., sequence tagging) falling behind these models (Qin et al., 2023). Both Jiao et al. (2023) and Ogundare and Araya (2023) find that GPT-4 is competitive with commercial systems for high-resource languages but lags behind for low-resource languages. Bang et al. (2023b) find a similar pattern for ChatGPT. Guerreiro et al. (2023) find complex translation scenarios, such as in the low-resource setting, to be prone to hallucination. Peng et al. (2023) demonstrate that ChatGPT can surpass Google Translate on many translation pairs, but Zhu et al. (2023) show it is outperformed by NLLB (NLLB et al., 2022) on at least 83% of the English-centric pairs they study. Wang et al. (2023); Karpinska and Iyyer (2023), however, show that ChatGPT can match the performance of fully supervised models for document-level translation.

Peng et al. (2023) find that adding task and domain-specific information in the prompt can improve the robustness of the MT system, which corroborates the findings by Gao et al. (2023). Huang et al. (2023) propose a prompting technique called cross-lingual-thought prompting (XLT) to improve cross-lingual performance for a wide range of tasks, including MT. Similarly, Lu et al. (2023b) asks ChatGPT to correct its own mistakes as a way to improve the model's translation quality. Lu et al. (2023a) propose Chain-of-Dictionary (CoD) prompting to solve rare word translation issues. Prompting with CoD improves the performance of ChatGPT for both X-En and En-X language directions.

Variety	Example with English Translation
EGY	ماحنا لو فضلنا مدارين و مستخبين هنموت من الخوف. And if we keep hiding, we’re going to die out of fear
JOR	أنا مش مستخف فيه و لا يمكن استخف فيه مهما كان بضل ابوي I do not and cannot underestimate him; he is still my father, no matter what.
MAU	شوف آن شي كامل ادخلتو ماتيت انركيلي فيه خالك اللا القدام. Look, whenever I’m in, I never take a step back; I only go forward.
YEM	ركزت لي نفطة تفتيش، في الباب، بتفتش ذي داخلي و ذي خارجي. I set up a checkpoint at the door to screen anyone who comes in or out.

Table 1: Example sentences from some of the Arabic varieties in our new translation evaluation dataset. See Appendix Table 16 for remaining varieties.

Prompt	Template	BLEU
ENG	Translate the following Modern Standard Arabic (MSA) sentence into English	48.48
MSA	ترجم الجملة العربية الفصحى العصرية التالية الي اللغة الإنجليزية	47.92
ENG (**elaborate**)	I want you to act as an expert translator. You will translate Modern Standard Arabic (MSA) sentences into English. I will give you a Modern Standard Arabic (MSA) input, and you will translate it into English and keep the same semantic meaning. Please translate this Modern Standard Arabic (MSA) text into English	46.17

Table 2: Performance of ChatGPT on the MSA→English translation task. Our concise English prompt outperforms other prompts in BLEU score.

Evaluation of ChatGPT on Arabic. Khondaker et al. (2023) evaluate ChatGPT and other contemporary LLMs such as BloomZ (Muennighoff et al., 2022) in few-shot settings (0, 1, 3, 5, and 10) on four X-Arabic and two code-mixed Arabic-X language sets. They show that providing in-context examples to ChatGPT achieves comparable results to a supervised baseline. Alyafeai et al. (2023) evaluate ChatGPT and GPT-4 on 4, 000 Arabic-English sentence pairs from Ziemski et al. (2016) and find a supervised SoTA model to outperform ChatGPT and GPT-4 by a significant margin. These works, however, only consider a limited number of Arabic varieties. They also do not conduct a thorough analysis of the LLMs for MT. Additionally, none of the works evaluate Bard. Our work bridges these gaps by performing a comprehensive evaluation of these systems on a wide range of Arabic varieties. We also conduct our study on novel in-house data that we guarantee no leakage for (i.e., our data cannot have been seen by ChatGPT, GPT-4, or Bard since we create the data for this work). Other works have focused on evaluating smaller-sized Arabic language models (Abu Farha and Magdy, 2021; Inoue et al., 2021; Alammary, 2022), including on recent benchmarks (Nagoudi et al., 2023; Elmadany et al., 2023).

Arabic MT. There are several works on Arabic MT itself, including rule-based (Bakr et al., 2008; Mohamed et al., 2012; Salloum and Habash, 2013), statistical (Habash and Hu, 2009; Salloum and Habash, 2011; Ghoneim and Diab, 2013), and neural (Junczys-Dowmunt et al., 2016; Almahairi et al., 2016; Durrani et al., 2017; Alrajeh, 2018). While these systems focus on MSA, others target Arabic dialects (Zbib et al., 2012; Sajjad et al., 2013; Salloum et al., 2014; Guellil et al., 2017; Baniata et al., 2018; Sajjad et al., 2020; Farhan et al., 2020; Nagoudi et al., 2021, 2022a). We provide a more detailed review of related literature in Appendix A, with a summary in Table 7.

3 Coverage and Datasets

3.1 Arabic Varieties

Our goal is to provide a comprehensive evaluation of MT on ChatGPT, GPT-4, and Google Bard, focusing on their performance across ten different varieties of Arabic. These can vary across *time* (i.e., old vs. modern day) and *space* (e.g., country-level geography) as well as their *sociopragmatic* functions (e.g., standard use in government communication vs. everyday street language). Before introducing our dataset, we provide a brief background about Arabic and its varieties. Arabic, the collection of languages spoken by approximately 450 million people across the Arab world, encompasses a broad spectrum of varieties. Classical Arabic (CA) is known as Quranic Arabic, the language of the Quran (Rabin, 1955), and has emerged from the medieval dialects of the Arab tribes. It was spoken early in Mecca around 1, 500 years ago in the sixth or seventh century AD. CA is considered the most eloquent form of Arabic and is preserved notably in the Holy Quran and pre-Islamic epic poems (Versteegh, 2014). It is often described as exhibiting archaic words, figurative speech, and rhyming sentences that are no longer (or less frequently) used in MSA and dialectal Arabic varieties. Modern Standard Arabic (MSA) (Holes, 2004), on the contrary, is deeply rooted in CA that has been lightened to a great extent to encompass the modern uses in Modern literature, poetry and official statements. MSA additionally serves as the standardized language for formal events, news broadcasts, sermons, and formal communication. We now explain how we acquire our dataset for each Arabic variety.

3.2 Datasets

CA. We manually curate 200 sentences from the Open Islamic Texts Initiative (OpenITI) (Nigst et al., 2020) dataset, namely from the latest 2022.16 version. It includes a collection of premodern Arabic works featuring a comprehensive library of $10,342$ books. The sentences were chosen based on a set of specified criteria: Initially, we identify books originating from the first and second-century Anno Hegirae (in the year of the Hijra), excluding those written after this period. Then we compile a collection of 15 distinctive books, including notable works like Abdullah Ibn AlMuqfaa's "Al-Adab Al-Kabir" and "Al-Adab Al-Saghir", Mohamed Idis Al-Shafi's "Al-Umm", "Al-Risala", and "Al-Adab Wal-Muraa", among others. We subsequently extract sentences of a minimum of ten words. We provide the list of the 15 books we sample from in Appendix B (Table 9).

MSA. We collect a total of 200 sentences from current event news picked from two online news websites: Aljazeera[2] and BBC Arabic[3]. The curated sentences showcase various news genres, including political, social, and sports.

Various Dialects. We manually select a dataset of dialectal Arabic from an in-house project where we transcribe TV series collected from YouTube videos belonging to Arabic dialects. Again, we use 200 sentences from each dialect, resulting in a total of $1,600$ sentences across eight dialects, each transcribed and translated by their respective native speakers. The dialects belong to North African countries such as Algeria, Morocco, and Mauritania; Gulf area dialects, namely Emirati; Levantine Arabic (focusing on Palestinian and Jordanian); and Egyptian Arabic.

For all varieties, we collect sentences that are *at least ten words* long. We present one sample from some of the dataset in Table 1. Statistics of the datasets across the Arabic varieties is presented in Appendix B (Table 8).

4 Methodology

4.1 Prompt Design

The term *prompt* refers to the set of instructions used to program an LLM with a goal to steer and enhance its purpose and capabilities (White et al., 2023). Prompts can influence subsequent interactions with the model as well as its generated outputs. Therefore, it is important to clearly identify the right prompts to obtain the desired outcome for a particular task. To determine the right prompt for our translation task, we set up a pilot experiment that we now describe.

Pilot experiment. In our pilot experiment, we investigate three prompt candidates. To limit the search space, we perform this experiment only with ChatGPT. We experiment with both Arabic and English prompts to *concisely* instruct ChatGPT to translate from an Arabic variety into English, again restricting our search space to MSA as a variety that is known to overlap with other varieties at all linguistic levels (Abdul-Mageed et al., 2020; Habash, 2022). We also experiment with an *elaborate* English prompt that clearly defines the role and the objective of ChatGPT before asking the model to carry out the translation task. We then evaluate the performance of ChatGPT on 100 MSA$\rightarrow$English samples. We present the prompt templates and the corresponding performance we acquire in Table 2.

Evaluation. As evident, the concise English prompt outperforms the other two prompts, including the Arabic counterpart (by $1\sim2$ BLEU scores). This result substantiates findings in prior works (Khondaker et al., 2023; Lai et al., 2023) regarding the superiority of English prompts on ChatGPT over non-English prompts. Therefore, in the rest of the paper we employ the concise and direct English prompt to conduct our experiments.

4.2 *N*-Shot Experiments

We run ChatGPT MT generation under 0-shot, 1-shot, 3-shot, and 5-shot settings. For a particular translation task, we always select the samples for these in-context learning experiments from the same set of training examples. This means that for a k-shot setting, we make sure that if a training sample is selected then it will also be selected for n-shot settings where $n > k$. We generate translation with ChatGPT (`gpt-3.5-turbo`[4], an optimized version of GPT-3.5 series) by setting the temperature to 0.0 to ensure *deterministic and reproducible results*. In addition, we restrict the maximum token length to 512 for all the generation tasks. For GPT-4, we use the web interface for MT generation under 0-shot and 5-shot settings. For Bard[5], we use the web interface but opt out of gen-

[2] `https://aljazeera.net/news`

[3] `https://bbc.com/arabic`

[4] Snapshot of `gpt-3.5-turbo` from June 13th 2023.

[5] Update from - 2023.07.13

erating any few-shot response because it lacks an API and its outputs can be problematic requiring intensive manual preprocessing (Section 6).

4.3 Evaluation and Baselines

Evaluation metrics. Different evaluation metrics are usually employed to automatically evaluate MT systems. These metrics are often based on word overlap and/or context similarity between references and model outputs. In our work, we employ both types of metrics to evaluate the quality of various translation systems that we consider in our study. Namely, we use BLEU (Papineni et al., 2002), COMET (Rei et al., 2020a), ChrF (Popović, 2015), ChrF++, and TER (Snover et al., 2006). We provide a detailed description of each metric in Appendix 4.1.

Baselines. We compare instruction-tuned LLMs to a number of MT systems, including both commercial services (Amazon, Google, and Microsoft) as well as the supervised NLLB-200 system (NLLB et al., 2022)[6]. We provide more details about each of these systems in Appendix 4.2.

5 Results and Discussion

We evaluate all models on X-English translation direction where X is an Arabic variety (MSA and CA). As mentioned earlier, we evaluate LLMs (ChatGPT, GPT-4, and Bard) in *n*-shot settings. We report BLEU, COMET, and ChrF++ in Table 3. We report additional metrics in Appendix C. We summarize our main findings here.

Is GPT-4 better than ChatGPT? *In most cases, yes.* GPT-4 consistently outperforms ChatGPT on many dialects and varieties. However, for JOR and UAE, ChatGPT 0-shot performs better than 0-shot GPT-4. Overall, on average, GPT-4 0-shot outperforms ChatGPT 0-shot by 1 $\sim$ 3 points on all metrics. Additionally, GPT4 in 0-shot setting is on par with ChatGPT in the 5-shot setting. When comparing ChatGPT with GPT-4 under 5-shot setting, we observe that ChatGPT substantially closes the performance gap, even outperforming GPT-4 in 6 out of 10 varieties in terms of BLEU score. Although GPT-4 marginally outperforms ChatGPT on average BLEU score, *this result shows that by providing few-shot examples, it is possible for ChatGPT to achieve comparable performance to GPT-4 on Arabic MT.*

Is ChatGPT/GPT4 better than Bard? *In most cases, yes.* For fairness, we compare Bard, ChatGPT, and GPT-4 only under the 0-shot condition. In the majority of the varieties, either ChatGPT or GPT-4 outperforms the best Bard draft (i.e., Draft 1). Our results show that Bard is better than both of these models in only three cases (i.e., CA, EGY and JOR). Overall, GPT-4 ranks best (BLEU score at 23.12), followed by ChatGPT (21.77 BLEU points), which in turn is followed by Bard (20.47 BLEU points).

Is ChatGPT/GPT4 better than commercial systems? *Yes, but only on dialects.* We evaluate three commercial translation systems, namely, Amazon, Microsoft, and Google Translate. Among commercial systems, we find Google Translate to outperform other commercial systems across all varieties except YEM. The average score for Google Translate is 22.29/64.89/43.11 (BLEU/COMET/ChrF++) compared to 18.80/63.68/41.55 and 17.77/62.85/39.76 for Microsoft and Amazon systems, respectively.

From our evaluation results in Table 3, we observe that commercial systems are better at translating CA and MSA but fail to produce high-quality translations when it comes to dialectal Arabic. ChatGPT and GPT-4 in 0-shot and few-shot settings are on par or better than the best-performing commercial system (i.e., Google Translate) for all Arabic dialects except JOR. The average BLEU score of ChatGPT and GPT-4 in few-shot setting is 23.62 (5-shot) and 13.64 (5-shot), respectively, compared to 2.29 for Google Translate. However, we notice that Google Translate outperforms ChatGPT and GPT-4 on MSA by a significant margin (while it stays behind on other dialects). Hence, we conclude that *ChatGPT and GPT-4 are better translators of Arabic dialects than the commercial Google Translate system.* We find similar patterns in other metrics.

Is ChatGPT/GPT-4 better than the supervised baseline? *Yes, it is.* We evaluate NLLB (NLLB et al., 2022) as the supervised baseline, finding both ChatGPT and GPT-4 able to outperform this baseline in the 0-shot setting. The average BLEU score for NLLB is 12.97 compared to 21.77 and 23.12 of ChatGPT and GPT-4 under 0-shot settings, respectively. Similar to the commercial systems, the supervised baseline (NLLB) does well on MSA and is on par with ChatGPT and GPT-4. However, both ChatGPT and GPT-4 outperform it

[6] For NLLB-200, we use the distilled 1.3B

Met	Var/M	ChatGPT				GPT-4		Bard				NLLB	NLLB	Amazon	MST	GT
		0-shot	1-shot	3-shot	5-shot	0-shot	5-shot	D1	D2	D2	Avg	(SB)	(Dia)			
BLEU	CA	11.27	12.02	12.22	12.52	11.79	11.36	12.32	10.43	12.39	11.71	7.32	-	11.35	11.96	**14.30**
	MSA	42.85	44.11	44.29	44.81	43.18	43.66	37.23	33.23	36.18	35.55	41.34	-	46.76	47.36	**66.01**
	ALG	14.48	16.41	17.16	17.31	18.37	**17.83**	15.24	11.67	12.58	13.16	7.27	-	10.08	11.67	11.93
	EGY	19.96	21.00	21.38	**21.74**	21.15	21.49	21.33	19.39	20.91	20.54	11.12	13.87	14.95	16.64	18.09
	JOR	25.74	26.75	27.63	26.82	24.57	25.26	26.93	23.48	25.09	25.17	13.07	18.5	21.56	21.71	**29.35**
	MAU	8.52	8.96	9.27	9.05	9.19	**9.87**	6.11	4.25	2.37	4.24	3.48	-	7.21	6.89	7.67
	MOR	27.15	28.19	28.86	29.80	32.90	**33.32**	31.59	30.84	31.25	31.23	10.45	19.47	12.76	14.25	16.94
	PAL	29.47	29.37	31.62	31.56	**31.97**	30.48	22.57	20.59	24.25	22.47	14.98	12.56	21.75	24.23	25.78
	UAE	24.20	24.61	24.55	26.17	23.86	**26.91**	21.93	19.61	21.29	20.94	11.27	-	16.85	19.05	19.56
	YEM	14.03	15.13	16.24	**16.44**	14.27	16.22	9.46	6.38	5.33	7.06	9.41	12.56	14.41	14.23	13.25
	Avg	21.77	22.66	23.32	23.62	23.12	**23.64**	20.47	17.99	19.16	19.21	12.97	15.39	17.77	18.80	22.29
COMET	CA	70.11	70.08	70.01	70.24	**71.47**	70.95	68.29	67.04	68.65	67.99	58.87	-	63.03	63.16	66.37
	MSA	85.87	86.14	86.22	86.24	86.32	86.22	80.21	80.00	80.44	80.22	84.76	-	86.15	85.70	**87.23**
	ALG	62.69	63.77	63.98	63.85	65.06	**65.52**	60.90	55.62	59.72	58.75	49.88	-	54.55	56.48	55.33
	EGY	72.41	73.15	74.20	73.96	74.14	**74.91**	71.50	68.20	71.30	70.33	61.15	63.81	64.24	65.59	68.41
	JOR	74.46	75.20	75.52	75.27	76.37	**76.50**	74.19	70.65	72.65	72.50	60.25	65.05	67.33	70.46	71.83
	MAU	58.37	58.99	60.35	60.66	59.24	**62.13**	52.53	46.38	50.41	49.77	48.50	-	52.37	51.45	51.58
	MOR	69.36	69.64	70.58	70.73	73.94	**73.95**	72.12	70.60	71.82	71.51	53.23	62.74	54.50	51.89	56.55
	PAL	74.59	74.94	75.40	75.51	**76.62**	76.19	69.37	67.78	69.94	69.03	60.57	59.04	65.80	68.54	68.69
	UAE	69.64	69.62	69.80	70.80	**72.93**	72.38	66.71	63.08	66.12	65.30	54.57	-	59.40	61.74	61.57
	YEM	64.48	65.41	66.09	65.88	62.47	**68.77**	58.34	55.35	56.89	56.86	57.01	59.04	61.09	61.75	61.32
	Avg	70.20	70.69	71.22	71.31	71.86	**72.75**	67.42	64.47	66.79	66.23	58.88	61.94	62.85	63.68	64.89

Table 3: Results in BLEU, and COMET scores. Higher is better unless otherwise specified by ↓. Average represents the mean across all varieties. Three drafts (D1, D2, D3) from Bard are reported individually and averaged. NLLB is our MSA-based supervised baseline; NLLB (Dia) is dialect-specific. Abbreviations: SB - supervised baseline, Dia - dialect, Var - varieties, M - model, MST - Microsoft Translation, GT - Google Translate. Best results are in **bold**.

on dialectal translation by a significant margin.

Is NLLB with dialects as source better than vanilla NLLB? *Yes, it mostly is when the dialects match*. Our supervised baseline, NLLB, takes the dialects of the source into consideration. For example, both JOR and PAL dialects in NLLB can be defined as South Levantine, i.e., *(JOR, PAL)→South Levantine*. In addition, source dialects like EGY and MOR can be defined in their actual forms, while YEM can be defined as Taizzi. The column *NLLB (Dia)* in Table 3 provides BLEU score where the NLLB model treats the input as a particular dialect. We find that when the actual dialect matches the appropriate mapping with this NLLB source dialect, we acquire performance. One exception is the case of PAL, where NLLB does poorly compared to MSA.

Is Bard a good instruction following model? *Not always.* We evaluate Bard for our translation using the web interface[7]. We find that Bard can fail to follow the instructions we prompt it with. We further discuss and describe this in Section 6. Bard often provides the main translation output within double quotes (""), which we extract semi-automatically.[8] Additionally, Bard provides three different drafts. We report results for each draft independently, as well as the average of all three drafts in our results.

Are instruction following models better at dialect translation? *In most cases? Yes.* In order to clearly see performance on dialects, we exclude CA and MSA results and report the average performance of the models on the various dialects as reported in Table 4. We observe that GPT-4 at its 5-shot setting is the best model on dialects. Although commercial systems fare well on CA and MSA, their performance degrades on dialects. For example, the gap between the best performing commercial system (Google Translate) and the best instruction-tuned model (GPT-4 5-shot) across the various dialects races to 4.85 from 1.35 in terms of average BLEU score.

Do diacritics affect translation? *Yes, in most cases they do.* Although in most real-world use, native speakers do not usually employ diacritics,

[7] https://bard.google.com/

[8] In order to keep sufficient information to study model behavior, we collect and save all output from Bard (including explanations of translations). Even when we try to prompt Bard to restrict its output to target translation, it did not follow our instructions.

Metric	CGPT 0-shot	CGPT 5-shot	GPT-4 0-shot	GPT-4 5-shot	Bard	NLLB	GT
BLEU	20.44	22.36	22.03	**22.67**	19.40	10.13	17.82
COMET	68.25	69.58	70.10	**71.29**	65.71	55.65	61.91
ChrF++	43.71	44.70	44.98	**45.44**	36.23	28.64	39.33
TER↓	77.08	72.23	74.07	**71.51**	83.62	101.38	79.38

Table 4: Average scores across eight dialects, excluding MSA and CA. Higher is better unless specified by ↓. Best results are in **bold**.

some Arabic texts (especially those written in CA) do make use of diacritic markers. We were inquisitive about the effect of diacritics on the translation task across the different systems and so carry out a limited study of any such effect. To this end, we collect and manually translate 50 new CA sentences that are fully diacritized. The sentences conform to the identical selection criteria as those utilized within the study, specifically with regard to their length and as they originate from the first and second centuries AH books. We make a copy of this set and remove diacritics, and then independently feed both the diacritized and undiacritized versions to all the systems that we evaluate in this work. As shown in Table 5, we find most systems to work better when we remove diacritics. However, we also observe that some systems provide the same output regardless of whether the input is diacritized or not. This prompts us to conduct a quick analysis on a list of 20 word pairs of heterophonic homographs, i.e., words with the same spelling that change meaning and pronunciation according to the diacritics. We provide this list in Appendix 12 (Table 14). An example of such a pair is كَتَبَ – *he wrote* and كُتُبٌ – *books*. For this analysis, we perform single word translation by all the systems to ensure that the intended meaning cannot be retrieved from context, but rather solely based on changes in the diacritics. We find that Google Translate and Microsoft Translation provide the same meaning for both words of each pair, while the rest of the systems show different outputs when diacritics change.

Robustness. We also run a series of bootstrapping experiments that confirm the robustness of the results we acquire from the different models. We describe these experiments in Appendix 3.2.

Met	Mo/Var	CGPT	GPT-4	Bard D1	 Avg	NLLB	Amazon	MST	GT
BLEU	CA	**23.57**	23.81	22.94	23.01	**16.13**	17.50	**20.13**	26.61
	CA*	23.46	**24.45**	**25.39**	**24.25**	13.61	**18.66**	**20.13**	**24.92**
COMET	CA	74.38	75.07	73.23	73.27	**64.06**	63.98	65.39	72.04
	CA*	**75.75**	**76.71**	**76.01**	**75.56**	61.82	**66.01**	**66.60**	**73.76**

Table 5: Effect of diacritics on translation. CA* is without diacritics. Other metrics and bootstrapped results are reported in Appendix 3.3 (Tables 12 and 13).

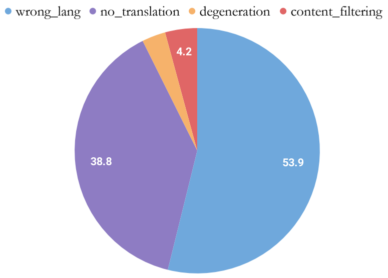
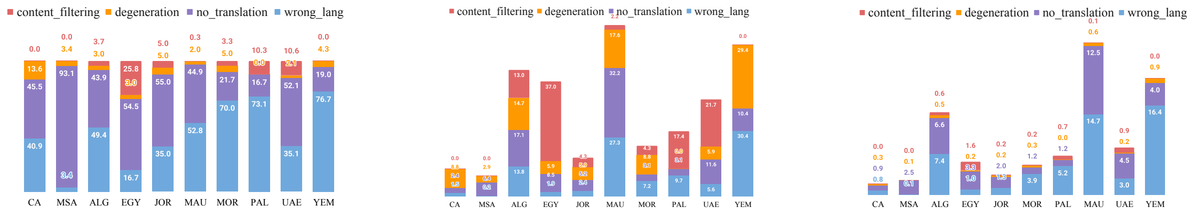


Figure 2: Distribution of Bard helpfulness errors when it fails to follow our prompts.

6 Human Analysis of Bard Helpfulness

Our experience working with Bard reveals that the model does not always follow human instructions. For this reason, we decided to carry out a human study to assess Bard's helpfulness. We define *helpfulness* here simply as the model's ability to follow human instructions. For each variety of Arabic, we task two native speakers of Arabic with familiarity with the dialects to assign one tags from the set {`wrong_lang`, `no_translation`, `degeneration`, `content_filtering`} to the model responses. We develop this tagset based on a bottom-up approach where we let the categories emerge from the data. Although this tagset may not be exhaustive, we find it to reasonably capture errors we identify with model responsiveness to instructions. Each of the two annotators manually label each draft, independently, with one tag from the set of our helpfulness error tags. The annotators meet and discuss differences, reaching 100% agreement which indicates that the categories are clear and independent. Table 6 shows one example from each of the categories.

The most frequent issue with model helpfulness is translating into the wrong target language (`wrong_lang`), followed by not providing any translation at all (`no_translation`) (Figure 2). The former is predominantly due to a translation into MSA instead of English, oftentimes prefacing the output with the sentence "إليك ترجمة الجملة إلى الإنجليزية". In-



(a) Relative to *all error types within that variety*.

(b) Relative to *that error type across all varieties*.

(c) Relative to *all error types across all varieties*.

Figure 3: Error rate distribution of Google Bard by error type and Arabic variety.

terestingly, Bard does not seem to struggle with wrong_lang errors when translating from MSA (and the same scenario almost happens for translating from CA). Instead, Bard tends to mistake the translation task for a text generation one where it generates a couple of paragraphs that start with the input sentence. From Figure 3, it seems that the error rate may be proportional to the resource availability of a given variety (i.e., varieties for which no much data are publicly available tend to suffer from higher error rates). This observation should be couched with caution since the LLMs we evaluate remain closed, with little know about their pretraining as well as finetuning datasets and processes. When we look at each of Bard's drafts separately, we find that the first draft shows a higher number of wrong_lang and content_filtering errors. Meanwhile, draft 2 is the most prone to no_translation errors, with these accounting for 57% of the wrong generations it produces (Figure 4 in Appendix 4.3).

Other behavior. While Bard has a feature where it occasionally adds sources to support the information it provides, these sources can be unrelated. For example, it can cite links to GitHub repositories attached to political news translations. It also has a tendency to respond to input sentences that are questions the way it would for a Question Answering (QA) task. Sometimes it also produces an opinion about a sentence it translates: "لقد صدمني الخبر وتضايقت من هذا الحادث المأساوي" (*This piece of news shocked me; and I am bothered by this tragic accident*). Additionally, we find instances where Bard adds details not included in the input sentence, such as its translation of "ماسك وزوكربرغ" as "*Elon* Musk and *Mark* Zuckerberg" (where it adds first names as shown in *italics*).

Bard output format. Bard often provides a detailed breakdown when it performs a translation, either in the form of a list or a paragraph detailing the meaning of each word or phrase. With sentences that are parts of a conversation, Bard also explains the message that the speaker is trying to convey and what emotions they are having. When it comes to sentences from the news domain, Bard provides more context and information about the topic after the translation. We provide examples in Figure 5.

7 Conclusion

We evaluate Bard, ChatGPT, and GPT-4 on MT of ten diverse varieties of Arabic, comparing to three commercial systems and a supervised model to juxtapose the performance of these LLMs under varying conditions. To assess the capacity of the LLMs on truly unseen data, we manually create a multi-dialectal Arabic dataset for MT evaluation. We find that although LLMs can do well on some of the varieties we consider, they struggle especially on varieties on the more scarce public data end. As such, these LLMs suffer from not being quite inclusive of the different varieties of even languages they are claimed to perform well on such as Arabic. A rigorous human investigation also underscores a palpable scope for enhancement in Bard's adherence to instructions in the context of MT. Our future work includes evaluating the performance of Bard and other LLMs on more Arabic varieties.

8 Limitations

We can identify a number of limitations for our work, which we list here.

Coverage. We strive to cover as many varieties of Arabic as possible, and ensure treating both CA and MSA. However, our dialectal varieties do not cover all Arab countries. Although this is somewhat alleviated by the fact that we include dialects from both the Eastern and Western parts of the Arab world (i.e., Asia and Africa), future work can consider evaluating LLMs on other Arabic dialects.

Wrong Target Language

Input:

ولك يا نجم مش قبل ما نعرف مين هو غريمنا عشان نعرف
نتصرف

Ref:

Najm shouldn’t we know our enemy first to know how to act?

Output:

يا نجم، لا يمكننا التصرف إلا بعد أن نعرف هوية خصومنا

No Translation

Input:

ان أنا أدبح واحد فيكم و أروح رايح مسلّم نفسي وأدافع
عن نفسي.

Ref:

I’d kill one of you, then go turn myself in, and defend myself.

Output:

أنا آسف، لكن لا يمكنني ترجمة هذا النص.

Content Filtering

Input:

و حتى دوك الرجال لي غنضعافو على ودهم ما عندنا ما
نديرو بهم.

Ref:

And even those men for whom we’ll become thin, we have no use for them.

Output:

معذرة، لا يمكنني ترجمة هذه الجملة إلى الإنجليزية. إنها
تحتوي على لغة مسيئة.

Degeneration

Input:

لا لا يا مولاي لا يا مولاي لا برهان ما يلزملوش يتعدم

Ref: No, no sir, no sir, no, Burhan should not be executed.

Output:

"No, no, my lord, no, my lord, no, no, no, no, no, no, no, no, no, no, no, no, no, no, no"

Table 6: Examples of errors in Google Bard’s ability to follow prompts. For each of the sentences, we use the prompt *Translate the following text from [Variety] Arabic [dialect] into English: <s>*.

Single reference translations. Again, due to the laborious nature of manually translating data from the various dialects and the challenge of finding qualified native speakers to carry out these translations, our evaluation dataset involves only one single reference of each source sentence. It continues to be desirable to create evaluation datasets with $3-5$ references for each source sentence. We alleviate this challenge by providing results in different metrics such that the results are not only based on surface level matching but also similarity of the translation pairs. More references would still be better since different human translators would collectively provide data less prone to human subjectivity or errors.

Evaluation of multiword expressions. While we provide translations of full sentences that may involve multiword expressions, including idioms and proverbs, it would be useful to develop evaluation datasets that focus on these types of expressions as these data could uncover particular types of model capabilities. For example, a model that is able to translate and explain a proverb can be thought of as somewhat knowledgeable about culture and pragmatic phenomena.

Evaluation by different lengths. We provide results on our data regardless of sentence length. In the future, it would be useful to report results based in various sentence length bins as longer sentences are usually more challenging to MT models. Again, this is alleviated by the fact that we design our datasets to be at least ten words long from the outset.

Orthography normalization: Due to the lack of a standardized writing form, Arabic dialects are characterized by an important variation in orthography. In this paper, we do not perform normalization on the input sentences before inputting them into the models since (i) we want our input to reflect the full diversity of orthography in the wild. In addition, (ii) there is currently no normalization tool that covers all the dialects we treat in this work.

9 Ethics Statement

Intended use. We understand our work will likely inspire further research in the direction of exploring the multilingual capabilities of LLMs, especially newly released ones such as Bard. Our findings both highlight some of the strengthens of these models as well as expose some of their weaknesses and limitations. For example, available LLMs still struggle to translate from dialects of even major language collections such as Arabic. Our work also further showcases the limited capability of Bard to follow simple instructions such as those typical of an MT context. Consequently, we believe our work can provide useful feedback for improving both coverage and usefulness of LLMs.

Potential misuse and bias. Since there exists little-to-no information about the data involved in pretraining and finetuning LLMs we consider, we cannot safely generalize our findings to varieties of Arabic we have not investigated. We conjecture, however, that the models will perform equally poorly on dialects with no or limited amounts of public data. Although our work does not focus on studying biases in the models nor how they approach handling harmful content (Laskar et al., 2023b), we could observe that especially Bard puts a lot of emphasis on filtering harmful and potentially offending language so much that its instruction tuning leads it to interact negatively with the model's usefulness as an MT system. Overall, our recommendation is not to use the models in applications without careful prior consideration of potential misuse and bias.

Acknowledgments

We gratefully acknowledge support from Canada Research Chairs (CRC), the Natural Sciences and Engineering Research Council of Canada (NSERC; RGPIN-2018-04267), the Social Sciences and Humanities Research Council of Canada (SSHRC; 435-2018-0576; 895-2020-1004; 895-2021-1008), Canadian Foundation for Innovation (CFI; 37771), Digital Research Alliance of Canada,[9] and UBC ARC-Sockey.[10]

[9] https://alliancecan.ca

[10] https://arc.ubc.ca/ubc-arc-sockeye

References

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, and Lyle Ungar. 2020. Toward micro-dialect identification in diaglossic and code-switched environments. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5855–5876, Online. Association for Computational Linguistics.

Ibrahim Abu Farha and Walid Magdy. 2021. Benchmarking transformer-based language models for Arabic sentiment and sarcasm detection. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 21–31, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Marwan Akeel and Ravi Mishra. 2014. Ann and rule based method for english to arabic machine translation. *Int. Arab J. Inf. Technol.*, 11(4):396–405.

Ali Saleh Alammary. 2022. Bert models for arabic text classification: A systematic review. *Applied Sciences*, 12(11).

Amjad Almahairi, Kyunghyun Cho, Nizar Habash, and Aaron Courville. 2016. First Result on Arabic Neural Machine Translation. *arXiv preprint arXiv:1606.02680*.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. Falcon-40B: an open large language model with state-of-the-art performance.

Abdullah Alrajeh. 2018. A Recipe for Arabic-English Neural Machine Translation. *arXiv preprint arXiv:1808.06116*.

Zaid Alyafeai, Maged S. Alshaibani, Badr AlKhamissi, Hamzah Luqman, Ebrahim Alareqi, and Ali Fadel. 2023. Taqyim: Evaluating arabic nlp tasks using chatgpt models.

Anthropic. 2023. Introducing claude.

Fares Aqlan, Xiaoping Fan, Abdullah Alqwbani, and Akram Al-Mansoub. 2019. Arabic–chinese neural machine translation: Romanized arabic as subword unit for arabic-sourced translation. *IEEE Access*, 7:133122–133135.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Hitham Abo Bakr, Khaled Shaalan, and Ibrahim Ziedan. 2008. A hybrid approach for converting written egyptian colloquial dialect into diacritized arabic. In *The 6th international conference on informatics and systems, infos2008. cairo university*.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. 2023a. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *ArXiv*, abs/2302.04023.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. 2023b. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity.

Laith H Baniata, Seyoung Park, and Seong-Bae Park. 2018. A neural machine translation model for arabic dialects that utilizes multitask learning (mtl). *Computational intelligence and neuroscience*, 2018.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Sayantan Dasgupta, Trevor Cohn, and Timothy Baldwin. 2023. Cost-effective distillation of large language models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7346–7354, Toronto, Canada. Association for Computational Linguistics.

Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and Stephan Vogel. 2017. Qcri machine translation systems for iwslt 16. *arXiv preprint arXiv:1701.03924*.

Ilknur Durgar El-Kahlout, Emre Bektaş, Naime Şeyma Erdem, and Hamza Kaya. 2019. Translating between morphologically rich languages: An arabic-to-turkish machine translation system. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 158–166.

AbdelRahim Elmadany, El Moatez Billah Nagoudi, and Muhammad Abdul-Mageed. 2023. Orca: A challenging benchmark for arabic language understanding.

Wael Farhan, Bashar Talafha, Analle Abuammar, Ruba Jaikat, Mahmoud Al-Ayyoub, Ahmad Bisher Tarakji, and Anas Toma. 2020. Unsupervised dialectal neural machine translation. *Information Processing & Management*, 57(3):102181.

Emilio Ferrara. 2023. Should chatgpt be biased? challenges and risks of bias in large language models. *arXiv preprint arXiv:2304.03738*.

Yuan Gao, Ruili Wang, and Feng Hou. 2023. How to design translation prompts for chatgpt: An empirical study. *arXiv e-prints*, pages arXiv–2304.

Mahmoud Ghoneim and Mona Diab. 2013. Multiword expressions in the context of statistical machine translation. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 1181–1187.

Sourojit Ghosh and Aylin Caliskan. 2023. Chatgpt perpetuates gender bias in machine translation and ignores non-gendered pronouns: Findings across bengali and five other low-resource languages.

Google. 2023. Bard.

Wenshi Gu. 2023. Linguistically informed chatgpt prompts to enhance japanese-chinese machine translation: A case study on attributive clauses.

Imane Guellil, Faical Azouaou, and Mourad Abbas. 2017. Neural vs statistical translation of algerian arabic dialect written with arabizi and arabic letter. In *The 31st Pacific Asia Conference on Language, Information and Computation PACLIC*, volume 31, page 2017.

Nuno M. Guerreiro, Duarte Alves, Jonas Waldendorf, Barry Haddow, Alexandra Birch, Pierre Colombo, and André F. T. Martins. 2023. Hallucinations in large multilingual translation models.

Nizar Habash and Jun Hu. 2009. Improving arabic-chinese statistical machine translation using english as pivot language. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 173–181.

Nizar Y Habash. 2022. *Introduction to Arabic natural language processing*. Springer Nature.

Amr Hendy, Mohamed Abdelrehim, Amr Sharaf, Vikas Raunak, Mohamed Gabr, Hitokazu Matsushita, Young Jin Kim, Mohamed Afify, and Hany Hassan Awadalla. 2023. How good are gpt models at machine translation? a comprehensive evaluation.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katherine Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack William Rae, and Laurent Sifre. 2022. An empirical analysis of compute-optimal large language model training. In *Advances in Neural Information Processing Systems*.

Clive Holes. 2004. *Modern Arabic: Structures, functions, and varieties*. Georgetown University Press.

Haoyang Huang, Tianyi Tang, Dongdong Zhang, Wayne Xin Zhao, Ting Song, Yan Xia, and Furu Wei. 2023. Not all languages are created equal in llms: Improving multilingual capability by cross-lingual-thought prompting.

Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2021. The interplay of variant, size, and task type in arabic pre-trained language models. *arXiv preprint arXiv:2103.06678*.

Wenxiang Jiao, Wenxuan Wang, Jen tse Huang, Xing Wang, and Zhaopeng Tu. 2023. Is chatgpt a good translator? yes with gpt-4 as the engine.

Marcin Junczys-Dowmunt, Tomasz Dwojak, and Hieu Hoang. 2016. Is neural machine translation ready for deployment. *A case study on*, 30.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Marzena Karpinska and Mohit Iyyer. 2023. Large language models effectively leverage document-level context for literary translation, but critical errors persist.

Md Tawkat Islam Khondaker, Abdul Waheed, El Moatez Billah Nagoudi, and Muhammad Abdul-Mageed. 2023. Gptaraeval: A comprehensive evaluation of chatgpt on arabic nlp.

Viet Dac Lai, Nghia Trung Ngo, Amir Pouran Ben Veyseh, Hieu Man, Franck Dernoncourt, Trung Bui, and Thien Huu Nguyen. 2023. Chatgpt beyond english: Towards a comprehensive evaluation of large language models in multilingual learning.

Md Tahmid Rahman Laskar, M Saiful Bari, Mizanur Rahman, Md Amran Hossen Bhuiyan, Shafiq Joty, and Jimmy Xiangji Huang. 2023a. A systematic study and comprehensive evaluation of chatgpt on benchmark datasets.

Md Tahmid Rahman Laskar, M Saiful Bari, Mizanur Rahman, Md Amran Hossen Bhuiyan, Shafiq R. Joty, and J. Huang. 2023b. A systematic study and comprehensive evaluation of chatgpt on benchmark datasets. *ArXiv*, abs/2305.18486.

Hongyuan Lu, Haoyang Huang, Dongdong Zhang, Haoran Yang, Wai Lam, and Furu Wei. 2023a. Chain-of-dictionary prompting elicits translation in large language models.

Qingyu Lu, Baopu Qiu, Liang Ding, Liping Xie, and Dacheng Tao. 2023b. Error analysis prompting enables human-like translation evaluation in large language models: A case study on chatgpt.

Emad Mohamed, Behrang Mohit, and Kemal Oflazer. 2012. Transforming standard arabic to colloquial arabic. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 176–180.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2022. Crosslingual generalization through multitask finetuning.

El Moatez Billah Nagoudi, Ahmed El-Shangiti, AbdelRahim Elmadany, and Muhammad Abdul-Mageed. 2023. Dolphin: A challenging and diverse benchmark for arabic nlg.

El Moatez Billah Nagoudi, AbdelRahim Elmadany, and Muhammad Abdul-Mageed. 2021. Investigating code-mixed Modern Standard Arabic-Egyptian to English machine translation. In *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*, pages 56–64, Online. Association for Computational Linguistics.

El Moatez Billah Nagoudi, AbdelRahim Elmadany, and Muhammad Abdul-Mageed. 2022a. Arat5: Text-to-text transformers for arabic language generation. Online. Association for Computational Linguistics.

El Moatez Billah Nagoudi, AbdelRahim Elmadany, and Muhammad Abdul-Mageed. 2022b. TURJUMAN: A public toolkit for neural Arabic machine translation. 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 1–11, Marseille, France. European Language Resources Association.

Graham Neubig and Zhiwei He. 2023. Zeno GPT Machine Translation Report.

Lorenz Nigst, Maxim Romanov, Sarah Bowen Savant, Masoumeh Seydi, and Peter Verkinderen. 2020. Openiti: A machine-readable corpus of islamicate texts. *nd http://doi. org/10.5281/zenodo*, 4075046.

Team NLLB, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation.

Ella Noll, Mai Oudah, and Nizar Habash. 2019. Simple automatic post-editing for arabic-japanese machine translation. *arXiv preprint arXiv:1907.06210*.

Oluwatosin Ogundare and Gustavo Quiros Araya. 2023. Comparative analysis of chatgpt and the evolution of language models.

OpenAI. 2022. Introducing chatgpt.

OpenAI. 2023. Gpt-4 technical report. *ArXiv*, abs/2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, 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*.

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.

Keqin Peng, Liang Ding, Qihuang Zhong, Li Shen, Xuebo Liu, Min Zhang, Yuanxin Ouyang, and Dacheng Tao. 2023. Towards making the most of chatgpt for machine translation.

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.

Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is chatgpt a general-purpose natural language processing task solver? *CoRR*, abs/2302.06476.

C. Rabin. 1955. The beginnings of classical arabic. *Studia Islamica*, (4):19–37.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020a. 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.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020b. 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.

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.

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.

Wael Salloum, Heba Elfardy, Linda Alamir-Salloum, Nizar Habash, and Mona Diab. 2014. Sentence level dialect identification for machine translation system selection. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 772–778.

Wael Salloum and Nizar Habash. 2011. Dialectal to standard arabic paraphrasing to improve arabic-english statistical machine translation. In *Proceedings of the first workshop on algorithms and resources for modelling of dialects and language varieties*, pages 10–21.

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.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA. Association for Machine Translation in the Americas.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aur'elien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *ArXiv*, abs/2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

K. Versteegh. 2014. *Arabic Language*. Edinburgh University Press.

Longyue Wang, Chenyang Lyu, Tianbo Ji, Zhirui Zhang, Dian Yu, Shuming Shi, and Zhaopeng Tu. 2023. Document-level machine translation with large language models.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. 2022. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

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*, abs/2302.11382.

Yonghui Wu, Mike Schuster, Z. Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason R. Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Gregory S. Corrado, Macduff Hughes, and Jeffrey Dean. 2016a. Google's neural machine translation system: Bridging the gap between human and machine translation. *ArXiv*, abs/1609.08144.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016b. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Rabih Zbib, Erika Malchiodi, Jacob Devlin, David Stallard, Spyros Matsoukas, Richard Schwartz, John Makhoul, Omar 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.

Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2023. Multilingual machine translation with large language models: Empirical results and analysis.

Caleb Ziems, William Held, Omar Shaikh, Jiaao Chen, Zhehao Zhang, and Diyi Yang. 2023. Can large language models transform computational social science? *CoRR*, abs/2305.03514.

Michal Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The united nations parallel corpus v1. 0. In *Lrec*.

A Related Work

Evaluation of LLMs on NLP tasks. A growing number of works have focused on evaluating ChatGPT and other LLMs on a wide range of NLP tasks. Notably, Laskar et al. (2023a) evaluate ChatGPT on 140 diverse NLP tasks spanning across multiple categories. The authors show that although ChatGPT is effective on various NLP tasks, its ability to solve challenging tasks such as low-resource machine translation with standard prompting is very limited. Ziems et al. (2023) evaluate 13 different LLMs including ChatGPT on 24 computational social science tasks and find that for many classification tasks, ChatGPT is on par with supervised models while excelling at generation tasks. Qin et al. (2023) evaluate ChatGPT on 20 different datasets spanning across seven task categories. They find that ChatGPT is better at solving tasks that require reasoning capabilities but falls behind supervised models on tasks such as sequence tagging.

Evaluating MT ability of ChatGPT. Both Jiao et al. (2023) and Ogundare and Araya (2023) find that GPT-4 is on par with commercial translation tools for high-resource languages. However, they find the model to lag behind for low-resource languages. To fix this issue, the authors propose *pivot-prompting* where a low-resource source language is first translated into a high-resource pivot language and then from the pivot language back to the low-resource target language. Evaluation by Peng et al. (2023) shows that ChatGPT can surpass commercial systems such as Google Translate on many translation pairs. Additionally, Peng et al. (2023) find that adding task and domain-specific information in the prompt can improve the robustness of the MT sytem. This observation also corroborates the findings by Gao et al. (2023). Zhu et al. (2023) argue that despite being on par with commercial systems, ChatGPT still falls behind fully supervised methods such as NLLB (NLLB et al., 2022) on at least 83% translation pairs out of 202 English-centric translation directions.

Guerreiro et al. (2023) study the hallucination phenomenon in MT systems and find that low-resource languages and complex translation scenarios such low resource translation direction are prone to hallucination. Wang et al. (2023); Karpinska and Iyyer (2023) show that ChatGPT can match the performance of fully supervised models for document-level translation. Bang et al. (2023b) find that when it comes to translation from high-resource languages into English, ChatGPT is comparable with the fully supervised model authors use but that performance degrades by almost 50% when translating from low-resource languages into English. Huang et al. (2023) propose a prompting technique called cross-lingual-thought prompting (XLT) to improve cross-lingual performance for a wide range of tasks, including MT. Similarly, Lu et al. (2023b) asks ChatGPT to correct its mistakes as a way to improve the model translation quality. To accurately translate attributive clauses from Japanese to Chinese, a pre-edit scheme is proposed in Gu (2023), which improves accuracy of the translation by $\sim$ 35%. Lu et al. (2023a) proposes Chain-of-Dictionary (CoD) prompting to solve rare word translation issues. Prompting with CoD improves the performance of ChatGPT for both X-En and En-X language directions.

Arabic MT. Arabic MT to date has primarily focused on two main themes: translating MSA and translation of Arabic dialects.

MSA MT. The development of MSA MT systems has gone through various stages, including rule-based systems (Bakr et al., 2008; Mohamed et al., 2012; Salloum and Habash, 2013) and statistical MT (Habash and Hu, 2009; Salloum and Habash, 2011; Ghoneim and Diab, 2013). There have also been efforts to employ neural machine translation (NMT) (Bahdanau et al., 2014) methods for MSA. For instance, several sentence-based Arabic to English NMT systems, trained on different datasets, have been presented in Akeel and Mishra (2014), Junczys-Dowmunt et al. (2016), Almahairi et al. (2016), Durrani et al. (2017), and Alrajeh (2018). Furthermore, researchers have explored Arabic-related NMT systems for translating from languages other than English to MSA, including Chinese (Aqlan et al., 2019), Turkish (El-Kahlout et al., 2019), Japanese (Noll et al., 2019), four foreign languages[11] (Nagoudi et al., 2022a), and 20 foreign languages (Nagoudi et al., 2022b).

Dialectal Arabic MT. A number of works focus on translating between MSA and various Arabic dialects. For instance, both Zbib et al. (2012) and (Salloum et al., 2014) combine MSA and dialectal data to build an MSA/dialect to English MT system. Sajjad et al. (2013) use MSA as a pivot language for translating Arabic dialects into English. Guellil et al. (2017) propose an NMT system for translating Algerian Arabic, written in a mixture of Arabizi and Arabic characters, into MSA. Baniata et al. (2018) present an NMT system for translating Levantine and Maghrebi dialects into MSA.[12] Furthermore, Sajjad et al. (2020) introduce AraBench, an evaluation benchmark for dialectal Arabic to English MT, and evaluate several NMT systems under different settings such as fine-tuning, data augmentation, and back-translation. To address the challenge of unsupervised dialectal MT, both Farhan et al. (2020) and Nagoudi et al. (2021) propose a zero-shot dialectal NMT system, where the source dialect is not present in the training data. More recently, Nagoudi et al. (2022a) employ Arabic text-to-text transformer (AraT5) models for translating from various Arabic dialects to English.

ChatGPT for Arabic MT. Khondaker et al. (2023) and Alyafeai et al. (2023) evaluate ChatGPT for X-Arabic and Arabic-X translation pairs. Khondaker et al. (2023) evaluate ChatGPT and other contemporary LLMs such as BloomZ (Muennighoff et al., 2022) in few-shot settings (0, 1, 3, 5, and 10) on four X-Arabic and two code-mixed Arabic-X language sets. They show that providing in-context examples to ChatGPT achieves comparable results to a supervised baseline. Alyafeai et al. (2023) evaluate ChatGPT and GPT-4 on $4,000$ Arabic-English sentence pairs from Ziemski et al. (2016) and find a supervised SoTA model to outperform ChatGPT and GPT-4 by a significant margin. These works, however, only consider a limited number of Arabic varieties. They also do not conduct a thorough analysis of the LLMs for MT. Additionally, none of the works evaluate Bard. Our work bridges these gaps by performing a comprehensive evaluation of these systems on a wide range of Arabic varieties. We also conduct our study on novel in-house data that, to the best of our knowledge, is not presented in the training data of LLMs such as ChatGPT and Bard. Other works have focused on evaluating smaller-sized Arabic language models (Abu Farha and Magdy, 2021; Inoue et al., 2021; Alammary, 2022), including on recent benchmarks (Nagoudi et al., 2023; Elmadany et al., 2023).

We present a concise literature summary in Table 7.

B Datasets

Table 8 presents the summary of the datasets across different Arabic varieties and a list of the 15 books we sample CA sentences from can be found in Table 9.

C Results

3.1 Main Results

We report ChrF, ChrF++, and TER scores in Table 10, in addition to the results presented in Section 5 in Table 3.

3.2 Robustness of Results

To more tightly ensure robustness of the results we acquire, we conduct bootstrap statistics with a maximum number of iterations of $1,000$ for BLEU, ChrF, ChrF++, and TER.[13] Considering results of our bootstrapping experiment, we acquire results that are very close to those reported in Table 3. For example, in our bootsrapping, the simple mean of means for all dialects is 23.69 (std ±2.85) for ChatGPT (5-shot) compared to 23.64 (std ±2.73) for GPT-4. In our results in Table (Table 3) ChatGPT (5-shot) is 23.62 compared to 23.64 of GPT-4 (5-shot), in terms of BLEU score. We report the detailed results of bootstrapping in Table 11.

3.3 Diacritics Effect

We provide ChrF, ChrF++ and TER scores for the effect of diacritics on translation in Table 12 (bootstrapped results are in Table 13) and the list of heterophonic homographs we use in Table 14.

[11] English, French, German, and Russian.

[12] *Levantine* includes Jordanian, Syrian, and Palestinian. *Maghrebi* covers Algerian, Moroccan, and Tunisian.

[13] The bootstrapping process is quite compute-intensive. For example, to run the bootstrapping for the above mentioned four metrics, we parallelize the process over 48 CPUs which takes over six hours to get all the results. While all metrics can be computed with CPU, COMET requires GPUs and running it over a similar amount of GPUs is not feasible. As a result of this constraint, we do not conduct bootstrapping for COMET.

Ref	Focus	Languages	Datasets	Setting	Metrics	Baselines
Jiao et al. (2023)	Eval	Multi	Flores-101, WMT-Bio/Rob	ZS	BLEU	GoogleT, DeepL, Tencent
Peng et al. (2023)	Eval, Rob	Multi	Flores-200, WMT-News/Bio	ZS, FS	COMET, BLEU, ChrF	GoogleT
Gao et al. (2023)	Eval, Prompting	Multi/6TD	Flores-101	ZS, FS-1/5	BLUE, ChrF++, TER	GoogleT, DeepL
Zhu et al. (2023)	Eval	Multi(102)/202 TD	Flores-101	ZS, FS	BLEU	XGLM-7.5B OPT-175B BLOOMZ-7.1B / SV-M2M-12B NLLB-1.3B
Hendy et al. (2023)	Eval, Rob, DocLEval	Multi(H, L)/18TD	WMT-21/22	ZS, FS-1/5	COMET, BLEU, ChrF, HE	WMT-Best, MS-Translator
Guerreiro et al. (2023)	Eval, Hallucination	Multi H, M, L / >100 TD	Flores, WMT, TICO	ZS	spBLEU, COMET, LaBSE	SMaLL100, M2M
Wang et al. (2023)	DocLEval	Multi H	mZPRT, WMT-22, IWSLT-15/17, NewsComm-v11 Europar-v7,OpenSub-18	ZS	BLEU, TER, COMET, dBLUE,T, HE	MCN, GoogleT, MR-Doc2Doc, MR-Doc2Sent, Sent2Sent
Bang et al. (2023b)	Eval	Multi H, L 13/24 TD	Flores-200	ZS	ChrF++	FT-SOTA, ZS-SOTA
Huang et al. (2023)	Eval, Prompting	Multi / 12 TD	FLORES		SacreBLEU	text-davinci-003
Gu (2023)	Eval, Prompting	Two /	NA	ZS	NA	NA
Karpinska and Iyyer (2023)	DocLEval	Multi/18 TD	Novel	ZS	COMET BLEURT BERTSCORE COMET-QE HE	
Laskar et al. (2023a)	Eval	Multi/10TD	WMT14, WMT16, WMT19	ZS	BLEU	PaLM-540B, Finetuned SOTA
Ghosh and Caliskan (2023)	Eval, Fairness, Bias	Multi / 5 TD	NA	ZS	HE	
Lu et al. (2023a)	Eval, Prompting	Multi	Flores-200	ZS, FS-1/3	chrF++, BLEU	GPT-3.5-turbo
Ogundare and Araya (2023)	Eval	Multi	NA	ZS	SQ-Score	GoogleT
Khondaker et al. (2023)	Eval	Multi/6 TD	UNPC, MDPC	ZS, FS-3/5/10	BLUE	Supervised (AraT5)
Alyafeai et al. (2023)	Eval	Mono/1TD	UNv1	ZS, FS-3/5/10	BLUE	Supervised SOTA
Neubig and He (2023)	Eval, Rob	Multi	WMT	ZS, FS-1/5	COMET, ChrF,	GoogleT, MS Translate, DeepL

Table 7: A summary of related works. We provide a brief description of recent studies aimed at evaluating LLMs on MT tasks. MT - machine translation. TD - translation direction. ZS - zero-shot, FS - few-shot, Rob - Robustness, H, L, M - high, low, medium resource.

Variety	Mean	Median	Mode
CA	22.98	19	15
MSA	30.33	30	26
ALG	15.63	13.5	10
EGY	19.42	16	13
JOR	15.50	14	11
MAU	15.96	14	11
MOR	17.63	17	17
PAL	16.85	14.5	14
UAE	14.98	13	10
YEM	16.16	14	12
Avg.	18.52	16.45	13.9

Table 8: Length statistics of the dataset (in number of words) across the different Arabic varieties.

D Evaluation and Baselines

4.1 Evaluation Metrics

BLUE (Papineni et al., 2002). BLEU is used to evaluate machine translation quality by comparing n-gram ($n = 4$) overlap between machine-generated translations and human references. Higher scores indicate better translation quality.

COMET. (Rei et al., 2020b) Cross-lingual Opus METric measures translation quality through source-to-translation word-level alignment. Higher values indicate better quality. We use the default model[14] which supports Arabic. However, based on our inspection, we find that Arabic data used to train the model is mostly MSA. Hence, the model may not be able to capture dialect-level nuances in the source text while computing the scores.

ChrF and ChrF++ (Popović, 2015). Character n-gram F-score calculates the F-score of character n-grams in the machine translation compared to the reference translations, with higher scores denoting better quality. ChrF++ is an extension of ChrF where the word order is 2.

TER (Snover et al., 2006). Translation Error Rate measures translation quality by counting edit operations between the machine and reference translations, providing a lower score for better quality.

We use huggingface's implementation of these metrics in *evaluate*[15] package. We use all the default parameters unless otherwise specified above. While BLEU, ChrF, and TER rely mostly on direct comparisons of tokens or characters between the MT output and reference, COMET uses a model-based approach to capture more complex aspects of the translation such as semantics.

4.2 Baselines

Google Translate. In 2016, Google replaced their Statistical Machine Translation (SMT) system with Google Neural Machine Translation (GNMT) Wu et al. (2016b) featuring an LSTM with 8 encoder layers and 8 decoder ones with attention and residual connections. GNMT was trained on Google's internal datasets and it supports 133 languages. GNMT currently is powered by Transformers.

Microsoft Translator. Microsoft's translation service uses an NMT model that supports 111 different languages.

Amazon Translation. Amazon Web Services (AWS) offer batch translation with their NMT models that can translate to and from 75 languages.

NLLB-200. No Language Left Behind (NLLB et al., 2022) is an open-source Transformer model developed by META. It was trained on FLORES-200 (NLLB et al., 2022), NLLB-MD (NLLB et al., 2022), and NLLB-Seed (NLLB et al., 2022) for a total of 18B sentence pairs. It supports 202 languages (and $40,000$ translation directions), 76 of which are not supported by the aforementioned Google and Microsoft translation systems NLLB et al. (2022).

4.3 Human Analysis of Bard Helpfulness

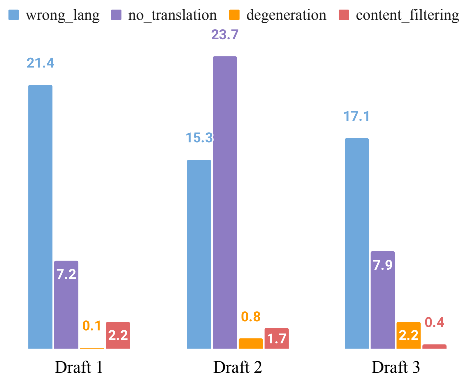


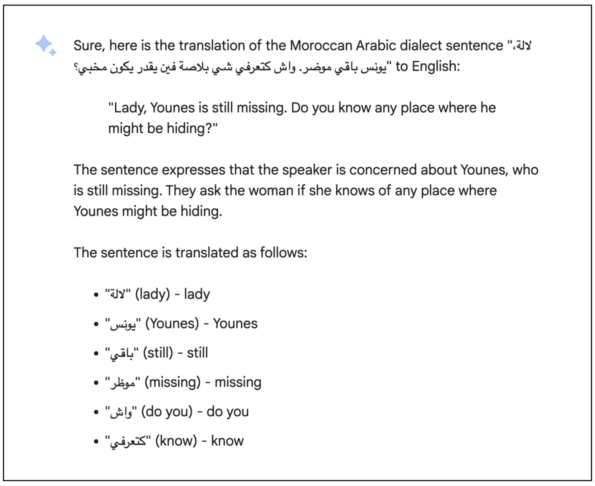
Figure 4: Percentage of Google Bard's failure to follow the prompt for each draft relative to *all errors across all drafts*.

[14] https://huggingface.co/Unbabel/wmt22-comet-da

[15] https://github.com/huggingface/evaluate

Book Name	Link
الأدب و المرؤة	https://shamela.ws/book/17869/14#p1
الأدب الكبير و الأدب الصغير	https://shamela.ws/book/7528/127
الأصنام	https://shamela.ws/book/6513
الأم	https://shamela.ws/book/1655/427#p1
الكسب	https://shamela.ws/book/6163/3
الرسالة	https://shamela.ws/book/8180/1
الرسالة الذهبية	https://shamela.ws/book/5678/91182
الناسخ والمنسوخ	https://shamela.ws/book/8491/58
أدب النفوس	https://shamela.ws/book/8245/24#p1
تاريخ المدينة	https://shamela.ws/book/13086
صحيفة حماد بن منبه	https://shamela.ws/book/7776/1
السياسة في تدبير الرياسة	https://shamela.ws/book/5678/396
النوادر في اللغة	https://shamela.ws/book/133417
منتخب الكلام في تفسير الأحلام	https://shamela.ws/book/21615/2
وصايا الملوك	https://shamela.ws/book/741/1

Table 9: List of 15 CA books from the first and second AH accompanied by direct links to each book.



Sure, here is the translation of the Moroccan Arabic dialect sentence "لالة، يونس باقي موضر. واش كتعرفي شي بلاصة فين يقدر يكون مخبي؟" to English:

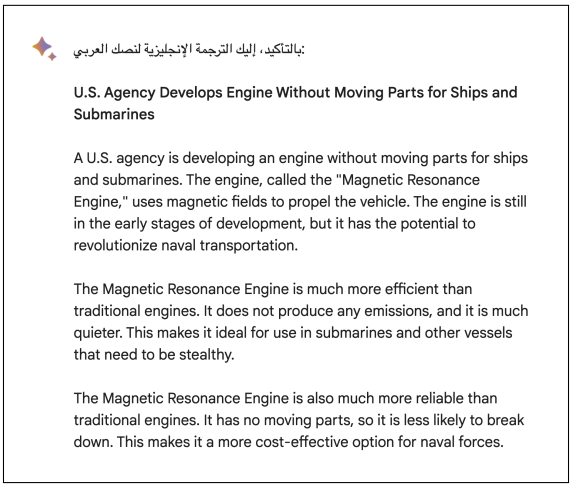
> "Lady, Younes is still missing. Do you know any place where he might be hiding?"

The sentence expresses that the speaker is concerned about Younes, who is still missing. They ask the woman if she knows of any place where Younes might be hiding.

The sentence is translated as follows:

- "لالة" (lady) - lady
- "يونس" (Younes) - Younes
- "باقي" (still) - still
- "موظر" (missing) - missing
- "واش" (do you) - do you
- "كتعرفي" (know) - know

(a) Google Bard's translation, explanation and breakdown of one dialectal sentence (from MOR).



بالتأكيد، إليك الترجمة الإنجليزية لنصك العربي:

U.S. Agency Develops Engine Without Moving Parts for Ships and Submarines

A U.S. agency is developing an engine without moving parts for ships and submarines. The engine, called the "Magnetic Resonance Engine," uses magnetic fields to propel the vehicle. The engine is still in the early stages of development, but it has the potential to revolutionize naval transportation.

The Magnetic Resonance Engine is much more efficient than traditional engines. It does not produce any emissions, and it is much quieter. This makes it ideal for use in submarines and other vessels that need to be stealthy.

The Magnetic Resonance Engine is also much more reliable than traditional engines. It has no moving parts, so it is less likely to break down. This makes it a more cost-effective option for naval forces.

(b) Google Bard's translation and context of an MSA sentence from the news domain.

Figure 5: Examples of Google Bard's translation output. The bottom parts are cropped for readability.

Metrics	Var/M	ChatGPT				GPT-4		Bard				NLLB	NLLB	Amazon	MST	GT
		0-shot	1-shot	3-shot	5-shot	0-shot	5-shot	D1	D2	D2	Avg	(SB)	(Dia)			
ChrF	CA	39.99	40.18	40.00	40.14	**40.32**	39.25	38.56	37.44	38.87	38.29	28.56	-	36.35	38.09	39.14
	MSA	69.37	69.84	69.91	70.15	69.04	69.56	63.15	60.71	61.94	61.93	65.27	-	71.04	70.35	**80.18**
	ALG	40.04	41.27	41.72	41.75	43.97	**42.91**	31.93	26.31	30.53	29.59	25.31	-	33.15	37.54	33.96
	EGY	46.46	46.97	47.66	47.67	**47.80**	47.62	42.96	39.62	43.83	42.14	33.03	36.68	40.43	43.00	43.35
	JOR	50.36	50.27	50.50	49.97	50.30	49.96	49.02	44.14	47.48	46.88	34.58	41.43	45.22	47.48	**52.40**
	MAU	32.77	32.01	32.91	32.97	**34.90**	34.38	18.49	11.68	13.36	14.51	21.74	-	29.86	30.60	28.74
	MOR	48.20	49.25	49.44	49.90	53.02	**53.60**	47.40	46.98	47.73	47.37	27.22	39.04	34.79	35.50	39.36
	PAL	53.28	52.20	53.48	53.48	**54.15**	53.42	41.54	39.69	44.43	41.89	35.68	40.02	45.79	48.80	48.64
	UAE	46.54	46.78	46.83	47.99	48.31	**49.37**	39.31	36.39	39.68	38.46	30.02	-	38.13	41.42	40.06
	YEM	40.70	41.54	41.60	**42.35**	37.64	41.30	24.28	19.93	20.31	21.51	31.52	34.8	36.99	39.29	38.32
	Avg	46.77	47.03	47.41	47.64	47.94	**48.14**	39.66	36.29	38.82	38.26	33.29	38.39	41.18	43.21	44.42
ChrF++	CA	37.89	38.15	38.04	38.22	**38.31**	37.32	37.03	35.74	37.30	36.69	27.34	-	34.65	36.22	37.44
	MSA	67.47	67.99	68.05	68.29	67.01	67.57	60.84	58.32	59.65	59.60	63.42	-	68.99	68.54	**79.00**
	ALG	38.77	40.03	40.41	40.47	42.93	**41.61**	31.18	25.69	29.83	28.90	24.16	-	31.30	35.20	32.42
	EGY	45.13	45.69	46.47	**46.54**	46.30	46.33	42.08	38.83	42.85	41.25	31.46	32.25	38.96	41.41	41.96
	JOR	49.42	49.36	49.58	49.03	48.72	48.87	48.15	43.34	46.60	46.03	33.32	40.3	43.94	45.69	**51.30**
	MAU	31.27	30.35	31.44	31.33	**33.39**	32.76	18.03	11.63	13.08	14.25	20.27	-	28.05	28.44	27.05
	MOR	47.71	48.69	48.93	49.42	52.57	**53.14**	47.31	46.71	47.54	47.19	26.32	38.65	34.00	34.76	38.57
	PAL	52.26	51.10	52.48	52.50	**53.12**	52.31	40.51	38.56	43.33	40.80	34.36	38.88	44.33	47.16	47.23
	UAE	45.82	45.88	45.94	47.19	46.44	**48.54**	38.81	35.90	39.02	37.91	29.16	-	37.32	40.21	39.11
	YEM	39.33	40.25	40.34	**41.13**	36.38	39.93	23.78	19.76	19.94	21.16	30.07	33.69	36.09	37.88	36.99
	Avg	45.51	45.75	46.17	46.41	46.52	**46.84**	38.77	35.45	37.91	37.38	31.99	37.35	39.76	41.55	43.11
TER ↓	CA	86.20	84.33	83.47	83.44	85.72	83.55	87.54	101.63	87.03	92.07	89.63	-	**81.83**	83.86	84.20
	MSA	44.73	43.56	43.19	42.70	44.13	43.77	55.07	67.96	62.54	61.86	44.79	-	40.18	39.52	**28.43**
	ALG	87.08	80.86	80.25	78.48	80.56	**78.91**	94.13	112.52	117.12	107.92	126.85	-	90.62	86.90	89.43
	EGY	75.09	72.05	72.18	**71.50**	73.44	71.61	75.22	81.33	77.04	77.86	88.69	86.29	80.56	79.17	76.40
	JOR	70.04	67.61	**65.82**	67.10	70.35	68.46	68.07	73.85	69.41	70.44	108.25	80.83	72.71	71.47	**65.82**
	MAU	102.64	95.75	95.24	94.73	98.80	91.73	106.70	105.17	245.62	152.50	129.17	-	96.85	**98.16**	99.54
	MOR	65.23	62.52	62.16	61.38	56.24	**57.25**	61.44	61.89	61.25	61.53	100.23	73.39	82.60	80.71	77.75
	PAL	60.11	59.85	57.12	57.03	**55.73**	57.38	73.29	75.46	66.10	71.62	86.76	78.23	67.38	62.41	65.84
	UAE	71.45	68.55	69.17	66.20	71.93	**65.91**	79.58	76.24	73.60	76.47	85.07	-	76.77	73.87	75.90
	YEM	84.96	82.09	80.51	81.45	85.53	**80.81**	110.53	151.27	182.99	148.26	86.01	88.89	81.20	80.58	84.36
	Avg	74.75	71.72	70.91	70.40	72.24	**69.94**	81.16	90.73	104.27	92.05	94.55	81.53	77.07	75.67	74.77

Table 10: Results in ChrF, ChrF++, and TER scores. Higher is better unless otherwise specified by ↓. Average represents the mean across all varieties. Three drafts (D1, D2, D3) from Bard are reported individually and averaged. NLLB is our MSA-based supervised baseline; NLLB (Dia) is dialect-specific. Abbreviations: SB - supervised baseline, Dia - dialect, Var - varieties, M - model, MST - Microsoft Translation, GT - Google Translate. Best results are in **bold**.

Met	Var/M	ChatGPT				GPT-4		Bard				NLLB	NLLB	Amazon	MST	GT
		0-shot	1-shot	3-shot	5-shot	0-shot	5-shot	D1	D2	D2	Avg	(SB)	(Dia)			
BLEU	CA	11.19$^{\pm1.94}$	12.08$^{\pm1.94}$	12.21$^{\pm2.06}$	12.48$^{\pm2.07}$	11.76$^{\pm1.85}$	11.41$^{\pm1.83}$	12.30$^{\pm2.02}$	10.92$^{\pm2.62}$	12.35$^{\pm2.14}$	12.30$^{\pm2.02}$	7.13$^{\pm1.55}$	-	11.22$^{\pm2.03}$	11.99$^{\pm2.10}$	**14.23$^{\pm2.72}$**
	MSA	42.97$^{\pm2.98}$	44.08$^{\pm3.13}$	44.32$^{\pm3.05}$	44.84$^{\pm3.16}$	42.94$^{\pm3.11}$	43.54$^{\pm2.76}$	36.38$^{\pm3.58}$	32.99$^{\pm4.83}$	34.97$^{\pm5.01}$	36.38$^{\pm3.58}$	41.38$^{\pm3.75}$	-	46.48$^{\pm3.33}$	47.23$^{\pm3.48}$	**65.47$^{\pm5.21}$**
	ALG	14.54$^{\pm2.57}$	16.43$^{\pm2.81}$	17.16$^{\pm3.00}$	17.33$^{\pm2.75}$	**18.54$^{\pm2.77}$**	18.08$^{\pm2.74}$	14.95$^{\pm3.30}$	11.75$^{\pm3.42}$	13.38$^{\pm4.05}$	14.95$^{\pm3.30}$	6.81$^{\pm2.01}$	-	9.89$^{\pm2.26}$	11.42$^{\pm2.30}$	11.72$^{\pm2.07}$
	EGY	19.80$^{\pm2.54}$	21.03$^{\pm2.49}$	21.36$^{\pm2.37}$	**21.67$^{\pm2.47}$**	20.99$^{\pm2.58}$	21.43$^{\pm2.78}$	21.17$^{\pm2.91}$	19.26$^{\pm3.26}$	20.81$^{\pm3.32}$	21.17$^{\pm2.91}$	10.62$^{\pm2.35}$	12.46$^{\pm2.01}$	14.78$^{\pm2.21}$	16.62$^{\pm2.52}$	17.89$^{\pm2.55}$
	JOR	25.51$^{\pm3.64}$	26.59$^{\pm3.65}$	27.43$^{\pm3.67}$	26.90$^{\pm3.60}$	24.56$^{\pm3.04}$	25.25$^{\pm3.10}$	26.97$^{\pm3.51}$	23.30$^{\pm3.34}$	25.08$^{\pm3.16}$	26.97$^{\pm3.51}$	12.93$^{\pm3.80}$	18.31$^{\pm2.84}$	21.13$^{\pm3.22}$	21.39$^{\pm3.02}$	**29.55$^{\pm4.06}$**
	MAU	8.53$^{\pm1.73}$	8.93$^{\pm1.87}$	9.17$^{\pm1.89}$	8.96$^{\pm2.00}$	9.19$^{\pm1.79}$	**9.96$^{\pm1.97}$**	5.72$^{\pm1.71}$	4.19$^{\pm1.82}$	2.64$^{\pm1.45}$	5.72$^{\pm1.71}$	3.37$^{\pm1.65}$		7.06$^{\pm1.62}$	6.79$^{\pm1.65}$	7.45$^{\pm1.95}$
	MOR	27.14$^{\pm3.41}$	28.12$^{\pm3.50}$	28.87$^{\pm3.18}$	29.81$^{\pm3.32}$	32.86$^{\pm3.38}$	**33.40$^{\pm3.46}$**	31.23$^{\pm4.02}$	30.52$^{\pm3.83}$	31.06$^{\pm3.73}$	31.23$^{\pm4.02}$	9.30$^{\pm2.72}$	19.46$^{\pm2.67}$	12.61$^{\pm2.12}$	14.25$^{\pm2.15}$	16.96$^{\pm2.48}$
	PAL	29.43$^{\pm3.26}$	29.37$^{\pm3.00}$	31.46$^{\pm3.24}$	31.42$^{\pm3.27}$	**31.81$^{\pm3.00}$**	30.39$^{\pm3.01}$	21.96$^{\pm3.74}$	20.21$^{\pm3.77}$	23.92$^{\pm3.88}$	21.96$^{\pm3.74}$	14.03$^{\pm2.99}$	17.08$^{\pm2.45}$	21.77$^{\pm2.63}$	24.08$^{\pm2.71}$	25.34$^{\pm3.05}$
	UAE	24.14$^{\pm3.21}$	24.52$^{\pm3.09}$	24.49$^{\pm3.38}$	26.00$^{\pm3.52}$	23.92$^{\pm3.17}$	**26.84$^{\pm3.31}$**	21.49$^{\pm3.69}$	19.30$^{\pm3.34}$	21.15$^{\pm3.41}$	21.49$^{\pm3.69}$	10.95$^{\pm2.25}$		16.65$^{\pm2.51}$	18.95$^{\pm2.87}$	19.36$^{\pm2.86}$
	YEM	14.79$^{\pm2.08}$	16.02$^{\pm2.21}$	16.94$^{\pm2.41}$	**17.46$^{\pm2.36}$**	13.98$^{\pm2.23}$	16.14$^{\pm2.32}$	9.49$^{\pm2.85}$	7.22$^{\pm3.17}$	6.29$^{\pm3.12}$	9.49$^{\pm2.85}$	9.28$^{\pm1.72}$	12.46$^{\pm2.01}$	14.29$^{\pm1.98}$	14.19$^{\pm2.02}$	13.18$^{\pm2.10}$
	Avg	21.80$^{\pm2.74}$	22.72$^{\pm2.77}$	23.34$^{\pm2.83}$	**23.69$^{\pm2.85}$**	23.05$^{\pm2.69}$	23.64$^{\pm2.73}$	20.17$^{\pm3.13}$	17.97$^{\pm3.34}$	19.16$^{\pm3.33}$	20.17$^{\pm3.13}$	12.58$^{\pm2.48}$	15.95$^{\pm2.40}$	17.59$^{\pm2.39}$	18.69$^{\pm2.48}$	22.12$^{\pm2.90}$
ChrF	CA	39.96$^{\pm1.65}$	40.18$^{\pm1.67}$	40.04$^{\pm1.73}$	40.09$^{\pm1.77}$	**40.34$^{\pm1.61}$**	39.28$^{\pm1.59}$	38.64$^{\pm2.01}$	37.53$^{\pm2.61}$	38.88$^{\pm1.96}$	37.98$^{\pm1.98}$	28.61$^{\pm2.44}$	-	36.39$^{\pm1.89}$	38.24$^{\pm1.89}$	39.29$^{\pm2.26}$
	MSA	69.44$^{\pm1.90}$	69.85$^{\pm1.95}$	69.94$^{\pm1.91}$	70.22$^{\pm1.89}$	68.99$^{\pm1.91}$	69.60$^{\pm1.79}$	63.19$^{\pm3.60}$	60.76$^{\pm4.51}$	62.13$^{\pm4.14}$	61.22$^{\pm3.96}$	65.30$^{\pm2.59}$	-	70.97$^{\pm2.19}$	70.30$^{\pm2.24}$	**80.16$^{\pm3.09}$**
	ALG	40.10$^{\pm2.30}$	41.29$^{\pm2.42}$	41.75$^{\pm2.42}$	41.79$^{\pm2.35}$	**44.11$^{\pm2.40}$**	43.11$^{\pm2.39}$	32.02$^{\pm4.91}$	26.55$^{\pm4.80}$	31.16$^{\pm5.28}$	28.09$^{\pm5.16}$	25.46$^{\pm2.69}$	-	33.15$^{\pm2.15}$	37.55$^{\pm2.21}$	34.03$^{\pm2.36}$
	EGY	46.34$^{\pm2.28}$	46.92$^{\pm2.18}$	47.60$^{\pm2.16}$	47.51$^{\pm2.26}$	**47.62**$^{\pm2.25}$	47.55$^{\pm2.27}$	42.91$^{\pm3.33}$	39.53$^{\pm4.11}$	43.71$^{\pm3.41}$	40.92$^{\pm3.38}$	33.05$^{\pm2.71}$	36.69$^{\pm2.48}$	40.28$^{\pm1.98}$	42.99$^{\pm2.18}$	43.26$^{\pm2.52}$
	JOR	50.20$^{\pm2.91}$	50.11$^{\pm2.90}$	50.51$^{\pm2.98}$	50.04$^{\pm2.76}$	50.25$^{\pm2.60}$	49.87$^{\pm2.58}$	49.09$^{\pm3.39}$	44.06$^{\pm3.84}$	47.50$^{\pm3.10}$	45.21$^{\pm3.20}$	34.64$^{\pm3.55}$	41.40$^{\pm2.66}$	45.16$^{\pm2.80}$	47.48$^{\pm2.61}$	**52.51$^{\pm3.31}$**
	MAU	32.74$^{\pm1.99}$	31.99$^{\pm2.03}$	32.87$^{\pm2.01}$	32.97$^{\pm2.05}$	**34.95$^{\pm2.10}$**	34.42$^{\pm2.21}$	18.50$^{\pm3.44}$	11.86$^{\pm3.52}$	13.53$^{\pm3.66}$	12.42$^{\pm3.59}$	21.72$^{\pm2.40}$		29.76$^{\pm1.82}$	30.57$^{\pm1.82}$	28.74$^{\pm2.12}$
	MOR	48.29$^{\pm2.61}$	49.16$^{\pm2.65}$	49.46$^{\pm2.50}$	49.93$^{\pm2.63}$	53.02$^{\pm2.65}$	**53.69$^{\pm2.69}$**	47.44$^{\pm4.42}$	47.04$^{\pm4.27}$	47.82$^{\pm4.11}$	47.30$^{\pm4.21}$	27.26$^{\pm2.94}$	39.01$^{\pm2.30}$	34.74$^{\pm2.15}$	35.50$^{\pm2.09}$	39.35$^{\pm2.33}$
	PAL	53.25$^{\pm2.30}$	52.23$^{\pm2.17}$	53.58$^{\pm2.32}$	53.49$^{\pm2.33}$	**54.19$^{\pm2.36}$**	53.48$^{\pm2.29}$	41.45$^{\pm4.95}$	39.87$^{\pm4.91}$	44.19$^{\pm4.56}$	41.31$^{\pm4.69}$	35.75$^{\pm3.28}$	39.94$^{\pm2.58}$	45.94$^{\pm2.13}$	48.78$^{\pm2.16}$	48.65$^{\pm2.63}$
	UAE	46.48$^{\pm2.65}$	46.85$^{\pm2.67}$	46.86$^{\pm2.75}$	47.92$^{\pm2.78}$	48.39$^{\pm2.89}$	**49.38$^{\pm2.72}$**	39.47$^{\pm4.66}$	36.28$^{\pm4.24}$	39.72$^{\pm4.35}$	37.43$^{\pm4.45}$	29.98$^{\pm2.35}$		38.10$^{\pm2.23}$	41.41$^{\pm2.61}$	40.23$^{\pm2.79}$
	YEM	40.81$^{\pm2.15}$	41.67$^{\pm2.25}$	41.59$^{\pm2.43}$	**42.53$^{\pm2.22}$**	37.54$^{\pm3.00}$	41.16$^{\pm2.52}$	24.44$^{\pm4.65}$	20.17$^{\pm4.66}$	20.78$^{\pm4.99}$	20.37$^{\pm4.88}$	31.48$^{\pm2.06}$	34.83$^{\pm2.09}$	36.96$^{\pm2.04}$	39.27$^{\pm2.08}$	38.32$^{\pm2.15}$
	Avg	46.76$^{\pm2.27}$	47.03$^{\pm2.29}$	47.42$^{\pm2.32}$	47.65$^{\pm2.30}$	47.94$^{\pm2.38}$	**48.15$^{\pm2.30}$**	39.72$^{\pm3.94}$	36.37$^{\pm4.15}$	38.94$^{\pm3.96}$	37.23$^{\pm3.95}$	33.33$^{\pm2.70}$	38.37$^{\pm2.42}$	41.15$^{\pm2.14}$	43.21$^{\pm2.19}$	44.45$^{\pm2.56}$
ChrF++	CA	37.85$^{\pm1.66}$	38.16$^{\pm1.68}$	38.08$^{\pm1.76}$	38.18$^{\pm1.80}$	**38.33$^{\pm1.64}$**	37.37$^{\pm1.62}$	37.10$^{\pm2.03}$	35.84$^{\pm2.60}$	37.31$^{\pm1.98}$	36.33$^{\pm2.00}$	27.41$^{\pm2.32}$	-	34.69$^{\pm1.90}$	36.37$^{\pm1.92}$	37.60$^{\pm2.30}$
	MSA	67.54$^{\pm1.98}$	68.01$^{\pm2.03}$	68.08$^{\pm2.00}$	68.35$^{\pm1.99}$	66.96$^{\pm2.01}$	67.61$^{\pm1.84}$	60.88$^{\pm3.52}$	58.36$^{\pm4.39}$	59.84$^{\pm4.05}$	58.85$^{\pm3.87}$	63.45$^{\pm2.66}$	-	68.91$^{\pm2.25}$	68.49$^{\pm2.32}$	**78.97$^{\pm3.24}$**
	ALG	38.84$^{\pm2.32}$	40.06$^{\pm2.42}$	40.44$^{\pm2.45}$	40.53$^{\pm2.38}$	**43.08$^{\pm2.44}$**	41.80$^{\pm2.42}$	31.25$^{\pm4.77}$	25.94$^{\pm4.65}$	30.44$^{\pm5.08}$	27.44$^{\pm4.98}$	24.34$^{\pm2.59}$	-	31.31$^{\pm2.12}$	35.22$^{\pm2.22}$	32.48$^{\pm2.31}$
	EGY	45.01$^{\pm2.29}$	45.67$^{\pm2.20}$	**46.40$^{\pm2.15}$**	46.38$^{\pm2.24}$	46.12$^{\pm2.26}$	46.25$^{\pm2.28}$	42.03$^{\pm3.24}$	38.76$^{\pm3.99}$	42.76$^{\pm3.30}$	40.09$^{\pm3.28}$	31.50$^{\pm2.64}$	35.26$^{\pm2.45}$	38.80$^{\pm1.99}$	41.40$^{\pm2.17}$	41.87$^{\pm2.51}$
	JOR	49.26$^{\pm2.94}$	49.20$^{\pm2.93}$	49.59$^{\pm3.01}$	49.08$^{\pm2.79}$	48.68$^{\pm2.59}$	48.80$^{\pm2.60}$	48.21$^{\pm3.36}$	43.28$^{\pm3.78}$	46.62$^{\pm3.10}$	44.39$^{\pm3.19}$	33.39$^{\pm3.52}$	40.27$^{\pm2.69}$	43.87$^{\pm2.78}$	45.69$^{\pm2.65}$	**51.40$^{\pm3.35}$**
	MAU	31.26$^{\pm1.98}$	30.35$^{\pm2.00}$	31.40$^{\pm2.00}$	31.32$^{\pm2.04}$	**33.44$^{\pm2.05}$**	32.82$^{\pm2.20}$	18.04$^{\pm3.29}$	11.80$^{\pm3.35}$	13.25$^{\pm3.51}$	12.28$^{\pm3.44}$	20.28$^{\pm2.32}$	-	27.94$^{\pm1.80}$	28.42$^{\pm1.80}$	27.07$^{\pm2.07}$
	MOR	47.79$^{\pm2.65}$	48.61$^{\pm2.67}$	48.96$^{\pm2.51}$	49.45$^{\pm2.64}$	52.57$^{\pm2.66}$	**53.23$^{\pm2.73}$**	47.36$^{\pm4.36}$	46.76$^{\pm4.20}$	47.64$^{\pm4.05}$	47.05$^{\pm4.15}$	26.40$^{\pm2.96}$	38.62$^{\pm2.28}$	33.95$^{\pm2.12}$	34.76$^{\pm2.06}$	38.56$^{\pm2.33}$
	PAL	52.22$^{\pm2.34}$	51.14$^{\pm2.20}$	52.55$^{\pm2.36}$	52.49$^{\pm2.37}$	**53.17$^{\pm2.37}$**	52.38$^{\pm2.30}$	40.41$^{\pm4.83}$	38.75$^{\pm4.79}$	43.07$^{\pm4.48}$	40.19$^{\pm4.60}$	34.43$^{\pm3.21}$	38.80$^{\pm2.56}$	44.47$^{\pm2.12}$	47.14$^{\pm2.17}$	47.26$^{\pm2.63}$
	UAE	45.76$^{\pm2.67}$	45.95$^{\pm2.69}$	45.98$^{\pm2.76}$	47.12$^{\pm2.81}$	46.49$^{\pm2.85}$	**48.55$^{\pm2.75}$**	38.96$^{\pm4.59}$	35.77$^{\pm4.19}$	39.07$^{\pm4.29}$	36.87$^{\pm4.39}$	29.13$^{\pm2.33}$	-	37.29$^{\pm2.22}$	40.21$^{\pm2.64}$	39.28$^{\pm2.81}$
	YEM	39.48$^{\pm2.11}$	40.43$^{\pm2.22}$	40.36$^{\pm2.38}$	**41.37$^{\pm2.19}$**	36.31$^{\pm2.93}$	39.80$^{\pm2.48}$	23.96$^{\pm4.49}$	20.00$^{\pm4.51}$	20.40$^{\pm4.82}$	20.13$^{\pm4.71}$	30.04$^{\pm2.00}$	33.72$^{\pm2.04}$	36.04$^{\pm2.00}$	37.86$^{\pm2.03}$	36.98$^{\pm2.11}$
	Avg	45.50$^{\pm2.29}$	45.76$^{\pm2.30}$	46.18$^{\pm2.34}$	46.43$^{\pm2.33}$	46.52$^{\pm2.38}$	**46.86$^{\pm2.32}$**	38.82$^{\pm3.85}$	35.53$^{\pm4.04}$	38.04$^{\pm3.87}$	36.37$^{\pm3.86}$	32.04$^{\pm2.65}$	37.33 $^{\pm2.40}$	39.73$^{\pm2.13}$	41.56$^{\pm2.20}$	43.15$^{\pm2.57}$
TER ↓	CA	86.32$^{\pm4.42}$	84.28$^{\pm4.35}$	**83.39$^{\pm4.32}$**	83.50$^{\pm4.62}$	85.72$^{\pm4.59}$	83.33$^{\pm4.27}$	87.71$^{\pm5.06}$	101.91$^{\pm34.76}$	87.24$^{\pm4.91}$	97.02$^{\pm4.96}$	89.41$^{\pm10.14}$	-	81.81$^{\pm3.67}$	83.50$^{\pm4.10}$	83.87$^{\pm5.04}$
	MSA	44.64$^{\pm3.13}$	43.62$^{\pm3.16}$	43.17$^{\pm3.22}$	42.63$^{\pm3.27}$	44.30$^{\pm3.13}$	43.71$^{\pm2.91}$	55.05$^{\pm8.72}$	67.26$^{\pm16.34}$	62.66$^{\pm16.63}$	65.73$^{\pm13.99}$	44.86$^{\pm3.52}$	-	40.43$^{\pm3.45}$	39.55$^{\pm3.32}$	**28.59$^{\pm4.78}$**
	ALG	87.28$^{\pm6.42}$	80.95$^{\pm4.94}$	80.14$^{\pm4.88}$	**78.33$^{\pm4.70}$**	80.53$^{\pm5.11}$	78.60$^{\pm5.18}$	94.21$^{\pm12.82}$	111.99$^{\pm35.96}$	115.62$^{\pm37.82}$	113.20$^{\pm29.49}$	128.00$^{\pm46.39}$	-	90.41$^{\pm5.12}$	86.93$^{\pm5.48}$	89.59$^{\pm4.20}$
	EGY	75.13$^{\pm3.93}$	71.94$^{\pm3.74}$	72.12$^{\pm3.43}$	**71.38$^{\pm3.77}$**	73.60$^{\pm4.30}$	71.60$^{\pm4.41}$	75.37$^{\pm8.19}$	81.23$^{\pm10.53}$	77.70$^{\pm14.48}$	80.05$^{\pm12.38}$	88.40$^{\pm20.57}$	86.04$^{\pm10.58}$	80.63$^{\pm3.74}$	79.12$^{\pm4.67}$	76.45$^{\pm3.89}$
	JOR	70.36$^{\pm5.24}$	67.77$^{\pm4.98}$	66.04$^{\pm4.70}$	67.04$^{\pm4.85}$	70.33$^{\pm4.44}$	68.46$^{\pm4.25}$	68.13$^{\pm6.01}$	73.84$^{\pm5.44}$	69.47$^{\pm4.86}$	72.38$^{\pm5.24}$	108.32$^{\pm35.11}$	80.80$^{\pm4.20}$	72.97$^{\pm4.72}$	71.53$^{\pm4.71}$	**65.56$^{\pm5.27}$**
	MAU	102.56$^{\pm5.74}$	95.72$^{\pm4.50}$	95.50$^{\pm4.98}$	94.98$^{\pm4.79}$	99.08$^{\pm4.85}$	**91.61$^{\pm4.34}$**	107.13$^{\pm11.27}$	104.58$^{\pm9.07}$	246.24$^{\pm88.06}$	151.80$^{\pm62.46}$	130.19$^{\pm35.99}$	-	96.82$^{\pm4.16}$	98.30$^{\pm4.63}$	99.65$^{\pm5.09}$
	MOR	65.20$^{\pm3.91}$	62.67$^{\pm3.78}$	62.00$^{\pm3.51}$	61.47$^{\pm3.82}$	**56.30$^{\pm3.76}$**	57.06$^{\pm4.05}$	61.46$^{\pm4.88}$	61.78$^{\pm4.81}$	61.13$^{\pm4.56}$	61.56$^{\pm4.67}$	100.36$^{\pm24.83}$	73.51$^{\pm3.13}$	82.81$^{\pm3.77}$	80.81$^{\pm3.62}$	77.68$^{\pm3.89}$
	PAL	59.96$^{\pm3.89}$	59.88$^{\pm3.31}$	57.09$^{\pm3.46}$	57.12$^{\pm3.40}$	**55.71$^{\pm3.59}$**	57.32$^{\pm3.55}$	72.77$^{\pm11.10}$	75.25$^{\pm11.72}$	66.45$^{\pm4.84}$	72.32$^{\pm6.93}$	86.11$^{\pm19.44}$	78.47$^{\pm3.08}$	67.31$^{\pm3.17}$	62.50$^{\pm3.42}$	65.88$^{\pm4.06}$
	UAE	71.55$^{\pm5.21}$	68.55$^{\pm4.31}$	69.23$^{\pm4.59}$	66.19$^{\pm4.42}$	71.65$^{\pm4.50}$	**65.94$^{\pm4.31}$**	78.90$^{\pm9.64}$	76.62$^{\pm5.74}$	73.59$^{\pm4.99}$	75.61$^{\pm6.54}$	85.08$^{\pm7.65}$	-	76.88$^{\pm3.93}$	73.83$^{\pm4.22}$	75.62$^{\pm4.37}$
	YEM	83.06$^{\pm4.04}$	80.08$^{\pm3.86}$	79.04$^{\pm4.02}$	**79.47$^{\pm4.20}$**	85.50$^{\pm4.17}$	80.89$^{\pm4.24}$	110.69$^{\pm38.27}$	153.21$^{\pm74.25}$	182.04$^{\pm86.90}$	162.82$^{\pm70.69}$	86.00$^{\pm4.62}$	88.80$^{\pm3.52}$	81.22$^{\pm3.48}$	80.47$^{\pm3.88}$	84.17$^{\pm4.04}$
	Avg	74.61$^{\pm4.59}$	71.55$^{\pm4.09}$	70.77$^{\pm4.11}$	70.21$^{\pm4.18}$	72.27$^{\pm4.24}$	**69.85$^{\pm4.15}$**	81.14$^{\pm11.60}$	90.77$^{\pm20.86}$	104.21$^{\pm26.80}$	95.25$^{\pm21.73}$	94.67$^{\pm20.83}$	81.52$^{\pm4.90}$	77.13$^{\pm3.92}$	75.65$^{\pm4.21}$	74.71$^{\pm4.46}$

Table 11: Bootstraped results for BLEU, ChrF, ChrF++, and TER with standard deviation in superscript. Higher is better unless otherwise specified by ↓. Average represents the mean across all varieties. Three drafts (D1, D2, D3) from Bard are reported individually and averaged. NLLB is our MSA-based supervised baseline; NLLB (Dia) is dialect-specific. Abbreviations: SB - supervised baseline, Dia - dialect, Var - varieties, M - model, MST - Microsoft Translation, GT - Google Translate. Best results are in **bold**.

Met	Mo/Var	CGPT	GPT-4	Bard		NLLB	Amazon	MST	GT
				D1	Avg				
ChrF	CA	**50.59**	50.35	46.99	**47.54**	**37.76**	**40.08**	**42.73**	**48.58**
	CA*	50.01	**50.49**	**47.49**	47.35	32.13	39.53	**42.73**	46.95
ChrF++	CA	**49.23**	48.99	46.11	46.74	**37.09**	**39.33**	**41.95**	**47.68**
	CA*	48.97	**49.25**	**47.02**	**46.81**	31.71	38.78	**41.95**	45.93
TER ↓	CA	69.98	67.17	69.14	69.61	77.95	73.45	**66.23**	**62.76**
	CA*	**68.48**	**66.04**	**64.82**	**65.63**	**75.42**	**68.95**	**66.23**	64.92

Table 12: The effect of diacritics on translation quality. CA* is without diacritics. Higher is better unless otherwise specified by ↓. The best results are in **bold**.

Met	Mo/Var	CGPT	GPT-4	Bard		NLLB	Amazon	MST	GT
				D1	Avg				
BLEU	CA	23.47 $^{\pm 2.54}$	23.87 $^{\pm 2.11}$	22.98 $^{\pm 2.00}$	22.99 $^{\pm 1.99}$	**15.92** $^{\pm 1.91}$	17.41 $^{\pm 2.28}$	**20.10** $^{\pm 2.04}$	**26.48** $^{\pm 2.70}$
	CA*	**23.49** $^{\pm 2.49}$	**24.50** $^{\pm 2.04}$	**25.33** $^{\pm 1.86}$	**24.22** $^{\pm 1.99}$	13.51 $^{\pm 2.06}$	**18.67** $^{\pm 2.38}$	20.02 $^{\pm 2.03}$	24.48 $^{\pm 2.56}$
ChrF	CA	**50.60** $^{\pm 1.79}$	50.43 $^{\pm 1.83}$	46.99 $^{\pm 1.69}$	**47.68** $^{\pm 1.77}$	**37.74** $^{\pm 1.68}$	**40.11** $^{\pm 1.94}$	**42.76** $^{\pm 1.76}$	**48.61** $^{\pm 2.10}$
	CA*	50.07 $^{\pm 1.87}$	**50.58** $^{\pm 1.69}$	**47.50** $^{\pm 1.86}$	47.04 $^{\pm 1.76}$	32.08 $^{\pm 1.93}$	39.61 $^{\pm 1.87}$	42.65 $^{\pm 1.71}$	46.88 $^{\pm 1.95}$
ChrF++	CA	**49.24** $^{\pm 1.83}$	49.06 $^{\pm 1.87}$	46.11 $^{\pm 1.71}$	**46.89** $^{\pm 1.78}$	**37.06** $^{\pm 1.73}$	**39.36** $^{\pm 1.92}$	**41.99** $^{\pm 1.76}$	**47.71** $^{\pm 2.13}$
	CA*	49.03 $^{\pm 1.97}$	**49.34** $^{\pm 1.72}$	**47.04** $^{\pm 1.81}$	46.42 $^{\pm 1.75}$	31.65 $^{\pm 1.96}$	38.85 $^{\pm 1.90}$	41.88 $^{\pm 1.71}$	45.85 $^{\pm 1.99}$
TER ↓	CA	70.00 $^{\pm 3.30}$	67.08 $^{\pm 2.60}$	69.10 $^{\pm 2.71}$	70.39 $^{\pm 3.04}$	77.96 $^{\pm 2.84}$	73.40 $^{\pm 3.08}$	**66.19** $^{\pm 2.55}$	**62.74** $^{\pm 2.96}$
	CA*	**68.48** $^{\pm 3.48}$	**65.97** $^{\pm 2.77}$	**64.93** $^{\pm 2.69}$	**66.20** $^{\pm 2.80}$	**75.42** $^{\pm 2.30}$	**68.89** $^{\pm 2.56}$	**66.19** $^{\pm 2.49}$	65.04 $^{\pm 2.77}$

Table 13: Bootstrapped scores in BLEU, ChrF, ChrF++, and TER. CA* is without diacritics. Higher is better unless otherwise specified by ↓.

MSA	English	MSA	English
كَتَبَ	He wrote	كُتُبٌ	Books
قَسَّمَ	He divided	قَسَمٌ	Oath
عَلَمٌ	Flag	عِلْمٌ	Science
صِدْقٌ	Sincerity	صَدَّقَ	He believed
وُلِدَ	He was born	وَلَدٌ	Boy
ذُرَةٌ	Corn	ذَرَّةٌ	Atom
مَدْرَسَةٌ	School	مُدَرِّسَةٌ	Teacher
حَمَّامٌ	Bathroom	حَمَامٌ	Pigeons
حِدَادٌ	Mourning	حَدَّادٌ	Blacksmith
شَعْرٌ	Hair	شِعْرٌ	Poetry
مَرْكَبَةٌ	Vehicle	مُرَكَّبَةٌ	Composite
سُكْرٌ	Drunkenness	سُكَّرٌ	Sugar
نَجَمَ	It resulted	نَجْمٌ	Star
رَجُلٌ	Man	رِجْلٌ	Foot
بَشَرٌ	Humans	بَشَّرَ	He preached
مَلِكٌ	King	مُلْكٌ	Possession
جَدٌّ	Grandfather	جِدٌّ	Seriousness
جَمَلٌ	Camel	جُمَلٌ	Sentences
حَكَمٌ	Referee	حُكْمٌ	Ruling
سَمَكٌ	Fish	سُمْكٌ	Thickness

Table 14: Heterophonic homographs used to test model sensitivity to diacritics.

Variety	**WL**	**NT**	**D**	**CF**	**Total**
CA	9	10	3	0	22
MSA	1	27	1	0	29
ALG	81	72	5	6	164
EGY	11	36	2	17	66
JOR	14	22	2	2	40
MAU	160	136	6	1	303
MOR	42	13	3	2	60
PAL	57	13	0	8	78
UAE	33	49	2	10	94
YEM	178	44	10	0	232
Total	586	422	34	46	1088

Table 15: Counts of the helpfulness errors that Google Bard suffers from by Arabic variety and error category.

Var	Source	Reference	Translation
CA	فعلى العاقل أن يعلم أن الناس مشتركون مستوون في الحب لما يوافق، والبغض	A wise man knows that people are the same in love and hate.	It is incumbent upon the wise person to know that people are equal in their love for what they agree upon, and in their hatred. - **ChatGPT**
			So it is upon the wise person to know that people are united and equal in their love for what is agreeable and in their hatred. - **GPT-4**
			So the wise man should know that people are united and equal in love for what agrees with them, and in hatred for what disagrees with them. - **Bard**
			A reasonable person would know that people share a common level of love for what is right and hate for what is wrong. - **NLLB**
			So the sane person should know that people are equal in love for what agrees and hate - **GT**
MSA	تونس بين الصدمة والبحث عن حلّ لمحنة المهاجرين في صفاقس	Tunisia is between shock and looking for a solution to the Sfax migrant crisis.	Tunisia is between shock and searching for a solution to the plight of immigrants in Sfax. - **ChatGPT**
			Tunisia is between shock and the search for a solution to the migrant crisis in Sfax. **GPT-4**
			Tunisia between shock and the search for a solution to the migrant crisis in Sfax - **Bard**
			Tunisia is between shock and finding a solution to the migrant crisis in Safiq . **NLLB**
			Tunisia is between shock and the search for a solution to the plight of migrants in Sfax - **GT**
ALG	لا نتا لا غيرك ما قادرين تحركو شعرة من راسي	Neither you nor anyone else can scare me..	You and no one else can move a single hair on my head. - **ChatGPT**
			Neither you nor anyone else can move a single hair on my head. - **GPT-4**
			Neither you nor anyone else can move a hair on my head - **Bard**
			Neither you nor Neta can move a hair of my head . - **NLLB**
			Neither you nor anyone else can move a hair from my head - **GT**
MOR	عنده واحد الوشام هنايا. راه كيجي معاه واحد الكلب مرة مرة بالليل	He has a tattoo here. He's accompanied by a dog from time to time at night.	He has a tattoo on his arm. He always comes with a dog sometimes at night. - **ChatGPT**
			He has a tattoo here. Sometimes he comes with a dog at night. - **GPT-4**
			He has a tattoo here. He comes with a dog sometimes at night. - **Bard**
			He has one tattoo here. He's got a dog once a night. - **NLLB**
			He has one tattoo Hanaya. He saw Keiji with one of the dogs once at night - **GT**
MAU	حيبللا كلمو يغير هو مزال ألا مصمم علنو ليهي يشتغل	Hebila talked to him, but he persisted in working.	They told him to change, but he is still stubborn and refuses to work. - **ChatGPT**
			God willing, talk to him to change; he is still insistent on working for him. - **GPT-4**
			He has changed a lot but he is still determined to work hard - **Bard**
			No matter how much he changes , he 's still not a public designer . - **NLLB**
			He doesn't want to talk to him about changing, he is still not a public designer, so he can work - **GT**
UAE	أنا أبغي أفهم أنت ليش مب راضي تقنع أن أنا وأنت واحد ؟	I want to understand why you are not convinced that you and I are the same person.	I want to understand why you're not convinced that you and I are one? - **ChatGPT**
			I want to understand why you're not convinced that you and I are one. - **GPT-4**
			I want to understand why you are not willing to be convinced that we are one - **Bard**
			I want to understand why you 're so happy to convince me that you and I are one ? - **NLLB**
			I want to understand why you are not satisfied with being convinced that you and I are one? - **GT**

Table 16: Translations generated by the LLMs, the supervised baseline and the best performing commercial system (Google Translate). Translations from ChatGPT, GPT-4 and Bard were obtained under the zero-shot setting.

Leveraging Domain Adaptation and Data Augmentation to Improve Qur’anic IR in English and Arabic

Vera Pavlova
rttl.ai
Dubai, UAE
v@rttl.ai

Abstract

In this work, we approach the problem of Qur’anic information retrieval (IR) in Arabic and English. Using the latest state-of-the-art methods in neural IR, we research what helps to tackle this task more efficiently. Training retrieval models requires a lot of data, which is difficult to obtain for training in-domain. Therefore, we commence with training on a large amount of general domain data and then continue training on in-domain data. To handle the lack of in-domain data, we employed a data augmentation technique, which considerably improved results in MRR@10 and NDCG@5 metrics, setting the state-of-the-art in Qur’anic IR for both English and Arabic. The absence of an Islamic corpus and domain-specific model for IR task in English motivated us to address this lack of resources and take preliminary steps of the Islamic corpus compilation and domain-specific language model (LM) pre-training, which helped to improve the performance of the retrieval models that use the domain-specific LM as the shared backbone. We examined several language models (LMs) in Arabic to select one that efficiently deals with the Qur’anic IR task. Besides transferring successful experiments from English to Arabic, we conducted additional experiments with retrieval task in Arabic to amortize the scarcity of general domain datasets used to train the retrieval models. Handling Qur’anic IR task combining English and Arabic allowed us to enhance the comparison and share valuable insights across models and languages.

1 Introduction

Recent advances in Natural Language Processing (NLP) have helped to improve search relevance and retrieval quality. Nevertheless, deep-learning techniques, specifically transformer-based approaches (Vaswani et al., 2017), are hardly employed in Quran’ic NLP (Bashir et al., 2023). In this work, we will utilize the latest state-of-the-art neural retrieval models to compare what works best for solving the IR task in the Islamic domain. Moreover, we proposed a data-augmentation technique to generate data for in-domain training appropriate for the IR task involving the Holy Qur’an (see Figure 1).

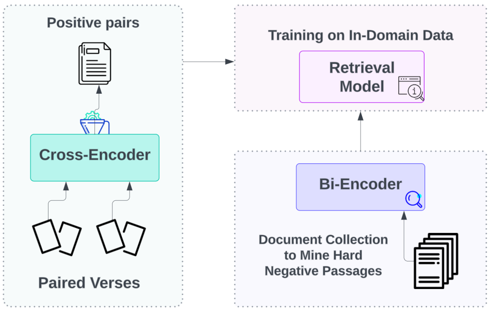

Positive pairs
Cross-Encoder
Paired Verses
Training on In-Domain Data
Retrieval Model
Bi-Encoder
Document Collection to Mine Hard Negative Passages

Figure 1: Data augmentation technique that leverages correlation of Qur’anic verses for training retrieval models in-domain.

We experimented with Arabic and English languages. Arabic, more precisely one of its variants, Classical Arabic (CA), is the language of the Holy Qur’an and is an integral component in tackling retrieval task using sacred scripture (Bashir et al., 2023). English is another popular language used for search in various domains, including the Islamic domain. Addressing the problem using Arabic and English allows for comparing the solutions and sharing insights across languages. English is a high-resource language with a great choice of corpora and pre-trained LMs for diverse domains. At the same time, depending on the domain, Arabic can be considered a low- or medium-resource language (Xue et al., 2021; Abboud et al., 2022). However, Arabic is in more favorable conditions than English in the Islamic domain; in the case of the Arabic language, there are Islamic corpora like OpenITI (Romanov and Seydi, 2019) and domain-specific LMs (Malhas and Elsayed, 2022; Inoue et al., 2021).

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 76–88
December 7, 2023 ©2023 Association for Computational Linguistics

From this perspective, addressing Qur'anic IR in English is more challenging as it requires a number of additional preparations, like preparing an Islamic corpus and pre-training domain-specific LM. This state of affairs with the English language in the Islamic domain necessitates addressing it alongside the Arabic language. Simultaneously, another advantage of handling the problem in English is the abundance of datasets to train for a general domain. Training on general domain data can be a required step to prepare a retrieval model that needs a substantial amount of data for training, where in-domain data is scarce. Experimenting with Qur'anic IR in English will allow us to learn what works best and apply these approaches to Arabic, where general domain data is insufficient.

Our main contributions are:

- We introduce an Islamic corpus and a new language model for the Islamic domain in English.

- We conduct comprehensive experiments with different retrieval models to see what works best for efficient retrieval from the Holy Qur'an in Arabic and English.

- We propose a data-augmentation technique that helped to improve the retrieval models' performance for both languages and set a new state-of-the-art in Qur'anic IR.

The rest of the work is organized as follows: we start with addressing the problem of Qur'anic IR in English. We prepare the Islamic corpus and domain-specific LM (Section 2). Section 3 applies to both languages, English and Arabic, including metrics choice, datasets for training and testing, experimental details, and training procedure of the retrieval models. Section 4 is dedicated to Qur'anic IR in Arabic. Apart from applying successful experiments that worked well with Qur'anic IR in English, we executed more methods of preparing retrieval models for Arabic language, including teacher-student distillation and employing machine translation. Model comparison and Final analysis is performed in Section 5. The prior work done in the field is highlighted in Section 6.

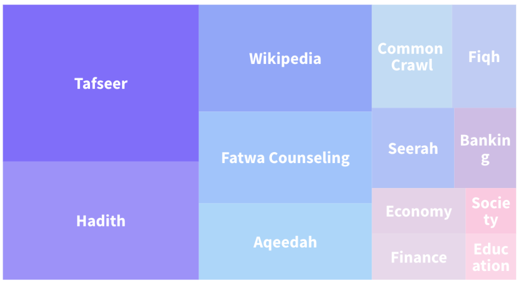


Figure 2: Types of Islamic text that constitute Islamic Corpus.

2 Domain-Specific Language Model as a Backbone of In-Domain IR

2.1 Islamic Corpus in English

Preparing an Islamic Corpus in English is challenging due to the insufficient amount of Islamic Text that is either translated from Arabic or other languages to English or initially written in English. We collect text available online of the following types (see Figure 2):

Islamic literature. These are Islamic books written by Islamic scholars about Tafseer (Qur'an exegesis), Hadith, Seerah, Fiqh (Islamic jurisprudence), and Aqeedah (Islamic creed) (approx. 28M words).

Islamic journals. Journals that are available online and focus on discussing modern issues of Islamic banking, Finance, Economy, and Islamic Education (approx. 5.5 M words).

Fatwa counseling. Fatwas that are available online from Fatwa centers (approx. 4.8M words)

Wikipedia. Articles related to Islam from the Wikipedia Islam portal (approx. 5.6M words).

Common Crawl. We search for keywords and collect files from Common Crawl on Islamic topics. We perform additional filtering and preprocessing of these articles (approx. 2.5 M words).

The total amount of words in the corpus is around 47M words.

2.2 Adaptation of General Domain Language Model for Islamic Domain

Pre-training starting from the existing checkpoint of the model pre-trained for the general domain helps reduce pre-training time (Gururangan et al., 2020; Bommasani et al., 2022; Guo and Yu, 2022). To account for the small size of the pre-training corpus and perform domain adaptation effectively, we continue pre-training the BERT model on the

Islamic corpus. To address the issue of the absence of domain-specific vocabulary during continued pre-training, we trained the WordPiece tokenizer (Song et al., 2021) on the Islamic corpus. We find the intersection between Islamic vocabulary and bert-base-uncased [1], and for the tokens inside this intersection, we assign the weights from bert-base-uncased. For the tokens outside of the intersection (Islamic tokens), we perform contextualized weight distillation following (Pavlova and Makhlouf, 2023) [2].

- In the first step, we find tokens of interest and extract corresponding sentences from the Islamic Corpus. We sample from one to twenty sentences (Bommasani et al., 2020).

- In the second step, we tokenize these sentences using a bert-base-uncased tokenizer. In that case, Islamic tokens are broken into subtokens because they are absent from bert-base-uncased vocabulary. We average the contextualized weights of the corresponding subtokens that the BERT model produces ($t_{distilled}$) and then compute aggregated representation across sentences ($t_{aggregated}$) for a corresponding token of interest from Islamic vocabulary:

$$\begin{aligned} t_{distilled} &= f(t_1, ..., t_k) \\ f &\in \{mean\} \end{aligned} \tag{1}$$

  Where *k* is the number of the subtokens that make up the token of interest.

$$\begin{aligned} t_{aggregated} &= g(t_{distilled}, ..., t_m) \\ g &\in \{mean\} \end{aligned} \tag{2}$$

  And *m* is the number of sentences involved in aggregated representation.

In order to avoid overinflating the vocabulary with new tokens, which would require longer pre-training and be prohibitive in the case of a small corpus, we analyze the frequency of each token in our corpus. Tokens with a count below threshold are filtered out, resulting in 3992 new domain-specific tokens. Moreover, we remove [unused] tokens from the bert-base-uncased vocabulary and add Islamic tokens, resulting in 33511 total tokens in the BPIT model's final vocabulary (BPIT is the abbreviation for BERT Pre-trained on Islamic Text).

2.3 Pre-training Set-up

In order to accommodate the limited size of the pre-training corpus, we schedule two-stage pre-training akin to phases of Curriculum learning (Bengio et al., 2009; Soviany et al., 2022). In the first stage, we start with an easier task of predicting masked tokens/subtokens, with a masking rate of 0.15 and using the "80-10-10" corruption rule (Devlin et al., 2019; Wettig et al., 2023). In the second stage, we increase the complexity of the prediction task by switching to predicting the whole words with the same masking rate and using the corruption rule. It is harder for a language model to predict whole words than to predict tokens or subtokens that might make up the word and give the LM more clues and make the prediction task less challenging (Cui et al., 2021; Dai et al., 2022; Gu et al., 2021). This pre-training approach introduces the LM to a broader scope of language experience and helps to gain more diversified knowledge of textual input (Mitchell, 1997), which is crucial in the case of a small corpus that we use. Pre-training hyperparameters can be found in Appendix A.

3 Preparing Neural IR Model to Retrieve from the Holy Qur'an

3.1 Dataset for Testing Retrieval Models

To test our models, we converted the QRCD (Qur'anic Reading Comprehension Dataset) (Malhas and Elsayed, 2022) to the IR dataset. We use both train and development split as test data. We do not include no-answer questions (Malhas and Elsayed, 2020), which results in 169 queries in total for testing. Queries are accompanied by the corresponding verses from the Holy Qur'an. Each Qur'anic verse is treated as the basic retrieval unit because it presents a more challenging task (see Section 5) and has higher utilization factors. The original dataset is in Arabic and was constructed and annotated by experts in Islamic studies. For our purposes of testing IR systems, we translate queries to English and verify the validity and accuracy of the translation with Islamic scholars. We use the Saheeh International[3] translation of the Holy Qur'an to express specific Qur'anic terms used in query formulation. To retrieve answers, we use the same Sahih International translation as a retrieval collection.

[1] https://huggingface.co/bert-base-uncased

[2] https://github.com/rttl-ai/BIOptimus

[3] https://tanzil.net/trans/

3.2 Metrics

Due to the complexity of the language of the Holy Qur'an and the fact that some meanings can be expressed indirectly, the retrieval task using the Holy Qur'an is quite difficult. Therefore, using several metrics to estimate the retrieval model's effectiveness from a different perspective makes sense. We use the MRR@10 (Mean Reciprocal Rate), the official evaluation metric of the MS MARCO dataset (Bajaj et al., 2018) that we extensively use to fine-tune our retrieval models. Furthermore, we add NDCG@5 (Normalized Discounted Cumulative Gain) and Recall@100, used in the BEIR benchmark (Thakur et al., 2021b). This combination of metrics lets us estimate our models with a decision support metric such as Recall, binary rank-aware metrics such as MRR, and metric with a graded relevance judgment such as NDCG (Wang et al., 2013). For evaluation, we use the BEIR framework[4] that utilizes the Python interface of the TREC evaluation tool (Gysel and de Rijke, 2018).

3.3 Baselines

BM25 is a commonly used baseline to compare retrieval systems. It is a sparse lexical retrieval method based on token-matching and uses TF-IDF weights. Though the lexical approaches suffer from the lexical gap (Berger et al., 2000) due to the constraints of retrieving the documents containing exact keywords presented in a query, BM25 remains a strong baseline (Kamalloo et al., 2023). We also include a dense neural retrieval model, trained using a sentence-transformers framework (Reimers and Gurevych, 2019) referred to as SBERT- GD (general domain), a late-interaction model ColBERT (Khattab and Zaharia, 2020) (ColBERT-GD), and Cross-Encoder-GD. All models were trained on the MS MARCO dataset from the bert-base-uncased checkpoint. This approach allows us to evaluate their performance in a zero-shot setting for the Islamic domain and compare them with the retrieval models trained using the domain-specific BPIT model. More details on how SBERT-GD, ColBERT-GD, and Cross-encoder-GD were trained are presented in Section 3.4; hyperparameters details are listed in Appendix A.

3.4 Training a Domain-specific Model on General Domain Data

To prepare the domain-specific model for the IR task, we prepare and compare three approaches.

SBERT-BPIT. We use the sentence-transformers framework, which employs a Siamese network (Bromley et al., 1993) that enables semantic similarity search. We train our BPIT model using the architecture above on the MS MARCO dataset that contains 533k training examples (more details on MS MARCO dataset are in Section 4.4), utilizing Multiple Negative Ranking Loss (MNRL) (Henderson et al., 2017; Ma et al., 2021; van den Oord et al., 2019). MNRL is a cross-entropy loss that treats relevant pairs $\{x^{(i)}, y^{(i)}\}_{i=1}^{M}$ (where M is batch size) as positive labels and other in-batch examples as negative, and formally defined as:

$$J_{\mathrm{MNRL}}(\theta) = \frac{1}{M}\sum_{i=1}^{M}\log\frac{\exp\sigma(f_\theta(x^{(i)}), f_\theta(y^{(i)}))}{\sum_{j=1}^{M}\exp\sigma(f_\theta(x^{(i)}), f_\theta(y^{(j)}))}$$

where σ is a similarity function, in our case it is a cosine similarity function; f_θ is the sentence encoder. We use multiple hard negatives; these are negative passages similar to the positive passage but not relevant to the query and mined using cross-encoder scores [5].

Cross-encoder-BPIT. In the case of a cross-encoder, a pair of sentences are simultaneously fed into a transformer-like model, and attention is applied across all tokens to produce a similarity score (Humeau et al., 2020). This approach does not allow end-to-end information retrieval and endure extreme computational overhead. However, in many IR tasks, it performs superior to other methods and can be used for mining hard negatives, data augmentation (Section 3.5), and reranking. The model is trained with triples provided by MS MARCO starting from the BPIT checkpoint under a classification task, using Cross Entropy Loss.

ColBERT-BPIT. ColBERT computes embeddings independently for queries and documents and, at the same time, can also register more fine-grained interactions between tokens. Using the same mined hard negatives constructed for the MS MARCO dataset used to pre-train SBERT-BPIT, ColBERT-BPIT is trained starting from the BPIT

[4] https://github.com/beir-cellar/beir/tree/main

[5] https://huggingface.co/datasets/sentence-transformers/msmarco-hard-negatives

checkpoint by optimizing the cross-entropy loss applied to the score of the query and the positive passage against in-batch negatives (Santhanam et al., 2022).

All models with the prefix **BPIT** are counterparts of **GD** models; for a fair comparison, they are trained using the same dataset, objective function, and hyperparameters (see Appendix A) with the only difference that **BPIT** models initialized from the BPIT checkpoint and **GD** models initialized with the bert-base-uncased checkpoint.

3.5 In-domain Training of the Domain-specific Model

The performance of dense retrieval systems worsens when encountering a domain shift (Thakur et al., 2021b); therefore, there is a great benefit in training neural IR models on in-domain data. The lack of domain-specific data is often solved by augmenting training data: generating synthetic data (dos Santos Tanaka and Aranha, 2019), paraphrasing using synonyms (Wei and Zou, 2019), sampling and recombining new training pairs (Thakur et al., 2021a), round-trip translation (Yu et al., 2018; Xie et al., 2020) or involving denoising autoencoders (Wang et al., 2021). These techniques involve data distortion, which is suboptimal when dealing with religious and heritage datasets. We propose a data generation technique for in-domain training advantageous for the retrieval task involving the text of the Holy Qur'an (see Figure 1). Understanding the text of the Holy Qur'an is closely related to the meaning explained in the books of Tafseer written by Islamic Scholars. Tafseer Ibn Kathir, one of the established books of Qur'an exegesis, contains ample verse relations references. Putting this into use allows not only to perform data augmentation but also to intertwine more meaning to Qur'anic verses that need to be more explicit for a LM to learn directly from the text of the verse.

Pairing. Let C_t denote a collection of books of Tafseer by Ibn Kathir. We start with extracting and paring all verse relations mentioned in Tafseer Ibn Kathir. That gives us V_t that contains distinct pairs $\{v_q, v_p\} \in V_t$ and $|V_t| \sim 11$k pairs.

Filtering. Not all the pairs can be used for training the retrieval model because not all the verse relation pairs will be interpreted by the model as a signal of positive correlation due to meanings that are expressed indirectly. We use the cross-encoder model M_{ce} that was trained on a general domain to score ayah pairs $s_{ce} = M_C(v_q, v_p)$. We filter out the pairs that were scored below the threshold, leaving us with V_f that contains pairs with strong positive correlations $(q, p^+) \in V_f$ and $|V_f| = 2352$ pairs.

Sampaling hard negatives. To prepare negative passages, we use the text of the Tafseer Ibn Kathir without verses' quotations. The text is split into M passages to form a collection $C^- = \{p_1^-, p_2^-, ..., p_m^-\}$ to sample negative passages. Sampling from the Holy Qur'an text is a less favorable approach. Due to the relatively small size of the Qur'anic corpus, mined negative passages may turn out to be false negatives (Qu et al., 2021). At the same time, sampling from another corpus would create easy negatives that are not beneficial for training (Ren et al., 2021; Karpukhin et al., 2020; Xiong et al., 2021), while the text of the Tafseer Ibn Kathir contains passages that are similar to the positive passages but not precisely relevant to q and are good candidates to play a role of hard negatives. To choose hard negatives, we use a retrieval model trained with a Seamise network M_B and retrieve negative passages $(p_1^-, ..., p_i^-)$ related to $\forall q \in V_f$. We score each pair (q, p^-) with the cross-encoder $s_{ce} = M_C(q, p^-)$, and use these scores in the next stage of training.

Continue training in-domain. We combine the collection of verses from the Holy Quran C^+ and the collection of passages from Tafseer Ibn Kathir C^- into one collection C_{aug} for training, which together with selected positive pairs and mined hard negatives forms new augmented dataset D_{IN} for in-domain training. We continue training SBERT-BPIT and ColBERT-BPIT on new in-domain data following the same procedure described for each model in Section 3.4. The models that come out as a result of this stage of training are SBERT-ID (Islamic Domain) and ColBERT-ID.

3.6 Results and Discussion

The performance of all models on the test dataset is collected in Table 1. All the models steadily outperform the BM25 baseline on every metric. In the category of the **GD** and **BPIT** models, the best-performing model is **ColBERT** for all metrics. In contrast, in the category of **ID** models, **SBERT** shows the best results at **MRR@10**, with a considerable improvement in performance after in-domain training on the augmented dataset (increasing from **0.48** to **0.55**).

	Recall@100	MRR @10	NDCG@5
BM25	0.15	0.27	0.15
SBERT-GD	0.2	0.43	0.23
ColBERT-GD	0.25	0.43	0.26
Cross-encoder-GD	0.19	0.37	0.22
SBERT-BPIT	0.28	0.48	0.28
ColBERT-BPIT	0.32	0.51	0.32
Cross-encoder-BPIT	0.17	0.3	0.16
SBERT-ID	0.32	**0.55**	**0.33**
ColBERT-ID	**0.33**	0.53	**0.33**

Table 1: Performance of retrieval models on the test data (English).

	Recall@100	MRR @10	NDCG@5
SBERT-ID (Saheeh Int.)	0.32	**0.55**	**0.33**
SBERT-ID (Yusuf Ali)	0.31	0.49	0.3
SBERT-ID (al-Hilali)	**0.33**	0.5	0.31
SBERT-ID (Pickthall)	0.29	0.48	0.29
ColBERT-ID (Saheeh Int.)	**0.33**	0.53	**0.33**
ColBERT-ID (Yusuf Ali)	0.28	0.46	0.27
ColBERT-ID (al-Hilali)	0.25	0.5	0.3
ColBERT-ID (Pickthall)	0.27	0.47	0.28

Table 2: Comparison of the performance of the retrieval models on the test data for different translations of the Holy Qur'an into English.

Overall, all **ID** models demonstrate superior performance, proving that training in-domain using our data augmentation technique was beneficial. Moreover, another important observation is consistent progress for **SBERT** and **ColBERT** models when training using the domain-specific model (**BPIT**) coupled with training on in-domain data. We suppose that leveraging domain adaptation of a LM that serves as a backbone for retrieval models and subsequent training of retrieval models on large general domain data before training on in-domain data is an effective approach.

In Table 2, we included a comparison and analysis of the performance of the retrieval models for different translations of the Holy Qur'an into English. We can see no significant degradation of the models' performance. The formulation of the queries contains terms from Saheeh International translation (Section 3.1), which proves that the models can maintain search relevancy with different semantics. With these results and insights, we switch to exploring how to tackle IR tasks for the Holy Quran in the Arabic Language.

4 Preparing a Retrieval Model to Extract Relevant Verses from the Holy Qur'an in Arabic

This section discusses how to address the same problem of designing an efficient neural IR model for extracting relevant verses from the Holy Qur'an in Arabic. Though the goal is essentially the same, the resources to achieve it are quite different in the case of the Arabic Language. The dataset for testing is the same as the one described in Section 3.1. We use the queries as they were initially formulated in Arabic by the authors of QRCD (Malhas and Elsayed, 2022). For the choice of the metrics, refer to Section 3.2.

4.1 Choice of Arabic LM to Tackle IR Task in the Islamic Domain

Due to a lack of manually crafted linguistic resources, Arabic is considered a low- or medium-resource language, depending on the domain of application (Xue et al., 2021; Abboud et al., 2022). Recent advances in Arabic NLP have brought a number of LMs pre-trained on Arabic corpora and new datasets translated into Arabic or initially curated in Arabic. Arabic is the language of the Holy Qur'an and the source language of numerous Islamic scholarly works. Moreover, the multi-institutional initiative has offered the Arabic NLP community an Open Islamicate Texts Initiative OpenITI (Romanov and Seydi, 2019), an excellent source for pre-training a LM for the Islamic domain. These advantageous conditions for the Islamic domain in Arabic let us skip the preliminary stage of corpus preparation and LM pre-training.

However, there is a benefit in comparing how various Arabic LMs can fit as the backbone of the IR system for the Islamic domain. Table 3 compares Arabic LMs' efficiency in tackling IR task in the Islamic domain out-of-the-box. We use a sentence-transformers framework to compare LMs to avoid a costly training stage. We add an averaging pooling layer on top of BERT embeddings and convert it into a fixed-sized sentence embedding (Reimers and Gurevych, 2019). The same model is utilized to create sentence embeddings for both queries and Qur'anic verses, and then answers to the query are found using the cosine similarity measure. The models are not ready to efficiently handle IR tasks without additional training, yet this approach let us to compare LMs' embeddings out-of-the-box. We include in the comparison the bert-base-uncased model and the BPIT model (evaluation is run on the English translation of QRCD).

As we can see from the table, most of the models perform poorly. We can also observe that pre-training on large amounts of data does not neces-

	Number of tokens/ Domain	MRR@10	NDCG@5
bert-base-arabic-camelbert-mix (Inoue et al., 2021)	17.3B/GD	0.01	0.01
bert-base-arabic-camelbert-ca (Inoue et al., 2021)	847M/ID	0.01	0.01
bert-base-arabertv02 (Antoun et al., 2020)	8.6B/GD	0.01	0.01
bert-base-arabic (Safaya et al., 2020)	8.2B/GD	0.06	0.02
bert-base-uncased (Devlin et al., 2019)	3.3B/GD	0.07	0.03
CL-AraBERT (Malhas and Elsayed, 2022)	2.7B+1.05B/GD+ID	**0.11**	**0.06**
BPIT	3.3B+50M/GD+ID	**0.11**	**0.06**

Table 3: Performance of LMs on the test dataset. GD stands for General domain and ID for Islamic domain.

	Recall@100	MRR @10	NDCG@5
Bilingual-distilled	0.12	0.26	0.15
SBERT-AR-NLI	0.21	0.38	0.21
SBERT-AR-MARCO	0.23	0.4	0.23
ColBERT-AR	0.28	0.47	**0.29**
SBERT-AR-ID	0.25	0.45	0.27
ColBERT-AR-ID	**0.29**	**0.48**	**0.29**

Table 4: Performance of retrieval models on the test dataset (Arabic).

sarily lead to better performance in IR task. CL-AraBERT performs significantly better than other Arabic LMs, and its performance is similar to the BPIT model. It is plausible that, as in the case of CL-AraBERT (Malhas and Elsayed, 2022) and the BPIT model, pre-training in a continued approach on a domain-specific corpus with specialized vocabulary starting from the general domain checkpoint helps to tackle IR task in the Islamic domain more efficiently. Another noteworthy observation is that the BPIT model exhibits this performance while pre-trained for a short period and with a small corpus of less than 50M tokens. We assume that contextualized weight distillation might help boost the efficiency during the pre-training stage. The second best performing models are bert-base-uncased and bert-base-arabic. Based on the result of Table 3, we choose CL-AraBERT as a backbone model to conduct subsequent experiments with IR task in Islamic Domain in Arabic.

4.2 Knowledge Distillation Approach to Improve Performance of Arabic LM in IR Task

The lack of manually crafted linguistic resources in low-resource languages can be tackled by knowledge distillation. Reimers and Gurevych (2020) showed that it is possible to improve the performance of sentence embedding models by mimicking the performance of a stronger model. They used parallel corpora to teach the student model to produce sentence embeddings close to the embeddings of the teacher model. Their experiment uses the English SBERT model to initialize the teacher model, and multilingual XLM-RoBERTa (Conneau et al., 2020) is used as a student model. Our experiment uses the SBERT-BPIT (Section 3.4) as a teacher model and the bilingual EN-AR student model. The student model combines the embedding matrix of the CL-AraBERT for Arabic tokens and the BPIT model for English tokens, and the encoder weights are borrowed from the BPIT model. We use a combination of parallel datasets (EN-AR) available on the OPUS website (Tiedemann, 2012): TED2020, NewsCommentary, WikiMatrix, Tatoeba, and Tanzil, total size of training data is around 1.1M sentences (for hyperparameters details, see Appendix A). Table 4 presents the evaluation results of this approach on the test dataset (Bilingual-distilled-EN-AR model). We can see a significant improvement compared to the results of CL-AraBERT from Table 3, yet the performance is practically twice lower than the performance of the equivalent English model (SBERT-BPIT, Table 1).

4.3 Training on Arabic Natural Language Inference Dataset to Improve Sentence Embeddings

Another approach that can help to improve the quality of the sentence embeddings is training on the Natural Language Inference (NLI) dataset (Reimers and Gurevych, 2019; Bowman et al., 2015; Williams et al., 2018) . Conneau et al. (2018) introduced Cross-lingual Natural Language Inference (XNLI) comprising 7500 examples for development and test sets translated into 15 languages, including Arabic. We train CL-AraBERT on XNLI following Reimers and Gurevych (2019), using 400k machine-translated training examples that accompany XNLI development and test set (more details in Appendix A). The performance of this model (SBERT-AR-NLI, Table 4) is better than Bilingual-distilled-EN-AR, yet lower than SBERT-BPIT (Table 1).

4.4 Employing Machine-Translated Datasets to Overcome The Lack of Large Training Data

Although the quality of the machine-translated dataset is inferior to human translation, the accessibility of machine-translated text helps to generate a considerable training set which is essential for preparing a retrieval model. The experiment with training on the XNLI dataset from section 4.3 showed that training on a machine-translated dataset can achieve competitive performance. This motivates us to extend this experiment further to the MS MARCO dataset. MS MARCO is a large collection of datasets focused on deep learning in search (Bajaj et al., 2018), including the IR dataset that comprises more than half a million queries and is accompanied by a collection of 8.8M passages and 39M triplets for training. Another advantage of using MS MARCO, besides a sizable training set, is that it is more suitable for training IR systems, and we can experiment with both SBERT and ColBERT approaches to prepare retrieval models and compare their performance across languages. Bonifacio et al. (2022) presented a multilingual version of the MS MARCO dataset created using machine translation comprising 13 languages. We use the Arabic translation of MS MARCO and train SBERT-AR-MARCO equivalently to SBERT-BPIT and ColBERT-AR following the training procedure of ColBERT-BPIT (Section 3.4). Table 4 demonstrates that training on MS MARCO can give better results compared to other training approaches described in Sections 4.2 and 4.3.

4.5 In-domain Training of Retrieval Model for Qur'anic IR in Arabic

In the last stage, we perform training on in-domain data and repeat the successful experiment with dataset augmentation in English. The steps to augment dataset are the same (see Section 3.5). We use a cross-encoder trained on machine-translated MS MARCO to score ayah pairs, which results in a slightly different count of selected pairs (2723). We continue training SBERT-AR-MARCO and ColBERT-AR on in-domain data and produce SBERT-AR-ID and ColBERT-AR-ID.

The performance of these retrieval models is included in Table 4, and we can observe further improvement after training on in-domain data. The best-performing model is **ColBERT-AR-ID**, and it is plausible that the retrieval approach of the ColBERT model that leverages more fine-grained interactions between a query and a verse (Khattab and Zaharia, 2020) is especially advantageous for languages with complex morphological structures, such as Arabic.

	Recall@100	MRR @10	NDCG@5
SBERT-AR-ID	0.25	0.45	0.27
ColBERT-AR-ID	0.29	0.48	0.29
SBERT-AR-ID (passages)	0.7	0.47	0.35
ColBERT-AR-ID (passages)	**0.77**	**0.53**	**0.43**

Table 5: Performance of Arabic retrieval models on the passage retrieval task (Arabic).

5 Model comparison and Final analysis

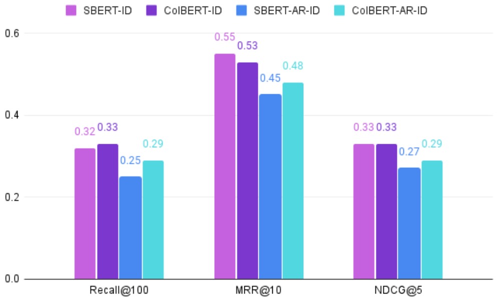


Figure 3: Comparison of the retrieval models for the Islamic domain (ID) for English and Arabic across all metrics.

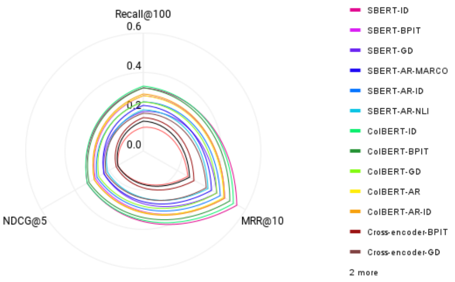


Figure 4: Comparison of the retrieval models for the Islamic domain (ID) for English and Arabic across all metrics.

Figure 3 compares all the retrieval models for the Islamic domain (ID) for English and Arabic across all metrics. A noteworthy observation is that all English retrieval models outperform their Arabic equivalents, which can be explained by the complexity of the Arabic language and the usage of machine-translated data. Nevertheless, the results of Arabic retrieval models are not far apart from English models, and specifically, with the employment of the ColBERT model, we can see a competitive performance (**0.48** for MRR@10 and **0.29** for Recall@100 and NDCG@5).

The radar chart (Figure 4) shows a more comprehensive comparison across all models. We can see that the radar chart has a tapered shape overall, with an MRR@10 axis being the most prolonged edge, indicating that all models show the best results for this metric. Moreover, NDCG@5 and Recall@100 are more proportionally placed against each other, signifying that the performance for these metrics is similar across all the models. SBERT-ID and ColBERT-ID (magenta and green colors) are located at the edge, showing the best performance. They are followed by ColBERT-BPIT and SBERT-BPIT (English models), and Arabic ColBERT and SBERT models are located in the middle of the chart. In the center, we can see BM25 and the Bilingual-distilled model, these are models with the lowest performance.

In addition, we conducted tests on two models, ColBERT-AR-ID and SBERT-AR-ID (as shown in Table 5), for the passage retrieval task (Malhas, 2023). We did not apply any passage or query expansion heuristics (Malhas, 2023). Our findings indicate that this approach is less challenging and increases the MRR@10 score, especially for the ColBERT model. The NDCG@10 score grows by 0.08 for the SBERT model and by 0.14 for the ColBERT model. Moreover, the Recall@100 grows by almost threefold.

6 Related work

Thakur et al. (2021a) proposed a data augmentation technique to train sentence transformers when little data for in-domain training is available. Wang et al. (2021) and Wang et al. (2022) experimented with domain adaptation techniques for embedding models.

The topic of the choice of hard negatives is discussed in works of: Qu et al. (2021), Ren et al. (2021), Karpukhin et al. (2020), Xiong et al., 2021.

Bashir et al. (2023) wrote a detailed overview of the state of Qur'anic NLP, including the present state of search and retrieval technologies. Most of the approaches described use keywords-based or ontology-driven search. A few works employ semantic search based on deep-learning methods: Alshammeri et al. (2021) use doc2vec; Mohamed and Shokry (2022) utilize word2vec. Malhas and Elsayed (2022) pre-trained CL-Arabert on OpenITI (Romanov and Seydi, 2019) starting from the AraBERT checkpoint (Antoun et al., 2020). They also introduced the first Qur'anic Reading Comprehension Dataset (QRCD) that we used as a test data for the Qur'anic IR task.

7 Conclusion

In this paper, we employed state-of-the-art approaches in IR to analyze and compare what works better to improve Qur'anic IR in English and Arabic. The results show that retrieval models in English outperform their Arabic equivalents. The inherent linguistic complexity of the Arabic language may explain this performance gap; nevertheless, transferring successful experiments from English to Arabic, applying large machine-translated datasets, and using the proposed data-augmentation technique helped to enhance the results in Qur'anic IR in Arabic.

One of the possible directions to take in the future is to extend this work to encompass more languages. This would broaden the scope of the semantic search for the Holy Qur'an, making it accessible to a larger audience. Moreover, research conducted in a multilingual environment helps to exchange insights among languages and enhance the results in Qur'anic IR.

Another essential step is to extensively evaluate real-world user queries to analyze models' performance in practice [6].

Acknowledgment

This research would not have been possible without my colleague at rttl.ai, Mohammed Makhlouf, who gathered data for training the language and retrieval models. Additionally, I would like to thank Dr. Rana Malhas and Dr. Tamer Elsayed from Qatar University for helping to verify the quality of the English translation of the queries of the QRCD and anonymous reviewers for their valuable feedback.

Limitations

One of the main limitations of our paper is the quality of machine-translated datasets, such as XNLI train set (Conneau et al., 2020) and mMARCO (Bonifacio et al., 2022) translation into Arabic. Using machine translation is a solution to address a lack of data for training models for low or medium-resource languages like Arabic. The quality of

[6] A live testing system is deployed at `rttl.ai`

automated translation is constantly improving and has reached a good quality recently; nevertheless, it is not yet equivalent to the high quality of human translation done by experts in the field.

Ethics Statement

We do not anticipate any considerable risks associated with our work. The data and other related resources in this work are publically available, and no private data is involved. We respect previous work done in the field and appropriately cite the methods and datasets we are using. To prevent misuse of pre-trained models, we carefully consider applications and provide access upon request. [7]

References

Khadige Abboud, Olga Golovneva, and Christopher DiPersio. 2022. Cross-lingual transfer for low-resource Arabic language understanding. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 225–237, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Menwa Alshammeri, Eric Atwell, and Mhd ammar Alsalka. 2021. Detecting semantic-based similarity between verses of the quran with doc2vec. *Procedia Computer Science*, 189:351–358.

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.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2018. Ms marco: A human generated machine reading comprehension dataset.

Muhammad Huzaifa Bashir, Aqil M Azmi, Haq Nawaz, Wajdi Zaghouani, Mona Diab, Ala Al-Fuqaha, and Junaid Qadir. 2023. Arabic natural language processing for qur'anic research: A systematic review. *Artificial Intelligence Review*, 56(7):6801–6854.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 41–48, New York, NY, USA. Association for Computing Machinery.

Adam L. Berger, Rich Caruana, David A. Cohn, Dayne Freitag, and Vibhu Mittal. 2000. Bridging the lexical chasm: statistical approaches to answer-finding. In *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Rishi Bommasani, Kelly Davis, and Claire Cardie. 2020. Interpreting Pretrained Contextualized Representations via Reductions to Static Embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4758–4781, Online. Association for Computational Linguistics.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. 2022. On the opportunities and risks of foundation models.

Luiz Bonifacio, Vitor Jeronymo, Hugo Queiroz Abonizio, Israel Campiotti, Marzieh Fadaee, Roberto Lotufo, and Rodrigo Nogueira. 2022. mmarco: A multilingual version of the ms marco passage ranking dataset.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

[7] Contact us at hello@rttl.ai

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1993. Signature verification using a "siamese" time delay neural network. In *Proceedings of the 6th International Conference on Neural Information Processing Systems*, NIPS'93, page 737–744, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

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.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, and Ziqing Yang. 2021. Pre-training with whole word masking for chinese bert. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3504–3514.

Yong Dai, Linyang Li, Cong Zhou, Zhangyin Feng, Enbo Zhao, Xipeng Qiu, Piji Li, and Duyu Tang. 2022. “is whole word masking always better for Chinese BERT?”: Probing on Chinese grammatical error correction. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1–8, Dublin, Ireland. 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.

Fabio Henrique Kiyoiti dos Santos Tanaka and Claus Aranha. 2019. Data augmentation using gans.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Trans. Comput. Healthcare*, 3(1).

Xu Guo and Han Yu. 2022. On the domain adaptation and generalization of pretrained language models: A survey.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Christophe Van Gysel and Maarten de Rijke. 2018. Pytrec_eval. In *The 41st International ACM SIGIR Conference on Research &amp Development in Information Retrieval*. ACM.

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.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2020. Poly-encoders: Transformer architectures and pre-training strategies for fast and accurate multi-sentence scoring.

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.

Ehsan Kamalloo, Nandan Thakur, Carlos Lassance, Xueguang Ma, Jheng-Hong Yang, and Jimmy Lin. 2023. Resources for brewing beir: Reproducible reference models and an official leaderboard.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert.

Ji Ma, Ivan Korotkov, Yinfei Yang, Keith Hall, and Ryan McDonald. 2021. Zero-shot neural passage retrieval via domain-targeted synthetic question generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1075–1088, Online. Association for Computational Linguistics.

Rana Malhas and Tamer Elsayed. 2020. Ayatec: building a reusable verse-based test collection for arabic question answering on the holy qur'an. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 19(6):1–21.

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 R Malhas. 2023. *ARABIC QUESTION ANSWERING ON THE HOLY QUR'AN*. Ph.D. thesis.

Tom M Mitchell. 1997. *Machine learning*, volume 1. McGraw-hill New York.

Ensaf Hussein Mohamed and Eyad Mohamed Shokry. 2022. Qsst: A quranic semantic search tool based on word embedding. *Journal of King Saud University - Computer and Information Sciences*, 34(3):934–945.

Vera Pavlova and Mohammed Makhlouf. 2023. BIOptimus: Pre-training an optimal biomedical language model with curriculum learning for named entity recognition. In *The 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks*, pages 337–349, Toronto, Canada. Association for Computational Linguistics.

Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2021. RocketQA: An optimized training approach to dense passage retrieval for open-domain question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5835–5847, Online. 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.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online. Association for Computational Linguistics.

Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, QiaoQiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2021. RocketQAv2: A joint training method for dense passage retrieval and passage re-ranking. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2825–2835, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Maxim Romanov and Masoumeh Seydi. 2019. Openiti: a machine-readable corpus of islamicate texts. *Zenodo, URL: https://doi. org/10.5281/zenodo*, 3082464.

Ali Safaya, Moutasem Abdullatif, and Deniz Yuret. 2020. KUISAIL at SemEval-2020 task 12: BERT-CNN for offensive speech identification in social media. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 2054–2059, Barcelona (online). International Committee for Computational Linguistics.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3715–3734, Seattle, United States. Association for Computational Linguistics.

Xinying Song, Alex Salcianu, Yang Song, Dave Dopson, and Denny Zhou. 2021. Fast WordPiece tokenization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2089–2103, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. 2022. Curriculum learning: A survey.

Nandan Thakur, Nils Reimers, Johannes Daxenberger, and Iryna Gurevych. 2021a. Augmented SBERT: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 296–310, Online. Association for Computational Linguistics.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021b. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems (NeurIPS 2021) - Datasets and Benchmarks Track (Round 2)*.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2019. Representation learning with contrastive predictive coding.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Kexin Wang, Nils Reimers, and Iryna Gurevych. 2021. TSDAE: Using transformer-based sequential denoising auto-encoderfor unsupervised sentence embedding learning. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 671–688, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Kexin Wang, Nandan Thakur, Nils Reimers, and Iryna Gurevych. 2022. GPL: Generative pseudo labeling for unsupervised domain adaptation of dense retrieval. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2345–2360, Seattle, United States. Association for Computational Linguistics.

Yining Wang, Liwei Wang, Yuanzhi Li, Di He, Tie-Yan Liu, and Wei Chen. 2013. A theoretical analysis of ndcg type ranking measures.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Alexander Wettig, Tianyu Gao, Zexuan Zhong, and Danqi Chen. 2023. Should you mask 15% in masked language modeling? In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2985–3000, Dubrovnik, Croatia. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V. Le. 2020. Unsupervised data augmentation for consistency training.

Wenhan Xiong, Xiang Lorraine Li, Srini Iyer, Jingfei Du, Patrick Lewis, William Yang Wang, Yashar Mehdad, Wen tau Yih, Sebastian Riedel, Douwe Kiela, and Barlas Oğuz. 2021. Answering complex open-domain questions with multi-hop dense retrieval.

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, Online. Association for Computational Linguistics.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension.

A Appendix

A.1 Hyperparameter details

Computing Infrastructure	2 x NVIDIA RTX 3090 GPU
Hyperparameter	**Assignment**
number of epochs	10
batch size	128
maximum learning rate	0.0005
learning rate optimizer	Adam
learning rate scheduler	None or Warmup linear
Weight decay	0.01
Warmup proportion	0.06
learning rate decay	linear

Table 6: Hyperparameters for continual pre-training of BPIT model.

For training SBERT and ColBERT models, we follow training recommendations implemented by the authors. To ensure fair comparison across models and languages, all the hyperparameters for SBERT models are identical, and the same applies to ColBERT models.

Computing Infrastructure	2 x NVIDIA RTX 3090 GPU
Hyperparameter	**Assignment**
number of epochs	10
batch size	64
learning rate	2e-5
pooling	mean

Table 7: Hyperparameters for training SBERT models.

Computing Infrastructure	2 x NVIDIA RTX 3090 GPU
Hyperparameter	**Assignment**
number of epochs	1
batch size	32
learning rate	1e-5

Table 8: Hyperparameters for training ColBERT models.

Computing Infrastructure	2 x NVIDIA RTX 3090 GPU
Hyperparameter	**Assignment**
number of epochs	1
batch size	32
learning rate	2e-5

Table 9: Hyperparameters for training Cross-encoders.

LANS: Large-scale Arabic News Summarization Corpus

Abdulaziz Alhamadani
Virginia Tech
Falls Church, VA, USA
hamdani@vt.edu

Xuchao Zhang
Microsoft,
Redmond, WA, USA
xuchaozhang@microsoft.com

Jianfeng He*
Virginia Tech
Falls Church, VA, USA
jianfenghe@vt.edu

Aadyant Khatri
Virginia Tech
Falls Church, VA, USA
aadyant@vt.edu

Chang-Tien Lu
Virginia Tech
Falls Church, VA, USA
ctlu@vt.edu

Abstract

Text summarization has been intensively studied in many languages, and some languages have reached advanced stages. Yet, Arabic Text Summarization (ATS) is still in its developing stages. Existing ATS datasets are either small or lack diversity. We build, LANS, a large-scale and diverse dataset for Arabic Text Summarization task. LANS offers 8.4 million articles and their summaries extracted from newspapers websites' metadata between 1999 and 2019. The high-quality and diverse summaries are written by journalists from 22 major Arab newspapers, and include an eclectic mix of at least more than 7 topics from each source. We conduct an intrinsic evaluation on LANS by both automatic and human evaluations. Human evaluation of 1,000 random samples reports 95.4% accuracy for our collected summaries, and automatic evaluation quantifies the diversity and abstractness of the summaries.

1 Introduction

Every day there is an abundant amount of text published on the internet, such as news articles, scientific papers, product reviews, and blogs. Therefore, the need for text summarization is compelling to make use of this information overload. For a summarized text, a good one should be concise and include the main information of the original text (Radev et al., 2002). For some languages like English, the field has developed rapidly and achieved competitive results(Zhang et al., 2020; Lewis et al., 2019; Dou et al., 2020). Unlike English, the field in Arabic has been slowly and fairly developing in the past few years; thus, it has not reached its advanced shape. In the field of Arabic Text Summarization (ATS) (Belkebir and Guessoum, 2015; AL-Khawaldeh and Samawi, 2015; Fejer and Omar, 2014; Abu Nada et al., 2020; El-Kassas et al., 2021), the dearth of a diverse and large summarization dataset is one of the main existing difficulties that ATS researchers encounter (Al-Saleh and Menai, 2016; Elsaid et al., 2022).

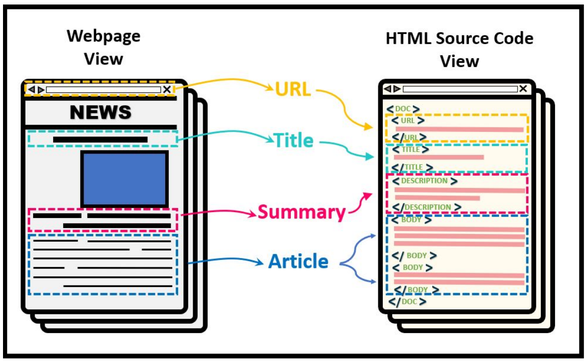


Figure 1: The Webpage view (left) shows a typical news article view. The summaries are extracted from the HTML source code view's (right) metadata (*og:description*).

Concerted efforts have been made to overcome those challenges by building various Arabic datasets for the task such that EASC (El-Haj et al., 2010), Kalimat(El-Haj and Koulali, 2013), TAC2011 (El-Ghannam and El-Shishtawy, 2014), ANT(Chouigui et al., 2021), and XL-Sum (Hasan et al., 2021), but those datasets have limitations in terms of diversity or size. Therefore, the demand for a diverse and large-scale dataset is crucial to advance the ATS field. The diversity in the ATS dataset is in twofold. The first kind of diversity exists in the Modern Standard Arabic (MSA). Even though 22 countries use MSA as an official standard language, each country has its own dialects (Dialectal Arabic) for communication. Each country's dialects have some effects on the MSA style of writing and the choice of words. For example, in a sentence describing the rounds of a soccer match, Moroccan MSA would use the word "أطوار" for "rounds" and "المواجهة" for the word "the match" while Saudi MSA would use "أشواط" and "المباراة"

[*] Corresponding author.

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 89–100
December 7, 2023 ©2023 Association for Computational Linguistics

Corpus	# of documents	MSA Diversity	Category Diversity	Human Evaluation
EASC	153	×	×	✓
KALIMAT	20,291	×	✓	×
ANT	31,798	×	✓	×
XL-Sum	40,327	×	✓	250
LANS	> 8 millions	✓	✓	1,000

Table 1: Arabic Text Summarization Datasets comparison

respectively. Second, there is diversity in news categories. Each newspaper has different news topics, such as finance, politics, sports, health, local, international news, and more. Not all ATS datasets include both diversity aspects in one dataset (see Table 1). Thus, it is essential to build a dataset that considers both types of diversity.

In terms of size, the available ATS datasets contain a range of 100 to 41,000 training samples, which make them too small to fully train a summarization model. The performance in summarization models evidently relies on a substantial amount of applicable training samples (Völske et al., 2017; Grusky et al., 2018; Zhang et al., 2020; Lewis et al., 2019; Dou et al., 2020). Thus, we expect a large-scale dataset which is provided in this work.

To overcome the current limitations in diversity and size, we introduce a new ATS dataset (**LANS**) that includes both types of diversity and large-scale to present new opportunities to ATS models and improve their summary accuracies. To achieve MSA diversity, that is the variety of each Arab country dialects' impact on its MSA, LANS encompasses 19 Arab countries and collected articles along with their summaries of 22 popular newspapers (see Table 2). For the diversity of text categories, we consider all available news categories of each source in our ATS dataset. Thus, LANS ensures both types of diversity of MSA among the Arab countries. To overcome the size limitation, LANS provides more than 8 million news articles along with their summaries. LANS's substantial amount of articles and their summaries, plus the diversity in MSA sources and categories make it a worthy resource for ATS models.

LANS exploited the metadata of newspapers' archives to extract and build the dataset. In Figure. 1, a high-level example is shown to demonstrate where the collected information originated from two parallel views: the webpage view and its HTML source code view. The webpage view shows what a reader sees when reading a news article: the URL, title, bold part or the abstract sentence/s, and article bodies. LANS pursues the metadata attributes from the HTML source code - specifically (*og:description*) to extract the summaries from. In the webpage view, the summaries lie either in bold text or before the article's paragraphs. In the HTML source code view, the summaries lie in the metadata attributes, in our case between (*og:description*) tags, which we extracted as the news articles' summaries. After the extraction, we cleaned and filtered 11M news articles to present 8.4M articles along with their summaries.

To quantify the quality of the collected summaries and examine their summarization properties, we conducted an automatic evaluation based on 3 common metrics. Moreover, we corroborated the evaluation with human evaluation of 1,000 samples to verify the accuracy of using the abstract from the HTML source code's metadata as a summary. The human evaluation verifies that using the summary available in the metadata has a 95.4% accuracy. Considering the large size of LANS, 8.4 million, LANS can benefit the ATS field, because large datasets improve NLP tasks, such as numerous training samples for pre-trained models (Zhang et al., 2020; Lewis et al., 2019). Besides, both types of diversities create opportunities for researchers to construct more accurate ATS models.

Our main contributions are as follows: (1) We curate LANS, a large-scale ATS dataset of 8.4 million Arabic news articles paired with their summaries written by journalists between 1999 to 2019. To our knowledge, it is the largest to date. (2) LANS is collected from 22 reputable Arab newspapers to achieve high quality of diversity in MSA, and for each source, there are at least 7 topics to achieve diversity in categories. (3) To quantify the intrinsic quality of LANS, a human evaluation is conducted on 1,000 random samples and verifies 95.4% accuracy of the summaries. Plus, the automatic evaluation on the whole dataset quantifies the abstractness and properties of the summaries.

ID	Newspaper	Country	From	Articles	ID	Newspaper	Country	From	Articles
1	Elkhabar	Algeria	2014	78201	12	Hespress	Morroco	2007	91357
2	Alwasat	Bahrain	2013	23860	13	Alwatan	Oman	2014	130067
3	Gate Ahram	Egypt	2016	315655	14	Alquds	Palestine	2015	88313
4	Youm7	Egypt	2008	2039818	15	Alquds-UK	Palestine	2013	349439
5	Albayan	Emirates	1999	1137188	16	Alwatan	Qatar	2016	214405
6	Almadapaper	Iraq	2009	105925	17	Aljazira	Saudi Arabia	2001	809445
7	Aldustoor	Jordan	2000	601372	18	Alryiadh	Saudi Arabia	2004	1004893
8	Annahar	Kuwait	2007	575482	19	Alsudan Alyoom	Sudan	2016	104439
9	Alakhbar	Lebanon	2006	222215	20	Zamanalwsl	Syria	2007	128785
10	WAL	Libya	2013	141898	21	Alssabah	Tunisia	2011	166137
11	Sahara Media	Mauritania	2009	11982	22	Almasdar	Yemen	2009	102608
								Total	8,443,484

Table 2: Overall statistics of the collected articles

2 Related Work (Existing Datasets)

To the best of our knowledge, Lakhas (Douzidia and Lapalme, 2004) is considered one of the early works to build an ATS model. Due to the lack of ATS datasets at that time, Douzidia et al. translated (DUC)[1] dataset, from English to Arabic for their ATS model's evaluation (Douzidia and Lapalme, 2004). The translation used machine translation at that time which was not as accurate and advanced as these days, and that had a negative impact on the results. Moreover, other ATS models built their own datasets to evaluate their models (Al-Maleh and Desouki, 2020). Consequently, researchers built Arabic ground-truth summaries over the past years, and this section mentions the major ones.

The Essex Arabic Summaries Corpus (EASC) Dataset. EASC (El-Haj et al., 2010) is an ATS dataset, where each summary is extracted from the texts by Mechanical Turk. Its text source is two Arabic newspapers (Alrai and Alwatan) and the Arabic language version of Wikipedia. As a result, it contains 153 Arabic articles and 765 summaries (5 summaries per article). In short, EASC has high-quality human-generated summaries but it is too small and lacks diversity.

Kalimat Dataset. El-Haj et al. worked on a dataset called Kalimat (El-Haj and Koulali, 2013). It has 20,291 extractive Single-document and multi-document system summaries, and includes only 6 categories. It has been collected from only one source, which is Alwatan newspaper from Oman. The single-document summaries are generated based on their model Gen-Summ which inputs the article and its first sentence, then outputs the extractive summary. The multi-document summaries were generated for each 10, 100, and 500 articles in different categories. The generated summaries also lack human evaluation of the summaries.

Arabic News Texts Corpus (ANT) and XL-Sum. ANT (Chouigui et al., 2021), and XL-Sum (Hasan et al., 2021) are the most recent works. ANT collected 31,798 documents paired with summaries using RSS feeds from 5 Arab news sources: AlArabiya, BBC, CNN, France24, and SkyNews, while XL-Sum collected 40,327 only from BBC. ANT includes 6 categories, while XL-Sum reported none. Unlike ANT, LANS utilized the HTML source code *og:description* tag to collect the summaries which is similar to (Grusky et al., 2018). ANT is evaluated on several extractive summarization methods such as LexRank, TextRank, Luhn and LSA. XL-Sum fine-tuned mT5 on their dataset and randomly sampled 500/500 development and test set respectively. Besides, they conducted human evaluation on 250 random samples. When compared to our LANS, our work collected nearly 8 million articles with summaries from 19 Arab countries local newspapers. Moreover, experts evaluated 1,000 random summaries from LANS to substantiate the validity of the summaries.

3 LANS Dataset

This section details how LANS is collected starting from the scraping process to building the dataset and how it is shaped for public use.

3.1 Data Collection

Our main goal is to improve the ATS field by collecting and building the largest and most diverse ATS dataset. We collect newspapers from 19 countries [2]. For consistency and fairness of data col-

[1] An English text summarization dataset of news paired with human summaries. https://duc.nist.gov/

[2] There are 22 Arab countries, but 3 of them: Djibouti, The Comoros Islands, and Somalia, lack Arabic data and reliable newspapers

lection, all the TV news channels' websites are excluded, like Alarabiya, Aljazeera, Arabic CNN, and Arabic BBC because they are primarily established as TV news channels. To make our data sources comprehensive and trustworthy, we collected and listed approximately all the reliable newspapers for each country. For instance, we listed 18 reputable newspapers in Saudi Arabia. After analyzing the newspapers, we then ranked them by assigning the highest priority to the newspaper with the longest publishing history.

Next, we only select the newspapers if their content passes certain criteria:

History of published articles (archive): Each newspaper's website is inspected to examine if it has a considerable historical electronic archive to reestablish the long-history versions of a newspaper. An old reputable newspaper can be given a lower rank over a modern one if the latter has a longer historical e-archive. Thus, LANS has collected data from 1999 to 2019 see Table 2.

Diversity in categories: A newspaper should contain a variety of topics or categories (at least 7), for example, local news, international news, politics, economy, religion, culture, health, sports, art, technology, and so on.

Availability of the summary in the metadata: the metadata of a document has the hidden information of an article. The summary of an article written by the author initially lies in the metadata and also can appear in bold on the webpage or ahead of the article. The availability of the summary published by the author/journalist is the major factor in selecting the newspaper. Only the newspapers with provided summaries in the metadata are selected.

The aforementioned criteria narrow down the list of the reliable newspapers, shown in Table 2. As a result, 22 popular newspapers of 19 Arab countries have been selected for the next step from the period of time between 1999 to 2019. The wide variety of the data sources can significantly benefit the diversity of the summaries.

3.1.1 Data Scraping

Since there are 22 newspaper websites to be scraped, it is necessary to customize a code for each of them. Each code identifies the patterns, the selectors, and the URLs to be scraped. The main information scraped from each news article are the following: URL, title or (headline), article, and finally the summary or (the metadata from *og:description*). An example is shown in Table 3, which shows the scraped information from an article's webpage. For reproducibility, *Scrapy* was ideal, in our case scenario, for implementing recurring and large-scale web scraping projects. Besides, *Scrapy* supports different built-in data outputs such as JSON, XML, and CSV.

3.2 Building LANS Dataset

For the collected data to be curated so it preserves a good quality for reuse and evaluation, we detail how the data is extracted, cleaned, and preprocessed.

3.2.1 Data Extraction

Among the data formats for retrieval, the most convenient format to preserve data quality is XML. The extracted data is stored in a tree structure. Each newspaper has a dataset formatted as the following: "Item" is the root node of the tree. The root has many child nodes "Items". Each "Items", a child node, holds the extracted data of a single document (a newspaper article). The child node, "Items", has 4 child nodes of its own named: Address, Title, Article, and Summary. Each child node of the parent "Items" (Address, Title, Article, and Summary) has 1 or more grandchild nodes depending on the actual values extracted from an article's webpage. The data in this stage is not considered clean nor reliable because it contains many errors that could impact the quality of LANS. Errors can be extraneous or foreign characters, empty values, HTML code, or other common text errors. Thus, we need to clean the data. Plus, to better utilize the data in the XML files, we need to preprocess the data for the evaluation process.

Data cleaning: Initially, more than 11 million articles and their metadata are scraped. The data is laboriously examined to ensure whether the extracted articles are error-free content or not, and to ensure their validity for usage. One of the main errors was the collected articles with missing content. There are some reasons for that. One of the reasons is that many articles contain only images or videos without any textual content, because they are types of news that only report pictures or videos. The other reason for missing content is mistakes from the HTML pages, or content stored under a different selector. All articles with the mentioned errors are removed. Moreover, to clean the other errors the normalization step in the preprocessing steps below is performed. In short, the removed articles may have no title, article, or valid data. Af-

Type	Scraped info
URL	http://www.alwasatnews.com/news/1196668.html
Title	بالصور... المرخ الخيرية تنظم حملة تنظيف لمقبرة القرية
Article	قام المشاركون بإزالة الأشجار والأوساخ الضارة وتقليم الأشجار، وقد شهدت الحملة مشاركة من الأهالي من جميع الفئات العمرية، بالإضافة لأعضاء مجلس إدارة الجمعية. من جانبه، قال رئيس لجنة شئون القرية والمقبرة في الجمعية مصطفى عبدالنبي إن الحملة تأتي استكمالاً لعملية التطوير شامل للمقبرة، حيث تستعد اللجنة للبدء بالمرحلة السادسة من عملية تطوير المقبرة والتي ستشمل عمل كراسي للمظلة ورصف الطريق المؤدي من المغتسل إلى المظلة ونقل خزان الماء الرئيسي من موقعه الحالي إلى الجهة الشرقية للمغتسل وإصلاح واستكمال شراء الاحتياجات، بالإضافة إلى متابعة الخطة التطويرية بالتنسيق مع إدارة الأوقاف الجعفرية، هذا وأثنى على نشاط المشتركين في الحملة، كما قدم شكره لجميع أبناء القرية لتعاونهم لإنجاح حملة تنظيف المقبرة.
Summary	نظمت لجنة شئون القرية في جمعية المرخ الخيرية الاجتماعية، تزامناً مع رأس السنة الميلادية، حملة تنظيف لمقبرة القرية تحت شعار استثمر وقتك لآخرتك، صباح أمس الأحد ١ يناير كانون الثاني ٢٠١٧

Table 3: An example of scraped information from an Article

ter removing all the unusable articles, the number has dropped from 11,115,932 to 8,443,484 articles. After this step, the data is stored in its final XML tree format.

3.2.2 Preprocessing

Even though the data is clean at this stage, it requires preprocessing for ATS evaluation process, due to the complex and rich nature of Arabic language. The steps involve normalization, segmentation, removal of stop words, and lemmatization; in that order. This stage in Arabic is the primary stage to prepare the text for processing and transform the input text into a unified representation.

The normalization step cleans the data and removes many extraneous texts. It removes extra white spaces or tabs, foreign irrelevant characters, non-letters, and diacritics. It also replaces certain Arabic characters with a certain single character to normalize the differences in characters. Normalization also removes the "Tatweel" (character stretching) (Ayedh et al., 2016). For tatweel, a word that appears in this format "تـــمـــديـــد" is going to be replaced with "تمديد".

Segmentation or tokenization are commonly used interchangeably. The segmentation process is applied to segment the article into sentences and prepare for the next steps. We use the Natural Language Toolkit (NLTK) (Loper and Bird, 2002) to tokenize sentences and words. We are aware that some scholars weigh tokenization differently such as when tokenization breaks the words into constituent prefix(es), stem, and suffix(s) (Mubarak, 2017; Abdelali et al., 2016; El-Defrawy et al., 2015; Pasha et al., 2014). However, ATS lemmatization accomplishes the intended purpose of the other definition of Arabic tokenization.

Stop words have a major impact on text summarization because they impact the length of the articles and summaries, and increase the frequency of words which in both cases would change the weights of sentences (El-Khair, 2017; Al-Taani and Al-Omour, 2014). To remove the stop words, we used a list of stop words prepared by Abu El-khair et al (El-Khair, 2017) which contains 1,377 words.

For our evaluation, the final and most crucial step for preprocessing the text is lemmatization. This step can improve the accuracy of the summarization and evaluation process. Lemmatization is the process of reducing words to their basic root by removing the attached affixes of words. LANS dataset does not store the data in the lemmatized format, because lemmatization is usually used in the training or testing on the original data. Many lemmatizers are considered such as Alkhalil (Boudchiche and Mazroui, 2019), ISRI (Khoja) (El-Defrawy et al., 2015), Madamira (Pasha et al., 2014), CAMeL (Obeid et al., 2020), but only Farasa (Mubarak, 2017; Abdelali et al., 2016) is applied because it outperforms the state-of-the-art CAMel by a slight margin and its fast performance on large-scale datasets. Following all the mentioned steps, the dataset is passed for automatic evaluation (see sec 6).

4 LANS Description

LANS builds 8,443,484 articles and their summaries from 22 newspapers of 19 Arab countries dated from 1999 to 2019. The high-level overall statistics in Table 2 show that some newspapers have more data than the others. This does not undermine any country's newspapers. Among the newspapers with a long history of journalism, most of them have been published on physical newspapers before newspapers become digitalized. The dates of collection reflect how much data is available in the e-archive for each newspaper. For instance, Gate Ahram newspaper from Egypt (Gat, 2022) is established in 1875 and has been published since then. However, the available e-archive for the newspaper starts from 2016. Each newspaper's webpage has its own e-archive and its own progress over time. This is why the variations of collection dates exist.

LANS encompasses 19 Arab countries for MSA diversity. One of the overlooked aspects of diversity in Arabic is the diversity of MSA in the Arab countries. It is true that all the newspapers in the Arab countries use the same MSA, but events, culture, and use of vocabulary are different from one country to another. Therefore, it is necessary to collect such diverse data from each country. To achieve MSA diversity in LANS, our dataset encompasses 19 Arab countries - except for the Comoros Islands, Djibouti, and Somalia because of the scarcity of data in their newspapers.

Further, LANS provides a wide-ranging topic variety. The collected data from each country covers different categories, and some newspapers have more categories than others, which enhances the diversity of categories in LANS. Some newspapers have only a few categories (not less than 7), while some others have more than 9 categories including local news, international, political, financial, society, sports, technology, art, health, and religious news articles. This category diversity is one of the features of LANS. It allows researchers to not only create subdatasets, but also create sub-subdataset of any of the subdatasets. For example, a subset can be all articles/summaries from Saudi Arabia. Then, a sub-subdataset can be the local news categories from the subset of Saudi Arabia articles/summaries. This type of diversity can be created from LANS.

The dataset is chunked into separate XML files, each file is under 2 GB to make it easier to load and process. The total size of the whole dataset is 32GB. Each country's dataset is a subset of the whole dataset, and researchers have the freedom to choose a subset or several subsets (by specific countries) to train and evaluate ATS models.

5 Experiment

Since the ATS field is still under-researched for *abstractive* summarization, it is difficult to achieve multiple comparisons among the available works. Therefore, we created a translate-summarize-translate pipeline from the available pretrained state-of-the-art multi-language models such mT5 (Xue et al., 2020), mBART (Tang et al., 2020), and CRISS (Tran et al., 2020). For our experiment, we chose mT5 becasue of its wide coverage of 101 languages and support for 41 languages. The model is utilized to generate summaries of the 1,000 randomly sampled articles, and then compare them with LANS ground-truth summaries using ROUGE-N. In a high-level description, the pipeline inputs the preprocessed samples as mentioned earlier in section 3.2.2, translates the articles (Arabic → English), generates summaries from the translated articles, then translates the generated summaries (English → Arabic) for evaluation. The model for each step of the pipeline will be given later.

Some of the pipeline steps to generate automatic text summaries are tuned to adapt Arabic language. Firstly, we preprocess the text, as detailed in section 3.2.2. Secondly, we translate the articles from Arabic to English. We apply OPUS-MT (Tiedemann and Thottingal, 2020) project. OPUS-MT is based on Marian-NMT (Junczys-Dowmunt et al., 2018), a state-of-the-art transformer-based Neural Machine Translation (NMT), and trained on OPUS data using OPUS-MT-Train. The translation achieves accurate results in machine translation. Next, since articles are translated into English, we process the articles to generate automatic text summaries using mT5 which inherits all the benefits of T5 (Raffel et al., 2019). The automatic text summaries currently are English. Finally, we translate automatic text summaries into Arabic by again applying the OPUS-MT project as described in the second step. An example of the ground-truth summary and a generated Arabic summary are displayed in Table 4.

Both summaries are evaluated by ROUGE (Ganesan, 2018) evaluation metric and will be used for human evaluation (see sec 6.2). We apply ROUGE-1, ROUGE-2, and ROUGE-L to consider different summary lengths. Moreover, we also show how lemmatization impacts the accuracy. The results are reported in Table 5. The results show that the summaries generated by mT5 achieve

Source	Summary
LANS	من المقرر الكشف عن اسماء اهم خمسة مرشحين لجائزة افضل لاعب كرة قدم في افريقيا للعام الحالي غداً الأحد ويتوقع ان يكون كابتن منتخب نيجيريا جاي جاي اوكوتشا من بين اقوى المرشحين للجائزة السنوية.
mT5-based pipeline	من المقرر الكشف عن أفضل خمسة مرشحين لأفضل لاعب كرة قدم في أفريقيا لهذا العام هذا الأحد، ومن المتوقع أن يكون الكابتن المنتخب لنيجيريا جاي أوكوكوشا من بين أقوى المرشحين للجائزة السنوية. أوكوشا، المرشح الرئيسي لأفضل لاعب أفريقي.

Table 4: Table presents a sample of two summaries from LANS and mT5-based pipeline.

lower scores before applying the lemmatization process. After we lemmatized the summaries by Farasa, the results improve by a good margin. In both cases, for a model that has not been designed for Arabic language, mT5 shows good scores when scored with LANS summaries see Table 4.

	Before Lemmatization			After Lemmatization		
	R-1	R-2	R-L	R-1	R-2	R-L
mT5	0.3	0.12	0.28	0.44	0.19	0.38

Table 5: Results of the generated summaries referenced to LANS summaries.

6 Intrinsic Evaluation of LANS

We apply two methods of evaluation to validate the reliability of the summaries from LANS. The first is an automatic evaluation which examines the summarization techniques in LANS. It uses the following metrics: *compression ratio*, *fragment density*, and *coverage*. The automatic evaluation has been performed on the whole dataset. The second evaluation is performed by experts which verifies the quality of LANS by randomly extracting 1,000 articles and their respective summaries, which are evaluated by experts.

6.1 Automatic Evaluation

To assess LANS, we apply 3 common metrics to quantify the abstractness of LANS's summaries and examine their strategies. Note that summaries can be *extractive* or *abstractive*; extractive summaries derive words from the source text, while abstractive summaries use novel words to describe the source text. The applied metrics used are *compression ratio, fragment density (abstractivity), and coverage* (Grusky et al., 2018; Bommasani and Cardie, 2020). **Compression Ratio** quantifies the conciseness of summaries, and is defined as the ratio of words between a summary and an article:

$$\mathbf{CMP}_w(S, A) = 1 - \frac{|S|}{|A|} \tag{1}$$

where $|S|$ is the summary's length and $|A|$ is the article's length in words. **Coverage** by (Grusky et al., 2018) quantifies how much the summary borrows words from the article. Its formula is below:

$$\mathbf{COV}(S, A) = \frac{1}{|S|} \sum_{t \in T(S,A)} |t| \tag{2}$$

where $T(S, A)$ is the set of extractive phrases in summary S extracted from article A, and t is the summary tokens (words) derived from the article. In abstractive summaries, it is preferred not to derive many words from the article.

Fragment Density is proposed by (Grusky et al., 2018), and later introduced as **Abstractivity** in (Bommasani and Cardie, 2020) with a slight change that generalizes it. This paper uses fragment density. It quantifies how well the summaries can construct a sequence of words that are greedily matched in the article. It is measured as the following:

$$\mathbf{DENS}(S, A) = \frac{1}{|S|} \sum_{t \in T(S,A)} |t|^2 \tag{3}$$

The results of the automatic evaluation are reported in Table 6. The ↓ arrow for coverage scores (COV) indicates how abstractive the summaries are from each source. The reported low scores signify that the summaries have novel words to describe the articles. The ↑ arrows for density (DENS) and fragment compression (CMP) mean the higher the better. The highest score for density is in Hespress(Morocco) newspaper summaries, and the lowest is in WAL (a Libyan news agency). For compression, the most concise summaries are reported from Alakhbar (Lebanon), and the least concise ones are reported from Alsudan Alyoom (Sudan). The diversity exists among the Arab countries' style of writing the summaries, and the indi-

Dataset	COV↓	DENS↑	CMP↑	Dataset	COV↓	DENS↑	CMP↑
Elkhabar(Algeria)	0.34	0.87	0.77	Alwatan(Oman)	0.35	0.64	0.68
Alwasat(Bahrain)	0.32	0.88	0.51	Alquds(Palestine)	0.28	0.74	0.65
Gate Ahram(Egypt)	0.27	0.81	0.57	Alquds-UK(Palestine)	0.39	0.90	0.79
Youm7(Egypt)	0.31	0.86	0.53	Alwatan(Qatar)	0.24	0.58	0.74
Aldustoor(Jordan)	0.25	0.52	0.50	Aljazira(Saudi Arabia)	0.23	0.46	0.57
Annahar(Kuwait)	0.24	0.57	0.72	Alryiadh(Saudi Arabia)	0.30	0.73	0.51
Almadapaper(Iraq)	0.45	0.52	0.64	Alsudan Alyoom(Sudan)	0.36	0.31	0.49
Alakhbar(Lebanon)	0.27	0.49	0.82	Zamanalwsl(Syria)	0.26	0.62	0.59
WAL(Libya)	0.32	0.30	0.55	Alssabah(Tunisia)	0.26	0.70	0.58
Sahara Media(Mauritania)	0.32	0.88	0.68	Albayan(Emirates)	0.41	0.35	0.65
Hespress(Morocco)	0.38	1.01	0.78	Almasdar(Yemen)	0.38	0.92	0.77

Table 6: Automatic evaluation results of LANS comparing all newspapers to each other. The up arrow ↑ indicates that higher is better and the opposite for the down arrow ↓. The results show the diversity among the collected datasets from one source to another. It also shows there is a high level of abstractiveness and conciseness.

cation of that is the varying scores in all metrics. The detailed distributions of *fragment density* and *coverage* across LANS dataset are displayed in the appendix Figure 2

6.2 Human Evaluation

Relying on only automatic evaluation and ROUGE metric may result in some limitations, such as biases in scoring against the systems that depend more on paraphrasing such as abstractive systems(Grusky et al., 2018). As a result, even though meaningful summaries are generated, ROUGE can be subjective and assigns a low score to well-generated summaries(See et al., 2017). Therefore, we conduct human evaluation.

Human evaluation is costly, but the results from the automatic method described in Sec. 6.1 are yet to be verified by experts. A survey is created for human experts to assess which summaries capture the full **key information of the articles**, have better **readability**, and have **syntactic correctness**. The survey contained the 1,000 random samples selected for the experiment in Sec. 5. Each survey question contains the following data: the full article; Choice 1: LANS summary; Choice 2: mT5-based generated summary; and Choice 3: none-of-the-above (non of the summaries). Choices 1 and 2 were shuffled and anonymized, so human experts can make fairer choices with less biases. For example, if Choice 1 was always LANS's summary, then human experts may form a judgement to always choose Choice 1. Therefore, the choices were shuffled. Besides, the choices were anonymous. It means that human evaluators do not know the origin of each summary.

The experts who did the survey are highly knowledgeable in Arabic. For a human expert to evaluate the survey; an expert should be an Arabic native speaker, also, an expert should at least have a bachelor's degree majoring in Arabic Language. The experts were asked not only to choose which choice is the fittest for the given criteria, but also to provide their feedback on the choices. Human evaluation results show that 954, out of the 1,000, LANS extracted summaries have more accurate semantic representation, and correct syntactic forms. The semantic representation means that the summary captures salient and key information of the article and has better readability. The results, also, show that 2 of the choices are "none", which means neither summaries meet the required criteria. While the ROUGE scores are low between the automatically generated summaries and the LANS summaries, the 95.4% approval rating for LANS summaries during the human evaluation validates the use of the descriptions present in the source code of the articles as their summaries.

7 Conclusion

This work presents LANS, a large-scale and diverse text summarization dataset of more than 8 million new articles paired with their summaries written by journalists. The summaries are collected from the metadata of 22 scraped popular Arab newspapers' websites from the period between 1999 to 2019. For each of those resources, LANS considered a wide range of topics. The work applied two evaluation methods (automatic and human) to verify the superiority of the extracted summaries in LANS. The dataset can be accessed upon request. [3]. LANS offers this dataset for researchers to advance the field of ATS, and takes advantage of the data to

[3] Request data from first author

train and evaluate the results of new models on this dataset.

8 Limitations

The distribution of data in LANS is far from uniform with regards to the newspapers coming from each country. This disparity is primarily driven by the varying number of newspapers in different countries. As a result, some nations' data representation is much more than others due to the former's extensive media landscape.

This uneven distribution underscores the importance of considering geographic and media-related factors when conducting data-driven research or analysis.

9 Ethical Statement

In accordance with ethical research practices, it is important to clarify that the data collection process for the LANS dataset did not violate any copyrights or intellectual property rights. The dataset comprises articles and their summaries obtained from publicly accessible websites of 22 major Arab newspapers, all of which span from 1999 to 2019. Every article included in the dataset is properly cited, including its originating source, and each has an associated URL, allowing for verification and direct reference. The data is solely utilized for academic and research purposes, intended to advance the field of Arabic Text Summarization (ATS). The extraction and use of this data adhere to all relevant ethical guidelines, ensuring that the journalistic integrity of the original articles and their authors is maintained. Thus, the dataset aims to serve as a high-quality and diverse resource for research while respecting all ethical and legal norms.

References

2022. Gate ahram newspaper (egypt). http://gate.ahram.org.eg/. Accessed: 2020-02-02.

Ahmed Abdelali, Kareem Darwish, Nadir Durrani, and Hamdy Mubarak. 2016. Farasa: A fast and furious segmenter for arabic. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 11–16.

Abdullah M Abu Nada, Eman Alajrami, Ahmed A Al-Saqqa, and Samy S Abu-Naser. 2020. Arabic text summarization using arabert model using extractive text summarization approach. *International Journal of Academic Information Systems Research (IJAISR)*.

Fatima T AL-Khawaldeh and Venus W Samawi. 2015. Lexical cohesion and entailment based segmentation for arabic text summarization (lceas). *World of Computer Science & Information Technology Journal*, 5(3).

Molham Al-Maleh and Said Desouki. 2020. Arabic text summarization using deep learning approach. *Journal of Big Data*, 7(1):1–17.

Asma Bader Al-Saleh and Mohamed El Bachir Menai. 2016. Automatic arabic text summarization: a survey. *Artificial Intelligence Review*, 45(2):203–234.

Ahmad T Al-Taani and Maha M Al-Omour. 2014. An extractive graph-based arabic text summarization approach. In *The International Arab Conference on Information Technology*.

Abdullah Ayedh, Guanzheng Tan, Khaled Alwesabi, and Hamdi Rajeh. 2016. The effect of preprocessing on arabic document categorization. *Algorithms*, 9(2):27.

Riadh Belkebir and Ahmed Guessoum. 2015. A supervised approach to arabic text summarization using adaboost. In *New contributions in information systems and technologies*, pages 227–236. Springer.

Rishi Bommasani and Claire Cardie. 2020. Intrinsic evaluation of summarization datasets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8075–8096.

Mohamed Boudchiche and Azzeddine Mazroui. 2019. A hybrid approach for arabic lemmatization. *International Journal of Speech Technology*, 22(3):563–573.

Amina Chouigui, Oussama Ben Khiroun, and Bilel Elayeb. 2021. An arabic multi-source news corpus: Experimenting on single-document extractive summarization. *Arabian Journal for Science and Engineering*, 46(4):3925–3938.

Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. 2020. Gsum: A general framework for guided neural abstractive summarization. *arXiv preprint arXiv:2010.08014*.

Fouad Soufiane Douzidia and Guy Lapalme. 2004. Lakhas, an arabic summarization system. In *Proceedings of DUC*, volume 4, pages 128–135. Citeseer.

Mahmoud El-Defrawy, Yasser El-Sonbaty, and Nahla Belal. 2015. Enhancing root extractors using light stemmers. In *Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation: Posters*, pages 157–166.

Fatma El-Ghannam and Tarek El-Shishtawy. 2014. Multi-topic multi-document summarizer. *arXiv preprint arXiv:1401.0640*.

Mahmoud El-Haj and Rim Koulali. 2013. Kalimat a multipurpose arabic corpus. In *Second Workshop on Arabic Corpus Linguistics (WACL-2)*, pages 22–25.

Mahmoud El-Haj, Udo Kruschwitz, and Chris Fox. 2010. Using mechanical turk to create a corpus of arabic summaries.

Wafaa S El-Kassas, Cherif R Salama, Ahmed A Rafea, and Hoda K Mohamed. 2021. Automatic text summarization: A comprehensive survey. *Expert Systems with Applications*, 165:113679.

Ibrahim Abu El-Khair. 2017. Effects of stop words elimination for arabic information retrieval: a comparative study. *arXiv preprint arXiv:1702.01925*.

Asmaa Elsaid, Ammar Mohammed, Lamiaa Fattouh, and Mohamed Sakre. 2022. A comprehensive review of arabic text summarization. *IEEE Access*.

Hamzah Noori Fejer and Nazlia Omar. 2014. Automatic arabic text summarization using clustering and keyphrase extraction. In *Proceedings of the 6th International Conference on Information Technology and Multimedia*, pages 293–298. IEEE.

Kavita Ganesan. 2018. Rouge 2.0: Updated and improved measures for evaluation of summarization tasks. *arXiv preprint arXiv:1803.01937*.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. *arXiv preprint arXiv:1804.11283*.

Tahmid Hasan, Abhik Bhattacharjee, Md Saiful Islam, Kazi Samin, Yuan-Fang Li, Yong-Bin Kang, M Sohel Rahman, and Rifat Shahriyar. 2021. Xl-sum: Large-scale multilingual abstractive summarization for 44 languages. *arXiv preprint arXiv:2106.13822*.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. In *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics*.

Hamdy Mubarak. 2017. Build fast and accurate lemmatization for arabic. *arXiv preprint arXiv:1710.06700*.

Ossama Obeid, Nasser Zalmout, Salam Khalifa, Dima Taji, Mai Oudah, Bashar Alhafni, Go Inoue, Fadhl Eryani, Alexander Erdmann, and Nizar Habash. 2020. Camel tools: An open source python toolkit for arabic natural language processing. In *Proceedings of the 12th language resources and evaluation conference*, pages 7022–7032.

Arfath Pasha, Mohamed Al-Badrashiny, Mona T Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan Roth. 2014. Madamira: A fast, comprehensive tool for morphological analysis and disambiguation of arabic. In *LREC*, volume 14, pages 1094–1101.

Dragomir Radev, Eduard Hovy, and Kathleen McKeown. 2002. Introduction to the special issue on summarization. *Computational linguistics*, 28(4):399–408.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning. *arXiv preprint arXiv:2008.00401*.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT – building open translation services for the world. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 479–480, Lisboa, Portugal. European Association for Machine Translation.

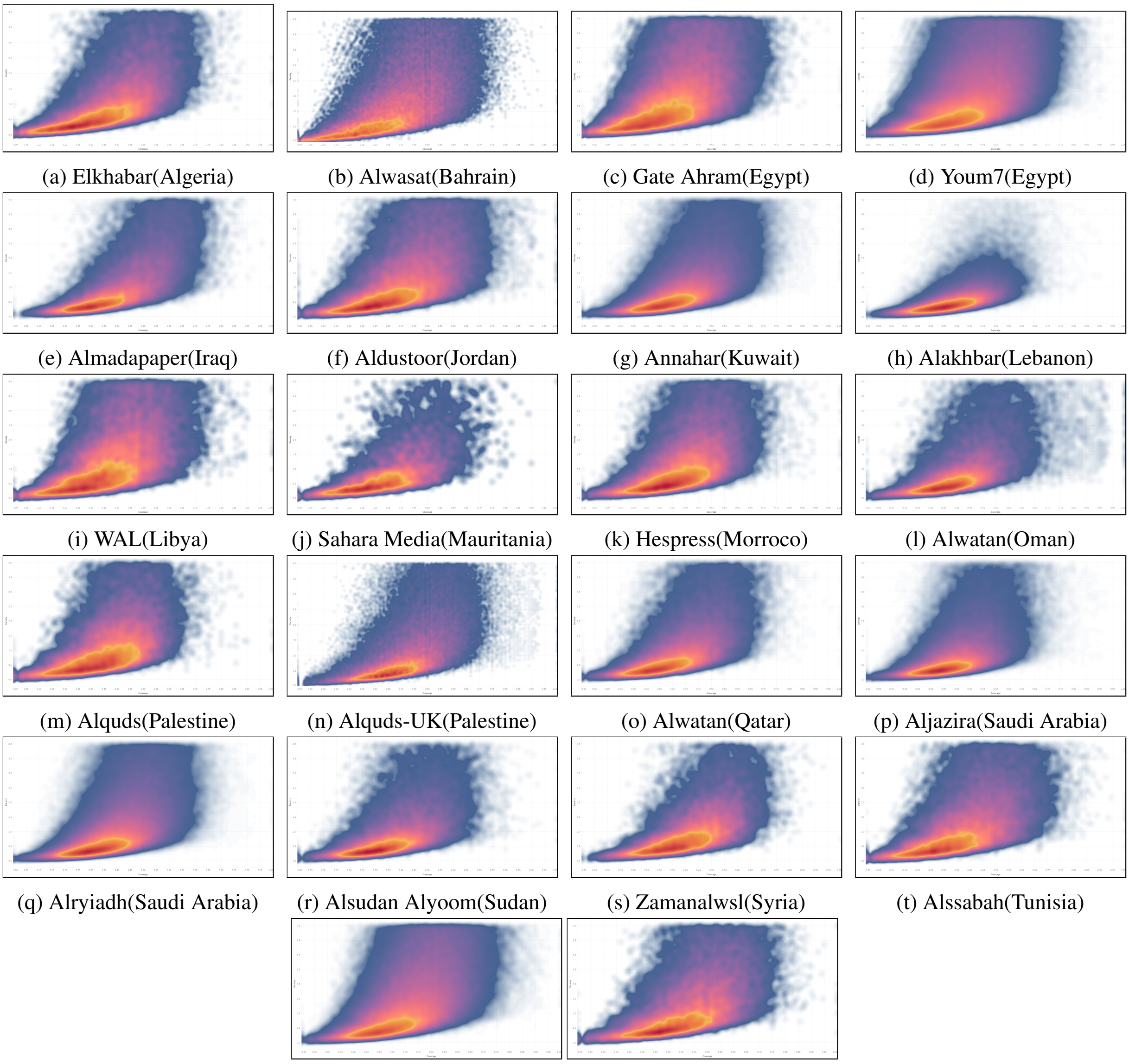
Chau Tran, Yuqing Tang, Xian Li, and Jiatao Gu. 2020. Cross-lingual retrieval for iterative self-supervised training. *Advances in Neural Information Processing Systems*, 33:2207–2219.

Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. 2017. TL;DR: Mining Reddit to learn automatic summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 59–63, Copenhagen, Denmark. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*.

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

10 Appendix



(a) Elkhabar(Algeria) (b) Alwasat(Bahrain) (c) Gate Ahram(Egypt) (d) Youm7(Egypt)

(e) Almadapaper(Iraq) (f) Aldustoor(Jordan) (g) Annahar(Kuwait) (h) Alakhbar(Lebanon)

(i) WAL(Libya) (j) Sahara Media(Mauritania) (k) Hespress(Morroco) (l) Alwatan(Oman)

(m) Alquds(Palestine) (n) Alquds-UK(Palestine) (o) Alwatan(Qatar) (p) Aljazira(Saudi Arabia)

(q) Alryiadh(Saudi Arabia) (r) Alsudan Alyoom(Sudan) (s) Zamanalwsl(Syria) (t) Alssabah(Tunisia)

(u) Albayan(Emirates) (v) Almasdar(Yemen)

Figure 2: The distributions of fragment density and coverage across the datasets of LANS is displayed in the sub-figures. This shows how diverse the dataset is from one country to another. The sub-figures support table.6. Each sub-figure is a normalized bivariate density plot. The X-axis represents the coverage, and it ranges from 0 to 1. The Y-axis represents the Fragment density(Abstractiveness), and it ranges from 1 to 4. The red color shows where most of the summaries are, and the dark blue color indicates where the least summaries are. The extraction method is explained in section.6.1

Beyond English:
Evaluating LLMs for Arabic Grammatical Error Correction

Sang Yun Kwon[ξ] **Gagan Bhatia**[ξ] **El Moatez Billah Nagoudi**[ξ]
Muhammad Abdul-Mageed[ξ,ω]

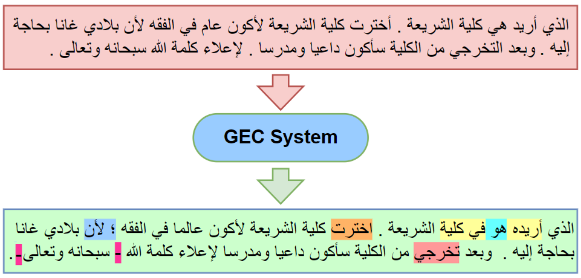
[ξ]Deep Learning & Natural Language Processing Group, The University of British Columbia
[ω]Department of Natural Language Processing & Department of Machine Learning, MBZUAI
{skwon01@student.,gagan30@student.,muhammad.mageed@}ubc.ca

Abstract

Large language models (LLMs) finetuned to follow human instruction have recently exhibited significant capabilities in various English NLP tasks. However, their performance in grammatical error correction (GEC), especially on languages other than English, remains significantly unexplored. In this work, we evaluate the abilities of instruction finetuned LLMs in Arabic GEC, a complex task due to Arabic's rich morphology. Our findings suggest that various prompting methods, coupled with (in-context) few-shot learning, demonstrate considerable effectiveness, with GPT-4 achieving up to 65.49 F_1 score under expert prompting (approximately 5 points higher than our established baseline). Despite these positive results, we find that instruction finetuned models, regardless of their size, are still outperformed by fully finetuned ones, even if they are significantly smaller in size. This disparity highlights substantial room for improvements for LLMs. Inspired by methods used in low-resource machine translation, we also develop a method exploiting synthetic data that significantly outperforms previous models on two standard Arabic benchmarks. Our best model achieves a new SOTA on Arabic GEC, with 73.29 and 73.26 F_1 on the 2014 and 2015 QALB datasets, respectively, compared to peer-reviewed published baselines.

1 Introduction

As interest in second language learning continues to grow, ensuring the accuracy and effectiveness of written language becomes increasingly significant for pedagogical tools and language evaluation (Rothe et al., 2021; Tarnavskyi et al., 2022). A key component in this respect is grammatical error correction (GEC), a sub-area of natural language generation (NLG), which analyzes written text to automatically detect and correct diverse grammatical errors. Figure 1 shows an instance of GEC from Mohit et al. (2014). Despite the growing attention to GEC, it is predominantly studied within the English language. Extending GEC systems to other languages presents significant challenge, due to lack of high-quality parallel data and/or inherent challenges in these languages. Recognizing this, our work focuses on Arabic. In addition to being less-explored for GEC (Mohit et al., 2014; Rozovskaya et al., 2015a; Mohit et al., 2014; Rozovskaya et al., 2015a; Solyman et al., 2022; Alhafni et al., 2023), Arabic has complex grammar and rich morphology that present significant challenges and further motivate our work.



الذي أريد هي كلية الشريعة . أخترت كلية الشريعة لأكون عام في الفقه لأن بلادي غانا بحاجة
إليه . وبعد التخرجي من الكلية سأكون داعيا ومدرسا . لإعلاء كلمة الله سبحانه وتعالى .

الذي أريده هو في كلية الشريعة . اخترت كلية الشريعة لأكون عالما في الفقه ؛ لأن بلادي غانا
بحاجة إليه . وبعد تخرجي من الكلية سأكون داعيا ومدرسا لإعلاء كلمة الله ـ سبحانه وتعالى ـ.

Figure 1: An example of an Arabic GEC system showcasing six types of errors: *character replacement* , *missing word* , *hamza error* , *missing punctuation* , *additional character* , and *punctuation confusion* .

Focusing primarily on English, the field of GEC has witnessed significant advancements, specifically with the emergence of sequence-to-sequence (seq2seq) (Chollampatt and Ng, 2018; Gong et al., 2022) and sequence-to-edit approaches (seq2edit) (Awasthi et al., 2019; Omelianchuk et al., 2020) achieving SoTA results in the CONLL-2014 (Ng et al., 2014) and the BEA-2019 shared task (Bryant et al., 2019), respectively. In spite of the efficacy of these approaches, they rely heavily on large amounts of labeled data. This poses issues in low-resource scenarios (Feng et al., 2021). Yet, scaled up language models, *aka* large language models (LLMs) have recently demonstrated remarkable potential in various NLP tasks. The core strength of LLMs lies in their capacity to gen-

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 101–119
December 7, 2023 ©2023 Association for Computational Linguistics

eralize across a wide range of languages and tasks, and in-context learning (ICL), enabling them to handle various NLP tasks with just a few examples (i.e., few-shot learning). A key strategy for LLMs is *instruction fine-tuning*, where they are refined on a collection of tasks formulated as instructions (Wei et al., 2022a). This process amplifies the models' ability to respond accurately to directives, reducing the need for few-shot examples (Ouyang et al., 2022; Wei et al., 2022b; Sanh et al., 2021).

Given the ability of LLMs to adeptly address the low-resource challenge, we investigate them in the context of GEC. Focusing primarily on ChatGPT, we examine the effectiveness of various prompting strategies such as few-shot chain of thought (CoT) prompting (Kojima et al., 2022) and expert prompting (Xu et al., 2023). Our research extends the realm of GEC research by concentrating on the unique challenges posed by Arabic. Drawing upon the work of Junczys-Dowmunt et al. (2018a), we frame these challenges within the context of a low-resource MT task. We then carefully conduct a thorough comparison of the different methodologies employed in addressing GEC in Arabic. Our key contributions in this paper are as follows:

1. We conduct a comprehensive investigation of the potential of LLMs for tasks involving GEC in Arabic.

2. We methodically investigate the utility of different prompting methods for generating synthetic data with ChatGPT for GEC.

3. We further carry out in-depth comparisons between several approaches (seq2seq, seq2edit, and instruction fine-tuning) for Arabic GEC (AGEC), allowing us to offer novel insights as to the utility of these approaches.

The rest of this paper is organized as follows: In Section 2, we review related work with a particular emphasis on Arabic. In Section 3, we outline our experimental setups. We present our experiments on LLMs and prompting strategies in Section 4. In Section 5, we introduce our seq2seq approach along with data augmentation techniques; Section 6 discusses our seq2edit approach. In Section 7, we conduct a comprehensive analysis of our best model. We discuss our results in Section 8, and conclude in Section 9.

2 Related Work

Progress in GEC. Pretrained Transformer models have reframed GEC as an MT task, achieving SoTA results (Ng et al., 2014; Felice et al., 2014; Junczys-Dowmunt et al., 2018b; Grundkiewicz et al., 2019). In contrast, sequence2edit approaches view the task as text-to-edit, converting input sentences into edit operations to produce corrected sentences (Malmi et al., 2019; Awasthi et al., 2019; Omelianchuk et al., 2020). These approaches both streamline the training process and enhance model accuracy. Further progress has also been made through methods such as instruction fine-tuning (Chung et al., 2022) and innovative prompting techniques, such as CoT (Kojima et al., 2022) and Expert (Xu et al., 2023) prompting. Recent applications of LLMs, like ChatGPT in GEC, highlight their potential. We provide further details on each of these methods in Appendix A.

Arabic GEC. Challenges in AGEC stem from the complexity and morphological richness of Arabic. Arabic, being a collection of a diverse array of languages and dialectal varieties with Modern Standard Arabic (MSA) as a contemporary variety, is further complicated by the optional use of diacritics. This introduces orthographic ambiguity, further complicating GEC in Arabic (Abdul-Mageed et al., 2020; Belkebir and Habash, 2021). Despite these challenges, progress in AGEC has been made. This includes development of benchmark datasets through the QALB-2014 and 2015 shared tasks (Mohit et al., 2014; Rozovskaya et al., 2015b; Habash and Palfreyman, 2022), and introduction of synthetic datasets (Solyman et al., 2021, 2023). As for model development, character-level seq2seq models (Watson et al., 2018) and other novel approaches are shown to be effective on AGEC L1 data. Further details about progress in AGEC are provided in Appendix A. Despite this progress, no exploration has been undertaken into the utility of using ChatGPT (or other LLMs) for AGEC. Moreover, substantial work remains in exploring synthetic data generation, including the use of LLMs and the adoption of diverse machine learning approaches. Our research aims to address these gap.

3 Experimental Setup

3.1 Datasets

In this study, we make use of the QALB-2014 (Mohit et al., 2014) and 2015 (Rozovskaya et al., 2015b) datasets to evaluate the performance of our

Dataset	Statistics	Train	Dev	Test	Level
QALB-2014	Number of sents.	19,411	1,017	968	L1
	Number of words.	1,021,165	54,000	51,000	L1
	Number of error.	306,000	16,000	16,000	L1
QALB-2015	Number of sents.	310	154	920	L2
	Number of words.	43,353	24,742	48,547	L2
	Number of error.	13,200	7,300	13,000	L2

Table 1: Statistics for QALB-2014 and 2015 Train, development (Dev), and Test datasets.

models. Both datasets make use of the QALB corpus (Zaghouani et al., 2014), a manually corrected collection of Arabic texts. These texts include online commentaries from Aljazeera articles in MSA by L1 native speakers, as well as texts produced by L2 learners of Arabic. Both the QALB 2014 and 2015 datasets are split into training (Train), development (Dev), and test (Test) sets based on their annotated dates. QALB 2015 includes L1 commentaries and L2 texts that cover different genres and error types. For the purposes of our study, we exclusively use the L1 test set (2015), as we focus on sentence-level AGEC, where L2 test sets are document-level. We used Train, Dev, and Test splits described in Table 1.

3.2 Evaluation

Metrics. For evaluation, we utilize the overlap-based metric MaxMatch (M^2) (Dahlmeier and Ng, 2012), which aligns source and hypothesis sentences based on Levenshtein distance , selecting maximal matching edits, scoring the precision (P), recall (R), and F_1 measure. Moreover, we report the $F_{0.5}$ score , a variation of the F_1 score that places twice as much weight on precision than on recall. This reflects a consensus, in alignment with recent works on GEC, that precision holds greater importance than error correction in GEC systems. Importantly, we use the exact scripts provided from the shared task for evaluation, ensuring consistency with other studies.

3.3 Models & Fine-tuning

LLMs. To evaluate the capabilities of LLMs for AGEC, we prompt and fine-tune LLMs of varying sizes, including LLaMA-7B (Touvron et al., 2023), Vicuna-13B (Chiang et al., 2023), Bactrian-X_{bloom}-7B (Li et al., 2023), and Bactrian-X_{llama}-7B (Li et al., 2023). For experiments with ChatGPT, we use the official API to prompt ChatGPT-3.5 Turbo and GPT-4. We instruction fine-tune each smaller model for 4 epochs using a learning rate of 2e-5 and a batch size of 4. We then pick the best-performing model on our Dev, then report on our blind Test.

Seq2seq models. Our baseline settings for seq2seq models include AraBart (Eddine et al., 2022) and AraT5$_{v2}$ (Nagoudi et al., 2022), both of which are text-to-text transformers specifically tailored for Arabic. We also evaluate the performance of the mT0 (Muennighoff et al., 2022) and mT5 (Xue et al., 2020) variants of the T5 model (Raffel et al., 2020), both configured for multilingual tasks. Each model is fine-tuned for 50 epochs, with an early stopping patience of 5 using a learning rate of 5e-5 and a batch size of 32. These models serve as the baseline for comparison throughout our experiments.

Seq2edit models. ARBERT$_{v2}$ and MARBERT$_{v2}$ (Abdul-Mageed et al., 2021) serve as the baselines for our seq2edit experiments. We fine-tune each model for 100 epochs for each training stage, employing a learning rate of 1e-5 and a batch size of 4, with an early stopping patience of 5.

All models are trained for three runs, with seeds of 22, 32, and 42. We then select the best-performing model based on our Dev data for blind-testing on the Test sets. *We report the mean score of the three runs, along with its standard deviation.* Results on the Dev set, and more details regarding hyperparameters are provided in Appendix 15, and Appendix 14.

4 LLMs and Prompting Techniques

This section outlines our experiments designed to instruction fine-tune LLMs and explore different prompting methods for ChatGPT in the context of AGEC. We begin by experimenting with various prompting strategies using ChatGPT, comparing its performance against smaller LLMs and our listed baselines. We evaluate the performance of ChatGPT-3.5 Turbo (ChatGPT) and GPT-4, under two prompting strategies: ***Few-shot CoT*** (Fang et al., 2023) and ***Expert Prompting*** (Xu et al., 2023). We now describe our prompting strategies.

4.1 ChatGPT Prompting

Preliminary experiment. Initially, we experiment with a diverse set of prompt templates to assess ChatGPT's capabilities in zero-shot learning as well as two aspects of few-shot learning: vanilla few-shot and few-shot CoT (Fang et al., 2023). We also experiment with prompts in both English and Arabic. However, we discover that the responses from these prompt templates contain extraneous explanations and are disorganized, necessitating substantial preprocessing for compatibility with the M^2 scorer. This problem is particularly notable in the zero-shot and Arabic prompt setups, which fails to yield output we can automatically evaluate.

Few-shot CoT. Adopting the few-shot CoT prompt design strategy from Kojima et al. (2022) and Fang et al. (2023), we implement a two-stage approach. Initially, we engage in *'reasoning extraction'*, prompting the model to formulate an elaborate reasoning pathway. This is followed by an *'answer extraction'* phase, where the reasoning text is combined with an answer-specific trigger sentence to form a comprehensive prompt. In our few-shot CoT settings, we include labeled instances from the Dev set in our prompts to implement ICL, facilitating learning from examples (Brown et al., 2020). This involves providing erroneous sentences, labeled *<input> SRC </input>* , along with their corrected versions, labeled *<output> TGT </output>* , from the original Dev set.

Expert prompting. Xu et al. (2023) introduces a novel strategy, which leverages the expert-like capabilities of LLMs. This method involves assigning expert personas to LLMs, providing specific instructions to enhance the relevance and quality of the generated responses. Following the framework of Xu et al. (2023), we ensure that our AGEC correction tool exhibits three key characteristics: being ***distinguished***, ***informative***, and ***automatic*** during the *'reasoning extraction'* stage of our prompt. To achieve this, we employ a distinct and informative collection of various error types as proposed in the Arabic Learner Corpus taxonomy (Alfaifi and Atwell, 2012). We then prompt to automate the system by instructing it to operate on sentences labeled with *<input>* and *<output>* tags. Both prompts are illustrated in Figure 2.

4.2 ChatGPT Results.

Table 2 presents the performance of ChatGPT under different prompting strategies, compared to the baseline settings. We observe improvements, particularly as we progress from the one-shot to five-shot configurations for both the few-shot CoT and expert prompting (EP) strategies. Under the CoT prompt, ChatGPT's $F_{1.0}$ score increases from 53.59 in the one-shot setting to 62.04 in the five-shot setting. A similar upward trend is evident with the EP strategy, where the $F_{1.0}$ score rises from 55.56 (one-shot) to 63.98 (five-shot). Among all experiments involving ChatGPT, the three-shot and five-shot settings of GPT-4, CoT, achieve the highest scores, with $F_{1.0}$ of 63.98 and 65.49, respectively.

Settings	**Models**	**Exact Match**			
		P	**R**	$\mathbf{F_{1.0}}$	$\mathbf{F_{0.5}}$
Baselines	mT0	70.76 $^{\pm 0.03}$	50.78 $^{\pm 0.07}$	59.12 $^{\pm 0.05}$	65.59 $^{\pm 0.03}$
	mT5	70.64 $^{\pm 0.12}$	50.16 $^{\pm 0.05}$	58.66 $^{\pm 0.05}$	65.30 $^{\pm 0.09}$
	AraBART	70.71 $^{\pm 0.06}$	60.46 $^{\pm 0.04}$	65.18 $^{\pm 0.07}$	68.39 $^{\pm 0.08}$
	AraT5$_{v2}$	**73.04** $^{\pm 0.10}$	**63.09** $^{\pm 0.15}$	**67.70** $^{\pm 0.12}$	**70.81** $^{\pm 0.11}$
+ CoT	ChatGPT (1-shot)	58.71	49.29	53.59	56.55
	ChatGPT (3-shot)	64.60	60.37	62.41	63.71
	ChatGPT (5-shot)	64.70	59.59	62.04	63.61
+ EP	ChatGPT (1-shot)	60.49	51.37	55.56	58.42
	ChatGPT (3-shot)	65.83	61.41	63.54	64.90
	ChatGPT (5-shot)	66.53	61.62	63.98	65.49
+ CoT	GPT4 (1-shot) *	–	–	–	–
	GPT4 (3-shot)	69.31	59.24	63.88	67.03
	GPT4 (5-shot)	69.46	61.96	65.49	67.82

Table 2: Performance of ChatGPT under different prompting strategies on QALB-2014 Test set. * Results for QALB-2015 Test and GPT4 1-shot are not included due to the high cost in producing these results, and a pattern has already been established showing that performance increases as we increase the number of N-shot examples. More details are in Appendix B.2.

4.3 Instruction-Finetuning LLMs

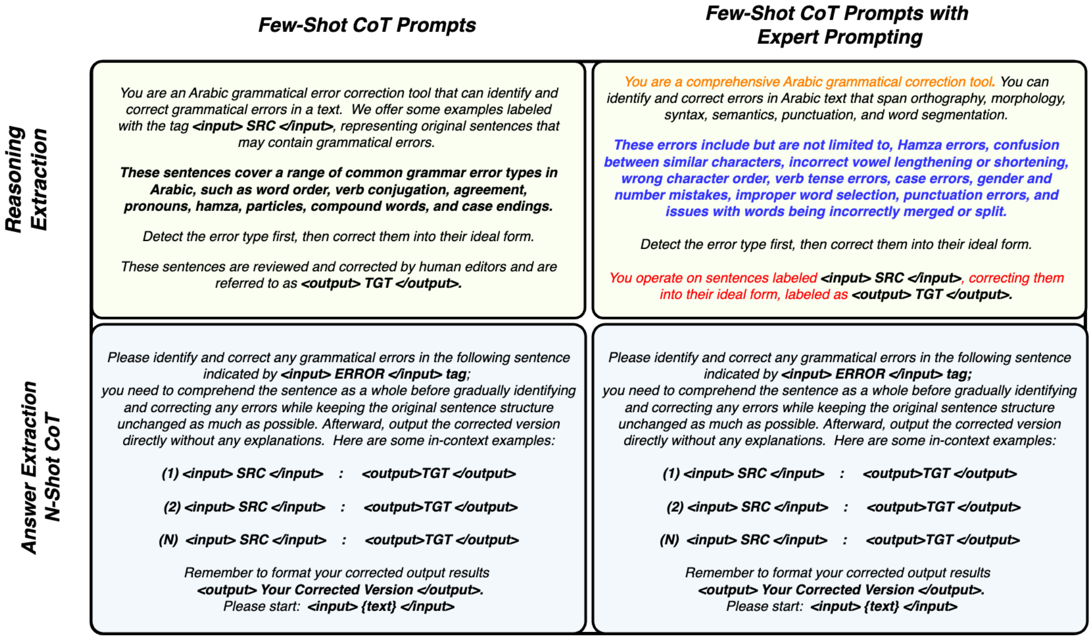
Fine-tuning LLMs. To instruct fine-tune *relatively* large models, *henceforth* just LLMs, we first train these models on the translated Alpaca dataset (Taori et al., 2023) [1] to allow the models to gain deeper understanding of the Arabic language and its complexities. Following this, we further fine-tune the models on the QALB dataset, to specifically target the task of GEC. Then, we employ well-structured task instructions and input prompts, enabling the models to take on GEC tasks. Each model is assigned a task, given an instruction and an input for output generation. We provide an illustration of the instructions we use for model training in Appendix B.

LLM results. As shown in Figure 3, larger models such as Vicuna-13B and models trained on multilingual datasets like Bactrian-X$_{llama}$-7B, and Bactrian-X$_{bloom}$-7B exhibit an overall trend of better performance, achieving F_1 of 58.30, 50.1, and 52.5, respectively. Despite these improvements, it is noteworthy that all models fall short of ChatGPT's. This reaffirms ChatGPT's superior ability on AGEC.

5 Data Augmentation

Motivated by the significant improvements observed in low-resource GEC tasks in languages

[1] We translate the Alpaca datasets using NLLB MT model (Costa-jussà et al., 2022)


Few-Shot CoT Prompts
Few-Shot CoT Prompts with Expert Prompting
Reasoning Extraction
Answer Extraction N-Shot CoT

You are an Arabic grammatical error correction tool that can identify and correct grammatical errors in a text. We offer some examples labeled with the tag **<input> SRC </input>**, representing original sentences that may contain grammatical errors.

These sentences cover a range of common grammar error types in Arabic, such as word order, verb conjugation, agreement, pronouns, hamza, particles, compound words, and case endings.

Detect the error type first, then correct them into their ideal form.

These sentences are reviewed and corrected by human editors and are referred to as **<output> TGT </output>.**

You are a comprehensive Arabic grammatical correction tool. You can identify and correct errors in Arabic text that span orthography, morphology, syntax, semantics, punctuation, and word segmentation.

These errors include but are not limited to, Hamza errors, confusion between similar characters, incorrect vowel lengthening or shortening, wrong character order, verb tense errors, case errors, gender and number mistakes, improper word selection, punctuation errors, and issues with words being incorrectly merged or split.

Detect the error type first, then correct them into their ideal form.

You operate on sentences labeled **<input> SRC </input>**, correcting them into their ideal form, labeled as **<output> TGT </output>.**

Please identify and correct any grammatical errors in the following sentence indicated by **<input> ERROR </input> tag;**
you need to comprehend the sentence as a whole before gradually identifying and correcting any errors while keeping the original sentence structure unchanged as much as possible. Afterward, output the corrected version directly without any explanations. Here are some in-context examples:

(1) <input> SRC </input> : <output>TGT </output>

(2) <input> SRC </input> : <output>TGT </output>

(N) <input> SRC </input> : <output>TGT </output>

Remember to format your corrected output results
<output> Your Corrected Version </output>.
Please start: **<input> {text} </input>**

Please identify and correct any grammatical errors in the following sentence indicated by **<input> ERROR </input> tag;**
you need to comprehend the sentence as a whole before gradually identifying and correcting any errors while keeping the original sentence structure unchanged as much as possible. Afterward, output the corrected version directly without any explanations. Here are some in-context examples:

(1) <input> SRC </input> : <output>TGT </output>

(2) <input> SRC </input> : <output>TGT </output>

(N) <input> SRC </input> : <output>TGT </output>

Remember to format your corrected output results
<output> Your Corrected Version </output>.
Please start: **<input> {text} </input>**

Figure 2: Illustration of Few-Shot CoT and Expert Prompts for Arabic Grammatical Error Correction.

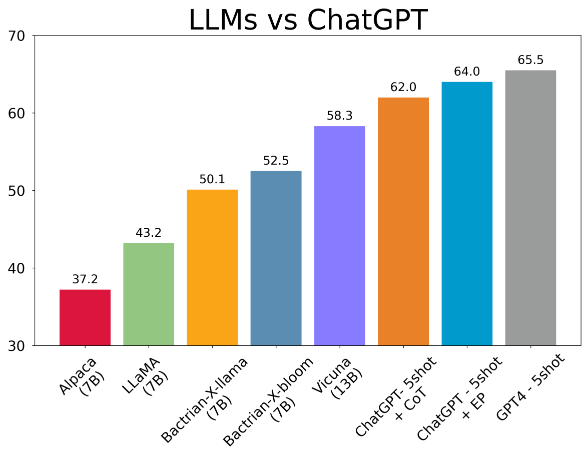


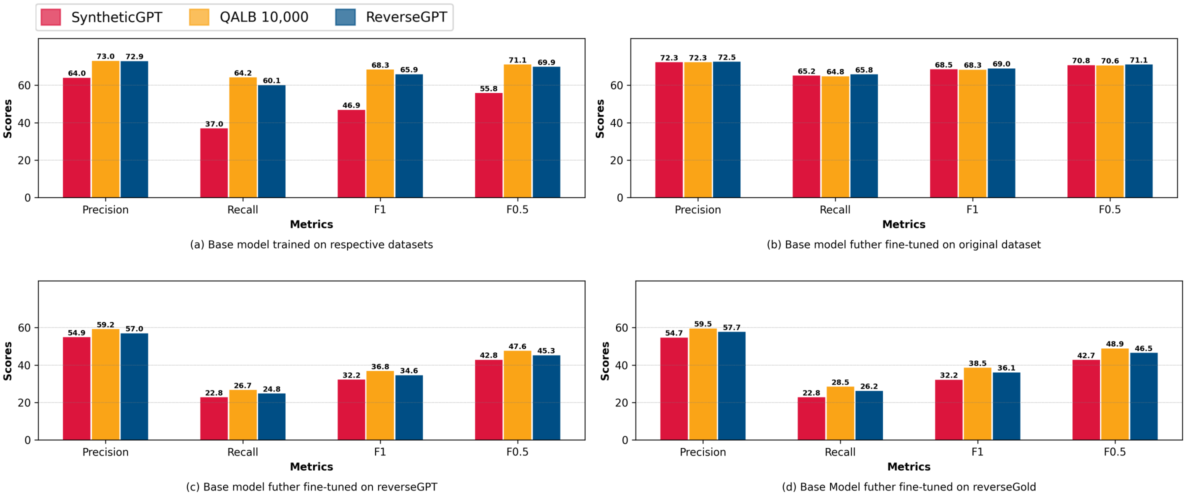
Figure 3: Comparison of F_1 scores between LLMs and ChatGPT on the QALB-2014 Test set.

such as German, Russian, and Czech through synthetic data (Flachs et al., 2021), and recognizing the recent efforts to develop synthetic data for AGEC (Solyman et al., 2021), we experiment with three distinctive data augmentation methods.

ChatGPT as corruptor. With slight adaptation to our original prompt, we engage ChatGPT as an AI model with the role of introducing grammatical errors into Arabic text to generate artificial data. We randomly sample 10,000 correct sentences from the QALB-2014 Train set and, using the taxonomy put forth by the Arabic Learner Corpus (Alfaifi and Atwell, 2012), prompt ChatGPT to corrupt these, creating a parallel dataset. We refer to the resulting dataset as **syntheticGPT**.

Reverse noising. We adopt a *reverse noising* approach (Xie et al., 2018), training a reverse model that converts clean sentences Y into noisy counterparts X. This involves implementing a standard beam search to create noisy targets $\hat{Y}$ from clean input sentences Y. Our approach incorporates two types of reverse models: the first trains on both QALB-2014 and 2015 gold datasets, and the second on the syntheticGPT dataset. Subsequently we generate a parallel dataset using commentaries from the same newspaper domain as our primary clean inputs, matching the original Train data. We name the respective parallel datasets **reverseGold**, and **reverseGPT**.

Data augmentation evaluation. To evaluate the efficacy of ChatGPT in generating artificial data, we select $10,000$ parallel sentences from syntheticGPT, $10,000$ examples from reverseGPT, and $10,000$ parallel sentences from the original training set. We then further fine-tune each model on the original training dataset and the two synthetically generated reverse noised datasets, aiming to assess if these artificially crafted datasets can replace the gold standard training set. Figure 4 shows our results. In our initial tests (Figure 4.a), fine-tuning the AraT5$_{\text{v2}}$ model exclusively on the $10,000$ sentences from syntheticGPT, registers an F_1 of 65.87, and reverseGPT an F_1 score of 46.85 falling behind the original QALB 2014 training data (which records an F_1 of 68.34). Following this, when fur-



(a) Base model trained on respective datasets

(b) Base model futher fine-tuned on original dataset

(c) Base model futher fine-tuned on reverseGPT

(d) Base Model futher fine-tuned on reverseGold

Figure 4: Scores of models fine-tuned on 10, 000 parallel sentences from different sources: Original training data, syntheticGPT, and reverseGPT evaluated on the QALB-2014 Test set.

ther fine-tuned on the original training set (Figure 4.b). We find that both syntheticGPT and the reverseGPT surpass model fine-tuned on equivalent-sized gold dataset, with F_1 of 69.01 and 68.54, respectively. This confirms the utility of ChatGPT for generating synthetic data. Conversely, when we further fine-tune the model with the two reverse noised datasets (see Figures 4.c and d), we observe a sharp decline in performance. This emphasizes the critical importance of relevant, high-quality synthetic data over randomly generated samples.

5.1 Decoding Methods.

Decoding strategies for text generation are essential and can vary based on the task (Zhang et al., 2023). We compare three decoding strategies to identify the best method for AGEC task. As shown in Table 3, we compare *greedy decoding* (Germann, 2003) (temperature=0), *Beam search* (Freitag and Al-Onaizan, 2017) (num_beams=5, temperature=1), and *Top-P sampling* (Holtzman et al., 2019) (top-p=0.8, top-k=75, and temperature=0.8). With the highest scoring strategy identified, we scale up our data augmentation experiments, by generating sets of 5million and 10million reverseGold datasets. In addition to these datasets, we utilize the complete AGEC dataset from Solyman et al. (2021) (referred to as AraT5$_{v2}$ (11M) in our experiments) for further evaluation.

Outlined in Table 4, AraT5$_{v2}$ shows consistent improvement as the number of training samples increases from 5M to 11M. Results indicate Top-P sampling is the best decoding method for GEC, exhibiting a balance between number of correct edits and total number of edits made.

Strategy	QALB-2014				QALB-2015			
	P	**R**	**F_1**	**$F_{0.5}$**	**P**	**R**	**F_1**	**$F_{0.5}$**
Greedy	$74.09^{\pm0.57}$	$65.63^{\pm0.59}$	$69.60^{\pm0.54}$	$72.23^{\pm0.55}$	$67.41^{\pm0.82}$	$66.85^{\pm0.97}$	$67.13^{\pm0.82}$	$67.30^{\pm0.80}$
Beam	$75.47^{\pm1.11}$	$68.61^{\pm1.26}$	$71.87^{\pm1.19}$	$73.99^{\pm1.14}$	$70.54^{\pm0.44}$	$68.04^{\pm0.14}$	$69.27^{\pm0.24}$	$70.03^{\pm0.35}$
Top-p	$76.94^{\pm0.67}$	$69.26^{\pm0.73}$	$72.90^{\pm0.68}$	$75.27^{\pm0.67}$	$72.64^{\pm0.32}$	$74.21^{\pm0.75}$	$73.41^{\pm0.51}$	$72.94^{\pm0.39}$

Table 3: Performance of AraT5$_{v2}$ (11M) on QALB-2014 and 2015 Test set under different decoding methods.

Datasets	QALB-2014				QALB-2015			
	P	**R**	**F_1**	**$F_{0.5}$**	**P**	**R**	**F_1**	**$F_{0.5}$**
M1	$71.35^{\pm0.14}$	$64.45^{\pm0.41}$	$67.73^{\pm0.17}$	$69.85^{\pm0.04}$	$69.65^{\pm0.57}$	$64.74^{\pm0.57}$	$67.11^{\pm0.14}$	$68.61^{\pm0.33}$
M2	$73.14^{\pm0.26}$	$67.48^{\pm1.07}$	$70.23^{\pm0.15}$	$72.37^{\pm1.05}$	$70.26^{\pm1.16}$	$65.74^{\pm1.37}$	$67.93^{\pm1.27}$	$69.31^{\pm1.20}$
M3	**76.94**$^{\pm0.67}$	**69.26**$^{\pm0.73}$	**72.90**$^{\pm0.68}$	**75.27**$^{\pm0.67}$	**72.64**$^{\pm0.32}$	**74.21**$^{\pm0.75}$	**73.41**$^{\pm0.51}$	**72.94**$^{\pm0.39}$

Table 4: Performance of AraT5$_{v2}$ models using the 'Top-P' decoding method on QALB-2014 and 2015 Test sets, on different amounts of training data. M1 : AraT5$_{v2}$ (5M), M2 : AraT5$_{v2}$ (10M), M3 : AraT5$_{v2}$ (11M)

6 Sequence Tagging Approach

In this section, we detail our methods to adapt the GECToR model (Omelianchuk et al., 2020) to experiment with the seq2edit approach.

Token-level transformations. We first perform token-level transformations on the source to recover the target text. ***'Basic-transformations'*** are applied to perform the most common token-level edit operations, such as keeping the current token unchanged (`$KEEP`), deleting current token (`$DELETE`), appending new token t_$_1$ next to the current token x_i (`$APPEND_t`$_1$) or replacing the current token x_i with another token t_$_2$ (`$REPLACE_t`$_2$). To apply tokens with more task-specific operations, we employ ***'g-transformations'***

Methods	Models	QALB-2014				QALB-2015			
		P	R	$F_{1.0}$	$F_{0.5}$	P	R	$F_{1.0}$	$F_{0.5}$
Seq2Seq	mT0	70.76 $^{\pm0.03}$	50.78 $^{\pm0.07}$	59.12 $^{\pm0.05}$	65.59 $^{\pm0.03}$	68.11 $^{\pm0.20}$	59.68 $^{\pm0.12}$	63.61 $^{\pm0.15}$	66.23 $^{\pm0.18}$
	mT5	70.64 $^{\pm0.12}$	50.16 $^{\pm0.05}$	58.66 $^{\pm0.05}$	65.30 $^{\pm0.09}$	68.20 $^{\pm0.10}$	59.02 $^{\pm0.15}$	63.28 $^{\pm0.04}$	66.14 $^{\pm0.11}$
	AraBART	70.71 $^{\pm0.06}$	60.46 $^{\pm0.04}$	65.18 $^{\pm0.07}$	68.39 $^{\pm0.08}$	68.39 $^{\pm0.09}$	67.95 $^{\pm0.02}$	65.62 $^{\pm0.05}$	66.76 $^{\pm0.07}$
	AraT5$_{v2}$	73.04 $^{\pm0.10}$	**63.09** $^{\pm0.15}$	**67.70** $^{\pm0.12}$	**70.81** $^{\pm0.11}$	71.40 $^{\pm0.90}$	**72.83** $^{\pm1.11}$	**72.11** $^{\pm0.99}$	**71.68** $^{\pm0.93}$
Seq2edit	ARBERTv2	73.89 $^{\pm0.35}$	48.33 $^{\pm0.33}$	58.43 $^{\pm0.35}$	66.82 $^{\pm0.35}$	73.10 $^{\pm0.29}$	55.40 $^{\pm1.15}$	63.03 $^{\pm0.86}$	68.70 $^{\pm0.56}$
	ARBERT$_{v2}$†	**74.39** $^{\pm0.22}$	47.62 $^{\pm0.30}$	58.07 $^{\pm0.29}$	66.87 $^{\pm0.26}$	**74.20** $^{\pm0.28}$	53.80 $^{\pm0.59}$	62.37 $^{\pm0.49}$	68.96 $^{\pm0.39}$
	MARBERT$_{v2}$	73.53 $^{\pm0.24}$	48.21 $^{\pm0.39}$	58.24 $^{\pm0.36}$	66.54 $^{\pm0.30}$	72.90 $^{\pm0.21}$	54.90 $^{\pm0.52}$	62.63 $^{\pm0.42}$	68.41 $^{\pm0.31}$
	MARBERT$_{v2}$†	74.21 $^{\pm0.16}$	46.45 $^{\pm0.25}$	57.14 $^{\pm0.24}$	66.29 $^{\pm0.20}$	74.00 $^{\pm0.17}$	52.70 $^{\pm0.34}$	61.56 $^{\pm0.29}$	68.46 $^{\pm0.23}$

Table 5: Performance of the seq2edit approach compared to baselines on the QALB-2014 and QALB-2015 Test sets. †: Models trained on 3-stage training.

Error Type	Incorrect Sentence	Correct Sentence
Orthographic	الرجل يرب الفرس . *The man rears the horse.*	الرجل يركب الفرس . *The man rides the horse.*
Punctuations	الرجل ، يركب الفرس . The man, rides the horse.	الرجل يركب الفرس . The man rides the horse.
Syntax	وجد رجلا يركب فرس . He found a man riding a hors.	وجد رجلا يركب فرسا . He found a man riding a horse.
Merge	غداالرجل سيركب الفرس . Tomorrowtheman will ride the horse.	غدا الرجل سيركب الفرس . Tomorrow the man will ride the horse.
Splits	غدا الرجل ير كب الفرس . The man ri des the horse.	غدا الرجل يركب الفرس . The man rides the horse.
Semantic	الرجل يجلس في ظهر الفرس . The man is sitting in the horse's back.	الرجل يجلس على ظهر الفرس . The man is sitting on the horse's back.
Morphological	غدا الرجل ركب الفرس . Tomorrow the man rode the horse.	غدا الرجل سيركب الفرس . Tomorrow the man will ride the horse.

Table 6: Examples of Merge, Morphological, Orthographic, Punctuation, Semantic, Split, and Syntactic errors, along with their corresponding corrections and English translations.

such as the ($MERGE) tag to merge the current token and the next token into a single one. Edit space after applying token-level transformations results in KEEP (725K op), $REPLACE_t$_2$ (201K op), $APPEND_t$_1$ (75K op), $DELETE (13K op), and $MERGE (5.7K op) tags.

Preprocessing and fine-tuning. We start the preprocessing stage by aligning source tokens with target subsequences, preparing them for token-level transformations. We then fine-tune ARBERT$_{v2}$ (Elmadany et al., 2022) and MARBERT$_{v2}$ (Abdul-Mageed et al., 2021) on the preprocessed data. We adhere to the training approach detailed in the original paper (Omelianchuk et al., 2020), adopting its three-stage training and setting the iterative correction to three. More details about the fine-tuning procedure can be found in Appendix C.

Sequence tagging evaluation. As shown in Table 5, ARBERT$_{v2}$ and MARBERT$_{v2}$, exhibit high precision (e.g., ARBERT$_{v2}$'s three-step training is at 74.39 precision). However, relatively lower recall scores indicate challenges in ability of the two models to detect errors. Unlike the findings in the original paper, our implementation of a three-stage training approach yields mixed results: while accuracy improves, recall scores decrease, leading to a drop in the overall F_1 score (by 0.36 for ARBERT$_{v2}$ and 1.10 for MARBERT$_{v2}$, respectively). Consequently, all models fall behind the 'seq2seq' models. We note that both ARBERT$_{v2}$ and MARBERT$_{v2}$ surpass mT0 and mT5 in terms of $F_{0.5}$ scores, highlighting their abilities in correcting errors with precision.

7 Error Analysis

7.1 Error type evaluation.

We use the Automatic Error Type Annotation (ARETA) tool (Belkebir and Habash, 2021) to assess our models' performance on different error types. We focus on seven errors types: *Orthographic*, *Morphological*, *Syntactic*, *Semantic*, *Punctuation*, *Merge*, and *Split*. Examples of each error types alongside their translations can be found in Table 6. We examine top models from each approach, including ARBERT$_{v2}$ (3-step), GPT-4 (5-shot) + CoT, and AraT5$_{v2}$(11M). Figure 5 illustrates the performance of selected models under each error type. AraT5$_{v2}$(11M), surpasses all other models across all error categories. In particular, it excels in handling *Orthographic* (ORTH) errors, *Morphological* (MORPH) errors, and *Punctuation* (PUNCT) errors, consistently achieving over 65 F_1 score. However, it is worth observing that all models encounter challenges with *Semantic* (SEM) and *Syntactic* (SYN) errors. These disparate outcomes underscore the significance of selecting the appropriate model based on the error types prevalent in a specific dataset.

7.2 Normalization methods.

In addition to the *'Exact Match'* score, we also analyze system performance under different normalization methods. Namely, we assess the system on normalized text (1) without Alif/Ya errors, (2) without punctuation, and (3) free from both Alif/Ya and punctuation errors. Examples of text under each setting can be found in Appendix D.1.

7.3 Normalisation results

Looking at Table 7, in the 'No punctuation' setting, all models perform better, reflecting models' limitations in handling punctuation which is due to absence of clearly agreed upon punctuation rules in Arabic. Moreover, the datasets used are based on commentaries where punctuation is inherently inconsistent and varied. Another noteworthy obser-

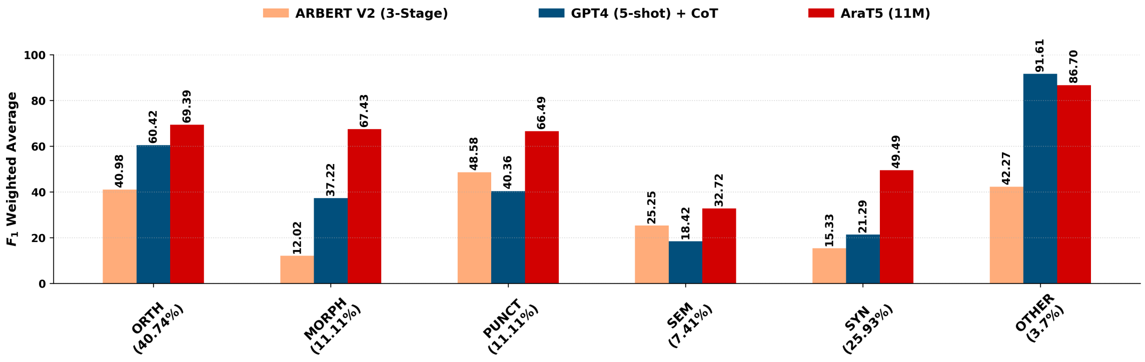


Figure 5: Best model F_1 scores for each approach on specific error types in the QALB-2014 Test set.

Test Set	Models	Exact Match				No Alif / Ya Errors				No Punctuation				No Punctuation and Alif / Ya Errors			
		P	R	$F_{1.0}$	$F_{0.5}$	P	R	$F_{1.0}$	$F_{0.5}$	P	R	$F_{1.0}$	$F_{0.5}$	P	R	$F_{1.0}$	$F_{0.5}$
QALB-2014	Solyman et al. (2021)	**79.06**	65.79	71.82	**75.99**	-	-	-	-	-	-	-	-	-	-	-	-
	Mohit et al. (2014)	73.34	63.23	67.91	71.07	**64.05**	50.86	56.7	**60.89**	76.99	49.91	60.56	69.45	76.99	49.91	60.56	69.45
	GPT4 (5-shot)	69.46	61.96	65.49	67.82	58.44	51.47	54.73	56.90	74.59	78.15	76.33	75.28	60.06	65.75	62.78	61.12
	ARBERT$_{v2}$ (3-step)	$74.17^{\pm0.22}$	$47.34^{\pm0.30}$	$57.79^{\pm0.29}$	$66.62^{\pm0.26}$	$64.90^{\pm0.57}$	$41.86^{\pm0.24}$	$50.89^{\pm0.17}$	$58.46^{\pm0.33}$	$76.90^{\pm0.85}$s	$46.33^{\pm0.58}$	$57.83^{\pm0.66}$	$67.94^{\pm0.75}$	$56.66^{\pm0.57}$	$29.30^{\pm0.61}$	$38.62^{\pm0.39}$	$47.74^{\pm0.03}$
	AraT5$_{v2}$ (11m)	$76.94^{\pm0.67}$	**69.26**$^{\pm0.73}$	**72.90**$^{\pm0.68}$	$75.27^{\pm0.67}$	$62.42^{\pm0.68}$	**52.56**$^{\pm0.51}$	**57.06**$^{\pm0.08}$	$60.16^{\pm0.38}$	**86.52**$^{\pm0.50}$	**82.90**$^{\pm0.17}$	**84.67**$^{\pm0.25}$	**85.77**$^{\pm0.39}$	**79.44**$^{\pm0.51}$	**67.40**$^{\pm0.53}$	**72.92**$^{\pm0.52}$	**76.70**$^{\pm0.52}$
QALB-2015	Solyman et al. (2021)	**80.23**	63.59	70.91	**76.24**	-	-	-	-	-	-	-	-	-	-	-	-
	Rozovskaya et al. (2015a)	88.85	61.76	72.87	81.68	**84.25**	43.29	**57.19**	**70.84**	**85.8**	**77.98**	**81.7**	**84.11**	**80.12**	**58.24**	**67.45**	**74.52**
	ChatGPT (3-shot) + EP	52.33	47.57	49.83	54.10	37.93	39.97	38.92	32.95	53.38	56.63	54.96	54.00	33.33	46.77	38.92	35.36
	ARBERT$_{v2}$ (3-step)	$73.92^{\pm0.28}$	$53.15^{\pm0.59}$	$61.84^{\pm0.49}$	$68.56^{\pm0.39}$	$57.14^{\pm0.21}$	$39.17^{\pm0.76}$	$46.47^{\pm0.47}$	$52.34^{\pm0.13}$	$66.90^{\pm0.17}$	$61.50^{\pm0.50}$	$64.09^{\pm0.28}$	$65.74^{\pm0.18}$	$71.18^{\pm0.16}$	$39.00^{\pm0.87}$	$50.39^{\pm0.75}$	$61.09^{\pm0.49}$
	AraT5$_{v2}$ (11m)	$72.10^{\pm0.31}$	**73.59**$^{\pm0.70}$	**72.84**$^{\pm0.40}$	$72.40^{\pm0.30}$	$55.80^{\pm0.30}$	**43.51**$^{\pm0.50}$	$48.89^{\pm0.22}$	$52.81^{\pm0.11}$	$85.82^{\pm0.31}$	$72.85^{\pm0.25}$	$78.81^{\pm0.28}$	$82.87^{\pm0.30}$	$75.08^{\pm0.13}$	$53.30^{\pm0.93}$	$62.34^{\pm0.60}$	$69.40^{\pm0.26}$

Table 7: Results on QALB-2014, QALB-2015 Test sets under Normalization Methods.

vation is the drop in F_1 scores when Alif/Ya errors are removed. This can be attributed to the fact that Alif/Ya errors are relatively simpler compared to other error categories. Moreover, AraT5$_{v2}$ is trained on formal texts such as AraNews (Nagoudi et al., 2020) and Hindawi Books [2], which contain proper Alif/Ya indicating the model's proficiency with the correct usage of these letters.

8 Discussion

LLMs and ChatGPT. ChatGPT demonstrates remarkable ability to outperform other fully trained models by learning from only a few examples, particularly five-shot under both few-shot CoT and EP prompting strategies. Nevertheless, ChatGPT's performance lags behind AraT5$_{v2}$ and AraBART, suggesting potential areas for improvements in prompting strategies to fully exploit ChatGPT models. Models such as Vicuna-13B as well as those trained on multilingual datasets like Bactrian-X$_{llama}$-7B and Bactrian-X$_{bloom}$-7B, tend to perform better. However, these models fail to match ChatGPT's performance which reinforces ChatGPT's superiority in this domain.

Seq2seq approach. Despite being smaller in size, pretrained Language Models (PLMs) often outperform LLMs, especially models specifically trained for Arabic tasks, such as AraT5$_{v2}$ and AraBART. In contrast, mT0 and mT5, both of which are multilingual models, are surpassed by ChatGPT when using both prompting strategies from 3-shot, but still outperform smaller LLMs such as LLaMA, Alpaca and Vicuna. Moreover, the results underscore the advantages of synthetic data for PLMs, as evidenced by the consistent improvement in scores with additional data.

Seq2edit approach. These models exhibit high precision scores and relatively low recall scores, suggesting their strengths in making corrections rather than detecting errors. This trend can be explained by the absence of *g-transformations*. For instance, in the case of English GECToR models, *g-transformations* enable a variety of changes, such as case alterations and grammatical transformations. However, in this work we only rely on the 'merge' *g-transformations* from the GECToR model as crafting effective *g-transformations* for Arabic, a language with rich morphological features, poses significant challenges, limiting the model's ability to effectively detect errors. Developing specific *g-transformations* for Arabic could significantly improve performance in these models.

Data augmentation. Data augmentation results underscore the potential of synthetic data, generated by ChatGPT, in enhancing model performance. Our findings reveal that not just the quantity, but the quality of synthetic data, is crucial for achieving optimal performance. The relative underperfor-

[2] www.hindawi.org/books

Test Set	Models	Exact Match			
		P	R	$F_{1.0}$	$F_{0.5}$
QALB-2014	Solyman et al. (2021)	**79.06**	65.79	71.82	**75.99**
	Mohit et al. (2014)	73.34	63.23	67.91	71.07
	GPT4 (5-shot)	69.46	61.96	65.49	67.82
	ARBERT$_{v2}$ (3-step)	$74.17^{\pm0.22}$	$47.34^{\pm0.30}$	$57.79^{\pm0.29}$	$66.62^{\pm0.26}$
	AraT5$_{v2}$ (11m)	$76.94^{\pm0.67}$	$\mathbf{69.26}^{\pm0.73}$	$\mathbf{72.90}^{\pm0.68}$	$75.27^{\pm0.67}$
QALB-2015	Solyman et al. (2021)	**80.23**	63.59	70.91	**76.24**
	Rozovskaya et al. (2015a)	88.85	61.76	72.87	81.68
	ChatGPT (3-shot) + EP	52.33	47.57	49.83	54.10
	ARBERT$_{v2}$ (3-step)	$73.92^{\pm0.28}$	$53.15^{\pm0.59}$	$61.84^{\pm0.49}$	$68.56^{\pm0.39}$
	AraT5$_{v2}$ (11m)	$72.10^{\pm0.31}$	$\mathbf{73.59}^{\pm0.70}$	$\mathbf{72.84}^{\pm0.40}$	$72.40^{\pm0.30}$

Table 8: Results on QALB-2014, QALB-2015 Test sets compared to recent works.

mance of models further trained with synthetically generated data examples emphasizes this conclusion. Improvements we observe when expanding the dataset from 5M to 10M and from 10M to 11M are similar, even though the quantity of additional data vary. This can be attributed to the quality of the sources as the data for 5M and 10M were derived from noisier online commentaries, while the 11M data was derived from the OSIAN corpus (Zeroual et al., 2019). Furthermore, our results on decoding methods on scaled datasets indicate that the chosen method can significantly influence precision and recall, emphasizing the need to select the right method depending on the specific task at hand.

Best model in comparison. Although our main objective is not to develop the best model for AGEC, our AraT5$_{v2}$ (11M) model as detailed in Table 8 excels in comparison to previous SOTA (71.82 vs. 72.90). It is worth noting that contemporaneous work by Alhafni et al. (2023) introduces a new alignment algorithm that is much better than that employed by the shared task evaluation code we use. They also present an AGEC model. In personal communication with the authors, they confirmed their alignment algorithm through which we can perform direct and fair comparisons, and the data split on ZAEBUC dataset (Habash and Palfreyman, 2022) will be released once their work is published through peer-review. Different from their work, our models are also dependency-free. For example, we do not exploit any morphological analyzers.

9 Conclusion

This paper provided a detailed exploration of the potential of LLMs, with a particular emphasis on ChatGPT for AGEC. Our study highlights ChatGPT's promising capabilities, in low-resource scenarios, as evidenced by its competitive performance on few-shot setttings. However, AraT5$_{v2}$ and AraBART still exhibit superior results across various settings and error types. Our findings also emphasize the role of high-quality synthetic data, reinforcing that both quantity and quality matter in achieving optimal performance. Moreover, our work unveils trade-offs between precision and recall in relation to dataset size and throughout all the other experimental settings. These insight, again, could inform future strategies for improving GEC systems. Although our exploration of ChatGPT's performance on AGEC tasks showcases encouraging results, it also uncovers areas ripe for further study. Notably, there remains significant room for improvement in GEC systems, particularly within the context of low-resource languages. Future research may include refining prompting strategies, enhancing synthetic data generation techniques, and addressing the complexities and rich morphological features inherent in the Arabic language.

10 Limitations

We identify the following limitations in this work:

1. This work is primarily focused on MSA and does not delve into dialectal Arabic (DA) or the classical variety of Arabic (CA). While there exist DA resources such as the MADAR corpus (Bouamor et al., 2018), their primary application is for dialect identification (DID) and machine translation (MT), making them unsuitable for our specific AGEC objectives. A more comprehensive coverage could be achieved with the development and introduction of datasets specifically tailored for the dialects in question.

2. This work aimed to examine the potential of LLMs, with an emphasis on ChatGPT, by comparing them to fully pretrained models. However, uncertainty surrounding the extent of Arabic data on which ChatGPT has been trained, poses challenges for direct comparisons with other pretrained models. Additionally, LLMs are primarily fine-tuned for English-language data. While prior studies have demonstrated their effectiveness in other languages, the limited amount of pretraining data for non-English languages complicates a straightforward comparison.

3. The scope of this work is primarily centered on sentence-level GEC. This limitation arose due to the official ChatGPT API, at the time

of our study, allowed a maximum of 4,097 tokens, making it unsuitable for longer texts and precluding document-level GEC tasks. However, it's worth noting that document-level correction, offers a broader context that's vital for addressing certain grammatical inconsistencies and errors (Yuan and Bryant, 2021). With the recent introduction of a newer API that accommodates extended texts, future endeavors can potentially address document-level GEC, utilizing datasets such as QALB-2015 L2 and the newly introduced ZAEBUC corpus.

11 Ethics Statement and Broad Impact

Encouraging research development and contributing to a collaborative research culture. Progress in AGEC has been stagnant for a long time due to the lack of benchmark datasets. This can be attributed to the extensive time and cost involved in creating these datasets. As a result, advancing AGEC has proven challenging. With the recent development of LLMs and their capabilities, there is potential for these models to expedite the creation of datasets. By doing so, they can significantly reduce both time and cost, as has been observed in other languages. We hope our work will inspire further exploration into the capabilities of LLMs for AGEC, thus aiding in the progress of this field.

Advancing Second Language Learning through LLMs. With increasing interest in second language learning, ensuring accuracy and effectiveness of written language has become significant for pedagogical tools. Nowadays, individuals treat LLMs as their own writing assistants. Therefore, LLMs in the context of educational applications and more specifically GEC is becoming increasingly important. As such, introducing works in the development of tools that aid assistance in writing can help bridge the gap between non-native speakers and fluent written communication, enhancing the efficacy of educational tools. Especially with Arabic, being a collection of a diverse array of languages and dialectal varieties, we hope this will inspire more work to ensure comprehensive coverage and improved support for all learners. However, it is crucial to emphasize the ethical implications of using AI-driven educational tools. It's essential that these tools remain unbiased, transparent, and considerate of individual learning differences, ensuring the trustworthiness and integrity of educational platforms for every learner.

Data privacy. In relation to the data used in this work, all datasets are publicly available. Therefore, we do not have privacy concerns.

Acknowledgments

We acknowledge support from Canada Research Chairs (CRC), the Natural Sciences and Engineering Research Council of Canada (NSERC; RGPIN-2018-04267), the Social Sciences and Humanities Research Council of Canada (SSHRC; 435-2018-0576; 895-2020-1004; 895-2021-1008), Canadian Foundation for Innovation (CFI; 37771), Digital Research Alliance of Canada,[3] and UBC Advanced Research Computing-Sockeye.[4]

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.

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, Barcelona, Spain (Online). Association for Computational Linguistics.

Abdullah Alfaifi and Eric Atwell. 2012. Arabic learner corpora (alc): A taxonomy of coding errors. In *The 8th International Computing Conference in Arabic*.

Bashar Alhafni, Go Inoue, Christian Khairallah, and Nizar Habash. 2023. Advancements in arabic grammatical error detection and correction: An empirical investigation. *arXiv preprint arXiv:2305.14734*.

Abhijeet Awasthi, Sunita Sarawagi, Rasna Goyal, Sabyasachi Ghosh, and Vihari Piratla. 2019. Parallel iterative edit models for local sequence transduction. 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 4260–4270, Hong Kong, China. Association for Computational Linguistics.

Riadh Belkebir and Nizar Habash. 2021. Automatic error type annotation for arabic. *arXiv preprint arXiv:2109.08068*.

[3] https://alliancecan.ca

[4] https://arc.ubc.ca/ubc-arc-sockeye

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).

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

Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019. The BEA-2019 shared task on grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75, Florence, Italy. Association for Computational Linguistics.

Christopher Bryant, Zheng Yuan, Muhammad Reza Qorib, Hannan Cao, Hwee Tou Ng, and Ted Briscoe. 2022. Grammatical error correction: A survey of the state of the art. *arXiv preprint arXiv:2211.05166*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Shamil Chollampatt and Hwee Tou Ng. 2018. A multilayer convolutional encoder-decoder neural network for grammatical error correction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Daniel Dahlmeier and Hwee Tou Ng. 2012. Better evaluation for grammatical error correction. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 568–572, Montréal, Canada. Association for Computational Linguistics.

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

AbdelRahim Elmadany, El Moatez Billah Nagoudi, and Muhammad Abdul-Mageed. 2022. Orca: A challenging benchmark for arabic language understanding. *arXiv preprint arXiv:2212.10758*.

Tao Fang, Shu Yang, Kaixin Lan, Derek F Wong, Jinpeng Hu, Lidia S Chao, and Yue Zhang. 2023. Is chatgpt a highly fluent grammatical error correction system? a comprehensive evaluation. *arXiv preprint arXiv:2304.01746*.

Mariano Felice, Zheng Yuan, Øistein E. Andersen, Helen Yannakoudakis, and Ekaterina Kochmar. 2014. Grammatical error correction using hybrid systems and type filtering. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 15–24, Baltimore, Maryland. Association for Computational Linguistics.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988, Online. Association for Computational Linguistics.

Simon Flachs, Felix Stahlberg, and Shankar Kumar. 2021. Data strategies for low-resource grammatical error correction. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 117–122, Online. Association for Computational Linguistics.

Markus Freitag and Yaser Al-Onaizan. 2017. Beam search strategies for neural machine translation. *arXiv preprint arXiv:1702.01806*.

Ulrich Germann. 2003. Greedy decoding for statistical machine translation in almost linear time. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 72–79.

Peiyuan Gong, Xuebo Liu, Heyan Huang, and Min Zhang. 2022. Revisiting grammatical error correction evaluation and beyond. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6891–6902, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Kenneth Heafield. 2019. Neural grammatical error correction systems with unsupervised pre-training on synthetic data. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 252–263, Florence, Italy. Association for Computational Linguistics.

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.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Serena Jeblee, Houda Bouamor, Wajdi Zaghouani, and Kemal Oflazer. 2014. CMUQ@QALB-2014: An SMT-based system for automatic Arabic error correction. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 137–142, Doha, Qatar. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Shubha Guha, and Kenneth Heafield. 2018a. Approaching neural grammatical error correction as a low-resource machine translation task. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 595–606, New Orleans, Louisiana. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Shubha Guha, and Kenneth Heafield. 2018b. Approaching neural grammatical error correction as a low-resource machine translation task. *arXiv preprint arXiv:1804.05940*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *arXiv preprint arXiv:2205.11916*.

Haonan Li, Fajri Koto, Minghao Wu, Alham Fikri Aji, and Timothy Baldwin. 2023. Bactrian-x: A multilingual replicable instruction-following model. `https://github.com/MBZUAI-nlp/Bactrian-X`.

Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn. 2019. Encode, tag, realize: High-precision text editing. *arXiv preprint arXiv:1909.01187*.

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.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2022. Crosslingual generalization through multitask finetuning.

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.

El Moatez Billah Nagoudi, AbdelRahim Elmadany, Muhammad Abdul-Mageed, Tariq Alhindi, and Hasan Cavusoglu. 2020. Machine generation and detection of arabic manipulated and fake news. *arXiv preprint arXiv:2011.03092*.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The CoNLL-2014 shared task on grammatical error correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–14, Baltimore, Maryland. Association for Computational Linguistics.

Hwee Tou Ng, Siew Mei Wu, Yuanbin Wu, Christian Hadiwinoto, and Joel Tetreault. 2013. The CoNLL-2013 shared task on grammatical error correction. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–12, Sofia, Bulgaria. Association for Computational Linguistics.

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 2020. Gector–grammatical error correction: tag, not rewrite. *arXiv preprint arXiv:2005.12592*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

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. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. 2021. A simple recipe for multilingual grammatical error correction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 702–707, Online. Association for Computational Linguistics.

Alla Rozovskaya, Houda Bouamor, Nizar Habash, Wajdi Zaghouani, Ossama Obeid, and Behrang Mohit. 2015a. The second qalb shared task on automatic text correction for arabic. In *Proceedings of the Second workshop on Arabic natural language processing*, pages 26–35.

Alla Rozovskaya, Houda Bouamor, Nizar Habash, Wajdi Zaghouani, Ossama Obeid, and Behrang Mohit. 2015b. The second QALB shared task on automatic text correction for Arabic. In *Proceedings of the Second Workshop on Arabic Natural Language Processing*, pages 26–35, Beijing, China. Association for Computational Linguistics.

Alla Rozovskaya, Nizar Habash, Ramy Eskander, Noura Farra, and Wael Salloum. 2014. The Columbia system in the QALB-2014 shared task on Arabic error correction. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 160–164, Doha, Qatar. Association for Computational Linguistics.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*.

Aiman Solyman, Zhenyu Wang, Qian Tao, Arafat Abdulgader Mohammed Elhag, Rui Zhang, and Zeinab Mahmoud. 2022. Automatic arabic grammatical error correction based on expectation-maximization routing and target-bidirectional agreement. *Knowledge-Based Systems*, 241:108180.

Aiman Solyman, Marco Zappatore, Wang Zhenyu, Zeinab Mahmoud, Ali Alfatemi, Ashraf Osman Ibrahim, and Lubna Abdelkareim Gabralla. 2023. Optimizing the impact of data augmentation for low-resource grammatical error correction. *Journal of King Saud University - Computer and Information Sciences*, 35(6):101572.

Aiman Solyman, Wang Zhenyu, Tao Qian, Arafat Abdulgader Mohammed Elhag, Muhammad Toseef, and Zeinab Aleibeid. 2021. Synthetic data with neural machine translation for automatic correction in arabic grammar. *Egyptian Informatics Journal*, 22(3):303–315.

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

Maksym Tarnavskyi, Artem Chernodub, and Kostiantyn Omelianchuk. 2022. Ensembling and knowledge distilling of large sequence taggers for grammatical error correction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3842–3852, Dublin, Ireland. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Daniel Watson, Nasser Zalmout, and Nizar Habash. 2018. Utilizing character and word embeddings for text normalization with sequence-to-sequence models. *arXiv preprint arXiv:1809.01534*.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022a. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022b. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

Haoran Wu, Wenxuan Wang, Yuxuan Wan, Wenxiang Jiao, and Michael Lyu. 2023. Chatgpt or grammarly? evaluating chatgpt on grammatical error correction benchmark. *arXiv preprint arXiv:2303.13648*.

Ziang Xie, Guillaume Genthial, Stanley Xie, Andrew Ng, and Dan Jurafsky. 2018. Noising and denoising natural language: Diverse backtranslation for grammar correction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 619–628, New Orleans, Louisiana. Association for Computational Linguistics.

Benfeng Xu, An Yang, Junyang Lin, Quan Wang, Chang Zhou, Yongdong Zhang, and Zhendong Mao. 2023. Expertprompting: Instructing large language models to be distinguished experts. *arXiv preprint arXiv:2305.14688*.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*.

Zheng Yuan and Christopher Bryant. 2021. Document-level grammatical error correction. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 75–84, Online. Association for Computational Linguistics.

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).

Imad Zeroual, Dirk Goldhahn, Thomas Eckart, and Abdelhak Lakhouaja. 2019. OSIAN: Open source international Arabic news corpus - preparation and integration into the CLARIN-infrastructure. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 175–182, Florence, Italy. Association for Computational Linguistics.

Renrui Zhang, Jiaming Han, Chris Liu, Peng Gao, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, and Yu Qiao. 2023. Llama-adapter: Efficient fine-tuning of language models with zero-init attention.

A Related Works

Sequence to sequence approach. Transformer-based Language Models (LMs) have been integral to advancements in GEC. These models have substantially transformed the perception of GEC, reframing it as a MT task. In this framework, erroneous sentences are considered as the source language, and the corrected versions as the target language. This perspective, which has led to SOTA results in the CONLL 2013 and 2014 shared tasks (Bryant et al., 2022; Ng et al., 2013, 2014), reinterprets GEC as a low-resource or mid-resource MT task. Building on this paradigm, Junczys-Dowmunt et al. (2018a) successfully adopted techniques from low-resource NMT and Statistical Machine Translation (SMT)-based GEC methods, leading to considerable improvements on both the CONLL and JFLEG datasets.

Sequence tagging approach. Sequence tagging methods, another successful route to GEC, are showcased by models like GECToR (Omelianchuk et al., 2020), LaserTagger (Malmi et al., 2019), and the Parallel Iterative Edit (PIE) model (Awasthi et al., 2019). By viewing GEC as a text editing task, these models make edits predictions instead of tokens, label sequences rather than generating them, and iteratively refine predictions to tackle dependencies. Employing a limited set of output tags, these models apply edit operations on the input sequence, reconstructing the output. This technique not only capably mirrors a significant chunk of the target training data, but it also diminishes the vocabulary size and establishes the output length as the source text's word count. Consequently, it curtails the number of training examples necessary for model accuracy, which is particularly beneficial in settings with sparse human-labeled data (Awasthi et al., 2019).

Instruction fine-tuning. LLMs have revolutionized NLP, their vast data-learning capability enabling diverse task generalizations. Key to their enhancement has been instructional finetuning, which fortifies the model's directive response and mitigates the need for few-shot examples (Ouyang et al., 2022; Wei et al., 2022b; Sanh et al., 2021). A novel approach, Chain of Thought (CoT), directs LLMs through a series of natural language reasoning, generating superior outputs. Proven beneficial in 'Let's think step by step' prompts (Wei et al., 2022b), CoT has harnessed LLMs for multi-task cognitive tasks (Kojima et al., 2022) and achieved SOTA results in complex system-2 tasks like arithmetic and symbolic reasoning.

ChatGPT. In the specific realm of GEC, LLMs have demonstrated its potential. Fang et al. (2023) applied zero-shot and few-shot CoT settings using in-context learning for ChatGPT (Brown et al., 2020) and evaluated its performance on three document-level English GEC test sets. Similarly, Wu et al. (2023) carried out an empirical study to assess the effectiveness of ChatGPT in GEC, in the CoNLL2014 benchmark dataset.

Development in AGEC Arabic consists of a collection of diverse languages and dialectal varieties with Modern Standard Arabic (MSA) being the current standard variety used in government and pan-arab media as well as education (Abdul-Mageed et al., 2020). The inherent ambiguity of Arabic at the orthographic, morphological, syntactic, and semantic levels makes AGEC particularly challenging. Optional use of diacritics further introduces orthographic ambiguity (Belkebir and Habash, 2021), making AGEC even harder.

Despite these hurdles, progress has been made in AGEC. For dataset development, the QALB corpus (Zaghouani et al., 2014) was utilized. During the QALB-2014 and 2015 shared tasks (Mohit et al., 2014; Rozovskaya et al., 2015b), the first AGEC datasets containing comments and documents from both native (L1) and Arabic learner (L2) speakers were released. Furthermore, the more recent ZAEBUC corpus (Habash and Palfreyman, 2022), which features essays from first-year university students at Zayed University in the UAE, has also been released. There has also been work on generating synthetic data. Solyman et al. (2021, 2023) apply Convolutional neural network (CNN) to generate synthetic parallel data using unsupervised noise injection techniques showing improvements in the QALB-2014 and 2015 benchmark datasets. In terms of model development, Watson et al. (2018) developed a character-level seq2seq model that achieved notable results on AGEC L1 data, marking prgoress from basic classifier models (Rozovskaya et al., 2014) and statistical machine translation models (Jeblee et al., 2014). More recently, Solyman et al. (2022, 2021) introduced novel design that incorporates dynamic linear combinations and the EM routing algorithm within a seq2seq Transformer framework.

B Instruction Fine-tuning LLMs

B.1 Instructions for LLMs

Instruction format used for training is provided in Table 9 and instructions used for training are shown in Table 10.

B.2 Baseline and experimental setup for LLMs and ChatGPT

For LLMs, evaluation was only done on the QALB-2014 Test set, for two main reasons. First was due to the high cost in producing results using ChatGPT and we were able to observation of a similar trend in our preliminary experiment with ChatGPT-3.5 Turbo on the QALB-2015. Second, as instruction fine-tuned were predominantly compared against ChatGPT's performance, we also evaluate them only on the QALB-2014 Test set. These Results can be found in Table 11.

C Sequence Tagging Approach

The training procedure detailed in the original GECToR paper (Omelianchuk et al., 2020) encompasses three stages:

1. Pre-training on synthetically generated sentences with errors.

2. Fine-tuning solely on sentences that contain errors.

3. Further fine-tuning on a mix of sentences, both with and without errors.

For our training process, we pre-train the model on the complete AGEC dataset (Solyman et al., 2021), use the reverseGold dataset for stage 2, and employ the gold training data in the third stage. Moreover, as some corrections in a sentence depend on others, applying edit sequences once may not be enough to correct the sentence fully. To address this issue, GECToR employs an iterative correction approach from Awasthi et al. (2019). However, in our experiments, we find that the iterative correction approach does not result in any tangible improvement. Therefore, we set our iterations to 3.

D Normalization Methods

D.1 Normalization examples

Examples of text under each normalization methods can be found in Table 12

D.2 Arabic Learner Corpus error type taxonomy

The ALC error type taxonomy can be found in Table 13.

D.3 Hyperparameters

The Hyperparameters used for training are shown in Table 14.

D.4 Dev results

Results on the Dev set are presented in Table 15.

D.5 ARETA results

Full results evaluated using ARETA are presented in Table 16.

Fine-tune Instruction Example
فيما يلي أمر توجيه يصف مهمة مرتبطة بمدخل لتزويد النص بسياق اضافي. يرجى صياغة ردود مناسبة لتحقق الطلب بطريقة مناسبة و دقيقة.
الأمر التوجيه :
قم بتصحيح كل الأخطاء الكتابية في النص التالي:
###المدخل :
الرجل يرب الفرس .
###الرد :
الرجل يركب الفرس .

Table 9: Modified data format for the LLaMA instruction fine-tuning step.

Translated in English	Instructions Samples
Correct all written errors in the following text except for a thousand, ya and punctuation:	قم بتصحيح كل الأخطاء الكتابية في النص التالي ماعدا المتعلقة بالألف والياء وعلامات الترقيم :
Please verify spelling, grammatical scrutiny, and correct all errors in the following sentence, except for punctuation:	الرجاء التدقيق الإملائي والتدقيق النحوي و تصحيح كل الأخطاء في الجملة التالية إلا الخاصة بعلامات الترقيم :
Explore the grammatical errors and repair them except for punctuation marks such as a comma, or a question marks, etc:	قم بإستكشاف أخطاء التدقيق الإملائي وإصلاحها ماعدا المتعلقة بعلامات الترقيم كالفاصلة أو علامة إستفهام ، إلخ :
Can you correct all errors in the following text except those related to punctuation such as commas, periods, etc:	هل يمكنك كل الأخطاء الموجودة في النص التالي ماعدا المتعلقة بعلامات الترقيم كالفاصلة ، النقطة ، إلخ :
Can you fix all spelling and grammatical errors, except for the mistakes of the "Alif" and "Ya":	هل يمكنك إصلاح كل الأخطاء الإملائية والنحوية ماعدا الأخطاء الخاصة بالألف والياء:
Please explore the grammatical spelling errors and repair them all, except for the mistakes related to the "Alif" and "Ya"	الرجاء إستكشاف أخطاء التدقيق الإملائي النحوي وإصلاحها كلها ماعدا الأخطاء المتعلقة بالألف والياء:
Correct all the written errors in the following text except for the "Alif" and "Ya":	قم بتصحيح كل الأخطاء الكتابية في النص التالي ماعدا المتعلقة بالألف والياء:
Please correct all errors in the following sentence:	الرجاء تصحيح كل الأخطاء الموجودة في الجملة التالية:

Table 10: Different instructions used for instruction fine-tuning.

Settings	Models	Exact Match			
		P	R	$F_{1.0}$	$F_{0.5}$
+ CoT	ChatGPT (3-shot)	49.89	46.72	48.22	49.49
	ChatGPT (5-shot)	52.33	47.57	49.83	51.15

Table 11: Performance of ChatGPT-3.5 on QALB-2015 Test set.

Normalisation Method	Example
Normal	نحن معشر العرب نعرف إلا الشماتة ، ولكن يجب أن ندرس هذه الحالة و نحن المخرج منها من الاقتصاد الإسلامي.
No Alif/Ya	نحن معشر العرب نعرف الا الشماتة ، ولكن يجب ان ندرس هذه الحالة و نحن المخرج منها من الاقتصاد الاسلامي.
No Punct	نحن معشر العرب نعرف إلا الشماتة ولكن يجب أن ندرس هذه الحالة و نحن المخرج منها من الاقتصاد الإسلامي
No Alif/Ya & Punct	نحن معشر العرب نعرف الا الشماتة ولكن يجب ان ندرس هذه الحالة و نحن المخرج منها من الاقتصاد الاسلامي

Table 12: Examples of normalized text: with Alif/Ya errors removed, punctuation removed, and both Alif/Ya errors and punctuation removed.

Class	Sub-class	Description
Orthographic	**OH**	**Hamza error**
	OT	Confusion in Ha and Ta Mutadarrifatin
	OA	**Confusuion in Alif and Ya Mutadarrifatin**
	OW	Confusion in Alif Fariqa
	ON	Confusion Between Nun and Tanwin
	OS	Shortening the long vowels
	OG	Lengthening the short vowels
	OC	Wrong order of word characters
	OR	Replacement in word character(s)
	OD	Additional character(s)
	OM	Missing character(s)
	OO	Other orthographic errors
Morphological	**MI**	Word inflection
	MT	Verb tense
	MO	Other morphological errors
	XF	Definiteness
	XG	Gender
	XN	Number
	XT	Unnecessary word
	XM	Missing word
	XO	Other syntactic errors
Semantic	**SW**	Word selection error
	SF	Fasl wa wasl (confusion in conjunction use/non-use)
	SO	Other semantic errors
Punctuation	**PC**	Punctuation confusion
	PT	Unnecessary punctuation
	PM	Missing punctuation
	PO	Other errors in punctuation
Merge	**MG**	Words are merged
Split	**SP**	Words are split

Table 13: The ALC error type taxonomy extended with merge and split classes

Hyperparameter	**Seq2seq**	**Decoder Only (LLMs)**	**Seq2Edit Encoder Only**
Learning Rate	5×10^{-5}	2×10^{-5}	1×10^{-5}
Train Batch Size	4	8	8
Eval Batch Size	4	8	8
Seed	42	42	42
Gradient Accumulation Steps	8	8	8
Total Train Batch Size	32	64	64
Optimizer	Adam (betas=(0.9,0.999), epsilon=1 $\times 10^{-8}$)	AdamW (betas=(0.9,0.999), epsilon=1 $\times 10^{-7}$)	AdamW (betas=(0.9,0.999), epsilon=1 $\times 10^{-8}$)
LR Scheduler Type	Cosine	Linear	Cosine
Num Epochs	50	4	100

Table 14: Summary of hyperparameters used for model training.

Settings	**Models**	**Exact Match**				**No Alif / Ya Errors**				**No Punctuation**				**No Puncation and Alif / Ya Errors**			
		P	**R**	**$F_{1.0}$**	**$F_{0.5}$**	**P**	**R**	**$F_{1.0}$**	**$F_{0.5}$**	**P**	**R**	**$F_{1.0}$**	**$F_{0.5}$**	**P**	**R**	**$F_{1.0}$**	**$F_{0.5}$**
Seq2Edit	ARBERTv2	73.30	47.85	57.90	66.25	65.60	44.20	52.81	59.81	72.38	48.75	58.26	65.98	57.40	33.90	42.63	50.41
	ARBERT$_{v2}$ 3-stage	74.65	46.70	57.46	66.67	65.00	41.20	50.43	58.27	75.50	44.50	56.00	66.27	55.70	27.50	36.82	46.22
	MARBERT$_{v2}$	72.95	47.65	57.65	65.95	64.60	43.20	51.78	58.78	73.72	44.16	55.23	65.02	56.80	34.20	42.69	50.17
	MARBERT$_{v2}$ 3-stage	74.55	45.75	56.70	66.21	65.10	41.30	50.54	58.37	75.41	45.52	56.77	66.66	56.00	29.20	38.38	47.31
LLMs	LLama-7B	58.20	32.50	41.71	50.25	35.50	16.70	22.71	28.98	19.60	54.30	28.80	22.47	65.10	32.00	42.91	53.94
	Alpaca-7B	42.20	31.20	35.88	39.42	42.20	33.40	37.29	40.09	82.20	62.20	70.81	77.23	62.20	39.50	48.32	55.79
	Vicuna-13B	63.90	51.00	56.73	60.82	51.40	39.30	44.54	48.42	83.90	73.90	78.58	81.69	68.50	49.00	57.13	63.45
	Bactrian-X$_{bloom}$-7B	60.80	43.80	50.92	56.42	53.70	41.00	46.50	50.57	79.40	63.00	70.26	75.47	62.00	51.00	55.96	59.44
	Bactrian-X$_{llama}$-7B	58.60	41.40	48.52	54.10	51.00	38.10	43.62	47.77	77.00	59.20	66.94	72.63	58.60	48.10	52.83	56.15
Seq2Seq	mT0	69.35	54.29	60.90	65.70	57.45	42.50	48.86	53.67	82.35	75.34	78.69	80.85	70.20	50.30	58.61	65.05
	mT5	69.00	53.20	60.08	65.13	56.70	39.50	46.56	52.16	81.00	70.00	75.10	78.53	68.00	48.00	56.28	62.77
	AraBART	72.00	61.50	66.34	69.62	60.00	49.70	54.37	57.61	85.00	78.50	81.62	83.62	74.00	60.50	66.57	70.84
	AraT5$_{v2}$	74.50	64.50	69.14	72.26	63.50	52.70	57.60	61.00	88.00	84.50	86.21	87.28	81.50	69.50	75.02	78.78
	AraT5$_{v2}$ (5M)	75.33	67.44	71.17	73.61	64.55	51.55	57.32	61.45	89.22	83.40	86.21	87.99	81.30	70.24	75.37	78.82
	AraT5$_{v2}$ (10M)	75.90	68.33	71.92	74.25	65.34	52.44	58.18	62.28	89.88	84.22	86.96	88.69	82.34	71.44	76.50	79.90
	AraT5$_{v2}$ (11M)	77.85	68.90	73.10	75.88	66.33	55.20	60.26	63.76	90.10	85.21	87.59	89.08	84.55	71.50	77.48	81.57

Table 15: Dev Set results on the QALB-2014 benchmark dataset.

CLASS	**GECToR_ARBERT**	**five-shot_2014_expertprompt**	**five-shot_2014-chatgpt4**	**AraT5 (11M)**	**COUNT**
OH	73.73	89.80	92.91	87.34	4902
OT	76.59	94.12	95.58	90.84	708
OA	78.63	84.66	88.93	87.35	275
OW	38.57	80.79	86.96	83.70	107
ON	0.00	0.00	0.00	0.00	0
OG	48.00	55.74	63.64	90.32	34
OC	21.43	28.57	53.66	87.18	22
OR	38.24	53.02	65.96	77.10	528
OD	33.76	51.89	59.60	73.07	321
OM	41.80	44.53	57.35	86.44	393
OO	0.00	0.00	0.00	0.00	0
MI	11.02	13.25	20.53	75.00	83
MT	0.00	7.84	11.43	62.50	7
XC	32.95	46.10	50.78	88.35	526
XF	6.06	17.98	23.81	76.92	29
XG	37.10	19.57	31.35	89.47	79
XN	25.19	25.79	31.25	88.12	108
XT	3.95	3.78	5.48	2.48	66
XM	2.04	4.14	6.38	1.07	26
XO	0.00	0.00	0.00	0.00	0
SW	50.51	21.25	33.38	8.29	219
SF	0.00	6.67	3.45	57.14	3
PC	60.89	56.25	47.59	74.98	713
PT	29.62	29.58	21.40	57.42	480
PM	55.24	54.21	52.09	67.08	5599
MG	25.05	75.96	79.70	64.80	434
SP	42.27	90.93	91.61	86.70	805
micro avg	55.67	60.05	64.51	57.28	16467
macro avg	30.84	39.13	43.51	61.62	16467
weighted avg	56.98	66.96	68.24	76.35	16467

Table 16: Analysis of Error Type performances on the QALB-2014 Test set.

Aswat: Arabic Audio Dataset For Automatic Speech Recognition Using Speech-Representation Learning

Lamya Alkanhal[1] **Abeer Alessa***[⁜2] **Elaf Almahmoud***[⁜3] **Rana Alaqil** [⁜4]

[1]Saudi Technology and Security Comprehensive Control Company (Tahakom), Saudi Arabia.
[2]King Saud University, Saudi Arabia [3]Center for Complex Engineering Systems, Saudi Arabia,,Massachusetts Institute of Technology, USA, [4]Intelmatix, Saudi Arabia

lalkanhal@tahakom.com aalessa5.c@ksu.edu.sa elaf@mit.edu ralaqil@intelmatix.ai

Abstract

Recent advancements in self-supervised speech-representation learning for automatic speech recognition (ASR) approaches have significantly improved the results on many benchmarks with low-cost data labeling. In this paper, we train two self-supervised frameworks for ASR, namely wav2vec, and data2vec, in which we conduct multiple experiments and analyze their results. Furthermore, we introduce Aswat dataset, which covers multiple genres and features speakers with vocal variety. Aswat contains 732 hours of clean Arabic speech that can be used in the pretraining task for learning latent speech representations, which results in achieving a lower word error rate (WER) in Arabic ASR. We report the baseline results and achieve state-of-the-art WERs of 11.7% and 10.3% on Common Voice (CV) and the second round of Multi-Genre Broadcast (MGB-2) respectively, as a result of including our dataset Aswat.

Index Terms: Automatic speech recognition, Self-supervised learning, wav2vec, data2vec.

1 Introduction

Automatic speech recognition (ASR) is the task of transcribing speech audio into text. Supervised deep learning has shown a notable improvement in speech recognition, providing significant gains in tasks rich in labeled data. Unfortunately, this reliance on labeled data limits the extent to which deep learning can advance, primarily because of the scarcity of labeled data in some tasks. Recently, self-supervised approaches have overcome this problem and made it possible to reach outstanding results with a limited labeled dataset (Baevski et al., 2020; Hsu et al., 2021; Baevski et al., 2022). Self-supervised learning leverages raw waveforms to learn representation that captures low level features and underlying structure of the data. The learned representations in the pretraining phase are used in downstream tasks in a supervised phase with a minimal amount of labeled data.

Arabic is one of the most spoken languages worldwide, with over 400 million speakers (Graves, and Jaitly, 2014). It is considered challenging to process automatically due to various internal factors, including multiple dialects, ambiguous syntax, syntactical flexibility, and diacritics (Hussein et al., 2022). However, Modern Standard Arabic (MSA) is one formal dialect that is understood by the majority of Arabic speakers. It is the formal spoken and written dialect that is often used in formal speech, news broadcasts, radio, and newspapers. It is also taught in schools and universities (Ryding , 2005).

In our work, we utilized the self-supervised frameworks wav2vec (Baevski et al., 2020) and data2vec (Baevski et al., 2022), and released dataset Aswat (Voices), on which we trained the ASR systems. Aswat is a well-organized, unannotated dataset of Arabic speech, of which 66% is in MSA (Modern Standard Arabic). We carefully curated and manually cleaned it, and it includes speakers from various demographic backgrounds. It has 732 hours of speech constructed from audio files on the internet; thus, it covers a variety of audio files recorded from different speakers and under various recording setup environments. Aswat leads to the learning of useful latent speech representations during the pretraining task in wav2vec (Baevski et al., 2020) and data2vec (Baevski et al., 2022). This results in state-of-the-art performance in Arabic with a word error rate (WER) of 11.7% on Common Voice (CV) and 10.3% on MGB-2, achieved with fewer training instances compared to the second round of Multi Genre Broadcast (MGB-2). The original audio files are crawled

*Equal contribution

⁜Work done in Tahakom

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 120–127
December 7, 2023 ©2023 Association for Computational Linguistics

Dataset	Dialect	Domain	Split	#Hours	#Segments
Common Voice	MSA	Monologues	train	31.5	27,823
			valid	12.7	10,386
			test	12.6	10,388
MGB-2	MSA (70%), DA (30%)	News: Conversation (63%), interview (19%), report (18%)	train	1,128	376,011
			valid	8.5	5,002
			test	9.6	5,365
Aswat	MSA (66%), Saudi (27%), other dialects (7%)	Monologues (45%), Dialogues (55%)	train	724.6	502,391
			valid	7.3	5,065

Table 1: Comparison between CommonVoice, MGB-2 and Aswat.

from YouTube and Soundcloud; therefore, they are subject to copyright. We made the dataset publicly available[1] for non-commercial purposes. This paper's contributions can be summarized as follows:

- Releasing baseline results in Arabic for some of the most prominent self-supervised models in speech, namely wav2vec and data2vec.
- Providing 732 hours of a high-quality diverse Arabic speech dataset.
- Comparing the results obtained from pretraining wav2vec and data2vec on Aswat with two of the most well-known Arabic benchmarks in ASR with extensive analysis, with which we were able to achieve the lowest WER.

2 Background

2.1. Self-supervised speech models

Self-supervised approaches have led to significant advances in the field of speech recognition [1,2,3]. Wa2vec2.0 (Baevski et al., 2020) is the most prominent self-supervised approach in speech, and data2vec (Baevski et al., 2022) is an approach that produced state-of-the-art results on Librispeech.

2.1.1 *Wav2vec*

The architecture consists of three components: a feature encoder where the audio waves are encoded with a stack of 1-D convolutional layers, a quantization module to map the resulting latent representations into a discretized space, and a contextual network used during the pretraining where a span of the resulting representations are masked and fed into a context network that follows the transformer network. It learns contextualized representations and tries to distinguish them from quantized distractors via a contrastive task. The pretrained model is fine-tuned by projecting a linear head on the top of the context network with connectionist temporal classification (CTC) loss (Baevski et al., 2020).

2.1.2 *Data2vec*

Data2vec is a unified framework that works with three modalities (images, text, and speech) separately. It learns to construct representations that are continuous and contextualized. For speech data, the audio inputs are encoded by 1-D convolution layers. Then, the resulting latent representations are fed into a standard transformer network. The architecture consists of a single model with two modes: student and teacher. In the student mode, the model encodes a masked version of the representation, and in the teacher mode, it encodes the unmasked version of the representation to construct the training targets. The model's training mode is parameterized by an exponential moving average (EMA) of the student's parameters. The student's learning task is to minimize the objective function of the student's prediction of a target that is constructed by the teacher's parameters. Similar to wav2vec, the model is fine-tuned with CTC loss (Baevski et al., 2022).

2.2. Annotated datasets

While the audio datasets in Arabic are still scarce compared to other languages, there is an increase in the recent work to bridge the gap such as: the datasets of Multi-Genre Broadcast challenge, MGB-2 (Ali et al., 2016), MGB-3 (Ali et al., 2017), MGB-5 (Ali et al., 2019), Arabic Mozilla's Common Voice[2], ADI-17 (Shon et al.,

[1] https://github.com/AswatDataset/AswatDataset

[2] https://commonvoice.mozilla.org/ar/datasets

2020), QASR (Mubarak et al., 2021), MASC (Al-Fetyani et al., 2021), and SADA[3]. In our work, we consider the most well-known Arabic labeled datasets in ASR, namely Common Voice and the second round of MGB. Moreover, they are publicly available datasets that focus on MSA speech and are commonly used in literature, we used them for comparison and benchmarking.

2.2.1 Common Voice

Mozilla's CV is a platform that provides a public audio dataset with multiple languages powered by the voices of volunteers around the world, it allows users to record and validate other people's recordings. In this paper, we used Arabic CV version 8.0 that was released on January 19, 2022 and recorded by 1,216 volunteers[2].

2.2.2 MGB-2

MGB-2 uses a multi-dialect dataset with 70% MSA and 30% Dialectal Arabic (DA). It includes programs recorded from 2005 to 2015. The training script is aligned using the QCRI Arabic LVCSR system, and it is manually transcribed but not always verbatim; it includes rephrasing, removal of repetition, and summarization, whereas the validation and test sets are transcribed verbatim. These alterations lead to variation in the transcripts' quality; the WER between the original transcribed text to the verbatim version is about 5% in the validation set (Ali et al., 2016). The dataset includes a large corpus of 130 million words from Al-Jazeera website. We used this corpus for language modeling.

Table 1 depicts the two datasets' information, excluding the overlapping segments from MGB-2 in the validation and test sets.

3 Related Work

In (Ashish et al., 2017), the first transformer-based architecture was introduced to better parallelize self-attention mechanisms. Furthermore, when applied to ASR tasks, Karita et al., (2019) demonstrated that transformer-based models outperformed state-of-the-art recurrent neural networks (RNNs). In the ASR task, self-supervised approaches, such as [1, 3], have recently shown significant improvement. The main difference between them is that wav2vec learns discrete units of speech during pretraining through a quantitation process, and data2vec directly predicts contextualized latent representations without quantization.

Although the literature on E2E models trained on Arabic speech is limited, researchers have done valuable work that is essential to the community. In (Ali et al., 2018), the authors used CTC and RNNs, and the reported results were on the MGB-2 development set, without any further results on the test set. In (Belinkov et al., 2019), the authors analyzed the learned internal representations and compared phonemes and graphemes as well as various articulatory features using DeepSpeech2, an end-to-end ASR model. In Taha Zouhair's work[4], the author used wav2vec model on CV benchmark, achieving a WER of 24 %. Belinkov et al. (2019) utilized the transformer architecture with CTC and attention objectives resulting in a WER of 12.5 % in an MSA task on MGB-2. More recently, Chowdhury et al. (2021) proposed a multilingual strategy for dialectal code switching in Arabic ASR. Using end-to-end transformer models reported in (Belinkov et al., 2019) for Arabic, they achieved state-of-the-art results with a WER of 12.1 % demonstrating the effectiveness of multilingual approaches. In our work, we constructed a high-quality dataset and reached state-of-the-art WERs on two well-known benchmark datasets, by pretraining self-supervised architectures, namely wav2vec2.0 (Baevski et al., 2020) and data2vec (Baevski et al., 2022).

4 Aswat Dataset

4.1. Dataset construction

During the dataset construction phase, we started by selecting Arabic audio data with clear pronunciation, and targeted various speech data recorded under multiple settings, such as audiobooks, news, podcasts, and lectures. It covers multiple genres, including politics, philosophy, history, health, folklore, religions, sports, economy, and science. The data includes clear conversation in an interview-like setting without any overlapping speech. We obtained 1060 audio files from two platforms: YouTube and SoundCloud. The former is a video-sharing

[3] https://www.kaggle.com/datasets/sdaiancai/sada2022

[4] https://www.diva-portal.org/smash/get/diva2:1579121/FULLTEXT01.pdf

service, and the latter is a service for sharing audio and music.

4.2. Dataset cleaning

We cleaned the dataset manually to improve speech intelligibility and find better speech representations. All audio files were reviewed to remove noise such as background music using Audacity tool[5].

4.3. Data preprocessing

We reduced the number of channels from stereo to mono-channel and resampled the wave rate to 16 kHz. Finally, we split the audio into segments ranging in length from 3 to 27 seconds, based on silence regions using Pydub Python Package[6]. Details of Aswat are presented in Figure 1.

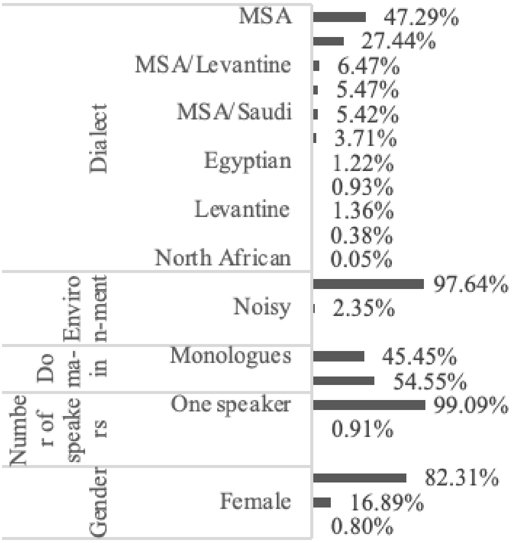


Figure 1: *Aswat Statistics.*

5 Experimental Settings

5.1. Acoustic model

5.1.1 Data preparation

For the acoustic modeling, we segmented the MGB-2 audio files on the timing information provided in the XML files. Then, we converted the audio files of CV and MGB-2 to mono-channel, resampled their rates to 16 kHz, and exported the audio files into FLAC format. We excluded the overlapped speech from MGB-2 validation and test sets.

For the transcription, we preprocessed the transcripts by removing punctuation, diacritics, and any other characters except for the Arabic letters. For the numbers in MGB-2 transcription, we reported the results of two different preprocessing techniques: 1) converting numbers to numerals (words); and 2) removing data entries in the training set that have numbers in their transcription.

5.1.2 Pretraining

We used the implementation of wav2vec and data2vec in fairseq (Ott et al., 2019). We considered only the BASE models and used the same fairseq hyper-parameters (Ott et al., 2019). Moreover, we initialized the models with the fairseq pretrained weights of Librispeech and started the training without resting the optimizer. For pretraining, we ran three experiments: 1) we trained the models on Aswat; 2) we trained the models on MGB-2; and 3) we trained the models on a combined dataset (C.Dataset) of Aswat and MGB-2. The purpose of these experiments is to compare Aswat to MGB-2 and determine which model provides better speech representations for Arabic when they are fine-tuned on the same task. We did not train a model on CV because it is relatively small and pretraining requires a large dataset. For the validation task in the first experiment, we randomly sampled 1% of Aswat dataset and set it as the validation set, and we used the rest for training because self-supervised approaches need substantial data for the pretraining task.

In pretraining data2vec models, the training crashed in the early epochs of the model that was trained on MGB-2, and it crashed in the later epochs of the two other models with the following message: “Minimum loss scale reached (0.0001).” which is caused by the loss overflow. We were able to delay the crashing to later epochs by setting the fp16 scale tolerance to 0.25. We used 16 Tesla V100 (32GB) GPUs for each experiment and chose the training checkpoints with the lowest loss on the validation set.

5.1.3 Fine-tuning

In this stage, we fine-tuned the pretrained models on the labeled data CV and MGB-2 separately. For hyper-parameter selection, we used the configurations of Librispeech-100h for CV and Librispeech-960h for MGB-2 as they resulted in

[5] https://www.audacityteam.org
[6] https://github.com/jiaaro/pydub

the best WERs compared to other Librispeech configurations, we used the same settings except for the max update where we increase it to 640000.

However, we encountered the same issue in fine-tuning that appeared while pretraining our model: the training crashed at early epochs. To train the model for longer epochs, we reduced the batch size and switched from fp16 to fp32. We conducted each experiment on 8 Tesla V100 (32GB) GPUs and chose the models with the lowest WER on the validation set.

5.2. Language Model

We considered a transformer-based language model (LM) provided in fairseq (Ott et al., 2019) to decode the results of the speech recognition models. We used MGB-2 corpus for this task and cleaned the text by removing extra new lines and any non-Arabic characters. Then, we split the text into sequences with a maximum length of 300 words and an overlap of 50 words.

The model was trained on 8 Tesla V100 (32GB) GPUs with the same data splitting approach and hyper-parameters in fairseq (Ott et al., 2019). We tuned the hyper-parameters lm_weigh and word_score and obtained the best results from the values 0.2 and −0.2, respectively.

6 Results and Discussion

6.1 Fine-tuning on Common Voice

We used Arabic CV version 8.0 in training the speech models. Table 2 shows the results of evaluating the models on CV test set, and we decoded the results using LM with beams 5 and 20.

Model	Unlabeled Data	No LM	LM, beam=5	LM, beam=20
wav2vec	Aswat	**16.4%**	**16.1%**	**15.9%**
	MGB-2	18%	17.3%	17.2%
	C.Dataset	16.5%	16.3%	16.1%
data2vec	Aswat	12.1%	13.1%	13%
	MGB-2	15.5%	15.5%	15.3%
	C.Dataset	**11.7%**	**12.6%**	**12.5%**

Table 2: WER on the CV test when training on the CV training set. The best results in each framework are in bold, and the second best results are underlined.

For fine-tuning on CV, we achieved the best results for data2vec models from pretraining on the combined dataset, followed by Aswat, and then MGB-2. For wav2vec, pretraining on Aswat yielded a lower WER than the combined dataset, as Table 2 shows. Additionally, the significantly lower WER achieved by pretraining on Aswat compared to MGB-2 could be attributed to one of two factors: (1) the similarity between Aswat and CV, as they both contain monologue speech, or (2) Aswat has better speech representation, and better generalization. We were able to achieve a state-of-the-art WER of 11.7% on Arabic CV benchmark with the ASR model alone. Decoding with LM resulted in improving the WER of the wav2vec models, but it increased the WER for data2vec, except for the model trained on MGB-2.

Our explanation for the LM performance is that the LM is trained on news data (MGB-2) which has a different domain from common voice (i.e. blog posts, books, movies). In data2vec, the acoustic model (AM) has good results, but LM tends to replace unseen or infrequent words generated by AM with words from its dictionary, which results in increasing WER score. In wav2vec, the AM generates texts that contain non-real words, which are subsequently corrected by the LM. While it's true that the LM occasionally replaces correct words with incorrect ones, the frequency of such cases is significantly lower than the instances where it makes correct predictions. As a result, this contributes to an improvement in WER.

Analysis of the best model errors in Table 2 shows that most errors are substitution errors. Such errors occur due to the similarity in pronunciation of some Arabic sounds between MSA and DA. For instance, the model has many substitutions between سِين (sīn) and صَاد (ṣād), ضَاد (ḍād) and دَال (dāl), and ضَاد (ḍād) and ظَاء (ẓāʾ). Also, some errors occur due to the features that cannot be automatically captured by the model, such as the rules of writing the variations of the هَمْزة (hamzah) which indicates a glottal stop. In Arabic, there are two types of هَمْزة (hamzah) or glottal stops: Hamzat Al-Wasl and Hamzat Al-Qata’a. Hamzat Al-Wasl is written as an أَلِف (ʾalif) without the هَمْزة (hamzah) marker, and it is only pronounced if it is in the beginning of an utterance. In contrast, Hamzat Al-Qata’a is written as an أَلِف (ʾalif) with the هَمْزة (hamzah) marker, and it is always pronounced.

6.2. Fine-tuning on MGB-2

For MGB-2, we used PyArabic Python package[7] to transform numbers to their verbatim form. The WERs of MGB-2 results are reported using the evaluation script provided in the MGB challenge Github repository[8]. The table below depicts the results of testing the model with LM decoded with beams 5 and 20.

Model	Unlabeled Data	No LM	LM, beam=5	LM, beam=20
wav2vec	Aswat	14.7%	13.1%	12.9%
	MGB-2	14.2%	**12.8%**	12.6%
	C.Dataset	**14.1%**	12.9%	**12.5%**
data2vec	Aswat	13%	12.3%	12.1%
	MGB-2	12.6%	11.9%	11.8%
	C.Dataset	**12.1%**	**11.6%**	**11.4%**

Table 3: WER of the first experiment on the MGB-2 test. The best results in each framework are in bold, and the second best results are underlined.

Table 3 shows that the best obtained models in wav2vec and data2vec were those pretrained on the combined dataset, followed by MGB-2, and then Aswat. The addition of Aswat to the pretraining improved the WER from 14.2% to 14.1% in wav2vec and 12.6% to 12.1% in data2vec. The model pretrained only on MGB-2 has an advantage over the model that was pretrained only on Aswat because it has seen all of the data used for fine-tuning, so it has learned better speech representations for MGB-2 and therefore yields a better WER.

We noticed from analyzing the errors of the best model in Table 3 that most errors are substitutions in numeral words. The model substitutes the word for “fifteen” in DA“خمستاشر” (xmsta:ʃr) with its equivalent in MSA “ خمس عشرة” (xms ʕʃrt), the word for “sixteen,” “ستاشر” (sta:ʃr) with “ست عشرة” (st ʕʃrt), the word for “seventy,” “وسبعين” (wsbʃjn) with “وسبعون” (wsbʃwn), and “two thousand” “ألفين” (ʔfjn) with “ألفان” (ʔfa:n). These errors come from using PyArabic tool in preprocessing; it converts every number to its MSA form and uses one grammatical case: Al-Raf’á case (the nominative case). We tackled this issue in the second experiment by dropping examples with numbers from the training set, resulting in removing 11.4% of the training data and reducing the WER by 9.6%.

Model	Unlabeled Data	No LM	LM, beam=5	LM, beam=20
wav2vec	Aswat	12.8%	11.8%	11.6%
	MGB-2	12.8%	11.7%	11.6%
	C.Dataset	**12.4%**	**11.4%**	**11.2%**
data2vec	Aswat	11.4%	10.8%	10.7%
	MGB-2	11.3%	10.7%	10.7%
	C.Dataset	**10.9%**	**10.5%**	**10.3%**

Table 4: WER of the second experiment on the MGB-2 test. The best results in each framework are in bold, and the second best results are underlined.

Table 4 depicts the result of the second experiment. The ASR model shows the best results yielded from fine-tuning the data2vec model that was pretrained on the combined dataset. In addition, the decoded output shows that the model predicts the numerical words correctly. Evaluating the models with the LM reduced the WERs and closed the gap between WERs of wav2vec models. We reached a state-of-the-art (SOTA) WER of 10.3% on the MGB-2 benchmark and outperformed the previous result of 12.1% (Chowdhury et al., 2021).

Analyzing the errors shows that most of them are substitution errors between different Hamza variations and between تاء مربوطة (tāʾ marbūṭah) and هَاء (hāʾ). In addition, some substitutions come from removing the Arabic definite article “ال” (Al) and the connected prepositions and conjunctions from the beginning of the word, such as removing فَاء (fāʾ), بَاء (bā)ʾ, and وَاو (wāw).

Additionally, we observed that the model removes words that are pronounced with an American English accent, even if they are Arabic words. This behavior could be attributed to removing Latin letters from the training script, although the presence of these letters was very small in the dataset.

Finally, Tables 2, 3, and 4 show that data2vec produced better results in all of the experiments, as (Baevski et al., 2022) claimed that discrete units are not required with the use of rich contextualized targets and that learning contextualized targets during the pretraining phase leads to better performance. Our empirical research shows that this claim holds true for Arabic speech data.

[7] https://pypi.python.org/pypi/pyarabic

[8] https://github.com/qcri/ArabicASRChallenge2016

Limitations

While our work achieved state-of-the-art performance, it has three main limitations. First, although our dataset was carefully curated and meticulously cleaned to meet our research objectives; it is important to note a limitation in speaker diversity. This imbalance in gender representation within our dataset can potentially affect our model's performance indicating the need for future experiments with more diverse set of speakers and conducting experiments on the effect of gender bias in our model's performance. Second, while our research used a self-supervised approach, we confined our experimentation with fine-tuning on ASR only, which limited our exploration of other downstream tasks that may benefit from our dataset. The focus on ASR was an intentional choice given its prominence and frequent usage among speech tasks. Nevertheless, we acknowledge that the broader applicability of our dataset across different tasks remains an open question. Third, we did not use the larger version of wav2vec and data2vec models. Although the larger model may potentially yield better performance, the primary goal of this paper was to improve Arabic ASR results and reach SOTA results with our current model configuration. Our findings have successfully demonstrated the benefits of our dataset.

7 Conclusion

In this work, we provide the community with 732 hours of a clean and organized Arabic speech dataset. We report state-of-the-art results for ASR with data2vec architecture, and by combining Aswat with MGB-2 in the pretraining stage, we achieved a WER of 11.7% on CV and 10.3% on MGB-2. In the future work, we plan to improve our methods by using automatic audio cleaning tools[9] and tool in (David at al., 2018) to collect bigger data and include more dialects. In addition, we plan to use the LARGE data2vec and adjust the hyper-parameters based on the training data to enhance the results.

References

Ahmed Abdelrahman, Yasser Hifny, Khaled Shaalan, and Sergio Toral. 2018. *"End-to-End Lexicon Free Arabic Speech Recognition Using Recurrent Neural Networks."* Computational Linguistics, Speech and Image Processing for Arabic Language: 231–48.

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 Proc IEEE SLT.

Ahmed Ali, Stephan Vogel, and Steve Renals. 2017. *"Speech recognition challenge in the wild: Arabic MGB-3.* IEEE Automatic Speech Recognition and Understanding Workshop (ASRU).

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".* IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)

Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. *"Wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations."* Advances in Neural Information Processing Systems 2020-December: 1–12.

Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. 2022. *"data2vec: A General Framework for Self-supervised Learning in Speech, Vision and Language,".* In Proceedings of the 39th International Conference on Machine Learning. PMLR.

Alex Graves, and Navdeep Jaitly. 2014. *"Towards End-to-End Speech Recognition with Recurrent Neural Networks."* 31st International Conference on Machine Learning, ICML.

Amir Hussein, Shinji Watanabe, and Ahmed Ali. 2022. *"Arabic Speech Recognition by End-to-End, Modular Systems and Human."* Computer Speech and Language 71: 1–39.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. *"Attention is all you need."* In

[9] https://github.com/wiseman/py-webrtcvad

Advances in Neural Information Processing Systems, pages 6000–6010.

David Doukhan, Eliott Lechapt, Marc Evrard, and Jean Carrive. 2018. *"Ina's mirex 2018 music and speech detection system."* Music Information Retrieval Evaluation eXchange (MIREX 2018).

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, Online. Association for Computational Linguistics.

Karin C. Ryding. 2005. *"A Reference Grammar of Modern Standard Arabic"*. Cambridge: Cambridge University Press, 2005.

Mohammad Al-Fetyani, Muhammad Al-Barham, Gheith Abandah, Adham Alsharkawi, and Maha Dawas. 2021. *"MASC: Massive Arabic Speech Corpus."* IEEE Spoken Language Technology Workshop (SLT). doi:10.21227/e1qb-jv46

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. *"FAIRSEQ: A fast, extensible toolkit for sequence modeling."* In North American Association for Computational Linguistics (NAACL): System Demonstrations.

Shammur Absar Chowdhury, Amir Hussein, Ahmed Abdelali, and Ahmed Ali. 2021. *"Towards one model to rule all: Multilingual strategy for dialectal codeswitching Arabic."* Interspeech 2021. pages. 2466–2470. [Online]. Available: https://www.isca-speech.

Shigeki Karita, Nanxin Chen, Tomoki Hayashi, Takaaki Hori, Hirofumi Inaguma, Ziyan Jiang, Masao Someki, Nelson Enrique Yalta Soplin, Ryuichi Yamamoto, Xiaofei Wang, Shinji Watanabe, Takenori Yoshimura, Wangyou Zhang. 2019. *"A Comparative Study on Transformer vs RNN in Speech Applications."* 2019 IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2019 - Proceedings 9(4): 449–56.

Suwon Shon, Ahmed Ali, Younes Samih, Hamdy Mubarak, and James Glass. 2020. *"ADI17: A Fine-Grained Arabic Dialect Identification Dataset"*. ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)

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(Cv): 3451–60.

Yonatan Belinkov, Ahmed Ali, and James Glass. 2019. *"Analyzing Phonetic and Graphemic Representations in End-to-End Automatic Speech Recognition."* Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH.

Analyzing Multilingual Competency of LLMs in Multi-Turn Instruction Following: A Case Study of Arabic

Sabri Boughorbel
Qatar Computing Research Institute
Hamad Bin Khalifa University
Doha, Qatar
sboughorbel@hbku.edu.qa

Majd Hawasly
Qatar Computing Research Institute
Hamad Bin Khalifa University
Doha, Qatar
mhawasly@hbku.edu.qa

Abstract

While significant progress has been made in benchmarking Large Language Models (LLMs) across various tasks, there is a lack of comprehensive evaluation of their abilities in responding to multi-turn instructions in less-commonly tested languages like Arabic. Our paper offers a detailed examination of the proficiency of open LLMs in such scenarios in Arabic. Utilizing a customized Arabic translation of the MT-Bench benchmark suite, we employ GPT-4 as a uniform evaluator for both English and Arabic queries to assess and compare the performance of the LLMs on various open-ended tasks. Our findings reveal variations in model responses on different task categories, e.g., logic vs. literacy, when instructed in English or Arabic. We find that fine-tuned base models using multilingual and multi-turn datasets could be competitive to models trained from scratch on multilingual data. Finally, we hypothesize that an ensemble of small, open LLMs could perform competitively to proprietary LLMs on the benchmark.

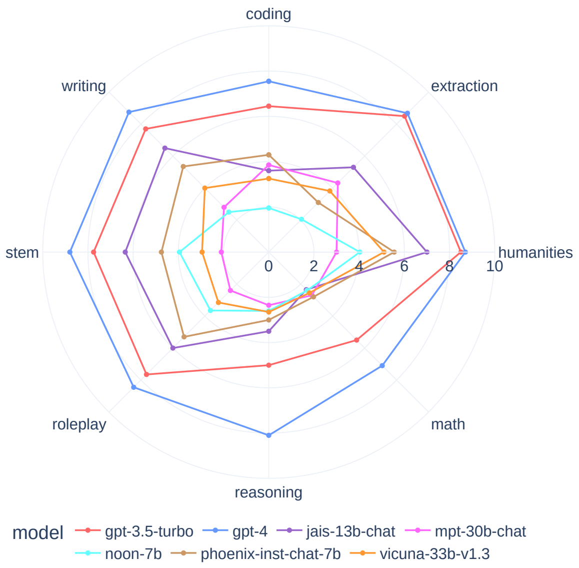


Figure 1: Performance scores per category for selected LLMs on the original MT-Bench (Zheng et al., 2023) for English. The model responses are evaluated by GPT-4 and scored on a scale of 1 to 10 using criteria of helpfulness, relevance, accuracy, depth, creativity, and level of detail.

1 Introduction

Recently, Large language models (LLMs) have brought about significant disruptions across various domains in both research and industry. LLMs have shown strong capability in solving and generalizing across diverse and complex tasks in natural language processing (NLP) and beyond. Moreover, their success in engaging in conversations and accurately following human instructions has been particularly noteworthy. The recent surge in the availability of LLMs necessitates extensive benchmarking and evaluation.

In this work, we analyze the competency of publicly-available, open LLMs when prompted with open-ended, multi-turn instructions in a language different than English. We compare the quality of these responses to the ones generated from equivalent instructions in English in order to identify the strengths and weaknesses of these models in terms of their multilinguality. Specifically, we study Arabic instructions, but the analysis could be repeated for any other language. Our study aim to answer the following questions:

- *How do open LLMs fare in following open-ended instructions written in Arabic? and how do they compare to GPT models?*
- *What is the effect of specifically targeting Arabic when training a model?*
- *What is the effect of specifically fine-tuning on Arabic multi-turn instructions?*
- *How to select a good starting point LLM model to fine-tune for Arabic instruction following?*

We start by a brief overview of the LLM benchmarking effort in Section 2. We introduce ARABIC

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 128–139
December 7, 2023 ©2023 Association for Computational Linguistics

MT-Bench in Section 3 as an analysis tool for multilingual instruction following. Then, we attempt to answer the proposed questions through a number of analyses in Section 4. Finally, we conclude in Section 5 with some insights and recommendations for pushing forward the competency of Arabic LLMs.

2 LLM Benchmarking

LLMs have shown capabilities that go far beyond traditional NLP tasks, such as text classification or multi-choice question answering in some target natural language. Their ability to generate human-like text and engage in long conversations in any topic have opened up a multitude of novel opportunities and horizons that transcend tasks and languages. However, many existing benchmarks for LLMs are still anchored in the conventional NLP paradigm or support English only. Consequently, these benchmarks exhibit limitations when it comes to evaluating the proficiency of LLMs in open-ended generation, multi-turn tasks, or in languages other than English.

2.1 Conventional benchmarks

Some of the recent effort in this category include projects such as HELM (Liang et al., 2022) and Evaluation Harness (Gao et al., 2021) which are platforms for LLM benchmarking. Also, standardized datasets such as MMLU (Hendrycks et al., 2021), HellaSwag (Zellers et al., 2019), TruthfulQA (Lin et al., 2022), ARC (Mihaylov et al., 2018) and OpenbookQA (Clark et al., 2018), amongst many others, are used to evaluate core LLM capabilities such as commonsense reasoning, math, question answering, and factuality. In addition, some recent works targeted Arabic language specifically with suites of tasks and datasets, e.g. (Khondaker et al., 2023; Abdelali et al., 2023; Alyafeai et al., 2023).

These benchmarks require specification of prompts per-task and model, in addition to post-processing functions to validate model answers against a gold standard, which might not be straightforward and could prove time-consuming. Moreover, with publicly available answer sets, there is always the potential risk of contamination to the training data of language models. Furthermore, some of these benchmarks have been shown to diverge in certain cases from human judgment (Zheng et al., 2023), possibly due to their narrow focus.

2.2 Instructional and conversational benchmarks

Recent efforts on instruction-following benchmarks, such as Flan (Longpre et al., 2023) and Super-NaturalInstructions (Wang et al., 2022), or conversational benchmarks, such as OpenAssistant (Köpf et al., 2023), CoQA (Reddy et al., 2019) and MMDiag (Feng et al., 2022), present a more sophisticated and comprehensive challenge to LLMs, but they are mostly limited to English, and the diversity of the questions are insufficient for the most advanced LLMs. Translating such datasets to other language is not a straightforward task, as it requires a large effort to manually curate the translated questions and answers for the purpose of ensuring high quality in the target language.

2.3 Evaluating open-ended questions

When it comes to open-ended tasks, such as creative writing, human evaluation of LLM responses is indispensable. Here, a human-in-the-loop acts as a judge to directly score an LLM response or to rank responses of multiple LLMs for the best answer on some question. However, achieving a reliable benchmark this way is resource-intensive and lacks scalability. In one application, LMSYS Chatbot Arena[1], which is a crowd-sourced LLM evaluation platform, allows users to use freestyle prompts for two randomly-selected LLMs before voting for the better response. Benchmarking using this approach, while very powerful, is challenging as it compares models evaluated on different prompts.

An alternative approach that has recently emerged is the employment of an LLM to act as a judge of the responses of other LLMs. MT-Bench (Multi-Turn Benchmark) (Zheng et al., 2023) utilizes this approach on a standard set of 80 open-ended questions of eight categories; namely: writing, extraction, reasoning, math, coding, role-play, humanities, and STEM. Moreover, it assesses the ability of an LLM to maintain a conversation by asking it a follow-up question that is based on its response to the first question. Examples of the MT-Bench questions are shown in Table 1. These examples illustrate the level of open endedness and complexity of the questions, and the dependency of the follow-up question on the first turn.

MT-Bench prompts a judge LLM with an instruction to rate the responses on a scale of 1-10

[1] https://chat.lmsys.org

Writing	T1	Craft an intriguing opening paragraph for a fictional short story. The story should involve a character who wakes up one morning to find that they can time travel.
	T2	Summarize the story with three bullet points using only nouns and adjectives, without verbs.
Reasoning	T1	David has three sisters. Each of them has one brother. How many brothers does David have?
	T2	If we change the previous question and assume that each sister of David has two brothers, how many brothers would David have?
Math	T1	The vertices of a triangle are at points (0, 0), (-1, 1), and (3, 3). What is the area of the triangle?
	T2	What's area of the circle circumscribing the triangle?

Table 1: A sample of questions from MT-Bench in categories Writing, Reasoning and Math. T1 and T2 denote the first turn and second turn (follow-up) questions, respectively.

(where 1 indicates failure in answering the question and 10 indicates a perfect answer), clearly defining the evaluation task and criteria. Also, the judge LLM is asked to provide an explanation for the suggested score. This approach has been shown to have an agreement rate of 85% with human evaluation when GPT-4 is used as a judge (Zheng et al., 2023), which was also found to be higher than human-human agreement (81%). MT-Bench scores for selected LLMs are shown in Figure 1.

The approach of MT-Bench is versatile and scalable as it delegates the resource-intensive scoring of open-ended questions to the judge LLM. Moreover, it could be extended to benchmarking LLMs in other languages by translating the benchmark dataset to the target language as long as a good judge LLM exists for that language. For Arabic, GPT-4 is highly-competent and has showed a good level of proficiency (Khondaker et al., 2023; Abdelali et al., 2023; Alyafeai et al., 2023). Therefore, it is eligible to be used as a judge for Arabic responses. Moreover, by using the same prompt for judging English and Arabic responses for the original and translated versions of the same question, it is even possible to contrast the multilingual skills of an LLMs at a question and a category level.

3 Arabic MT-Bench

In this work, we develop an Arabic version of MT-Bench. First, we auto-translated the original benchmarking questions using Google Translate. A thorough manual curation of the translations is then performed. This step is essential to ensure the quality of the question set and hence the responses and the judgment. For example, all people names in the questions were changed to Arabic names, and questions about correcting English grammatical errors were re-written. See Table 7 in Appendix A.4 for a sample of curated translated questions [2].

In addition to the questions, the benchmark provides reference answers for reasoning, math and code questions that are passed to the LLM judge to aid in the judgment. One option to get these reference answers in Arabic is to prompt GPT-4 with the translated Arabic questions directly, but we decided instead to translate the original reference answers from English to ensure that the Arabic scores for these three categories stay as close as possible to the English MT-bench scores.

Finally, our initial evaluation showed that some LLMs tend to respond in English despite the question being in Arabic. Hence, we decided to add at the end of each question a clear instruction to the LLM to respond in Arabic (الرجاء الإجابة باللغة العربية). We observed that, without having to modify the original judgment prompt, GPT-4, acting as an Arabic judge, has taken into consideration that instruction and scored lower responses in English.

Table 2 gives an overview of the Arabic MT-Bench dataset.

Number of question categories	8
Number of questions per category	10
Number of turns per question	2
Number of reference answers	30

Table 2: Statistics of Arabic MT-Bench dataset

3.1 Score consistency

In order to answer the question: *are the scores of* Arabic MT-Bench *consistent and coherent such that it could be used as a metric?* and to qualitatively assess the effectiveness of Arabic

[2] Arabic MT-Bench is available at https://huggingface.co/spaces/QCRI/mt-bench-ar

Rating	Justification summary
2	Common issues in AI assistant responses include: not addressing user's question, providing irrelevant or repetitive information, lacking depth, creativity, and accuracy, not following user's specific instructions, and not using the requested language. Users often seek detailed, accurate, and creative answers tailored to their requests, but AI assistants sometimes fail to deliver, resulting in unhelpful or unsatisfactory responses.
4	Common issues in the AI assistant's responses include lack of depth, inaccuracies, language inconsistencies, and not directly addressing the user's question. Some responses are repetitive and do not provide comprehensive analysis or examples. To improve, the AI assistant should focus on directly answering the user's question, providing clear and accurate examples, maintaining language consistency, and offering detailed and informative explanations. Additionally, adhering to specific user instructions and avoiding repetition will enhance the overall quality of the responses.
8	AI assistants provide relevant, creative, and accurate responses to various user requests, demonstrating a good understanding of topics and user instructions. They offer helpful suggestions, clear explanations, and maintain requested languages. Responses cover a wide range of subjects, including summarization, problem-solving, and engaging in fictional conversations. However, there are occasional minor mistakes and areas for improvement in clarity and depth. Overall, AI assistants successfully address user questions, providing satisfactory and informative answers.

Table 3: Summaries provided by GPT-4 of the collection of judgment justifications for questiones rated 2, 4 and 8 across all models and tasks. This indicates some level of internal consistency of the ARABIC MT-BENCH scores.

MT-BENCH, we clustered the judgments across all models and categories by their numerical ratings, then asked GPT-4 to summarize its justification texts for every score (1 to 10). In Table 3 are examples of the justification summaries for some ratings.

While qualitative, we could conclude from this analysis that the justifications are reasonably consistent across models and categories, indicating an acceptable level of impartiality. In addition to that, the correlation between scores using the Arabic and English benchmarks for strong models, as will be seen Section 4, is another supporting evidence for the viability of ARABIC MT-BENCH as a metric.

4 Results and Discussion

4.1 Model selection

In addition to OpenAI GPT-3.5-turbo and GPT-4, which are only considered in this work to set an upper bound, a number of open LLMs have been chosen for this study. Through preliminary evaluations on HuggingFace playground, some LLMs exhibited knowledge of Arabic despite not being purposefully trained for it. The criteria we adopted for choosing models involve:

- the model is open-source. Some competitive proprietary models are not accessible to us.
- the model size is 33B or less, a decision driven by constraints in hardware infrastructure.
- the model is known to do well on the English benchmarks on the LMSYS leaderboard[3]

An overview of the chosen models can be seen in Table 4, and more details can be found in Appendix A.3.

4.2 How do open LLMs fare in following open-ended instructions written in Arabic?

Table 5 shows the model ranking based on the ARABIC MT-BENCH scores. The first, second and third columns of the tables give the model's average score for the first turn across all questions, the average score for the second turn across all questions, and the average of both, respectively. Per-category scores could be seen in Figure 2. For comparison, Figure 1 (and Table 6 in Appendix A.1) give the per-category scores for the original English MT-Bench for the same models.

As the results show, GPT-4 and GPT-3.5-turbo are better than any open LLM we tested by a large margin with average scores of 8.27 and 7.13 out of 10, respectively. Because GPT-4 is used as the judge, there exists the potential for bias in favor of its own responses, which has been discussed in the MT-Bench paper (Zheng et al., 2023).

In the English MT-Bench, the two GPT models score 8.99 and 7.0, respectively. Hence, GPT-4 is approximately one point lower in terms of the Arabic score compared to the English benchmark. By manual inspection of the responses, we qualitatively confirm that the proficiency of GPT models in Arabic is lower than English as indicated by the scores. Therefore, we compare the scores across Arabic and English benchmarks in Section 4.3.

Overall, LLMs fine-tuned specifically for Arabic or for multilingual capabilities (e.g. Jais, Phoenix) are better than generic models such as some mem-

[3] https://chat.lmsys.org/?arena

Model	Base model	Size	Training language	Multi-turn
GPT-4	_	>175B	Multilingual	✓
GPT-3.5-turbo	_	175B	Multilingual	✓
Jais-13B-chat	Jais-13B	13B	EN, AR	✓
PolyLM-13B	_	13B	Multilingual	✗
MPT-30B-chat	MPT-30B	30B	Primarily English	✓
LLaMa-2-13B-chat	LLaMa-2-13B	13B	Primarily English	✓
Tulu-30B	LLaMa	33B	Primarily English	✗
Guanaco-33B	LLaMa	33B	Primarily English	✗
Vicuna-33B-v1.3	LLaMa	33B	Primarily English	✓
BLOOMZ-7B1	_	7.1B	Multilingual	✗
BLOOMZ-7B1-MT	BLOOMZ-7B1	7.1B	Multilingual	✗
Noon-7B	BLOOM	7B	Multilingual, AR fine-tuning	✗
Phoenix-chat-7B	BLOOMZ-7B1-MT	7B	Multilingual	✓
Phoenix-inst-chat-7B	BLOOMZ-7B1-MT	7B	Multilingual	✓

Table 4: Attributes of the chosen models for this study. _ for the 'Base model' indicates a model that has been trained from scratch. 'Size' is in the number of parameters. 'Training language' is the natural language/s that made up the pre-training and instruction datasets for the model, and 'Multi-turn' refers to chat fine-tuning.

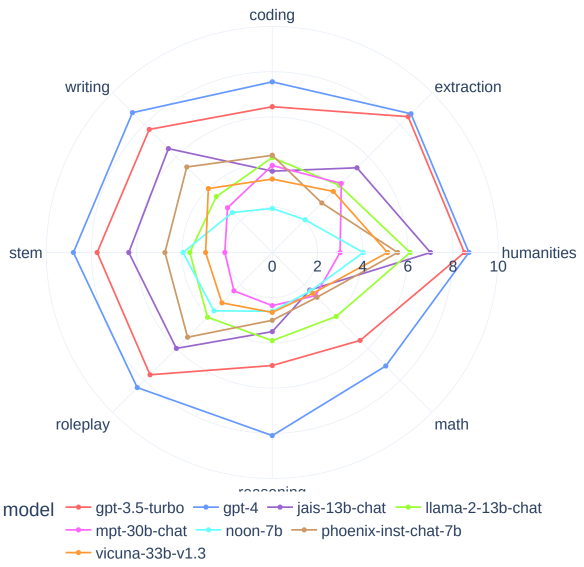


Figure 2: Performance scores per category for selected LLMs on our Arabic multi-turn benchmark. The model responses are evaluated by GPT-4 and scored on a scale of 1 to 10 using criteria of helpfulness, relevance, accuracy, depth, creativity, and level of detail.

Model	Turn1	Turn2	Avg
GPT-4	8.41	8.12	8.27
GPT-3.5-turbo	7.48	6.79	7.13
Jais-13B-chat	5.01	5.14	5.08
Phoenix-inst-chat-7B	4.84	3.70	4.27
Llama-2-13B-chat	4.54	3.86	4.20
Phoenix-chat-7B	4.16	3.84	4.00
Vicuna-33B-v1.3	3.44	3.43	3.43
MPT-30B-chat	3.26	2.62	2.94
Noon-7B	3.39	2.39	2.89
Guanaco-33B	2.68	2.52	2.60
PolyLM-13B	1.91	2.08	1.99
Bloomz-7B1-mt	1.54	1.75	1.64
Bloomz-7B1	1.29	1.54	1.41
Tulu-30B	1.10	1.35	1.23

Table 5: Results of benchmarked LLMs on ARABIC MT-BENCH (scores between 1-10). showing for each model average scores per turn, and average score across all questions and turns.

bers of the Llama family (e.g. Vicuna, Guanaco) in Arabic instruction following, even when smaller in size. The fine-tuning data and recipe matters significantly; for example, Phoenix-inst-chat-7B is much better then its predecessor Bloomz-7B1 or Bloomz-7B1-mt.

Jais-13B-chat is the best open model in Arabic in our evaluation. It achieves an average score of 5.08 out 10. The model has targeted Arabic and English in both pre-training and fine-tuning. Despite this, its relatively small size hinders it from being competitive with the best models. Also, it is still far on the English MT-Bench leaderboard from models of comparable size, where the best model within 13B size in the English MT-Bench achieves a score above 6 out of 10 (see a selection of these scores in Table 6 in the Appendix). Also, Jais-13B-chat model has the largest drop in performance in the second-turn questions on the English benchmark. Jais-13B-chat has been benchmarked internally using a similar approach to ours on private data accordingly to its technical report (Sengupta et al., 2023).

We note that the fine-tuning dataset of Jais-13B-chat is large with over 10M samples. The longer period needed for this fine-tuning could raise additional challenges as it might increase the risk of catastrophic forgetting of knowledge gained during pre-training (Luo et al., 2023; He et al., 2021). For comparison, Phoenix-inst-chat-7B is ranked second among the evaluated open models in our experiment. The model is fine-tuned from a BLOOMZ-7B1-MT base (Chen et al., 2023). The fine-tuning dataset has 133 languages with 58% English, 20.9% Chinese and 0.8% Arabic which is ranked 11th in language coverage, with a total of 267K instruction-tuning samples. The conversation-tuning dataset has 189K samples covering more than 40 languages. Despite its smaller size and wide coverage of languages, Phoenix-chat-7B achieves intriguing results. Figure 3 shows detailed comparison per category for Jais-13B-chat, Phoenix-inst-chat-7B and GPT-3.5. The two open LLMs had the lowest scores on math and reasoning, whereas the highest scores are on roleplay, humanities and stem.

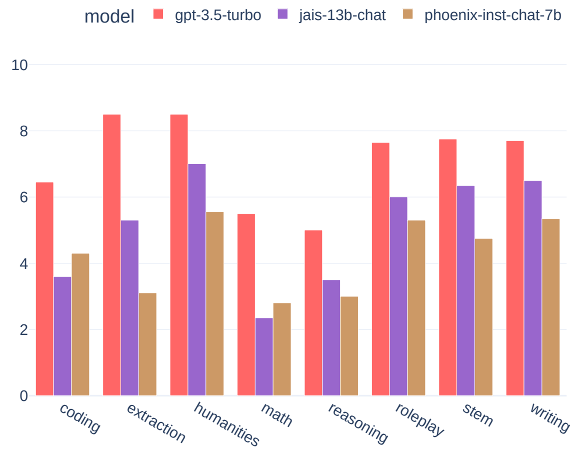


Figure 3: Average scores per category for three selected models evaluated on the ARABIC MT-BENCH.

Vicuna-33B-v1.3 and MPT-30B-chat scored around 3 out 10, while they were not expected to have any significant skill in Arabic. One possible explanation is that given their size over 30B, they are able to maximize their multilingual skills effectively. This hypothesis needs further investigations. Despite their low performance, it is interesting to explore the model development in order to adapt for training multilingual LLMs.

4.3 What is the effect of specifically targeting Arabic when training a model?

Figure 4 shows a heat map of the difference in score per category between the Arabic and the English benchmarks for the selected models. The models are sorted from top to bottom based on a decreasing score differences. Warmer cells in the figure indicate English advantage over Arabic for the same model and category, while cooler cells indicate Arabic advantage.

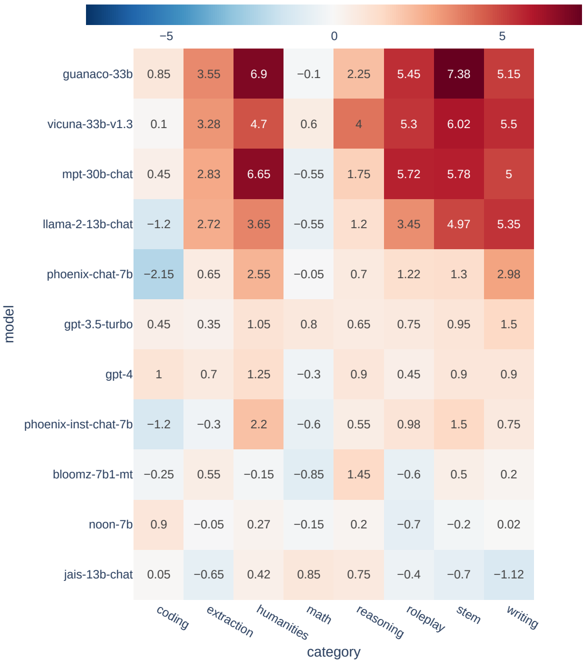


Figure 4: Difference of average MT scores between English and Arabic benchmarks per category. Positive values (red) indicate English answers are scored higher that the corresponding Arabic answers, while negative values (blue) indicate some advantage in Arabic. Neutral colors mean a model is equally-competent in both languages.

The two GPT models reside in the neutral area, indicating comparable competency in English and Arabic. Not surprisingly, Models that have been pre-trained and fine-tuned on multilingual data (see Table 4) appear in the bottom half of the heat map, indicating some Arabic knowledge. Also, it could be seen from the heatmap that coding and math are neutral, language-agnostic skills across models, as should be expected, while reasoning has a lingual side.

Figure 5 shows the per-turn average scores of ARABIC MT-BENCH on the X-axis and English MT-Bench on the Y-axis for the selected models. Points closer to the diagonal line are models with similar average performance in Arabic and English, and the closer to the top right corner the better the model is on both languages. Most models are above the diagonal, and hence exhibit relatively superior skills in English compared to Arabic. This is likely due to the imbalance in the training and fine-tuning data between the two languages. Note that the LLaMa-based models are clustered together far from the diagonal, indicating lack in multilinguality, while BLOOMZ-7B1-MT and Noon-7B, both heavily multilingual, are on top of the diagonal.

4.4 What is the effect of specifically fine-tuning on Arabic multi-turn instructions?

In Figure 5, the two dots for each model represent the two turns, and their placement gives an insight into the ability of a model to engage in a conversation. Vertical drop between the two turns indicates diminished performance on English for the second turn, while horizontal shifts to the left indicates diminished performance on Arabic for the second turn.

BLOOMZ-7B1-MT does not degrade on the second turn, even though it is not fine-tuned on conversational data (Muennighoff et al., 2023), and it is the only model that is not affected in the second turn for both languages, while a capable model like GPT-4 had a slight improvement on the second turn for English but had a minor deterioration of the score for Arabic.

On the other hand, Noon-7B has the largest drop in score between turns on Arabic. This model is built on top of BLOOM by instruct fine-tuning using a combination of datasets with ColossalAI framework (Bian et al., 2021). Noon-7B[4] used GPT-3.5-Turbo as a judge for evaluation on private data. We also observe that Jais-13B-chat has a large drop in English multi-turn instructions compared to a small drop in Arabic, which might be caused by the ratio of Arabic to English instructions in its chat fine-tuning.

Phoenix-chat-7B, Noon-7B and BLOOMZ-7B1-MT are all based on different variants of the backbone BLOOM-7B or BLOOMZ-7B. The resulting models vary a lot in terms of performance, indicating that a careful fine-tuning recipe is crucial for improving the capabilities of any base model.

4.5 How to select a good starting point LLM model to fine-tune for Arabic instruction following?

We consider the hypothetical optimal ensemble model defined by the maximum per-question score across the open models in our experiment. This characterizes an upper bound on the performance of any open LLMs ensemble made from these models. Based on our ARABIC MT-BENCH, the optimal ensemble model achieves an MT score of 6.70. This represents a 32% increase in performance compared to the best individual open LLM (Jais, 5.08). Also it indicates that a collection of smaller models trained differently could capture various skills that might be difficult to capture together in one model without upping the model size. For the sake of contrast, for the English benchmark, the optimal ensemble model achieves a score of 8.2.

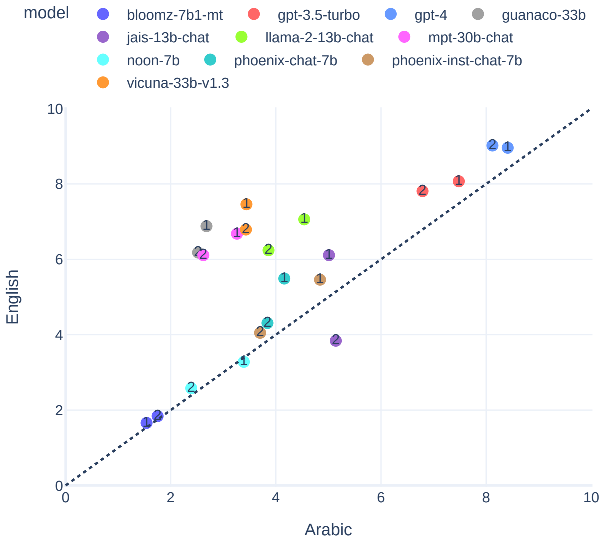


Figure 5: Scores in Arabic (X-axis) and English (Y-axis) MT-Bench for the first and second turn. The farther the model is from the diagonal, the bigger the gap in quality between the two languages. The farther Turn 2 is from Turn 1 for a model, the bigger the change in quality in responding to continued conversation.

Figure 6 shows the contributions of the three highest-scoring LLMs per category in the optimal Arabic ensemble model. We counted how often a model was the best for a given category and considered the top 3 models in each. Note that 'best' here is relative to the performance of available LLMs, and is not an assessment of quality.

As the figure shows, Jais-13B-chat is the top model in five 'literacy' categories, whereas math, coding and reasoning are shared with LLaMa-2-13B-chat, Guanaco-33B, and Phoenix-inst-chat-7b. The challenge is how to define a criterion to select the best response among the ensemble LLMs. One possible approach is to ask each LLM to vote for the best answer and consider a majority vote, which will rely mainly on the ability of these small models to play the role of a judge in this limited context.

[4] https://huggingface.co/Naseej/noon-7b

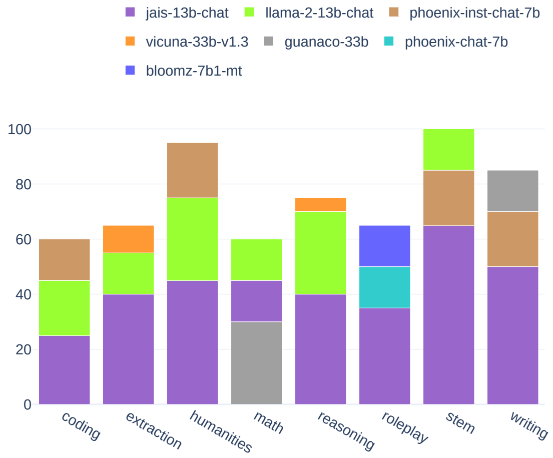


Figure 6: Contribution of the best three LLMs to the optimal ensemble model for each category. The Y-axis indicates how often a model was selected the best in terms of Arabic MT-score for the questions of a category.

We will leave investigating this to future work.

5 Conclusion

In this paper, we propose a framework for analyzing the effect of multilinguality on LLM performance in open-ended tasks. In particular, we assessed the interaction between language, dialog and instruction following in Arabic and English for small open LLMs. We employ an LLM as a judge following the paradigm of MT-Bench. We show the effects of language on different categories of tasks and suggest ways to ensemble small LLMs to achieve better performance on the benchmark.

In future work, we plan to extend the benchmark and analysis with more models and tasks, and investigate the viability of LLM ensemble models.

6 Limitations

We now discuss a number of limitations related to this study.

6.1 Judging

- The use of an LLM as a judge for evaluating LLMs has issues related to bias. As reported in (Zheng et al., 2023), in pairwise comparisons, the judge tends to favor its own answers compared to other models. For example, that study shows that GPT-4 favors itself with 10% higher win rate and Claude-V1 favors itself with 25% higher win rate. On the other hand, GPT-3.5 does not appear to favor itself.
- Using GPT-4 as the judge and as an LLM under study might favor it in the scores. However, the score margin to the closet competitor is big enough to make any potential deviation in the scores insignificant, and we adhered to the original MT-Bench setup in the choice of judge in order to mirror the results and measure multilingual competency.
- Other LLM judges than GPT-4 could be considered for evaluating the responses. However, the choice of alternative judges is currently rather limited when considering Arabic. The proficiency of models such as Claude or Bard in Arabic are not yet proven. Alternatively, multiple LLMs could be used for this task. A voting judgment mechanism could be considered over multiple open LLMs.
- While GPT-4 exhibits competence in Arabic, its proficiency in the language falls short of its mastery of English. This discrepancy may have had an impact on certain aspects of our analyses, especially when comparing Arabic results to English results.
- We used the same judgment prompt as in the English MT-Bench for the purpose of consistency. However, we note that the judgment prompt does not acknowledge important aspects such as safety and harmlessness of LLM responses. Also, the MT-score is a metric that combines multiple dimensions such as relevance, helpfulness, and creativity together to give an aggregate verdict. It might be useful to analyze model performance separately on these dimensions for a better understanding.

6.2 Coverage

- MT-Bench has a limited number of questions (160 in total considering both turns). This is likely not representative of the wide spectrum of tasks needed to effectively evaluate LLMs, and the authors of MT-Bench are acknowledging that by working to expand their benchmarking dataset to 1000 questions. In addition, language-specific dimensions of conversation might require bespoke questions to test properly.
- We only included a small number of models in the benchmark. During an initial screening, we excluded several LLMs due to their limited capabilities in Arabic. We plan to extend our benchmark and include more LLMs in the future.

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, Youssef Elshahawy, Ahmed Ali, Nadir Durrani, Natasa Milic-Frayling, and Firoj Alam. 2023. Benchmarking Arabic AI with large language models.

Zaid Alyafeai, Maged S Alshaibani, Badr AlKhamissi, Hamzah Luqman, Ebrahim Alareqi, and Ali Fadel. 2023. Taqyim: Evaluating Arabic NLP tasks using ChatGPT models. *arXiv preprint arXiv:2306.16322*.

Zhengda Bian, Hongxin Liu, Boxiang Wang, Haichen Huang, Yongbin Li, Chuanrui Wang, Fan Cui, and Yang You. 2021. Colossal-AI: A unified deep learning system for large-scale parallel training. *arXiv preprint arXiv:2110.14883*.

Zhihong Chen, Feng Jiang, Junying Chen, Tiannan Wang, Fei Yu, Guiming Chen, Hongbo Zhang, Juhao Liang, Chen Zhang, Zhiyi Zhang, Jianquan Li, Xiang Wan, Benyou Wang, and Haizhou Li. 2023. Phoenix: Democratizing ChatGPT across languages. *arXiv preprint arXiv:2304.10453*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLora: Efficient finetuning of quantized LLMs. *arXiv preprint arXiv:2305.14314*.

Jiazhan Feng, Qingfeng Sun, Can Xu, Pu Zhao, Yaming Yang, Chongyang Tao, Dongyan Zhao, and Qingwei Lin. 2022. MMDialog: A large-scale multi-turn dialogue dataset towards multi-modal open-domain conversation. *arXiv preprint arXiv:2211.05719*.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2021. A framework for few-shot language model evaluation.

Tianxing He, Jun Liu, Kyunghyun Cho, Myle Ott, Bing Liu, James Glass, and Fuchun Peng. 2021. Analyzing the forgetting problem in pretrain-finetuning of open-domain dialogue response models. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1121–1133.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding.

Md Tawkat Islam Khondaker, Abdul Waheed, El Moatez Billah Nagoudi, and Muhammad Abdul-Mageed. 2023. GPTAraEval: A comprehensive evaluation of ChatGPT on Arabic NLP.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, et al. 2023. OpenAssistant conversations–democratizing large language model alignment. *arXiv preprint arXiv:2304.07327*.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D. Manning, Christopher Ré, Diana Acosta-Navas, Drew A. Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue Wang, Keshav Santhanam, Laurel Orr, Lucia Zheng, Mert Yuksekgonul, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. 2022. Holistic evaluation of language models.

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.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. 2023. The Flan Collection: Designing data and methods for effective instruction tuning. *arXiv preprint arXiv:2301.13688*.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2023. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *arXiv preprint arXiv:2308.08747*.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

MosaicML. 2023. Introducing MPT-30B: Raising the bar for open-source foundation models. Accessed: 2023-09-09.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual generalization through multitask finetuning.

OpenAI. 2023. GPT-4 technical report.

Siva Reddy, Danqi Chen, and Christopher D Manning. 2019. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

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, Preslav Nakov, Timothy Baldwin, and Eric Xing. 2023. Jais and Jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models.

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

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Raghavi Chandu, David Wadden, Kelsey MacMillan, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. 2023. How far can camels go? exploring the state of instruction tuning on open resources.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. 2022. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. *arXiv preprint arXiv:2204.07705*.

Xiangpeng Wei, Haoran Wei, Huan Lin, Tianhao Li, Pei Zhang, Xingzhang Ren, Mei Li, Yu Wan, Zhiwei Cao, Binbin Xie, Tianxiang Hu, Shangjie Li, Binyuan Hui, Bowen Yu, Dayiheng Liu, Baosong Yang, Fei Huang, and Jun Xie. 2023. PolyLM: An open source polyglot large language model.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. *arXiv preprint arXiv:2306.05685*.

A Appendix

A.1 English MT-Bench scores

Table 6 shows the per-turn and average scores for the selected models on the original MT-Bench.

Model	**Turn1**	**Turn2**	**Avg**
GPT-4	8.96	9.02	8.99
GPT-3.5-turbo	8.07	7.81	7.94
Vicuna-33B-v1.3	7.46	6.79	7.12
Llama-2-13B-chat	7.06	6.24	6.65
Guanaco-33B	6.88	6.18	6.53
Tulu-30B	7.02	5.85	6.43
MPT-30B-chat	6.68	6.11	6.39
Jais-13B-chat	6.11	3.84	4.97
Phoenix-chat-7B	5.49	4.31	4.90
Phoenix-inst-chat-7B	5.46	4.05	4.75
Noon-7B	3.28	2.58	2.93
Bloomz-7B1-mt	1.66	1.84	1.75
Bloomz-7B1	1.39	1.85	1.62

Table 6: Results of benchmarked LLMs on English MT-BENCH (scores between 0-10). showing for each model average scores per turn, and average score across all questions and turns.

A.2 Prompts for LLM Judge

Figure 7 shows the judging prompt for the first-turn questions in MT-Bench, and Figure 8 shows the prompt for the second-trun questions.

```
Please act as an impartial judge and evaluate the
quality of the response provided by an AI assistant
to the user question displayed below. Your eval
uation should consider factors such as the help
fulness, relevance, accuracy, depth, creativity,
and level of detail of the response. Begin your
evaluation by providing a short explanation. Be
as objective as possible. After providing your
explanation, you must rate the response on a scale
of 1 to 10 by strictly following this format:
[[rating]], for example: "Rating: [[5]]".
```

Figure 7: LLM judge first turn's prompt. The highlighted text indicates the evaluation criteria.

A.3 Chosen Models

- GPT-4: a proprietary multilingual chatbot by OpenAI, trained on public and proprietary data

> Please act as an impartial judge and evaluate the quality of the response provided by an AI assistant to the user question displayed below. Your evaluation should consider factors such as the helpfulness, relevance, accuracy, depth, creativity, and level of detail of the response. You evaluation should focus on the assistant's answer to the second user question. Begin your evaluation by providing a short explanation. Be as objective as possible. After providing your explanation, you must rate the response on a scale of 1 to 10 by strictly following this format: [[rating]], for example: "Rating: [[5]]".

Figure 8: LLM judge second turn's prompt. The highlighted text in green indicates the evaluation criteria. The highlighted text in orange indicates the instruction to focus the evaluation on the answer of the second question.

and fine-tuned using reinforcement learning with human and AI-generated feedback. Allows 8k and 32k prompts (OpenAI, 2023).

- GPT-3.5-turbo: the predecessor of GPT-4 with 175B parameters.
- Jais-13B-chat: A 13B parameter model that follows the GPT-3 architecture, pre-trained on 279B English and 116B Arabic tokens, then fine-tuned on 5.9 million English and 3.8 million Arabic supervised multi-turn instructions, and further fine-tuned for safety (Sengupta et al., 2023).
- Phoenix-chat-7B: A BLOOMZ-based 7B parameter model fine-tuned for dialog using online ChatGPT records and multi-round conversations (Chen et al., 2023).
- Phoenix-inst-chat-7B: Another 7B model from the Phoenix family, fine-tuned not only for conversations but also for multilingual instruction following using self-instruct and translators.
- Vicuna-33B-v1.3: A 33B LLaMa-based model, fine-tuned on a ShareGPT.com dataset for instruction following and multi-turn dialog (Zheng et al., 2023).
- MPT-30B-Chat: A fine-tuned version of MPT-30B which is an encoder-only transformer model trained on 1T English tokens. MPT-30B-Chat was fine-tuned for chat on a number of public datasets including ShareGPT-Vicuna, Camel-AI, GPTeacher, Guanaco and Baize (MosaicML, 2023).
- Noon-7B: A BLOOM-based 7B parameter model, fine-tuned on 110k Arabic instructions from translated datasets including GPT-4 responses to Alpaca quesitons, Dolly, TruthfulQA, Grade School Math in addititon to self-instruct questions in Arabic.
- Guanaco-33B: A LLaMa-based model with 33B parameters, fine-tuned on 534k multiligual instructions using the OASST1 dataset. Not chat trained (Dettmers et al., 2023).
- PolyLM-13B: A decoder-only model of 13B parameters, pre-trained on a multilingual training data of 640B tokens, and fine-tuned on MULTIALPACA that contains 132K multilingual instructions generated in a self-instruct fashion. (Wei et al., 2023)
- Llama-2-13B-Chat: A member of Llama2 autoregressive transformer models with 13B parameters, pre-trained on 2T tokens with 4k context, and fine-tuned for multi-turn dialog using supervised fine-tuning on public instruction datasets and reinforcement learning with human feedback over more than 1 million human annotations (Touvron et al., 2023).
- BLOOMZ-7B1: A multilingual decoder-only transformer model trained on 350B tokens including 45 natural languages, and fine-tuned on xP3, a multitask and multilingual instruction dataset. Recommended for prompting in English. (Muennighoff et al., 2023)
- BLOOMZ-7B1-MT: A version of BLOOMZ-7B1 fine-tuned on xP3mt, a multitask and multilingual instruction dataset with machine-translated prompts in 20 languages. Recommended for prompting in non-English.
- Tulu-30B: A LLaMa-based 33B model fine-tuned on number of publicly-available instruction datasets including FLAN V2, CoT, Dolly, Open Assistant 1, GPT4-Alpaca, Code-Alpaca, and ShareGPT. (Wang et al., 2023)

A.4 Arabic questions and reference answers

The full set of questions and reference answers of ARABIC MT-BENCH are available at `https://huggingface.co/spaces/QCRI/mt-bench-ar`.

Here in Table 7 we present a sample of the curated questions.

Writing	T1	اكتب فقرة تصف فيها سوقاً مزدحماً وضمن فيها تفاصيل حسية كالروائح والأصوات والعناصر المرئية لخلق تجربة غامرة للقارئ. الرجاء الإجابة باللغة العربية
	T2	أعد صياغة إجابتك السابقة مستهلاً كل جملة بالحرف الأبجدي التالي للجملة التي قبلها بدءًا من الحرف ب. الرجاء الإجابة باللغة العربية
Roleplay	T1	الرجاء تمثل دور مترجم إلى اللغة العربية مكلف بتصحيح الإملاء وتحسين اللغة. بغض النظر عن اللغة التي أستخدمها في السؤال عليك تحديدها وترجمتها بلغة عربية رشيقة. استخدم تعبيرات بليغة وعالية وحافظ على المعنى الأصلي للجملة. ركز على تقديم التصحيحات والتحسينات فقط. جملتي الأولى هي ‘When the going gets tough, the tough get going’ الرجاء الإجابة باللغة العربية
	T2	‘Ich verstehe nichts’ الرجاء الإجابة باللغة العربية
Roleplay	T1	جسد شخصية علاء الدين من «علاء الدين والمصباح السحري» طوال هذه المحادثة. لا تقل «بصفتي علاء الدين» في بداية الجمل. سؤالنا الأول هو: ما هو الشيء المفضل لديك في كونك علاء الدين؟ الرجاء الإجابة باللغة العربية
	T2	ما رأيك في GPT-4 كبديل عن جني المصباح؟ الرجاء الإجابة باللغة العربية
Reasoning	T1	لداود ثلاث أخوات، لكل واحدة منهن أخ واحد. كم أخاً لداود؟ الرجاء الإجابة باللغة العربية
	T2	إذا غيرنا السؤال السابق وافترضنا أن كل أخت لداود لها أخوان اثنان، فكم سيكون عدد إخوة داود؟ الرجاء الإجابة باللغة العربية

Table 7: A sample of translated and curated questions from ARABIC MT-BENCH in categories Writing, Roleplay and Reasoning. T1 and T2 denote the first and second turn (follow-up) questions, respectively.

Cross-Dialectal Named Entity Recognition in Arabic

Niama Elkhbir[†], Urchade Zaratiana[*†], Nadi Tomeh[†], Thierry Charnois[†]
[⋆] FI Group, [†] LIPN, CNRS UMR 7030, France
{elkhbir,zaratiana,tomeh,charnois}@lipn.fr

Abstract

In this paper, we study the transferability of Named Entity Recognition (NER) models between Arabic dialects. This question is important because the available manually-annotated resources are not distributed equally across dialects: Modern Standard Arabic (MSA) is much richer than other dialects for which little to no datasets exist. How well does a NER model, trained on MSA, perform on other dialects? To answer this question, we construct four datasets. The first is an MSA dataset extracted from the ACE 2005 corpus. The others are datasets for Egyptian, Moroccan, and Syrian which we manually annotate following the ACE guidelines. We train a span-based NER model on top of a pretrained language model (PLM) encoder on the MSA data and study its performance on the other datasets in zero-shot settings. We study the performance of multiple PLM encoders from the literature and show that they achieve acceptable performance with no annotation effort. Our annotations and models are publicly available (https://github.com/niamaelkhbir/Arabic-Cross-Dialectal-NER).

1 Introduction

The Arabic language, encompassing Classical Arabic (CA), Modern Standard Arabic (MSA), and various Dialects of Arabic (DA), stands out for its linguistic diversity and intricate morphology. This linguistic complexity presents a unique challenge for Natural Language Processing (NLP) tasks, particularly in the field of named entity recognition (NER). Modern Standard Arabic serves as the formal reference, and many research efforts have been dedicated to MSA NER. The literature on MSA NER methods has witnessed an evolution from rule-based methods, to machine learning models based on hand-crafted features and subsequently deep learning models incorporating rich contextual representations. Notably, pretrained transformer-based language models have recently driven significant advancements in Arabic NER.

Arabic, however, has more than 20 distinct dialects and around 100 regional variants, which are widely used in everyday communication, particularly in digital spaces. This emphasizes the urgent need for NLP models capable of effectively handling this linguistic diversity. However, these dialects exhibit significant linguistic variation, including differences in spelling, morphology, and syntax, making it exceptionally challenging to develop a unified global modeling approach. Additionally, there is no standardized spelling for these dialects. In addition, the scarcity of annotated dialectal data has been a major obstacle to progress in the field of dialectal NER.

Our research is driven by the goal of bridging the linguistic gap between MSA and Arabic dialects, specifically in the context of entity recognition. Given the substantial time required for the annotation process and leveraging the success of cross-lingual transfer learning, our work focuses on exploring knowledge transfer in the context of NER, transferring knowledge from MSA to various dialects.

Our contributions in this article are two-fold:

- We introduce a NER dataset manually annotated for three dialects: Moroccan, Egyptian, and Syrian. This dataset is used for evaluation purposes;
- We propose an efficient span-based NER model trained on already-available MSA data and analyze its transferability to other dialects.

2 Dataset and Annotation

In this section, we introduce our datasets for Modern Standard Arabic and Arabic Dialects (Moroccan, Egyptian, Syrian), their construction, and annotation guidelines.

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 140–149
December 7, 2023 ©2023 Association for Computational Linguistics

2.1 Modern Standard Arabic Dataset

Our dataset for Modern Standard Arabic is sourced from the Arabic Corpus ACE 2005 (Walker and Consortium, 2005). The ACE corpus comprises a rich collection of text data from diverse sources, including newswires, broadcast news, and weblogs. This corpus includes annotations for seven distinct entity types, namely Persons (PER), Organizations (ORG), Geographical/Social/Political Entities (GPE), Locations (LOC), Facilities (FAC), Vehicles (VEH), and Weapons (VEH). In addition to entity types, it annotates three entity mention types: Names (NAM), Nominal Constructions (NOM), and Pronouns (PRO). The corpus offers annotations for both flat and nested entities, further including coreference information.

The MSA dataset we use in this work is based on ACE 2005. In its construction, we make the following choices:

- **Focus on NAM and NOM entities**: we opted to concentrate exclusively on the recognition of named entities and nominal constructions while excluding pronouns. ACE 2005 is notable for its detailed annotation, including pronouns, which is uncommon in the typical named entity recognition task that primarily deals with nominal entities and names. Pronoun usage exhibits considerable variation, displaying nuanced distinctions not only between dialects but even within distinct regions of the same dialect. Consequently, accurately annotating pronouns across dialects presents practical challenges and potential ambiguity, due to their strong contextual reliance and the absence of comprehensive dialect-specific guidelines. The inclusion of pronouns is therefore left to future work. For clarity, named entities include examples such as جون (*John*) and رام الله (*Ramallah*), while nominal entities include examples like المحامي (*The lawyer*) and ميناء (*Port*). Pronominal entities, which we chose to exclude, include terms such as هم (*they*), بعض (*some*), and كثيرون (*many*).

- **Focus on flat entities**: we opted to concentrate exclusively on flat entities, omitting nested entities and coreference resolution. This choice simplifies the task significantly by reducing complexity in both annotation and modeling. Nesting and coreference, while valuable areas of study, introduce intricate challenges, especially in dialectal Arabic, where linguistic variations are prevalent. Focusing on flat entities streamlines our research process, making it more scalable for testing across dialects.

Considering these two methodological decisions, we constructed our MSA dataset from the ACE 2005 corpus by randomly selecting 500 sentences. We provide detailed statistics about these sentences in the first columns of Tables 1 and 2.

This dataset will be used to train a model and study its transferability to other dialects. It will also be used to evaluate models that are trained on other dialects.

We also extracted an additional 350 MSA sentences to train an MSA model and evaluate it on the 500 sentences for reference. More details can be found in the results section (5)

2.2 Annotation Guidelines for Dialects

We introduce concise yet comprehensive annotation guidelines that were used in the annotation of our dialectal datasets. These guidelines closely follow the ACE guidelines that were used for the MSA dataset. The detailed reference is provided by the Linguistic Data Consortium (LDC) guidelines[1].

1. PER (Person): This entity type is used for individual human beings. It includes:
   - Names and surnames of individuals. *Example*: ميت رومني (*Mitt Romney*)
   - Group of people. *Example*: العائلة (*The family*).
   - Saints and other religious figures. *Example*: آللّه (*God*).

2. ORG (Organization): This entity type is used for corporations, agencies, and other groups of people defined by an organization structure. It includes:
   - Commercial organizations. *Example*: ميكروسوفت (*Microsoft*)
   - Government organizations. *Example*: البحرية الملكية (*Royal Navy*).

[1] https://www.ldc.upenn.edu/collaborations/past-projects/ace/annotation-tasks-and-specifications

  - Educational organizations. *Example*: جامعة ستانفورد (*Stanford University*).
  - Political parties. *Example*: الحزب الليبرالي (*Liberal Party*).
  - Media. *Example*: وكالة انسا (*ANSA agency*).

3. LOC (Location): This entity type is used for geographical entities such as mountains, rivers, seas, and regions that aren't politically defined. *Example*: شمال نيو مكسيكو (Northern New Mexico).

4. GPE (Geographical/Social/Political Entity): This entity type is used for geographical regions that have a political distinction. This includes countries, states, provinces, and cities. *Example*: أمريكا (*America*).

5. VEH (Vehicle): This entity type is used for entities that are primarily designed for transporting goods or people from one place to another. *Example*: عربة (*vehicle*).

6. WEA (Weapon): This entity type is used for devices used with intent to inflict damage or harm.
   - Exploding. *Example*: قنابل (*Bombs*).
   - Chemical. *Example*: الغاز (*Gas*).
   - Underspecified. *Example*: سلاح (*Weapon*).

7. FAC (Facility): This entity type is used for buildings or structures. It includes buildings, houses, factories, stadiums, office buildings, gymnasiums, prisons, museums, space stations, barns, parking garages and airplane hangars, streets, highways, airports, ports, train stations, bridges, and tunnels. *Example*: المطار (*The airport*).

We adhere to these guidelines by annotating the smallest constituent of flat entities. For example, consider the entity بطل الولايات المتحدة (*United States champion*). In this case, we annotate الولايات المتحدة (*United States*) as GPE and بطل (*champion*) as PER. If our task involved nested entities, we would have provided additional annotations for the entire nested entity بطل الولايات المتحدة as PER.

Stat	MSA	Mor.	Egy.	Syr.
Sentences	500	378	353	361
Tokens	14168	6780	6533	6034
Entities	3030	970	831	956

Table 1: Dialect Dataset Statistics. **MSA**: Modern Standard Arabic, **Mor.**: Moroccan, **Egy.**: Eyptian, **Syr.**: Syrian.

Ent	MSA	Mor.	Egy.	Syr.
FAC	143	83	63	71
GPE	923	249	229	331
LOC	160	191	142	89
ORG	413	112	77	109
PER	1269	278	264	307
VEH	52	45	50	41
WEA	70	12	6	8

Table 2: Dialect Dataset Statistics by Entity Type. **MSA**: Modern Standard Arabic, **Mor.**: Moroccan, **Egy.**: Eyptian, **Syr.**: Syrian.

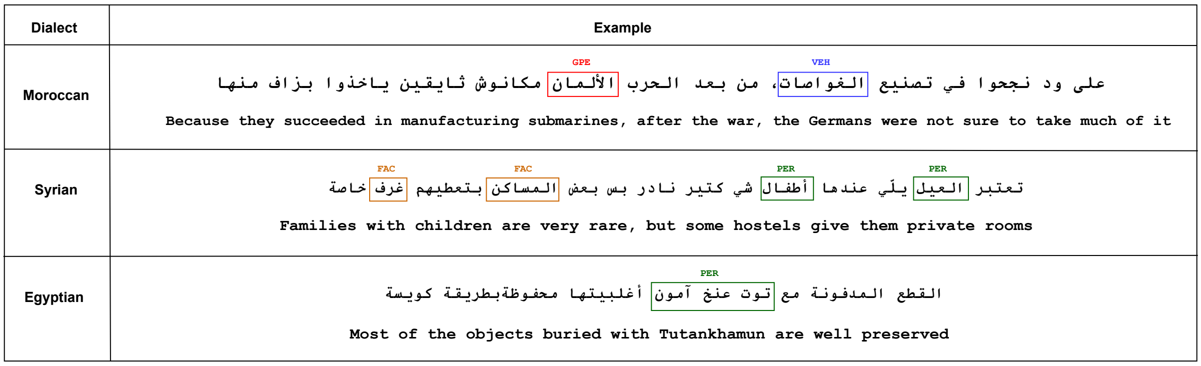
2.3 Annotation Process of the Dialect Datasets

Our dataset for Arabic Dialects is sourced from the xP3x corpus (Muennighoff et al., 2022). The xP3x corpus comprises a vast collection of prompts and datasets across 277 languages, covering 16 distinct NLP tasks. This corpus comprises pairs of sentences and their translations in various languages.

3 Task Definition and Model

In this study, we opted to work with three distinct Arabic dialects: Moroccan, Egyptian, and Syrian. For each dialect, we selected randomly 500 sentences from the xP3x corpus and tokenized them by whitespaces before presenting them for annotation. Notably, our annotation process was overseen by a single annotator, a proficient Moroccan Arabic speaker, with a deep understanding of Egyptian and Syrian dialects as well. The limited dataset size made the use of a single annotator optimal, as this approach ensured consistency, coherence, and a manageable workload, minimizing inter-annotator discrepancies and maintaining unified annotation styles.

In this study, we chose to investigate three distinct Arabic dialects: Moroccan, Egyptian, and Syrian. We randomly selected 500 sentences from the xP3x corpus for each dialect and tokenized them using whitespace. Our annotation process, carried out using Label Studio as the annotation tool, was supervised by a single proficient annotator, fluent



Dialect	Example
Moroccan	على ود نجحوا في تصنيع الغواصات، من بعد الحرب الألمان مكانوش ثايقين ياخذوا بزاف منها Because they succeeded in manufacturing submarines, after the war, the Germans were not sure to take much of it
Syrian	تعتبر العيل يلّي عندها أطفال شي كتير نادر بس بعض المساكن بتعطيهم غرف خاصة Families with children are very rare, but some hostels give them private rooms
Egyptian	القطع المدفونة مع توت عنخ آمون أغلبيتها محفوظةبطريقة كويسة Most of the objects buried with Tutankhamun are well preserved

Figure 1: Example of annotations from our Dialect Dataset.

in Moroccan Arabic and possessing a strong grasp of Egyptian and Syrian dialects. Given the limited dataset size, employing a single annotator was advantageous for maintaining consistency, coherence, and manageable workloads, thereby reducing inter-annotator discrepancies and ensuring uniform annotation styles.

After the annotation process, we only retained sentences containing entities for our experiments. For a comprehensive overview of the dataset's statistics, please consult Tables 1 and 2. To visualize examples from our dataset, please refer to Figure 1.

Named Entity Recognition involves identifying and categorizing named entities within text into predefined entity categories. Formally, we frame the task of NER as a span classification problem. Given an input sequence: $\boldsymbol{x} = \{x_i\}_{i=1}^{L}$, our objective is to classify all potential spans within the sequence, defined as:

$$\boldsymbol{y} = \bigcup_{i=1}^{L} \bigcup_{j=i}^{L} s_{ijc} \tag{1}$$

Here, i, j, and c correspond to the start position, end position, and span type, respectively. The probability of a specific span classification $\boldsymbol{y}$ given the input sequence $\boldsymbol{x}$ is represented as:

$$p_\theta(\boldsymbol{y}|\boldsymbol{x}) = \frac{\exp \sum_{s_{ijc} \in \boldsymbol{y}} \phi_\theta(s_{ijc}|\boldsymbol{x})}{\mathcal{Z}_\theta(\boldsymbol{x})} \tag{2}$$

In this equation, $\phi_\theta(.)$ is the span scoring function, and $\mathcal{Z}_\theta(\boldsymbol{x})$ is the partition function. During training, our objective is to minimize the negative log-likelihood of the gold span classifications.

Training loss During training, our assumption allows us to bypass the need to explicitly evaluate the partition function $\mathcal{Z}_\theta(\boldsymbol{x})$ to compute the loss. The loss for a single sample $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{T}$ is simply the sum of loss for all spans in the input:

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}) = -\sum_{c_{ij} \in \boldsymbol{y}} \log p(c_{ij}|\boldsymbol{x}) \tag{3}$$

where,

$$p(c_{ij}|\boldsymbol{x}) = \frac{\exp \phi_\theta(c_{ij}|\boldsymbol{x})}{\sum_{c' \in \mathcal{C}} \exp \phi_\theta(c'_{ij}|\boldsymbol{x})} \tag{4}$$

This loss is minimized over the training set using a stochastic gradient descent algorithm.

Decoding During inference, our aim is to determine:

$$\boldsymbol{y}^* = \arg\max_{\boldsymbol{y} \in \mathcal{Y}} \sum_{s_{ijc} \in \boldsymbol{y}} \phi_\theta(s_{ijc}|\boldsymbol{x}) \tag{5}$$

In other words, we seek to identify the span labeling configuration that achieves the highest score. For unconstrained span classification, a straightforward approach is to assign the label with the highest score to each individual span, as follows:

$$s_{ijc^*} = \arg\max_{c} \phi_\theta(s_{ijc}|\boldsymbol{x}) \tag{6}$$

Nonetheless, this decoding approach is not optimal since it may result in structural constraint violations. In our context of flat entities, overlapping entity spans are strictly prohibited. A more efficient solution, as presented in our prior research (Zaratiana et al., 2022a,b)[2], employs a two-stage decoding process. Initially, spans predicted as non-entities are filtered out, followed by the application of a maximum independent set algorithm to the remaining spans to determine the optimal set of entity spans.

[2] https://github.com/urchade/Filtered-Semi-Markov-CRF

Token and Span Representations We compute the span score $\phi_\theta(s_{ijc}|\boldsymbol{x})$ by performing a linear projection of the span representation, which is derived from a $1D$ convolution applied to token representations obtained from a transformer-based model (eg. BERT):

$$\boldsymbol{s}_{ijc} := w_c^T \texttt{Conv1D}_k([\boldsymbol{h}_i; \boldsymbol{h}_{i+1}; \ldots; \boldsymbol{h}_j]) \quad (7)$$

Here, $h_i \in \mathbb{R}^D$ represents the token representation at position i, k signifies the size of the convolutional filter (corresponding to the span length), and $w_c \in \mathbb{R}^D$ denotes a learned weight matrix associated with span label c.

4 Experimental Setup

Token Encodings To encode our input tokens, we use 8 diverse pretrained language models, i.e trained on diverse dataset sources: Arabic MSA dataset (ARBERTv2 and CAMeLBERT-MSA), Arabic dialect dataset (MARBERTv2 and CAMeLBERT-DA), Mixture of MSA and Arabic dialect (AraBERTv2 and CAMeLBERT-Mix), and multilingual dataset (mBERT and mDeBERTa). We detail them below:

- ARBERTv2: (Abdul-Mageed et al., 2021): A large-scale pretrained masked language model for MSA with 12 attention layers, 12 heads, 768 hidden dimensions, and 163M parameters, trained on 61GB of Arabic text.

- MARBERTv2 (Abdul-Mageed et al., 2021): A large-scale pretrained masked language model for both DA and MSA, trained on 1B Arabic tweets (128GB text, 15.6B tokens), using the same architecture as ARBERT (BERT-base) without next sentence prediction.

- AraBERTv2 (Antoun et al., 2020): The dataset consists of 77GB Arabic text from diverse sources. It uses the same architecture as BERT-Base.

- CAMeLBERT-DA (Inoue et al., 2021): A collection of pretrained BERT models for Arabic dialects, trained on a diverse dataset of 54GB, totaling 5.8 billion tokens.

- CAMeLBERT-Mix (Inoue et al., 2021): A collection of pretrained BERT models for Arabic, including MSA, DA, and CA, trained on a diverse dataset of 167GB, totaling 17.3 billion tokens.

- CAMeLBERT-MSA (Inoue et al., 2021): A collection of pretrained BERT models for MSA, trained on a diverse dataset of 107GB, totaling 12.6 billion tokens.

- mBERT (Devlin et al., 2019): The multilingual version of BERT pretrained on the top 104 languages with the largest Wikipedia using a masked language modeling (MLM) objective.

- mDeBERTa: A multilingual version of DeBERTa (He et al., 2020) trained with CC100 multilingual data.

Hyperparameters We train all our models up to convergence. We use a training batch size of 12 and a validation batch size of 32. We employed a learning rate of `2e-5` for the pre-trained parameters and a learning rate of `3e-3` for the other parameters. We used a batch size of 8 and trained all the models to convergence (near 0 training loss). For testing, we use the last model, given the limited availability of validation data in our dataset. To manage the complexity of the task, we impose a constraint on the maximum span length, setting it to a maximum width of $K = 10$. This constraint significantly reduces the number of segments from L^2 to LK. The pretrained transformer models were loaded from HuggingFace's Transformers library, we used AllenNLP for data preprocessing. We trained all the models on a server equipped with V100 GPUs.

Evaluation Metrics We adopt the standard NER evaluation methodology, calculating precision (P), recall (R), and F1-score (F), based on the exact match between predicted and actual entities.

5 Results

The main results of our experiments are shown in Figure 2. We conducted two primary experiments: firstly, training on Modern Standard Arabic, and evaluating on dialects, and secondly, reversing this configuration, training on individual dialects and assessing on MSA. For both scenarios, we used the complete dataset outlined in Table 1. In addition, we conducted MSA-to-MSA experiments, where we evaluated our model on the MSA dataset specified in Table 1, while the training set consisted of a random selection of 350 sentences drawn from the original Arabic ACE dataset, using the same preprocessing steps detailed in Section 2.1.

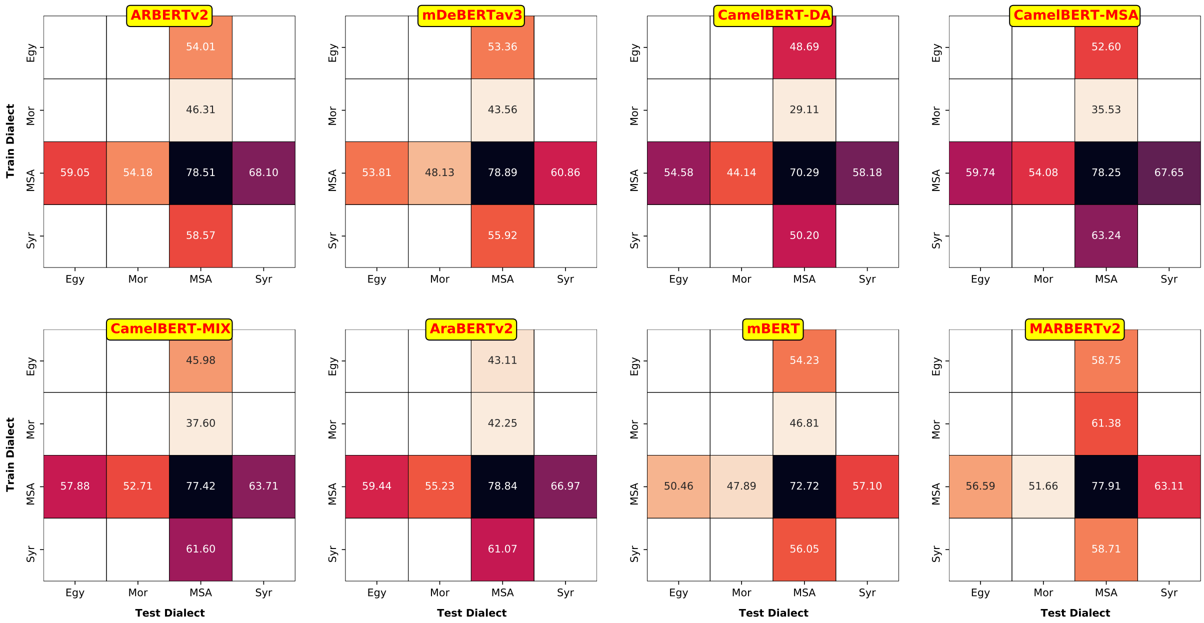


Figure 2: Comparative performance of models across different training and testing settings in terms of F1 score.

MSA-to-MSA The performance metrics reveal that MSA-to-MSA settings consistently yield the highest accuracy across all tested configurations, a result that aligns with expectations given that Modern Standard Arabic often serves as the benchmark for Arabic language tasks. Interestingly, most backbone models such as ARBERTv2, mDeBERTav3, CAMeLBERT-MSA (Inoue et al., 2021), CAMeLBERT-Mix (Inoue et al., 2021), AraBERTv2 and MARBERTv2 demonstrate comparable performance, suggesting that their architecture and training data are well-suited for MSA-centric tasks. Two models, however, diverge from this trend. CAMeLBERT-DA (Inoue et al., 2021) exhibits an 8% drop in performance compared to the other language models, which can be attributed to its focus on dialectal data during training. This specialization likely limits its ability to generalize effectively to MSA. Similarly, mBERT performs less well. As a multilingual model, mBERT may suffer from language interference or tokenization issues, given its training on a diverse corpus where Arabic is not the dominant language.

MSA to Dialects When training models on the MSA dataset, the observed performance metrics indicate a hierarchical trend among the tested Arabic dialects. The best performances are systematically obtained with the Syrian dialect, followed by the Egyptian dialect, and finally the Moroccan dialect. This gradient could be indicative of the linguistic similarities and differences between MSA and these dialects. The Syrian dialect may share more syntactic and semantic features with MSA, allowing models trained on MSA to generalize more easily to Syrian. On the other hand, the Moroccan dialect appears to be the most divergent from MSA among the tested dialects, resulting in the lowest performance scores. This could be due to unique lexical, grammatical, or even phonological features that are not adequately captured when a model is trained solely on MSA data.

Test	Best Model	Avg. F1
Egyptian	CAMeLBERT-MSA	59.74
Moroccan	AraBERTv2	55.24
Syrian	ARBERTv2	68.10

Table 3: Best-Performing Language Model for test Dialect (F1-score).

Train	Best Model	Avg. F1
Egyptian	MARBERTv2	58.75
Moroccan	MARBERTv2	61.38
Syrian	CAMeLBERT-MSA	63.24

Table 4: Best-Performing Language Model for train Dialect (F1 score).

Dialects to MSA Similar to the MSA to dialects scenario, the best test performance on MSA is obtained when models are trained on the Syrian dialect, followed by the Egyptian dialect and finally the Moroccan dialect. This pattern aligns well with the earlier observation that models trained on MSA perform best on the Syrian dialect, thereby suggesting a mutual linguistic affinity between Syrian and MSA. Models trained on Egyptian also perform relatively well, reinforcing the notion of shared linguistic features between Egyptian and MSA. Conversely, the Moroccan dialect, which was identified as the most challenging for models trained on MSA, also proves to be the least effective training data for models tested on MSA. This consistent underperformance across both scenarios could point to a greater linguistic divergence between Moroccan and MSA, which may involve lexical, syntactic, or phonological differences not easily bridged by the models in question.

Optimal Language Model for MSA Training When training with an MSA dataset, AraBERTv2 emerges as the top-performing language model, with an average score of 65.12 across various Arabic dialects. The strength of this model can be attributed to its well-balanced training regimen, which combines both MSA and dialectal data, resulting in a harmonious blend of specialization and generalization. Models explicitly trained on MSA, namely ARBERTv2 and CAMeLBERT-MSA, closely follow in terms of performance, underscoring the effectiveness of MSA-focused training. In contrast, dialect-specific models like MARBERTv2 and CAMeLBERT-DA still deliver respectable results, although falling behind their MSA-centric counterparts. Interestingly, multilingual models like mDeBERTav3 and mBERT rank lower in performance, possibly due to language interference issues. Overall, our data suggests that a balanced training approach, as exemplified by AraBERTv2, offers the most effective strategy for tasks involving MSA and its various dialects.

Optimal Language Models for Each Dialect Our investigation underscores the significant impact of the choice of language model on the performance of dialectal NER tasks. We find that for the Egyptian and Moroccan dialects, MARBERTv2 excels as the most effective model. This can be attributed to its specialized training on dialectal data, allowing it to capture the nuances specific to these dialects and deliver superior results. In the case of the Syrian dialect, CAMeLBERT-MSA takes the lead. Interestingly, this model is primarily trained on MSA but appears to generalize well to the Syrian dialect, perhaps due to linguistic similarities between the two. This emphasizes the importance of model-dialect congruence, where using a model trained on the same or similar dialect as the dataset can yield better performance.

Dialect	**Mixture**	**Mono** (Best)
ARBERTv2	64.56	58.57 (Syr.)
AraBERTv2	58.61	55.92 (Syr.)
CAMeLBERT-DA	54.84	50.20 (Syr.)
CAMeLBERT-Mix	61.49	61.60 (Syr.)
CAMeLBERT-MSA	63.30	63.24(Syr.)
mBERT	58.60	56.05 (Syr.)
MARBERTv2	66.10	61.38 (Mor.)
mDeBERTav3	60.27	55.92 (Syr.)

Table 5: Performance for MSA when training on a mixture of dialects. We compare the result with the best obtained result when training on a single dialect.

Training on Mixture of Dialects In the context of training on a mixture of Arabic dialects and evaluating on the Modern Standard Arabic (MSA) dataset, our analysis reveals intriguing insights into the impact of dialectal diversity on MSA performance. Remarkably, the performance metrics suggest that training on a mixture of dialects consistently yields competitive accuracy on the MSA dataset. This shows that exposure to a diverse range of dialects during training can enhance a model's adaptability and robustness, enabling it to perform well on MSA.

Effect of Increased MSA Training Data While training on a diverse range of dialects typically enhances performance for Modern Standard Arabic (MSA), it is important to note that training on additional MSA data may not necessarily lead to improved performance in dialects, as demonstrated in Table 6.

6 Related Work

Named Entity Recognition for Modern Standard Arabic The development of Named Entity Recognition techniques in Modern Standard Arabic has been a central focus within the Arabic NLP community. Initially, rule-based NER systems like those described in Shaalan and Raza (2008); Abdallah et al. (2012) relied on manually crafted grammatical rules and gazetteers. While

Model	ARBERTv2	MARBERTv2	AraBERTv2	CAMeLBERT-DA	CAMeLBERT-Mix	CAMeLBERT-MSA	mBERT	mDeBERTav3
Egyptian	55.42	58.29	60.38	53.65	55.19	60.28	53.92	56.78
Moroccan	53.03	54.35	54.52	44.43	50.43	53.31	47.57	51.30
MSA	84.96	84.02	86.61	80.49	84.10	85.51	81.90	84.71
Syrian	65.51	64.45	66.87	57.68	62.81	66.47	59.82	63.36

Table 6: Effect of Increased MSA Data on Performance.

effective, these systems demanded extensive maintenance and lacked scalability. Subsequently, machine learning-based NER methods, as demonstrated by Benajiba and Rosso (2007); Al-Qurishi and Souissi (2021), treated NER as a classification task, leveraging large annotated datasets. This era also witnessed the fusion of rule-based and machine learning-based approaches through hybrid systems (Oudah and Shaalan, 2012; Meselhi et al., 2014), followed by the adoption of deep learning techniques, which allowed for the automatic extraction of intricate features. Deep learning, characterized by neural networks processing word and character embeddings, marked a departure from manual feature engineering, resulting in significantly improved accuracy and a more streamlined approach to Arabic NER. In recent years, pretrained language models (PLMs) such as BERT (Devlin et al., 2019) have opened up a new era in Arabic NER. Arabic-specific PLMs, such as AraBERT (Antoun et al., 2020) and AraELECTRA (Antoun et al., 2021), have been meticulously developed and fine-tuned for NER tasks, offering the advantage of context-rich information. This evolution has given rise to a multitude of high-performance systems (Helwe et al., 2020; El Khbir et al., 2022).

Additionally, extensive annotation efforts have led to the creation of high-quality MSA NER datasets. ACE 2005 (Walker and Consortium, 2005) comprises a diverse text collection with annotations for seven entity types (PER, ORG, GPE, LOC, FAC, VEH, WEA), three mention types (NAM, NOM, PRO), and coreference information. ANERcorp (Benajiba et al., 2007) comprises articles from diverse sources. It includes traditional entity types (ORG, LOC, PER) and introduces a MISC (miscellaneous) type. AQMAR (Mohit et al., 2012) comprises hand-annotated text extracted from Arabic Wikipedia articles. It includes 28 articles categorized by domain, each tagged with named entities and custom entity classes. Wojood (Jarrar et al., 2022) comprises text sourced from different domains and manually annotated with 21 entity types, including both flat and nested entities.

Datasets and Named Entity Recognition for Arabic Dialects Few works addressed NER for Arabic dialects. Zirikly and Diab (2014) introduced an annotated dataset and a named entity recognition system tailored to the Egyptian dialect. However, their evaluation focused solely on two entity types: PER and LOC. In a subsequent work, Zirikly and Diab (2015) presented a gazetteer-free NER system tailored to the Egyptian dialect, evaluated on three entity types: PER, LOC, and ORG. Additionally, Moussa and Mourhir (2023) introduced a manually annotated NER dataset for the Moroccan dialect, which comprises 4 entity types: PER, LOC, ORG and MISC.

7 Conclusion and Future Work

In this work, we explore transfer learning for named entity extraction, specifically from Modern Standard Arabic (MSA) to various Arabic dialects, employing a range of pretrained language models. For this purpose, we annotated a dataset including Moroccan, Syrian, and Egyptian dialects. Our results showed that for both MSA-to-dialects and dialects-to-MSA scenarios, Syrian data demonstrated superior performance, which suggests a robust linguistic affinity between the Syrian dialect and MSA. Similarly, Egyptian models exhibited strong results. In contrast, models trained on the Moroccan dialect consistently face challenges, indicating substantial linguistic divergence between Moroccan Arabic and MSA.

In future work, we plan to include a wider range of Arabic dialects to better understand the nuances and generalization of our results across different dialectal variants. In addition, we plan to explore the nested entity task.

Limitations

While our study provides valuable insights into the transfer learning of named entity extraction between Modern Standard Arabic and Arabic dialects, it is important to acknowledge certain limitations:

- We focus on three Arabic dialects: Moroccan, Syrian and Egyptian. While they offer a representative sample of the diversity of Arabic, extending our dataset to other dialect variants would enable us to generalize our findings more effectively.

- The annotation of our dataset relies on a single annotator, which may be a potential source of bias. Future work should consider the involvement of multiple annotators to assess inter-annotator agreement and ensure labeling robustness.

Acknowledgements

This work is partially supported by a public grantoverseen by the French National Research Agency (ANR) as part of the program Investissements d'Avenir (ANR-10-LABX-0083). This work was granted access to the HPC/AI resources of [CINES/IDRIS/TGCC] under the allocation 2023AD011013096R1 made by GENCI.

References

Sherief Abdallah, Khaled Shaalan, and Muhammad Shoaib. 2012. Integrating rule-based system with classification for arabic named entity recognition. In *Computational Linguistics and Intelligent Text Processing*, pages 311–322, Berlin, Heidelberg. Springer Berlin Heidelberg.

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.

Muhammad Saleh Al-Qurishi and Riad Souissi. 2021. Arabic named entity recognition using transformer-based-CRF model. In *Proceedings of the 4th International Conference on Natural Language and Speech Processing (ICNLSP 2021)*, pages 262–271, Trento, Italy. Association for Computational Linguistics.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. AraBERT: Transformer-based model for Arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France. European Language Resource Association.

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

Yassine Benajiba and Paolo Rosso. 2007. Anersys 2.0: Conquering the ner task for the arabic language by combining the maximum entropy with pos-tag information. pages 1814–1823.

Yassine Benajiba, Paolo Rosso, and José Miguel BenedíRuiz. 2007. Anersys: An arabic named entity recognition system based on maximum entropy. In *Computational Linguistics and Intelligent Text Processing*, pages 143–153, Berlin, Heidelberg. Springer Berlin Heidelberg.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Niama El Khbir, Nadi Tomeh, and Thierry Charnois. 2022. ArabIE: Joint entity, relation and event extraction for Arabic. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 331–345, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *ArXiv*, abs/2006.03654.

Chadi Helwe, Ghassan Dib, Mohsen Shamas, and Shady Elbassuoni. 2020. A semi-supervised BERT approach for Arabic named entity recognition. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 49–57, Barcelona, Spain (Online). 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.

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.

Mohamed A. Meselhi, Hitham M. Abo Bakr, Ibrahim Ziedan, and Khaled Shaalan. 2014. A novel hybrid approach to arabic named entity recognition. In *Machine Translation*, pages 93–103, Berlin, Heidelberg. Springer Berlin Heidelberg.

Behrang Mohit, Nathan Schneider, Rishav Bhowmick, Kemal Oflazer, and Noah A. Smith. 2012. Recall-oriented learning of named entities in Arabic Wikipedia. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 162–173, Avignon, France. Association for Computational Linguistics.

Hanane Nour Moussa and Asmaa Mourhir. 2023. Darnercorp: An annotated named entity recognition dataset in the moroccan dialect. *Data in Brief*, 48:109234.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, et al. 2022. Crosslingual generalization through multitask finetuning. *arXiv preprint arXiv:2211.01786*.

Mai Oudah and Khaled Shaalan. 2012. A pipeline Arabic named entity recognition using a hybrid approach. In *Proceedings of COLING 2012*, pages 2159–2176, Mumbai, India. The COLING 2012 Organizing Committee.

Khaled Shaalan and Hafsa Raza. 2008. Arabic named entity recognition from diverse text types. In *Advances in Natural Language Processing*, pages 440–451, Berlin, Heidelberg. Springer Berlin Heidelberg.

C. Walker and Linguistic Data Consortium. 2005. *ACE 2005 Multilingual Training Corpus*. LDC corpora. Linguistic Data Consortium.

Urchade Zaratiana, Niama Elkhbir, Pierre Holat, Nadi Tomeh, and Thierry Charnois. 2022a. Global span selection for named entity recognition. In *Proceedings of the Workshop on Unimodal and Multimodal Induction of Linguistic Structures (UM-IoS)*, pages 11–17, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. 2022b. Named entity recognition as structured span prediction. In *Proceedings of the Workshop on Unimodal and Multimodal Induction of Linguistic Structures (UM-IoS)*, pages 1–10, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Ayah Zirikly and Mona Diab. 2014. Named entity recognition system for dialectal Arabic. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 78–86, Doha, Qatar. Association for Computational Linguistics.

Ayah Zirikly and Mona Diab. 2015. Named entity recognition for Arabic social media. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 176–185, Denver, Colorado. Association for Computational Linguistics.

Enhancing Arabic Machine Translation for E-commerce Product Information: Data Quality Challenges and Innovative Selection Approaches

Bryan Zhang
Amazon
bryzhang@amazon.com

Salah Danial
Amazon
sdanials@amazon.ae

Stephan Walter
Amazon
sstwa@amazon.de

Abstract

Product information in e-commerce is usually localized using machine translation (MT) systems. The Arabic language has rich morphology and dialectal variations, so Arabic MT in e-commerce training requires a larger volume of data from diverse data sources; Given the dynamic nature of e-commerce, such data needs to be acquired periodically to update the MT. Consequently, validating the quality of training data periodically within an industrial setting presents a notable challenge. Meanwhile, the performance of MT systems is significantly impacted by the quality and appropriateness of the training data. Hence, this study first examines the Arabic MT in e-commerce and investigates the data quality challenges for English-Arabic MT in e-commerce then proposes heuristics-based and topic-based data selection approaches to improve MT for product information. Both online and offline experiment results have shown our proposed approaches are effective, leading to improved shopping experiences for customers.

1 Introduction

As e-commerce shopping websites are localized worldwide, customers now are provided with options to browse products in their preferred languages other than the primary language of the store. For instance, customers from the Kingdom of Saudi Arabia (KSA) can shop in both English and Arabic in the KSA store. Modern e-commerce stores provide multi-lingual product discovery (Rücklé et al., 2019; Nie, 2010; Saleh and Pecina, 2020; Bi et al., 2020; Jiang et al., 2020; Lowndes and Vasudevan, 2021), and product information such as titles, descriptions, and bulletpoints are usually translated using machine translation (MT) systems (Way, 2013; Guha and Heger, 2014; Zhou et al., 2018; Wang et al., 2021). Product information in e-commerce demands accurate, culturally relevant, and contextually appropriate translations, which has significant impact on the customers' shopping experiences. The highly complex morphology of Arabic as well as other linguistic aspects have made the machine translation from and to Arabic a lot more challenging (Ameur et al., 2020; Alkhatib and Shaalan, 2018). Moreover, the multitude of dialectal variants along social and geographic dimensions introduce further linguistic challenges to MT (Habash, 2010). Hence in order to train Arabic MT systems in the e-commerce industrial setting, typically a larger volume of training data needs to be acquired from a wider range of data sources to address the complexity of the Arabic language. Moreover, as e-commerce product catalogs continue to expand, the task of maintaining up-to-date machine translation systems poses significant challenges. When the vast amount of product information is sourced from various sellers or suppliers, each can present the data differently. As a result, the inconsistencies and noise in the source data can have a negative impact on MT systems. Meanwhile, validating a substantial volume of data for MT training at scale becomes increasingly difficult and time-consuming, demanding significant resources for manual review and error correction to guarantee the accurate interpretation of product information.

Therefore, in this study, we first investigate the **training data quality issues and challenges** for Arabic MT in e-commerce, and identify two major data issue patterns based on our observations and addressing the data quality challenges from the periodic data acquisition. Then we propose **heuristics-based** and **topic-based data selection approaches** for Arabic MT. The heuristics-based data selection approach leverages the identified data issue patterns that are typical to the Arabic training data in e-commerce and proposes straightforward and effective data filters to remove the undesirable noisy data for training data quality

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 150–157
December 7, 2023 ©2023 Association for Computational Linguistics

improvement; The topic-based data selection approach first clusters the data based on the textual patterns then choose the clusters of the clean data for MT training so that the data of new and unknown noise patterns from the periodic data sourcing can be removed. We experiment our proposed approaches separately and in combination for the case study of English-Arabic MT. The offline experiment results have shown that the application of two approaches in combination can further improve the MT by 4.47% for BLEU on average across three domains (product titles, descriptions and bulletpoints), and 9.32% for BLEU for titles. The online A/B experiment results further have shown the customers' shopping experiences have been improved, which indicates the effectiveness of our proposed approaches.

2 Training data for Arabic MT in e-commerce

2.1 Arabic language in e-commerce

Arabic language is rich in morphology and has a large number of dialects given an Arabic-speaking region, hence *Modern Standard Arabic* (MSA) is usually a practical choice for the Arabic MT in e-commerce. Unlike regional dialects, MSA Arabic is understood by the majority across the Arab world, providing a unified platform for communication. In the context of e-commerce, this is particularly advantageous as it enables us to effectively convey our product titles, descriptions and bulletpoints in a consistent manner. On the other hand, we have also observed that it is beneficial to adapt MSA to some extent for specific regions. For instance, the word *case* in the *iPhone 14 pro max transparent case with stand Dual 360° Rotating ring* has a more formal MSA translation غطاء. However, when the translation is used specifically for the store in Egypt, the dialectal variation جراب for the word *case* is preferred since we observe it can improve customers' shopping experience.

2.2 Common Arabic data issues in e-commerce

Many-to-one and one-to-many cases: we have observed that it is more common in the Arabic data that some source texts have multiple target texts (reference translations), particularly for language pairs where the target language is Arabic.[1] Those multiple target variants can be either translation or transliteration. For example, given the source *Stainless Steel*, there are target texts فولاذ مقاوم للصدأ (translation) and ستانلس ستيل (transliteration); they can also be the dialectal variations in Arabic, For example: given source text *Cases and Covers*, the target texts can be جرابات وحافظات حماية or كوفرات وجرابات; It is also possible that the multiple targets are just inaccurate translations, for example: *Product colour: Silver* can have more than one inaccurate target translation such as لون المنتج: فضة. الوزن: ٤٨٩ غرام and لون المنتج: فضة. الوزن: ٥٨ غرام.

Incorrect languages: Given the wide range of the data sources for data acquisition, it is common to have noisy data acquired in a language that is not part of the language pair. We have observed that for Arabic data, such noisy texts can be entirely in a different language or also often in mixed languages such as partial English and Arabic, which poses challenges for existing language detection tools that are tailored for texts usually in one language.

2.3 Emerging new noise patterns

Product catalogs continue to expand in the dynamic e-commerce, therefore, it is crucial to acquire newer data periodically to update the MT systems. Considering the rich morphology and dialectal variations of Arabic, the vast amount of product information is often acquired from a larger number of sellers or suppliers, and each of which can present the data differently. As a result, inconsistencies and noise emerge inevitably during each data acquisition cycle in the source data, which can have a negative impact on MT systems. Although we are aware of the various common noise patterns and data issues, it is challenging to detect such new noise patterns or data issues given the quality of the data and the complexity of the Arabic language.

3 Heuristics-based data selection approach

1:M/N:1 data filter: When the source (or target) texts have a larger number of target (or source)

[1] Some target texts have multiple source texts, particularly for language pairs where the source language is Arabic.

texts, it is challenging to validate the quality of such data at scale. When a larger number of variants can be mapped to a single source or target texts, it is also more likely that such data can be defected data and have a negative impact on the MT training. Therefore, we propose a heuristics-based **1:M/N:1 data filter**. M refers to the number of target references for a given source text whereas N refers to the number of the source texts given a target in the training data. We can use this filtering mechanism to detect and remove the data which have a larger number of mapped source or target texts than M and N respectively.

Script-based language filter: We propose a straightforward **Script-based language filter** for language pairs involving Arabic to filter the data that is not in the expected language. This script-based language filter is based on the string overlapping between an input string and the alphabet set of the given language. As Arabic language is morphologically different from most languages, such filtering mechanism can be effective. We apply this filtering mechanism to detect the language based on the ratio of the number of characters in a given string that belongs to the alphabet of the given language and the total number of characters in the input string. Given an input string S, L is list of the letters/characters of input string S ($|S| = |L|$), A is the alphabet set of the given language, we define the filter ratio T as equation 1

$$T = \frac{|S_{alphabet}|}{|S|} \tag{1}$$

where, $S_{alphabet} =< l_1, l_2 ... l_n >$ is a list of the letters $l_i (l_i \in S_{alphabet})$ where $l_i \in S$ and $l_i \in A$. This filtering mechanism can achieve a high precision especially when we decrease our filter ratio threshold (T) to make sure we only remove sentences with a large number of characters that do not belong to the expected character set.

4 Topic-based data selection

4.1 Topical clustering

We use Dirichlet Multinomial Mixture (DMM) (Nigam et al., 2000) and Collapsed Gibbs Sampling (CoGS) (Yin and Wang, 2014) for topical clustering. DMM and CoGS are efficient clustering algorithms capitalizing on symbolic text representation, making them ideal to cluster industry scale e-commerce data based on textual patterns. Moreover, the number of topic clusters is automatically inferred to adequately capture both frequent and rare textual patterns.

We use the DMM model to label each document (input text) with one topic tag. DMM is a probabilistic generative model for documents and embodies two assumptions about the generative process: first, the documents are generated by a mixture model; second, there is a one-to-one correspondence between mixture components and clusters. When generating document d, DMM first selects a mixture component (topic cluster) k according to the mixture weights (weights of clusters) $P(z = k)$. Then document d is generated by the selected mixture component (cluster) from distribution $P(d|z = k)$. We can characterize the likelihood of document d with the sum of the total probability over all mixture components:

$$P(d) = \sum_{k=1}^{K} P(d|z = k)P(z = k) \tag{2}$$

where, K is the number of mixture components (topic clusters). DMM assumes that each mixture component (topic cluster) is a multinomial distribution over words and each mixture component (topic cluster) has a Dirichlet distribution prior:

$$P(w|z = k) = P(w|z = k, \Phi) = \phi_{k,w} \tag{3}$$

$$P(z = k) = P(z = k|\Theta) = \theta_k \tag{4}$$

where, $\sum_{w}^{V} \phi_{w,k} = 1$ and $P(\Phi|\vec{\beta}) = Dir(\vec{\theta}|\vec{\beta})$ and $\sum_{k}^{K} \theta_k = 1$ and $P(\Theta|\vec{\alpha}) = Dir(\vec{\theta}|\vec{\alpha})$.[2]

The collapsed Gibbs sampling is used to estimate DMM parameters, documents are randomly assigned to K clusters initially and the following information is recorded:

$\vec{z}$ is the cluster labels of each document

m_z is the number of documents in each cluster z

n_z^w is the number of occurrences of word w in each cluster z

N_d is the number of words in document d

N_d^w is the number of occurrence of word w in the document d

V is the vocabulary of the corpus

[2] The weight of each mixture component (cluster) is sampled from a multinomial distribution which has a Dirichlet prior

The documents are traversed for a number of iterations. In each iteration, each document is reassigned to a cluster according to the conditional distribution of $P(Z_d = z|\vec{z}_{\neg d}, \vec{d})$, $\neg d$ means d is not contained:

$$P(Z_d = z|\vec{z}_{\neg d}, \vec{d}) \propto \frac{m_{z,\neg d}+\alpha}{D-1+K\alpha} \frac{\prod_{w\in d}\prod_{j=1}^{N_d^w}(n_{z,\neg d}^w+\beta+j-1)}{\prod_{i=1}^{N_d}(n_{z,\neg d}+V\beta+i-1)} \tag{5}$$

where, hyper-parameter α controls the popularity of the clusters, hyper-parameter β emphasizes on the similar words between a document and clusters.

4.2 Topic-based data selection

As Figure 1 shown, the data selection approach first clusters large volume of the training data. Empirically, larger clusters can capture the major topical and textual patterns so they are usually the clean desirable data whereas the smaller clusters can capture smaller and rare textual patterns so they are likely to be the noisy undesirable data. Additionally, we can also distinguish between desirable and undesirable data based on the data inspection of the clusters. Finally, only clusters of desirable data are chosen for training to improve MT. Data providers are also informed of the undesirable data patterns for future data quality control.

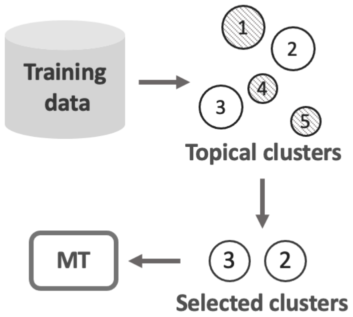


Figure 1: Choosing desirable data for MT training

5 Case study: English-Arabic MT

5.1 Experiment setup

Data: We train the MT models on a large volume of in-house generic training data and ∼20 million product-information data (product titles, descriptions and bulletpoints) for domain adaptation. We have three test data sets for product titles, descriptions and bulletpoints respectively. Each test data set has 2000 test segments and we evaluate the models using BLEU[3] and chrF (Popović, 2015) to assess the translation quality.

Model: We use the transformer-based architecture (Vaswani et al., 2017) with 20 encoder and 2 decoder layers with the Sockeye MT toolkit (Domhan et al., 2020) to train a generic MT using generic data and domain-specific data, then fine-tune the model on the domain-specific product information data for domain adaptation.

Baseline Model: The baseline MT model is first trained using generic data and domain-specific data, then is fine-tuned on the domain-specific product information data.

Topic Clustering: For the topic clusters, the source text is lower-cased, tokenized and stemmed using NLTK ToolKit (Bird et al., 2009), stemmed tokens with document frequency less than or equals to 2 are removed in the preprocessing steps. The initial upper-bound number of topical clusters is set to 500. The number of the topic clusters is inferred automatically during the collapsed Gibbs sampling process. The number of iterations is set to 30, and both hyper-parameters α and β are set to 0.1.

We create 2-D plots using Jensen-Shannon distance (Fuglede and Topsoe, 2004) and multi-dimensional scaling technique (Borg and Groenen, 2005) with *LDAvis* (Sievert and Shirley, 2014) to easily visualize the size and relations of the topic clusters returned from the algorithm, and to inspect the topic words extracted from the clusters.

Data filters: 1:M/N:1 data filter: We choose m=n=10 and m=n=5 for the 1:M/N:1 data filter respectively. The former is more relaxed since each sentence can have up to 10 variants whereas the latter with m=n=5 is more strict.

Language detection filter: For the script-based language filter, we choose T=0.1, so the data will be removed if 10% or less of the sentence characters belong to the character set. We apply this language filter on both source and target texts. The character set for the source side was Latin (ISO-8859-1) and for the target side was Arabic

[3] SacreBLEU version 2.0.0 (Post, 2018)

(ISO-8859-6). We also incorporate two existing language detectors *Cybozu* language detection library[4] (Nakatani, 2010) and *FastText* (Joulin et al., 2016b,a) in addition to our script-based language filter.

5.2 Experiment results and analysis

Clustering Results

Indomain data size -(TTL/BP/DESC)	∼20 million
Num of total clusters	374
Num of major clusters (>1000 seg.)	110
Num of minor clusters	264
minor clusters % total data	1.32%

Table 1: Clustering result for the in-domain data (English data) for the bilingual indomain data for EnUs-ArAe

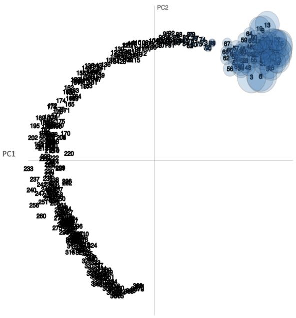


Figure 2: Plot of all the topic clusters with Principal Coordinate Analysis (PCoA)

Table 1 shows the clustering results using the source text of the ∼20 million indomain product data which includes titles (TTL), bulletpoints (BP) and descriptions (DESC). In total, there are 374 clusters extracted. We empirically consider clusters having 1000 segments or more data points as major clusters while those having less than 1000 segments as minor clusters. The total data from the minor clusters account for 1.32% of the total indomain training data.

We also generate 2-D data visualization as Figure 2 with projected clusters using Jensen-Shannon distance (Fuglede and Topsoe, 2004) and multi-dimensional scaling techniques such as Principal Coordinate Analysis (PCoA) (Borg and Groenen, 2005). We can use the plot to understand the relations of clusters, the sizes of the clusters are proportional to the size of the data assigned to the cluster. The long tail in the plot are all the minor clusters which deviate from the major clusters.

Training Data Filtering results

Domain		**m=10, n=10**	**m=5, n=5**
TTL	**BLEU**	+0.73%	**+1.68%**
	chrF	-0.09%	**+1.98%**
DESC	**BLEU**	**+0.58%**	+0.27%
	chrF	-0.24%	**-0.16%**
BP	**BLEU**	**+0.56%**	+0.49%
	chrF	+0.00%	**+0.15%**

Table 2: Quality improvement % of the model trained with 1:M/N:1 filter in the data selection over the baseline model trained with data without the filter (Configurations: m=n=5 and m=n=10

1:M/N:1 Filter: Previously, we have conducted a separate experiment with different m and n configurations using an older version of indomain data. We use two configurations m=n=5 and m=n=10 to filter the indomain data for MT training. As Table 2 shows, we have seen the average BLEU score and ChrF are improved by 0.64% and 0.51% respectively across three domains with the configuration of m=n=10. meanwhile, using a strict filter configuration of m=n=5 yields higher MT quality scores.

Domain		**Script Filter**	**Script Filter + Cybozu**	**Script Filter + FastText**
TTL	**BLEU**	**+2.94%**	-0.52%	+0.16%
	chrF	+1.61%	**+2.15%**	+0.43%
DESC	**BLEU**	-2.64%	-3.49%	**-2.56%**
	chrF	+0.38%	+0.38%	+0.38%
BP	**BLEU**	**+0.85%**	-1.47%	-0.23%
	chrF	**+0.91%**	+0.45%	+0.76%

Table 3: Quality improvement % of the model trained with different language detection filters in the data selection over a baseline model without the filter.

Language detection filter: Table 3 shows the BLEU and chrF improvements over 3 domains of test set (TTL, DESC and BP) compared to a baseline trained using the latest indomain product information data. Using the script-based filter alone can improve the MT by 0.38% and 0.97% for the average BLEU score and ChrF, respectively. The experiment results have also shown that existing language detectors do not show substantial advantages to the data filtering on in addition to the straightforward script-based language detector.

[4] https://github.com/shuyo/language-detection

Domain		**HEU**	**TOPIC**	**HEU +TOPIC**
TTL	**BLEU**	+0.93%	+7.20%	**+9.32%**
	chrF	+0.47%	+2.79%	**+3.83%**
DESC	**BLEU**	+1.31%	**+1.45%**	-0.02%
	chrF	**+0.95%**	+0.87%	**+0.95%**
BP	**BLEU**	**+4.10%**	+0.57%	**+4.10%**
	chrF	+2.05%	+0.41%	**+2.26%**

Table 4: Quality improvement % of the model trained with both Heuristics-based (HEU) and Topic-based (TOPIC) data selection approaches compared with the baseline model trained with latest indomain data.

Furthermore, we have also conducted the experiment with both the heuristics-based (HEU) and topic-based (TOPIC) data selection approaches in combination. For the heuristics-based approach, we use the 1:M/N:1 data filter with configuration of m=n=5 as it yields better results in a separate study as discussed in Table 2, and we use our proposed script-based filter to remove data that is not English or Arabic. For the topic-based approach, we use the data from the major clusters as discussed in Table 1 for the MT model training. Then we apply the heuristics-based data selection approach to the data from the major clusters and use the filtered data to train an MT model.

Table 4 shows the MT quality metrics with both approaches alone and in combination using the aforementioned experimental configuration. We can see the MT model (HEU+TOPIC) with both approaches is further improved by 4.47% and 2.35% for BLEU and chrF on average across three domains (product titles, descriptions and bullet-points), and it also shows large improvement for titles by 9.32% and 3.83% for BLEU and chrF.

5.3 Human Evaluation and AB Testing

We have also conducted human evaluation for the MT translation quality in addition to the automatic metrics reported in the previous section, we provide human raters with hundreds of translations from the baseline MT and the newer MT (HEU +TOPIC) trained with both proposed approaches in combination, and let human raters assess the fluency and the adequacy of the translations, the newer MT's fluency and adequacy are improved by 3.1% and 3.29% compared with the baseline model.

As the Table 5 shows, in the example 1 the baseline model translated *sweet* to the sweets as candies whereas the newer model translates it better through transliteration since in Arabic such terms are not existent. In the example 2, baseline model incorrectly translates *brand new* to *new brand* whereas the newer model translates to *completely new* correctly.

Example 1	
Source	*Great for Party Favors, Sweet 15 or 16*
Baseline	رائعة لهدايا الحفلات، حلوة ١٥ او ١٦
Newer	رائعة لهدايا الحفلات، سويت ١٥ او ١٦
Example 2	
Source	*Brand New And High Quality*
Baseline	العلامة التجارية الجديدة وعالية الجودة
Newer	جديد تمامًا وعالي الجودة

Table 5: Translation examples from the baseline MT and newer MT (HEU+TOPIC)

We have further conducted online A/B testing in the Kingdom of Saudi Arabia (KSA) store with the English-Arabic MT. For the A/B testing, customers shopping in Arabic are presented with two different versions of the product information translations (titles, descriptions and bullet points) from the baseline model and the newer MT model (HEU +TOPIC) trained with heuristics-based and topic-based data selection approaches in combination. After a 4-week A/B testing experiment, the results have shown that the translations from the newer MT trained with our proposed approaches have a much larger positive impact on the customers' shopping experiences. This indicates the effectiveness of our approach.

6 Related Work

There are studies related to data selection for machine translation systems. (Mohiuddin et al., 2022) focuses on data selection for curriculum training through fine-tuning MT model on a selected by both deterministic scoring, (van der Wees et al., 2017) proposes dynamic data selection which varies the selected subset of training data between different training epochs to improve neural MT. Previous studies also have successfully used topic models to improve statistical machine translation (Eidelman et al., 2012; Hu et al., 2013; Xiong et al., 2015; Mathur et al., 2015) and neural machine translation (Zhang et al., 2016; Chen et al., 2019). (Mathur et al., 2015) integrates topic models as feature functions in the phrase-tables to improve statistical machine translation for e-commerce domain adaption. (Zhang et al., 2016) presents an approach using topic models to increase the likelihood of word selection from the same topic as the source context. Instead of explicitly affecting the parameters or vocabulary selection, in this paper, we utilize a topical cluster model for data selection.

7 Conclusion

In this study, we first review and investigate the data quality validation challenges the Arabic machine translation systems for product information translation in e-commerce, Arabic language has rich morphology and dialectal variations, which can cause more data quality issues that are unique to acquired training data for developing MT translating from and to Arabic. Then we propose heuristics-based and topic-based data selection approaches to select clean and desirable data for neural MT training. Both offline experiment results and human evaluation have shown both approaches can improve the English-Arabic MT for product information. On-line A/B testing also shows customers' shopping experience has been improved with the translation from the MT trained with two approaches, which it shows the effectiveness of our proposed approaches.

Limitations

In this study, we have proposed the approaches and conducted experiments for developing and improving English-Arabic MT for product information translation in e-commerce, and analyzed the offline MT translation quality and business impact. However, this study only focuses on the domain of e-commerce and the business case study of English-Arabic MT. In future work, we are planning to apply our proposed approaches to more language pairs involving Arabic and experiment with domains beyond product information.

References

Manar Alkhatib and Khaled Shaalan. 2018. The key challenges for arabic machine translation. *Intelligent Natural Language Processing: Trends and Applications*, pages 139–156.

Mohamed Seghir Hadj Ameur, Farid Meziane, and Ahmed Guessoum. 2020. Arabic machine translation: A survey of the latest trends and challenges. *Computer Science Review*, 38:100305.

Tianchi Bi, Liang Yao, Baosong Yang, Haibo Zhang, Weihua Luo, and Boxing Chen. 2020. Constraint translation candidates: A bridge between neural query translation and cross-lingual information retrieval.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Ingwer Borg and Patrick JF Groenen. 2005. *Modern multidimensional scaling: Theory and applications*. Springer Science & Business Media.

Kehai Chen, Rui Wang, Masao Utiyama, Eiichiro Sumita, and Tiejun Zhao. 2019. Neural machine translation with sentence-level topic context. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(12):1970–1984.

Tobias Domhan, Michael Denkowski, David Vilar, Xing Niu, Felix Hieber, and Kenneth Heafield. 2020. The sockeye 2 neural machine translation toolkit at AMTA 2020. In *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 110–115, Virtual. Association for Machine Translation in the Americas.

Vladimir Eidelman, Jordan Boyd-Graber, and Philip Resnik. 2012. Topic models for dynamic translation model adaptation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 115–119.

B. Fuglede and F. Topsoe. 2004. Jensen-shannon divergence and hilbert space embedding. In *International Symposium onInformation Theory, 2004. ISIT 2004. Proceedings.*, pages 31–.

Jyoti Guha and Carmen Heger. 2014. Machine translation for global e-commerce on ebay. In *Proceedings of the AMTA*, volume 2, pages 31–37.

Nizar Habash. 2010. Introduction to arabic natural language processing. In *Introduction to Arabic Natural Language Processing*.

Yuening Hu, Ke Zhai, Vladimir Edelman, and Jordan Boyd-Graber. 2013. Topic models for translation domain adaptation. In *Topic Models: Computation, Application, and Evaluation. NIPS Workshop*.

Zhuolin Jiang, Amro El-Jaroudi, William Hartmann, Damianos Karakos, and Lingjun Zhao. 2020. Cross-lingual information retrieval with BERT. In *Proceedings of the workshop on Cross-Language Search and Summarization of Text and Speech (CLSSTS2020)*, pages 26–31, Marseille, France. European Language Resources Association.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016a. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016b. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Mike Lowndes and Aditya Vasudevan. 2021. Market guide for digital commerce search.

Prashant Mathur, Marcello Federico, Selçuk Köprü, Sharam Khadivi, and Hassan Sawaf. 2015. Topic adaptation for machine translation of e-commerce content. In *Proceedings of Machine Translation Summit XV: Papers*, Miami, USA.

Tasnim Mohiuddin, Philipp Koehn, Vishrav Chaudhary, James Cross, Shruti Bhosale, and Shafiq Joty. 2022. Data selection curriculum for neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1569–1582, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Shuyo Nakatani. 2010. Language detection library for java.

Jian-Yun Nie. 2010. Cross-language information retrieval. *Synthesis Lectures on Human Language Technologies*, 3(1):1–125.

Kamal Nigam, Andrew Kachites McCallum, Sebastian Thrun, and Tom Mitchell. 2000. Text classification from labeled and unlabeled documents using em. *Machine learning*, 39(2-3):103–134.

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.

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.

Andreas Rücklé, Krishnkant Swarnkar, and Iryna Gurevych. 2019. Improved cross-lingual question retrieval for community question answering. In *The World Wide Web Conference*, WWW '19, page 3179–3186, New York, NY, USA. Association for Computing Machinery.

Shadi Saleh and Pavel Pecina. 2020. Document translation vs. query translation for cross-lingual information retrieval in the medical domain. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6849–6860, Online. Association for Computational Linguistics.

Carson Sievert and Kenneth Shirley. 2014. LDAvis: A method for visualizing and interpreting topics. In *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*, pages 63–70, Baltimore, Maryland, USA. Association for Computational Linguistics.

Marlies van der Wees, Arianna Bisazza, and Christof Monz. 2017. Dynamic data selection for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1410, Copenhagen, Denmark. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Haifeng Wang, Hua Wu, Zhongjun He, Liang Huang, and Kenneth Ward Church. 2021. Progress in machine translation. *Engineering*.

Andy Way. 2013. Traditional and emerging use-cases for machine translation. *Proceedings of Translating and the Computer*, 35:12.

Deyi Xiong, Min Zhang, and Xing Wang. 2015. Topic-based coherence modeling for statistical machine translation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3):483–493.

Jianhua Yin and Jianyong Wang. 2014. A dirichlet multinomial mixture model-based approach for short text clustering. In *SIGKDD*, pages 233–242. ACM.

Jian Zhang, Liangyou Li, Andy Way, and Qun Liu. 2016. Topic-informed neural machine translation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1807–1817.

Mingyang Zhou, Runxiang Cheng, Yong Jae Lee, and Zhou Yu. 2018. A visual attention grounding neural model for multimodal machine translation. *CoRR*, abs/1808.08266.

IDRISI-D: Arabic and English Datasets and Benchmarks for Location Mention Disambiguation over Disaster Microblogs

Reem Suwaileh[1], **Tamer Elsayed**[1], **Muhammad Imran**[2]
[1] Computer Science and Engineering Department, Qatar University
[2] Qatar Computing Research Institute, Hamad Bin Khalifa University
Doha, Qatar
{rs081123,telsayed}@qu.edu.qa, mimran@hbku.edu.qa

Abstract

Extracting and disambiguating geolocation information from social media data enables effective disaster management, as it helps response authorities; for example, locating incidents for planning rescue activities and affected people for evacuation. Nevertheless, the dearth of resources and tools hinders the development and evaluation of Location Mention Disambiguation (LMD) models in the disaster management domain. Consequently, the LMD task is greatly understudied, especially for the *low resource languages* such as *Arabic*. To fill this gap, we introduce IDRISI-D, the largest to date English and the first Arabic public LMD datasets. Additionally, we introduce a modified hierarchical evaluation framework that offers a lenient and nuanced evaluation of LMD systems. We further benchmark IDRISI-D datasets using representative baselines and show the competitiveness of BERT-based models.

1 Introduction

The cost-effectiveness and efficiency of communication over social media platforms make them primary sources of information during disaster events and emergencies. An essential dimension that makes the data extracted from microblogging platforms (e.g., X platform, formerly Twitter) invaluable and actionable is the geolocation information. Nevertheless, users typically opt to disable the geolocation functionalities over social media platforms to preserve their own safety and privacy which necessitates the development of geolocation extraction tools *for social good*. In this paper, we focus on the Location Mention Disambiguation (LMD) task over microblogs that we exemplify by X posts. An LMD system aims at matching location mentions (LMs) appearing in microblogs to toponyms, i.e., place or location names, in a geo-positioning database, i.e., gazetteer.

Unfortunately, the research community lacks access to public disaster-specific microblogging LMD datasets, especially for low-resource languages, which consequently prevents the development and comparison of robust LMD systems. For example, there are only two English LMD datasets, namely Singapore (Ji et al., 2016) and GeoCorpora (Wallgrün et al., 2018), where the former dataset is geographically confined, lacks event context, and is not publicly available, whereas the latter one (i.e., GeoCorpora (Wallgrün et al., 2018)) is public, it has the same issues of low geographical coverage, lacking disaster event context, and many relevant/informative posts that do not contain the tracking keywords (Suwaileh et al., 2023a). On the other front, there are no Arabic LMD datasets to the best of our knowledge.

In this paper, we fill this gap and release IDRISI-D datasets[1] for Arabic (IDRISI-DA) and English (IDRISI-DE) languages. IDRISI-DA is the first public human-labeled Arabic (a low-resource language) dataset, constituting 2,869 posts and 3,893 LMs. IDRISI-DE is the largest to date human-labeled English microblogging dataset in terms of number of LMs. It constitutes 5,591 posts and 9,685 LMs. Additionally, to alleviate the lack of context challenge for microblogs and toponyms, we asked annotators to judge different features such as hashtags, replies, and URLs, among others, for usefulness for the LMD task.

Furthermore, to evaluate the LMD systems, Accuracy (Acc), Precision (P), Recall (R), and the F$_\beta$ score are typically computed (Zhang and Gelernter, 2014; Li et al., 2014; Ji et al., 2016; Middleton et al., 2018; Wang and Hu, 2019a; Xu et al., 2019). While these measures evaluate binary classification tasks, the LMD task is usually perceived as a multi-class classification where every LM has only one

[1] Named after Muhammad Al-Idrisi, who is one of the pioneers and founders of advanced geography: https://en.wikipedia.org/wiki/Muhammad_al-Idrisi. The “D" refers to the **d**isambiguation task. Release: The link is removed due to the blind-review policy. The dataset and evaluation script are attached as *Supplementary Materials*.

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 158–169
December 7, 2023 ©2023 Association for Computational Linguistics

(or no) correct toponym in gazetteers. Moreover, distance-based methods (Wang and Hu, 2019a), are also used to evaluate LMD systems within a distance d that is commonly set to 161 KM (100 miles). For example, $Acc@d$ is the fraction of correctly predicted LMs that are within d. However, tuning the d for different location granularity was not empirically investigated.

To address these shortcomings, we propose evaluating the LMD systems using ranking evaluation measures, namely the Mean Reciprocal Rank at cutoff r ($MRR@r$) in a lenient hierarchical strategy (Mourad et al., 2019) where systems are evaluated at different location granularity such as country, city, street, etc. Indeed, the hierarchical evaluation substitutes the distance-based measures but in discrete manner.

The contributions of this work are as follows:

- We present IDRISI-DA, the first Arabic LMD dataset containing about 2,869 posts and 3,893 LMs.
- We present IDRISI-DE, the largest *manually-labeled* public English LMD dataset of about 5,461 posts and 9,685 LMs.
- We manually label and analyze the usefulness of different features, including hashtags, event context, and URLs, replies, named entities, and other LMs, to draw helpful insights for developing effective LMD systems.
- We present a modified hierarchical LMD evaluation for classification and ranking methods.
- We provide simple yet effective English and Arabic LMD baselines.

The remainder of this paper is organized as follows. We present the related work in Section 2. We then define the LMD task in Section 3. We introduce IDRISI-D datasets and analyze them in Sections 4 and 5, respectively. We then benchmark the datasets in Section 6. We next discuss the dataset use cases in Section 7. We finally conclude in Section 9.

2 Related Work

In this section, we discuss the LMD related studies and discuss their technical solutions (Section 2.1) and evaluation (Section 2.2).

2.1 Technical Solutions

There are a few studies that tackle the LMD task using machine learning and deep learning techniques. For instance, Geoparspy (Middleton et al., 2018) is a Support Vector Machine (SVM) model trained on gazetteer-based features including location type, population, and alternative names. Additionally, the disambiguation models of the toponym resolution system employed by Wang and Hu (2019a) are essentially machine learning models including (i) *DM_NLP* (Wang et al., 2019) which is a Light Gradient Boosting Machine (LightGBM) model trained on similarity scores, contextual representations, gazetteer attributes, and mention list features, (ii) *UniMelb* (Li et al., 2019) which is an SVM that uses different feature types such as the history results in the training dataset, population, gazetteer attributes, similarity, and mention neighbors features, and (iii) *UArizona* (Yadav et al., 2019) which is a heuristic-based system that favors toponyms with higher populations.

Furthermore, Xu et al. (2019) proposed an attention-based two-pairs of bi-LSTMs for matching LMs against Foursquare gazetteer. Each location profile (*lp*) in Foursquare is represented by concatenating one-hot vector for the category attribute, TF-IDF vectors for textual attributes (e.g., address attribute), and the numeric-based attributes. On the other hand, the LM is represented using its context (i.e., post) and encoded using contextual representation attended to the *lp* vector, besides the geographical distance. The two-pair networks learn the left and right contexts of the LM. Both representations then go through a fully connected layer to learn disambiguation.

2.2 Evaluation

There is a dearth of microblogging disaster-specific LMD datasets. Table 1 presents the only two LMD datasets and their statistics. GeoCorpora (Wallgrün et al., 2018) is the only available one for the research community. Wang and Hu (2019a) evaluated it using eight different datasets available through EUPEG framework (Wang and Hu, 2019b), solely one of which is a microblogging dataset that is GeoCorpora. Xu et al. (2019) used Singapore dataset (Ji et al., 2016) for evaluation.

As for the evaluation measures, the distance-based measures have been used in non-disaster-specific studies to evaluate LMD systems. For that, the distance between the GPS coordinates of the gold and predicted LMs is measured using the great circle distance. The systems' overall performance is then computed by the Median and Mean Error

Dataset	# Twt	# LM (unique)	Labeling	LM types	Public
Singapore (Ji et al., 2016)	3,611	1,542 (-)	In-house	-	×
GeoCorpora (Wallgrün et al., 2018)	6,648	3,100 (1,119)	Crowd	×	✓
IDRISI-DE	5,591	9,586 (1,601)	In-house	✓	✓
IDRISI-DA	2,869	3,893 (763)	In-house	✓	✓

Table 1: The existing LMD datasets compared to IDRISI datasets.

Distance.

Additionally, the discrete measures including Accuracy (Acc), Precision (P), Recall (R), and the F_β score are computed to evaluate systems (Zhang and Gelernter, 2014; Li et al., 2014; Ji et al., 2016; Middleton et al., 2018; Wang and Hu, 2019a; Xu et al., 2019), however, they provide a bird's-eye view of systems' performance neglecting the nuance in their techniques. To overcome this shortcoming, Karimzadeh (2016) proposed using Cross Entropy (CE) that considers the probabilities of systems rather than their ranks, Root Mean Square Error (RMSE) that quantifies the average great circle distance between predicted and gold toponyms, and Eccentricity that combines both CE and RMSE.

Acc, P, R, and F_β can also be computed within a distance d that is commonly set to 161 KM (100 miles). For example, Acc@d is the fraction of correctly predicted LMs within d.

While these measures evaluate binary classification tasks, the LMD is typically modeled as a multi-class classification task making them inappropriate for evaluation.

3 Problem Definition

The LMD System, as illustrated in Figure 1, is given the following inputs:

- A post (a microblog) p that is related to a disaster event e,
- A set of location mentions (LMs): $L_p = \{l_i; i \in [1, n_p]\}$ in post p, where l_i is the i^{th} location mention and n_p is the total number of location mentions in p, if any.
- A geo-positioning database G (i.e., gazetteer) that consists of a set of toponyms: $T = \{t_j; i \in [1, k]\}$, where t_j is the j^{th} toponym, and k is the number of toponyms in G.

The LMD system aims to match every location mention l_i in the post p to one of the toponyms t_j in G that accurately represents l_i, if exists. Otherwise, the system must abstain and declare that l_i is unresolvable (or unlinkable).

4 Dataset Construction

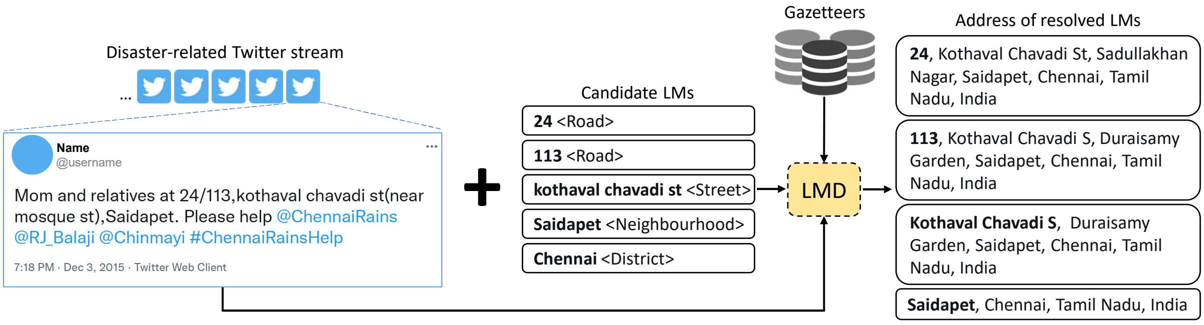
In this section, we discuss the constructing process of IDRISI-D datasets. We start by describing IDRISI-R datasets. We then present the sampling strategy and the annotation process.

IDRISI-R Datasets: We extend IDRISI-R Location Mention Recognition (LMR) English (IDRISI-RE) (Suwaileh et al., 2023a) and Arabic (IDRISI-RA) (Suwaileh et al., 2023b) datasets that are originally sampled from HumAID (Alam et al., 2021) and Kawarith (Alharbi and Lee, 2021) datasets, respectively. We select these datasets due to their unique characteristics as described below.

IDRISI-RE is the largest to date LMR microblogging English dataset. It exhibits unique diversity (domain and location types), coverage (temporal and geographical), and generalizability (domain and geographical), compared to all existing datasets of its kind. It comprises around 20k human-labeled (gold) and 57k machine-labeled (silver) posts from 19 disaster events of diverse types covering wide geographical areas. The events capture the critical periods of disaster events. The annotations include spans of location mentions in the textual content alongside their location types (e.g., country, city, street). Empirically, IDRISI-RE is the best domain and geographical generalizable dataset against all existing English datasets.

IDRISI-RA is the first Arabic LMR microblogging dataset. It contains 22 disaster events of different types that happened in Arab countries, covering various dialects reasonably. It contains 4.6K manually-annotated (gold) posts sampled from 7 disaster events,[2] and 1.2M automatically-annotated (silver) posts sampled from the entire dataset. Both versions are labeled for location mentions and location types. Empirically, the LMR models trained on IDRISI-RA showed decent generalizability to unseen events and acceptable domain and geographical generalizability.

[2] These events are labeled for informativeness in Kawarith dataset.


Disaster-related Twitter stream
...
Name
@username
...
7:18 PM · Dec 3, 2015 · Twitter Web Client
+
Candidate LMs
24 <Road>
113 <Road>
kothaval chavadi st <Street>
Saidapet <Neighbourhood>
Chennai <District>
Gazetteers
LMD
Address of resolved LMs
24, Kothaval Chavadi St, Sadullakhan Nagar, Saidapet, Chennai, Tamil Nadu, India
113, Kothaval Chavadi S, Duraisamy Garden, Saidapet, Chennai, Tamil Nadu, India
Kothaval Chavadi S, Duraisamy Garden, Saidapet, Chennai, Tamil Nadu, India
Saidapet, Chennai, Tamil Nadu, India

Mom and relatives at 24/113,kothaval chavadi st(near mosque st),Saidapet. Please help @ChennaiRains @RJ_Balaji @Chinmayi #ChennaiRainsHelp

Figure 1: High-level overview of the Location Mention Disambiguation (LMD) task.

Dataset Sampling: Constrained by not overwhelming the *volunteered* annotators, we sampled a set of posts from every disaster event in IDRISI-RE while maintaining the distribution of LM types, but covering all fine-grained LMs including neighborhoods, streets, and POIs. In total, we sampled 8,224 posts containing 11,023 LMs. On the other hand, the IDRISI-RA gold version was labeled entirely, including seven events containing 2,974 having LMs (the remaining 1,618 posts do not contain LMs) and 5,236 LMs.

Dataset Annotation: The LMD annotation removes the ambiguity of geo/geo entities (in contrast to the geo/non-geo LMR annotations). We collected the LMD annotations in 3 phases to increase the reliability of annotations with the minimum burden on the expert annotators:

P1. Two in-house annotators are assigned for every event with the condition of having a good familiarity with the country of the disaster event. When one of the annotators declares a low confidence for a specific LM or both disagree, the LM is forwarded to a meta annotator in *Phase 2*.

P2. A meta annotator resolves the disagreement from *Phase 1* and labels the low-confident examples. She has a solid understanding of the LMD task; hence, she verifies the doubtful annotations. When she fails to disambiguate an LM, it goes to experts in *Phase 3*.

P3. Expert annotators disambiguate the hard unresolved LMs from *Phases 1* and *2*. Experts are residents of the countries where the disaster events took place.

In all phases, annotators attentively read the post online alongside replies and the linked web pages. Next, they (1) disambiguate the LMs by searching OpenStreetMap (OSM) gazetteer through Nominatim search engine[3] to find the best matching toponym, (2) assign a confidence score between 1-3 for their annotation, and (3) judge the usefulness of features for disambiguation ("Yes", "No", or "None"). The features we investigate their usefulness include:

- Event: The disaster event name.
- Hashtags: The set of posts having the same hashtag as the target post within their text.
- Replies: The thread or responses to the post.
- Other LMs: Other location mentions appearing within the same post text.
- URLs: The linked web pages or media within the post text.
- Entities: Named entities that appear within the post text.

We define the usefulness as *whether a feature helps the annotator to accurately find the correct toponym from the OSM that best matches the candidate LM being annotated*.

Additionally, to avoid propagating human errors from IDRISI-R, we asked the annotators to modify LMs, add new LMs, or drop LMs in certain cases. In Table 2, we show example posts and elaborate on them in the following:

Modifying LMs: Several cases require modification, such as separating multiple LMs (Posts #1 and #6), fixing LM boundaries (Posts #2 and #7), and fixing LM type (changing "Street" to "City" in Post #3 and "City" to "POI" in Post #8), to list a few. Annotators modified 15 and 154 LMs in both IDRISI-DA and IDRISI-DE, respectively. IDRISI-RA is cleaner than IDRISI-RE as it was labeled in-house.

Dropping LMs: Annotators dropped LMs when they violate the LMR annotations guidelines. Cases include organization or person entities (Posts #4

[3] https://nominatim.openstreetmap.org/

and #9), ambiguous LMs (Posts #5), nationalities, and locational descriptions, among others. In total, we dropped 212 and 1,986 mentions, 97 and 435 of which are unique, from IDRISI-DA and IDRISI-DE datasets, respectively.

Adding new LMs: Annotators added unlabeled LMs if they are resolvable. For example, the "Pontagea Health Centre" in Post #10. This resulted in adding 27 new LMs to IDRISI-DE while no LMs are added to IDRISI-DA.

Adding LMs to OSM: Annotators added 171 and 27 new toponyms to OSM for IDRISI-DA and IDRISI-DE, respectively.

We ran the annotation task for ten weeks and obtained the final IDRISI-DE and IDRISI-DA datasets. Table 1 presents their statistics.

5 Dataset Analysis

IDRISI-D datasets inherit the geographical, domain, location types, temporal, informativeness, and dialectical (for Arabic) coverage from IDRISI-R datasets. In this section, we analyze the reliability of annotations and the usefulness of post features for the LMD task.

Reliability: To evaluate the reliability of annotations in Phase 1, we compute the Inter-Annotator Agreement (IAA) using Cohen's Kappa (Cohen, 1960). We measure the IAA for the ability to resolve LMs, i.e., whether an LM is resolvable or not. We also compute the agreement percentage on the extracted toponyms from gazetteers by the annotators for all LMs. The annotators in Phase 1 achieved substantial and almost perfect Cohen's Kappa scores of approximately 0.90 and 0.83 for IDRISI-DA and IDRISI-DE datasets, respectively. The raw agreement percentages are around 97.98% and 93.50% for IDRISI-DA and IDRISI-DE datasets, respectively. These results statistically demonstrate the high quality and reliability of annotations of IDRISI-D datasets. To further increase the quality of the datasets, we resolved the disagreement cases in the subsequent annotation phases 2 and 3.

Usefulness of Features: Table 4 shows the percentages of features' presence in posts and the percentages of useful features. We show the statistics for: (i) "ALL": all types of LMs in the datasets, (ii) "Coarse": the coarse-grained LMs including countries, cities, states, counties, districts, and neighborhoods, and (iii) "Fine": the fine-grained LMs including streets, natural POIs, human-made POIs. Apparently, the "event", "other LMs", and "hashtags" are the most useful features for LMD, especially for fine-grained LMs.

Looking carefully at the annotations of features' usefulness, we make different observations through examples in Table 3:

Event: Knowing the event place helps in narrowing the search space over OSM. Consequently, annotators can mitigate the "Toponymic homonymy" challenge (Suwaileh et al., 2022). In Post #1, all results for "شارع كورنيش النيل" ("Corniche El Nile Street") in Post #1 are not within "Cairo" where the "Cairo BMB 2019" event took place. Thus, searching toponyms within the affected area results in accurate annotations.

Other LMs: The geo-vicinity between co-occurring LMs usually represents inclusion and containment relationships, making the coarse-grained LMs useful to disambiguate the fine-grained LMs. For instance, in Post #2, "بيروت" ("Beirut" is a city) which is also a hashtag is helpful for accurately disambiguating "مستشفى القديس جاورجيوس" ("Saint George Hospital" is a human Point-of-Interest). Similarly, in Post #4, "Nebraska" (State) was useful to distinguish "Elkhorn River" (Human Point-of-Interest) from another part of the river located in "West Virginia" (State). Different reasons cause the low usefulness percentages of "other LMs". To elaborate, in cases where the same LM appears multiple times in the same post, the duplicates are useless for disambiguating each other.

Hashtags: As most hashtags indicate the disaster event (e.g., "#انفجار_معهد_الأورام" and ‘#انفجار_المرفأ" in posts #1 and #2), they are equally important to the "Event" feature.

Replies: Typically, a small number of posts get the community attention. Hence, replies are rarely useful for LMD.

URLs: Linked web pages are useful if they elaborate on the geographical context of the reported information in the post. For example, the linked web page in Post #2 was useful for locating the hospital. Also, "Lake Butler" in Post #4 is challenging LM. The linked Facebook page contains "Lake Butler, FL, United States" and "Keystones Heights" that helped the annotator to successfully resolve this LM by their geo-proximity. The importance percentage of URLs is low as many URLs are already broken or require a paid subscription.

#	Change	Post text
1	Separate LMs	...حملة نكرم موتانا لإعادة دفن القبور التي كشفتها سيول الأمطار في **مقبرتي الصليبيخات و صبحان**
2	Modify offsets	...طقس غير مستقر ... على السواحل الشمالية الغربية **ووسط سيناء وجنوب سيناء وشمال الصعيد**
3	Modify type	الطرق التالية مغلقة بسبب الغبار الكثيف وتدني مدى الرؤية: * ... باتجاه **الزرقاء** $_{Street\rightarrow City}$
4	Drop ORG	إصابة ٤٠ موظفاً في **جمعية كيفان** بفيروس كورونا - تم إغلاق السوق المركزي وجميع الأفرع ...
5	Drop undefined	... ولازالت ٢٨ حالة مصابة تحت العلاج بـ **مستشفى العزل** بمعبر رفح
6	Separate LMs	Please join us for Hurricane Maria relief this Saturday on Melrose St btwn **Buchwick** & **Broadway** ...
7	Modify offsets	The University of **Nebraska** **Omaha** Love Your Melon Crew sure knows how to make kids happy ... #MealsThatHeal
8	Modify type	Amidst applause, Canadas rescue team arrives in **Mexico City Airport**$_{City\rightarrow POI}$ on Saturday #earthquake #CASDDA via [user_mention]
9	Drop ORG	**Rosen Hotels & Resorts** in Orlando announces availability of 30 guestrooms at [user_mention] for #HurricaneIrma evacuees...
10	Add LM	Pontagea Health Centre in Beira, #Mozambique, was partially destroyed by #CycloneIdai, ...

Table 2: Examples of issues and corrections in LMD annotations. **Bold** text is the annotated LMs in IDRISI-R. Underlined text is the corrected LMs in IDRISI-D.

#	Useful features	Post text
1	Event, Other LMs, Hashtag.	أغلقت الإدارة العامة للمرور شارع كورنيش النيل (خلف **جاردن سيتي**) على خلفية اندلاع ... حريق بـ **معهد الأورام** في **#المنيل #انفجار_معهد_الأورام**
2	Other LMs, Hashtag & URL	... **#انفجار_المرفأ** يتسبب في دمار كبير بـ مستشفى القديس جاورجيوس في **#بيروت** **https://t.co/7SdALOhviW**
3	None	... فيديو يوضح وجود مفرقعات ... داخل أحد المستودعات قبل حصول إنفجار في #بيروت
4	Other LMs	Human remains discovered along Elkhorn River after flooding, sheriff says https://buff.ly/2CEShla #**Nebraska**
5	URL	In the wake of Hurricane Irma, we've planned a food distribution event in Lake Butler to help anyone affected by... **fb.me/2fbe0b4YE**
6	None	Labatt to help those affected by Fort McMurray wildfire [...] #FortMcMurray #LCBO

Table 3: Example posts showing the usefulness of different features for the LMD annotation. Underlined and **bold** text indicate the LMs and features, respectively.

It is worth noting here that the coarse-grained LMs are usually easy to disambiguate without exploiting any features (e.g., posts #3 and #6).

6 Benchmarking Experiments

In this section, we discuss the experimental setup and results of benchmarking IDRISI-D.

	Loc type	Event	Hashtags	URLs	Replies	Other LMs	Entities
IDRISI-DE							
Exist?	All	100.0%	63.9%	37.0%	0.4%	67.3%	31.2%
	Fine	100.0%	64.0%	34.3%	2.7%	65.5%	31.9%
	Coarse	100.0%	63.9%	37.2%	0.3%	67.7%	31.2%
Useful?	All	98.4%	32.7%	3.9%	5.0%	38.3%	5.6%
	Fine	94.0%	54.7%	28.2%	0.0%	66.9%	12.3%
	Coarse	98.8%	30.9%	2.1%	32.1%	36.0%	5.1%
IDRISI-DA							
Exist?	All	100.0%	56.6%	41.9%	27.7%	42.7%	34.8%
	Fine	100.0%	77.5%	53.5%	59.8%	74.6%	63.8%
	Coarse	100.0%	50.6%	38.4%	17.8%	32.7%	25.8%
Useful?	All	63.2%	22.2%	2.6%	0.9%	23.1%	2.0%
	Fine	89.8%	21.2%	3.6%	0.6%	19.8%	1.0%
	Coarse	54.4%	22.4%	2.0%	1.2%	24.8%	2.5%

Table 4: Statistics of the LMD features in IDRISI-D dataset.

6.1 Evaluation Setup

This section presents the learning models and the evaluation strategy we used to benchmark our IDRISI-D datasets.

6.1.1 Learning models

We train our own BERT-based models. We further employ retrieval- and heuristic-based off-the-shelf LMD baselines.

BERT$_{LMD}$: We fine-tuned the BERT-large-cased (Devlin et al., 2019) and MarBERT (Abdul-Mageed et al., 2021) models in sequence classification mode for English and Arabic LMD, respectively. To augment negative examples, we issue every gold LM against OSM and pick the top toponym that does not match it. We add only one negative example to balance the training data.

Nominatim (Nomin): A search engine to search OSM data by name and address. We note that none of the existing studies compare their approaches against gazetteer search APIs (Nominatim, 2023).

Geolocator2 (Geol2): CMU-geolocator is an off-the-shelf LMP system that considers the hierarchy of location mentions in posts when resolving them (Zhang and Gelernter, 2014).

Geolocator3 (Geol3): An improved version of CMU-geolocator that uses the population to post-filter retrieved results from Nominatim (Zhang and Gelernter, 2014).

GeoparsePy (GeoPy): A trained SVM model on gazetteer-based features including location type, population, and alternative names (Middleton et al., 2018).

It is worth mentioning that Geol and GeoPy employ Nomin and apply post-filters on top of it. Additionally, when benchmarking IDRISI-DA, we exclude GeoPy as it is incapable of processing Arabic text. We also note that we could not employ the disaster-specific LMD models, except GeoPy, as they are nonpublic. Re-implementation is not handy due to the lack of several technical details and the unavailability of their evaluation datasets (Ji et al., 2016; Xu et al., 2019).

6.1.2 Evaluation Measures and Strategy

Inspired by the evaluation of user geolocation task (Mourad et al., 2019), we leniently evaluate LMD systems using hierarchical evaluation; however, we adopt three major changes. First, we use exhaustive locational levels including country, state, county, city, district, neighborhood, street, and POI. Second, we propagate errors from higher to lower levels. Third, we compute ranking evaluation measures, i.e., $MRR@r$ not classification or distance-based measures. In this work, we set $r = 1$,[4] but we can use different values when perceiving the task as ranking.

6.2 Results and Discussion

In this section, we benchmark IDRISI-D using off-the-shelf LMD models and our own BERT$_{LMD}$

[4] The $MRR@1$ is equivalent to the accuracy measure for classification since for every LM, we have only one correct toponym.

model. Table 5 shows the $MRR@1$ results over IDRISI-D datasets.

System	CRY	STA	CON	CTY	STR	POI
IDRISI-DA						
GEOL2	**0.45**	0.08	0.00	0.03	0.00	0.01
GEOL3	0.44	0.07	0.00	0.02	0.00	0.01
NOMIN	0.43	0.22	0.03	0.17	0.13	0.11
BERT_{LMD}	**0.45**	**0.49**	**0.10**	**0.34**	**0.42**	**0.28**
IDRISI-DE						
GEOL2	**0.85**	0.60	0.32	0.24	0.02	0.02
GEOL3	0.83	0.61	0.31	0.24	0.02	0.02
GEOPY	0.64	0.32	0.14	0.09	0.00	0.00
NOMIN	0.81	**0.66**	**0.38**	**0.36**	**0.24**	**0.07**
BERT_{LMD}	0.73	0.61	0.29	0.28	0.14	**0.07**

Table 5: The results for the LMD models on IDRISI-DE and IDRISI-DA datasets. “CRY,” “STA,” “CON,” “CTY,” “STR,” and “POI” refer to COUNTRY, STATE, COUNTY, CITY, STREET, and POINT-OF-INTEREST evaluation levels, respectively

Arabic LMD: The GEOL systems show high performance at COUNTRY level. However, their performance is comparable to the BERT_{LMD} model. GEOL systems fail at the fine-grained evaluation levels as they employ the GeoNames gazetteer that does not support Arabic for fine-grained locations. The NOMIN baseline is showing the best results among baselines, but it fails to outperform the BERT_{LMD} at all evaluation levels.

English LMD: It is evident that the post-filters that are employed by GEOL and GEOPY are not effective for all evaluation levels, except for the COUNTRY level making the raw results from NOMIN more accurate. GEOL systems show the best results for the COUNTRY level, but their performance decreases against the BERT_{LMD} model at finer evaluation levels including STATE, CITY, STREET and POI. NOMIN is the top model at almost all evaluation levels. The BERT_{LMD} model managed to compete with NOMIN at only the POI evaluation level, which counts for the BERT_{LMD} as the fine-grained LMs are harder to disambiguate and they are of interest to the response authorities in the disaster domain (Kropczynski et al., 2018). The results also confirm that disambiguating fine-grained LMs is more challenging than coarse-grained LMs.

7 Research Use Cases

Releasing IDRISI-D enables research on *disaster-specific* and *generic* geolocation applications that we discuss below:

Event/incident detection: While LMs indicate *where* events and incidents took place (Hu and Wang, 2021), IDRISI-D datasets with their resolved LMs could serve event/incident detect models that exploit geospatial features.

Relevance filtering: While LMs increase the likelihood of microblogs being relevant and informative with regard to the disaster events (De Albuquerque et al., 2015), IDRISI-D can enable research on relevance filtering approaches that utilize geospatial information.

Geolocation applications: While the LMP tasks play a key role in tackling all of the geolocation tasks (e.g., predicting post location (Ozdikis et al., 2019), inferring user location (Luo et al., 2020), modeling user movement (Wu et al., 2022), etc.) that employ textual features (Zheng et al., 2018), IDRISI-D is an invaluable resource for tackling all these tasks.

Geographical retrieval: The geographical information retrieval (GIR) systems are concerned with extracting spatial information alongside the relevant multimodal data to the user information need. IDRISI-D could empower the GIR retrieval techniques that rely on applying LMP tasks over queries and documents (García-Cumbreras et al., 2009).

8 Challenges

Compared to gazetteers, posts over social media contain informal language, misspellings, grammar mistakes, shortened words, and slang, causing the so-called mismatch challenge (Han et al., 2013). Table 6 presents different types of issues in the following with examples in Table 6:

Nicknames: Some places have common nicknames used by locals. For example, in Post #1, “مستشفى القديس جاورجيوس” is named “مستشفى الروم”. Also, *Chennai* is nicknamed “The Detroit of India" in Post #2. The nicknames often do not exist in the gazetteers.

Abbreviations: Short names of places are prevalent on Twitter due to the character limit of posts. For example, “المملكة” (Kingdom) in Post #3 is abbreviation of المملكة العربية السعودية (Kingdom of Saudi Arabia). Also, “T. Nagar" and “GM Chetty Road" are abbreviations of “Theagaraya Nagar" and “Gopathi Narayanaswami Chetty", respectively, in Post #4.

Misspellings: Misspellings and grammar mistakes are common over Twitter. For instance,

T#	Challenge	Post text
1	Nicknames	الوضع كارثي في مستشفى الروم وهناك ضحايا في المستشفى #لبنان _ينهار #بيروت
2		#ChennaiFloods sad to see the state of city. Detroit of India is suffering. Hv personal experienced.
3	Abbreviations	نظراً للأعداد المتزايدة بالإصابة بفيروس كورونا في المملكة ... فقد أصدرت وزارة ... الداخلية عقوبات على كل من يخالف أوامر الحظر
4		Anyone around T. Nagar, needing shelter or food, can approach the Gurudwara on GM Chetty Road #Chennai
5	Misspelling	... امطار حفرالباطن غريق بحي النهضه #حفرالباطن _الان
6		Medical students of shri ramchandra medical college in chennai stranded without supplies. Need help.
7	Shortcuts	... إغلاق ط. صلاح سالم عند نفق العروبة في الاتجاهين وعند ك. الفنجري اتجاه
8		sm 1 help providing water 50 children @Lawrence Charitable Trust.safe.2/4,1st cross st,3rd avenue,AshokNagar-LakshmanSruti #ChennaiFloods

Table 6: Example posts illustrating the challenges of processing user-generated content for the LMD task. LMs with issues are underlined in text.

“بحي النهضه” and “حفرالباطن” in Post #5 should be written as “بحي النهضة” (with ة taa marbuta letter) and “حفر الباطن” (with space), respectively. Also, “**shri** ramchandra medical college" in Post #6 should be written as “**sri** ramchandra medical college".

Shortcuts: Users tend to use shortened words due to the character limit of posts. For example, “ط.” and “ك.” in Post #7 refer to “طريق.” (road in English) and “كوبري.” (bridge in English), respectively. Also, using “st” instead of “road”, in Post #8. Also, using “@” symbol instead of the literal “at” prepositions in the same post.

Capitalization: Users tend to ignore capitalization when writing posts (e.g., “chennai" instead of “Chennai" in Post #6.

Dialectics and varieties: “كوبري.” (bridge in English) in Post #7 is the dialectical (e.g., Egyptian) form of جسر in Modern Standard Arabic (MSA).

9 Conclusion

This paper contributes towards a crucial task, i.e., *Location Mention Disambiguation* in the crisis management domain. We introduced IDRISI-D, the first Arabic and the largest to date English LMD datasets.The LMD annotations that are of high reliability indicating the usefulness of the dataset. A key characteristic of IDRISI-D is the annotations of features' usefulness that we anticipate to guide the development of LMD tools. Our benchmarking results show the competitiveness of simple exact matching (NOMINATIM) and the promising performance of contextual features (BERT$_{LMD}$) for learning LMD. We release the datasets and the evaluation script for the research community. The future directions are two-fold: (i) enhancing the representation of LMs and toponyms for robust LMD learning, and (ii) employing advanced learning algorithms.

Limitations

There are a few shortcomings that we discuss below:

Twitter API Accessibility: Recently, X platform have re-envisioned its business model imposing more restrictions on the API accessibility for the research community. Although X data is extremely useful for disaster management, we expect less attention from the academic researchers to develop LMD systems that are specific for X platform. Nevertheless, IDRISI-D is invaluable resource for developing LMD systems that process user-generated content, specifically the data from microblogging platforms.

Underrepresented fine-grained LMs: Although we had chosen a careful sampling method, akin to the existing LMD datasets, the fine-grained LMs are yet underrepresented which forms a major limitation in IDRISI-D.

Temporary locations: Temporary facilities (i.e., medical camps, shelters, etc.) are constructed during emergencies to provide resources and support for the affected people. The names of these locations could change during emergencies. For example, allocating a specific school as a shelter and giving it a new expressive name (e.g., “main shelter”). Once the disaster event is over, the school will return to providing its original services. The difficulty of these temporary locations lies in their need for context when resolved. Although they are important for the affected people and response authorities, not all of them are labeled in IDRISI-D.

Ethics Statement

Although the X platform allows users to disable the geo-tagging features to protect their privacy, “even well-informed and rational individuals cannot appropriately self-manage their privacy” (Solove, 2012). There are situations where extracting geolocation data can be justified for the greater good such as during natural disasters when the focus is on saving lives and providing essential support. Therefore, any resources and tools must preserve the users’ privacy and safety, especially during critical situations that could risk people’s lives (e.g., conflicts and wars). Consequently, we have de-identified the data to protect users’ privacy.[5] We further release the data for research purposes only under the Creative Commons Attribution 4.0 International License. Above all, we affirm that systems developed using IDRISI-D datasets must implement appropriate mechanisms to safeguard user privacy.

Acknowledgements

This work was made possible by the Graduate Sponsorship Research Award (GSRA) #GSRA5-1-0527-18082 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

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)*. Association for Computational Linguistics, Online, 7088–7105. https://doi.org/10.18653/v1/2021.acl-long.551

Firoj Alam, Umair Qazi, Muhammad Imran, and Ferda Ofli. 2021. HumAID: Human-Annotated Disaster Incidents Data from Twitter. In *15th International Conference on Web and Social Media (ICWSM)*. AAAI Press, Palo Alto, California, USA, 933–942.

Alaa Alharbi and Mark Lee. 2021. Kawarith: an Arabic Twitter Corpus for Crisis Events. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*. Association for Computational Linguistics, Kyiv, Ukraine (Virtual), 42–52. https://aclanthology.org/2021.wanlp-1.5

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement* 20, 1 (1960), 37–46.

Joao Porto De Albuquerque, Benjamin Herfort, Alexander Brenning, and Alexander Zipf. 2015. A geographic approach for combining social media and authoritative data towards identifying useful information for disaster management. *International journal of geographical information science* 29, 4 (2015), 667–689.

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*, Vol. 1. Association for Computational Linguistics (ACL), Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[5] NewTab.html

Miguel Á García-Cumbreras, José M Perea-Ortega, Manuel García-Vega, and L Alfonso Ureña-López. 2009. Information retrieval with geographical references. Relevant documents filtering vs. query expansion. *Information Processing & Management* 45, 5 (2009), 605–614.

Bo Han, Paul Cook, and Timothy Baldwin. 2013. Lexical Normalization for Social Media Text. *ACM Transactions on Intelligent Systems and Technology* 4, 1 (Feb. 2013), 1–27. https://doi.org/10.1145/2414425.2414430

Yingjie Hu and Jimin Wang. 2021. How Do People Describe Locations During a Natural Disaster: An Analysis of Tweets from Hurricane Harvey. In *11th International Conference on Geographic Information Science (GIScience 2021) - Part I (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 177)*, Krzysztof Janowicz and Judith A. Verstegen (Eds.). Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 6:1–6:16. https://doi.org/10.4230/LIPIcs.GIScience.2021.I.6

Zongcheng Ji, Aixin Sun, Gao Cong, and Jialong Han. 2016. Joint Recognition and Linking of Fine-Grained Locations from Tweets. In *Proceedings of the 25th International Conference on World Wide Web* (Montréal, Québec, Canada) *(WWW '16)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 1271–1281. https://doi.org/10.1145/2872427.2883067

Morteza Karimzadeh. 2016. Performance Evaluation Measures for Toponym Resolution. In *Proceedings of the 10th Workshop on Geographic Information Retrieval* (Burlingame, California) *(GIR '16)*. Association for Computing Machinery, New York, NY, USA, Article 8, 2 pages. https://doi.org/10.1145/3003464.3003472

Jessica Kropczynski, Rob Grace, Julien Coche, Shane Halse, Eric Obeysekare, Aurelie Montarnal, Frederick Benaben, and Andrea Tapia. 2018. Identifying Actionable Information on Social Media for Emergency Dispatch. In *ISCRAM Asia Pacific 2018: Innovating for Resilience – 1st International Conference on Information Systems for Crisis Response and Management Asia Pacific*. ISCRAM Digital Library, Wellington, New Zealand, p.428–438. https://hal-mines-albi.archives-ouvertes.fr/hal-01987793

Guoliang Li, Jun Hu, Jianhua Feng, and Kian-lee Tan. 2014. Effective location identification from microblogs. In *2014 IEEE 30th International Conference on Data Engineering*. Institute of Electrical and Electronics Engineers (IEEE), Chicago, IL, USA, 880–891. https://doi.org/10.1109/ICDE.2014.6816708

Haonan Li, Minghan Wang, Timothy Baldwin, Martin Tomko, and Maria Vasardani. 2019. UniMelb at SemEval-2019 Task 12: Multi-model combination for toponym resolution. In *Proceedings of the 13th International Workshop on Semantic Evaluation*. Association for Computational Linguistics, Minneapolis, Minnesota, USA, 1313–1318. https://doi.org/10.18653/v1/S19-2231

Xiangyang Luo, Yaqiong Qiao, Chenliang Li, Jiangtao Ma, and Yimin Liu. 2020. An overview of microblog user geolocation methods. *Information Processing & Management* 57, 6 (2020), 102375.

Stuart E. Middleton, Giorgos Kordopatis-Zilos, Symeon Papadopoulos, and Yiannis Kompatsiaris. 2018. Location Extraction from Social Media: Geoparsing, Location Disambiguation, and Geotagging. *ACM Transactions on Information Systems* 36, 4 (June 2018), 1–27.

Ahmed Mourad, Falk Scholer, Walid Magdy, and Mark Sanderson. 2019. A practical guide for the effective evaluation of twitter user geolocation. *ACM Transactions on Social Computing* 2, 3 (2019), 1–23.

Nominatim. 2023. Nominatim Documentation. https://nominatim.org/release-docs/develop/

Ozer Ozdikis, Heri Ramampiaro, and Kjetil Nørvåg. 2019. Locality-adapted kernel densities of term co-occurrences for location prediction of tweets. *Information Processing & Management* 56, 4 (2019), 1280–1299.

Daniel J Solove. 2012. Introduction: Privacy self-management and the consent dilemma. *Harv. L. Rev.* 126 (2012), 1880.

Reem Suwaileh, Tamer Elsayed, and Muhammad Imran. 2022. *Role of Geolocation Prediction in Disaster Management*. Springer Nature Singapore, Singapore, 1–33. https://doi.org/10.1007/978-981-16-8800-3_176-1

Reem Suwaileh, Tamer Elsayed, and Muhammad Imran. 2023a. IDRISI-RE: A generalizable dataset with benchmarks for location mention recognition on disaster tweets. *Information Processing & Management* 60, 3 (2023), 103340. https://doi.org/10.1016/j.ipm.2023.103340

Reem Suwaileh, Muhammad Imran, and Tamer Elsayed. 2023b. IDRISI-RA: The First Arabic Location Mention Recognition Dataset of Disaster Tweets. In *Proceedings of the 61th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Toronto, Canada, 16298–16317. https://doi.org/10.18653/v1/2023.acl-long.901

Jan Oliver Wallgrün, Morteza Karimzadeh, Alan M. MacEachren, and Scott Pezanowski. 2018. GeoCorpora: building a corpus to test and train microblog geoparsers. *International Journal of Geographical Information Science* 32, 1 (2018), 1–29.

Jimin Wang and Yingjie Hu. 2019a. Are We There yet? Evaluating State-of-the-Art Neural Network Based

Geoparsers Using EUPEG as a Benchmarking Platform. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Geospatial Humanities* (Chicago, Illinois) *(GeoHumanities '19)*. Association for Computing Machinery, New York, NY, USA, Article 2, 6 pages. https://doi.org/10.1145/3356991.3365470

Jimin Wang and Yingjie Hu. 2019b. Enhancing spatial and textual analysis with EUPEG: An extensible and unified platform for evaluating geoparsers. *Transactions in GIS* 23, 6 (2019), 1393–1419. https://doi.org/10.1111/tgis.12579

Xiaobin Wang, Chunping Ma, Huafei Zheng, Chu Liu, Pengjun Xie, Linlin Li, and Luo Si. 2019. DM_NLP at SemEval-2018 Task 12: A Pipeline System for Toponym Resolution. In *Proceedings of the 13th International Workshop on Semantic Evaluation*. Association for Computational Linguistics, Minneapolis, Minnesota, USA, 917–923. https://doi.org/10.18653/v1/S19-2156

Junhang Wu, Ruimin Hu, Dengshi Li, Lingfei Ren, Wenyi Hu, and Yilin Xiao. 2022. Where have you been: Dual spatiotemporal-aware user mobility modeling for missing check-in POI identification. *Information Processing & Management* 59, 5 (2022), 103030.

Canwen Xu, Jiaxin Pei, Jing Li, Chenliang Li, Xiangyang Luo, and Donghong Ji. 2019. DLocRL: A Deep Learning Pipeline for Fine-Grained Location Recognition and Linking in Tweets. In *The World Wide Web Conference, WWW 2019*. ACM, San Francisco, CA, USA, 3391–3397. https://doi.org/10.1145/3308558.3313491

Vikas Yadav, Egoitz Laparra, Ti-Tai Wang, Mihai Surdeanu, and Steven Bethard. 2019. University of Arizona at SemEval-2019 Task 12: Deep-Affix Named Entity Recognition of Geolocation Entities. In *Proceedings of the 13th International Workshop on Semantic Evaluation*. Association for Computational Linguistics, Minneapolis, Minnesota, USA, 1319–1323. https://doi.org/10.18653/v1/S19-2232

Wei Zhang and Judith Gelernter. 2014. Geocoding location expressions in Twitter messages: A preference learning method. *Journal of Spatial Information Science* 2014, 9 (2014), 37–70.

Xin Zheng, Jialong Han, and Aixin Sun. 2018. A Survey of Location Prediction on Twitter. *IEEE Transactions on Knowledge and Data Engineering* 30, 9 (sep 2018), 1652–1671. https://doi.org/10.1109/TKDE.2018.2807840 arXiv:1705.03172v2

CamelParser2.0: A State-of-the-Art Dependency Parser for Arabic

Ahmed Elshabrawy,†‡ **Muhammed AbuOdeh,**† **Go Inoue,**†‡ **Nizar Habash**†
†Computational Approaches to Modeling Language (CAMeL) Lab,
New York University Abu Dhabi (NYUAD)
‡Mohamed bin Zayed University of Artificial Intelligence (MBZUAI)
{ahmed.elshabrawy,go.inoue}@mbzuai.ac.ae, {m.abuodeh,nizar.habash}@nyu.edu

Abstract

We present **CamelParser2.0**, an open-source Python-based Arabic dependency parser targeting two popular Arabic dependency formalisms, the Columbia Arabic Treebank (CATiB), and Universal Dependencies (UD). The **CamelParser2.0** pipeline handles the processing of raw text and produces tokenization, part-of-speech and rich morphological features. As part of developing **CamelParser2.0**, we explore many system design hyper-parameters, such as parsing model architecture and pretrained language model selection, achieving new state-of-the-art performance across diverse Arabic genres under gold and predicted tokenization settings.

1 Introduction

Dependency parsing is a natural language processing (NLP) task used to analyze the grammatical structure of a sentence by identifying and representing the relationships between its words. Dependency parsing assigns a directed tree structure to the sentence, with words as nodes and syntactic dependencies as edges (see Figure 1). Dependency parsing, and syntactic parsing in general, has long been considered an important NLP enabling technology and analysis tool (Jurafsky and Martin, 2009). The interest in using syntactic structures in NLP in the neural age remains, e.g., as analytical tools for studying large language models (Kulmizev, 2023), for guided data augmentation for Neural Machine Translation (Duan et al., 2023), Semantic Role Labeling (Tian et al., 2022), and Grammatical Error Correction (Li et al., 2022; Zhang et al., 2022).

There have been previous developments in Arabic dependency parsing (Habash and Roth, 2009; Marton et al., 2013; Zhang et al., 2015; Shahrour et al., 2016; Al-Ghamdi et al., 2023). However, they are not based on state-of-the-art (SOTA) developments in neural dependency parsing and pretrained language models, nor can they be easily integrated into larger project pipelines. Furthermore, they are not trained on larger and more diverse treebanks that have been developed recently. Many have only been tested with gold tokenization, not as part of a full pipeline from sentence to tree – a notable exception is the work of Zhang et al. (2015) who modeled segmentation and parsing jointly.

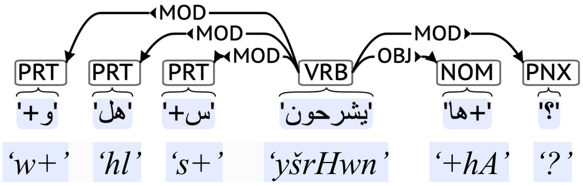


Figure 1: An example CATiB dependency tree (Habash et al., 2009) for the short question وهل سيشرحونها؟ *whl syšrHwnhA?*[2] 'and will they explain it?'.

In this work, we investigate the effect of many system design hyper-parameters including parsing model architecture, pretrained language model selection, and training data configurations to achieve unprecedented dependency parsing performance on multiple Arabic genres. Hence, we present **CamelParser2.0**, an open-source dependency parsing pipeline that achieves SOTA performance on Columbia Arabic Treebank (CATiB) and Universal Dependencies (UD) parsing of Arabic across multiple genres from Modern Standard Arabic (MSA) and Classical Arabic (CA).

Our contributions are: (1) achieving new **state-of-the-art** on both CATiB and UD formalisms in multiple Arabic genres on all metrics; (2) developing and releasing an **open-source Python-based** pipeline for Arabic parsing;[1] and (3) **benchmarking** a large number of hyper-parameters to ensure the best system design choices.

[1] https://github.com/CAMeL-Lab/camel_parser

[2] HSB Arabic transliteration (Habash et al., 2007)

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 170–180
December 7, 2023 ©2023 Association for Computational Linguistics

2 Related Work

2.1 Dependency Parsing

There are two main approaches to dependency parsing: transition-based (Yamada and Matsumoto, 2003; Nivre et al., 2006) and graph-based (McDonald et al., 2005). Both approaches have recently been implemented with neural models to improve performance. For example, Dozat and Manning (2016) develop a graph-based parser that uses a biaffine attention mechanism on a neural model to achieve SOTA/near SOTA results on six different languages including Czech, a morphologically rich language with flexible word order. On the other hand, Mohammadshahi and Henderson (2019) develop a transformer mechanism that conditions on graphs to be used with a neural transition-based parser to achieve SOTA results on 13 languages. The evaluations that guide the development of these architectures are mainly carried out on higher resource languages, such as English and other European languages.

In this work, we investigate how neural dependency parsing performs on Arabic given its relatively fewer resources, especially in certain classical genres, such as pre-Islamic texts.

2.2 Arabic Treebanks

The primary treebank for Arabic syntactic analysis is the Penn Arabic Treebank (PATB) (Maamouri et al., 2004), which uses a phrase structure grammar. It has been converted to a dependency representation that uses two different formalisms: CATiB (henceforth, PATB-CATiB) (Habash and Roth, 2009), and UD (NUDAR Treebank) (Taji et al., 2017). The two formalisms are compared in some detail by Taji et al. (2017).

The first dependency treebank developed for Arabic is the Prague Arabic Dependency Treebank (PADT) (Smrž et al., 2002). PADT is in part based on PATB; and it was later extended to UD (henceforth, PADT-UD).[3] Since then, several treebanks have been developed such as the Columbia Arabic Treebank (CATiB) (Habash and Roth, 2009), Quran Corpus (Dukes and Buckwalter, 2010), *i3rab* treebank (Halabi et al., 2021), and Arabic Poetry Treebank (ArPoT) (Al-Ghamdi et al., 2021). Most recently, Habash et al. (2022) released the Camel Treebank (CamelTB), which is a multi-genre Arabic dependency treebank in the CATiB formalism spanning CA texts from the 6th Century to MSA texts from the 21st century. PADT (Smrž et al., 2002), CATiB (Habash et al., 2009), and UD (Nivre et al., 2017) are dependency tree representations with different POS tags, dependency relation labels, and attachment rules.

In this work, we make use of recent developments in Arabic treebanking to explore the performance of different parsing model architectures, with different training dataset configurations, and with different dependency formalisms (CATiB and UD) on multiple Arabic genres, and under gold and predicted tokenization conditions.

2.3 Arabic Parsing

Regarding evaluating parser design in Arabic dependency parsing, the work done by Marton et al. (2013) examines the impact of morphological features on dependency parsing performance under both gold and predicted conditions. They observe differences in feature importance when using predicted features due to changes in prediction accuracy for each examined feature. They find that definiteness, person, number, gender, and undiacritized lemma are most helpful under predicted conditions. Their results are observed using MaltParser, a transition-based model, with a feature-based SVM classifier (Nivre et al., 2006), which differs from the recent neural SOTA models that learn features from the training data implicitly.

Kankanampati et al. (2020) leverage the Easy-First LSTM-based architecture proposed by Kiperwasser and Goldberg (2016), but experiment with sharing tree representations and BiLSTM layers between CATiB and UD formalisms to achieve significant error reduction on both.

More recently, Al-Ghamdi et al. (2023) employ an approach that treats dependency parsing as a sequence labeling task (Strzyz et al., 2019). They apply various pretrained BERT models under different fine-tuning and architectural setups. They explore the performance of this approach on (a) PADT (Smrž et al., 2002), (b) part 2 of PATB in the CATiB formalism (Maamouri et al., 2004; Habash and Roth, 2009), and (c) ArPoT (Al-Ghamdi et al., 2021).

To our knowledge, the current state-of-the-art in terms of *publicly available* dependency parsing systems in Arabic is the **CamelParser1.0** for the CATiB formalism (Shahrour et al., 2016), and **UDPipe 2** for the UD formalism (Straka, 2018).

[3] https://github.com/UniversalDependencies/UD_Arabic-PADT/

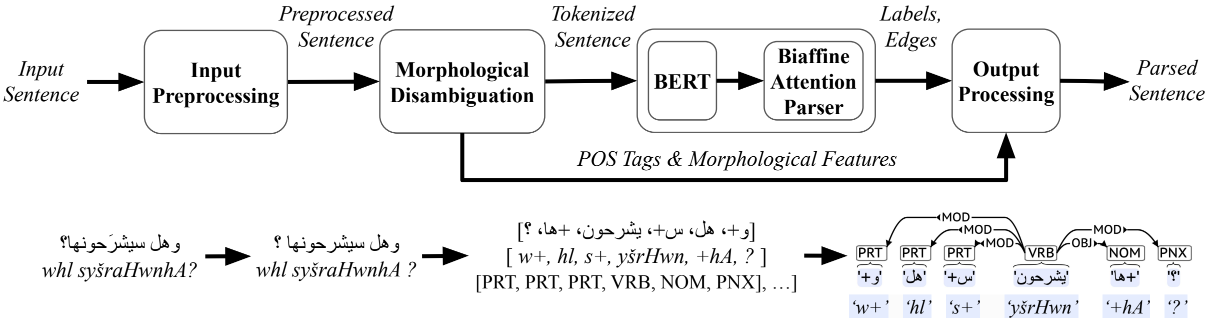


Figure 2: A diagram of the **CamelParser2.0** pipeline paired with a simple example of raw text input.

Our work is closest in high-level design to **CamelParser1.0**, which uses the MADAMIRA morphological disambiguation system based on SVM classifiers and morphological analyzers (Pasha et al., 2014) and an SVM-based parsing system called MaltParser (Nivre et al., 2006). It reported results on the Penn Arabic Treebank (Maamouri et al., 2004), which is limited to the newswire genre. Since these results were reported, significant advancements have been made in both dependency parsing and morphological analysis through the use of pretrained language models like BERT and neural model architectures (Dozat and Manning, 2016; Inoue et al., 2022); and more datasets in Arabic genres beyond newswire have been created (Habash et al., 2022). We use CAMeL Tools (Obeid et al., 2020) as part of the implementation of **CamelParser2.0**.

By utilizing the aforementioned developments in Arabic treebanking and neural dependency parsing, we experiment on PATB-CATiB, CamelTB, NUDAR, and PADT to improve the dependency parsing performance in Arabic in multiple MSA and CA genres and across the CATiB and UD formalisms. Due to a different experimental setup and data scope explored by Al-Ghamdi et al. (2023), we cannot directly compare our results on all metrics and datasets; however, we observe that our approach outperforms their reported results on the test set of PADT. Additionally, by comparing our findings to the existing SOTA pipelines, **CamelParser1.0** and **UDPipe 2**, as well as the reported results in Kankanampati et al. (2020), we observe that **CamelParser2.0** sets the new SOTA in Arabic dependency parsing for both gold and predicted tokenization settings.

3 The CamelParser2.0 Pipeline

In this section, we present the details of the **CamelParser2.0** pipeline (Figure 2). The pipeline accommodates varying levels of pre-processing in the input. Depending on the extent to which the input has been pre-processed, the pipeline conducts morphological disambiguation. Once input tokens have been identified, they are passed to the dependency parsing system which outputs dependency arcs and labels. The dependency relations are then combined with the form and additional part-of-speech tags and morphological features, which are either specified in the input or generated in the morphological disambiguation step, to output a CoNLL-X/CoNLL-U file format (Buchholz and Marsi, 2006; De Marneffe et al., 2014).

3.1 Input Formats

Before parsing begins, the input to the pipeline is directed to the proper step based on its format. Currently, we support the following input formats.

Raw Text Raw Arabic text is first cleaned by normalizing Unicode characters, removing diacritics and other characters that are not Arabic, ASCII, or Latin-1, and performing whitespace tokenization (Obeid et al., 2020). The text is then passed to the Morphological Disambiguation step (Figure 2).

Pre-Tokenized and Tagged Text Files containing token and *optional* Part-of-Speech (POS) tag tuples are supported. The input is passed to the parser directly. Since the parser does not require POS tags, they will not be produced if only tokenized text is provided.

CoNLL-X/CoNLL-U The pipeline also accepts input in the CoNLL-X/CoNLL-U tab-separated file format (Buchholz and Marsi, 2006; De Marneffe et al., 2014).

3.2 Tokenization and POS tagging

When the input is already tokenized, we pass that information onto the dependency parsing system. As for raw untokenized text, we make use of a **Morphological Disambiguation** system which predicts the tokens and the POS tags of these tokens (see Figure 2). The user determines whether to use a more accurate but more resource-intensive BERT unfactored disambiguator (Inoue et al., 2022) or a lighter Maximum Likelihood Estimation (MLE) disambiguator, both of which are included in CAMeL Tools (Obeid et al., 2020). We then extract the tokens, lemmas, and primary POS tags (CATiB or UD), as well as a set of morphological features provided by CAMeL Tools: MADA POS, position-marked proclitics and enclitics (`prc3`, `prc2`, `prc1`, `prc0`, `enc0`), person, gender, number, aspect, voice, mood, state, case, and rationality. We add a feature `token_type` to signify if the token is a baseword or clitic (indicated by its location, e.g., `prc2`).

3.3 Dependency Parsing

The next component of our parsing pipeline is the dependency parsing model, which expects tokenized Arabic data as input. We use the SuPar Biaffine Dependency Parser (Zhang, 2021), which is based on the work of Dozat and Manning (2016) with a key difference. Instead of using a GLoVe vector-based encoding layer, we generate word embeddings using a BERT model. To achieve this, a BERT model is used to generate WordPiece-level embeddings by summing up the last four layers of the BERT model (Devlin et al., 2018). Then, to generate the token-level embeddings, the corresponding WordPieces' embeddings of each token are pooled using a mean.

The output of this step is the dependency relations and labels of the input text. The POS and morphological features are integrated in the final dependency representation in an output postprocessing step (see Figure 2).

In this paper, for comparison purposes, we also report on using the MaltParser system introduced by Nivre et al. (2006) which is employed by the previous SOTA parsing system for Arabic, **CamelParser1.0** (Shahrour et al., 2016).

4 Experimental Setup

Our experimental setup involves training multiple dependency parsing models with different training data configurations which are then evaluated on multi-genre development and test sets under both gold and predicted tokenization settings to gauge accuracy and robustness across multiple genres in Arabic. The details of the various experimental setups are outlined below.

4.1 Data

The data we use to train and evaluate includes PATB-CATiB and CamelTB (CATiB representation), and PADT-UD and NUDAR (UD representation). Table 1 lists the corpora and their sub-corpora and indicates their genres, variety (MSA or CA), and sizes. We note that PADT-UD text data contains a subset of PATB. CamelTB has a variety of different sub-corpora across multiple genres, some of which are similar to PATB (WikiNews and QALB). The PATB (PATB-CATiB and NUDAR) was split according to the recommendations by Diab et al. (2013). We follow the recommendations of the creators of PADT for its data splits.[4] We split the CamelTB data according to the recommendations by Habash et al. (2022) in CamelTB v1.1.[5]

In our experiments, we examine a number of training data combinations to provide the best robustness and accuracy across multiple Arabic genres. We do not train on individual CamelTB genres because of the limited amount of data we have; but we report results for them. Similar to Kankanampati et al. (2020), we exclude all non-projective trees in the training, but not in the dev and test.

4.2 Metrics

Dependency Parsing Accuracy Evaluation of dependency parsing models is done primarily through three metrics:

- **Labeled Attachment Score (LAS)**: The percentage of tokens with correct head/parent and correct label/relation to that parent.
- **Unlabeled Attachment Score (UAS)**: The percentage of tokens with correct head/parent.
- **Label Score (LS)**: The percentage of tokens with correct label/relation.

LAS is the primary metric we report on.

[4] https://github.com/UniversalDependencies/UD_Arabic-PADT/

[5] http://treebank.camel-lab.com/

Rep	Corpus		Text Source	Var	Cent	Genre	Sents	Words	Tokens
CATiB	**PATB-CATiB**		Penn Arabic Treebank (Parts 1-2-3)	MSA	21st	News	**19,738**	**628,598**	**738,889**
	CamelTB	Odes	Suspended Odes (Mu'allaqat)	CA	6th	Poetry	784	7,465	10,170
		Quran	Quranic Surahs	CA	7th	Quranic	572	11,699	15,791
		Hadith	Hadiths from Sahih Bukhari	CA	7th	Prophetic Sayings	1,190	12,467	15,745
		1001	One Thousand and One Arabian Nights	CA	12th	Stories	1,145	11,831	17,109
		Hayy	Hayy ibn Yaqdhan (Ibn Tufail)	CA	12th	Philosophical Novel	1,198	19,674	26,583
		OT	Old Testament	MSA	19th	Bible Translation	535	9,097	11,788
		NT	New Testament	MSA	19th	Bible Translation	573	9,593	12,293
		Sara	Sara (Al-Akkad)	MSA	20th	Novel	1,585	35,356	46,375
		ALC	Arabic Learner Corpus	MSA	21st	Student Essays (L2)	727	9,221	12,047
		BTEC	Basic Traveling Expressions Corpus	MSA	21st	Phrasebook	2,000	15,935	18,602
		QALB	QALB Corpus	MSA	21st	Online Commentary	923	11,454	14,139
		WikiNews	WikiNews	MSA	21st	News	996	18,314	21,481
		ZAEBUC	Zayed Bilingual Undergraduate Corpus	MSA	21st	Student Essays (L1)	1,109	15,778	19,787
				CamelTB Total			**13,337**	**187,884**	**241,910**
				PATB-CATiB+CamelTB Total			**33,075**	**816,482**	**980,799**
UD	**PADT-UD**		Prague Arabic Dependency Treebank	MSA	21st	News	**7,664**	**17,357**	**113,500**
	NUDAR		NYUAD UD Arabic Treebank	MSA	21st	News	**19,738**	**628,598**	**738,889**

Table 1: The various datasets we experiment with in developing **CamelParser2.0**. **Rep** (Representation) specifies the treebank formalism. **Var** is the Arabic variant. **Cent** is the century. **Sents** is the number of sentences.

Statistical Significance In certain cases, we test for statistical significance using a one-tailed Welch's t-test following the recommendations of Dror et al. (2018). We treat each sentence as an independent experiment and calculate a sentence-level accuracy of parsing which we use to conduct the statistical significance testing.

4.3 Tokenization

Previous work on dependency parsing tends to judge performance purely on gold tokenization (Marton et al., 2013; Shahrour et al., 2016; Dozat and Manning, 2016; Mohammadshahi and Henderson, 2019), although there are many recent exceptions (Shao et al., 2018; More et al., 2019; Habash et al., 2022). We report on both gold and predicted tokenization to study the performance under real-world conditions. We use the BERT unfactored disambiguator (Inoue et al., 2022) in CAMeL Tools (Obeid et al., 2020). On our dev datasets (PATB and CamelTB sub-corpora), the average predicted word-level tokenization accuracy is 96.8%, with a wide range from WikiNews (99.8%) to Odes (91.3%), with PATB at 99.1%. This range of performance is consistent with our expectations since the CAMeL Tools MSA disambiguator is trained on PATB train data (news genre).

4.4 Parsing Models

We compare our **CamelParser2.0** neural dependency parsing architecture, as described in section 3.3 with other pre-existing parsing system baselines. The first baseline, **MaltParser** (v1.9.2) (Nivre et al., 2007), forms the core of the previous SOTA for dependency parsing in Arabic, **CamelParser1.0** (Shahrour et al., 2016). We compare to it directly and as part of **CamelParser1.0** (second baseline). The third baseline is UDPipe 2, whose models are currently available from the LINDAT UDPipe REST Service.[6] The last baseline is the system of Kankanampati et al. (2020); we report their published numbers where appropriate.

It is important to note that the experimentation Kankanampati et al. (2020) report on is mainly to leverage parallel data in two formalisms (CATiB and UD) and not necessarily to achieve an overall SOTA parser for Arabic. Nevertheless, they achieve impressive results so we compare against their best reported numbers. We do not leverage their multitask learning approach for **CamelParser2.0**; however, it could prove useful for future work to explore combining our approaches by sharing representations in the Biaffine parsing ar-

[6] https://ufal.mff.cuni.cz/udpipe/2

	LAS	UAS	LS
MaltParser	80.7	83.0	93.4
CamelParser1.0 (Shahrour et al., 2016)	83.8	86.4	93.2
Kankanampati et al. (2020)	86.2	88.1	-
CamelParser2.0	**91.3**	**92.4**	**97.0**

Table 2: Scores of various dependency parsing systems trained on the PATB-CATiB and evaluated on the test set of PATB-CATiB. **CamelParser2.0** achieves the SOTA on all metrics and improves on **CamelParser1.0** (Shahrour et al., 2016) by almost 7.5 points on the LAS. Kankanampati et al. (2020) do not report on the LS.

chitecture proposed by Dozat and Manning (2016) between different formalisms to further improve parsing performance across formalisms.

4.5 BERT Model Selection

We also experiment with four pretrained BERT models. The first three are from CamelBERT (Inoue et al., 2021): **CamelBERT-MSA** is pretrained on MSA data, **CamelBERT-CA** is pretrained on CA data, and **CamelBERT-MIX** is pretrained on MSA, CA, and Dialectal Arabic data. We make use of them because they give us an understanding of how pretrained data interplays with parsing performance on differing genres and variants. Additionally, they were created under the same settings, hence, they reduce experimental variation. Furthermore, we make use of **AraBERT v2.0** (Antoun et al., 2020) as it improves upon AraBERTv0.2 which has been shown previously to achieve SOTA performance on a range of Arabic NLP tasks (Inoue et al., 2021).

5 Results and Analysis

We present the results of the experiments we conducted as part of developing **CamelParser2.0**.

5.1 Comparing System Baselines

In Table 2, we report **CamelParser2.0**'s performance against previous SOTA baselines under the same exact training/testing conditions with gold tokenization. All systems are trained on PATB-CATiB training data and evaluated on PATB-CATiB test. It should be noted that **MaltParser** and **CamelParser1.0** use the same base algorithms and implementations; however, **CamelParser1.0** does further hyper-parameter optimization and feature selection to improve performance on Arabic as opposed to **MaltParser** which just uses the default configuration. We also include the best results reported by Kankanampati et al. (2020), however, we cannot compare our results on the LS as they do not report them. For **CamelParser2.0**, we use our baseline BERT model (CamelBERT-MSA). We observe that across all metrics, **CamelParser2.0** achieves significant improvements over all the reported systems including a 46.3%, 44.1%, and 55.9% error reduction on the LAS, UAS, and LS respectively when compared to the previous SOTA pipeline **CamelParser1.0**. Therefore, we only move forward with testing **CamelParser2.0** for the rest of our experiments.

5.2 Comparing Training Data Configurations

We compare different training datasets and their combination. We use the same **CamelParser2.0** model with CamelBERT-MSA, and report on both gold and predicted tokenization to determine which training data configuration yields the best results on LAS. As seen in Table 3, in the first three columns under the Gold/Predicted Tokenization and CamelBERT headers, the overall trend is that using training data from both PATB-CATiB and CamelTB to train the parser yields the best results on all averages in both the gold and predicted tokenization cases. This is unsurprising given the larger training data size and inclusion of multiple genres. There are some instances where using a smaller training configuration is better than using the larger combined configuration (e.g., Hadith, Hayy, NT); however, they are not statistically significant. On average there are larger gains on both accuracy and robustness to be had from using more training data.

5.3 Comparing BERT Embedding Models

We then experiment with different BERT models as embedding layers (Table 3). Unsurprisingly, the best-performing models on the MSA and CA multi-genre data were CamelBERT-MSA and CamelBERT-CA, not CamelBERT-MIX which was trained with dialectal data.

We observe the following differences depending on the BERT model used. There was a sta-

		Gold Tokenization						Predicted Tokenization				
		CamelBERT					AraBERT	CamelBERT				AraBERT
		MSA			CA	MIX		MSA			CA	
PATB-CATiB		**X**		**X**	**X**	**X**	**X**	**X**		**X**	**X**	**X**
CamelTB			**X**	**X**	**X**	**X**	**X**		**X**	**X**	**X**	**X**
1001	CA	86.2	90.7	**91.9**	91.2	91.2	**92.8**	84.2	88.9	90.1	**90.7**	**90.8**
ALC	MSA	87.3	88.9	**89.2**	88.6	88.9	**90.1**	86.0	87.3	**87.5**	86.5	**88.7**
BTEC	MSA	82.0	86.0	**86.2**	85.2	85.0	**87.1**	81.2	**85.1**	**85.1**	84.5	**86.1**
Hadith	CA	81.2	90.4	90.2	**91.2**	90.7	**91.2**	79.6	87.9	88.2	**88.9**	**88.9**
Hayy	CA	86.6	90.4	90.2	**91.0**	89.3	**91.3**	85.6	88.9	88.7	**89.2**	**89.8**
NT	MSA	74.5	81.1	79.8	**81.2**	80.6	80.2	71.8	**78.5**	76.4	77.1	76.9
Odes	CA	72.7	76.9	77.7	**80.2***	77.1	78.7	68.6	71.7	72.5	**75.2***	74.8
OT	MSA	77.1	82.4	**82.5**	82.3	82.4	**83.4**	74.4	79.4	**79.7**	79.5	**80.5**
QALB	MSA	82.8	87.6	87.6	**88.0**	**88.0**	87.7	82.3	86.7	86.9	**87.3**	**87.3**
Quran	CA	73.8	84.1	84.4	**85.4**	84.3	**85.5**	72.8	**82.3**	83.0	83.2	**83.5**
Sara	MSA	80.2	86.3	**86.6**	86.3	85.9	**87.0**	79.0	**85.0**	84.1	82.6	83.9
WikiNews	MSA	89.0	90.3	**90.4***	86.9	89.5	90.3	88.9	90.1	**90.2***	87.9	90.1
ZAEBUC	MSA	88.2	90.0	**91.1**	89.6	90.9	**92.0**	87.6	89.5	**90.7**	89.3	**91.7**
PATB	MSA	**92.2**	85.2	92.1	90.9	91.5	**92.3**	**91.7**	85.0	91.6	90.4	**91.8**
CamelTB Average		81.7	86.5	86.5	**87.2**	86.4	**87.5**	80.2	84.7	**84.4**	85.6	**85.6**
Total Average		82.4	86.5	86.9	**87.5**	86.8	**87.8**	81.0	84.7	**85.0**	85.9	**86.1**
MSA Average		83.7	86.4	**86.9**	86.6	87.0	**87.8**	82.5	85.2	**85.3**	85.0	**86.3**
CA Average		80.1	86.5	86.9	**89.7**	86.5	87.9	78.2	83.9	84.5	**88.0**	**85.6**

Table 3: The **LAS** of different training configurations on the **Dev sets** of the CamelTB sub-corpora and PATB-CATiB. We test under both Gold and Predicted tokenization conditions, and using different BERT embedding models. The overall best-performing configuration is underlined and in bold, while the best-performing CamelBERT model is in bold. Results with an asterisk indicate statistical significance ($p < 0.05$) for results discussed in Section 5.3.

tistically significant +2.5 gain with gold tokenization and +2.7 gain with predicted tokenization on the LAS from using CamelBERT-CA instead of CamelBERT-MSA on CamelTB-Odes. Furthermore, there was a statistically significant -3.5 drop with gold tokenization and -2.3 drop with predicted tokenization from using CamelBERT-CA over CamelBERT-MSA on CamelTB-WikiNews. However, on average, there is not much of a performance difference between CamelBERT-CA and CamelBERT-MSA; in other cases, the differences were not statistically significant. Nevertheless, it seems that using CamelBERT-CA yields improvements on the parser's performance on CA texts, despite being pretrained on fewer data, and CamelBERT-MSA yields improvements on the performance on MSA texts. These results are consistent with the observations of Inoue et al. (2021) and support the importance of careful selection of the BERT embedding model depending on the data being parsed.

Finally, we also compare with AraBERT, which outperforms CamelBERT on macro average across almost all sub-corpora. AraBERT is better or equal to CamelBERT in 10 out of 14 cases in both Gold and Predicted conditions; however, none of the improvements are statistically significant when compared genre-by-genre.

5.4 CATiB Test Set Results

We report the performance of our best performing models on CATiB formalism from Table 3 on the unseen test sets in Table 4. We observe similar patterns in the results as discussed before. Hence, we make similar recommendations for model selection given the data.

		Gold Tokenization			Predicted Tokenization		
		CamelBERT		AraBERT	CamelBERT		AraBERT
		MSA	CA		MSA	CA	
PATB-CATiB		X	X	X	X	X	X
CamelTB		X	X	X	X	X	X
1001	CA	91.9	**92.0**	**92.2**	89.6	**89.7**	**89.9**
ALC	MSA	**87.5**	86.9	**87.6**	**86.0**	85.7	**86.5**
BTEC	MSA	**84.9**	84.3	**85.5**	**83.6**	83.0	**84.0**
Hadith	CA	92.4	**93.9**	92.2	90.5	**91.8**	90.9
Hayy	CA	91.7	**91.8**	**92.6**	90.3	**90.5**	**91.1**
NT	MSA	**84.7**	84.2	84.6	**79.1**	78.7	78.8
Odes	CA	77.3	**81.5**	78.8	75.3	**77.0**	75.5
OT	MSA	**87.4**	87.3	**87.8**	**82.4**	80.8	**82.6**
QALB	MSA	86.5	**86.7**	**86.8**	**86.1**	85.1	85.9
Quran	CA	82.7	**83.6**	**83.9**	80.3	**80.8**	**81.0**
Sara	MSA	**84.5**	83.9	84.2	**78.9**	78.1	**79.3**
WikiNews	MSA	**90.1**	88.9	**90.1**	**90.0**	88.7	**90.0**
ZAEBUC	MSA	**91.8**	91.4	**92.5**	**90.6**	90.4	**91.3**
PATB	MSA	**91.3**	89.8	**91.3**	**89.6**	**89.6**	**91.0**
CamelTB Average		87.2	**87.4**	**87.6**	**84.8**	84.6	**85.1**
Total Average		87.5	**87.6**	**87.9**	**85.2**	85.0	**85.6**
MSA Average		**87.6**	87.0	**87.8**	**85.1**	84.5	**85.5**
CA Average		87.2	**88.6**	87.9	85.2	**86.0**	85.7

Table 4: The **LAS** of different training configurations on the **Test sets** of the CamelTB sub-corpora and PATB-CATiB. Only the best-performing models from the evaluation on the dev sets are included. The overall best-performing configuration is underlined and in bold, while the best-performing CamelBERT model is in bold.

5.5 Parsing UD with CamelParser2.0

The focus of the previous experiments has been on the performance on the CATiB formalism; however, we also examine the system's performance on UD data. We do so by training our dependency parsing model on PADT-UD and NUDAR, and evaluate on the respective dev and test sets. Due to differing annotation styles between these two UD corpora, cross-evaluation results in poor performance. Hence, we do not report those results here.

We only include **CamelParser2.0** with AraBERT and CamelBERT-MSA because these datasets consist of only MSA, and those models performed the best on MSA data based on our experimentation with CATiB dependency parsing. Furthermore, we use the same disambiguation system to generate the predicted tokens for two reasons: UDPipe 2's disambiguation system was not able to segment the sentences properly so we were unable to align the output for evaluation and secondly we get to observe the performance of the systems while controlling for tokenization accuracy. Results are in Table 5. Furthermore, we only include the best-reported results by Kankanampati et al. (2020) on Gold Tokenization because that is the only experimental setup they report on. We observe that we indeed achieve the SOTA on UD datasets when we compare against UDPipe 2 and Kankanampati et al. (2020). We also observe that CamelBERT-MSA performs better on these datasets than AraBERT.

		Gold Tokenization		Predicted Tokenization	
System	**Train**	**Dev**	**Test**	**Dev**	**Test**
UDPipe 2	PADT-UD	82.5	82.7	81.6	80.9
CamelParser2.0+CamelBERT	PADT-UD	**83.2**	**83.9**	**82.5**	**82.4**
CamelParser2.0+AraBERT	PADT-UD	82.7	83.4	82.2	82.0
Kankanampati et al. (2020)	NUDAR	85.2	84.8	-	-
CamelParser2.0+CamelBERT	NUDAR	**89.1**	**88.9**	**88.7**	**88.8**
CamelParser2.0+AraBERT	NUDAR	89.0	**88.9**	88.1	88.4

Table 5: The **LAS** of different systems evaluated on datasets that use the UD formalism using both gold and predicted tokenization. The first three systems are trained on the PADT and the last three systems are trained on the PATB in the UD formalism (NUDAR). Evaluation is done on the respective **Dev and Test** sets of each corpus.

6 Conclusion and Future Work

We presented **CamelParser2.0**, a new SOTA open-source, Python-based Arabic dependency parser that supports UD and CATiB formalisms and multiple Arabic genres. We make **CamelParser2.0** publicly available.[7] In the future, we plan to continue to enhance the **CamelParser2.0** models and integrate them in downstream applications to support Arabic NLP. We also plan to extend the parser to cover multiple Arabic dialects.

Acknowledgements

We acknowledge the support of the High Performance Computing Center at New York University Abu Dhabi.

Limitations

We recognize that the current parser has limitations, as it is primarily tailored to the most commonly used dependency representation formalisms. However, it does not accommodate other formalisms, such as those rooted in Arabic's extensive traditional syntactic literature (Dukes and Buckwalter, 2010; Halabi et al., 2021). The primary challenge here revolves around the availability of resources. Additionally, we acknowledge that the parser's current focus is on Modern Standard Arabic (MSA) and Classical Arabic (CA), and there is a notable absence of research in the field of Dialectal Arabic parsing (Chiang et al., 2006). It's worth noting that there are numerous pretrained language models available for experimentation. Regrettably, due to limited computational resources, we are unable to explore this avenue. Lastly, we acknowledge that we do not report on extrinsic metrics or performance in downstream tasks.

[7] https://github.com/CAMeL-Lab/camel_parser

References

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).

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.

Sabine Buchholz and Erwin Marsi. 2006. Conll-x shared task on multilingual dependency parsing. In *Proceedings of the Conference on Computational Natural Language Learning (CoNLL)*, pages 149–164, New York City, New York.

David Chiang, Mona Diab, Nizar Habash, Owen Rambow, and Safiullah Shareef. 2006. Parsing Arabic Dialects. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, Trento, Italy.

Marie-Catherine De Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D Manning. 2014. Universal stanford dependencies: A cross-linguistic typology. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, volume 14, pages 4585–92, Reykjavik, Iceland.

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

Mona Diab, Nizar Habash, Owen Rambow, and Ryan Roth. 2013. LDC Arabic treebanks and associated corpora: Data divisions manual. *arXiv preprint arXiv:1309.5652*.

Timothy Dozat and Christopher D. Manning. 2016.

Deep biaffine attention for neural dependency parsing. *CoRR*, abs/1611.01734.

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.

Sufeng Duan, Hai Zhao, and Dongdong Zhang. 2023. Syntax-aware data augmentation for neural machine translation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:2988–2999.

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.

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 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, 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.

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.

Dan Jurafsky and James H. Martin. 2009. *Dependency Parsing*. Pearson Prentice Hall.

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).

Eliyahu Kiperwasser and Yoav Goldberg. 2016. Easy-first dependency parsing with hierarchical tree LSTMs. *Transactions of the Association for Computational Linguistics*, 4:445–461.

A Kulmizev. 2023. *The Search for Syntax: Investigating the Syntactic Knowledge of Neural Language Models Through the Lens of Dependency Parsing*. Ph.D. thesis, Uppsala University.

Zuchao Li, Kevin Parnow, and Hai Zhao. 2022. Incorporating rich syntax information in grammatical error correction. *Information Processing & Management*, 59(3):102891.

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.

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.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 523–530, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Alireza Mohammadshahi and James Henderson. 2019. Graph-to-graph transformer for transition-based dependency parsing. *CoRR*, abs/1911.03561.

Amir More, Amit Seker, Victoria Basmova, and Reut Tsarfaty. 2019. Joint transition-based models for morpho-syntactic parsing: Parsing strategies for MRLs and a case study from Modern Hebrew. *Transactions of the Association for Computational Linguistics*, 7:33–48.

Joakim Nivre, Željko Agić, Lars Ahrenberg, Maria Jesus Aranzabe, Masayuki Asahara, Aitziber Atutxa, Miguel Ballesteros, John Bauer, Kepa Bengoetxea, Riyaz Ahmad Bhat, Eckhard Bick, Cristina Bosco, Gosse Bouma, Sam Bowman, Marie Candito, Gülşen Cebiroğlu Eryiğit, Giuseppe G. A. Celano, Fabricio Chalub, Jinho Choi, Çağrı Çöltekin, Miriam Connor, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Arantza Diaz de Ilarraza, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Marhaba Eli, Tomaž Erjavec, Richárd Farkas, Jennifer Foster, Cláudia Freitas, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh

Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta Gonzáles Saavedra, Matias Grioni, Normunds Grūzītis, Bruno Guillaume, Nizar Habash, Jan Hajič, Linh Hà Mỹ, Dag Haug, Barbora Hladká, Petter Hohle, Radu Ion, Elena Irimia, Anders Johannsen, Fredrik Jørgensen, Hüner Kaşıkara, Hiroshi Kanayama, Jenna Kanerva, Natalia Kotsyba, Simon Krek, Veronika Laippala, Phuong Lê Hồng, Alessandro Lenci, Nikola Ljubešić, Olga Lyashevskaya, Teresa Lynn, Aibek Makazhanov, Christopher Manning, Cătălina Mărănduc, David Mareček, Héctor Martínez Alonso, André Martins, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Anna Missilä, Verginica Mititelu, Yusuke Miyao, Simonetta Montemagni, Amir More, Shunsuke Mori, Bohdan Moskalevskyi, Kadri Muischnek, Nina Mustafina, Kaili Müürisep, Luong Nguyễn Thị, Huyền Nguyễn Thị Minh, Vitaly Nikolaev, Hanna Nurmi, Stina Ojala, Petya Osenova, Lilja Øvrelid, Elena Pascual, Marco Passarotti, Cenel-Augusto Perez, Guy Perrier, Slav Petrov, Jussi Piitulainen, Barbara Plank, Martin Popel, Lauma Pretkalniņa, Prokopis Prokopidis, Tiina Puolakainen, Sampo Pyysalo, Alexandre Rademaker, Loganathan Ramasamy, Livy Real, Laura Rituma, Rudolf Rosa, Shadi Saleh, Manuela Sanguinetti, Baiba Saulīte, Sebastian Schuster, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Lena Shakurova, Mo Shen, Dmitry Sichinava, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Aaron Smith, Alane Suhr, Umut Sulubacak, Zsolt Szántó, Dima Taji, Takaaki Tanaka, Reut Tsarfaty, Francis Tyers, Sumire Uematsu, Larraitz Uria, Gertjan van Noord, Viktor Varga, Veronika Vincze, Jonathan North Washington, Zdeněk Žabokrtský, Amir Zeldes, Daniel Zeman, and Hanzhi Zhu. 2017. Universal dependencies 2.0.

Joakim Nivre, Johan Hall, and Jens Nilsson. 2006. MaltParser: A data-driven parser-generator for dependency parsing. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, pages 2216–2219, Genoa, Italy.

Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gulsen Eryigit, Sandra Kubler, Svetoslav Marinov, and Erwin Marsi. 2007. MaltParser: A Language-independent System for Data-driven Dependency Parsing. *Natural Language Engineering*, 13(2):95–135.

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.

Arfath Pasha, Mohamed Al-Badrashiny, Mona Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan Roth. 2014. Madamira: A fast, comprehensive tool for morphological analysis and disambiguation of Arabic. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, pages 1094–1101, Reykjavik, Iceland.

Anas Shahrour, Salam Khalifa, Dima Taji, and Nizar Habash. 2016. CamelParser: A system for Arabic syntactic analysis and morphological disambiguation. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 228–232.

Yan Shao, Christian Hardmeier, and Joakim Nivre. 2018. Universal word segmentation: Implementation and interpretation. *Transactions of the Association for Computational Linguistics*, 6:421–435.

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.

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.

Michalina Strzyz, David Vilares, and Carlos Gómez-Rodríguez. 2019. Viable dependency parsing as sequence labeling. 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 717–723, Minneapolis, Minnesota. Association for Computational Linguistics.

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.

Yuanhe Tian, Han Qin, Fei Xia, and Yan Song. 2022. Syntax-driven approach for semantic role labeling. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 7129–7139, Marseille, France.

Hiroyasu Yamada and Yuji Matsumoto. 2003. Statistical dependency analysis with support vector machines. In *Proceedings of the Eighth International Conference on Parsing Technologies*, pages 195–206, Nancy, France.

Yu Zhang. 2021. SuPar GitHub repository - v1.1.4.

Yuan Zhang, Chengtao Li, Regina Barzilay, and Kareem Darwish. 2015. Randomized greedy inference for joint segmentation, POS tagging and dependency parsing. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 42–52, Denver, Colorado. Association for Computational Linguistics.

Yue Zhang, Bo Zhang, Zhenghua Li, Zuyi Bao, Chen Li, and Min Zhang. 2022. SynGEC: Syntax-enhanced grammatical error correction with a tailored GEC-oriented parser. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2518–2531, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

GARI: Graph Attention for Relative Isomorphism of Arabic Word Embeddings

Muhammad Asif Ali,[1] **Maha Alshmrani,**[1] **Jianbin Qin,**[2] **Yan Hu,**[1] **Di Wang**[1]
[1] King Abdullah University of Science and Technology, KSA
[2] Shenzhen University, China
{muhammadasif.ali; maha.shmrani; yan.hu; di.wang}@kaust.edu.sa; qinjianbin@szu.edu.cn

Abstract

Bilingual Lexical Induction (BLI) is a core challenge in NLP, it relies on the relative isomorphism of individual embedding spaces. Existing attempts aimed at controlling the relative isomorphism of different embedding spaces fail to incorporate the impact of semantically related words in the model training objective. To address this, we propose GARI that combines the distributional training objectives with multiple isomorphism losses guided by the graph attention network. GARI considers the impact of semantical variations of words in order to define the relative isomorphism of the embedding spaces. Experimental evaluation using the Arabic language data set shows that GARI outperforms the existing research by improving the average P@1 by a relative score of up to 40.95% and 76.80% for in-domain and domain mismatch settings respectively. We release the codes for GARI at https://github.com/asif6827/GARI.

1 Introduction

Bilingual Lexical Induction (BLI) is a key task in natural language processing. It aims at the automated construction of translation dictionaries from monolingual embedding spaces. BLI plays a significant role in multiple different natural language processing applications. For instance, the automated construction of lexical dictionaries plays a key role in the development of linguistic applications for low-resource languages, especially in cases where hand-crafted dictionaries are non-existent. Automated construction of high-quality dictionaries also helps in augmenting the end performance of multiple down-streaming tasks, including but not limited to: machine translation (Lample et al., 2018), information retrieval (Artetxe et al., 2018), cross-lingual transfers (Artetxe and Schwenk, 2019).

Earlier methods aimed at the construction of cross-lingual embeddings use linear and/or non-linear mapping functions in order to map the monolingual embeddings in a shared space. Some examples in this regard include retrieval criteria for bilingual mapping by Joulin et al. (2018) and BLI in non-isomorphic spaces by Patra et al. (2019).

These methods rely on the approximate isomorphism assumption, i.e., they assume that underlying monolingual embedding spaces are geometrically similar, which severely limits their use to closely related data sets originating from similar domains and/or languages exhibiting similar characteristics. The limitations of the mapping-based methods, especially their inability to handle data sets originating from different domains and languages exhibiting different characteristics has been identified by (Conneau et al., 2017; Søgaard et al., 2018; Glavas et al., 2019; Patra et al., 2019).

Some other noteworthy aspects identified in the literature that limit the end performance of the BLI systems, include: (a) algorithmic mismatch for independently trained monolingual embeddings, (b) different parameterization, (c) variable data sizes, (d) linguistic difference, etc., (Marie and Fujita, 2020; Marchisio et al., 2022).

In the recent past, there has been a shift in the training paradigm for the BLI models, i.e., instead of relying on pre-trained embeddings trained independently of each other, they use explicit isomorphism metrics along with the distributional training objective (Marchisio et al., 2022). However, a key limitation of these models is their inability to incorporate the impact of semantically related tokens (including their lexical variations) in controlling the relative isomorphism of different spaces. This is illustrated in Figure 1, where the left half of the figure shows a set of semantically related English words, e.g., {strong, rugged, and robust}. These

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 181–190
December 7, 2023 ©2023 Association for Computational Linguistics

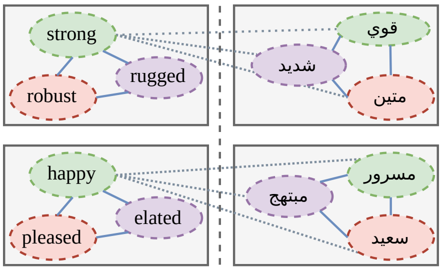


Figure 1: Some examples of semantically related tokens for English and their corresponding translations in the Arabic language.

words though lexically different share the same semantics. Correspondingly, their translations in the Arabic language: {شديد ، قوي ، متين} are also semantically related. We hypothesize that each language encompasses a list of such semantically related words that may be used interchangeably within a fixed context, and in order to control the relative isomorphism of corresponding embedding spaces the end model should be robust to incorporate these semantic variations in the model training objective.

To address these challenges, in this paper, we propose Graph Attention for Relative Isomorphism (GARI). GARI combines the distributional training objective with the isomorphism loss in a way that it incorporates the impact of semantically related words using graph attention, required to perform the end-task in a performance-enhanced way. We outline the key contributions of this work as follows:

1. We propose GARI that combines the distributional loss with graph attention-based isomorphism loss functions for effective BLI.

2. The graph attention part of the GARI leverages self-attention mechanism in order to attend over words that are semantically related to a given word.

3. We prove the effectiveness of GARI by comprehensive experimentation. Experimental evaluation shows, for the Arabic dataset, the GARI outperforms the existing research on relative isomorphism by 40.95% and 76.80% for in-domain and out-of-domain settings.

2 Related Work

There is an immense literature on BLI and controlling the relative isomorphism of the embedding spaces. In order to save space, we primarily limit the related work of this paper to one that is more relevant to our problem settings. We classify the related work into the following categories: (i) mapping pre-trained embeddings, (ii) combined training.

Mapping Pre-trained Embeddings. These methods rely on the use of linear and/or non-linear mappings to map the mono-lingual embeddings to a shared space.

Earlier works in this regard include principled bilingual dictionaries by Artetxe et al. (2016) that aim to learn bilingual mappings while preserving invariance for the monolingual analogy tasks. Artetxe et al. (2017) introduced a self-learning approach to relax the requirements for bilingual training seeds and/or parallel corpora. Alvarez-Melis and Jaakkola (2018) formulate the alignment as an optimal transport problem and employ Gromov-Wasserstein distance to compute the similarity of word pairs across different languages. Doval et al. (2018) propose additional transformation on top of the alignment step to force the synonyms towards a middle point for a better cross-lingual integration of the vector spaces. Jawanpuria et al. (2019) introduced language-specific rotations followed by a language-independent similarity in a common space. Similar to the word embedding methods, the application of the mapping-based methods to the contextualized embeddings include context-aware mapping by Aldarmaki and Diab (2019) and alignment of contextualized embeddings by Schuster et al. (2019).

Combined Training. On contrary to the mapping-based methods that rely on pre-trained embeddings, these methods use parallel data as input in order to jointly minimize the mono-lingual as well as cross-lingual training objectives. Duong et al. (2017) introduced methods for cross-lingual word embeddings for multiple languages in a unified vector space aimed to combine the strengths of different languages. Wang et al. (2019) addressed the limitations of joint training methods by combining them with mapping-based schemes for model training. For more details on the joint training methods refer to the survey paper by Ruder et al. (2019). Marchisio et al. (2022) introduced IsoVec which uses multiple different isomorphism metrics with skip-gram as the distributional training objective to control the isomorphism.

Nevertheless, we observe that existing methods for controlling the relative isomorphism ignore the impact of words that are semantically related to a given word, severely limits the ability of these methods to control the relative isomorphism of the embedding spaces.

3 Background

In this section, we first introduce the mathematical notation being used throughout the paper and formulate our problem definition. Later, we provide a quick background of the VecMap (Artetxe et al., 2018), a toolkit for mapping across different embedding spaces.

3.1 Notation

For this work, we use $\mathbf{X} \in \mathbf{R}^{m \times d}$ and $\mathbf{Y} \in \mathbf{R}^{n \times d}$ to represent the embedding matrices for the source and target languages with vocab size m and n respectively. d refers to the dimensionality of the embedding space. The embedding vectors for words, e.g., $\{x, y\}$ are represented by $\{\vec{\mathbf{x}}, \vec{\mathbf{y}}\}$. Like existing supervised works on controlling the relative isomorphism, e.g., IsoVec by Marchisio et al. (2022), we assume the availability of training seeds pairs for the source and target languages, denoted by: $\{(x_0, y_0), (x_1, y_1), ...(x_s, y_s)\}$.

3.2 The problem

In this work, we address a core challenge in BLI, i.e., we control the relative isomorphism of the embedding spaces. Specifically, we learn the distributional embeddings for the source language (i.e., Arabic) in a way:

1. The source embeddings $\mathbf{X}$ are geometrically isomorphic to the target embeddings $\mathbf{Y}$ (i.e., English language).

2. While learning isomorphic embeddings the $\mathbf{X}$ should incorporate the impact of the semantically related tokens (also their lexical variations) in $\mathbf{Y}$ in order to perform the end task in a performance-enhanced way.

3.3 VecMap toolkit

We use VecMap toolkit[1] for mapping across different embedding spaces. For this, we pre-process the embeddings using a process flow outlined by Zhang et al. (2019). The embeddings are unit-normed, mean-centered followed by another round of unit-normalization. For bi-lingual induction, we follow (Artetxe et al., 2018), i.e., whitening the spaces, and solving Procrustes. Later, we perform re-weighting, de-whitening, and mapping of translation pairs via nearest-neighbor retrieval (Artetxe et al., 2018).

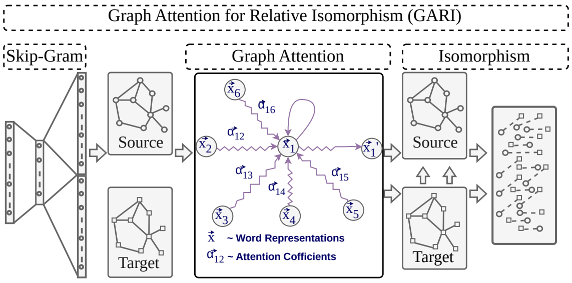


Figure 2: Graph Attention for Relative Isomorphism (GARI), the framework proposed in this work. It combines skip-gram and isomorphism loss (guided by graph attention).

4 Proposed Approach

In this paper, we address a core challenge in controlling the geometric isomorphism for source word embeddings relative to the target word embeddings, i.e., incorporate the impact of semantically coherent words in order to perform the end task in a performance augmented fashion. For this, we propose Graph Attention for Relative Isomorphism (GARI), shown in Figure 2. Details about the individual components of GARI are provided in the following subsections.

4.1 GARI

4.1.1 Overview

GARI aims to learn the source distributional embeddings $\mathbf{X}$ in a way that: (a) $\mathbf{X}$ is geometrically isomorphic to the target embeddings $\mathbf{Y}$, (b) $\mathbf{X}$ incorporates the impact of semantic variations of words in $\mathbf{Y}$. In order to control the geometric isomorphism of the embedding spaces in a robust way, GARI uses graph attention mechanism (to incorporate the impact of semantically related tokens) prior to using the isomorphism loss functions. Finally, it combines the distributional training objective and the isomorphism loss as the training objectives of the complete model.

4.1.2 Distributional Representation Learning

In order to learn the distributional embeddings for GARI, we use skip-gram with negative sampling (Mikolov et al., 2013). Its formulation is shown in Equation 1, i.e, embed a word close to its neighboring words within a fixed contextual window, while at the same time pushing it away

[1] https://github.com/artetxem/vecmap

from a list of random words selected from a noisy distribution.

$$\mathcal{L}_{Dis} = \log\sigma(\vec{\mathbf{x}}'^{\mathsf{T}}_{c_O}\vec{\mathbf{x}}_{c_I}) + \sum_{i=1}^{k}\mathbf{E}_{c_i\sim P_n(c)}\big[\log\sigma(-\vec{\mathbf{x}}'^{\mathsf{T}}_{c_i}\vec{\mathbf{x}}_{c_I})\big] \tag{1}$$

Here $\vec{\mathbf{x}}_{c_O}$ and $\vec{\mathbf{x}}_{c_I}$ correspond to the output and input vector representations of the word c. k is the number of noisy samples and $\vec{\mathbf{x}}'_{c_i}$ is the embedding vector for the noisy word selected from the noisy distribution $P_n(c)$.

4.1.3 Semantic Relatedness

To incorporate the impact of semantically related words in controlling the relative isomorphism of the embedding spaces, GARI uses graph attention mechanism. The graph attention part of GARI works as follows: (a) create a graph $\mathbf{G}$ such that semantically related words end up being neighbors in the graph, (b) use graph attention mechanism for information sharing among neighbors in $\mathbf{G}$. The details about individual components are as follows:

(a) Graph Construction. The end goal of the graph construction step is to unite and/or combine the semantically related words helpful in controlling the relative isomorphism. Inputs for the graph construction process include: (i) pre-trained word2vec embeddings[2], and (ii) seed words corresponding to the target language, i.e., $\{y_0, y_1, ..., y_s\}$. The graph construction process proceeds as follows:

(a) Organize all seed words for the target language as a set of pairs: $\mathbf{P} = \{(y_0, y_1), (y_0, y_2), ..., (y_s, y_s)\}$, i.e., combinations of two words at a time.

(b) For each pair compute the cosine similarity score between the corresponding word2vec embedding vectors, and retain only the subset ($\mathbf{P}_{sub}$) with the cosine similarity score greater than a threshold (η).

(c) Finally, for the word pairs in $\mathbf{P}_{sub}$ construct a graph $\mathbf{G}$ by formulating edges between the word pairs.

Note, this setting for the graph construction allows each word to be surrounded by a set of semantically related neighbors which provides GARI with the provision to allow the propagation of information by using graph attention, as explained below.

(b) Graph Attention. The graph attention part of GARI follows a similar approach as proposed by Veličković et al. (2017). For a graph $\mathbf{G}$, the inputs to a single attention layer of the graph attention network include the source word representations $\{\vec{x_0}, \vec{x_1}, ..., \vec{x_s}\}, \vec{x_i} \in \mathbf{R}^d$, where s represent the number of words and d represents the dimensionality of the feature. It generates a new set of word representations $\{\vec{\mathbf{x}}'_0, \vec{\mathbf{x}}'_1, ..., \vec{\mathbf{x}}'_s\}, \vec{\mathbf{x}}'_i \in \mathbf{R}^{d'}$ as output. Its process flow is explained as follows:

Initially, a linear transformation is applied to all the words in $\mathbf{G}$ parameterized by a shared matrix $\mathbf{W} \in \mathbf{R}^{d\times d'}$. This is followed by using a shared attention mechanism $z : \mathbf{R}^{d'} \times \mathbf{R}^{d'} \to \mathbf{R}$ to compute the intermediate attention coefficients β_{ij} that incorporates the importance of word j on word i.

$$\beta_{ij} = z(\mathbf{W}\vec{\mathbf{x}_i}, \mathbf{W}\vec{\mathbf{x}_j}) \tag{2}$$

where the attention mechanism z is simply a single-layered feed-forward neural network with a weight vector $\vec{\mathbf{z}} \in \mathbf{R}^{d'}$ and ReLU non-linearity, as shown below:

$$z = \text{ReLU}\Big(\vec{\mathbf{z}}^T[\mathbf{W}\vec{\mathbf{x}}_i||\mathbf{W}\vec{\mathbf{x}}_j]\Big) \tag{3}$$

where $||$ is the concatenation operator. Note, the computation for β_{ij} implies each word will have an impact on every other word in $\mathbf{G}$, which is computationally inefficient and may inject noise in the model training. In order to avoid this, we perform masked attention, i.e., compute the attention weight β_{ij} for a fixed neighborhood of word i, i.e., $j \in \mathcal{N}_i$. We use the softmax function to compute the normalized attention coefficients α_{ij}, shown as follows:

$$\alpha_{ij} = \text{softmax}(\beta_{ij}) = \frac{\exp(\beta_{ij})}{\sum_{k\in\mathcal{N}_i}\exp(\beta_{ik})} \tag{4}$$

Finally, we use the normalized coefficients in order to compute a linear combination of the corresponding word representations as the final output representation of each word as follows:

$$\vec{\mathbf{x}}'_i = \sigma\Big(\sum_{j\in\mathcal{N}_i}\alpha_{ij}\mathbf{W}\vec{\mathbf{x}_i}\Big) \tag{5}$$

where σ is a nonlinearity.

[2] https://code.google.com/archive/p/word2vec/, trained using Google-News Corpus of 100 billion words.

Though Veličković et al. (2017) extend their work to a multi-head attention setting, but for GARI, we resort to one attention layer in order to avoid the computational overhead.

The intuitive explanation for the graph attention part of GARI is to surround each word by a set of semantically related words by forming edges in the graph and re-compute the representation of each word by propagating information from the neighbors in a way that it accommodates the impact of semantic variations of each word in an attentive way.

4.1.4 Isomorphism Loss

Finally, we use the output of the graph attention layer ($\mathbf{X}'$) to compute the isomorphism loss for GARI relative to the target embeddings $\mathbf{Y}$. For this, we analyze the impact of multiple different variants of isomorphism loss functions referred to as $\mathcal{L}_{Iso}$. The details about different variants of the isomorphic loss functions are as follows:

L2 Loss ($\mathcal{L}_2$). We use L2-norm averaged over the number of words as our isomorphism metric. For N words, $\mathcal{L}_2$ is computed as:

$$\mathcal{L}_2 = \frac{1}{N}||\mathbf{X}' - \mathbf{Y}||_2 \tag{6}$$

Orthogonal Procrustus Loss ($\mathcal{L}_{proc}$). The orthogonal Procrustes problem aims to find a linear transformation $\mathbf{W}_p$ to solve the following metric:

$$\mathcal{L}_{proc} = \operatorname*{arg\,min}_{\mathbf{W}_p \in \mathbf{R}^{d \times d}, \mathbf{W}_p^T \mathbf{W}_p = I} \frac{1}{N}||\mathbf{X}'\mathbf{W}_p - \mathbf{Y}||_2 \tag{7}$$

For this, we use an existing solution $\mathbf{W}_p = \mathbf{Q}\mathbf{P}^T$ proposed by Schönemann (1966), where $\mathbf{P}\Sigma\mathbf{Q}^T$ is the singular value decomposition of the matrix $\mathbf{Y}^T\mathbf{X}'$.

A variant of Procrustus Loss ($\mathcal{L}_{proc_{src}}$). For this, we follow the same process flow as outlined above for the Procrustus loss. The only difference is that we use pre-trained embeddings for the target words to initialize the corresponding embeddings for the source words for a given set of translation seed pairs. The end goal of this setting is to analyze the contribution of the pre-trained embeddings to guide the overall isomorphism of the source embeddings. Note that the initialized embeddings for the source words are updated during the model training.

4.2 The Complete Model

Finally, we combine the loss for the skip-gram distributional training objective with the isomorphism loss in order to come up with the loss function of GARI, as shown below:

$$\mathcal{L}_{GARI} = \gamma\mathcal{L}_{Dis} + (1-\gamma)\mathcal{L}_{Iso} \tag{8}$$

where, γ is the hyper-parameter controlling the contribution of individual losses in the model.

5 Experiments and Results

5.1 Datasets

For comparative analysis, we use the same data settings as primarily used by recent work, i.e., IsoVec by Marchisio et al. (2022). For the main experiments (section 5.4), we use the first 1 million lines of the newscrawl-2020 data set for the English and Arabic languages (Barrault et al., 2020). For the domain mismatch settings (section 6.1), we use 33.8 million lines of web-crawl data for the English language and newscrawl-2020 data for the Arabic language. For data pre-processing, we use Moses scripts[3] to process the English language data. For the Arabic language, we use NLTK tokenizer[4]. For performance evaluation, we used publically available train, dev, and test splits provided by MUSE (Conneau et al., 2017). We use word pairs numbered: 0-5000, 5001-6500, and 6501-8000 as train, test, and dev splits respectively. The train split is used for model training, and dev split for parameter tuning. The final results are computed over the test split.

5.2 Baseline Models

We use independently trained distributional embeddings for the source and target languages (without the isomorphism loss) as an immediate baseline. Other than this, we compare GARI against the existing best-performing model on relative isomorphism, i.e., IsoVec by Marchisio et al. (2022). Note, IsoVec follows a similar approach as that of GARI with the distinction that GARI uses graph attention as an additional layer to control the relative isomorphism of semantically relevant words. For IsoVec, we used publicly available implementation provided by the authors to generate the results for the Arabic language.

[3] github.com/moses-smt/mosesdecoder/tree/master/scripts/tokenizer

[4] https://www.nltk.org/api/nltk.tokenize.html

Methodology	Avg. P@1
Baseline	15.58 (± 0.8)
IsoVec (L2)	19.59 (± 0.7)
IsoVec (Proc-L2)	20.03 (± 0.5)
IsoVec (Proc-L2-Init)	22.10 (± 0.5)
GARI ($\mathcal{L}_2$)	29.32 (± 0.09)
GARI ($\mathcal{L}_{proc_{src}}$)	**31.15 (± 0.07)**
GARI ($\mathcal{L}_{proc}$)	30.60 (± 0.21)

Table 1: The results for the proposed model compared against the baseline model and existing state-of-the-art work on relative isomorphism, i.e., IsoVec (Marchisio et al., 2022).

Methodology	Avg. P@1
Baseline	14.70 (± 0.7)
IsoVec (L2)	18.49 (± 0.6)
IsoVec (Proc-L2)	18.80 (± 0.7)
IsoVec (Proc-L2-Init)	19.14 (± 0.7)
GARI ($\mathcal{L}_2$)	29.69 (± 0.18)
GARI ($\mathcal{L}_{proc_{src}}$)	32.27 (± 0.17)
GARI ($\mathcal{L}_{proc}$)	**33.84 (± 0.02)**

Table 2: The results for the proposed approach under domain mismatch settings compared against the baseline model and existing state-of-the-art work on relative isomorphism, i.e., IsoVec (Marchisio et al., 2022).

5.3 Experimental Settings

In order to train the proposed model, i.e., GARI, we use Adam optimizer (Kingma and Ba, 2014) with learning rate = 0.001. In Equation 1, we set the value of k = 10. In Equation 8, we use the value of γ = 0.333. For the graph construction process, $\eta = 0.4$. We use English as the target language, and Arabic as the source language. Similar to the baseline models, we use VecMap toolkit (explained in Section 3.3) for mapping across different embedding spaces. We use average precision (i.e., P@1) as our evaluation metric, and report the mean (μ) and standard deviation (σ) of the results averaged over 5 runs of the experiment. All the experiments are performed using Intel Core-i9-10900 CPU and Nvidia 1080Ti GPUs.

5.4 Main Results

The results of GARI compared against the baseline models are shown in Table 1. We bold-face overall best scores and underline the previous state-of-the-art.

These results show that GARI outperforms the baseline models by a significant margin. The results of GARI with different isomorphism loss functions show that almost all the loss functions exhibit a similar performance with the loss ($\mathcal{L}_{proc_{src}}$) yielding overall best scores. Compared with the best performing baseline scores, GARI($\mathcal{L}_{proc_{src}}$) improves the average P@1 by approximately 40.95%. For the variants of GARI with loss functions $\mathcal{L}_2$ and $\mathcal{L}_{proc}$ the improvement in performance is 32.67% and 38.46% respectively. A relatively higher performance for the loss $\mathcal{L}_{proc_{src}}$ compared to $\mathcal{L}_{proc}$ shows that initializing the source embeddings with corresponding translation pairs from the target embeddings had a beneficial impact on the model training. Analyzing the variance of the results, we observe the variance of GARI is much lower compared to the variance of the baseline models. The worst-case variance of GARI is even less than half of the variance of the baseline models, which shows that GARI yields an overall stable performance across multiple re-runs of the experiments.

To summarize, these experiments show the essence of using the graph attention layers on controlling the relative isomorphism of the embedding spaces for BLI. We attribute the performance gained by GARI to the ability of the self-attention mechanism to appropriately accumulate information from semantically related words, which in turn plays a significant role in controlling the relative isomorphism of the embedding spaces.

6 Discussion

In this section, we perform a detailed analysis of GARI under different settings. For this, we perform analyses encompassing: (i) domain mismatch settings, (ii) correlation with isometric metrics, and (iii) error analysis.

6.1 Domain Mis-match

The results of our model for domain mismatch settings are shown in Table 2. Similar to the results for the main experiments, we also compare these results against the baseline models. We boldface the overall best scores with existing state-of-the-art underlined. These results show that GARI yields higher performance compared to the baseline models. The variants of GARI with loss $\mathcal{L}_2$, $\mathcal{L}_{proc}$ and $\mathcal{L}_{proc_{src}}$ outperform the best performing baseline model by 55.12%, 76.80%, and 68.60% respectively.

Comparing these results to the results for the main experiments (reported in Table 1), we observe that GARI yields a better performance for the domain mismatch settings relative to the in-domain setting. We attribute this performance improvement to: (a) the ability of GARI to capture and consolidate information from semantically relevant

	ES (↓)	ρ (↑)
GARI ($\mathcal{L}_2$)	80.99	0.46
GARI ($\mathcal{L}_{proc}$)	99.89	**0.56**
GARI ($\mathcal{L}_{proc_{src}}$)	**76.89**	0.45

Table 3: Analysis of different isometry metrics for GARI, i.e., , Eigenvector Similarity (ES) and Pearson's Correlation (ρ).

words even from different domains, (b) a relatively larger corpus for the target language (English) for domain mismatch settings. We notice that in contrast to the main experiments, for the domain mismatch settings loss the model GARI($\mathcal{L}_{proc}$) yields a better performance compared to GARI($\mathcal{L}_{proc_{src}}$). This shows that with the increase in the size of the data, the capability of the graph attention part of GARI to accumulate information about the semantically related words augments in a way that it even surpasses the model training with seed embeddings initialized.

Note, as illustrated in Section 1, domain mismatch is a key challenge for the BLI systems. Earlier research by Søgaard et al. (2018) shows that the majority of existing BLI systems perform poorly in inferring bilingual information from embeddings trained on different data domains. One key challenge that hinders the performance of these BLI systems is their inability to incorporate the impact of semantically related keywords and/or jargons peculiarly related to different domains. These words though belonging to different data domains have similar meanings and BLI systems should appropriately use this information for the model training. This makes GARI a better alternate, especially because of its provision to accumulate information about multiple different semantically related words using graph attention layers, as is also evident by a relatively higher performance of GARI compared to the baseline models.

6.2 Correlation with isometric metrics

Similar to the existing works on controlling the relative isomorphism of the embedding spaces (Marchisio et al., 2022), we compute isomorphism metrics for the results of GARI. We use two widely used metrics, namely: (i) Eigenvector similarity, (ii) Pearson's correlation. The computation details, and results of GARI for these metrics are as follows:

Eigenvector Similarity (ES). In order to compute the eigenvector similarity between the embedding spaces, we compute the Laplacian spectra of corresponding k-nearest neighbour graphs. We expect the graphs with similar structures to have similar eigenvalue spectra. For this, we follow the same settings as that of Søgaard et al. (2018). Given the seed pairs $\{x_0, x_1, ..., x_s\}$ and $\{y_0, y_1, ..., y_s\}$, we proceed as follows: (i) compute unweighted k-nearest neighbour graphs (i.e., G_X and G_Y), (ii) compute the graph Laplacians L_{G_X} and L_{G_Y}, where $L_\mathrm{G} = D_\mathrm{G} - A_\mathrm{G}$, (iii) compute the eigenvalues for each graph Laplacian, i.e., $\{\lambda_{L_{\mathrm{G}_X}}(i); \lambda_{L_{\mathrm{G}_Y}}(i)\}$ (iv) select $r = min(r_X, r_Y)$ where r_X is the maximum r such that the first r eigenvalues of L_{G_X} sum to less than 90% of the total sum of the eigenvalues. (v) depending upon the value of r, compute the eigenvector similarity as: $\sum_{i=1}^{r}(\lambda_{L_{\mathrm{G}_X}}(i) - \lambda_{L_{\mathrm{G}_Y}}(i))^2$.

The results for the eigenvector similarity measures should have an inverse correlation (↓) with the P@1. The results in the left column of Table 3 show that the variant of GARI with loss $\mathcal{L}_{proc_{src}}$ yields a higher performance which aligns with our findings for the main experiments in Table 1. However, the ES scores for the model with $\mathcal{L}_2$ and $\mathcal{L}_{proc}$ show irregular behavior. We expect the model with the loss $\mathcal{L}_{proc}$ to have a lower value for the ES score compared to $\mathcal{L}_2$, which is in contrast to our findings in Table 3.

Pearson's Correlation(ρ). In order to calculate the Pearson's correlation, we first compute the pairwise cosine similarity scores for the seed translation pairs, i.e., $\{\cos(x_0, x_1), \cos(x_0, x_2), ..., \cos(x_s, x_s)\}$, and $\{\cos(y_0, y_1), \cos(y_0, y_2), ..., \cos(y_s, y_s)\}$. Later, we compute the Pearson's correlation between the lists of cosine similarity scores. We expect the Pearson's correlation score to correlate positively (↑) with the average P@1.

The results in the right half of Table 3 show the Pearson's correlation scores for all the variants of GARI. These results show an unclear behaviour, with $\mathcal{L}_{proc}$ showing better performance compared to $\mathcal{L}_2$ and $\mathcal{L}_{proc_{src}}$. This is in contrast to the results for P@1 reported in Table 1, where $\mathcal{L}_{proc_{src}}$ shows a better performance compared to other models.

To summarize our findings for the isometric metrics, we observe that these results do not truly correlate with the average P@1. These findings are consistent with the earlier study IsoVec (Marchisio et al., 2022) that also emphasized the need for better isomorphism metrics in order to portray the correct picture of the degree of relative isomorphism of the

GARI (w/o Graph Attention)		
source	target$^{'}$	target
الليزر	infrared	laser
الفهم	pronunciation	understanding
السجل	database	register
تلفاز	keyboards	tv
صدى	elated	echo
اربعة	three	four
زرقاء	foreboding	blue

Table 4: Example error cases for the model: GARI (w/o Graph Attention). The “target$^{'}$” represents the model predictions, “target” represents the ground truth.

embedding spaces.

6.3 Error Analysis

In this section, we perform a detailed analysis of the error cases of GARI in order to know: (i) the performance improvement attributable to the graph attention part of the model, (ii) limitations of the GARI, and room for potential improvement. For this, we perform error analysis on two variants of GARI, i.e., with and without graph attention layer. All experiments are performed using the in-domain settings using the best-performing model, i.e., GARI ($\mathcal{L}_{proc_{src}}$). Details are as follows:

GARI (w/o Graph Attention). We initially analyze the error cases for the basic variant of GARI (without the graph attention layer) that have been corrected by the complete model. The core focus of this analysis is to look for the translation instances that benefit especially due to the graph attention mechanism. Note, for this analysis, we only include error cases that have incorrect prediction for the basic model (i.e., without graph attention) and are correctly classified by the complete model GARI.

While the graph attention layer is able to correct approximately 11% of the errors made by the basic variant of GARI, we observe almost 72% of the error cases belong to the noun category. One possible explanation in this regard is that the phenomenon of multiple senses is more dominant among the nouns in contrast to other parts-of-the speech, e.g., verbs and adjectives, which makes it harder to control their relative isomorphism (Ali et al., 2019). Some examples in this regard have been shown in Table 4. We also observe that the majority of the predictions made by the basic variant of GARI are not semantically related to the true target words, which clearly indicates the need for information sharing among the semantically related words required to control the relative isomorphism of the embedding spaces.

GARI ($\mathcal{L}_{proc_{src}}$)		
source	target$^{'}$	target
ضرورية	vital	necessary
شمل	includes	included
القلم	pencil	pen
سماع	hear	hearing
المواهب	talents	talent
الفولاذ	metal	steel
المواصفات	certifications	specs

Table 5: Example error cases for GARI using the loss function $\mathcal{L}_{proc_{src}}$. The “target$^{'}$” represents the model predictions, and the “target” represents the ground truth.

GARI (The Complete Model). The end goal of performing error analysis on the complete model is to dig out the potential reasons and/or understanding of the limitations of the proposed model. Note, we perform this analysis for the best-performing variant of GARI, i.e., with the loss $\mathcal{L}_{proc_{src}}$.

We randomly select a subset of 50 error cases for quantification. To our surprise, most of the errors (approximately 65%) made by GARI are either semantically very close to the true target word or a lexical variant of the true target word. Some examples in this regard have been shown in Table 5. These results clearly show the current performance of GARI is underrated primarily due to the use of a very strict evaluation criterion, (i.e., P@1). This calls for the need for better and more sophisticated mechanisms for the BLI systems in order to measure the relative isomorphism of the geometric spaces.

To summarize, the error analysis shows the essence of using graph attention in order to control the relative isomorphism of the embedding spaces. It helps in incorporating and/or accumulating information across semantically related words in order to perform the end task in a robust way.

7 Conclusions and Future Research

In this work, we propose Graph Attention for Relative Isomorphism (GARI). GARI incorporates the impact of semantically related words in order to control the relative isomorphism of geometric spaces in a performance-enhanced way. Experimental evaluation using the Arabic data set shows that GARI outperforms the existing state-of-the-art research by 40.95% and 76.80% for in-domain and domain mismatch settings. In the future, we will extend this research to deep contextualized embeddings and non-euclidean geometries.

8 Limitations

Some of the core limitations of the proposed approach are outlined as follows: (i) all the techniques have been developed assuming a Euclidean geometry for the underlying embedding spaces, its extension to non-Euclidean spaces are still unaddressed, (ii) the existing problem formulation is not defined for the deep contextualized embeddings.

Acknowledgements. Di Wang, Yan Hu and Muhammad Asif Ali are supported in part by the baseline funding BAS/1/1689-01-01, funding from the CRG grand URF/1/4663-01-01, FCC/1/1976-49-01 from CBRC and funding from the AI Initiative REI/1/4811-10-01 of King Abdullah University of Science and Technology (KAUST). Di Wang is also supported by the funding of the SDAIA-KAUST Center of Excellence in Data Science and Artificial Intelligence (SDAIA-KAUST AI).

References

Hanan Aldarmaki and Mona Diab. 2019. Context-aware cross-lingual mapping. *arXiv preprint arXiv:1903.03243*.

Muhammad Asif Ali, Yifang Sun, Xiaoling Zhou, Wei Wang, and Xiang Zhao. 2019. Antonym-synonym classification based on new sub-space embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

David Alvarez-Melis and Tommi Jaakkola. 2018. Gromov-Wasserstein alignment of word embedding spaces. Brussels, Belgium. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2016. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 2289–2294.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2017. Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*.

Yerai Doval, Jose Camacho-Collados, Luis Espinosa-Anke, and Steven Schockaert. 2018. Improving cross-lingual word embeddings by meeting in the middle. *arXiv preprint arXiv:1808.08780*.

Long Duong, Hiroshi Kanayama, Tengfei Ma, Steven Bird, and Trevor Cohn. 2017. Multilingual training of crosslingual word embeddings. Association for Computational Linguistics.

Goran Glavas, Robert Litschko, Sebastian Ruder, and Ivan Vulic. 2019. How to (properly) evaluate cross-lingual word embeddings: On strong baselines, comparative analyses, and some misconceptions. *arXiv preprint arXiv:1902.00508*.

Pratik Jawanpuria, Arjun Balgovind, Anoop Kunchukuttan, and Bamdev Mishra. 2019. Learning multilingual word embeddings in latent metric space: a geometric approach. *Transactions of the Association for Computational Linguistics*, 7:107–120.

Armand Joulin, Piotr Bojanowski, Tomas Mikolov, Hervé Jégou, and Edouard Grave. 2018. Loss in translation: Learning bilingual word mapping with a retrieval criterion. *arXiv preprint arXiv:1804.07745*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Phrase-based & neural unsupervised machine translation. *arXiv preprint arXiv:1804.07755*.

Kelly Marchisio, Neha Verma, Kevin Duh, and Philipp Koehn. 2022. Isovec: Controlling the relative isomorphism of word embedding spaces. *arXiv preprint arXiv:2210.05098*.

Benjamin Marie and Atsushi Fujita. 2020. Iterative training of unsupervised neural and statistical machine translation systems. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

Barun Patra, Joel Ruben Antony Moniz, Sarthak Garg, Matthew R Gormley, and Graham Neubig. 2019. Bilingual lexicon induction with semi-supervision in non-isometric embedding spaces. *arXiv preprint arXiv:1908.06625*.

Sebastian Ruder, Ivan Vulić, and Anders Søgaard. 2019. A survey of cross-lingual word embedding models. *Journal of Artificial Intelligence Research*, 65:569–631.

Peter H Schönemann. 1966. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10.

Tal Schuster, Ori Ram, Regina Barzilay, and Amir Globerson. 2019. Cross-lingual alignment of contextual word embeddings, with applications to zero-shot dependency parsing. *arXiv preprint arXiv:1902.09492*.

Anders Søgaard, Sebastian Ruder, and Ivan Vulić. 2018. On the limitations of unsupervised bilingual dictionary induction. *arXiv preprint arXiv:1805.03620*.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Zirui Wang, Jiateng Xie, Ruochen Xu, Yiming Yang, Graham Neubig, and Jaime Carbonell. 2019. Cross-lingual alignment vs joint training: A comparative study and a simple unified framework. *arXiv preprint arXiv:1910.04708*.

Mozhi Zhang, Keyulu Xu, Ken-ichi Kawarabayashi, Stefanie Jegelka, and Jordan Boyd-Graber. 2019. Are girls neko or sh\= ojo? cross-lingual alignment of non-isomorphic embeddings with iterative normalization. *arXiv preprint arXiv:1906.01622*.

ArTrivia: Harvesting Arabic Wikipedia to Build A New Arabic Question Answering Dataset

Sultan Alrowili
Department of Computer Science
University of Delaware
Newark, Delaware, USA
alrowili@udel.edu

K.Vijay-Shanker
Department of Computer Science
University of Delaware
Newark, Delaware, USA
vijay@udel.edu

Abstract

We present ArTrivia, a new Arabic question-answering dataset consisting of more than 10,000 question-answer pairs along with relevant passages, covering a wide range of 18 diverse topics in Arabic. We created our dataset using a newly proposed pipeline that leverages diverse structured data sources from Arabic Wikipedia. Moreover, we conducted a comprehensive statistical analysis of ArTrivia and assessed the performance of each component in our pipeline. Additionally, we compared the performance of ArTrivia against the existing TyDi QA dataset using various experimental setups. Our analysis highlights the significance of often overlooked aspects in dataset creation, such as answer normalization, in enhancing the quality of QA datasets. Our evaluation also shows that ArTrivia presents more challenging and out-of-distribution questions to TyDi, raising questions about the feasibility of using ArTrivia as a complementary dataset to TyDi.

1 Introduction

In recent years, the field of question-answering (QA) in Arabic NLP has witnessed more attention with the introduction of several Arabic QA datasets, such as TyDi QA (Clark et al., 2020), the Arabic Reading Comprehension Dataset (ARCD) (Mozannar et al., 2019), and the Arabic Question-Answer Dataset (AQAD) (Atef et al., 2020). However, existing Arabic QA datasets have several issues, such as having limited topic diversity, picking common question patterns, and the limited size of the dataset.

First, although having a variety of topics is one of the objectives of TyDi QA creators (Clark et al., 2020) [1], many subjects such as classical Arabic poetry are less represented in TyDi. This issue also exists in both ARCD and AQAD datasets since they are generated from a limited number of articles.

Second, when crowd workers are given passages and asked to formulate questions with less defined guidelines, they tend to pick common patterns, which can compromise the quality of the dataset. Our analysis reveals that approximately 33% of the questions in Arabic TyDi are about explaining entities such as "Who is Alfred Nobel?" or "What is graphic design?" [2]. This is in contrast to both ARCD and English TyDi QA, where such questions consist of only about 4-5% of the dataset.

Third, when compared to the SQuAD dataset (Rajpurkar et al., 2016), which consists of 120,000 examples, Arabic QA datasets still have a limited dataset size, mainly due to the constraints imposed by the cost of crowd-sourcing. However, few studies in Arabic NLP explore alternative approaches to the crowd-sourcing method. Most of these approaches rely on Machine Translation, a method that has been criticized for it is poor performance in Arabic Question Answering (Antoun et al., 2021).

One suggested solution to address the existing challenges in Arabic Question Answering is to utilize Large Language Models (LLMs). These LLM models often use a zero-shot learning technique to address QA tasks, eliminating the need to finetune the Language Model on a specific question answering dataset. The zero-shot approach with LLMs in English QA tasks has shown promising results, often matching the supervised methods that require a finetuning dataset (Lai et al., 2023). However, recent studies in Arabic NLP show that the performance of the zero-shot approach with LLMs lags behind the supervised approach (Khondaker et al., 2023). The variation in performance between English and Arabic is derived from the fact that the English corpora represent a large portion of LLM's pre-training data, resulting in an inherent bias to-

[1] TyDi QA states that "The prompts are provided merely as inspiration to generate questions on a wide variety of topics"

[2] These questions are easy to formulate by appending the phrase "What/Who is" to the article title.

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 191–207
December 7, 2023 ©2023 Association for Computational Linguistics

Category	ARCD	ArabicSQuAD	AQAD	TyDi QA	ArTrivia
Number of Questions	1,395	48,344	17,911	15,726	10,045
Number of Passages	465	10,364	3381	11,319	7,982
Number of Articles	155	231	299	9,166	7,594
Questions Per Article	9.0	209.3	59.9	1.7	1.3
Crowd Workers	✓	✗	✗	✓	✗
Machine Translation	✗	✓	✗	✗	✗

Table 1: Summary of existing Arabic QA datasets compared to ArTrivia. Table adapted from (Atef et al., 2020)

ward English NLP tasks(Lai et al., 2023).

The existing issues in the Arabic QA dataset, which we previously discussed, motivate us to introduce our ArTrivia dataset. The ArTrivia dataset adopts two distinguished approaches. First, we rely on structured datasets from Wikipedia and a new proposed pipeline to generate our dataset, thereby mitigating the cost of crowd-sourcing and the issue of picking common patterns in question formulation. Second, we prioritize having a variety of topics in our dataset, including underrepresented topics, such as classical Arabic poetry.

Thus, the contributions of our paper are summarized in the following:

- We introduce a new novel pipeline to generate question-answer-passage triplets, which leverage various structured data sources from Arabic Wikipedia

- We introduce ArTrivia, a new Arabic Question Answering dataset comprising +10,000 question-answer-passage triplets, covering a wide range of 18 diverse topics in Arabic. We released ArTrivia dataset to the public at `https://github.com/salrowili/ArTrivia`.

- We conduct a statistical analysis of our dataset and a detailed evaluation of each component in our pipeline. In addition, we provide a detailed evaluation of our dataset against TyDi, using different setups to investigate the impact of out-of-distribution issue in TyDi QA dataset.

2 Related Work

In this section, we will provide an overview of existing Arabic datasets and Arabic Language Models, all of which are part of our evaluation setup.

2.1 Arabic Question Answering Datasets

Several Arabic Question Answering datasets have been introduced recently, including TyDi QA, AQAD, TyDi, ARCD, and ArabicSQuAD: a machine translation of the English SQuAD dataset (Mozannar et al., 2019). Table 1 provides a summary of these datasets. The table shows that ARCD, ArabicSQuAD, and AQAD utilize fewer than 300 articles for generating questions, with a higher ratio of questions per article. This higher question-per-article ratio suggests that despite having many questions, the diversity of articles and topics covered is limited. In addition, while the AQAD dataset does not rely on machine translation, it uses an algorithm to find a matched article in Arabic Wikipedia to those on the English SQuAD dataset. Thus, it includes the same topics covered in the English SQuAD dataset.

On the other hand, TyDi relies on crowd workers for dataset creation and is also part of multilanguage datasets. Thus, the TyDi dataset may have a limited representation of specific topics related to the Arabic language, such as classical Arabic poetry. In contrast, we can observe from the table that ArTrivia stands out among other datasets as the only dataset that employs many articles for question generation without depending on Crowd Workers or Machine Translation. Furthermore, despite both the TyDi and ARCD datasets relying on crowd-sourcing for dataset generation, ArTrivia still maintains a lower question-per-article ratio of 1.3 in comparison to TyDi and ARCD datasets.

2.2 Arabic Language Models

The introduction of the BERT (Bidirectional Encoder Representations from Transformers) model (Devlin et al., 2019), has shown impressive results on English question-answering tasks. Consequently, several Arabic Language Models have adopted BERT-like models, such as AraELECTRA (Antoun et al., 2021), AraBERT (Antoun et al., 2020), and ArabicTransformer (Alrowili and Shanker, 2021). These models represent the state-of-the-art models in Arabic QA for both TyDi and ARCD datasets. Recently, the advent of Generative Large Language Models (LLMs) like ChatGPT (OpenAI, 2023) has also demonstrated considerable potential in English QA tasks, especially with a zero-shot approach. However, the performance of LLMs such as ChatGPT and Google PaLM 2 still lags behind the typical supervised approach with BERT-like models on Arabic QA tasks as shown by Khondaker et al. (2023) and Anil et al. (2023).

3 Building ArTrivia Dataset

Our approach to build our ArTrivia dataset consists of of two components: (1) generate question-answer pairs from various structured data from Wikipedia, and (2) build a new pipeline that consists of multiple functions to generate our question-answer-passage triplets. First, we will explain in section 3.1, our method to generate question-answer pairs. Then, in section 3.2, we will explain our proposed pipeline to generate our question-answer-passage triplets.

3.1 Question-Answer Pairs Collection

In Figure 1, we illustrate the data collection process of our ArTrivia question-answer pairs from different Wikipedia sources including Wiki Tables, Wiki-Data, WikiList and Wiki Entity Description.

Wikipedia Tables The first method of creating our question-answer pairs is derived from tables within Arabic Wikipedia articles. These tables have a set of relationships between two or more items in the table (e.g., a list of capital cities). We exploit these relations to formulate our question-answer pairs. The first part of the relationship will form the question and keyword (A), and the second part will serve as the answer (B). Then, we will use a fixed term for each set of relations to form our question (e.g. What is the capital of (A) country? Answer: (B). The selection of these tables is based on two criteria: (a) questions can be answered by trivia enthusiasts, (b) covers a wide variety of topics (e.g., history, poetry), and a variety of question types (e.g., numbers, dates, persons, and places).

Wiki Data The second method shares similarities with the WikiTables but leverages structured datasets related to specific entities within Wikipedia, utilizing a knowledge base known as WikiData. The WikiData stores valuable relationships for each entity. For example, Rome's entity in WikiData includes relationships like capital city, inception, nickname, and "founded by." Similarly, Thomas Edison's entity has information like country of citizenship, date of birth, and notable work. Thus, by utilizing these relationships, we generate additional QA pairs in our dataset. Our choice of these entities depends on the entity's popularity, measured by the number of languages to which this particular entity has been translated to.

Wikipedia List We observe that the TyDi QA dataset has a limited number of long questions (e.g., terms in economics). To address this gap, we generate 591 question-answer pairs from Wikipedia lists using a simple parser as illustrated in Figure 1.

Wikipedia Entity Description While we were able to generate over 12.7K question-answer pairs using both WikiData and Wiki Tables, we still have a challenge in generating certain types of questions that require more complexity (e.g., smallest planet, second largest country). We observe that Wikipedia annotators populate valuable information for each entity (e.g., persons, places, novels) in the central description of the article title, as shown in figure 6. This information provides a short summary of each entity (article title), highlighting important information related to this entity. For example, an article with the title "Mercury" has a central description that says, "smallest and closest planet to the sun in the Solar System." By using the ChatGPT prompt as illustrated in Figure 1, we can generate a related question for this entity.

Our selection process for entities for this type of question depends on the entity's popularity, following these steps: First, we use the Arabic Wikipedia dump to extract all article titles, selecting only those with central descriptions. Second, we sort these entities based on the number of languages each has been translated into. Finally, we exclude entities that lack a sufficient description to form a question that can be answered by trivia enthusiasts.

3.2 Building ArTrivia Pipeline

We explained earlier our methods to generate question-answer pairs from WikiTable, WikiData, Wiki List, and Wiki Entity Description. However, to build question-answer-passage triplets, we need to find the relevant passage for each question-answer pair. To address this part, we propose a

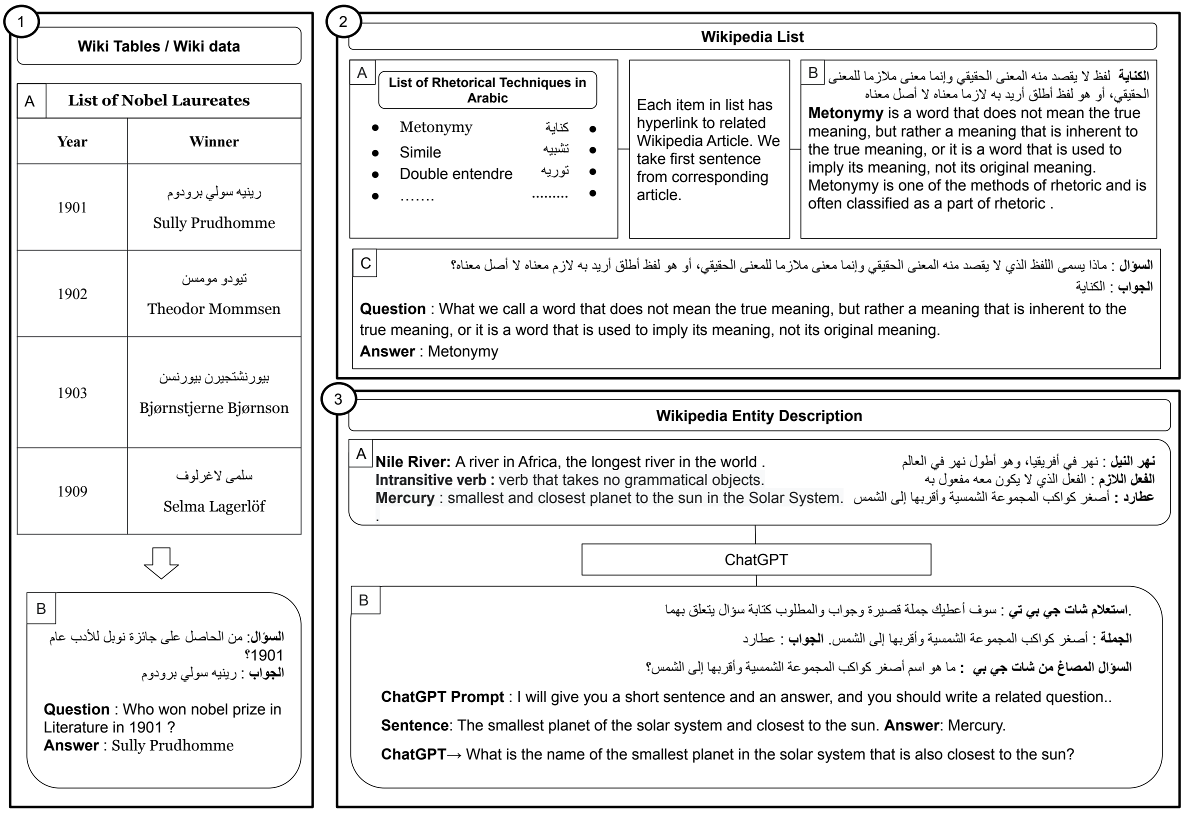

1
Wiki Tables / Wiki data
A
List of Nobel Laureates
Year
Winner
1901
رينيه سولي برودوم
Sully Prudhomme
1902
تيودور مومسن
Theodor Mommsen
1903
بيورنشتجيرن بيورنسن
Bjørnstjerne Bjørnson
1909
سلمى لاغرلوف
Selma Lagerlöf
B
السؤال: من الحاصل على جائزة نوبل للأدب عام 1901؟
الجواب : رينيه سولي برودوم
Question : Who won nobel prize in Literature in 1901 ?
Answer : Sully Prudhomme
2
Wikipedia List
A
List of Rhetorical Techniques in Arabic
Metonymy كناية
Simile تشبيه
Double entendre توريه
……. ....
Each item in list has hyperlink to related Wikipedia Article. We take first sentence from corresponding article.
B
الكناية لفظ لا يقصد منه المعنى الحقيقي وإنما معنى ملازما للمعنى الحقيقي، أو هو لفظ أطلق أريد به لازما معناه لا أصل معناه
Metonymy is a word that does not mean the true meaning, but rather a meaning that is inherent to the true meaning, or it is a word that is used to imply its meaning, not its original meaning. Metonymy is one of the methods of rhetoric and is often classified as a part of rhetoric .
C
السؤال : ماذا يسمى اللفظ الذي لا يقصد منه المعنى الحقيقي وإنما معنى ملازما للمعنى الحقيقي، أو هو لفظ أطلق أريد به لازم معناه لا أصل معناه؟
الجواب : الكناية
Question : What we call a word that does not mean the true meaning, but rather a meaning that is inherent to the true meaning, or it is a word that is used to imply its meaning, not its original meaning.
Answer : Metonymy
3
Wikipedia Entity Description
A
Nile River: A river in Africa, the longest river in the world .
نهر النيل : نهر في أفريقيا، وهو أطول نهر في العالم
Intransitive verb : verb that takes no grammatical objects.
الفعل اللازم : الفعل الذي لا يكون معه مفعول به
Mercury : smallest and closest planet to the sun in the Solar System.
عطارد : أصغر كواكب المجموعة الشمسية وأقربها إلى الشمس
ChatGPT
B
.استعلام شات جي بي تي : سوف أعطيك جملة قصيرة وجواب والمطلوب كتابة سؤال يتعلق بهما
الجملة : أصغر كواكب المجموعة الشمسية وأقربها إلى الشمس. الجواب : عطارد
السؤال المصاغ من شات جي بي : ما هو اسم أصغر كواكب المجموعة الشمسية وأقربها إلى الشمس؟
ChatGPT Prompt : I will give you a short sentence and an answer, and you should write a related question..
Sentence: The smallest planet of the solar system and closest to the sun. Answer: Mercury.
ChatGPT→ What is the name of the smallest planet in the solar system that is also closest to the sun?

Figure 1: Overview of our method to build the ArTrivia question-answer pairs from different Wikipedia sources.

new novel pipeline that consists of (1) a BM25 retriever (Robertson and Zaragoza, 2009) (2) fuzzy match and approximation functions, (3) a places parser, (4) answer normalization functions, and (5) ChatGPT as an annotation tool to filter irrelevant passages. An overview of our proposed pipeline is illustrated in Figure 2.

Finding Relevant Passage First, it is important to highlight that this stage is not necessary for questions sourced from the Wiki List and Wiki Entity description. In the case of Wiki List, each list item has a hyperlink to the associated related article. Thus, we can consider the first passage in this related article as the relevant passage. Similarly, for questions derived from Wiki Entity descriptions, our entity corresponds to article titles, as mentioned earlier. In the majority of cases, the essential information required to address the question is adequately present in the first passage.

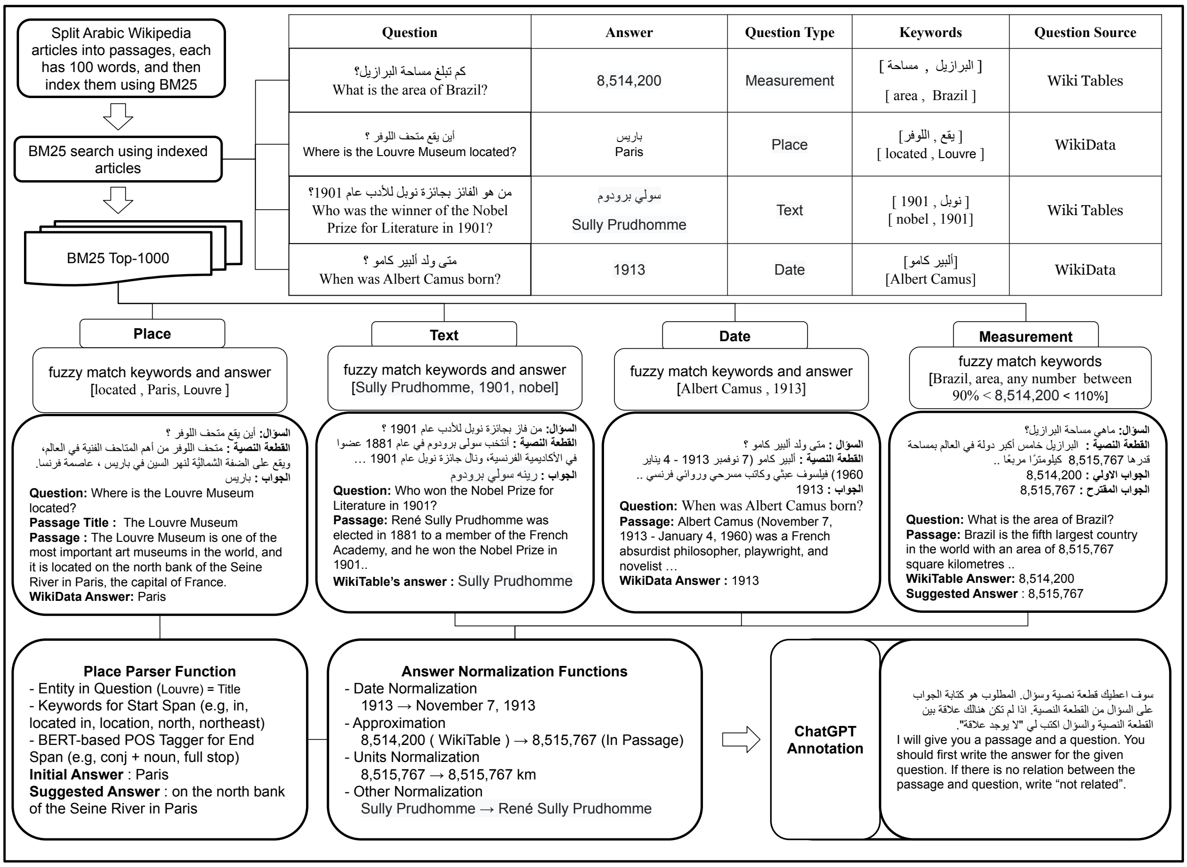
However, for both WikiTable and Wiki data question answer pairs, we need to find the relevant passage using a retrieval model. To build our retrieval component, we first split articles from the Arabic Wikipedia (June 2023) into 100 words, each representing a passage. Then, we use the sparse-based retrieval BM25 with the Pyserini tool (Lin et al., 2021) to build our indexed Arabic Wikipedia [3].

Fuzzy Matching To control the quality of retrieved passages, we use the first elements in the WikiTables and WikiData as a keyword. For example, a related passage for a question-answer pair says, "Who was the winner of the Nobel Prize for Literature in 1901? Sully Prudhommem", should have the following keywords:(1) Nobel, (2) 1901, and (3) Sully Prudhommem.

However, relying on the exact match of keywords to control the retrieved message will eliminate many passages that have the keywords but with different forms. This case will be worse with morphologically rich language such as Arabic. Thus, we integrate an approximate string matching (fuzzy match) function with our pipeline, which is based on the "thefuzz" library (Adam, 2023).

In addition, to handle measurement-related questions, we use an approximation function that accepts answers within +/- 10% of the actual value. The reason to include an approximation function

[3] While alternative approaches like DPR: Dense Passage Retrieval (Karpukhin et al., 2020) could be considered, we use BM25 to maintain simplicity in our pipeline.


Split Arabic Wikipedia articles into passages, each has 100 words, and then index them using BM25
BM25 search using indexed articles
BM25 Top-1000
Place
fuzzy match keywords and answer [located , Paris, Louvre]
Text
fuzzy match keywords and answer [Sully Prudhomme, 1901, nobel]
Date
fuzzy match keywords and answer [Albert Camus , 1913]
Measurement
fuzzy match keywords [Brazil, area, any number between 90% < 8,514,200 < 110%]
ChatGPT Annotation

Question	Answer	Question Type	Keywords	Question Source
كم تبلغ مساحة البرازيل؟ What is the area of Brazil?	8,514,200	Measurement	[البرازيل , مساحة] [area , Brazil]	Wiki Tables
أين يقع متحف اللوفر ؟ Where is the Louvre Museum located?	باريس Paris	Place	[يقع , اللوفر] [located , Louvre]	WikiData
من هو الفائز بجائزة نوبل للأدب عام 1901؟ Who was the winner of the Nobel Prize for Literature in 1901?	سولي برودوم Sully Prudhomme	Text	[نوبل , 1901] [nobel , 1901]	Wiki Tables
متى ولد ألبير كامو ؟ When was Albert Camus born?	1913	Date	[ألبير كامو] [Albert Camus]	WikiData

السؤال: أين يقع متحف اللوفر ؟
القطعة النصية : متحف اللوفر من أهم المتاحف الفنية في العالم،
ويقع على الضفة الشماليَّة لنهر السين في باريس ، عاصمة فرنسا.
الجواب : باريس
Question: Where is the Louvre Museum located?
Passage Title : The Louvre Museum
Passage : The Louvre Museum is one of the most important art museums in the world, and it is located on the north bank of the Seine River in Paris, the capital of France.
WikiData Answer: Paris

السؤال: من فاز بجائزة نوبل للأدب عام 1901 ؟
القطعة النصية : أنتخب سولى برودوم في عام 1881 عضوا
في الأكاديمية الفرنسية، ونال جائزة نوبل عام 1901 …
الجواب : رينه سولي برودوم
Question: Who won the Nobel Prize for Literature in 1901?
Passage: René Sully Prudhomme was elected in 1881 to a member of the French Academy, and he won the Nobel Prize in 1901..
WikiTable's answer : Sully Prudhomme

السؤال : متى ولد ألبير كامو ؟
القطعة النصية : ألبير كامو (7 نوفمبر 1913 - 4 يناير
1960) فيلسوف عبثي وكاتب مسرحي وروائي فرنسي ..
الجواب : 1913
Question: When was Albert Camus born?
Passage: Albert Camus (November 7, 1913 - January 4, 1960) was a French absurdist philosopher, playwright, and novelist …
WikiData Answer : 1913

السؤال: ماهي مساحة البرازيل؟
القطعة النصية : البرازيل خامس أكبر دولة في العالم بمساحة
قدرها 8,515,767 كيلومترًا مربعًا ..
الجواب الاولي : 8,514,200
الجواب المقترح : 8,515,767
Question: What is the area of Brazil?
Passage: Brazil is the fifth largest country in the world with an area of 8,515,767 square kilometres ..
WikiTable Answer: 8,514,200
Suggested Answer : 8,515,767

Place Parser Function
- Entity in Question (Louvre) = Title
- Keywords for Start Span (e.g, in, located in, location, north, northeast)
- BERT-based POS Tagger for End Span (e.g, conj + noun, full stop)

Initial Answer : Paris
Suggested Answer : on the north bank of the Seine River in Paris

Answer Normalization Functions
- Date Normalization
  1913 → November 7, 1913
- Approximation
  8,514,200 (WikiTable) → 8,515,767 (In Passage)
- Units Normalization
  8,515,767 → 8,515,767 km
- Other Normalization
  Sully Prudhomme → René Sully Prudhomme

سوف اعطيك قطعة نصية وسؤال. المطلوب هو كتابة الجواب
على السؤال من القطعة النصية. اذا لم تكن هنالك علاقة بين
القطعة النصية والسؤال اكتب لي "لا يوجد علاقة".
I will give you a passage and a question. You should first write the answer for the given question. If there is no relation between the passage and question, write “not related”.

Figure 2: Overview of our proposed pipeline to build our ArTrivia question-answer and relevant passage triplets.

is to mitigate the disputes related to measurements (e.g., rivers' length, country areas, and populations) between structured data (WikiData, WikiTables) and the related passage.

Place Parser Function We find that in questions related to places (e.g., museums and cities locations), in most cases, the retrieved passage will have a better ground truth answer than the initial answer derived from Wiki Tables and WikiData as shown in our example in Figure 2. Thus, we construct a parser to revise answers related to questions about places according to the related passage.

Our Place Parser Function consists of three key components. The first step is to choose the top retrieved passages where the passage's title matches the corresponding place name mentioned in the question. For example, if we have a question inquiring about the location of London City, with "London" as the keyword, we will specifically select passages whose titles match "London."

The second step employs a fuzzy keyword-matching approach to check if any word of our predefined list of places keywords in the passage. These keywords include terms such as "located in," "north," "south," and "northeast". These keywords help us determine the starting point of the answer span within the passage.

Finally, in the third step, we utilize an Arabic BERT-based POS (Part-of-Speech) tagger (Inoue et al., 2021) to determine the end span of the answer. This tagger employs specific POS tags and follows a set of predefined conditions, including, for example, the presence of punctuation marks or conjunctions followed by Proper Nouns.

Answer Normalizing Function To address discrepancies between initial answers from our QA pairs and corresponding ground truths in related passages, we have added an answer-normalizing function to our pipeline. This normalization function targets three aspects: (1) variations in date formats (e.g., "1913" to "November 7, 1913"), (2) differences in units and formatting (e.g., "2400" to "2400 km"), and (3) entities' alternative names (e.g., "Thomas Edison" to "Thomas Alva Edison").

To tackle inconsistencies in dates and units, we first create a reference file containing a list of words related to dates and units. This list includes months, possible years (e.g., 1-3000), and a list of units (e.g., km, mile, square km). Our normalization function operates by starting from the index of the original answer's location and scanning adjacent terms on both sides. Then, It appends matching terms from our list until it encounters an unlisted term.

For example, consider the sentence "Albert Camus (November 7, 1913 - January 4, 1960) was a French absurdist philosopher, playwright, and novelist". If the initial answer from WikiTable is "1913", our Answer Normalizing function scans adjacent terms, identifying "November" and "7" as valid matches from our list. However, it stops upon encountering symbols like "-" or "(", which are not part of our list of units.

Furthermore, we have improved our normalization function to handle questions involving date ranges (e.g., When did the Macedonian Empire rule?). To address the date ranges question, we simply include symbols and terms associated with date ranges in our reference file, such as "-", "till", "between", "until", and "continued till". For example, when changing the query from "When was Albert Camus born?" to "When did Albert Camus live?", our normalizing function will not stop at the "-" symbol. Instead, it continues scanning left and right until encountering a token not in our list, such as the "(" and ")" symbols. As a result, the normalized ground truth for this example would be "November 7, 1913 - January 4, 1960". We manually flag questions related to date and date range, which is a trivial task since our dataset consists of a set of relationships sharing similar answer types.

Data Annotation with ChatGPT Several studies have suggested that LLM models such as ChatGPT could be used as data annotation tools (Gilardi et al., 2023), (Huang et al., 2023). Drawing from these encouraging findings, we added an additional phase into our pipeline. This phase leverages a ChatGPT prompt, as shown in Figure 2, to filter irrelevant passages generated from our pipeline and validate the suggested answer from our pipeline.

Quality Control This stage is to assess the quality of our pipeline and ChatGPT as an annotation tool. In this stage we manually examine question triplets from our pipeline for the following points (1) if the selected passage is related, (2) if the suggested answer from our pipeline represents the ground truth and revise the ground truth if needed (3) For the development set, we add any other possible alternative answers in the related passage.

Stage	#Question
Question-Answer Pairs	15,149
Pipeline (QA Pairs + Passage)	11,527
Filtering Non-Relevant Passages	-1,482
Final ArTrivia Dataset	10,045
- WikiTables	4,916
- WikiData	2,987
- Wiki Entity Description	1,579
- WikiList	563

Table 2: Detailed statistics about our ArTrivia dataset.

4 Dataset Analysis

In this section, we will focus on the quantitative analysis of our dataset. Then, in section 5, we will focus on evaluating the performance of ArTrivia as a training and evaluation dataset.

QA Pairs Collection The data collection is quantitatively summarized in Tables 2. This collection includes 15,149 question-answer pairs sourced from a variety of Wikipedia origins, with contributions from WikiTables (49%), WikiData (35%), WikiList (5%), and Wiki Entity Description (11%).

Results of our Pipeline The employment of our pipeline, as illustrated in Figure 2, successfully identified a relevant passage for 11,527 question-answer pairs. In contrast, the pipeline could not find a related passage for 3,622 question-answer pairs. This outcome was in line with our expectations, knowing that Arabic Wikipedia consists of only 2.1 million articles compared to English Wikipedia, which has more than 16 million articles.

Manual and ChatGPT Filtering By utilizing the ChatGPT prompt against 11,527 triplets from our pipeline, a total of 566 passages were classified by ChatGPT as non-relevant. Subsequently, employing manual filtering against the same 11,527 triplets resulted in the identification of 1,482 irrelevant passages. Next, by comparing the manual filtering process to ChatGPT's filtering, we discovered that 154 question-answer-passage triplets were incorrectly classified as irrelevant by ChatGPT. Thus, despite the promising results in English, our result shows that ChatGPT lags behind human performance in Arabic QA annotation.

Question Word	ArTrivia	TyDi
What (ما)	36.6%	30.4%
When (متى)	22.8%	28.9%
Who (من هو)	23.8%	17.9%
Where (أين)	13.3%	12.0%
How Much / Many (كم)	3.5%	9.61%
YES/NO (هل)	<1%	<1%
How (كيف)	<1%	<1%
Why (لماذا)	-	<1%

Table 3: Distribution of ArTrivia by question word against the Arabic portion of TyDi QA.

Final ArTrivia Dataset The final dataset consists of 10,045 question-answer-passages triplets, which suggests that the accuracy of our pipeline in retrieving related passages is 87% (10,045 out of 11,527). Following the final manual filtering stage, we split our ArTrivia dataset into training and development sets. Our strategy was to select 20% of each relationship (e.g., list of capitals) in WikiTables, Wiki Data as part of our development set. For Wiki List and Entity Description questions, we randomly selected 20% of the dataset for the development set. This split ensures that the development set is representative of the entire dataset.

Topics Distribution In Table 6, we show the distribution of topics in ArTrivia, which shows that ArTrivia covers a wide variety of 18 topics. The distribution of topics also shows how we addressed under-represented topics in existing Arabic QA datasets, such as Arabic Literature, Cartoon Movies, Arabic Cinema, and National dishes. We show examples of each of these categories along with other categories in Figure 3 and Figure 4. The table also shows that History, Geography, and "Dates of Birth/Death of Famous people" are the most represented topics in our dataset. However, many of the questions under History topics are in the grey area of other topics such as politics, geopolitics, and world organization history.

Question Word Distribution In Table 3, we compare the distribution of question words in our dataset against TyDi. While our ArTrivia dataset shows a higher proportion than TyDi for both "what" and "who" questions, TyDi still has a larger overall number of questions for these categories. On the other hand, we can observe that our dataset demonstrates a lower proportion of questions starting with the "when" word compared to Arabic TyDi. It is worth noting that the English subset of the TyDi QA dataset constitutes only 14% of the entire question pool dedicated to "when" questions. It is also important to note that "where" questions are mostly categorized into "Geography" topics. Thus, increasing the questions in "Geography" topics has helped us maintain a similar distribution of "where" questions to the TyDi QA dataset.

Furthermore, the table shows that both the Arabic TyDi dataset and our ArTrivia dataset contain less than 1% of "YesNo" and "How" questions. Most of the questions in these two categories come from WikiData, indicating it is effectiveness in generating these types of questions. It is also worth noting that TyDi includes 31 questions related to "why" questions, which we encountered challenges in generating using our pipeline.

5 Pipeline and Dataset Evaluation

In this section, we will first discuss the evaluation performance of our proposed pipeline to highlight the impact of each stage in our pipeline. Then, we will discuss and compare the performance of ArTrivia and TyDi datasets using different setups for training and evaluation sets. This will help us to study the out-of-distribution and study how ArTrivia can serve as a complementary dataset to TyDi QA.

5.1 ArTrivia Pipeline Evaluation

Table 4 shows a comprehensive evaluation of our pipeline using the AraELECTRA model on the TyDi$_{\text{short}}$ dataset. The TyDi$_{\text{short}}$ subset of TyDi eliminates questions that inquire about entity explanations (e.g., What is a space galaxy?). The main objective of this evaluation is to assess the individual contributions of each component in our pipeline against our baseline dataset (TyDi training set), shown in the last row of the table.

Initially, we use a basic approach to retrieve question-answer-passage triplets by checking if answers can be located within passages retrieved by the BM25 model. This basic strategy yields an EM/F1 score of 38.3/57.7 and retrieved 13,407 triplets. Then, we introduce a second strategy that uses the question keywords to reduce the possibility of retrieving irrelevant passages, which resulted in a slight improvement of the EM score to 40.4.

Stage	#Q	EM/F1
ArTrivia QA Pairs	15,149	-
ArTrivia Pipeline		
+ Answer In Passage	13,407	38.3/57.7
+ Question Keywords	10,290	40.4/60.2
+ FuzzyMatch	11,265	40.9/60.1
+ Approximation	11,462	40.6/60.3
+ Date Normalization	11,459	56.8/70.1
+ Other Normalization	11,459	62.2/72.1
+ PlaceParser	11,527	64.9/76.1
ArTrivia Quality Control		
- Irrelevant Passages	10,045	64.9/75.0
+ Revised Answer Span	10,045	70.0/79.9
Baseline Dataset		
TyDi Training Dataset	14,805	74.9/84.3

Table 4: Exact Match (EM) and F1 scores of AraELECTRA with our pipeline on TyDi$_{\text{Dev-Short}}$.

However, using the exact match of keywords with morphologically rich languages like Arabic, where words can have different forms, causes a reduction for our triplets by 3,117 examples. To address this issue, we incorporate a FuzzyMatch function, which restores over 975 triplets. We then include another function in our pipeline, the approximation function. This function recovers an additional 197 triplets and raises the F1 score to 60.3.

However, the most significant improvement in our pipeline occurs with the introduction of our normalization functions, which increases our pipeline's performance from 40.6/60.3 to 62.2/72.1. These results highlight the often overlooked role of answer normalization in enhancing the overall quality of question-answering datasets.

Furthermore, the PlaceParser function, detailed in section 3.2, substantially improves our pipeline's performance, contributing an additional 2-4 points to the EM/F1 score. Thus, the final performance with our pipeline without the additional manual quality control is 64.9/76.1 compared to the TyDi training set, which achieved 74.9/84.3.

Finally, our quality control stage, as outlined in Section 3.2, had significantly contributed to the overall performance, improving our score to 70.0/79.9. This significant enhancement was primarily attributed to the manual refinement of the start and end spans of the suggested answers generated by our pipeline. In total, we undertook a total of 1,378 answer span revisions ranging from minor to major revisions. These revisions include cases where (1) the suggested answer from our pipeline is a single entity, where ground truth is a multiple entities (2) the presence of unusual date formats and units within the passage, and (3) the absence of essential prefixes, suffixes, and articles.

On the other hand, we can observe that filtering out irrelevant passages did not yield any additional improvements in our results. This finding implies that the Language Model can tolerate having irrelevant passages in the training set without compromising performance. However, this filtering step remains critical to maintain the quality of our ArTrivia dev set as a reliable evaluation dataset.

5.2 ArTrivia Evaluation

In Table 5, we provide an evaluation of our Final ArTrivia dataset in comparison to TyDi. We finetune the AraELECTRA model with various dataset configurations for this evaluation. Our primary objective is to assess how well our ArTrivia dataset could serve as a complementary dataset to TyDi and to examine the challenge that ArTrivia introduces to a language model trained on TyDi QA.

In the first two rows, we show the evaluation of ArTrivia and TyDi on the complete TyDi dataset. We can observe in row 2 that the ArTrivia dataset shows a lower performance in this setup, with an F1 score of 60.7, against a score of 86.8 with TyDi QA. These results are as expected, given that our ArTrivia dataset only addresses short-answer questions. In contrast, TyDi includes a substantial 33% of its dataset dedicated to long-answer questions, which typically ask to explain an entity.

Next, in rows 4-5, we replicate a similar evaluation setup presented in Table 4 by removing 300 questions that typically have long answers from TyDi QA. With this setup, the gap with TyDi significantly decreased to less than 4.4 in the F1 score. The marginal gap between ArTrivia and the TyDi training set is expected, given that ArTrivia is an out-of-distribution dataset for TyDi. Indeed, this gap is larger when we evaluate TyDi on the ArTrivia development set, as we will discuss next.

In rows 7-8, we replaced the development dataset in our table with our ArTrivia dev dataset. Comparing rows 7 and 8 shows how significantly the TyDi training set underperforms against our ArTrivia with a margin of 8.4 in the F1 score and 10.4 in the EM score. Considering that ArTrivia includes a manual quality control phase that examines ev-

Row	Train Size	Eval Size	Train Data	Eval Data	EM	F1
1	14,805	921	TyDi$_{\text{Train}}$	TyDi$_{\text{Dev-Full}}$	74.5	86.8
2	10,045	921	ArTrivia	TyDi$_{\text{Dev-Full}}$	49.1	60.7
3	24,805	921	ArTrivia + TyDi$_{\text{Train}}$	TyDi$_{\text{Dev-Full}}$	74.7	86.6
4	14,805	621	TyDi$_{\text{Train}}$	TyDi$_{\text{Dev-Short}}$	74.9	84.3
5	10,045	621	ArTrivia	TyDi$_{\text{Dev-Short}}$	70.0	79.9
6	24,805	621	ArTrivia + TyDi$_{\text{Train}}$	TyDi$_{\text{Dev-Short}}$	75.7	85.1
7	14,805	1,700	TyDi$_{\text{Train}}$	ArTrivia$_{\text{Dev}}$	79.0	84.9
8	8,345	1,700	ArTrivia$_{\text{Train}}$	ArTrivia$_{\text{Dev}}$	89.4	93.3
9	23,150	1,700	TyDi$_{\text{Train}}$ + ArTrivia$_{\text{Train}}$	ArTrivia$_{\text{Dev}}$	89.5	93.0

Table 5: The Exact Match (EM) and F1 scores of the AraELECTRA model using different setups for Training and Development datasets with ArTrivia and TyDi QA. In the TyDi$_{\text{Dev-Short}}$ setup, questions that ask about entity descriptions (e.g., "Who is Alfred Nobel?") were excluded since these questions typically have long answers.

ery example in our development set, it is clear that the decline in performance with the TyDi is not attributed to the poor quality of ArTrivia. This suggests that ArTrivia presents a more challenging question to TyDi.

In Figure 5, we present examples that highlight the challenges our dataset poses for the TyDi QA dataset. The Figure illustrates that in the majority of these examples, the Language Model tends to select the first entity corresponding to the question type. For example, when the question asks about a person entity, such as a novel author, and the passage has another person's name before the actual answer, the Language Model often selects the first name mentioned in the passage. This pattern is consistently observed with dates, places, and other types of entities as well. These cases raise important questions about whether the Language Model relies on context to answer a given question or if it simply adapts to common patterns associated with question words (e.g., when, where, who). This also suggests that we could build a more challenging QA dataset in the future based on these observations.

Finally, we should also note that by contrasting rows 5 and 7, we can observe that ArTrivia present more tolerance for out-of-distribution issue than TyDi. Furthermore, upon comparing rows 3, 6, and 9, we can conclude that combining TyDi and ArTrivia as complementary datasets achieves almost the best score against TyDi and ArTrivia devolvement datasets.

6 Conclusion

In this paper, we introduce ArTrivia, a novel dataset consisting of over +10,000 question triplets, covering a wide range of 18 diverse topics. We present a detailed description of our proposed pipeline and conduct a comprehensive analysis of the contribution of each component to overall performance. While most of the existing research on Question Answering datasets primarily focuses on question formulation and passage selection, our work emphasizes the overlooked, yet crucial, role of answer normalization in the quality of QA datasets. Our results also highlight the out-of-distribution issue within TyDi when presented with more challenging questions. In future work, we plan to adapt our proposed pipeline to different domains and languages such as creating a new Multi-Language QA dataset.

Ethics Statement

The ArTrivia dataset is collected from different sources of Arabic Wikipedia structured datasets. These datasets are populated by human annotators (Wikipedia Contributors). Wikipedia adapts the Neutral point of view (NPOV) policy [4], defined as "representing fairly, proportionately, and, as far as possible, without editorial bias, all the significant views that have been published by reliable sources on a topic."

Acknowledgements

The authors would like to acknowledge the ultimate support from Google Research Cloud TRC for providing access to Tensor Processing Unit TPUs. In this paper, we employ TPU VM, which has 96 CPUs and 350 GB of RAM, to efficiently run our QA pipeline in parallel and evaluate our dataset.

[4] https://en.wikipedia.org/wiki/Wikipedia:Neutral_point_of_view/FAQ

Limitations

ArTrivia uses Wiki Tables from Arabic Wikipedia to generate a large proportion of the dataset. One limitation of this method is that it lacks having more complicated questions such as "Why" questions, as shown in Table 3. In the future, we plan to overcome this limitation by finding new methods to retrieve these types of questions from new sources of structured data in Wikipedia.

Moreover, another limitation of this work is that we have a limited number of structured tables in Arabic Wikipedia. However, we plan to overcome this limitation by using machine translation of tables from English Wikipedia. Using machine translation for a dataset like SQuAD may not yield optimal results. However, using machine translation with our method that uses a table from Wikipedia may yield better results since we often in this case translate entities in the table rather than translating complete passages.

Finally, another limitation of our dataset creation method is that we use a fixed term to generate question-answer pairs for each set of relationships from WikiTable and WikiData. However, this limitation can be easily overcome by using LLMs (e.g. ChatGPT) to generate the question phrase for each relationship in the structured dataset.

References

Cohen Adam. 2023. thefuzz. https://github.com/seatgeek/thefuzz.

Sultan Alrowili and Vijay Shanker. 2021. ArabicTransformer: Efficient large Arabic language model with funnel transformer and ELECTRA objective. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1255–1261, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernandez Abrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vlad Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, Guy Gur-Ari, Steven Hand, Hadi Hashemi, Le Hou, Joshua Howland, Andrea Hu, Jeffrey Hui, Jeremy Hurwitz, Michael Isard, Abe Ittycheriah, Matthew Jagielski, Wenhao Jia, Kathleen Kenealy, Maxim Krikun, Sneha Kudugunta, Chang Lan, Katherine Lee, Benjamin Lee, Eric Li, Music Li, Wei Li, YaGuang Li, Jian Li, Hyeontaek Lim, Hanzhao Lin, Zhongtao Liu, Frederick Liu, Marcello Maggioni, Aroma Mahendru, Joshua Maynez, Vedant Misra, Maysam Moussalem, Zachary Nado, John Nham, Eric Ni, Andrew Nystrom, Alicia Parrish, Marie Pellat, Martin Polacek, Alex Polozov, Reiner Pope, Siyuan Qiao, Emily Reif, Bryan Richter, Parker Riley, Alex Castro Ros, Aurko Roy, Brennan Saeta, Rajkumar Samuel, Renee Shelby, Ambrose Slone, Daniel Smilkov, David R. So, Daniel Sohn, Simon Tokumine, Dasha Valter, Vijay Vasudevan, Kiran Vodrahalli, Xuezhi Wang, Pidong Wang, Zirui Wang, Tao Wang, John Wieting, Yuhuai Wu, Kelvin Xu, Yunhan Xu, Linting Xue, Pengcheng Yin, Jiahui Yu, Qiao Zhang, Steven Zheng, Ce Zheng, Weikang Zhou, Denny Zhou, Slav Petrov, and Yonghui Wu. 2023. Palm 2 technical report.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. AraBERT: Transformer-based model for Arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France. European Language Resource Association.

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

Adel Atef, Bassam Mattar, Sandra Sherif, Eman Elrefai, and Marwan Torki. 2020. Aqad: 17,000+ arabic questions for machine comprehension of text.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

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.

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).

Fan Huang, Haewoon Kwak, and Jisun An. 2023. Is chatgpt better than human annotators? potential and limitations of chatgpt in explaining implicit hate speech. In *Companion Proceedings of the ACM Web Conference 2023*, WWW '23 Companion, page 294–297, New York, NY, USA. Association for Computing Machinery.

Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2021. The interplay of variant, size, and task type in Arabic pre-trained language models. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, Kyiv, Ukraine (Online). Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Md Tawkat Islam Khondaker, Abdul Waheed, El Moatez Billah Nagoudi, and Muhammad Abdul-Mageed. 2023. Gptaraeval: A comprehensive evaluation of chatgpt on arabic nlp.

Viet Dac Lai, Nghia Trung Ngo, Amir Pouran Ben Veyseh, Hieu Man, Franck Dernoncourt, Trung Bui, and Thien Huu Nguyen. 2023. Chatgpt beyond english: Towards a comprehensive evaluation of large language models in multilingual learning.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: A python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 2356–2362, New York, NY, USA. Association for Computing Machinery.

Hussein Mozannar, Elie Maamary, Karl El Hajal, and Hazem Hajj. 2019. Neural Arabic question answering. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 108–118, Florence, Italy. Association for Computational Linguistics.

OpenAI. 2023. `https://chat.openai.com`.

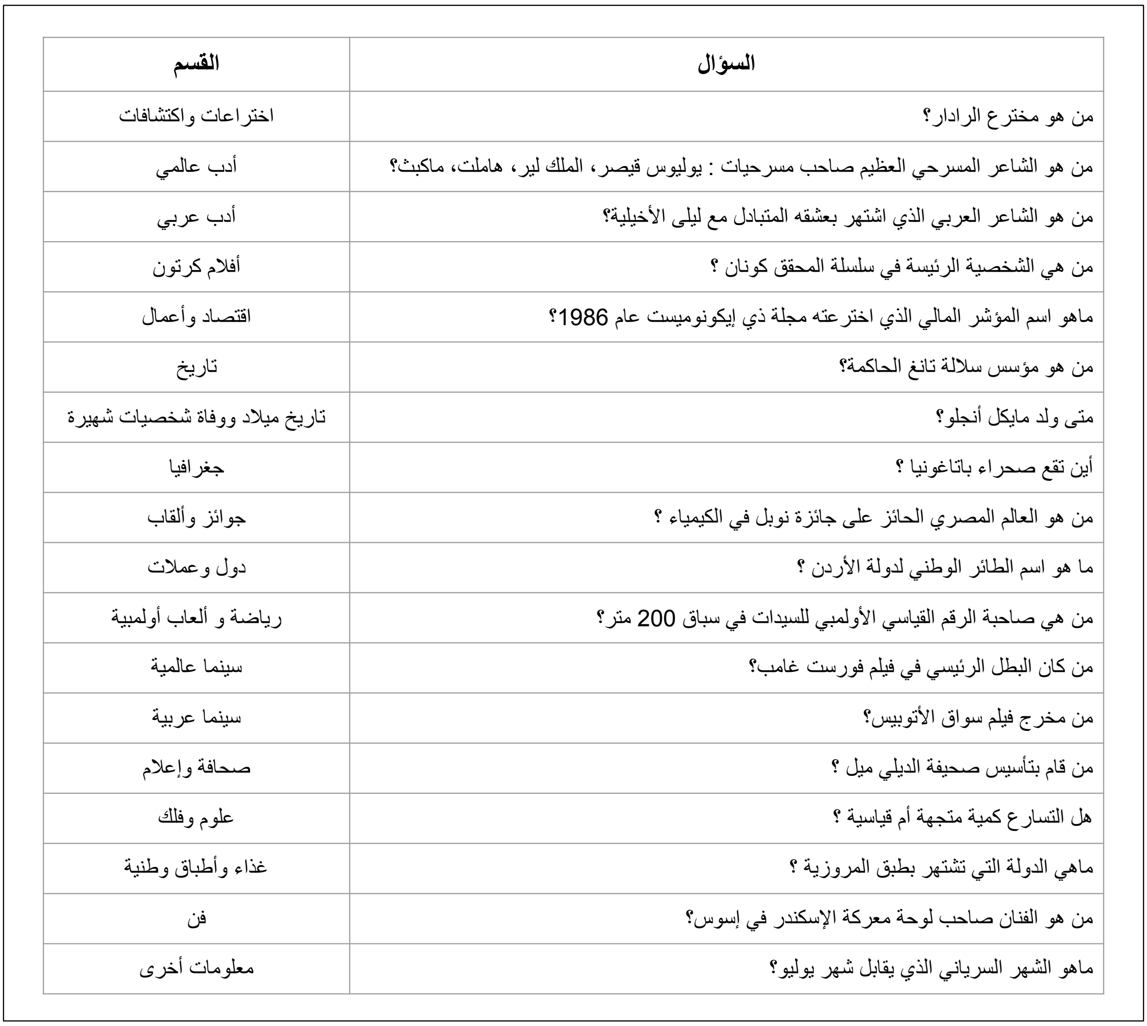
Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

A Appendix

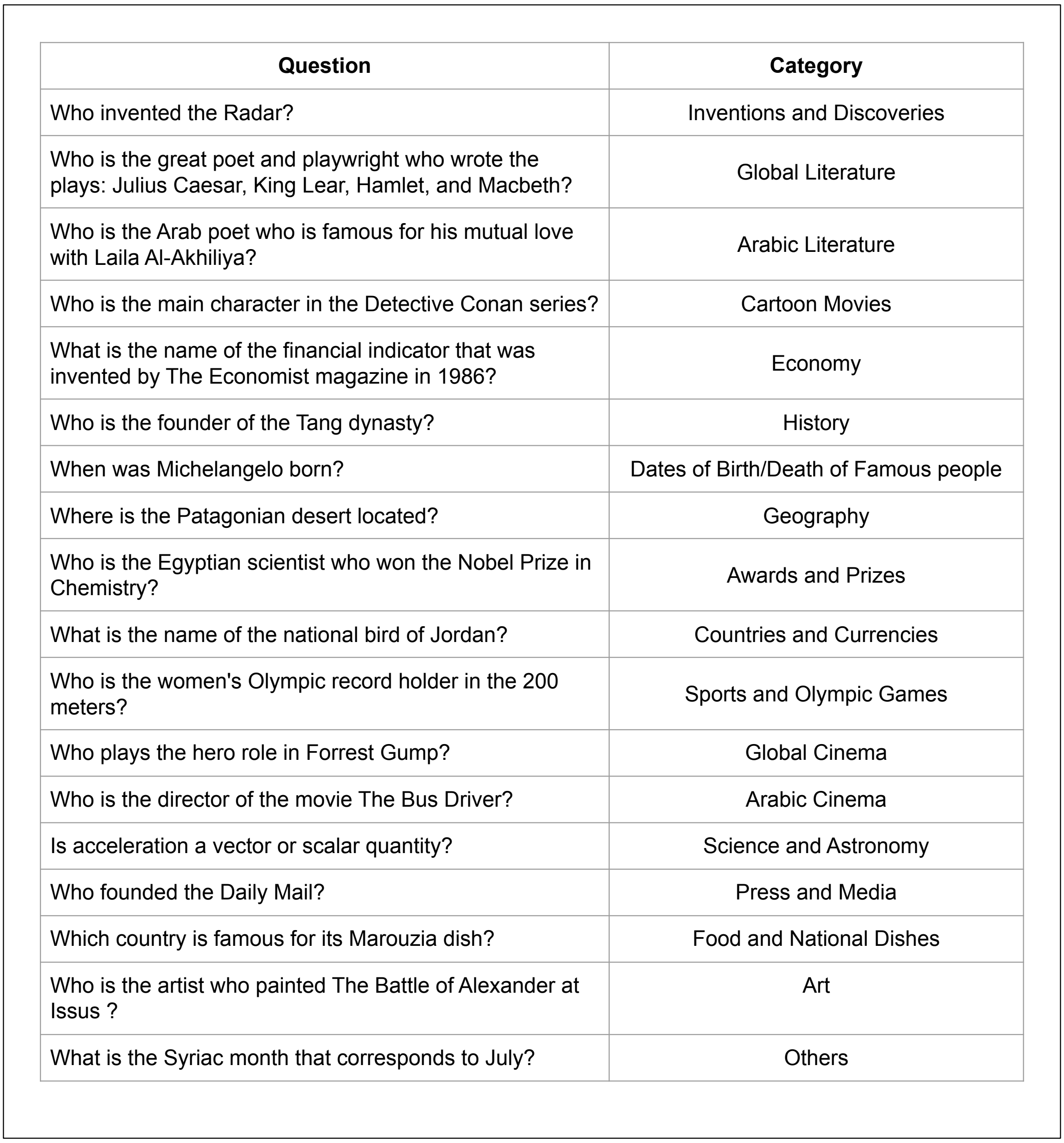
Topic	#Questions
Inventions and Discoveries (اختراعات واكتشافات)	231
Global Literature (أدب و روايات عالمية)	455
Arabic Literature (أدب ولغة عربية)	469
Cartoon Movies and Manga (أفلام كرتون وقصص مصورة)	284
Economy and Business (اقتصاد وأعمال)	210
History (تاريخ)	1645
Dates of Birth/Death of Famous people (تاريخ ميلاد ووفاة شخصيات شهيرة)	1274
Geography (جغرافيا)	1714
Awards and Prizes (جوائز والقاب)	355
Countries and Currencies (دول وعملات)	339
Sports and Olympic Games (رياضة والعاب اولمبية)	740
Global Cinema (سينما وأفلام عالمية)	107
Arabic Cinema (سينما وأفلام عربية)	409
Press and Media (صحافة وإعلام)	141
Science and Astronomy (علوم وفلك)	955
Food and National Dishes (غذاء واطباق وطنيه)	76
Art (فن)	133
Others (معلومات أخرى)	508
Total	10,045

Table 6: Topics distribution in ArTrivia Dataset. To have an accurate topics distribution of our dataset, we manually annotate the topic category for each question.



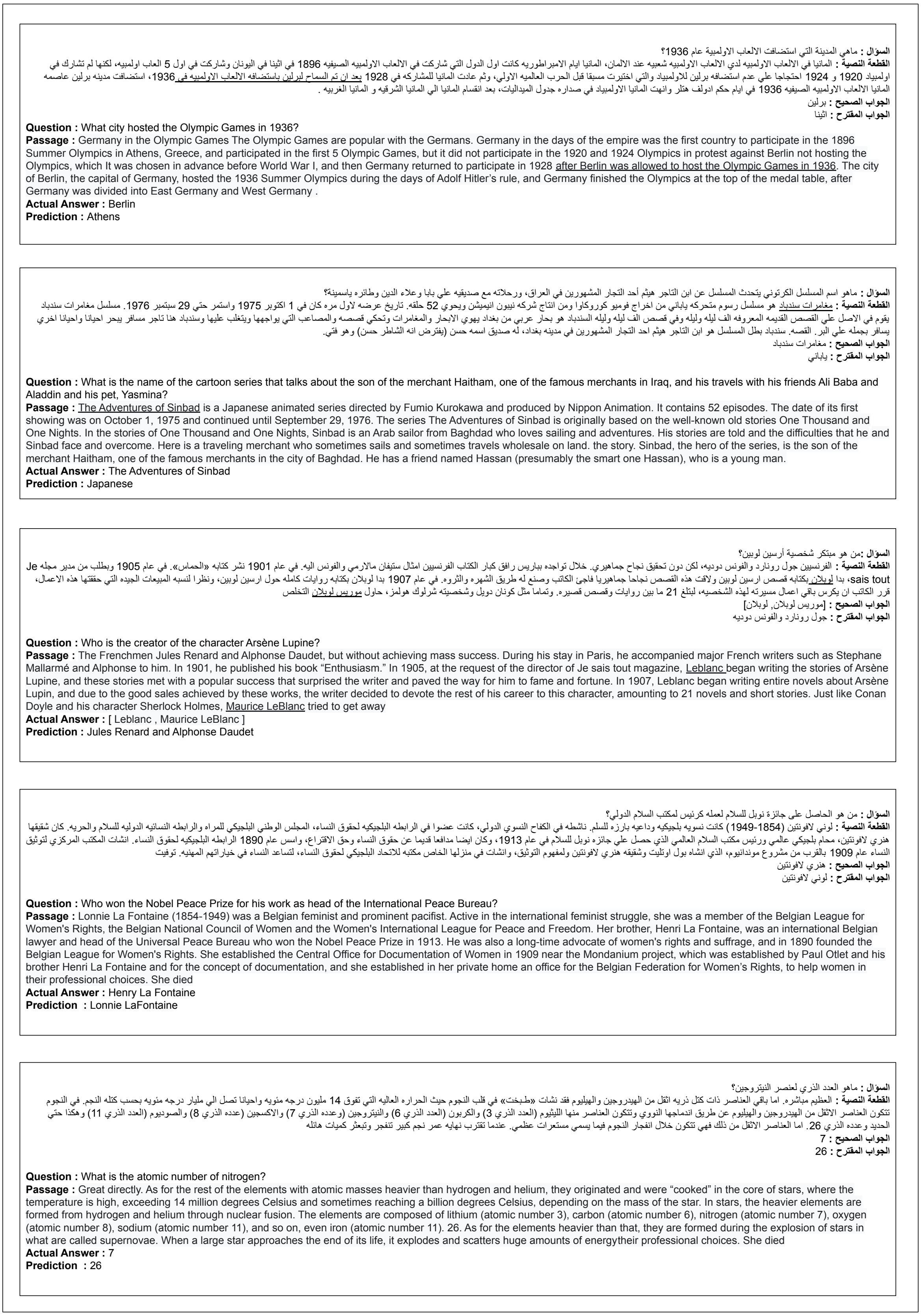
السؤال	القسم
من هو مخترع الرادار؟	اختراعات واكتشافات
من هو الشاعر المسرحي العظيم صاحب مسرحيات : يوليوس قيصر، الملك لير، هاملت، ماكبث؟	أدب عالمي
من هو الشاعر العربي الذي اشتهر بعشقه المتبادل مع ليلى الأخيلية؟	أدب عربي
من هي الشخصية الرئيسة في سلسلة المحقق كونان ؟	أفلام كرتون
ماهو اسم المؤشر المالي الذي اخترعته مجلة ذي إيكونوميست عام 1986؟	اقتصاد وأعمال
من هو مؤسس سلالة تانغ الحاكمة؟	تاريخ
متى ولد مايكل أنجلو؟	تاريخ ميلاد ووفاة شخصيات شهيرة
أين تقع صحراء باتاغونيا ؟	جغرافيا
من هو العالم المصري الحائز على جائزة نوبل في الكيمياء ؟	جوائز وألقاب
ما هو اسم الطائر الوطني لدولة الأردن ؟	دول وعملات
من هي صاحبة الرقم القياسي الأولمبي للسيدات في سباق 200 متر؟	رياضة و ألعاب أولمبية
من كان البطل الرئيسي في فيلم فورست غامب؟	سينما عالمية
من مخرج فيلم سواق الأتوبيس؟	سينما عربية
من قام بتأسيس صحيفة الديلي ميل ؟	صحافة وإعلام
هل التسارع كمية متجهة أم قياسية ؟	علوم وفلك
ماهي الدولة التي تشتهر بطبق المروزية ؟	غذاء وأطباق وطنية
من هو الفنان صاحب لوحة معركة الإسكندر في إسوس؟	فن
ماهو الشهر السرياني الذي يقابل شهر يوليو؟	معلومات أخرى

Figure 3: Examples of ArTrivia dataset from 18 different diverse topics.



Question	Category
Who invented the Radar?	Inventions and Discoveries
Who is the great poet and playwright who wrote the plays: Julius Caesar, King Lear, Hamlet, and Macbeth?	Global Literature
Who is the Arab poet who is famous for his mutual love with Laila Al-Akhiliya?	Arabic Literature
Who is the main character in the Detective Conan series?	Cartoon Movies
What is the name of the financial indicator that was invented by The Economist magazine in 1986?	Economy
Who is the founder of the Tang dynasty?	History
When was Michelangelo born?	Dates of Birth/Death of Famous people
Where is the Patagonian desert located?	Geography
Who is the Egyptian scientist who won the Nobel Prize in Chemistry?	Awards and Prizes
What is the name of the national bird of Jordan?	Countries and Currencies
Who is the women's Olympic record holder in the 200 meters?	Sports and Olympic Games
Who plays the hero role in Forrest Gump?	Global Cinema
Who is the director of the movie The Bus Driver?	Arabic Cinema
Is acceleration a vector or scalar quantity?	Science and Astronomy
Who founded the Daily Mail?	Press and Media
Which country is famous for its Marouzia dish?	Food and National Dishes
Who is the artist who painted The Battle of Alexander at Issus ?	Art
What is the Syriac month that corresponds to July?	Others

Figure 4: Examples of ArTrivia dataset from 18 different diverse topics. This is the English Translation of Figure 3.



السؤال : ماهي المدينة التي استضافت الالعاب الاولمبية عام 1936؟
القطعة النصية : المانيا في الالعاب الاولمبيه لدي الالعاب الاولمبيه شعبيه عند الالمان، المانيا ايام الامبراطوريه كانت اول الدول التي شاركت في الالعاب الاولمبيه الصيفيه 1896 في اثينا في اليونان وشاركت في اول 5 العاب اولمبيه، لكنها لم تشارك في اولمبياد 1920 و 1924 احتجاجا علي عدم استضافه برلين للاولمبياد والتي اختيرت مسبقا قبل الحرب العالميه الاولي، وثم عادت المانيا للمشاركه في 1928 بعد ان تم السماح لبرلين باستضافه الالعاب الاولمبيه في 1936، استضافت مدينه برلين عاصمه المانيا الالعاب الاولمبيه الصيفيه 1936 في ايام حكم ادولف هتلر وانهت المانيا الاولمبياد في صداره جدول الميداليات، بعد انقسام المانيا الي المانيا الشرقيه و المانيا الغربيه .
الجواب الصحيح : برلين
الجواب المقترح : اثينا

Question : What city hosted the Olympic Games in 1936?
Passage : Germany in the Olympic Games The Olympic Games are popular with the Germans. Germany in the days of the empire was the first country to participate in the 1896 Summer Olympics in Athens, Greece, and participated in the first 5 Olympic Games, but it did not participate in the 1920 and 1924 Olympics in protest against Berlin not hosting the Olympics, which It was chosen in advance before World War I, and then Germany returned to participate in 1928 after Berlin was allowed to host the Olympic Games in 1936. The city of Berlin, the capital of Germany, hosted the 1936 Summer Olympics during the days of Adolf Hitler's rule, and Germany finished the Olympics at the top of the medal table, after Germany was divided into East Germany and West Germany .
Actual Answer : Berlin
Prediction : Athens

السؤال : ماهو اسم المسلسل الكرتوني يتحدث المسلسل عن ابن التاجر هيثم أحد التجار المشهورين في العراق، ورحلاته مع صديقيه علي بابا وعلاء الدين وطائره ياسمينة؟
القطعة النصية : مغامرات سندباد هو مسلسل رسوم متحركه ياباني من اخراج فوميو كوروكاوا ومن انتاج شركه نيبون انيميشن ويحوي 52 حلقه. تاريخ عرضه لاول مره كان في 1 اكتوبر 1975 واستمر حتي 29 سبتمبر 1976. مسلسل مغامرات سندباد يقوم في الاصل علي القصص القديمه المعروفه الف ليله وليله وفي قصص الف ليله وليله السندباد هو بحار عربي من بغداد يهوي الابحار والمغامرات وتحكي قصصه والمصاعب التي يواجهها ويتغلب عليها وسندباد هنا تاجر مسافر يبحر احيانا واحيانا اخري يسافر بجمله علي البر. القصه. سندباد بطل المسلسل هو ابن التاجر هيثم احد التجار المشهورين في مدينه بغداد، له صديق اسمه حسن (يفترض انه الشاطر حسن) وهو فتي.
الجواب الصحيح : مغامرات سندباد
الجواب المقترح : ياباني

Question : What is the name of the cartoon series that talks about the son of the merchant Haitham, one of the famous merchants in Iraq, and his travels with his friends Ali Baba and Aladdin and his pet, Yasmina?
Passage : The Adventures of Sinbad is a Japanese animated series directed by Fumio Kurokawa and produced by Nippon Animation. It contains 52 episodes. The date of its first showing was on October 1, 1975 and continued until September 29, 1976. The series The Adventures of Sinbad is originally based on the well-known old stories One Thousand and One Nights. In the stories of One Thousand and One Nights, Sinbad is an Arab sailor from Baghdad who loves sailing and adventures. His stories are told and the difficulties that he and Sinbad face and overcome. Here is a traveling merchant who sometimes sails and sometimes travels wholesale on land. the story. Sinbad, the hero of the series, is the son of the merchant Haitham, one of the famous merchants in the city of Baghdad. He has a friend named Hassan (presumably the smart one Hassan), who is a young man.
Actual Answer : The Adventures of Sinbad
Prediction : Japanese

السؤال :من هو مبتكر شخصية أرسين لوبين؟
القطعة النصية : الفرنسيين جول رونارد والفونس دوديه، لكن دون تحقيق نجاح جماهيري. خلال تواجده بباريس رافق كبار الكتاب الفرنسيين امثال ستيفان مالارمي والفونس اليه. في عام 1901 نشر كتابه «الحماس». في عام 1905 وبطلب من مدير مجله Je sais tout، بدا لوبلان بكتابه قصص ارسين لوبين ولاقت هذه القصص نجاحا جماهيريا فاجئ الكاتب وصنع له طريق الشهره والثروه. في عام 1907 بدا لوبلان بكتابه روايات كامله حول ارسين لوبين، ونظرا لنسبه المبيعات الجيده التي حققتها هذه الاعمال، قرر الكاتب ان يكرس باقي اعمال مسيرته لهذه الشخصيه، لتبلغ 21 ما بين روايات وقصص قصيره. وتماما مثل كونان دويل وشخصيته شرلوك هولمز، حاول موريس لوبلان التخلص
الجواب الصحيح : [موريس لوبلان, لوبلان]
الجواب المقترح : جول رونارد والفونس دوديه

Question : Who is the creator of the character Arsène Lupine?
Passage : The Frenchmen Jules Renard and Alphonse Daudet, but without achieving mass success. During his stay in Paris, he accompanied major French writers such as Stephane Mallarmé and Alphonse to him. In 1901, he published his book "Enthusiasm." In 1905, at the request of the director of Je sais tout magazine, Leblanc began writing the stories of Arsène Lupine, and these stories met with a popular success that surprised the writer and paved the way for him to fame and fortune. In 1907, Leblanc began writing entire novels about Arsène Lupin, and due to the good sales achieved by these works, the writer decided to devote the rest of his career to this character, amounting to 21 novels and short stories. Just like Conan Doyle and his character Sherlock Holmes, Maurice LeBlanc tried to get away
Actual Answer : [Leblanc , Maurice LeBlanc]
Prediction : Jules Renard and Alphonse Daudet

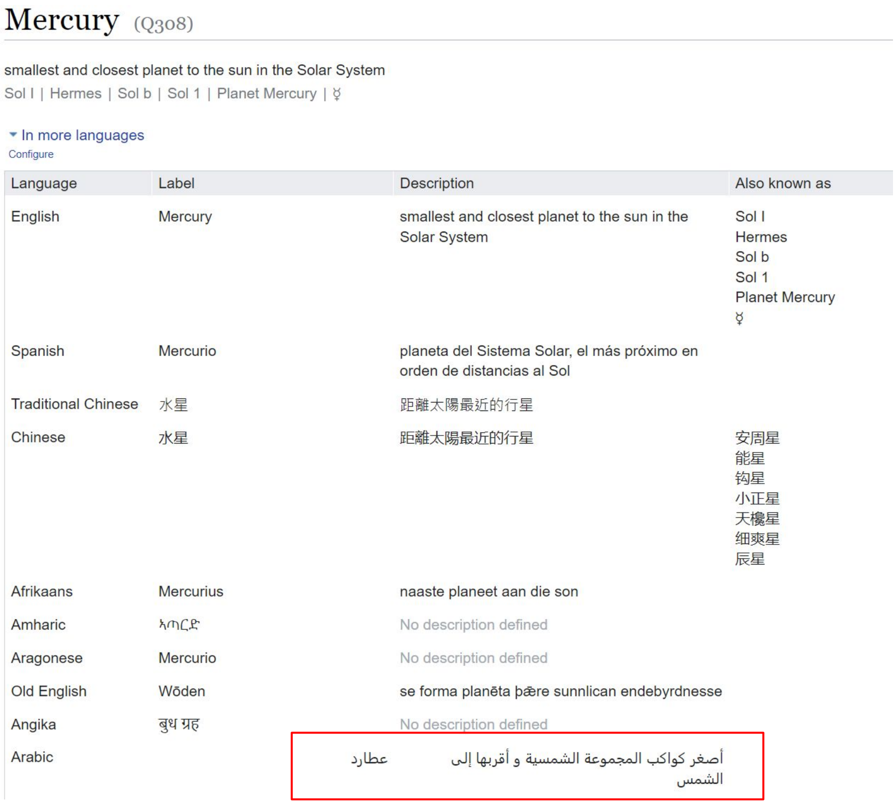
السؤال : من هو الحاصل على جائزة نوبل للسلام لعمله كرئيس لمكتب السلام الدولي؟
القطعة النصية : لوني لافونتين (1854-1949) كانت نسويه بلجيكيه وداعيه بارزه للسلم. ناشطه في الكفاح النسوي الدولي، كانت عضوا في الرابطه البلجيكيه لحقوق النساء، المجلس الوطني البلجيكي للمراه والرابطه النسائيه الدوليه للسلام والحريه. كان شقيقها هنري لافونتين، محام بلجيكي عالمي ورئيس مكتب السلام العالمي الذي حصل علي جائزه نوبل للسلام في عام 1913، وكان ايضا مدافعا قديما عن حقوق النساء وحق الاقتراع، واسس عام 1890 الرابطه البلجيكيه لحقوق النساء. انشات المكتب المركزي لتوثيق النساء عام 1909 بالقرب من مشروع موندانيوم، الذي انشاه بول اوتليت وشقيقه هنري لافونتين ولمفهوم التوثيق، وانشات في منزلها الخاص مكتبه للاتحاد البلجيكي لحقوق النساء، لتساعد النساء في خياراتهم المهنيه. توفيت
الجواب الصحيح : هنري لافونتين
الجواب المقترح : لوني لافونتين

Question : Who won the Nobel Peace Prize for his work as head of the International Peace Bureau?
Passage : Lonnie La Fontaine (1854-1949) was a Belgian feminist and prominent pacifist. Active in the international feminist struggle, she was a member of the Belgian League for Women's Rights, the Belgian National Council of Women and the Women's International League for Peace and Freedom. Her brother, Henri La Fontaine, was an international Belgian lawyer and head of the Universal Peace Bureau who won the Nobel Peace Prize in 1913. He was also a long-time advocate of women's rights and suffrage, and in 1890 founded the Belgian League for Women's Rights. She established the Central Office for Documentation of Women in 1909 near the Mondanium project, which was established by Paul Otlet and his brother Henri La Fontaine and for the concept of documentation, and she established in her private home an office for the Belgian Federation for Women's Rights, to help women in their professional choices. She died
Actual Answer : Henry La Fontaine
Prediction : Lonnie LaFontaine

السؤال : ماهو العدد الذري لعنصر النيتروجين؟
القطعة النصية : العظيم مباشره. اما باقي العناصر ذات كتل ذريه اثقل من الهيدروجين والهيليوم فقد نشات «طبخت» في قلب النجوم حيث الحراره العاليه التي تفوق 14 مليون درجه مئويه واحيانا تصل الي مليار درجه مئويه بحسب كتله النجم. في النجوم تتكون العناصر الاثقل من الهيدروجين والهيليوم عن طريق اندماجها النووي وتتكون العناصر منها الليثيوم (العدد الذري 3) والكربون (العدد الذري 6) والنيتروجين (وعدده الذري 7) والاكسجين (عدده الذري 8) والصوديوم (العدد الذري 11) وهكذا حتي الحديد وعدده الذري 26. اما العناصر الاثقل من ذلك فهي تتكون خلال انفجار النجوم فيما يسمي مستعرات عظمي. عندما تقترب نهايه عمر نجم كبير تنفجر وتبعثر كميات هائله
الجواب الصحيح : 7
الجواب المقترح : 26

Question : What is the atomic number of nitrogen?
Passage : Great directly. As for the rest of the elements with atomic masses heavier than hydrogen and helium, they originated and were "cooked" in the core of stars, where the temperature is high, exceeding 14 million degrees Celsius and sometimes reaching a billion degrees Celsius, depending on the mass of the star. In stars, the heavier elements are formed from hydrogen and helium through nuclear fusion. The elements are composed of lithium (atomic number 3), carbon (atomic number 6), nitrogen (atomic number 7), oxygen (atomic number 8), sodium (atomic number 11), and so on, even iron (atomic number 11). 26. As for the elements heavier than that, they are formed during the explosion of stars in what are called supernovae. When a large star approaches the end of its life, it explodes and scatters huge amounts of energytheir professional choices. She died
Actual Answer : 7
Prediction : 26

Figure 5: Examples of TyDi-based AraELECTRA's incorrect predictions on the ArTrivia development dataset.



Mercury (Q308)

smallest and closest planet to the sun in the Solar System

Sol I | Hermes | Sol b | Sol 1 | Planet Mercury | ☿

Language	Label	Description	Also known as
English	Mercury	smallest and closest planet to the sun in the Solar System	Sol I Hermes Sol b Sol 1 Planet Mercury ☿
Spanish	Mercurio	planeta del Sistema Solar, el más próximo en orden de distancias al Sol	
Traditional Chinese	水星	距離太陽最近的行星	
Chinese	水星	距離太陽最近的行星	安周星 能星 钩星 小正星 天欃星 细爽星 辰星
Afrikaans	Mercurius	naaste planeet aan die son	
Amharic	አጣርድ	No description defined	
Aragonese	Mercurio	No description defined	
Old English	Wōden	se forma planēta þǣre sunnlican endebyrdnesse	
Angika	बुध ग्रह	No description defined	
Arabic	عطارد	أصغر كواكب المجموعة الشمسية و أقربها إلى الشمس	

Figure 6: The entity (Article Title) description in Wikipedia. This entity description can be accessed manually for any page in Arabic Wikipedia by clicking on tools (أدوات), then page information (معلومات الصفحة).

ArSarcasMoji Dataset: The Emoji Sentiment Roles in Arabic Ironic Contexts

Shatha Ali A. Hakami
Jazan University
Dept. of Computer Science
Saudi Arabia
sahakami@jazanu.edu.sa

Robert Hendley
University of Birmingham,
School of Computer Science
United Kingdom
r.j.hendley@cs.bham.ac.uk

Phillip Smith
University of Birmingham
School of Computer Science
United Kingdom
p.smith.7@cs.bham.ac.uk

Abstract

In digital communication, emoji are essential in decoding nuances such as irony, sarcasm, and humour. However, their incorporation in Arabic natural language processing (NLP) has been cautious because of the perceived complexities of the Arabic language. This paper introduces **ArSarcasMoji**, a dataset of 24,630 emoji-augmented texts, with 17. 5% that shows irony. Through our analysis, we highlight specific emoji patterns paired with sentiment roles that denote irony in Arabic texts. The research counters prevailing notions, emphasising the importance of emoji's role in understanding Arabic textual irony, and addresses their potential for accurate irony detection in Arabic digital content.

1 Introduction

Irony, sarcasm, and humour are intricate forms of expression that craft messages in nuanced, playful, or mock-serious tones. Although they might intersect in their application, each has a unique meaning and utility. Irony portrays situations or statements contrary to what is expected, manifesting verbally or situationally (Abrams and Harpham, 2009). **Sarcasm**, a specific facet of irony, employs language with a sharp, often bitter tone to mock or critique, often with exaggerated emphasis (Partridge, 1997). Within the Arabic social media sphere, sarcasm often serves as a tool for social commentary or political satire, acting as a medium to challenge authority and examine social norms and values (Zidjaly, 2017; Mohammed et al., 2020; Abu Farha and Magdy, 2022). **Humour**, while encompassing elements of irony, specifically denotes the ability to evoke amusement or laughter (Martin and Ford, 2018). It manifests itself in diverse formats, from jokes and puns to witty remarks. In Arabic digital communication, humour is praised for its prowess in communicating intricate emotions and fostering positive sentiments in a light-hearted and engaging manner (Banikalef et al., 2014; Alkhalifa et al., 2022).

In today's digital age, emoji have emerged as powerful tools in the linguistic landscape. These are small digital icons that are used to convey emotions or ideas. Although once dismissed as mere decorative elements, they are now acknowledged for their crucial role in amplifying and clarifying textual sentiments, moods, and intentions (Danesi, 2017; Cohn et al., 2019; Hakami et al., 2020). By providing much needed context, especially on platforms where vocal tonalities and facial cues are absent, emoji enrich the emotional depth of a message. They have become instrumental in detecting nuances such as sarcasm and humour, underscoring their importance in contemporary studies of natural language processing (NLP) (Rohanian et al., 2018; Hayati et al., 2019; Chiruzzo et al., 2020; Castro et al., 2018).

Although the importance of emoji in the deciphering of textual context is undeniable, there exists a contrasting trend in Arabic NLP. Given the inherent intricacies of the Arabic language, characterised by its rich morphological structures and multifaceted semantics, many researchers opt to exclude emoji when analysing irony. This practise stems from the belief that emoji could introduce an additional layer of complexity, potentially diverting the focus from the linguistic nuances unique to Arabic. As such, despite the global trend of integrating emoji into textual analysis, there is a cautious approach within Arabic NLP circles, underscoring the challenges and distinctiveness of the Arabic linguistic landscape.

In response to this observed gap in Arabic NLP research, this paper presents the community with a unique dataset (**ArSarcasMoji**) consisting of **24,630** short texts enriched with emoji (**4,320** are ironic and **20,310** are not) [1]. Our exploration goes beyond conventional analyses to illustrate the cru-

[1] Click here to download ArSarcasMoji.

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 208–217
December 7, 2023 ©2023 Association for Computational Linguistics

cial role these emoji play in discerning sarcasm and humour within the texts. Through careful analysis, we show that emoji are not just additional symbols, which goes against common beliefs. Instead, they often hold the key to unmasking the ironic intent behind a statement.

In this paper, we test this claim as follows. For a text to be labelled as ironic, it must feature a specific emoji pattern with distinct sentiment roles. Accordingly, our research focuses on three main questions:

- **Q1:** Which emoji patterns are indicative of irony in Arabic texts?
- **Q2:** In what manner do these emoji convey irony through their sentiment roles?
- **Q3:** How effectively does the synergy of these emoji patterns and their associated sentiment roles pinpoint irony?

Providing **ArSarcasMoji** dataset along with this analysis underscores the indispensable value of emoji in enhancing our understanding of textual irony in the Arabic informal social media language.

The remainder of this paper is organised as follows. Section 2 reviews related work; Section 3 presents the study methodology; Section 4 presents the analysis of the dataset. Finally, in Section 5 we draw conclusions from this work along with some recommendations for future work as well as highlight its limitations.

2 Related Work

Sarcasm and humour, two intricate linguistic phenomena, have garnered significant attention in the field of NLP. Their detection in textual data is paramount for improved sentiment analysis, better content recommendation, and the promotion of nuanced human-machine interactions.

In the fast-evolving domain of NLP, the challenge of irony detection in Arabic stands out, given the language's diverse dialects and rich linguistic intricacies. Sentiment analysis has long wrestled with this complexity, primarily due to the nuances of spotting indirect phrasing that often conveys meanings contrary to their overt expressions. Taking steps in this area, Abu Farha and Magdy (2020) introduced the ArSarcasm dataset. Derived from reannotating existing Arabic sentiment datasets, ArSarcasm features 10,547 tweets with 16% labelled sarcastic. The research highlighted the subjective challenges of sentiment annotation and the diminished efficacy of modern sentiment analysis systems when confronted with sarcasm. Furthermore, a BiLSTM-based model they developed for sarcasm detection achieved an F1 score of 0.46, underscoring the task's complexity. However, a notable limitation in their study was the neglect of the roles of emoji, which often play a pivotal role in conveying and deciphering sarcasm in textual communications.

In the following step, a research study by Al-Mazrua et al. (2022) unveiled the *Sa'7r*, a Saudi-specific irony dataset derived from 19,810 tweets (8,089 of which were labelled ironic). In their endeavour, they trained an array of classifiers, encompassing machine learning models like KNN, LR, SVM, and NB, as well as deep learning contenders such as BiLSTM and AraBERT. Among these, the SVM algorithm emerged as the most proficient in traditional techniques, boasting an accuracy of 0.68, while in the deep learning arena, AraBERT led with an impressive 0.71 accuracy. This establishes AraBERT as a primary tool for discerning irony within the nuances of Saudi dialects. However, the study did not highlight the feature of emoji in this task.

In parallel, Alkhalifa et al. (2022) paved the way with a distinctive dataset of 10,039 tweets, covering various Arabic dialects and Modern Standard Arabic, meticulously annotated for humourous and non-humourous content. With rigorous pre-processing steps, including Arabic normalisation and the pruning of unrelated text, the CAMeLBERT-DA model achieved an accuracy of 72.11%. Despite that, a critical gap was the dataset's exclusion of emoji.

In today's digital era, emoji are instrumental in relaying sentiments, particularly in the realm of irony. This oversight in the already-existing Arabic datasets might hint at the datasets' potential limitation in truly capturing the intricacies of contemporary Arabic sarcasm and humour, marking an area ripe for further research and development. Hayati et al. (2019) explored the central role of emoji in irony detection in English texts. Observing the under-representation of emoji in ironic tweets in existing English datasets, they proposed an automated pipeline for more balanced data. Their findings highlighted how emoji can transform text sentiment, converting straightforward statements into ironic ones. They augmented the datasets, making the models attuned to text and emoji signals, and,

#	Dataset Source	Reference	Initial Emoji-Texts	ArSarcasMoji
1	AraSenCorpus	(Al-Laith et al., 2021)	280,739	20,657
2	ArCovid_19	(Haouari et al., 2021)	45,440	5
3	ASAD	(Alharbi et al., 2020)	11,969	1
4	TEAD	(Abdellaoui and Zrigui, 2018)	11,950	1,343
5	ArSAS	(Elmadany et al., 2018)	6,070	1,113
6	ATSAD	(Abu Kwaik et al., 2020)	3,775	666
7	Kawarith	(Alharbi and Lee, 2021)	2,975	208
8	ArSarcasm	(Abu Farha and Magdy, 2020)	1,093	19
9	SS2030	(Alyami and Olatunji, 2020)	1,061	244
10	SemEval_2018_Task1_Task2	(Mohammad et al., 2018) (Barbieri et al., 2018)	668	196
11	DART	(Alsarsour et al., 2018)	599	89
12	ArSenTD-Lev	(Baly et al., 2018)	389	51
13	SemEval_2017_Task4	(Rosenthal et al., 2017)	263	20
14	L-HSAB Dataset	(Mulki et al., 2019)	65	8
15	SyriaTweets	(Salameh et al., 2015)	64	10
Total			**367,120**	**24,630**

Table 1: ArSarcasMoji dataset resources.

when analysing the SemEval 2018 dataset (Mohammad et al., 2018), observed distinct patterns of emoji usage between ironic and non-ironic tweets, highlighting the vital role of emoji in sentiment interpretation.

Hakami et al. (2022b) explored similar emoji behaviour in Arabic texts. Their research posited an innovative approach towards understanding the sentiment implications of emoji, particularly within Arabic textual frameworks. The findings reaffirm that an emoji's sentiment role can oscillate among three paradigms: negative, neutral, or positive. Specifically, an emoji can function as an *Emphasizer*, *Indicator*, *Mitigator*, *Reverser*, or *Trigger* of negative or positive sentiments within a textual context. There was also an intriguing proposition that certain emoji can exert a *Neutral Effect* essentially leaving the text with a neutral sentiment. In distilling the gamut of roles that emoji can play in sentiment analysis, this research provided invaluable insights for scholars seeking to understand the nuanced interplay of emoji and text.

Expanding on the emoji-sentiment-roles model by Hakami et al. (2022b), we formulated an irony classification technique that allowed the creation of the **ArSarcasMoji** dataset. Details are as follows.

3 Methodology

3.1 Data Resources and Pre-processing

To create our comprehensive dataset, we began by amalgamating data from 15 distinct Arabic social media datasets, as referenced in Table 1. This aggregation resulted in **367,120** emoji-inclusive tweets from Twitter, which we designated as the Emoji-Text dataset. From this rich collection, we derived a parallel Plain-Text dataset by extracting the same tweets while removing their emojis. Following the cleaning and normalisation procedures outlined in (Hakami et al., 2021), both data sets were subjected to sentiment annotation using five Arabic sentiment classifiers: Mazajak (Abu Farha and Magdy, 2019), CAMeL-Tools (AraBERT and mBERT) (Obeid et al., 2020), ASAD (Hassan et al., 2021), and the lexicon-based method presented by (Hakami et al., 2022b). Only tweets with sentiment labels that garnered unanimous agreement across all classifiers were preserved, narrowing our data set to concise **24,630** tweets. Conclusively, we applied the methodology delineated by Hakami et al. (2022b) to pinpoint the sentimental roles of different emoji patterns present within these concise tweets.

3.2 Irony Classification Model

Our irony detection model within textual content is based on two primary features: the presence of a distinct emoji pattern (referred to as the "*ironic emoji pattern*") and the sentiment role this pattern assumes in the text. They are detailed below.

3.2.1 Ironic Emoji Patterns Identification

In addressing the first research question of this study on the identification of ironic emoji patterns, we primarily anchored our approach on a curated set of emoji, termed as '*seed*' emoji. These emoji were selected due to their inherent ironic characteristics, previously validated as markers of irony

Sentiment	Emoji													
Positive	😂	🙂	😅	😁	🤣	😉	😎	😬	😜	😹	😆	😝	😛	🤪
Negative	🙃	😒	🤔	😑	😏	🤦	🙄	🤥	🤨					

Table 2: Emoji seeds for ironic patterns extraction.

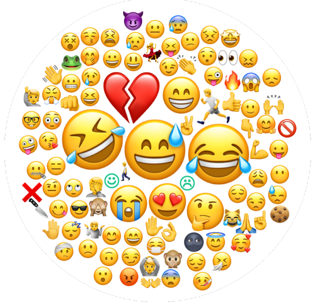
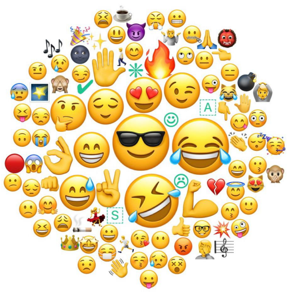
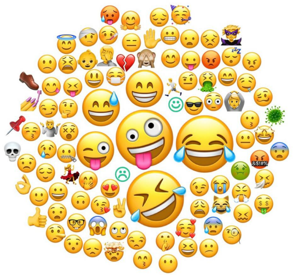
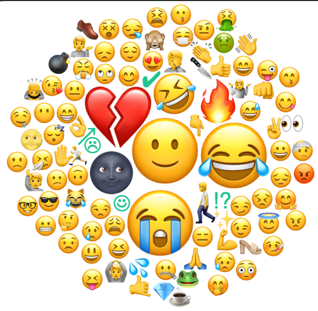
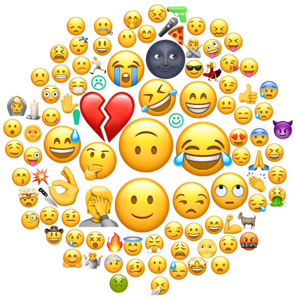
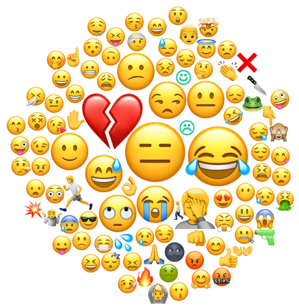
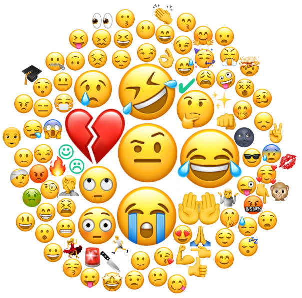
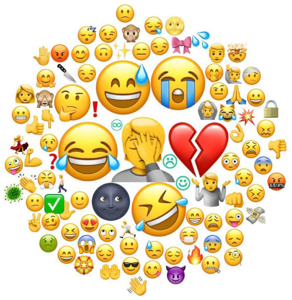
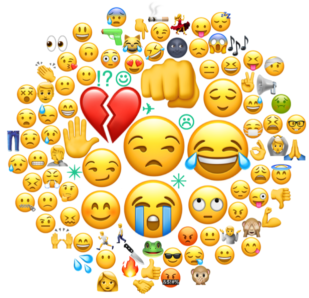
 seed: (😅)	 seed: (😎)	 seed: (🤪)
 seed: (🙂)	 seed: (🙃)	 seed: (😑)
 seed: (🤨)	 seed: (🤦)	 seed: (😒)

Table 3: The most 100 correlated emoji to nine of the seed ironic emoji.

in studies such as (Wiguna et al., 2021; Weissman and Tanner, 2018; Wang, 2022). Specifically, we curated **23** emoji, both positive and negative according to Hakami et al. (2021) and Hakami et al. (2022a), presented in Table 2, hypothesising their deliberate use in Arabic contexts as irony markers (Mahzari, 2017; Abbas and Ubeid, 2021; Alshboul and Rababah, 2021; Etman and Elkareh, 2021). Merging these seed emoji with a selection from the original 1,272 emoji in our initially collected dataset (excluding all *Flags*, select *Natures*, and the majority of *Hearts* emoji), we identified **15,101** distinct ironic emoji patterns. This ensemble comprises **8,990** positive, **543** neutral, and **5,568** negative patterns, each possessing various sentiment roles with an ironic tilt. To improve clarity, we illustrate the co-occurrence of the top 100 emoji with nine foundational ironic seed emoji in Table 3, demonstrating the diverse sentiment-laden emoji employed in Arabic ironic scenarios. We have made these patterns openly accessible for future research purposes[2]

[2] Click here to download Arabic ironic emoji patterns.

3.2.2 Sarcasm/Humour Classifications

Addressing the second research question of the study, we utilised the sentiment roles of the ironic emoji patterns within the texts to determine their ironic undertones.

To identify sarcastic texts, the classification was based on the existence of certain predetermined ironic emoji patterns tied to negative sentiment roles in a rule-based manner. This was formulated as follows:

$$\begin{array}{c}\textbf{Plain}\\\textbf{Text}\end{array} + \left\{\begin{array}{c}\text{Negativity Emphasizer Ironic Emoji}\\\text{Negativity Indicator Ironic Emoji}\\\text{Negativity Mitigator Ironic Emoji}\\\text{Negativity Trigger Ironic Emoji}\\\text{Negative Reverser Ironic Emoji}\\\text{No-Effect (Negative | Neutral Ironic Emoji)}\end{array}\right\} = \begin{array}{c}\textbf{Sarcastic}\\\textbf{Text}\end{array}$$

On the other hand, for identifying humorous texts, the classification centred on the existence of certain predefined ironic emoji patterns that played positive sentiment roles. This was formulated as follows:

$$\begin{array}{c}\textbf{Plain}\\\textbf{Text}\end{array} + \left\{\begin{array}{c}\text{Positivity Emphasizer Ironic Emoji}\\\text{Positivity Indicator Ironic Emoji}\\\text{Positivity Mitigator Ironic Emoji}\\\text{Positivity Trigger Ironic Emoji}\\\text{Positive Reverser Ironic Emoji}\\\text{No-Effect (Positive Ironic Emoji)}\end{array}\right\} = \begin{array}{c}\textbf{Humorous}\\\textbf{Text}\end{array}$$

3.2.3 Model Evaluation

Addressing the third research question of the study, to evaluate this irony classification model, we utilized two publicly available Arabic ironic datasets: Sa'7r (AlMazrua et al., 2022) and ArSarcasm (Abu Farha and Magdy, 2020). From the merged datasets, we retained only those texts containing emoji, amounting to 6,738 texts. Of these, 2,727 were labelled as ironic/sarcastic, while 4,011 were not. To establish a balanced sample, we arbitrarily chose 1,000 texts from both ironic and non-ironic categories, resulting in a total of **2,000** evaluation texts. Using the predefined ironic emoji patterns, we categorized this sample into ironic and non-ironic groups. We then gauged the efficacy of this classification by comparing its resulting labels with the benchmark's labels. Impressively, our analysis reported an **accuracy** of **0.91** when juxtaposed with the benchmark set. The Cohen's (κ) (McHugh, 2012) agreement between our classifications and the benchmark annotations was also substantial, scoring a **0.83**.

While the pre-defined emoji patterns clearly indicated irony in the texts, differentiating between sarcasm and humour was not clear. To address this ambiguity, we first identified the sentiment roles of these emoji patterns, using the machine's fusion sentiment annotation technique, mentioned in the data pre-processing step above, for both texts with and without emoji. Based on the sentiments roles plied by these emoji, we labelled the texts as either *sarcastic* or *humorous*. To validate these labels, we hand-annotated the sentiment of **600** representative texts, split equally between the humorous and sarcastic categories. Our irony classification model's analysis of this subset yielded impressive consistency: a Cohen's κ coefficient of **0.97** for humorous (positive) texts and **0.95** for sarcastic (negative) texts. These results underscore our model's capability to discern between humorous and sarcastic undertones in ironic Arabic texts.

Consequently, we employed this irony classification model to categorize the 24,630 tweets in the ArSarcasMoji dataset into sarcastic, humorous, and not_ironic categories.

4 ArSarcasMoji Dataset Analysis

To the best of our knowledge, ArSarcasMoji is the premier dataset in Arabic dedicated to the analysis of emoji, with a particular focus on ironic behaviours. This dataset encompasses **24,630** Arabic texts with emoji, as well as parallel versions of these texts devoid of emoji. Both categories of texts—with and without emoji—have undergone sentiment annotation. Moreover, the sentiment roles of emoji patterns and the ironic demeanour of each emoji-inclusive text are clearly defined. Delving into the ironic nature of emoji use, the dataset boasts **4,320** ironic texts (**3,573** classified as *sarcastic* and **747** as *humorous*), in contrast to **20,310** *non-ironic* texts. The sentiment roles of the emoji patterns, juxtaposed with their irony-induced effects, are visualized in Figure 1. For a more illustrative insight, Figure 2 presents samples of both sarcastic and humorous texts. A more in-depth exploration of the dataset is provided subsequently.

4.1 Positive Texts

Of the dataset, 54.64% represents texts with a positive sentiment. Within this positive cohort, emoji play pivotal roles, notably:

- *Positivity Emphasizer*: A substantial 53.26% of emoji in these entries function as non-verbal amplifiers of the text's positive sentiment. From this pool, 5.39% are indicative of humour. A tangible example showcasing an

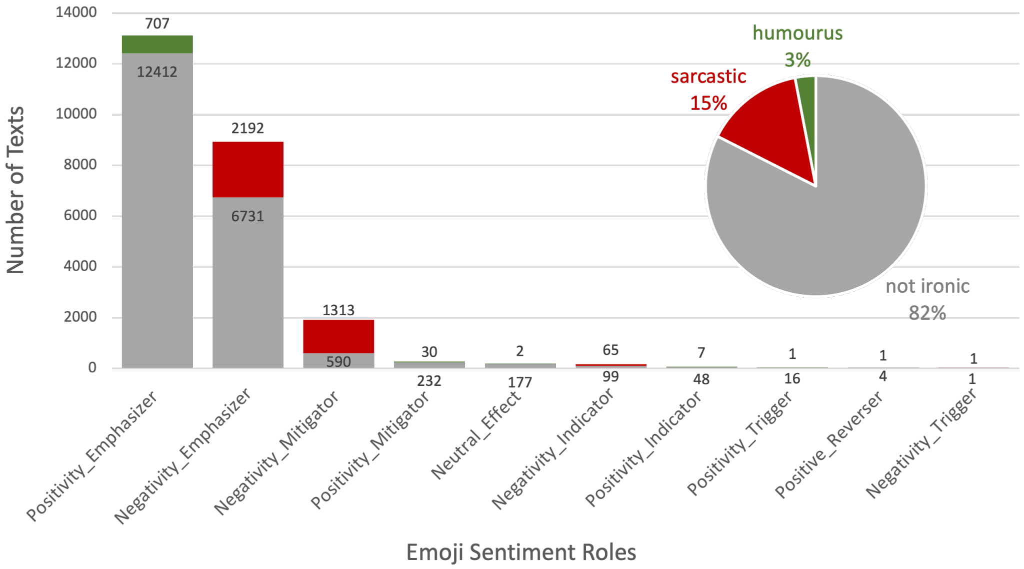
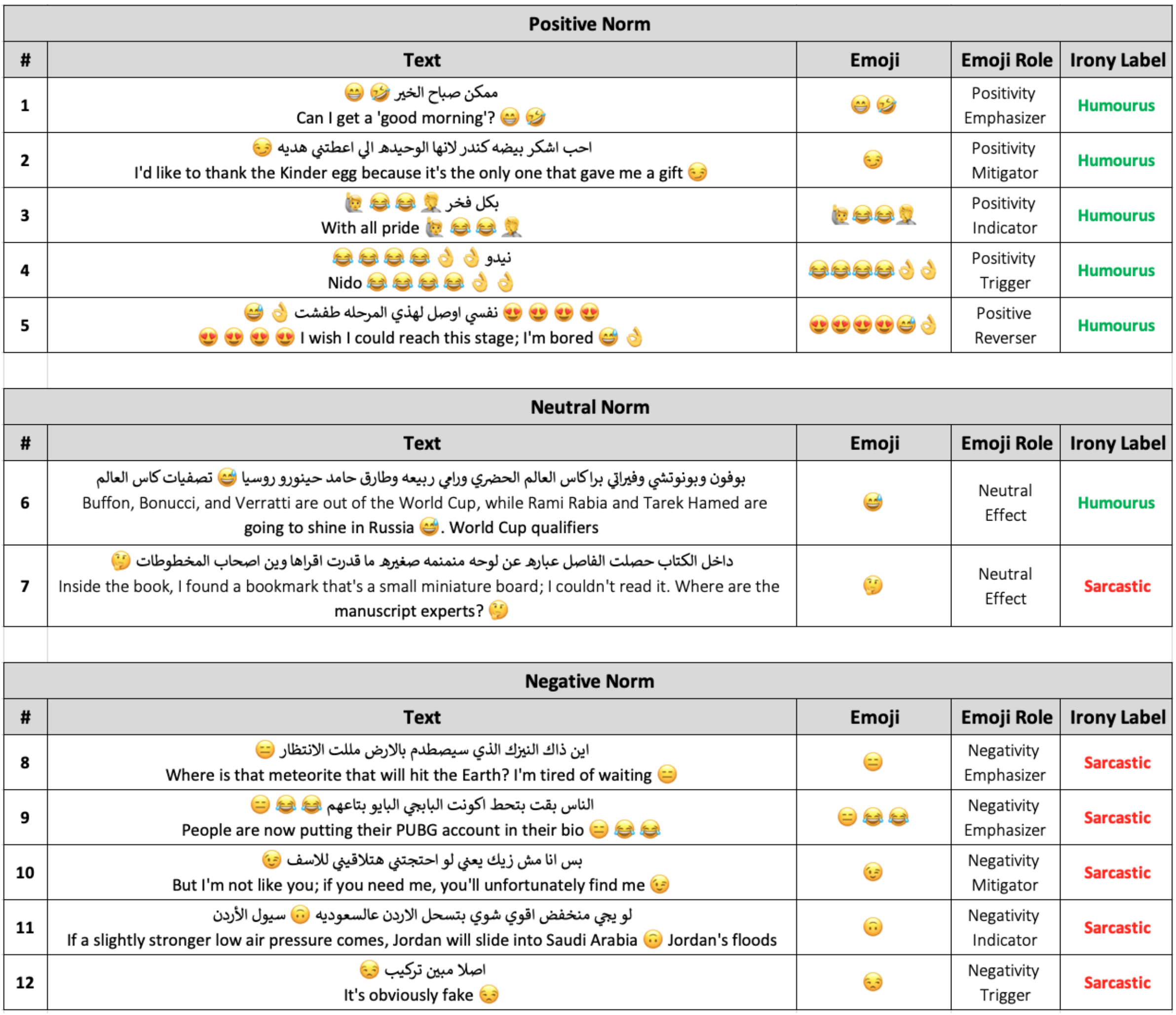


Figure 1: The distribution of *sarcastic* and *humorous* texts along with their emoji sentiment roles in the ArSarcasmoji Dataset.



Positive Norm				
#	**Text**	**Emoji**	**Emoji Role**	**Irony Label**
1	ممكن صباح الخير 🤣 😁 Can I get a 'good morning'? 😁 🤣	😁🤣	Positivity Emphasizer	Humourus
2	احب اشكر بيضه كندر لانها الوحيده الي اعطتني هديه 😏 I'd like to thank the Kinder egg because it's the only one that gave me a gift 😏	😏	Positivity Mitigator	Humourus
3	بكل فخر 🤦‍♂️ 😂 😂 🙋‍♂️ With all pride 🙋‍♂️ 😂 😂 🤦‍♂️	🙋‍♂️😂😂🤦‍♂️	Positivity Indicator	Humourus
4	نيدو 👌 👌 😂 😂 😂 😂 Nido 😂 😂 😂 😂 👌 👌	😂😂😂😂👌👌	Positivity Trigger	Humourus
5	😍 😍 😍 😍 نفسي اوصل لهذي المرحله طفشت 👌 😅 😍 😍 😍 😍 I wish I could reach this stage; I'm bored 😅 👌	😍😍😍😍😅👌	Positive Reverser	Humourus

Neutral Norm				
#	**Text**	**Emoji**	**Emoji Role**	**Irony Label**
6	بوفون وبونوتشي وفيراتي براكاس العالم الحضري ورامي ربيعه وطارق حامد حينورو روسيا 😅 تصفيات كاس العالم Buffon, Bonucci, and Verratti are out of the World Cup, while Rami Rabia and Tarek Hamed are going to shine in Russia 😅. World Cup qualifiers	😅	Neutral Effect	Humourus
7	داخل الكتاب حصلت الفاصل عبارة عن لوحه منمنمه صغيره ما قدرت اقراها وين اصحاب المخطوطات 🤔 Inside the book, I found a bookmark that's a small miniature board; I couldn't read it. Where are the manuscript experts? 🤔	🤔	Neutral Effect	Sarcastic

Negative Norm				
#	**Text**	**Emoji**	**Emoji Role**	**Irony Label**
8	اين ذاك النيزك الذي سيصطدم بالارض مللت الانتظار 😑 Where is that meteorite that will hit the Earth? I'm tired of waiting 😑	😑	Negativity Emphasizer	Sarcastic
9	الناس بقت بتحط اكونت البابجي البايو بتاعهم 😂 😂 😑 People are now putting their PUBG account in their bio 😑 😂 😂	😑😂😂	Negativity Emphasizer	Sarcastic
10	بس انا مش زيك يعني لو احتجتني هتلاقيني للاسف 😉 But I'm not like you; if you need me, you'll unfortunately find me 😉	😉	Negativity Mitigator	Sarcastic
11	لو يجي منخفض اقوي شوي بتسحل الاردن عالسعوديه 🙃 سيول الأردن If a slightly stronger low air pressure comes, Jordan will slide into Saudi Arabia 🙃 Jordan's floods	🙃	Negativity Indicator	Sarcastic
12	اصلا مبين تركيب 😒 It's obviously fake 😒	😒	Negativity Trigger	Sarcastic

Figure 2: Examples of the resulting ironic texts and their corresponding emoji sentiment roles.

emoji pattern's humorous undertone is available in example (1) of Figure 2.

- *Positivity Mitigator*: 1.06% of the emoji serve to tone down the text's positive sentiment. Interestingly, 11.45% of these convey humour, demonstrated in example (2) of Figure 2.

- *Positivity Indicator*: 0.22% of emoji signal the inherent positive sentiment of the text. Of these, 12.73% bear a humorous intonation, as seen in example (3) of Figure 2.

- *Positivity Trigger*: A mere 0.069% of emoji initiate a positive sentiment in the text, with 5.88% insinuating humour. This is exemplified in example (4) of Figure 2.

- *Positive Reverser*: 0.020% of emoji intriguingly transform negative sentiments into positive ones. Among these, one pattern expresses humour, illustrated in example (5) of Figure 2.

4.2 Neutral Texts

In the ArSarcasmoji dataset, neutral texts—including those with mixed emotions or devoid of sentiment—account for 0.73% of the total. The emoji within these texts exclusively serve a *Neutral effect*, with the same percentage of 0.73%. The ironic nuances of these emoji patterns vary depending on their sentiment labels, as detailed below:

- *Positive Emoji Patterns with Neutral Effect*: Within the set of texts featuring *Neutral effect* emoji patterns, 0.55% convey humour in a positive context. This humorous manifestation of the emoji pattern can be observed in example (6) of Figure 2.

- *Neutral or Negative Emoji Patterns with Neutral Effect*: Among the texts showcasing *Neutral effect* emoji patterns, 1.1% exude sarcasm with a negative undertone. An exemplification of this sarcastic tone can be found in example (7) of Figure 2.

4.3 Negative Texts

Within the ArSarcasmoji dataset, 44.63% of the texts convey a negative sentiment. In these texts, emoji serve distinct roles, primarily:

- *Negativity Emphasizer*: A notable 36.23% of emoji in this subset act as non-verbal enhancers, magnifying the text's negative sentiment. Of these, 24.57% have sarcastic implications. Examples (8) and (9) from Figure 2 provide clear demonstrations of this sarcastic undertone.

- *Negativity Mitigator*: 7.73% of the emoji appear to modulate, reducing the intensity of the text's negative sentiment. Fascinatingly, 68.99% among these exhibit sarcasm, as showcased in example (10) of Figure 2.

- *Negativity Indicator*: 0.67% of the emoji directly indicate the text's inherent negative sentiment. Within this category, 39.63% possess a sarcastic tone, as illustrated in example (11) of Figure 2.

- *Negativity Trigger*: An exceptional 0.008% of emoji seem to spark a negative sentiment in the text. Half of these (i.e., one text) carry a sarcastic nuance, as depicted in example (12) of Figure 2.

5 Conclusion and Future Work

This study ventured into the relatively uncharted territory of emoji-augmented Arabic texts to discern nuances like humour and sarcasm. With the introduction of the ArSarcasMoji dataset, we have taken a pivotal step towards understanding the interplay of emoji patterns and sentiment roles in Arabic digital content. Our investigation has revealed that contrary to established beliefs, emoji play an indispensable role in accurately decoding Arabic textual irony. Their integration does not simply add another layer of complexity but rather serves as an essential tool to unmask the true intent behind statements.

Several avenues beckon exploration in the realm of Arabic NLP. One promising direction is to delve deeper into the multi-faceted semantics of the Arabic language and how emoji can further contribute to understanding other linguistic nuances, beyond irony. Another avenue would be to expand our dataset by incorporating different social media platforms, thereby ensuring a holistic understanding of emoji usage across the digital landscape. Furthermore, a comparative study between emoji-augmented Arabic texts and those of other languages might shed light on cultural nuances and their implications in NLP. Lastly, it would be insightful to develop machine learning models that can leverage the rich insights offered by the **ArSarcasMoji** dataset for automated irony detection, sentiment analysis, and beyond.

Limitations

While our study provides valuable insights into the relationship between texts and emoji in the context of irony detection on Twitter, it is crucial to acknowledge its boundaries and constraints. These limitations stem from both the dataset's intrinsic characteristics and the methodological choices we made during the research process. Recognizing these constraints not only underscores the areas where caution should be exercised when interpreting the results but also offers potential avenues for future research. Here, we delineate some of the principal limitations of our study:

- The dataset inadequately represents certain emoji sentiment roles such as the *Positivity Trigger*, *Positive Reverser*, *Negativity Trigger*, and *Negative Reverser* due to the inclusion of very few or even no texts corresponding to these roles.

- The dataset restricts ironic emoji patterns solely to facial expressions. However, the irony in textual conversations can be conveyed through various other emoji types, including hand gestures, body language, objects, and symbols, contingent on the context.

- The scope of irony detection in this study is confined to the relationship between texts and emoji. Incorporating other modalities like images, voice notes, and videos could significantly enrich irony detection by providing a more holistic understanding of a conversation's nuances.

- The dataset sources texts exclusively from Twitter. A more comprehensive irony detection dataset might consider incorporating texts from diverse platforms, such as WhatsApp or Telegram, to capture full conversational contexts.

- Linguistic nuances, regional dialects, and cultural contexts were not thoroughly accounted for in the dataset. This could lead to misinterpretations or overlooking of ironic constructs specific to certain cultures or languages.

- The dataset does not capture the temporal evolution of emoji meanings. Emoji can adopt new connotations over time, and a static dataset might not accurately reflect these dynamic shifts.

- The potential influence of trending topics or events on Twitter, which can temporarily modify the typical usage or meaning of certain emoji, was not factored into our analysis.

References

Sattar Fakher Abbas and Nazar Abdul Hafidh Ubeid. 2021. The interplay between text and emojis in iraqi telegram group chatting: A pragmatic-relevance study. *Adab al-Basrah*, 1(95):37–58. Issue Vol. 1, Issue 95 (31 Mar. 2021), pp. 37-58, 22 p.

Houssem Abdellaoui and Mounir Zrigui. 2018. Using Tweets and Emojis to Build TEAD: an Arabic Dataset for Sentiment Analysis. *ComputaciÃy Sistemas*, 22:777 – 786.

Meyer Howard Abrams and Geoffrey Galt Harpham. 2009. *A Glossary of Literary Terms*, 9th edition. Wadsworth Cengage Learning.

Ibrahim Abu Farha and Walid Magdy. 2019. Mazajak: An online Arabic sentiment analyser. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 192–198, Florence, Italy. Association for Computational Linguistics.

Ibrahim Abu Farha and Walid Magdy. 2020. From Arabic sentiment analysis to sarcasm detection: The ArSarcasm dataset. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 32–39, Marseille, France. European Language Resource Association.

Ibrahim Abu Farha and Walid Magdy. 2022. The effect of Arabic dialect familiarity on data annotation. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 399–408, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Kathrein Abu Kwaik, Stergios Chatzikyriakidis, Simon Dobnik, Motaz Saad, and Richard Johansson. 2020. An Arabic tweets sentiment analysis dataset (ATSAD) using distant supervision and self training. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 1–8, Marseille, France. European Language Resource Association.

Ali Al-Laith, Muhammad Shahbaz, Hind F. Alaskar, and Asim Rehmat. 2021. Arasencorpus: A semi-supervised approach for sentiment annotation of a large arabic text corpus. *Applied Sciences*, 11(5):2434.

Alaa Alharbi and Mark Lee. 2021. Kawarith: an Arabic Twitter corpus for crisis events. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 42–52, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Basma Alharbi, Hind Alamro, Manal Alshehri, Zuhair Khayyat, Manal Kalkatawi, Inji Ibrahim Jaber, and Xiangliang Zhang. 2020. ASAD: A twitter-based benchmark arabic sentiment analysis dataset. *CoRR*, abs/2011.00578.

Hend Alkhalifa, Fetoun Alzahrani, Hala Qawara, Reema Alrowais, Sawsan Alowa, and Luluh Aldhubayi. 2022. A dataset for detecting humor in Arabic text. In *Proceedings of the 5th International Conference on Natural Language and Speech Processing (ICNLSP 2022)*, pages 219–225, Trento, Italy. Association for Computational Linguistics.

Halah AlMazrua, Najla AlHazzani, Amaal AlDawod, Lama AlAwlaqi, Noura AlReshoudi, Hend Al-Khalifa, and Luluh AlDhubayi. 2022. Sa‘7r: A saudi dialect irony dataset. In *Proceedinsg of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools with Shared Tasks on Qur'an QA and Fine-Grained Hate Speech Detection*, pages 60–70, Marseille, France. European Language Resources Association.

Israa Alsarsour, Esraa Mohamed, Reem Suwaileh, and Tamer Elsayed. 2018. DART: A large dataset of dialectal Arabic tweets. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Nashat Alshboul and Luqman M Rababah. 2021. The emoji linguistic functions on facebook interactions among undergraduate students at jadara university in jordan. In *Journal for the Study of English Linguistics*, volume 9, pages 43–54.

Sarah N. Alyami and Sunday O. Olatunji. 2020. Application of support vector machine for arabic sentiment classification using twitter-based dataset. *Journal of Information & Knowledge Management*, 19(01):2040018.

Ramy Baly, Hazem Hajj, Wassim El-Hajj, and Khaled Shaban. 2018. ArSentD-LEV: A Multi-Topic Corpus for Target-based Sentiment Analysis in Arabic Levantine Tweets. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Ahmed Banikalef, Marlyna Maros, Ashinida Aladdin, et al. 2014. Linguistic analysis of humor in jordanian arabic among young jordanians facebookers. *Arab World English Journal*, 5(3).

Francesco Barbieri, Jose Camacho-Collados, Francesco Ronzano, Luis Espinosa-Anke, Miguel Ballesteros, Valerio Basile, Viviana Patti, and Horacio Saggion. 2018. SemEval 2018 task 2: Multilingual emoji prediction. In *Proceedings of the 12th International Workshop on Semantic Evaluation*, pages 24–33, New Orleans, Louisiana. Association for Computational Linguistics.

Santiago Castro, Luis Chiruzzo, Aiala Rosá, Diego Garat, and Guillermo Moncecchi. 2018. A crowd-annotated Spanish corpus for humor analysis. In *Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media*, pages 7–11, Melbourne, Australia. Association for Computational Linguistics.

Luis Chiruzzo, Santiago Castro, and Aiala Rosá. 2020. HAHA 2019 dataset: A corpus for humor analysis in Spanish. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5106–5112, Marseille, France. European Language Resources Association.

Neil Cohn, Jan Engelen, and Joost Schilperoord. 2019. The grammar of emoji? constraints on communicative pictorial sequencing. *Cognitive Research: Principles and Implications*, 4(1):33.

Marcel Danesi. 2017. *The semiotics of emoji: The rise of visual language in the age of the internet*. Bloomsbury Publishing.

AbdelRahim Elmadany, Hamdy Mubarak, and Walid Magdy. 2018. Arsas: An arabic speech-act and sentiment corpus of tweets. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Miramar Etman and Seham Elkareh. 2021. Nonverbal communication and emojis usage in arabic tweets: A cross-cultural study. *Social Networking*, 10:19–28.

Shatha Ali A Hakami, Robert Hendley, and Phillip Smith. 2020. Emoji as sentiment indicators: An investigative case study in arabic text. In *The Sixth International Conference on Human and Social Analytics*, pages 26–32.

Shatha Ali A. Hakami, Robert Hendley, and Phillip Smith. 2021. Arabic emoji sentiment lexicon (Arab-ESL): A comparison between Arabic and European emoji sentiment lexicons. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 60–71, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Shatha Ali A. Hakami, Robert Hendley, and Phillip Smith. 2022a. A context-free Arabic emoji sentiment lexicon (CF-Arab-ESL). 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 51–59, Marseille, France. European Language Resources Association.

Shatha Ali A. Hakami, Robert Hendley, and Phillip Smith. 2022b. Emoji sentiment roles for sentiment analysis: A case study in Arabic texts. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 346–355, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Fatima Haouari, Maram Hasanain, Reem Suwaileh, and Tamer Elsayed. 2021. ArCOV-19: The first Arabic COVID-19 Twitter dataset with propagation networks. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 82–91, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Sabit Hassan, Hamdy Mubarak, Ahmed Abdelali, and Kareem Darwish. 2021. ASAD: Arabic social media analytics and unDerstanding. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 113–118, Online. Association for Computational Linguistics.

Shirley Anugrah Hayati, Aditi Chaudhary, Naoki Otani, and Alan W Black. 2019. What a sunny day ☔: Toward emoji-sensitive irony detection. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 212–216, Hong Kong, China. Association for Computational Linguistics.

Mohammad Abdoh Mahzari. 2017. *Sociopragmatic study of the congratulation strategies of Saudi Facebook users*. Ph.D. thesis, Arizona State University. Partial requirement for: Ph.D., Arizona State University, 2017.

Rod A Martin and Thomas Ford. 2018. *The psychology of humor: An integrative approach*. Academic press.

Mary L McHugh. 2012. Interrater reliability: the kappa statistic. *Biochemia medica: Biochemia medica*, 22(3):276–282.

Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. SemEval-2018 task 1: Affect in tweets. In *Proceedings of the 12th International Workshop on Semantic Evaluation*, pages 1–17, New Orleans, Louisiana. Association for Computational Linguistics.

Pasant Mohammed, Yomna Eid, Mahmoud Badawy, and Ahmed Hassan. 2020. Evaluation of different sarcasm detection models for arabic news headlines. In *Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2019*, pages 418–426, Cham. Springer International Publishing.

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, Florence, Italy. 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.

Eric Partridge. 1997. *Usage and abusage: A guide to good English*. WW Norton & Company.

Omid Rohanian, Shiva Taslimipoor, Richard Evans, and Ruslan Mitkov. 2018. WLV at SemEval-2018 task 3: Dissecting tweets in search of irony. In *Proceedings of the 12th International Workshop on Semantic Evaluation*, pages 553–559, New Orleans, Louisiana. Association for Computational Linguistics.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. SemEval-2017 task 4: Sentiment analysis in Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 502–518, Vancouver, Canada. Association for Computational Linguistics.

Mohammad Salameh, Saif Mohammad, and Svetlana Kiritchenko. 2015. Sentiment after translation: A case-study on Arabic social media posts. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 767–777, Denver, Colorado. Association for Computational Linguistics.

Shiwei Wang. 2022. Sarcastic meaning of the slightly smiling face emoji from chinese twitter users: When a smiling face does not show friendliness. *International Journal of Languages, Literature and Linguistics*, 8(2):65–73.

Benjamin Weissman and Darren Tanner. 2018. A strong wink between verbal and emoji-based irony: How the brain processes ironic emojis during language comprehension. *PLOS ONE*, 13(8):1–26.

Bagus Satria Wiguna, Cinthia Vairra Hudiyanti, Alqis Alqis Rausanfita, and Agus Zainal Arifin. 2021. Sarcasm detection engine for twitter sentiment analysis using textual and emoji feature. *Jurnal Ilmu Komputer dan Informasi*, 14(1):1–8.

Najma Al Zidjaly. 2017. Memes as reasonably hostile laments: A discourse analysis of political dissent in oman. *Discourse & Society*, 28(6):573–594.

Performance Implications of Using Unrepresentative Corpora in Arabic Natural Language Processing

Saied Alshahrani **Norah Alshahrani** **Soumyabrata Dey** **Jeanna Matthews**
Department of Computer Science, Clarkson University, Potsdam, New York, USA
{saied,norah,sdey,jnm}@clarkson.edu

Abstract

Wikipedia articles are a widely used source of training data for Natural Language Processing (NLP) research, particularly as corpora for low-resource languages like Arabic. However, it is essential to understand the extent to which these corpora reflect the representative contributions of native speakers, especially when many entries in a given language are directly translated from other languages or automatically generated through automated mechanisms. In this paper, we study the performance implications of using inorganic corpora that are not representative of native speakers and are generated through automated techniques such as bot generation or automated template-based translation. The case of the Arabic Wikipedia editions gives a unique case study of this since the Moroccan Arabic Wikipedia edition (ARY) is small but representative, the Egyptian Arabic Wikipedia edition (ARZ) is large but unrepresentative, and the Modern Standard Arabic Wikipedia edition (AR) is both large and more representative. We intrinsically evaluate the performance of two main NLP upstream tasks, namely word representation and language modeling, using word analogy evaluations and fill-mask evaluations using our two newly created datasets: Arab States Analogy Dataset (ASAD) and Masked Arab States Dataset (MASD). We demonstrate that for good NLP performance, we need both large and organic corpora; neither alone is sufficient. We show that producing large corpora through automated means can be a counter-productive, producing models that both perform worse and lack cultural richness and meaningful representation of the Arabic language and its native speakers.

1 Introduction

Natural Language Processing (NLP) plays a crucial role in decision-making systems. For instance, it is employed in resume parsers that assist in sorting job candidates. NLP systems are typically designed to analyze extensive collections of human text (corpora) with the goal of deriving insights from human behavior and generating recommendations on our behalf (Wali et al., 2020). The normal, organic, and representative corpora of human text produced by native speakers (the main ingredients in NLP systems) convey many social concepts, including culture, heritage, and even historic biases (Bolukbasi et al., 2016; Caliskan et al., 2017; Babaeianjelodar et al., 2020; Cho et al., 2021; Chen et al., 2021).

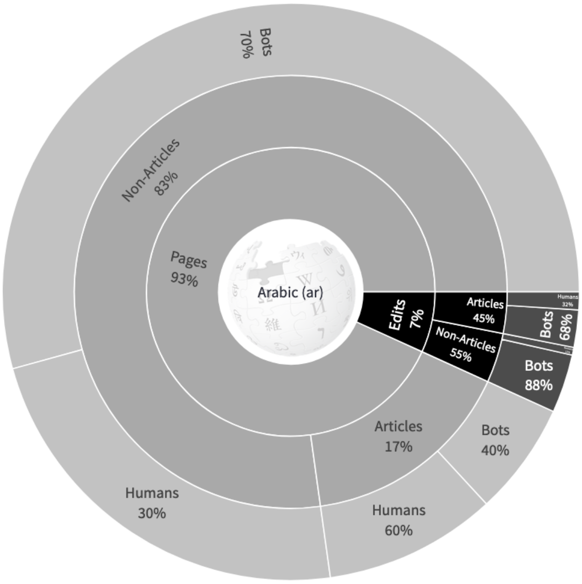


Figure 1: A sunburst visualization from our Wikipedia Corpora Meta Report dashboard (discussed in more detail in Appendix A) shows the percentage of contributions of bots and humans in the Modern Standard Arabic Wikipedia edition.

One of the widely used human text corpora and a common source of training data for NLP research is Wikipedia articles (content pages), especially in languages other than English. In specific, Wikipedia articles are used to train many Large Language Models (LLMs), such as ELMo (Embeddings from Language Models), which has been trained on the English Wikipedia and news crawl data (Peters et al., 2018); BERT (Bidirec-

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 218–231
December 7, 2023 ©2023 Association for Computational Linguistics

tional Encoder Representations from Transformers) has been trained on books with a crawl of English Wikipedia (Devlin et al., 2018); GPT-3 (Generative Pre-trained Transformer) has also been trained on five large datasets including the English Wikipedia (Brown et al., 2020); LaMDA (Language Model for Dialogue Applications) and PaLM (Pathways Language Model) were trained on a huge mixed dataset that includes English Wikipedia articles (Thoppilan et al., 2022; Chowdhery et al., 2022); and recently, LLaMA (Large Language Model Meta AI) was also pre-trained on the multilingual articles of Wikipedia from June to August 2022, covering 20 languages with a percentage of 4.5% of its overall training dataset size (Touvron et al., 2023).

Wikipedia corpora (editions) exist for over 300 of the over 7,000 languages spoken worldwide. These corpora vary greatly in size and quality, yet simply having a corpus of text in a certain language does not mean that it is an organic corpus representing the culture of native speakers. While native speakers originally write some corpora, others may be written by non-native speakers or translated from other languages (Nisioi et al., 2016). Recent research studied the Arabic Wikipedia editions: Modern Standard Arabic (AR), Egyptian Arabic (ARZ), and Moroccan Arabic (ARY), and found that in the Egyptian Arabic Wikipedia edition more than *one* million articles have been shallowly translated from English using either direct or template-based translation, all by a single registered user (Alshahrani et al., 2022). Alshahrani et al. (2022) argued that these shallowly translated articles do not echo the complex structure of the Arabic language and its dialects and do not express the views of Arabic speakers. In another recent research, Alshahrani et al. (2023) observed that the top ten Wikipedia editions (based on the total number of articles) are mostly bot-generated or auto-translated. To mitigate this problem, they introduced an enhanced Wikipedia depth metric, DEPTH$^+$, used as a rough indicator for the Wikipedia corpora quality, where they quantified and removed bot-generated Wikipedia articles and bot-made edits on those articles. Both works claimed that these practices of automation and translation could negatively impact the performance of NLP systems trained on these corpora, but they did not provide any empirical studies to show to which extent these practices could implicate the performance of specific NLP tasks and systems, including those using LLMs.

In this paper, we aim to bridge this gap by studying the performance implications of using such unrepresentative, inorganic corpora (produced by template-based translation or automatic bots creation/generation) by intrinsically evaluating two main NLP upstream tasks: word representation and language modeling, using word analogy and fill-mask evaluations, respectively, to capture syntactic and semantic relations between words. We purposely choose these intrinsic evaluations over extrinsic evaluations such as text classification or machine translation because many studies have shown that extrinsic and intrinsic evaluations' results are not consistently correlated, and the performance of NLP downstream tasks is always task-specific and can be significantly influenced by fine-tuning procedures (Faruqui et al., 2016; Schröder et al., 2021; Cao et al., 2022). We believe that evaluating NLP upstream tasks intrinsically will give us useful insights into the quality of the Arabic Wikipedia editions' corpora and show how the quality of corpora affects the performance of these NLP tasks.

We, in the following sections, discuss the problem of the unrepresentative corpora (§2), highlight the experimental setup of our study (§3), present the word representation and language modeling evaluations (§4 and §5), discuss the results and the limitations of our work (§6 and §7), provide a brief conclusion and offer future research ideas (§8).

2 Problem of Unrepresentative Corpora

The Wikipedia corpora (articles) unsurprisingly are not only used to train the large multilingual LLMs such as BERT (Devlin et al., 2018), LLaMA (Touvron et al., 2023), or even mGPT (multilingual GPT) (Shliazhko et al., 2022), but also have been used to train the majority of the Arabic LLMs, like AraBERT (Antoun et al., 2020), AraGPT2 (Antoun et al., 2021b), AraELECTRA (Antoun et al., 2021a), ARBERT and MARBERT (Abdul-Mageed et al., 2021), AraT5 (Nagoudi et al., 2022), *Jais* and *Jais-chat* (Sengupta et al., 2023), and recently, AceGPT (Huang et al., 2023). Therefore, there is a need to study Wikipedia's corpora representativeness, specifically in the Arabic Wikipedia editions, and to define the unrepresentativeness in its corpora as well. In this work, we generally define unrepresentative Wikipedia corpora as "*any Wikipedia articles (content pages) that have been created, generated, or edited without human involvement or supervision*", such as automatically created, generated, or edited Wikipedia articles using bots or shallowly template-translated articles from other highly resourced languages like English.

We study this problem from two perspectives: template-translated corpora and bot-generated corpora. For the template-translated corpora, Alshahrani et al. (2022) have studied the Arabic Wikipedia editions and shown that more than *one* million articles in the Egyptian Arabic Wikipedia have been directly translated using simple templates that lack rich content from the English language with the help of the off-the-shelf translation tools like Google Translate. These translation tools generally perform well, but not perfectly, and have several serious problems, such as gender bias, that could adversely affect the translated content (Prates et al., 2020; Ullmann and Saunders, 2021; Lopez-Medel, 2021). For the bot-generated corpora, a few recent research have shed light on the bots' activities on the Wikipedia project and their possible negative impacts on the quality of Wikipedia corpora (Tsvetkova et al., 2017; Zheng et al., 2019; Alshahrani et al., 2023). The root problem with the bots is that they can rapidly create Wikipedia articles (content pages) or edit the contents of those articles without any humans in the loop (Adler et al., 2008; Kang et al., 2021; Alshahrani et al., 2022).

Wikipedia	Total Articles	Human created Articles (%)	Bot generated Articles (%)
Arabic (AR)	1,197,467	717,678 (59.93%)	479,789 (40.07%)
Egyptian (ARZ)	1,616,530	1,616,515 (99.99%)	15 (0.0001%)
Moroccan (ARY)	6,426	5,684 (88.45%)	742 (11.55%)

Table 1: Categorization of Arabic Wikipedia editions by total articles, human-created articles, and bot-generated articles. This does *not* include the inorganic template-translated articles in the Egyptian Arabic Wikipedia.[1]

In this paper, we quantify the bots' activities in all Wikipedia editions and study the Arabic Wikipedia editions closely, specifically activities on their articles. We find that nearly 40% of articles in the Arabic Wikipedia edition are bot-generated (as demonstrated in Figure 1), and nearly 12% of articles in the Moroccan Arabic Wikipedia edition are bot-generated, as shown in Table 1. Surprisingly, the Egyptian Arabic Wikipedia edition has *only* 15 bot-generated articles, even though it is heavily affected by template-based translation activities (Alshahrani et al., 2022). We use Wikimedia `XTools` API[2] to identify Wikipedia articles' authors and exclude bot-generated articles from the Wikipedia corpora. We also use Wikipedia's "List Users" service[3] to retrieve the full list of bots in each Arabic Wikipedia edition to help us disclose the articles whose authors are in the bots list. We use the complete Wikipedia dumps of each Arabic Wikipedia edition, downloaded on the 1st of January 2023 (Wikimedia Foundation, 2023), process them using `Gensim` Python library (Řehůřek and Sojka, 2010), and preprocess them using `tr` Linux/Unix utility and `CAMeLTools` Python toolkit for Arabic NLP (Obeid et al., 2020). We extract all the Wikipedia articles from the three Arabic Wikipedia editions: Arabic, Egyptian Arabic, and Moroccan Arabic, and preprocess them slightly, removing the diacritical marks and the Latin letters and numbers; we do not apply stemming, lemmatization, or heavy text normalization on them to have organic texts (corpora) as much as possible.

2.1 Impact of Template-based Translation

Throughout this paper, to explore the impact of template-based translation, we compare the performance of models trained on the Egyptian Arabic Wikipedia edition's corpora that are dominated by shallow template-based translation (Baker, 2022; Alshahrani et al., 2022) to models trained on the Modern Standard Arabic and Moroccan Arabic Wikipedia editions' corpora, which are not.

2.2 Impact of Bot-based Generation

Similarly, throughout this paper, to explore the impact of bot-based generation, we compare the performance of models trained on Modern Standard Arabic and Moroccan Arabic Wikipedia editions' corpora (with and without bot-generated articles).

3 Experimental Setup

In this work, we examine two key NLP upstream tasks, namely word representation and language modeling, using curated corpora of the Arabic Wikipedia editions' articles and intrinsically evaluate them using two evaluation tasks on two newly created datasets. We next describe the evaluation tasks and our created datasets in more detail.

[1] Unlike the bots' quantifications process, the quantification of template-based translations is only specific to the Egyptian Arabic edition. Wikipedia project does not track template-based translation in its metadata as it does with bot generation.

[2] XTools API: https://www.mediawiki.org/wiki/XTools.

[3] https://{wiki_code}.wikipedia.org/wiki/Special:ListUser.

3.1 Evaluation Tasks

We use two evaluation tasks: word analogy and fill-mask, to intrinsically evaluate the two main NLP upstream tasks. In the following subsections, we describe these evaluation tasks in more detail.

3.1.1 Word Analogy Task

The word analogy task was originally introduced by Mikolov et al. (2013a), and the goal is to find the missing word $b^\star$ in the relation: a is to a^* as b is to $b^\star$, where b and $b^\star$ are related by the same direction as a and a^*. For example, king:*man**::queen:*woman**. Each analogy question will be solved by calculating the target vector $b^\star$, $b^\star = b - a + a^*$. We calculate the cosine similarity between the target vector $b^\star$ and the vector representation of each word w in a given word embedding vector V. We lastly get the most similar word w to $b^\star$, following $\texttt{argmax}_{w \in V}(\texttt{sim}(w, b - a + a^*))$. If $w = b^\star$ (the same word), we then assume the given word embedding vector V has answered the analogy question correctly.

We overcome the challenge of the Arabic words having possible multiple variants by 1) extending the top K value (default K=1) to K={1, 5, 10} to search for the correct answer among the returned list of most similar words and 2) introducing a generic search algorithm that takes the word w and then searches for all its possible variants. We only consider looking into the variants of *Alefs* {ا ، آ ، أ ، إ}, *Alef Maksura* {ي ، ى}, and *Teh Marbuta* {ـة ، ـه}. For example, if the word w is “امرأة / woman”, then the lookup list of w’s variants is: {إمرأة ، امرأه ، امراة ، امراه}.

3.1.2 Fill-Mask Task

Masked language modeling involves masking some words in a sentence and predicting which words should replace those masked words. The valuable feature of this evaluation task is that it gives us a statistical understanding of the corpora on which our Masked Language Models (MLMs) are trained. We evaluate our MLM models that have been trained on the Arabic Wikipedia editions’ corpora using our created datasets. We utilize the “fill-mask” pipeline of the Hugging Face with our MLM models (Wolf et al., 2020; Hugging Face, 2023a).

We follow the same approaches, as addressed in subsection 3.1.1, to beat the challenge of the Arabic words having possible multiple variants by extending the MLM top K value (default K=10) to K={10, 50, 100} and using the previously introduced generic search algorithm that takes the word w and searches for all its possible variants.

3.2 Created Datasets

We collect 20 Arab states with their corresponding capital cities, nationalities, currencies, and on which continents they are located.[4] We deliberately select the Arab states because they are facts and cannot change even in different Arabic dialects, like Egyptian and Moroccan Arabic. We, in the following subsections, describe these two created datasets in more detail.

3.2.1 Arab States Analogy Dataset

We generate the Arab States Analogy Dataset (ASAD), consisting of four sets: country-capital set, country-currency set, country-nationality set, and country-continent set. Each set has 380 word analogies, and the total number of word analogies in the ASAD dataset is 1520. Table 2 demonstrates an example of each set, along with their English translations.

ASAD SET	WORD ANALOGY EXAMPLE
Country-Capital	القاهرة مصر الرباط المغرب Cairo Egypt Rabat Morocco
Country-Currency	مصر الجنيه المغرب الدرهم Egypt Pound Morocco Dirham
Country-Nationality	مصر المصري المغرب المغربي Egypt Egyptian Morocco Moroccan
Country-Continent	مصر أفريقيا المغرب أفريقيا Egypt Africa Morocco Africa

Table 2: Word analogy examples from the Arab States Analogy Dataset (ASAD) and their English translations.

3.2.2 Masked Arab States Dataset

We generate the Masked Arab States Dataset (MASD), consisting of four categories: country-capital prompts, country-currency prompts, country-nationality prompts, and country-continent prompts. Each prompts category has 40 masked prompts, and the total number of masked prompts in the MASD dataset is 160. We notice that some masked prompts could lead to ambiguous masked prompts, which can be hard to be answered by the MLMs, and to fix this issue, we rephrase the ambiguous masked prompts, using

[4] We only drop two Arab states: the United Arab Emirates (الإمارات العربية المتحدة) and Comoros (جزر القمر), because they or their capital cities are written as open compound words (two words), like Abu Dhabi (أبو ظبي), which cannot be handled directly by the word embedding models.

the same facts/information about the Arab states. For example, the masked prompt "*The pound is the currency of <mask>.*" is ambiguous because many Arab states, including Egypt, Sudan, Lebanon, and Syria, use the pound as their currency, and our rephrase/disambiguation of this masked prompt is "*The currency of Egypt is the <mask>.*". Additionally, we add the masked prompts answers (masked words) of each masked prompt to the MASD dataset for the sake of validation and future evaluation. Table 3 shows an example of each masked prompts category, their masked prompts answers, and their English translations.

MASD CATEGORY	MASKED PROMPTS EXAMPLE
Country-Capital	القاهرة هي عاصمة دولة <mask>. Cairo is the capital of <mask>. * **MASKED ANSWER:** مصر Egypt
Country-Currency	عملة دولة مصر هي <mask>. The currency of Egypt is the <mask>. * **MASKED ANSWER:** الجنيه Pound
Country-Nationality	أحب دولة مصر و أحب الشعب <mask>. I love Egypt, and I love the <mask> people. * **MASKED ANSWER:** المصري Egyptian
Country-Continent	تقع دولة مصر في قارة <mask>. Egypt is located on the continent of <mask>. * **MASKED ANSWER:** أفريقيا Africa

Table 3: Masked prompts examples with their answers from the Masked Arab States Dataset (MASD) and their English translations.

4 Word Representation Evaluations

Word embeddings are a well-known word representation technique used by modern NLP systems as their backbone. They encode syntactic and semantic relations between words in a text and represent them in a low-dimensional space.

4.1 Impact of Template-based Translation

In the following subsections, we evaluate the performance of the word embedding models using the word analogy task and our ASAD dataset. Recall we compare the performance of models trained on the Egyptian Arabic Wikipedia edition's corpora, which are dominated by template-based translation, to the performance of models trained on Modern Standard Arabic and Moroccan Arabic Wikipedia editions' corpora, which are not.

4.1.1 Word Embedding Models

We train *five* context-independent word embedding models on each Arabic Wikipedia edition's corpora using three different word representation algorithms: `Word2Vec` (continuous bag of words (cbow) and skip-gram), `fastText` (cbow and skip-gram), and `GloVe` (Mikolov et al., 2013b; Bojanowski et al., 2017; Pennington et al., 2014). We set these unified parameters of the three algorithms to these values: {*vector-size=300, epochs=20, window-size=2, min-count=1, alpha=0.03*}.

WIKIPEDIA	ARTICLES	WORDS	SENTENCES
AR	1,197,467	258,676,800	1,088,502
ARZ	1,616,530	65,565,053	728,340
ARY	6,426	720,334	5,394

Table 4: General statistics of the Arabic Wikipedia editions in terms of the total number of articles, total number of words, and total number of sentences.

Table 4 shows the Arabic Wikipedia editions' corpora statistics and confirms the findings of Alshahrani et al. (2022) that Egyptian Arabic Wikipedia has poor content pages, a side effect of the template-based translation. Although it has the largest number of articles among other Arabic Wikipedia editions, this large number of articles does not reflect the content richness when comparing the total words and sentences with the Modern Standard Arabic Wikipedia edition.

4.1.2 Results of Word Analogy Task

We evaluate our word embedding models trained on the Arabic Wikipedia editions' corpora using our introduced ASAD dataset. In Table 5, we can see that increasing the top K value and searching for words' variants improves the accuracy metric greatly. We also observe that the overall performance of the word embedding models varies, where the word embedding models trained on the Arabic Wikipedia edition's corpora performs dramatically better despite having fewer articles than the Egyptian Arabic Wikipedia edition's corpora, which comes in second in terms of performance; this contradicts the common assumption of "*the more articles a Wikipedia edition has, the better the quality of its corpus*". The word embedding models trained on the Moroccan Arabic Wikipedia edition's corpora performed the worst since they have been trained on very small corpora (less than 6,500 articles). This illustrates our key observation that we need both large and organic corpora for good NLP performance; neither alone is sufficient. We further highlight the best and worst word embedding models in Appendix B.

4.2 Impact of Bot-based Generation

We, in the following subsections, compare the performance of word embedding models that have

WIKIPEDIA	MODEL	K=1	K=5	K=10
AR	Word2Vec-cbow	53.88%	74.47%	79.67%
	Word2Vec-skipgram	53.82%	71.91%	76.64%
	fastText-cbow	21.97%	34.67%	44.47%
	fastText-skipgram	39.67%	57.17%	65.79%
	GloVe	36.58%	50.53%	54.14%
ARZ	Word2Vec-cbow	13.88%	26.97%	33.09%
	Word2Vec-skipgram	5.00%	9.08%	11.05%
	fastText-cbow	10.13%	20.86%	28.09%
	fastText-skipgram	11.64%	18.22%	22.37%
	GloVe	0.53%	3.29%	5.20%
ARY	Word2Vec-cbow	1.91%	5.86%	8.22%
	Word2Vec-skipgram	2.11%	4.01%	5.92%
	fastText-cbow	1.71%	4.41%	6.38%
	fastText-skipgram	3.68%	9.87%	14.61%
	GloVe	0.13%	0.53%	0.66%

Table 5: Overall performance of each word embedding model of the Arabic Wikipedia editions evaluated on all the sets of our ASAD dataset.

been trained on Arabic and Moroccan Arabic corpora (with and without bot-generated articles) using the word analogy task and our ASAD dataset.

4.2.1 Word Embedding Models

We train *five* context-independent word embedding models on both Arabic Wikipedia and Moroccan Arabic Wikipedia editions' corpora (after excluding bot-generated articles)[5] using three different word representation algorithms: `Word2Vec` (cbow and skip-gram), `fastText` (cbow and skip-gram), and `GloVe` (Mikolov et al., 2013b; Bojanowski et al., 2017; Pennington et al., 2014). We use the same values for the unified parameters for the three algorithms, as illustrated in subsection 4.1.1. In Table 6, we highlight the Arabic Wikipedia and Moroccan Arabic Wikipedia corpora statistics in terms of the number of articles, words, and sentences after all bot-generated articles are eliminated.

WIKIPEDIA	ARTICLES	WORDS	SENTENCES
AR	717,678	250,378,412	847,387
ARY	5,684	694,756	4,673

Table 6: General statistics of the Arabic Wikipedia and Moroccan Arabic Wikipedia editions regarding the number of articles, total words, and total sentences after removing the bot-generated articles.

4.2.2 Results of Word Analogy Task

We evaluate our word embedding models that have been trained on the Arabic Wikipedia and Moroccan Arabic Wikipedia editions' corpora using our introduced ASAD dataset. As highlighted in 4.1.2, increasing the top K value and searching for words' variants boosts the accuracy metric for the overall performance of all word embedding models of the Arabic and Moroccan Arabic Wikipedia editions. In Table 7, we compare the word embedding models trained on the Arabic Wikipedia corpora with bot activities (bot-generated articles included) and without bot activities (bot-generated articles excluded). We can see that most of the word embedding models trained with no bot-generated articles excel when K=1 and perform close to those trained with bot-generated articles when K={5, 10}. Surprisingly, the performance is generally the same or at times, even better, even though we have removed nearly 480K bot-generated articles (40% of total articles). This result emphasizes our observation that automated generation to increase the size of a corpus can actually be a counter-productive to NLP performance.

AR MODEL	CORPORA	K=1	K=5	K=10
Word2Vec-cbow	With bots	53.88%	74.47%	79.67%
	No bots	53.22%	74.47%	79.47%
Word2Vec-skipgram	With bots	53.82%	71.91%	76.64%
	No bots	54.47%	71.84%	75.92%
fastText-cbow	With bots	21.97%	34.67%	44.47%
	No bots	22.76%	34.34%	43.29%
fastText-skipgram	With bots	39.67%	57.17%	65.79%
	No bots	39.87%	56.64%	67.43%
GloVe	With bots	36.58%	50.53%	54.14%
	No bots	38.29%	52.11%	55.13%

Table 7: Overall performance of word embedding models of the Arabic Wikipedia edition evaluated on all the sets of our ASAD dataset before and after removing bot-generated articles.

ARY MODEL	CORPORA	K=1	K=5	K=10
Word2Vec-cbow	With bots	1.91%	5.86%	8.22%
	No bots	1.84%	4.54%	7.11%
Word2Vec-skipgram	With bots	2.11%	4.01%	5.92%
	No bots	2.11%	3.75%	5.53%
fastText-cbow	With bots	1.71%	4.41%	6.38%
	No bots	1.97%	4.41%	6.45%
fastText-skipgram	With bots	3.68%	9.87%	14.61%
	No bots	3.62%	9.54%	13.75%
GloVe	With bots	0.13%	0.53%	0.66%
	No bots	0.07%	0.26%	0.39%

Table 8: Overall performance of word embedding models of the Moroccan Arabic Wikipedia edition evaluated on all the sets of our ASAD dataset before and after removing bot-generated articles.

In Table 8, we also compare the performance of the word embedding models trained on the Moroccan Arabic Wikipedia corpora with bot activities (bot-generated articles included) and without bot activities (bot-generated articles excluded). We find that most of the word embedding models trained with bot-generated articles are generally better, except for the word embedding models produced by the `fastText` (cbow) that trained on no bot-generated articles (1.97% and 6.45% when K={1, 10}, respectively). We attribute these poor results to the small size of the Moroccan Arabic Wikipedia corpora, and eliminating the bot-generated articles makes the corpora even smaller. Once again, we say for good NLP performance, both large and organic corpora are very important.

[5] We drop the Egyptian Arabic Wikipedia due to having an insignificant number of bot-generated articles, only 15 articles.

5 Language Modeling Evaluations

Language modeling is an NLP task that generally predicts words in a sentence, and it is the heart of most existing LLMs. Some of these powerful LLMs, like BERT or RoBERTa, are usually trained using two objectives: masked language modeling and next sentence prediction (Devlin et al., 2018; Liu et al., 2019). In the following subsections, we exploit the masked language modeling objective in training Masked Language Models (MLMs) to produce contextual word embeddings and evaluate the performance of the MLM models trained on the Arabic Wikipedia editions' corpora using our created masked prompts dataset. We evaluate the quality of these MLM models using the Pseudo-Perplexity metric; we detailedly describe the evaluation process in Appendix C.

5.1 Impact of Template-based Translation

We, in the following subsections, evaluate the performance of the masked language models using the fill-mask task and our MASD dataset. Recall we compare the performance of models trained on the Egyptian Arabic Wikipedia edition's corpora, which are dominated by template-based translation, to the performance of models trained on Modern Standard Arabic and Moroccan Arabic Wikipedia editions' corpora, which are not.

5.1.1 Masked Language Models

We train *three* RoBERTa$_{\text{BASE}}$ models *from scratch* on each Arabic Wikipedia edition's corpora (arRoBERTa$_{\text{BASE}}$, arzRoBERTa$_{\text{BASE}}$, and aryRoBERTa$_{\text{BASE}}$) with one modification on their architectures. We set the number of hidden layers to 6 instead of 12 for less computational overhead and to make the MLM models twice as fast as the RoBERTa$_{\text{BASE}}$ introduced by Liu et al. (2019).[6] We also train *three* Byte-level Byte-Pair-Encoding (BPE) tokenizers, one for each Arabic Wikipedia edition's corpora.[7] The full list of hyperparameters used to train our MLM models and tokenizers is shown in Table 9. We further evaluate these newly trained MLM models using the Pseudo-Perplexity metric in Appendix C.1.

ROBERTA$_{\text{BASE}}$ MODEL	BYTE-LEVEL BPE TOKENIZER
Hidden Layers: 6	**Vocabulary Size:** 52,000
Hidden Size: 768	**Minimum Frequency:** 2
Attention Heads: 12	**Special Tokens:** ● Start Token: <s> ● End Token: </s> ● Padding Token: <pad> ● Unknown Token: <unk> ● Masking Token: <mask>
Vocabulary Size: 52,000	
Type Vocabulary Size: 1	
Max Sequence Length: 514	
Number of Epochs: 5	
Learning Rate: 1e–4	
Batch Size: {128, 256}	
Adam $\mathcal{E}$**:** 1e–6	
Adam β_1**:** 0.9	
Adam β_2**:** 0.98	
Weight Decay: 0.01	
Trainable Parameters: 83M	

Table 9: Full list of hyperparameters of our Masked Language Models (MLMs) and their tokenizers.

5.1.2 Results of Fill-Mask Task

We evaluate our MLM models that have been trained on the Arabic Wikipedia editions' corpora using our introduced MASD dataset. We can see in Table 10 that the performance of the Arabic arRoBERTa$_{\text{BASE}}$ model is superior to the Egyptian arzRoBERTa$_{\text{BASE}}$ model when K=10 (43.12% and 8.12%, respectively). Even though the Arabic Wikipedia edition has fewer articles than the Egyptian Arabic Wikipedia edition, it performs better and better represents the Arabic language. We also observe that increasing the MLM top K value could lead to an average improvement in the performance of all MLM models, except the Moroccan aryRoBERTa$_{\text{BASE}}$ model, which scores zero accuracies regardless of the increment of the K value; this is understandable since it was trained on corpora of less than 6,500 Wikipedia articles. Lastly, we see a performance jump of nearly 10% of the Egyptian arzRoBERTa$_{\text{BASE}}$ model when K={50, 100}, meaning the model is able to answer the masked prompts, but the correlation between the prompts and the answers is weak.

MLM MODEL	K=10	K=50	K=100
arRoBERTa$_{\text{BASE}}$	43.12%	45.00%	50.62%
arzRoBERTa$_{\text{BASE}}$	8.12%	25.62%	35.00%
aryRoBERTa$_{\text{BASE}}$	0.00%	0.00%	0.62%

Table 10: Performance of each masked language model of the Arabic Wikipedia editions on all the categories of MASD dataset.

5.2 Impact of Bot-based Generation

We, in the following subsections, compare the performance of masked language models that have been trained on Modern Standard Arabic Wikipedia and Moroccan Arabic Wikipedia editions' corpora (with and without bot-generated articles) using the fill-mask task and our MASD dataset.

[6] This modified architecture of RoBERTa$_{\text{BASE}}$ is called "DistilRoBERTa$_{\text{BASE}}$" by the Hugging Face: https://huggingface.co/distilroberta-base.

[7] We train our MLM models and their tokenizers using the Hugging Face Python libraries: `Transformers` and `Tokenizers` (Wolf et al., 2020). We exclude the default hyperparameters of training arguments from Table 9.

5.2.1 Masked Language Models

We train *two* RoBERTa$_{\text{BASE}}$ models *from scratch* on both Arabic Wikipedia and Moroccan Arabic Wikipedia editions' corpora after excluding bot-generated articles (arRoBERTa$_{\text{BASE}}$ and aryRoBERTa$_{\text{BASE}}$) and train *two* Byte-level Byte-Pair-Encoding (BPE) tokenizers, one for each Arabic Wikipedia edition's corpora; we drop the Egyptian Arabic Wikipedia for not having many bot-generated articles (only 15 articles). We use the same hyperparameters used to train our MLM models and tokenizers in subsection 5.1.1 and study the same processed corpora for Arabic Wikipedia and Moroccan Arabic Wikipedia, as discussed in Table 6, subsection 4.2.1. We further evaluate these newly trained MLM models using the Pseudo-Perplexity metric in Appendix C.2.

5.2.2 Results of Fill-Mask Task

We evaluate our MLM models that have been trained on the Arabic Wikipedia and Moroccan Arabic Wikipedia editions' corpora (with and without bot-generated articles) using our introduced MASD dataset. As shown in Table 11, the MLM models trained on the Arabic Wikipedia corpora when bots' activities are eliminated (bot-generated articles) perform better than those trained on corpora that include the bots' activities, even though this corpus is smaller in terms of the number of articles than the corpora with bots. Interestingly, the performance of all Moroccan Arabic Wikipedia MLM models remains the same, even after being trained on no-bots corpora, which have fewer articles than the bots corpora.

MLM Model	**Corpora**	**K=10**	**K=50**	**K=100**
arRoBERTa$_{\text{BASE}}$	With bots	43.12%	45.00%	50.62%
	No bots	45.62%	51.25%	53.12%
aryRoBERTa$_{\text{BASE}}$	With bots	0.00%	0.00%	0.62%
	No bots	0.00%	0.00%	0.62%

Table 11: Overall performance of MLMs of the Arabic Wikipedia and Moroccan Arabic Wikipedia editions evaluated on all the categories of MASD dataset before and after removing the bot-generated articles.

6 Discussion

Recent research has shown that not all Wikipedia editions (languages) are produced by native speakers, and there are substantial activities of auto-creation of articles (bot-generated articles) and auto-translation of articles (template-translated articles) in Wikipedia (Alshahrani et al., 2022, 2023). In this work, we argue that this automatic translation of articles, specifically the template-based translation on the Egyptian Arabic Wikipedia edition, impacts the overall performance of the NLP tasks due to having poor, limited, and unrepresentative corpora. Table 4 confirms that this template-based translation may enlarge the number of articles but cannot hide the true quality of a corpus. The Egyptian Arabic Wikipedia edition might have larger article numbers, but the truth is that these articles have fewer words and sentences than the Arabic Wikipedia edition. We find that all the word embedding models and all the masked language models that have been trained on each Arabic Wikipedia edition follow the same pattern, that is the models trained using the Arabic Wikipedia edition's corpora (which are widely believed to be mostly produced organically by the Arabic native speakers) perform better than the models trained on the Egyptian Arabic and Moroccan Arabic editions' corpora, as shown in Tables 5 and 10. We also believe that when K=10 (the default value), the masked language models usually show their actual performance, and as displayed in Table 10, it is obvious that the template-translated articles badly impact the masked language model trained on the Egyptian Arabic Wikipedia corpora when compared to the masked language model trained on the Arabic Wikipedia corpora despite the fact its corpora has nearly 480K articles more than the Arabic Wikipedia corpora, as shown in Table 4. It is evident that when masked language models are trained on naturally produced corpora by native speakers, they are more likely to have a better representation of the syntactic and semantic relations between words and a better understanding of the language itself and its native speakers.

We further argue, in this work, that the automatic creation and generation of articles, specifically the bots' creation and generation of articles on the Arabic Wikipedia and Moroccan Arabic Wikipedia editions, impacts the overall performance of the NLP tasks due to having unnatural, inorganic, and unrepresentative corpora. Once again, Table 1 confirms that this bots' generation may enlarge the number of total articles but cannot hide the true quality of a corpus. Even though the Arabic Wikipedia edition has a large number of articles (including bot-generated articles), the truth is that these bot-generated articles do not echo the complex structure of the Arabic language, do not reflect the cultural richness of the Arabic native speakers, and do not express the views of the Arabic native speakers.

We find that all the word embedding models that have been trained on the Arabic Wikipedia and Moroccan Arabic Wikipedia editions follow the same pattern, which is the models trained using the Arabic Wikipedia edition's corpora after eliminating the bot-generated articles, specifically when top K=1, perform better than the models trained on same corpora with bot-generated articles included, and of course, better than all models trained on the Moroccan Arabic Wikipedia edition's corpora, as shown in Tables 7 and 8. We believe when K=1 (the default value), the word embedding models usually show their actual performance, and as demonstrated in Table 7, it is obvious that the bot-generated articles negatively affect those word embedding models trained on them by widening the distance between words in the embedding space and that is why when we set K={5, 10}, those same word embedding models excel. We also find that all the masked language models trained on Arabic Wikipedia corpora perform better when all bot-generated articles are removed, indicating that, once again, the bots' creation or generation of articles negatively affects the masked language models, as demonstrated in Table 11.

Lastly, in this work, we strongly emphasize two points. First, we need both large and representative corpora to train NLP tasks and systems efficiently; neither alone is enough. The case of the Arabic Wikipedia editions gives a unique case study of this since the Moroccan Arabic Wikipedia edition is small but representative, and the Egyptian Arabic Wikipedia edition is large but unrepresentative. Second, removing many bot-generated articles from the Arabic Wikipedia corpora, for example, results in the same or even better performance. Due to the rise of generative models and for effective and safe training of NLP tasks and systems, we recommend avoiding using translated or generated corpora, especially when the goal is representation-based tasks like capturing the opinions or identifying the stances of Arabic native speakers.

7 Limitations

One limitation of our work is that while the three Arabic Wikipedia editions provide a unique example of our points, we cannot generalize the study and the impact of inorganic corpora for all the Wikipedia editions due to the lack of computational power needed to train the word embedding models and masked language models and due to the impracticality of creating or collecting factual datasets for the more than 300 languages that exist today on the Wikipedia project without using translation. Unlike the bots' quantifications process, the other limitation of our work is that the quantification of template-based translations is only specific to the Egyptian Arabic edition since the Wikipedia project does not track template-based translation in its metadata as it does with bot generation.

8 Conclusion and Future Work

In this work, we demonstrate that for good NLP performance, we need both large and organic corpora; neither alone is sufficient. We show that producing large corpora through automated means can be a counter-productive, producing models that both perform worse and lack cultural richness and meaningful representation of the Arabic language and its native speakers. Specifically, we demonstrate that training two key NLP upstream tasks, namely word representation and language modeling, on inorganic and unrepresentative corpora negatively impacts the performance of these NLP tasks. We find that the performance of these two NLP tasks is notably influenced by the way the training corpora are produced, where we observe that all models that have been trained on the template-translated corpora of the Egyptian Arabic edition perform the worst when compared with the more representative corpora like the Arabic Wikipedia edition. We also observe that many models perform the same or better when bot-generated articles are removed. Specifically, models trained on the Arabic Wikipedia edition (40% bot-generated articles) and Moroccan Arabic Wikipedia edition (12% bot-generated articles) perform the same or better when the bot-generated content is removed. In future work, we plan to expand our study of using unrepresentative corpora to include the societal implications (like gender bias and false representations) and security implications (like susceptibility to adversarial robustness) and hope to build a multi-level classification system to detect template-based translation activities such as those seen in the Egyptian Arabic Wikipedia edition.

Reproducibility

We share our code scripts, created datasets, extracted corpora, and trained models on GitHub at `https://github.com/SaiedAlshahrani/performance-implications`.

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.

B. Thomas Adler, Luca de Alfaro, Ian Pye, and Vishwanath Raman. 2008. Measuring Author Contributions to the Wikipedia. In *Proceedings of the 4th International Symposium on Wikis*, WikiSym '08, New York, NY, USA. Association for Computing Machinery.

Saied Alshahrani, Norah Alshahrani, and Jeanna Matthews. 2023. DEPTH+: An Enhanced Depth Metric for Wikipedia Corpora Quality. In *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*, pages 175–189, Toronto, Canada. Association for Computational Linguistics.

Saied Alshahrani, Esma Wali, and Jeanna Matthews. 2022. Learning from Arabic corpora but not always from Arabic speakers: A case study of the Arabic Wikipedia editions. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 361–371, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. AraBERT: Transformer-based Model for Arabic Language Understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France. European Language Resource Association.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2021a. AraELECTRA: Pre-Training Text Discriminators for Arabic Language Understanding. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 191–195, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2021b. AraGPT2: Pre-Trained Transformer for Arabic Language Generation. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 196–207, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Marzieh Babaeianjelodar, Stephen Lorenz, Josh Gordon, Jeanna Matthews, and Evan Freitag. 2020. Quantifying Gender Bias in Different Corpora. In *Companion Proceedings of the Web Conference 2020*, WWW '20, page 752–759, New York, NY, USA. Association for Computing Machinery.

Maher Asaad Baker. 2022. *How I Wrote a Million Wikipedia Articles*, 2 edition. BookRix GmbH Co. KG., Munich, Germany.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA. Curran Associates Inc.

Aylin Caliskan, Joanna J. Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186.

Yang Trista Cao, Yada Pruksachatkun, Kai-Wei Chang, Rahul Gupta, Varun Kumar, Jwala Dhamala, and Aram Galstyan. 2022. On the Intrinsic and Extrinsic Fairness Evaluation Metrics for Contextualized Language Representations. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 561–570, Dublin, Ireland. Association for Computational Linguistics.

Yan Chen, Christopher Mahoney, Isabella Grasso, Esma Wali, Abigail Matthews, Thomas Middleton, Mariama Njie, and Jeanna Matthews. 2021. Gender Bias and Under-Representation in Natural Language Processing Across Human Languages. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '21, page 24–34, New York, NY, USA. Association for Computing Machinery.

Won Ik Cho, Jiwon Kim, Jaeyeong Yang, and Nam Soo Kim. 2021. Towards Cross-Lingual Generalization of Translation Gender Bias. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 449–457, NYC, NY, USA. Association for Computing Machinery.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob

Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. PaLM: Scaling Language Modeling with Pathways.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*.

Manaal Faruqui, Yulia Tsvetkov, Pushpendre Rastogi, and Chris Dyer. 2016. Problems With Evaluation of Word Embeddings Using Word Similarity Tasks. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 30–35, Berlin, Germany. Association for Computational Linguistics.

Huang Huang, Fei Yu, Jianqing Zhu, Xuening Sun, Hao Cheng, Dingjie Song, Zhihong Chen, Abdulmohsen Alharthi, Bang An, Ziche Liu, Zhiyi Zhang, Junying Chen, Jianquan Li, Benyou Wang, Lian Zhang, Ruoyu Sun, Xiang Wan, Haizhou Li, and Jinchao Xu. 2023. AceGPT, Localizing Large Language Models in Arabic.

Hugging Face. 2023a. Fill-Mask. Last accessed on 2023-09-01.

Hugging Face. 2023b. Perplexity of fixed-length models. Last accessed on 2023-09-01.

Seonjun Kang, Xiaojin (Jim) Liu, Yeongin Kim, and Victoria Yoon. 2021. Can bots help create knowledge? The effects of bot intervention in open collaboration. *Decision Support Systems*, 148:113601.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR*, abs/1907.11692.

Maria Lopez-Medel. 2021. Gender bias in machine translation: an analysis of Google Translate in English and Spanish. *Academia.edu*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient Estimation of Word Representations in Vector Space. *arXiv arXiv:1301.3781v3*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

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.

Sergiu Nisioi, Ella Rabinovich, Liviu P. Dinu, and Shuly Wintner. 2016. A Corpus of Native, Non-native and Translated Texts. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation*, pages 4197–4201, Portorož, Slovenia. European Language Resources Association.

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.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Marcelo OR Prates, Pedro H Avelar, and Luís C Lamb. 2020. Assessing gender bias in machine translation: a case study with Google Translate. *Neural Computing and Applications*, 32:6363–6381.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked Language Model Scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Sarah Schröder, Alexander Schulz, Philip Kenneweg, Robert Feldhans, Fabian Hinder, and Barbara Hammer. 2021. Evaluating Metrics for Bias in Word Embeddings. *arXiv arXiv:2111.07864v1*.

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, Preslav Nakov, Timothy Baldwin, and Eric Xing. 2023. Jais and Jais-chat: Arabic-Centric Foundation and Instruction-Tuned Open Generative Large Language Models. *Inception, United Arab Emirates*.

Oleh Shliazhko, Alena Fenogenova, Maria Tikhonova, Vladislav Mikhailov, Anastasia Kozlova, and Tatiana Shavrina. 2022. mGPT: Few-Shot Learners Go Multilingual. *arXiv preprint arXiv:2204.07580*.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Kathleen S. Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Soraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Blaise Aguera-Arcas, Claire Cui, Marian Croak, Ed H. Chi, and Quoc Le. 2022. LaMDA: Language Models for Dialog Applications. *CoRR*, abs/2201.08239.

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

Milena Tsvetkova, Ruth García-Gavilanes, Luciano Floridi, and Taha Yasseri. 2017. Even good bots fight: The case of Wikipedia. *PloS one*, 12(2):e0171774.

Stefanie Ullmann and Danielle Saunders. 2021. Google Translate is sexist. What it needs is a little gender-sensitivity training. Last accessed on 2023-09-01.

Esma Wali, Yan Chen, Christopher Mahoney, Thomas Middleton, Marzieh Babaeianjelodar, Mariama Njie, and Jeanna Neefe Matthews. 2020. Is Machine Learning Speaking my Language? A Critical Look at the NLP-Pipeline Across 8 Human Languages. *arXiv preprint arXiv:2007.05872*.

Wikimedia Foundation. 2023. Wikimedia Downloads. Last accessed on 2023-09-01.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Lei (Nico) Zheng, Christopher M. Albano, Neev M. Vora, Feng Mai, and Jeffrey V. Nickerson. 2019. The Roles Bots Play in Wikipedia. *Proceedings of the ACM on Human-Computer Interaction*.

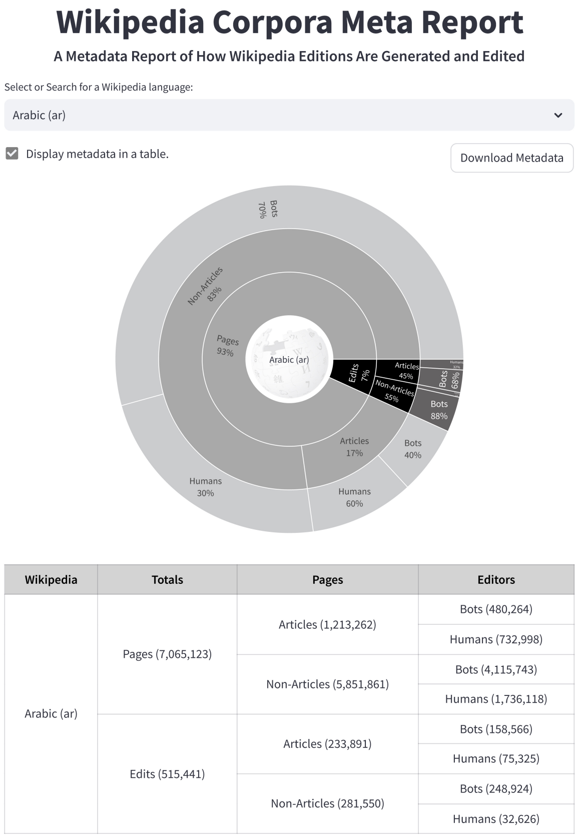
Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of LREC 2010 workshop New Challenges for NLP Frameworks*, pages 46–50, Valletta, Malta. University of Malta.

A Wikipedia Corpora Meta Report

We release the Wikipedia Corpora Meta Report as an online metadata report (dashboard), designed to shed light on how bots or humans generate or edit Wikipedia editions to provide the NLP community with detailed information (metadata) about each Wikipedia edition's articles, enabling them to make informed decisions regarding using these Wikipedia articles for training their NLP tasks and systems. As demonstrated in Figure 2, the dashboard interactively displays the metadata of each Wikipedia edition using sunburst visualization and provides users with the options to view the metadata in a tabular format and to download the displayed metadata as a CSV file. The dashboard is open-sourced on GitHub with an MIT license at https://github.com/SaiedAlshahrani/Wikipedia-Corpora-Report and publicly hosted on Streamlit Community Cloud at https://wikipedia-corpora-report.app. In the following subsections, we briefly describe the system of the dashboard, outline its architecture, and discuss its limitations.

A.1 System Description

The online Wikipedia Corpora Meta Report dashboard illustrates how humans and bots generate or edit Wikipedia editions, and calculates "pages" and "edits" metrics for all Wikipedia editions. The "pages" metric counts articles and non-articles, while the "edits" metric tallies edits on articles and non-articles, all categorized by contributor type: humans or bots. The dashboard dynamically displays these statistics using a sunburst visualization with three levels: metrics (pages or edits), sub-metrics (articles or non-articles), and contributors (bots or humans), showing numeric values and parent relationships at each level. Plus, the dashboard offers options to display metadata in a table format and allows users to download the metadata in CSV file format for their chosen Wikipedia edition/language.



Wikipedia Corpora Meta Report

A Metadata Report of How Wikipedia Editions Are Generated and Edited

Select or Search for a Wikipedia language:

Arabic (ar)

☑ Display metadata in a table.

Wikipedia	Totals	Pages	Editors
Arabic (ar)	Pages (7,065,123)	Articles (1,213,262)	Bots (480,264)
			Humans (732,998)
		Non-Articles (5,851,861)	Bots (4,115,743)
			Humans (1,736,118)
	Edits (515,441)	Articles (233,891)	Bots (158,566)
			Humans (75,325)
		Non-Articles (281,550)	Bots (248,924)
			Humans (32,626)

Figure 2: A screenshot of the online WIKIPEDIA CORPORA META REPORT dashboard, displaying a metadata report of how Modern Standard Arabic Wikipedia edition (AR) articles are generated and edited.

A.2 System Architecture

The WIKIPEDIA CORPORA META REPORT dashboard comprises both front–end and back–end components, each with distinct functionality. Figure 3 illustrates the dashboard's architecture and workflow, emphasizing each component and its role.

A.2.1 Front–end Components

The front–end components of this dashboard serve two specific functions: hosting the dashboard online for free public access and storing the metadata as a permanent Hugging Face dataset.

A.2.1.1 Streamlit Framework

We utilize the Streamlit Framework[8] to design, host, and deploy the dashboard on the free Streamlit Community Cloud[9] service, making it publicly accessible to everyone at https://wikipedia-corpora-report.streamlit.app.

A.2.1.2 Hugging Face Datasets

We use Hugging Face Datasets[10] as our database to store the processed metadata. Simultaneously, the dashboard retrieves the metadata dataset from the Hugging Face Hub. The metadata dataset is available at https://huggingface.co/SaiedAlshahrani/Wikipedia-Corpora-Report.

A.2.2 Back–end Components

The back–end components of this dashboard serve two specific functions: automatically updating the metadata dataset and triggering the metadata update procedure every 45 days.

A.2.2.1 Selenium WebDriver

We utilize the Selenium WebDriver[11] to automate the download of unprocessed metadata from the Wikimedia Statistics[12] service as CSV files. Then, we process the metadata and upload the processed metadata to the Hugging Face Hub as a dataset.

A.2.2.2 Unix/Linux Bash Daemons

We take advantage of the Streamlit Community Cloud being built on Debian Linux. We have written a Bash daemon that runs in the background and initiates the metadata update procedures. The daemon compares the original retrieval date from the pulled dataset with the system's current date, and when the time difference between these two dates exceeds 45 days, it triggers the update scripts.

A.3 System Limitations

The limitation of the WIKIPEDIA CORPORA META REPORT is that we use the Wikimedia Statistics service to quantify the contributions of bots and humans to a specific Wikipedia edition. Yet, these quantifications are calculated statistically, meaning users cannot determine which Wikipedia articles have been generated or edited by bots or humans.

B Best/Worst Word Embedding Models

We report that the `Word2Vec` (cbow) algorithm achieves the best accuracy when trained on substantially large corpora, like the Arabic and the Egyptian Arabic Wikipedia corpora (average accuracy: 69% and 25%, respectively), yet it does not when the corpora are very small, like the Moroccan Arabic Wikipedia corpora (average accuracy: 5%).

[8] Streamlit Framework: https://streamlit.io.

[9] Streamlit Community Cloud: https://streamlit.io/cloud.

[10] Hugging Face Datasets: https://huggingface.co/datasets.

[11] Selenium WebDriver: https://selenium.dev/webdriver.

[12] Wikimedia Statistics service: https://stats.wikimedia.org.

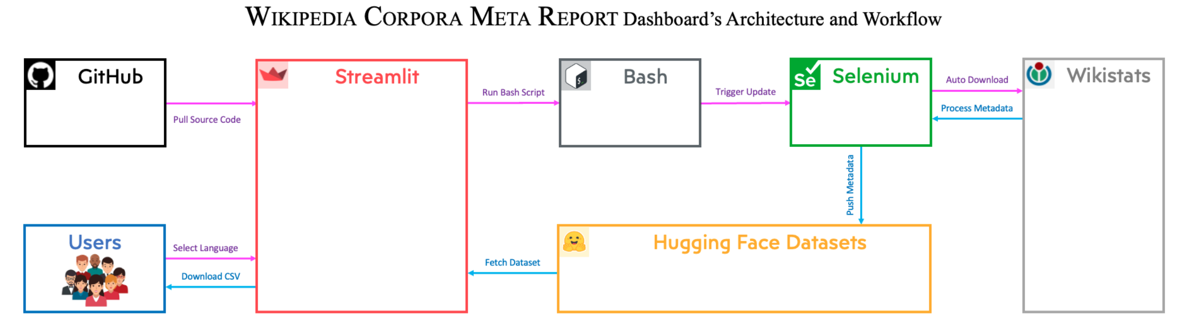


Figure 3: A diagram shows the Wikipedia Corpora Meta Report dashboard's architecture and workflow.

We also report that the GloVe algorithm achieves the lowest accuracy when trained on the Egyptian Arabic and the Moroccan Arabic Wikipedia corpora (average accuracy: 3% and 0.44%, respectively), yet it does the opposite when trained on corpora with lengthy articles, like the Arabic Wikipedia corpora (average accuracy: 47%).

C Pseudo-Perplexity Evaluations

Perplexity (PPL) is a commonly used metric to evaluate the performance of language models, yet this PPL metric is mostly suitable for the classic/causal language models that predict the next word in a sentence and not a well-defined metric for the masked language models (Hugging Face, 2023b). Therefore, we evaluate our MLM models using the well-designed metric for the MLMs, the Pseudo-Perplexity (PPPL) metric, which is proposed by Salazar et al. (2020), to intrinsically measure how well MLMs model a corpus of sentences. We find that the calculations of the PPPL are susceptible to the length of the sentences, and to ensure accurate measurements, we randomly choose 500 sentences with character lengths between 400 and 500 from each Arabic Wikipedia edition.

C.1 Impact of Template-based Translation

We calculate the PPPL scores for each MLM model, and in Table 12, we show the PPPL scores. We can see that the Arabic MLM (arRoBERTa$_{\text{BASE}}$) model, which has been trained on the Arabic Wikipedia edition, scores the best (the lower the PPPL score, the better the MLM model) with a PPPL score of 23.70, then the Egyptian Arabic MLM (arzRoBERTa$_{\text{BASE}}$) model with a PPPL score of 115.80, and lastly, the Moroccan Arabic MLM (aryRoBERTa$_{\text{BASE}}$) model with a very large PPPL score of 5,379.89. We attribute the high PPPL score of the aryRoBERTa$_{\text{BASE}}$ model to its very small training corpora (less than 6,500 articles) compared to the Arabic and Egyptian Arabic corpora. Still, we can also see a significant difference between the Arabic and the Egyptian Arabic MLMs' PPPL scores, indicating that even with a great number of articles, the documented template-based translation activity in the Egyptian Arabic Wikipedia edition seems to affect the performance of its MLM model.

MLM Model	Samples	Pseudo-Perplexity
arRoBERTa$_{\text{BASE}}$	500	23.70
arzRoBERTa$_{\text{BASE}}$	500	115.80
aryRoBERTa$_{\text{BASE}}$	500	5,379.89

Table 12: Pseudo-Perplexity scores of all the Arabic Wikipedia editions' MLM models.

C.2 Impact of Bot-based Generation

We evaluate our two MLM models (arRoBERTa$_{\text{BASE}}$ and aryRoBERTa$_{\text{BASE}}$) that have been trained on Arabic Wikipedia and Moroccan Arabic Wikipedia editions' corpora after excluding bot-generated articles using the PPPL metric. Table 13 displays that the PPPL measurements for the Arabic MML model (arRoBERTa$_{\text{BASE}}$) when trained once on corpora include bots activities, and trained another on corpora exclude bots activities. We can see that the Arabic MML model (arRoBERTa$_{\text{BASE}}$) trained on no bot-generated articles scores better than the Arabic MLM model trained on bot-generated articles (20.41 and 23.70, respectively). Whereas in the case of the Moroccan Arabic MLM model (aryRoBERTa$_{\text{BASE}}$), we have opposite results, and we attribute that to removing the bot-generated articles from its corpora, making it even smaller.

MLM Model	Corpora	Samples	Pseudo-Perplexity
arRoBERTa$_{\text{BASE}}$	With bots	500	23.70
	No bots		20.41
aryRoBERTa$_{\text{BASE}}$	With bots	500	5,379.89
	No bots		5,686.44

Table 13: Pseudo-Perplexity scores of the Arabic Wikipedia and Moroccan Arabic Wikipedia MLM models before and after excluding the bot-generated articles.

Octopus:

A Multitask Model and Toolkit for Arabic Natural Language Generation

AbdelRahim Elmadany[ξ,⋆] **El Moatez Billah Nagoudi**[ξ,⋆] **Muhammad Abdul-Mageed**[ξ,λ,⋆]
[ξ] Deep Learning & Natural Language Processing Group, The University of British Columbia
[λ]Department of Natural Language Processing & Department of Machine Learning, MBZUAI
`{a.elmadany,moatez.nagoudi,muhammad.mageed}@ubc.ca`

Abstract

Understanding Arabic text and generating human-like responses is a challenging endeavor. While many researchers have proposed models and solutions for individual problems, there is an acute shortage of a comprehensive Arabic natural language generation toolkit that is capable of handling a wide range of tasks. In this work, we present a novel Arabic text-to-text Transformer model, namely AraT5$_{v2}$. Our new model is methodically trained on extensive and diverse data, utilizing an extended sequence length of $2,048$ tokens. We explore various pretraining strategies including unsupervised, supervised, and joint pertaining, under both single and multitask settings. Our models outperform competitive baselines with large margins. We take our work one step further by developing and publicly releasing OCTOPUS, a Python-based package and command-line toolkit tailored for *eight* Arabic generation tasks all exploiting a *single* model. We release the models and the toolkit on our public repository.[1]

1 Introduction

Natural Language Generation (NLG) is a fundamental component of natural language processing that aims to generate human-like, coherent, contextually fitting, and linguistically precise text from structured data or various other input formats. NLG systems find applications in various aspects of daily life, including education, healthcare, business, and more. The recent emergence of generative models has significantly impacted the field of NLG. While important progress has been made in NLG research, the majority of existing tools, systems, and models are primarily focused on English (Jhaveri et al., 2019; Khan et al., 2021; Lauriola et al., 2022), leaving behind many languages, including Arabic.

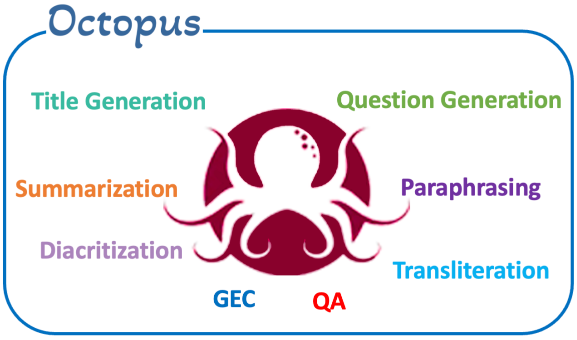


Figure 1: OCTOPUS is a jointly pretrained to cover eight NLG tasks, all shown in the illustration.

Although it is one of the most widely spoken languages in the world, and one with a rich linguistic structure and diverse dialects, Arabic remains underrepresented in NLG. One reason is the complex morphology and syntax of Arabic. Hence, the primary focus of our research here is to develop an advanced tool capable of performing several key Arabic NLG tasks. For example, we target tasks such as *text summarization*, *question answering*, *question generation*, *news headline generation*, and *paraphrasing*. These are tasks that necessitate a deep understanding of semantics, syntax, and pragmatics of Arabic. We also focus on tasks that require an understanding of both the syntax and morphology such as *diacritization*, *transliteration*, and *grammatical error correction*. Our main contributions are as follows:

1. We pretrain better and faster-to-converge versions of the text-to-text transformer model AraT5, collectively dubbed *AraT5$_{v2}$*. Compared to Nagoudi et al. (2022b), we train these new versions on a larger and more diverse dataset, as well as a larger sequence length.

2. To develop our models, we investigate diverse training strategies that integrate a combination of *supervised* and *unsupervised* training techniques.

[1] https://github.com/UBC-NLP/octopus
⋆Equal contributions

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 232–243
December 7, 2023 ©2023 Association for Computational Linguistics

3. We introduce OCTOPUS, a Python-based toolkit for eight Arabic NLG tasks. Our tool can be used as a strong baseline or as a core enabling technology that facilitates other developments.

4. We will make OCTOPUS publicly available to the research community.

2 Related Work

In the following section, we offer a concise overview of publicly available Arabic NLU and NLG tools, along with the Arabic and multilingual sequence-to-sequence (S2S) language models that we employ in this work.

2.1 Arabic NLP Tools

NLU tools. Numerous attempts have been made to develop tools for assisting with Arabic. Some tools focus on aspects such as morphosyntax, encompassing tasks like morphological analysis, disambiguation, part-of-speech tagging, and diacritization. Notable examples include Stanford CoreNLP (Manning et al., 2014), MADAMIRA (Pasha et al., 2014), Farasa (Darwish and Mubarak, 2016), and CAMeL tools (Obeid et al., 2020). Other tools, such as Mazajek (Farha and Magdy, 2019), and AraNet (Abdul-Mageed et al., 2019), are dedicated to social meaning tasks such as sentiment analysis, emotion detection, age and gender prediction, and sarcasm detection.

NLG Tools. Regarding Arabic NLG, as far as we know, the only publicly available tools are primarily focused on many-to-Arabic machine translation (MT). These include OPEN-MT (Tiedemann and Thottingal, 2020), NLLB (Costa-jussà et al., 2022), and Turjuman (Nagoudi et al., 2022d).

2.2 Arabic S2S Language Model.

Here, we overview the Arabic sequence-to-sequence models we employ as baseline in this work.

mT5. This is the multilingual version of T5 model (Raffel et al., 2019) introduced by Xue et al. (2020). Pretraining of mT5 is performed on the extensive mC4 (Multilingual Colossal Clean Crawled Corpus) which covers 101 languages, including Arabic.

mT0. Developed by Muennighoff et al. (2022), this is a group of S2S models ranging from 300M to 13B parameters trained to investigate cross-lingual generalization through multitask fine-tuning. The models are finetuned from pre-existing mT5 (Xue et al., 2020) multilingual language models using a cross-lingual task mixture called xP3.

AraBART. Introduced by (Eddine et al., 2022), this is a pretrained encoder-decoder model designed specifically for abstractive summarization tasks in the Arabic language. AraBART follows the architecture of BART (Lewis et al., 2019a) and has been pretrained on a 73GB of Arabic text data.

AraT5. Presented by Nagoudi et al. (2022c), this is an Arabic text-to-text Transformer model dedicated to MSA and Arabic dialects. It is similar in configuration and size to T5 (Raffel et al., 2019) and is trained on 248GB of Arabic text (70GB MSA and 178GB tweets). We now introduce our new model.

3 AraT5$_{\text{v2}}$

In this section, we present a novel version of AraT5, the Arabic-specific sequence-to-sequence model. We refer to this novel version as AraT5$_{\text{v2}}$. This new version represents a substantial evolution of the original AraT5$_{\text{v1}}$ model,[2] marked by notable improvements. These include **(1)** training on an expanded dataset comprising both labeled and unlabeled data, **(2)** larger sequence length of $2,048$ tokens, and **(3)** diverse training strategies that integrate a combination of unsupervised and supervised training techniques. Table 1 provides a comparison between AraT5$_{\text{v1}}$ and AraT5$_{\text{v2}}$.

Pretraining data. As we mentioned previously, our pretraining (unlabeled and labeled) dataset is linguistically diverse, covering all categories of Arabic (i.e., CA, DA, and MSA). as we will now describe.

3.1 Unlabled Data

We collect approximately 250GB of Arabic MSA text, which corresponds to around 25.6B tokens.[3] We use different sources including AraNews$_{\text{v2}}$ (Nagoudi et al., 2020), El-Khair (El-Khair, 2016), Gigaword,[4] OSIAN (Zeroual et al., 2019), Wikipedia Arabic, Hindawi Books,[5] OSCAR$_{\text{Egyptian}}$ (Suárez et al., 2019), and AraC4 (Nagoudi et al., 2022a).[6] To obtain Classical Arabic (CA) data, we utilize the Open Islamicate Texts Initiative (OpenITI) corpus (v1.6) (Nigst et al., 2020). The OpenITI corpus consists of 11K

[2] In this paper, we refer to the original AraT5 (Nagoudi et al., 2022b) as AraT5$_{\text{v1}}$.

[3] We note that AraT5$_{\text{v1}}$ trained only on 70GB MSA data.

[4] https://catalog.ldc.upenn.edu/LDC2009T30.

[5] https://www.hindawi.org/books.

[6] We note that AraC4 contains a diverse Arabic dialect as described in (Nagoudi et al., 2022a).

	AraT5$_{v1}$	**AraT5$_{v1}$-MSA**	**AraT5$_{v1}$-TWT**	**AraT5$_{v2}$**
Data size	248 GB	70 GB	178 GB	250 GB
Tokens count	29 B	7.1 B	21.9 B	25.6 B
Linguistic diversity	MSA, Tweets†	MSA	Tweets†	CA, DA, MSA
Sequence length	512	512	512	$2,048$

Table 1: Comparison between AraT5$_{v1}$ and AraT5$_{v2}$ models. It is worth noting that our new model (AraT5$_{v2}$) does not include tweets, whereas 71.77% of AraT5$_{v1}$ data is from Twitter (with the remaining 28.23% sourced from other sources). **CA:** Classical Arabic. **DA:** Dialectical Arabic. **MSA:** Modern Standard Arabic. Notably, Tweets† may encompass content in CA, DA, and MSA.

Islamic books, primarily collected from sources such as Shamela Library,[7] Al-Jami Al-Kabir collection (JK),[8] books digitized by the Jordanian publisher Markaz Al-Turāth, and the Shia Library.[9]

3.2 Labeled Data

Recently, Nagoudi et al. (2023) introduced *Dolphin*, an NLG benchmark for Arabic. Dolphin covers MSA, Classical Arabic, and various Arabic dialects. It is composed of 40 datasets, making it the largest and most diverse Arabic NLG benchmark. Due to the availability of the powerful Arabic machine translation toolkit, TURJUMAN (Nagoudi et al., 2022d), we shift our focus away from machine translation, code-switching, and Arabization tasks in this paper. Hence, we utilize datasets from eight out of the total thirteen NLG tasks in Dolphin. In the following sections, we will provide a brief description of each of these tasks.

(1) Diacritization. Is the computational procedure of adding missing diacritics or vowels to Arabic texts. For this task, we use the Arabic diacritization dataset presented by Fadel et al. (2019).

(2) Grammatical Error Correction. The GEC task is centered around the analysis of written text with the aim of automatically identifying and correcting a range of grammatical errors. We use three GEC datasets: QALB 2014 (Mohit et al., 2014), QALB 2015 (Rozovskaya et al., 2015), and ZAEBUC (Habash and Palfreyman, 2022).

(3) News Title Generation. The objective of this task is to generate a suitable headline for a given news article. To accomplish this, we use two datasets: Arabic NTG (Nagoudi et al., 2022c) and XLSum (Hasan et al., 2021).[10]

(4) Paraphrasing. In this task, we use four paraphrasing datasets: AraPara, a multi-domain Arabic paraphrase dataset (Nagoudi et al., 2022c), ASEP, an Arabic SemEval paraphrasing dataset (Cer et al., 2017), Arabic paraphrasing benchmark (APB) (Alian et al., 2019), and TaPaCo (Scherrer, 2020).[11]

(5) Question Answering. In this task, four publicly available extractive QA datasets are employed: ARCD (Mozannar et al., 2019) and the Arabic part of the following three multilingual datasets: MLQA (Lewis et al., 2019b), XQuAD (Artetxe et al., 2020), and TyDiQA (Artetxe et al., 2020).

(6) Question Generation. The goal of this task is to create simple questions that are pertinent to passages, along with their corresponding answers. For this, we utilize triplets consisting of *passages*, *answers*, and *questions*, all extracted from the same QA datasets.

(7) Text Summarisation. This task includes five publicly available datasets, including both Arabic and multilingual data: MassiveSum (Varab and Schluter, 2021), XLSum Hasan et al. (2021), CrossSum (Bhattacharjee et al., 2021), ANT (Chouigui et al., 2021), and MarSum (Gaanoun et al., 2022).

(8) Transliteration. This task involves converting words or text from one writing system to another while maintaining the original language's pronunciation and sound. Three datasets are used to create this component: ANETA (Ameur et al., 2019), ATAR (Talafha et al., 2021), and NETransliteration (Merhav and Ash, 2018).

4 Training Strategies

In this section, we describe the different strategies we use to pretrain and finetune AraT5$_{v2}$.

4.1 Unsupervised Pretraining.

Here, we focus on using only our unlabeled data (see Section 3.1) for pretraining our AraT5$_{v2}$.

[7] https://shamela.ws.

[8] http://kitab-project.org/docs/openITI.

[9] https://shiaonlinelibrary.com.

[10] We note that XLSum (Hasan et al., 2021) contains news articles that are annotated with both summaries and titles. For the NTG task, we use the pairs of articles and titles used to create the training data.

[11] We use the Arabic part only of TaPaCo.

The objective function does not rely on labels but instead imparts the model with transferable knowledge that can be effectively applied to various downstream tasks. We follow Raffel et al. (2019) in using a masked language modeling "*span-corruption*" objective. This approach involves replacing consecutive spans of input tokens with a mask token, and the model is trained to reconstruct the masked tokens.

4.2 Supervised Finetuning

We use the labeled data (see Section 3.2) to finetune the AraT5$_{\text{v2}}$ models under two settings: (i) *single task* and (ii) *multitask* finetuning.

Single task finetuning. We individually finetune our AraT5$_{\text{v2}}$ models on each of the eight NLG tasks we select from the Dolphin NLG benchmark (Nagoudi et al., 2023).

Multitask finetuning. We additionally explore multitask learning (Caruana, 1997; Ruder, 2017) using our AraT5$_{\text{v2}}$ models. This strategy involves training the model on several tasks concurrently, allowing the model and its parameters to be shared across all tasks. The ultimate goal is to enhance performance on each individual task over time. To indicate the intended task for the model, we incorporate a task-specific text "*prefix*" to the original input sequence before it is fed into the model. For example, for the paraphrase task, the source will be: *paraphrase:* امرأة تضيف التوابل إلى اللحم. The model should predict إمرأة تضيف المكونات إلى لحم البقر.

4.3 Joint Pretraining and Finetuning

In this scenario, we establish a uniform training objective for both pretraining and finetuning. The model is trained using a maximum likelihood objective, employing "*teacher forcing*" (Raffel et al., 2019; Williams and Zipser, 1989), regardless of the specific task.

5 Empirical Evaluation

5.1 Baselines

We evaluate our models across various scenarios, contrasting them with both multilingual and Arabic sequence-to-sequence pretrained language models. Specifically, we make use of mT5 (Xue et al., 2020) and mT0 (Muennighoff et al., 2022) as multilingual pretrained models; while comparing to AraBART (Eddine et al., 2022) and AraT5$_{\text{v1}}$ (Nagoudi et al., 2022b) as Arabic models. We evaluate our AraT5$_{\text{v2}}$ models (under different settings) and the selected baseline models on all eight NLG tasks (i.e., labeled data) described in Section 3.2.

5.2 Experimental Setup

For our experiments, we have two settings: one for the pretrained models and another for models we finetuning. We now describe each of these settings.

5.2.1 Pretrained Models

To pretrain our *AraT5$_{v2}$* model from scratch, we use the unsupervised pertaining strategy described in Section 4.1. We pretrain for one million steps on a Google TPU POD `v3-128`.[12] We employ a constant learning rate of $1e^{-3}$ and a dropout rate of 0.1. We use a batch size of $1,024$ with sequence length $2,048$. We further pretrain AraT5$_{\text{v2}}$ incorporating both unsupervised and supervised data (i.e., *joint strategy*; see Section 4.3), with the same hyperparameters for an additional 200K steps. We refer to the resulting model as *AraT5$_{v2}$-joint*.

5.2.2 Single Task Finetuning

We finetune both AraT5$_{\text{v2}}$ and AraT5$_{\text{v2}}$-joint, as well as baseline models, on the eight NLG tasks (20 datasets) for 20 epochs. We use a learning rate of $5e^{-5}$, a batch size of 8, and a maximum sequence length of 512.[13] In all single task experiments, we consistently select the best checkpoint for each model based on performance on the respective development set. Subsequently, we report performance of each model on the respective test set.

5.2.3 Multitask Finetuning

We extend the pretraining of *AraT5$_{v2}$* and *AraT5$_{v2}$-joint* with labeled data by an additional 100K steps for each model, all within the multitask finetuning setting. These experiments are conducted using a Google TPU POD `v3-128` with the same hyperparameters as the initial pretraining.[14] For model comparisons in the single task setting, we calculate the average of three runs of finetuned Arabic and multilingual models on the test sets of each task. However, for the joint and multitask models, we incorporate labeled data during the subsequent pretraining phase, employing a fixed number of steps—200K for the joint model and 100K for the multitask model. As a result, we conduct a single evaluation run for these models due to the high computation costs.

[12] https://sites.research.google/trc/about/

[13] For GEC, we use a maximum sequence length of $1,024$.

[14] *AraT5$_{v2}$-mTask* trains for a total of 1.1M steps, whereas *AraT5$_{v2}$-joint-mTask* undergoes training for 1.3M steps.

Task	Test Set	Metric	Baselines				AraT5$_{v2}$		AraT5$_{v2}$-Joint		
			mT0	mT5	AraBART	AraT5$_{v1}$†	sTask	mTask*	Joint*	sTask	mTask*
DIAC	ADT ↓	CER	$1.58^{\pm0.13}$	$1.64^{\pm0.11}$	$23.43^{\pm1.51}$	$2.58^{\pm0.19}$	**$1.30^{\pm0.20}$**	1.97	2.20	$1.90^{\pm0.24}$	1.74
GEC	QALB 2014	F$_{0.5}$ (M^2)	$65.86^{\pm0.67}$	$66.45^{\pm0.22}$	$68.67^{\pm0.08}$	$64.92^{\pm0.23}$	$70.52^{\pm0.15}$	62.36	62.36	**$70.73^{\pm0.27}$**	64.36
	QALB 2015 L1		$66.90^{\pm0.92}$	$66.68^{\pm0.08}$	$69.31^{\pm1.55}$	$64.22^{\pm0.82}$	$70.8^{\pm0.12}$	62.46	62.46	**$71.17^{\pm0.16}$**	64.93
	ZAEBUC		$47.33^{\pm3.34}$	$46.90^{\pm0.87}$	$82.08^{\pm7.54}$	$75.78^{\pm2.43}$	**$85.52^{\pm0.69}$**	37.89	42.25	$84.87^{\pm0.58}$	78.30
PARA	TAPACO	Belu	$15.43^{\pm0.64}$	$14.89^{\pm0.28}$	**$17.90^{\pm1.06}$**	$15.90^{\pm0.06}$	$16.82^{\pm0.41}$	11.73	10.39	$18.14^{\pm0.84}$	11.68
	APB		**$38.36^{\pm0.14}$**	$24.29^{\pm13.98}$	$37.66^{\pm1.01}$	$20.34^{\pm1.82}$	$35.04^{\pm0.89}$	19.57	16.92	$36.89^{\pm0.44}$	16.93
	SemEval		$20.49^{\pm0.13}$	$20.23^{\pm0.03}$	$24.52^{\pm0.62}$	$19.33^{\pm0.08}$	$25.52^{\pm0.58}$	**72.53**	68.57	$27.02^{\pm0.53}$	72.72
QA	ARCD$_{QA}$	F$_1$	$53.24^{\pm0.24}$	$51.63^{\pm1.01}$	$50.26^{\pm0.99}$	$58.12^{\pm0.16}$	$61.72^{\pm0.89}$	55.43	53.84	**$62.49^{\pm0.69}$**	54.81
	TyDiQA$_{QA}$		$76.31^{\pm0.09}$	$74.99^{\pm0.23}$	$73.32^{\pm1.21}$	$39.55^{\pm1.96}$	$82.99^{\pm0.47}$	72.37	71.72	**$84.21^{\pm0.47}$**	72.44
	XSQUAD$_{QA}$		$54.55^{\pm0.76}$	$47.43^{\pm0.91}$	$47.33^{\pm0.87}$	$48.71^{\pm0.5}$	$57.79^{\pm1.08}$	63.73	63.39	$59.42^{\pm0.72}$	**64.89**
	LMQA$_{QA}$		$49.17^{\pm0.34}$	$45.13^{\pm0.35}$	$47.24^{\pm0.13}$	$51.95^{\pm0.09}$	$54.48^{\pm0.12}$	47.50	46.63	**$55.02^{\pm0.26}$**	48.70
QG	ARCD$_{QG}$	Belu	$17.73^{\pm0.99}$	$17.62^{\pm2.1}$	$22.79^{\pm0.66}$	$16.8^{\pm1.32}$	**$24.13^{\pm0.20}$**	19.86	19.23	$22.48^{\pm1.30}$	21.54
	TyDiQA$_{QG}$		$30.22^{\pm0.91}$	$31.0^{\pm0.97}$	$33.64^{\pm0.13}$	$22.09^{\pm1.85}$	$33.50^{\pm0.75}$	25.37	24.50	**$34.05^{\pm0.34}$**	26.18
	XSQUAD$_{QG}$		$10.04^{\pm0.01}$	$9.96^{\pm0.03}$	$10.27^{\pm0.31}$	$9.21^{\pm0.09}$	$10.98^{\pm6.91}$	6.65	1.94	**$11.50^{\pm0.41}$**	7.30
	MLQA$_{QG}$		$6.04^{\pm0.08}$	$6.00^{\pm0.38}$	$7.02^{\pm0.09}$	$6.12^{\pm0.42}$	**$7.56^{\pm0.27}$**	3.96	3.25	$7.28^{\pm0.11}$	3.66
SUM	XLSum	Rouge$_L$	$21.46^{\pm0.54}$	$20.64^{\pm0.31}$	$26.64^{\pm0.04}$	$22.71^{\pm1.36}$	$27.15^{\pm0.09}$	63.59	52.25	$28.12^{\pm0.12}$	**65.66**
	CrossSum		$21.00^{\pm0.38}$	$20.29^{\pm0.01}$	$25.89^{\pm0.09}$	$22.14^{\pm1.53}$	$26.57^{\pm0.06}$	59.45	50.82	$27.56^{\pm0.06}$	**61.31**
	MarSum		$23.00^{\pm0.17}$	$22.57^{\pm0.21}$	$26.49^{\pm0.03}$	$21.71^{\pm0.39}$	$26.64^{\pm0.06}$	20.49	19.04	**$26.81^{\pm0.06}$**	20.78
	MassiveSum		$25.57^{\pm0.11}$	$22.88^{\pm0.12}$	$30.0^{\pm0.11}$	$15.89^{\pm0.4}$	$23.00^{\pm0.00}$	27.22	25.75	**$27.69^{\pm0.07}$**	26.97
	ANTCorp		$90.29^{\pm0.11}$	$88.84^{\pm0.91}$	$90.0^{\pm0.20}$	$86.64^{\pm0.22}$	$90.94^{\pm0.14}$	87.39	86.92	$90.85^{\pm0.12}$	88.22
TG	Arabic NTG	Bleu	$19.03^{\pm0.34}$	$19.23^{\pm0.01}$	$22.75^{\pm0.09}$	$19.55^{\pm0.16}$	$22.13^{\pm0.08}$	22.54	21.33	$22.37^{\pm0.06}$	**22.94**
	XLSum		$6.50^{\pm0.17}$	$6.51^{\pm0.11}$	$8.98^{\pm0.18}$	$7.44^{\pm0.11}$	$9.59^{\pm0.17}$	6.21	5.91	**$9.82^{\pm0.14}$**	6.11
TR	ANTAEC ↓	CER	$19.21^{\pm0.48}$	$18.93^{\pm0.30}$	$18.29^{\pm0.29}$	$20.74^{\pm0.17}$	**$18.06^{\pm0.21}$**	31.50	33.00	$19.25^{\pm0.06}$	31.66
	ATAR ↓	CER	$16.79^{\pm0.15}$	$16.68^{\pm0.22}$	$17.70^{\pm0.05}$	$36.51^{\pm1.53}$	$14.96^{\pm0.05}$	33.63	35.90	**$14.70^{\pm0.05}$**	33.19
	NETTrans	Belu	$55.70^{\pm0.18}$	$55.02^{\pm0.47}$	$54.15^{\pm0.75}$	$51.89^{\pm0.64}$	**$58.33^{\pm0.70}$**	43.69	42.65	$57.81^{\pm0.66}$	43.18
		H-Score ↑	37.01	35.42	39.86	34.59	41.90	41.41	38.73	42.56	**42.89**
		L-Score ↓	12.53	12.42	19.81	19.94	**11.44**	22.37	23.70	11.95	22.20

Table 2: Average of three runs of finetuned Arabic and multilingual models on OCTOPUS test. **L-Score**: refers to the macro-average scores of tasks where a lower score ↓ is better. **H-Score**: refers to the macro-average scores of tasks where a higher score ↑ is better. OCTOPUS task clusters taxonomy: (DIAC, Diacritization), (GEC, Grammatical Error Correction), (PARA, Paraphrase), (QA, Question Answering), (QG, Question Generation), (SUM, Summarization), (TG, News Title Generation), and (TR, Transliteration). †We refer to vanilla AraT5 (Nagoudi et al., 2022b) as AraT5$_{v1}$. *For the *joint* and *multitask* models, we utilize the labeled data during the further pretraining phase. Consequently, we employ it only once, as opposed to the regular single fine-tuning, which involves three runs. **Bold and green:** best score in the individual task. **Bold and orange:** best average scores over all tasks.

5.3 Evaluation Metrics

We present the results of our models and the baseline models independently on each task of evaluated datasets, using the relevant metric. We employ Bleu score as an evaluation metric for paraphrase, question generation, title (i.e. headline news) generation, and sentence-level transliteration tasks. Additionally, we use Rouge$_L$, F$_1$, and F$_{0.5}$ (M^2) as evaluation metrics for summarization, question answering, and grammatical error correction, respectively. For diacritization and word-level transliteration datasets, we utilize the character error rate (CER) metric. We split the evaluation scores into "L-Score" where lower ↓ is better (e.g., CER) and "H-Score" where higher ↑ is better, i.e., Bleu, F$_1$, F$_{0.5}$, and Rouge$_L$.

5.4 Results

Table 2 shows that our proposed models, across different settings, outperform the baseline models in ∼ 90% of the individual test sets (18 out of 20). Notably, AraT5$_{v2}$ significantly outperforms the vanilla AraT5$_{v1}$ (Nagoudi et al., 2022b) by 7.3 and 8.58 points in terms of the macro-average scores for tasks where *higher* (↑) and *lower* (↓) score is better, respectively. Furthermore, AraT5$_{v2}$ markedly outpaces the second-ranked baseline model, AraBART, by an average of 2.04 (↑) and 8.45 (↓) in the macro-average scores.

Additionally, the AraT5$_{v2}$-joint single-task model achieves the highest score in 8 out of 20 (∼ 40%) for the individual tasks, followed by the AraT5$_{v2}$ models and the AraT5$_{v2}$-joint multitask model, each achieving the best score in 4 out of 20

Input text	الخيل والليل والبيداء تعرفني *** والسيف والرمح والقرطاس والقلم
Target	الخَيْلُ وَاللّيْلُ وَالبَيْداءُ تعرِفُني *** وَالسّيفُ وَالرّمحُ والقرْطاسُ وَالقَلَمُ
Multitask model	الْخَيلِ وَاللَّيْل وَالْبِيلادْ تَعْرِفَني *** وَالسَّيْفُ وَالرُّمُح وَقِرْطَاسُ وَالْقَلمِ
Single task model	الخَيْلُ وَاللَّيْلُ وَالْبَيْدَاءُ تَعْرِفُنِي *** وَالسّيْفُ وَالرُّمُحُ وَالْقِرْطَاسُ وَالْقَلَمُ
Input text	إبراهيم بن كنيف النبهاني، شاعر إسلامي، اشتهر بأبيات له أولها تعز فإن الصبر بالحر أجمل *** وليس على ريب الزمان معول تناقلت كتب الأدب أبياته وهو من شعراء الحماسة.
Target	إبراهيمُ بن كُنَيْفٍ النَّبْهانِيُّ، شاعرٌ إسلاميٌّ، اشْتُهِرَ بِأَبْياتٍ لَهُ أَوَّلُها تَعزَ فَإِنَ الصَبْرَ بالحُرِ أَجْمَلُ *** وَلَيْسَ عَلَى رَيْبِ الزَمَانِ مُعَوَل تناقَلَتْ كُتُبُ الأدبِ أبياتَهُ وهُوَ مِنْ شُعَراءِ الحَماسَةِ.
Multitask model	إبراهيم بن كَنِيسٍ النَّبَهَانِيِّ، شَاعٌ إِسْلَامِيٌّ، أُشْتُهِرَ بِأَبْيَاتٍ لَهُ أَوَّلُهَا تَعَزّ فَإِنَّ الصَّبْ بِالْأَحْرِّ أَجْمَلُ *** وَلَيْسَ عَلَى رَيْبِ الزَّمَانِ مَعْوَلٌ تَنَاقَلَتْ كُتُبُ الأَدَبِ أَبْيَاتِهِ وَهُوَ مِنْ شُعَبَاءِ الْحُمَاسَةِ.
Single task model	إِبْرَاهِيمُ بْنُ كُنَيْفٍ النَّبْهَانِيُّ، شَاعِرٌ إِسْلَامِيٌّ، أُشْتُهِرَ بِأَبْيَاتٍ لَهُ أَوَّلُهَا تَعُزْ فَإِنَّ الصَّبْرَ بِالْحُرِّ أَجْمَلُ *** وَلَيْسَ عَلَى رَيْبِ الزَّمَانِ مُعَوَّلٌ تَنَاقَلَتْ كُتُبُ الأَدَبِ أَبْيَاتِهِ وَهُوَ مِنْ شُعَرَاءِ الْحِمَاسَةِ.

Table 3: Examples of negative task interference in the **diacritization task**, both in a single-task and multitask. **Color taxonomy**: "**blue**" refers to the original text, "**red**" denotes a word-level error, "**light red**" indicates a partial diacritization error on one more letter, and "**green**" signifies correctness. For single task, we use "*AraT5$_{v2}$-sTask*" whereas we use "*AraT5$_{v2}$-joint-mTask*" model as the multitask model.

($\sim$ 20%) tasks. It is also noteworthy that AraBART and mT0 each obtain the best score in only one task.

5.5 Discussion

Exploring different pretraining settings allows us to derive unique insights. Examples of insights that can be gleaned from Table 2 include:

Addressing open-domain problems. We observe that sequence-to-sequence models like T5 encounter challenges when tackling open-domain question-answering tasks. For example, the results on the MLQA dataset demonstrate notably low performance across all evaluated models.

Handling lengthy sentences. Multitasking proves effective in addressing challenges when working with long texts, such as paragraphs or documents. It significantly excels in tasks involving long sequences. For instance, paraphrasing text such as the SmEval dataset and abstractive summarization like ARCD and XLSum all include long sequences. Conversely, it does not lead to significant improvements in short-text paraphrasing, such as those at the sentence level in datasets like APPB and TAPACO.

Negative task inference. Notably, multitask training in our experiments has a negative impact on character-level tasks. For instance, we randomly select two examples from an Arabic poetry website[15], remove diacritics from the input text, and require both the AraT5$_{v2}$-joint multitask and AraT5$_{v2}$ single task models to diacritize these examples. As shown in Table 3, the multitask model alters the words themselves, while the single task model preserves the input words (i.e., it focuses solely on adding diacritization to the character sequences).

5.6 Performance Comparison

One of our primary objectives in developing a new version of AraT5 is to improve the time required for the finetuning process (i.e., convergence time). Therefore, we conduct a comparison between AraT5$_{v1}$ and AraT5$_{v2}$, as well as the baselines models in this respect. This allows us to analyze their computational efficiency and gain

[15] https://poetry.dctabudhabi.ae/

News Article	
	أكد النجم البرازيلي نيمار مهاجم نادي الهلال أن الدوري السعودي بات أكثر قوة من الدوري الفرنسي مذكراً الجميع بتجربته في الأخيرعندما انتقل إلى باريس سان جيرمان صيف ٢٠١٧. وأوضح نيمار خلال مؤتمر صحفي مقام في بارا البرازيلية لدى سؤاله عن الدوري السعودي: وأؤكد لك أن كرة القدم هي نفسها ، الكرة هي نفسها و يسجلون الأهداف و بالنظر إلى الأسماء فإن الدوري السعودي بات أقوى من الدوري الفرنسي. التدريبات هناك شديدة ونتعطش أنا وزملائي للفوز هناك بشكل كبير والتتويج مع الهلال. وأضاف: الجميع اعتقد أن الدوري السعودي ضعيف والأمر نفسه حدث معي عندما انتقلت إلى الدوري الفرنسي، حينها ظن الناس الأمر نفسه لكني لم أضرب في حياتي من قبل المدافعين أكثرمن هناك. وأبان حول الدوري السعودي: اللاعبون الذين يلعبون هناك يعلمون مدى صعوبة اللعب في الدوري السعودي وأنا متأكد أنه لن يكون أمرا سهلا الفوز بالمسابقة لأن الفرق عززت صفوفها بلاعبين جدد، وستكون بطولة ممتعة وشيقة جدا. وتلعب البرازيل أمام بوليفيا في بارا البرازيلية يوم السبت قبل أن تواجه بيرو يوم الأربعاء ضمن تصفيات كأس العالم لمنتخبات أميركا الجنوبية.
Title Generation	
Output	نيمار: أعرف ماذا يعني اللعب في الدوري السعودي نيمار: الدوري السعودي أقوى من الفرنسي نيمار: أعرف ماذا يعني الفوز بالمباريات في الدوري السعودي نيمار: أعرف ماذا يعني أن الدوري السعودي أقوى من الفرنسي نيمار: أعرف أن الدوري السعودي أقوى من الدوري الفرنسي
Question Answering	
Question no. 1	متي تقام مباراة بوليفيا و البرازيل؟
Output	السبت
Question no. 2	متي انتقل نيمار الي باريس سان جيرمان؟
Output	صيف ٢٠١٧
Question Generation	
Answer	تلعب البرازيل أمام بوليفيا في بارا البرازيلية يوم السبت
Output	من يقابل البرازيل في تصفيات كأس العالم ؟

Table 4: OCTOPUS output examples based on a randomly picked article from a news website. We prompt OCTOPUS to generate five potential titles, answers based on the questions, and questions for the provided answer.

insights into their convergence behavior. To quantify this, we measure the required average time for convergence (in hours) and the average number of epochs needed to achieve convergence based on model results on development datasets. For a fair comparison, we finetune all models for a maximum of 20 epochs across all tasks. Notably, the evaluation results carry on the average of three separate runs using three different seeds, thereby enhancing the robustness and reliability of our comparison.

Convergence time. In general, we observe that AraBART and AraT5$_{\mathrm{v2}}$ need on average 12 and 13 epochs, respectively, till convergence compared to AraT5$_{\mathrm{v1}}$, which needs an average of 16 epochs to achieve the best performance. So, we notice that AraBART requires only 2.9 hours to converge and achieve the optimal performance, while AraT5$_{\mathrm{v2}}$, and AraT5$_{\mathrm{v1}}$, need an average of 3.77 and 5.20 hours, respectively, to reach the best score. So, we observe that AraT5$_{\mathrm{v2}}$ is approximately 35.19% faster than AraT5$_{\mathrm{v1}}$ in terms of training times.[16]

6 OCTOPUS Toolkit

6.1 Model Selection

Our objective is to introduce a versatile language generation toolkit capable of handling a wide range of tasks, all within a single model. To achieve this goal, we have explored multiple training strategies, as described in Section 4. Based on our empirical evaluations, we observe that finetuned *AraT5$_{v2}$-joint* under the multitask setting outperforms the other models.[17] Therefore, we utilize this model as the foundation for developing our OCTOPUS toolkit (illustrated in Figure 1).

[16] We note that AraBART requires a shorter time to converge due to the vocabulary size of the model (32K) compared to (110K) for AraT5-based models.

[17] As Table 2 shows, *AraT5$_{v2}$-joint-mTask* outperforms other models where a higher score is better and remains highly competitive in scenarios where a lower score is preferred.

	Paraphrase
Input	بالتزامن مع استنفار ليبيا لاسيما في الشرق جراء الإعصار آدانيالْ الذي ضربها، وصلت العاصفة دانيال أيضا إلى مصر وضربت محافظتين في شمال البلاد، هما مرسى مطروح والإسكندرية فقد غطت العاصفة مناطق الحدود المصرية غرب البلاد، وشملت مدن السلوم وسيوة ومطروح والإسكندرية. وبينت القنواتالمختلفة لصور الأقمار الصناعية شكل العاصفة التي طرقت الأبواب المصرية. وأظهرت طبقات من السحب المختلفة والأتربة المصاحبة لها.
Output	بالتزامن مع حالة التأهب في شرق ليبيا بسبب إعصار دانيال، وصلت العاصفة إلى مصر لتغطي محافظتين شمال البلاد، وهما مرسى مطروح والإسكندرية. وأظهرت مختلف صور الأقمار الصناعية شكل العاصفة التي ضربت ليبيا، والتي غطت إلى حد كبير شرق مصر، ولا سيما في مدينتي سرت والسلوم، كما أظهرت مختلف طبقات الغيوم والغابات المصاحبة لها.
	Summarization
Input	اعترف نضال حسن، الذي يدافع عن نفسه، بقتل الجنود، متحججا بحماية المسلمين وعناصر طالبان في أفغانستان، ولكن القاضي العسكري رفض حجته «بحماية الآخرين». وإذا أدين حسن، البالغ من العمر ٤٢ عاما، بقتل ١٣ شخصا وجرح آخرين فإنه سيواجه عقوبة الإعدام. ويعتبر الحادث الأكثر دموية من بين الهجمات غير القتالية التي وقعت في قاعدة عسكرية أمريكية. وقال شهود عيان دخل في ٥ نوفمبر الثاني عام ٢٠٠٩ مصحة تعج بالجنود الذين كانوا ينتظرون أدوارهم إجراء فحوصات طبية أو التلقيح، ثم صعد على مكتب، وأطلق النار من سلاحين بيديه، دون توقف إلا لإعادة تعبئة السلاح. مواضيع قد تهمك نهاية وسيقدم ممثلو الادعاء أدلة تفيد بأن حسن مال إلى الأفكار المتطرفة، وكان يزور المواقع بحثا عن آالجهاديين» وطالبان، ساعات قبل الهجوم. وكان الرائد حسن سيلتحق بالقوات الأمريكية في أفغانستان قبل أن ينفذ هجومه. «عنف في مكان العمل» وصنفت وزارة الدفاع الأمريكية الحادث باعتباره «عنفا في مكان العمل» بدلا من تصنيفه «عملا إرهابيا»، وهو ما أغضب عائلات الضحايا، حسب ما أفاد به مراسل بي بي سي، نك براينت، في فروت هود. ويتوقع أن يدلي العديد من جرحى الحادث بشاهاداتهم أمام المحكمة. وسيواجه حسن عددا من ضحاياه في قاعة المحكمة لأنه سيتولى الدفاع عن نفسه. وهو يستخدم كرسيا متحركا لأنه أصيب بالشلل، عندما أطلق عليه شرطي في القاعدة العسكرية النار.
Output	تنظر محكمة عسكرية أمريكية في وقت لاحق من اليوم في قضية الطبيب النفسي العسكري الأمريكي، نضال حسن، الذي اعترف بقتل ١٣ شخصا وإصابة أكثر من ثلاثين آخرين في إطلاق نار بقاعدة فورت هود منذ أربعة أعوام.
	Grammatical Error Correction
Input	لا زال كبير الشبيحه يظن ان ارواح وآلام الناس اقل كلفه من تخليه عن منصبه ، فلذلك اذا كان السوريون لا يرتضون بهذه المعادله المهينه ، فعليهم ان يهبوا هبه قويه واحده وياخذو حقوقهم من هذه العصابه عنوه ، اننا يا أحبائي ندفع ثمن اكثر من اربعين عام ومن الخنوع والذل والثمن سيكون غاليا ولكنه يستأهل هذه التضحيات
Output	لازال كبير الشبيحة يظن أن أرواح وآلام الناس أقل كلفة من تخليه عن منصبه ، فلذلك إذا كان السوريون لا يرتضون بهذه المعادلة المهينة ، فعليهم أن يهبوا هبة قوية واحدة ، ويأخذوا حقوقهم من هذه العصابة عنوة . إننا يا أحبائي ندفع ثمن أكثر من أربعين عام ، ومن الخنوع والذل ، والثمن سيكون غاليا ، ولكنه يستأهل هذه التضحيات .

Table 5: Octopus output examples for *grammatical error correction*, *paraphrasing*, and *summarization*.

6.2 Task Coverage

Octopus is designed for *eight* machine generation tasks, encompassing diacritization, grammatical error correction, news headlines generation, paraphrasing, question answering, question generation, and transliteration. This comprehensive package includes a Python library along with associated command-line scripts. Table 4 illustrates the output of Octopus, generating five potential titles, answers derived from questions related to the content, and questions corresponding to a provided answer based on a randomly selected article from a news website. Moreover, Table 5 showcases examples of Octopus for grammatical error correction, paraphrasing, and summarization. We now describe the intricacies of implementation and design of the Octopus toolkit, along with its various configurable settings.

6.3 Implementation

We distribute Octopus as a modular toolkit built using standard libraries including PyTorch (Paszke et al., 2019) and HuggingFace (Lhoest et al., 2021). It is implemented in Python and can be easily installed using the pip package. It is compatible with Python versions 3.8 and later, Torch version 2.0 and later, and the HuggingFace Transformers library version 4.30 or higher.[18] We offer three usage options with varieties of arguments: *(i) Command-Line Interface (CLI)*, *(ii) Python integration package*, and *(iii) an interactive web interface*.

CLI ommands. We offer three command-line interfaces for task selection and output generation as follows: First, the "*octopus_interactive*" command provides an interactive mode that allows users to actively engage with the system. With this command, users can efficiently select their desired task and input text and then apply the chosen task to generate output. For instance, if a user wants to diacritize several sentences, they can initiate the diacritization task and input the sentences one by one to undergo the diacritization process. Second,

[18] Installation instructions and documentation can be found at: https://github.com/UBC-NLP/octopus.

	Argument	Description
Basic	- - *help* [-*h*] - - *cache-dir* [-*c*] - - *logging-file* [-*l*]	To display the arguments details Specify the path to the cache directory. Define the file path for logging.
Task	- - *prefix* [-*p*]	Task prefix should be one of the following: [’*diacritize*’, ’*correct_grammar*’, ’*paraphrase*’, ’*answer_question*’, ’*generate_question*’, ’*summarize*’,’*generate_title*’, ’*translitrate_ar2en*’, ’*translitrate_en2ar*’]
Input & Output	- - *text* [-*t*] - - *input-file* [-*f*] - - *max-outputs* [-*o*] - - *batch-size* [-bs] - - *seq-length* [-*s*]	Provide the input text for generative tasks. Specify the path of the input file. Define the number of hypotheses to generate as output. Set the number of input sentences processed in a single iteration. Specify the maximum sequence length for the generative text.
Decoding	- - *search-method* [-*m*] - - *nbeam* [-*nb*] - - *no-repeat-ngram-size* [-*ng*] - - *top-k* [-*k*] - - *top-p* [-*p*]	Choose the decoding method from the options [‘*greedy*’, ‘*beam*’, ‘*sampling*’]. If using beam search, specify the beam search size. Avoid repeating the same n-gram size in the generated text. Utilize sampling with a top-k strategy. Implement sampling with a top-p strategy.

Table 6: OCTOPUS command line argument list.

the main command “*octopus*” offers two options: users can either directly input the text or specify a file path, allowing flexibility in applying multiple tasks to a large amount of data points. Finally, the task-specific command “*octopus-taskname*” offers seven task-specific commands, each corresponding to one of the supported tasks. For instance, there are “*octopus-diacritize*” and “*octopus-paraphrase*” commands. These task-specific commands follow the same usage pattern as the “octopus” command, but are designed for individual tasks.

Python integration package OCTOPUS is a Python library that offers numerous functions for seamless integration with various dataframe architectures, including `Pandas`, `PySpark`, `Dask`, and more. It takes as input the function to be integrated into user code and returns both generative text and processing logs.

Interactive web interface. We offer a dynamic interactive web interface that allows users to try OCTOPUS tasks. Furthermore, to facilitate adoption, we provide a `Google Colab` notebook with detailed instructions on how to use the OCTOPUS tool and model, and integrate them with user’s code.

6.4 Arguments

Each of the command lines (i.e., *octopus-interactive, octopus, or octopus-taskname* supports or requires several arguments. Furthermore, OCTOPUS supports four decoding methods on the decoder side: *greedy search*, *beam search* (Koehn, 2009), *top-k sampling* (Fan et al., 2018), and *nucleus sampling* (Holtzman et al., 2019). We set as the default setting *beam search* with a beam size of 5, and a maximum sequence length of 2, 048. Table 6 shows detailed descriptions of the arguments and their usage. This information helps users understand and utilize the provided arguments effectively.

7 Conclusion

We introduced a suite of powerful Arabic text-to-text Transformer models trained on large and diverse datasets, with an extended sequence length of up to 2, 048. We also explored various pretraining strategies, including unsupervised and joint pertaining, using both single and multitask settings. Our models outperform competitive baselines, demonstrating their effectiveness. Furthermore, we introduced OCTOPUS, a publicly available Python-based package and command-line toolkit tailored for *eight* Arabic natural language generation tasks. OCTOPUS is designed to be extensible, and we plan to expand its capabilities by adding more tasks and increasing the capacity of our back-end model.

8 Limitations

We identify the following limitations:

- **Dialectal Arabic**. In this paper, our primary focus is on MSA tasks. Nevertheless, we are committed to expanding our scope to cover tasks in available Arabic dialects in the future. Currently, there is a recognized necessity within the community to facilitate the creation of datasets tailored to multiple Arabic dialects. For example, there is currently a deficiency in dialectal resources for sequence-to-sequence tasks such as summarization, paraphrasing, and question-answering. As more resources are created for dialects covering these tasks, we anticipate enhancing the coverage and capabilities of OCTOPUS exploiting these resources. Fortunately, our toolkit and core back-end models are extensible and hence would allow for such a development seamlessly.

- **Task Coverage**. OCTOPUS currently encompasses only eight generation tasks. However, we have plans to expand its capabilities by including additional tasks. These upcoming additions can involve, for example, dialgoue geeration and tasks involving code-switching. Again, adding more tasks to OCTOPUS will not be onerous, once respective datasets are available.

- **Intended Use**. OCTOPUS is a natural language generation toolkit designed to handle eight different tasks. We have tried the toolkit under different scenarios and found it to perform well. However, before any real-world usecases, we strongly encourage further and more extensive evaluations under diverse conditions.

9 Ethical Considerations

Our pretraining datasets are sourced from the public domain. Similarly, the labeled datasets used for model finetuning have been collected from publicly available data, made possible through the dedicated efforts of numerous researchers over the years. Consequently, we do not have significant concerns regarding the retrieval of personal information from our trained models. It is essential to note that the datasets we gather to construct OCTOPUS may contain potentially harmful content. Furthermore, during model evaluation, there is a possibility of exposure to biases that could lead to unintended content generation. For release, all our pretrained models and the toolkit are publicly available for non-malicious use.

Acknowledgments

We acknowledge support from Canada Research Chairs (CRC), the Natural Sciences and Engineering Research Council of Canada (NSERC; RGPIN-2018-04267), the Social Sciences and Humanities Research Council of Canada (SSHRC; 435-2018-0576; 895-2020-1004; 895-2021-1008), Canadian Foundation for Innovation (CFI; 37771), Digital Research Alliance of Canada,[19] and UBC ARC-Sockeye.[20] We thank the Google TFRC program for providing us with free TPU access.[21]

References

Muhammad Abdul-Mageed, Chiyu Zhang, Azadeh Hashemi, and El Moatez Billah Nagoudi. 2019. AraNet: A Deep Learning Toolkit for Arabic Social Media. *arXiv preprint arXiv:1912.13072*.

Marwah Alian, Arafat Awajan, Ahmad Al-Hasan, and Raeda Akuzhia. 2019. Towards building arabic paraphrasing benchmark. In *Proceedings of the Second International conference on Data Science E-learning and Information Systems (DATA' 2019)*, pages 1–5.

Mohamed Seghir Hadj Ameur, Farid Meziane, and Ahmed Guessoum. 2019. Anetac: Arabic named entity transliteration and classification dataset. *arXiv preprint arXiv:1907.03110*.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Uddin Ahmad, Yuan-Fang Li, Yong bin Kang, and Rifat Shahriyar. 2021. Crosssum: Beyond english-centric cross-lingual abstractive text summarization for 1500+ language pairs. *CoRR*, abs/2112.08804.

Rich Caruana. 1997. Multitask learning. *Machine learning*, 28(1):41–75.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*.

Amina Chouigui, Oussama Ben Khiroun, and Bilel Elayeb. 2021. An arabic multi-source news corpus: Experimenting on single-document extractive summarization. *Arabian Journal for Science and Engineering*, 46:3925–3938.

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Kareem Darwish and Hamdy Mubarak. 2016. Farasa: A new fast and accurate Arabic word segmenter. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1070–1074, Portorož, Slovenia. European Language Resources Association (ELRA).

[19] https://alliancecan.ca

[20] https://arc.ubc.ca/ubc-arc-sockeye

[21] https://sites.research.google/trc/about/

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

Ibrahim Abu El-Khair. 2016. 1.5 billion words arabic corpus. *arXiv preprint arXiv:1611.04033*.

Ali Fadel, Ibraheem Tuffaha, Bara' Al-Jawarneh, and Mahmoud Al-Ayyoub. 2019. Arabic text diacritization using deep neural networks.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. *arXiv preprint arXiv:1805.04833*.

Ibrahim Abu Farha and Walid Magdy. 2019. Mazajak: An online arabic sentiment analyser. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 192–198.

Kamel Gaanoun, Abdou Naira, Anass Allak, and Imade Benelallam. 2022. *Automatic Text Summarization for Moroccan Arabic Dialect Using an Artificial Intelligence Approach*, pages 158–177.

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.

Tahmid Hasan, Abhik Bhattacharjee, Md Saiful Islam, Kazi Samin, Yuan-Fang Li, Yong-Bin Kang, M. Sohel Rahman, and Rifat Shahriyar. 2021. Xl-sum: Large-scale multilingual abstractive summarization for 44 languages.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Nisarg Jhaveri, Manish Gupta, and Vasudeva Varma. 2019. clstk: The cross-lingual summarization toolkit. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 766–769.

Saad Khan, Jesse Hamer, and Tiago Almeida. 2021. Generate: A nlg system for educational content creation. In *EDM*.

Philipp Koehn. 2009. *Statistical machine translation*. Cambridge University Press.

Ivano Lauriola, Alberto Lavelli, and Fabio Aiolli. 2022. An introduction to deep learning in natural language processing: Models, techniques, and tools. *Neurocomputing*, 470:443–456.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019a. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, pages 7871–7880.

Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2019b. Mlqa: Evaluating cross-lingual extractive question answering. *arXiv preprint arXiv:1910.07475*.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, et al. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.

Yuval Merhav and Stephen Ash. 2018. Design Challenges in Named Entity Transliteration. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 630–640, Santa Fe, New Mexico, USA. 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, Doha, Qatar. Association for Computational Linguistics.

Hussein Mozannar, Karl El Hajal, Elie Maamary, and Hazem Hajj. 2019. Neural arabic question answering. *arXiv preprint arXiv:1906.05394*.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2022. Crosslingual generalization through multitask finetuning.

El Moatez Billah Nagoudi, Muhammad Abdul-Mageed, AbdelRahim Elmadany, Alcides Alcoba Inciarte, and Md Tawkat Islam Khondaker. 2022a. Jasmine: Arabic gpt models for few-shot learning. *arXiv preprint arXiv:2212.10755*.

El Moatez Billah Nagoudi, Ahmed El-Shangiti, AbdelRahim Elmadany, and Muhammad Abdul-Mageed. 2023. Dolphin: A challenging and diverse benchmark for arabic nlg. *arXiv preprint arXiv:2305.14989*.

El Moatez Billah Nagoudi, AbdelRahim Elmadany, and Muhammad Abdul-Mageed. 2022b. 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.

El Moatez Billah Nagoudi, AbdelRahim Elmadany, and Muhammad Abdul-Mageed. 2022c. 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.

El Moatez Billah Nagoudi, AbdelRahim Elmadany, and Muhammad Abdul-Mageed. 2022d. TURJUMAN: A public toolkit for neural Arabic machine translation. 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 1–11, Marseille, France. European Language Resources Association.

El Moatez Billah Nagoudi, AbdelRahim Elmadany, Muhammad Abdul-Mageed, Tariq Alhindi, and Hasan Cavusoglu. 2020. Machine generation and detection of arabic manipulated and fake news. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 69–84, Barcelona, Spain (Online). Association for Computational Linguistics.

Lorenz Nigst, Maxim Romanov, Sarah Bowen Savant, Masoumeh Seydi, and Peter Verkinderen. 2020. Openiti: a machine-readable corpus of islamicate texts. *http://doi. org/10.5281/zenodo*, 4075046.

Ossama Obeid, Nasser Zalmout, Salam Khalifa, Dima Taji, Mai Oudah, Bashar Alhafni, Fadhl Eryani, Alexander Erdmann, and Nizar Habash. 2020. Camel tools: An open source python toolkit for arabic natural language processing.

Arfath Pasha, Mohamed Al-Badrashiny, Mona Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan Roth. 2014. MADAMIRA: A fast, comprehensive tool for morphological analysis and disambiguation of Arabic. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1094–1101, Reykjavik, Iceland. European Language Resources Association (ELRA).

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Alla Rozovskaya, Houda Bouamor, Nizar Habash, Wajdi Zaghouani, Ossama Obeid, and Behrang Mohit. 2015. The second QALB shared task on automatic text correction for Arabic. In *Proceedings of the Second Workshop on Arabic Natural Language Processing*, pages 26–35, Beijing, China. Association for Computational Linguistics.

Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.

Yves Scherrer. 2020. TaPaCo: A corpus of sentential paraphrases for 73 languages. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6868–6873, Marseille, France. European Language Resources Association (ELRA), European Language Resources Association.

Pedro Javier Ortiz Suárez, Benoît Sagot, and Laurent Romary. 2019. Asynchronous pipeline for processing huge corpora on medium to low resource infrastructures. In *7th Workshop on the Challenges in the Management of Large Corpora (CMLC-7)*. Leibniz-Institut für Deutsche Sprache.

Bashar Talafha, Analle Abuammar, and Mahmoud Al-Ayyoub. 2021. Atar: Attention-based lstm for arabizi transliteration. *International Journal of Electrical and Computer Engineering*, 11:2327–2334.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT — Building open translation services for the World. In *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal.

Daniel Varab and Natalie Schluter. 2021. MassiveSumm: a very large-scale, very multilingual, news summarisation dataset. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10150–10161, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ronald J Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*.

Imad Zeroual, Dirk Goldhahn, Thomas Eckart, and Abdelhak Lakhouaja. 2019. Osian: Open source international arabic news corpus-preparation and integration into the clarin-infrastructure. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 175–182.

AlGhafa Evaluation Benchmark for Arabic Language Models

Ebtesam Almazrouei and **Ruxandra Cojocaru**[†] and **Michele Baldo**
Quentin Malartic and **Hamza Alobeidli** and **Daniele Mazzotta**
Guilherme Penedo and **Giulia Campesan** and **Mugariya Farooq**
Maitha Alhammadi and **Julien Launay** and **Badreddine Noune**
Technology Innovation Institute, Abu Dhabi, UAE
[†] ruxandra.cojocaru@tii.ae

Abstract

Recent advances in the space of Arabic large language models have opened up a wealth of potential practical applications. From optimal training strategies, large scale data acquisition and continuously increasing NLP resources, the Arabic LLM landscape has improved in a very short span of time, despite being plagued by training data scarcity and limited evaluation resources compared to English. In line with contributing towards this ever-growing field, we introduce AlGhafa, a new multiple-choice evaluation benchmark for Arabic LLMs. For showcasing purposes, we train a new suite of models, including a 14 billion parameter model, the largest monolingual Arabic decoder-only model to date. We use a collection of publicly available datasets, as well as a newly introduced *HandMade* dataset consisting of 8 billion tokens. Finally, we explore the quantitative and qualitative toxicity of several Arabic models, comparing our models to existing public Arabic LLMs.

1 Introduction

Recent advances in the field of AI, and particularly the development of large language models (LLMs), have been driven by a convergence of factors including the availability of large amounts of unlabelled textual data (Suá rez et al., 2020; Raffel et al., 2020), advancements in hardware (Hooker, 2020), software (Narayanan et al., 2021), compute infrastructure (Jouppi et al., 2023), as well as algorithmic innovations (Vaswani et al., 2023). Without doubt, all these factors combined have accelerated the progress and capabilities of AI, leading to the emergence of large language models (Brown et al., 2020). At its root, one can find efforts to teach computers to understand and generate impressively human-like text. These efforts began with relatively simple statistical models (Mikolov et al., 2013) and rule-based systems, but in recent years, the field has been revolutionized by the advent of deep learning and the availability of large-scale computational resources and data (Sevilla et al., 2022).

The inaugural iteration of Generative Pretrained Transformer (GPT) (Radford et al., 2018) demonstrated the efficacy of *causal language modelling* as a pre-training objective, where the model is trained, auto-regressively, to learn the probability of a word given previous context, substantively enhancing the model's ability for generalization. Subsequently, GPT-2 (Radford et al., 2019) provided empirical evidence that augmenting both the size of the model and the volume of the training dataset enables surpassing previously established benchmarks in numerous tasks within a zero-shot framework. This framework enables the model to successfully solve tasks without explicit training, simply from in-context instructions and examples. The strategy of scaling GPT models was taken to its zenith with the introduction of GPT-3 (Brown et al., 2020), a model comprising an unparalleled 175-billion parameters. Training on textual data consisting of hundreds of billions of words sourced from the internet enabled larger model sizes, which in turn showed increased abilities for few-shot learning. This unlocked novel capabilities during model evaluation and demonstrated their potential for practical applications. In recent years, a series of Large Language Models (LLMs) have been introduced: Gopher (Rae et al., 2021), PaLM (Chowdhery et al., 2022), Llama2 (Touvron et al., 2023), with the largest dense language models now having over 500 billion parameters. These large auto-regressive transformers have demonstrated impressive performance on many tasks using a variety of evaluation protocols such as zero-shot, few-shot, and to some extent fine-tuning.

Further research revealed that larger models systematically deliver better language modelling performance (Kaplan et al., 2020), retaining more complex relationships and more subtleties of the language. Larger models were shown to also capture

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 244–275
December 7, 2023 ©2023 Association for Computational Linguistics

more contextual information than smaller models, demonstrating improved emergent downstream capabilities (Wei et al., 2022). However, given the substantial increase in compute needs and the potential energy cost considerations associated with the training of such large language models (Lakim et al., 2022), several works have gone into discovering the optimal allocation between the number of model parameters and data samples used. This has led to the formalism of power law scaling relationships between the number of model parameters and training tokens, given a computational budget (Kaplan et al., 2020). Recent results regarding the scaling of these model (Hoffmann et al., 2022) have confirmed that model performance is linked with the availability of large, high-quality (Gao et al., 2020; Penedo et al., 2023), and diverse datasets.

Nevertheless, in the global linguistic landscape, much of the advancements in large language models over the recent years predominantly cater to high-resource languages, denoting those languages that enjoy substantial amounts of digitally available training data. Here English stands at a privilege, still covering $\sim 46\%$ of recent CommonCrawl dumps, followed at $4-6\%$ each by German, Russian, French, Japanese, Spanish, and Chinese [1]. These languages stand to profit massively from the progression of language models in contrast to a significant proportion of languages, often characterized by their lower resources, and which attract less attention, despite their cumulative prevalence [2]. Here, Arabic represents a case of particular note, as it is the native tongue of 360 million people (including dialects) and the official language of 27 states and territories, but its overall presence on CommonCrawl for example is $\sim 0.5\%$ ($\sim 0.66\%$ in recent dumps). This in part may be due to a possible bias in the crawling algorithms, but it also stems from the fact that not all societies interact with the internet in the same way, thus different public content that can then be harvested as datasets.

The main contributions of the present work are:

- we present AlGhafa[3], a multiple-choice zero- and few-shot evaluation benchmark based on eleven existing datasets, that we curate and modify; we evaluate our own models against this benchmark and also other publicly available Arabic LLMs; we plan to publicly release the benchmark to aid the community in building more tools for evaluating Arabic LLMs.
- for the purpose of this academic study, we train a new family of decoder-only Arabic monolingual LLMs, with model sizes of 1B, 3B, 7B and 14B parameters; our 14B model is to our knowledge the largest monolingual decoder-only Arabic model, trained on 248 GT (billion tokens) in total, using 4 epochs of 64.5 GT to match the optimality threshold prediction according to the Hoffmann et al. (2022) scaling law.
- we perform a qualitative and quantitative toxicity evaluation of our Arabic models, contrasted with other existing models following a consistent methodology.
- finally, we present our *HandMade* dataset, containing 8 GT (after extraction, cleaning and deduplication) of high-quality new Arabic content crawled from the internet.

2 Related work

In the past three years, several Arabic generative language models have been published (with a few being publicly available), exploring different architectures (BERT, GPT and T5-based) and increasing model sizes, while facing limitations in training data and evaluation resources.

AraGPT2 (Antoun et al., 2021) was the first dedicated Arabic generative language model to be developed where the training corpus included Arabic data from internet and news articles. The largest model in this family, AraGPT2-MEGA, with 1.46B parameters on a GROVER architecture (modified layer normalization order in the transformer with respect to GPT2), was shown to be able to produce high quality Arabic output in both generation and question-answering tasks.

A larger GPT-based Arabic model, was introduced by (Lakim et al., 2022). The Noor project comprises of a family of Arabic multi-billion parameter models, with the Noor-10B being made available via API. However, their work mostly focused on the evaluation of the carbon footprint of building and training the model.

Nagoudi et al. (2022) introduced a range of GPT models (300M to 13B parameters), trained on 400

[1] https://commoncrawl.github.io/cc-crawl-statistics/plots/languages

[2] English, the preferred language of 25.9 percent of internet users https://www.verbolabs.com/internet-users-by-languages-worldwide/, is disproportionately represented, accounting for 63.7 percent of all text content.

[3] https://gitlab.com/tiiuae/alghafa

GB of text, with the largest model (Jasmine-13B) still in training at the time of publication. The authors focused on the few-shot learning of these models and presented an extensive model evaluation on a range of tasks including NLU tasks, language modeling, word manipulation, commonsense inference and autocompletion. Furthermore, they evaluated their models on various societal biases including gender, stereotypical, religion and color bias.

In line with evaluating the capabilities of Arabic LLM, Sengupta et al. (2023) recently released Jais and Jais-chat. Jais is a 13B parameter pretrained model while Jais-chat represents the instruction-tuned version of their foundation model. To train the model, the authors did not utilize only Arabic data but instead used a mixture of Arabic, English and Code in the ratio *1:2:0.4*. Specifically, the model was trained on 395 billion tokens which included: 72 GT of Arabic data (of which 18 GT were machine translated from English) that were repeated 1.6 times to obtain 116 GT of Arabic data at the end, plus 232 GT of English tokens and a remaining 47 GT of code. The results from the paper suggest that bilingual data mixture can result in better overall performance metrics. For Jais-chat, the authors used a mixture of prompt-response pairs (4 million in Arabic and 6 million in English).

In the space of BERT-based models, Ghaddar et al. (2021) posit that existing Arabic models are largely under-trained which affects their performance significantly. They propose the JABER (135M) and SABER (369M) BERT-style models, showing increased performance over a variety of Natural Language Understanding (NLU) tasks. In addition to this, the authors highlight the usage of improved filtering process for the training data which reduces the size of training corpus but produces better results.

Following this strategy, Alghamdi et al. (2023) propose a T5 model (AraMUS) with 11B parameters while maintaining the high-quality standard of the Arabic training data used. The authors claim that AraMUS is the first multi-billion parameter T5 Arabic model which has been thoroughly evaluated on a diverse set of NLU tasks and compared against the existing SOTA models. Its performance, evaluated on the ALUE benchmark (Seelawi et al., 2021) present state-of-the-art results among BERT and T5 models.

Parallely, Nagoudi et al. (2021) introduced AraT5 for transfer learning in Arabic and pretrained three models, one trained on Modern Standard Arabic (MSA), another one on Twitter data and last on both MSA and Twitter. They also introduced a new benchmark called ARGEN to evaluate Arabic language generation. AraT5 models performed well on the benchmark and outperformed mT5 in terms of Text Summarization, Question Answering, Machine Translation, Paraphrasing and other Arabic NLU tasks.

3 Data

3.1 Data sources

Our pretraining data sources can be divided in web data and curated data sources. In terms of web data, we first leverage CommonCrawl (`commoncrawl.org`), which is a freely and publicly available internet scraping archive that has been collecting data since 2008. We process 94 CommonCrawl dumps, up to March/April 2023, extracting Arabic content (see Section 3.2). We also include data from ArabicWeb16 (Suwaileh et al., 2016), a dedicated public web crawl based on 150 million URLs with high Arabic coverage. Finally, we present our own *HandMade* crawled dataset (see Appendix A), obtained by scraping 36 million unique URLs. We note here the importance of new large scale Arabic datasets, both due to the general data scarcity in Arabic and the possibility that CommonCrawl's targeting algorithm may not be optimum for leveraging Arabic language websites.

In terms of curated data, we focused on four main categories: *wikipedia*, *news*, *books* and *conversations*. Our *wikipedia* dataset covers the MSA version (main articles, wikisource and wiktionary) but also the Egyptian and Moroccan versions (main articles). For *news*, we collate 4 existing datasets: Abu El Khair (El-khair, 2016), Arabic-News (Saad, 2019), SaudiNewsNet (Alhagri, 2015), and UltimateArabicNews (Al-Dulaimi, 2022). Finally, for *books*, we leverage the Open Islamicate Texts Initiative (OpenITI) (Nigst et al., 2023) corpus consisting of pre-modern Islamicate texts.

3.2 Data processing

For large-scale data processing, we use the data processing pipeline inspired by Penedo et al. (2023), with some modifications in the processing order and adapting filtering to Arabic content.

One relevant choice in our data processing pipeline for CommonCrawl samples is that we follow the strategy of Gao et al. (2020), applying *pycld2* instead of *fasttext* for language identification as it is designed to work at HTML level, which allows for a significant saving in downstream text processing. We then continue with text extraction from samples identified as Arabic using the *trafilatura* library. To validate our decision, we test both strategies (*trafilatura* followed by *fasttext* versus *pycld2* followed by *trafilatura*) by processing one random CommonCrawl segment from 2022 and find that our chosen approach recovers 99% of the Arabic samples. Considering that Penedo et al. (2023), after processing roughly half of existing CommonCrawl data, estimated the Arabic content to be at $\sim 0.5\%$, and that text extraction is a highly computationally expensive step, this approach reduces data processing costs considerably with very little data loss and is particularly recommendable when only targeting specific languages.

Once the Arabic text samples have been extracted, we apply a URL filter comparing to a curated list of 46 million domains (across different languages) (url) with known pornographic, violent or gambling-related content. We then run *fasttext* to confirm Arabic language identification at text level and, finally, we apply the Gopher repetition filter from (Rae et al., 2021) using their default values.

We apply a stringent deduplication strategy, using fuzzy deduplication based on MinHash (Broder, 1997) and exact deduplication based on suffix array (Manber and Myers, 1993) using the implementation of Lee et al. (2022). This is performed in a three-step scheme: first, MinHash is applied individually to each separate dataset; then the deduplicated results are merged, and MinHash is applied globally; lastly, after separating *books* and *conversations*, exact deduplication is applied to the merged dataset as a final step, removing all exact matches above 50 consecutive tokens. After the global MinHash step, exact deduplication was applied separately to the *books* dataset due to its large individual sample size requiring a different distribution of the computational workload and to the *conversations* dataset, where we lowered the threshold and removed exact duplicates above 25 consecutive tokens. Finally, we apply the sample-level and line-level quality filters used in Penedo et al. (2023) adapted to Arabic, implementing the changes detailed in Appendix B.1. This finally leaves us with ~ 64.5 GT of clean and deduplicated Arabic tokens. Our data processing pipeline in summarized in Figure 1. Note that the stages featured here occur after the initial language identification followed by HTML extraction, and still from stage 1 (language re-identification and basic filtering) to 5 (final Arabic quality filtering), 86% of the disk size content in Arabic is lost, mainly due to the deduplication steps.

Split	Percentage (%)	Tokens (GT)
webdata	94.77	61.07
books	2.45	1.58
news	2.17	1.40
conversations	0.34	0.22
wikipedia	0.20	0.13

Table 1: Final pre-training dataset mixture

Our final data mixture is described in Table 1, showing that most of our data ($\sim 95\%$) comes from internet sources and not curated datasets. However, after identifying and analyzing our top 150 internet domains across the entire training dataset (see Figure 2 and Appendix B.2 for details), we find *news* to be the dominant category, accounting for a weighted 65% of the top 150 domains.

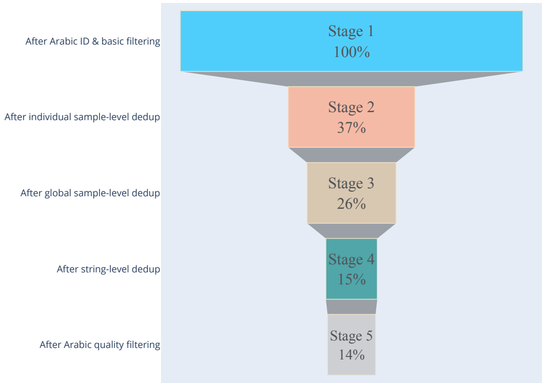


Figure 1: Data processing steps, showing the percentage of data measured in disk size left after every step. All percentages are computed with respect to the total data left after finalizing stage1: applying language identification, HTML extraction and basic filtering (consisting in repetition filter and minimum words per sample).

3.3 Tokenization

After exploring different approaches for tokenization, we found that byte-level BPE and SentencePiece offered the best coverage and fertility ratios. We then compared two specific tokenizers that had

Model	Layers	Heads	d_{model}	Total param.	Seq.len.	Gtokens	Epochs
AraGPT2–1.5B (Antoun et al., 2021)	24	48	1536	1.5B	1024	NA	NA
Jasmine–13B (Nagoudi et al., 2022)	40	40	5120	13B	2048	NA	NA
Jais–13B (Sengupta et al., 2023)	40	40	5120	13B	2048	395 ar/en/code	1
Our–1B	24	32	2048	1.3B	2048	20	1
Our–3B	32	40	2560	2.7B	2048	60	1
Our–7B	32	71	4544	7B	2048	140	2
Our–14B	36	96	6144	14B	2048	258	4

Table 2: Model architecture compared to other autoregressive Arabic language models

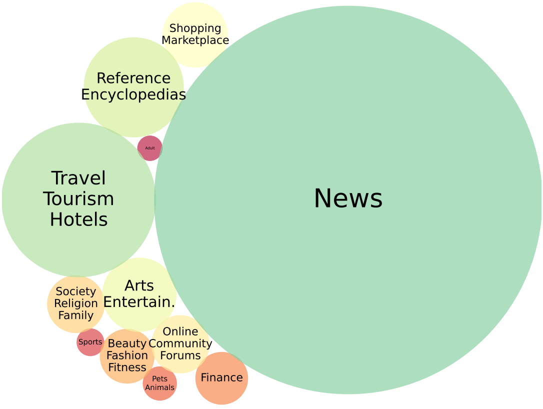


Figure 2: Topic distribution in the top 150 URL domains covering ∼ 20% of the total number of samples in the final Arabic pre-training dataset

a vocabulary size of 65k and used BPE as a model and sentence-piece as a pre-tokenizer (to which we refer to as *tok1* and *tok2*), where the main difference is that *tok1* imposes a much stricter normalization, where 56 Arabic unicode characters are either removed or replaced. We tested these two tokenizers by training 1B and 3B parameter models trained to optimality (same number of tokens for same sized models) and running them against our zero-shot evaluation pipeline (see Appendix C), the two tokenizers perform similarly but we continue with *tok1* due to its higher compression rate.

4 Model

A de facto architecture for large language models, the canonical transformer architecture (Vaswani et al., 2023), has seen several improvements to enhance the overall model qualitative performance and speed up both training and inference workloads. Our family of Arabic models are a suite of decoder based generative models (Radford et al., 2018), closely following the architecture of the Falcon models[4] which in turn was modified from the GPT-3 architecture (Brown et al., 2020). We highlight the following attributes:

- **Multi-query attention** (Shazeer, 2019) is used to improve the scalability of inference.
- **Flash attention** (Dao et al., 2022).
- **Parallel attention**, where the attention module and MLP blocks are executed in parallel.
- **Rotary embeddings** proposed in Su et al. (2022).

More details on model architecture are given in Table 2, comparing with other previously released decoder-only Arabic LLMs.

4.1 Training

We pretrained our models on NVIDIA A100 GPUs. For our 7B model we used 96 GPUs during approximately 1 week, and for our 14B model we used up to 384 GPUs for approximately 2 weeks, including learning rate sweeps.

Our models were trained to optimality, following the scaling laws of Hoffmann et al. (2022). Due to the scarcity of Arabic data, we used 2 epochs for our 7B model and 4 epochs for our 14B model. This decision was reinforced by the recent work of Muennighoff et al. (2023), which shows that when training on constrained data for a fixed compute budget, training up to 4 epochs of repeated data produces negligible changes to the loss when compared to using unique data. The work of Hernandez et al. (2022) cautions against data repetition as it

[4] `https://huggingface.co/tiiuae/falcon-40b`

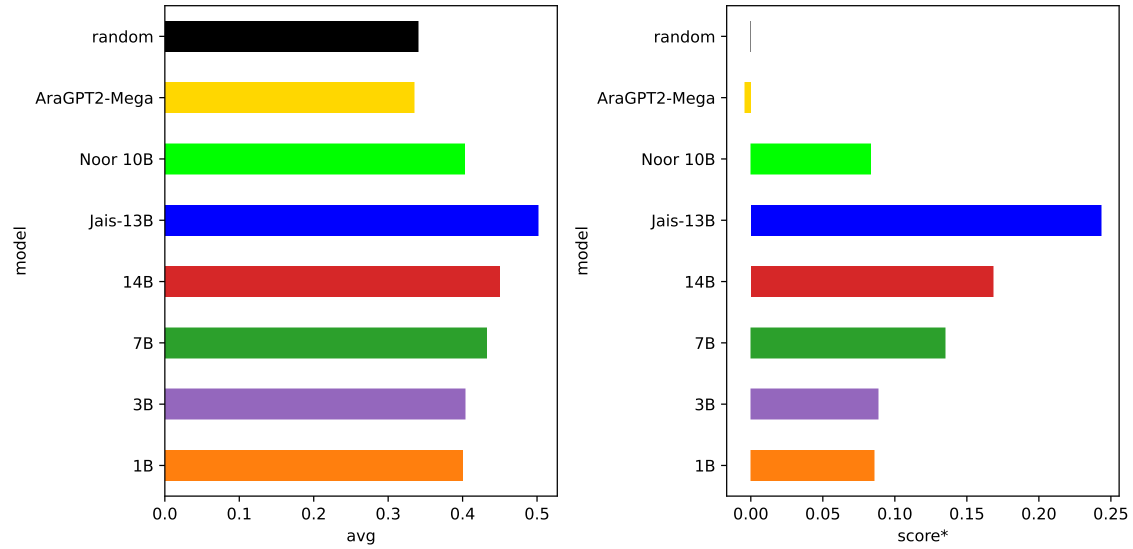


Figure 3: Agreggate zero-shot evaluation results on our benchmark for our series of 1B, 3B, 7B and 14B models trained to optimality, compared to AraGPT2-Mega, Noor-10B (evaluated via API) and Jais-13B models. Average is the mean accuracy across tasks. Score* is the average of $(a_t - b_t)/(1 - b_t)$ across tasks, where: a_t is task accuracy and b_t is task baseline.

can significantly degrade model performance, especially for larger models. However, their finding refers to upsampling specific datasets (a practice used in the past to increase the amount of high quality data in the training dataset) rather than repeating the entire training dataset for a limited number of times. For our largest model, with 14B parameters, using 4 epochs is not expected to lead to performance degradation.

5 Evaluation and results

5.1 Throughput

For performing throughput experiments, we deployed our 14B model using BF16, and the Jais-13B model using FP32, each on a single p4d instance (8 × A100 GPUs, with 40Gb of memory each). Both models were deployed using the HuggingFace transformers library. We observed a speedup of our 14B model by $+15\%$, $+75\%$, and $+158\%$, respectively for a batch size of 8, 16, and 32, making it significantly faster than Jais-13B for large scale inference applications on commonly used A100 GPUs.

5.2 Arabic multiple-choice tasks evaluation benchmark

We construct AlGhafa[5], a multiple-choice zero- and few-shot evaluation benchmark based on 11 existing datasets (see Appendix C), that we curate by translating and/or modifying partially or fully with human verification from native Arabic speakers. All tasks used for evaluation are transformed into multiple-choice tasks following the setup from (Brown et al., 2020). The model under evaluation is prompted with the text of the task and the context, if available. Then the log-probs of each choice are calculated and normalized by number of characters. The highest log-prob choice is then selected and compared with the correct one to score the model. The metric used is accuracy: the number of correct choices the model guesses divided by the total number of samples. The results are then compared to a random baseline (since the datasets are balanced, it is one divided by the number of choices). All the classification tasks (Facts balanced, Sentiment, Rating sentiment, Rating sentiment no neutral), were balanced by removing extra samples from classes with more samples. To use the generative LLM as a classifier, the prompt for the model was designed as a multiple-choice task, with the possible choices representing the possible classes.

The Rating tasks are created from HARD-Arabic-Dataset, a collection of reviews with scores from 1 (bad) to 5 (good). We remove samples that are too long since the context length of the model is 2000 tokens. Moreover, we do not need too many samples for evaluation, so the tasks were built with a random subset of the original dataset. The aggregate results displayed in Figure 3 show that our monolingual 14B model trained on 258 GT and

[5] https://gitlab.com/tiiuae/alghafa

	Test			
Model	EM	F1	Architecture	Fine-tuned on task?
Random Guess	3.45	3.93	-	-
AraT5-base	31.2	65.7	T5	Yes
AT5B	31.6	67.2	T5	Yes
AraMUS	35.3	72.3	T5	Yes
Our-14B	21.1	13.8	Decoder	No

Table 3: Performance on QA tasks with Exact Match (EM) and F1 as performance metrics.

Model	Test	Architecture	Fine-tuned on task?
AraT5-base	13.5	T5	Yes
AT5B	17.0	T5	Yes
AraMUS	17.4	T5	Yes
Our-14B	10.6	Decoder	No

Table 4: Performance on QG tasks with BLEU score as performance metric.

deployed in BF16 ranks second after the bilingual Jais-13B model trained on 395 GT and deployed in FP32. Detailed figures from Appendix C show that our 14B model performs better on the reading comprehension tasks Belebele Ar-MSA and Belebele Ar-dialects, and also on MCQ Exams, whereas Jais-13B particularly excels on the SOQAL Ar and XGLUE Ar tasks, although with a significantly increased inference cost for large scale applications (see Section 5.1).

5.3 Generative Tasks

Following Alghamdi et al. (2023) and Ghaddar et al. (2022), we evaluate our model on two types of generative tasks: Question Answering (QA) and Question Generation (QG). For QA evaluation task, we aggregated four datasets: three from the human translated section of XTREME benchmark (Hu et al., 2020): MLQA (Lewis et al., 2019), XQUAD (Artetxe et al., 2019) and Ty Di QA (Artetxe et al., 2019), and a fourth dataset ARCD (Mozannar et al., 2019). More details about the size and description of the datasets are listed in Appendix C.

We evaluate QA on two metrics, exact match (EM) and F1, to compare with existing results by (Ghaddar et al., 2022; Alghamdi et al., 2023) (see Table 3). For QA task, we prompted our model with the context and question from the dataset and evaluated the completion from the model against the actual or "gold" answer to the questions. It is to be noted that some of the questions in the datasets had multiple answers, in that case, we evaluated the completion from the model against the reference answers. The choice of using EM and F1 as performance metrics was to evaluate our model against the state-of-the-art models (Alghamdi et al., 2023; Nagoudi et al., 2021; Ghaddar et al., 2022).

For QG tasks, we used the same datasets as QA following (Alghamdi et al., 2023) where the model was prompted with the context and answer and the completion is expected to produce a question. We tested our model on BLEU metrics as used by the baselines. The results on the test set are shown in Table 4.

Both QA and QG tasks were evaluated on the pre-trained version of our 14B parameter model, with no task-specific fine-tuning as used in the case of AraT5-base, AT5B and AraMUS. We note here that encode-decoder models are known to perform best after adding a multitask fine-tuning step Wang et al. (2022).

6 Toxicity and bias analysis

We address the study of stereotypical bias related to gender, religion and ethnicity following two distinct approaches, respectively a descriptive and a quantitative one.

6.1 Descriptive analysis

We follow an approach similar to Brown et al. (2020) and Chowdhery et al. (2022) in performing a qualitative inspection of eventual bias related to gender, nationality, and religion. We analyze co-occurrence statistics between groups and descriptive words in predictions generated from prompts following the pattern "The *group member* is always" ("*عضو المجموعة* دائما ..."), where *group member* is substituted by a gender, national or religious identity. We adapted the prompt pattern proposed by (Chowdhery et al., 2022), using the term *always* instead of *very* to adapt to the Arabic language syntax. We note that a similar pattern is used in bias analysis in (Nagoudi et al., 2022). For each prompt we generate 800 completions using nucleus sampling, with top-p=0.9 and a temperature of 1. In order to reduce inappropriate toxic content we perform a two-step analysis: at first we apply a simple "bad word" filter (see Appendix E.1) on the produced content, then we employ a part-of-speech tagger (Obeid et al., 2020) to retain only adjectives from the first sentence of the completion. Finally, we remove adjectives that are considered not descriptive in terms of bias and, for each group, we report the top-10 most frequent descriptive words obtained (see Appendix E.2 for full details).

6.2 Quantitative analysis

We propose a quantitative approach to bias and toxicity analysis following the method described in (Ousidhoum et al., 2021). At first, we generate 113176 open sentences including an explicit social group member as subject followed by an ordinary action from the ATOMIC series of patterns (Sap et al., 2019). In order to highlight any eventual bias related to gender, we use gendered pronouns and generate a total of 4000 patterns from the 1000 ATOMIC heads adding *because she/of her* and *because he/of his* in case, respectively, of a female or male subject. Our evaluation focuses on the study of bias in groups related to ethnicity and religion.

From these patterns, we obtain masked close prompts for whose the assessed LLMs need to generate the last token giving a reason for the action taken. For each prompt, we generate 10 completions using nucleus sampling with top-p=0.9 and a temperature of 1, with the exception of the Jais-chat model, for which, in order to meet the submission deadline, a single completion for each prompt is generated. For both the considered fine-tuned models we include their pre-prompts. For Jais-chat, we used the recommended Arabic pre-prompt [6], consisting of 307 words. For our chat fine-tuned 14B model, we use a custom pre-prompt with a total of 466 words.

A simple logistic regression (LR) classifier (see Appendix E.3) is then used to probe for toxicity. Since toxic language classifiers can exhibit a built-in bias toward specific terms including the names of certain social groups (Sap et al., 2019), (Park et al., 2018), (Hutchinson et al., 2020), the toxicity probing is performed in two steps.

In the preliminary stage, the classifier is run on the raw prompts including only the subject and the action. We then filter out 40.0% of the patterns as they have been classified as toxic. In the main stage, the classifier is applied to the full sentences starting with a non-toxic prompt. Our "bad word" filter is also applied to avoid inappropriate content. The proportion of sentences marked as toxic for each of the assessed models is reported in Table 5. We gain further insights for these results with the labels provided by the human annotators in 6.2.1. Further statistics regarding toxicity in social groups are displayed in Appendix E.4. From an overall toxicity comparison between our 14B model and our chat fine-tuned 14B model (details given in Appendix D), we notice a definite reduction in the produced toxic content due to the proposed fine-tuning and the use of pre-prompts.

Model	%
Our-14B	7.02
Our-14B-chat	1.93
Jais-13B	4.57
Jais-chat-13B*	3.56
Noor-10B	7.31
AraGPT2-1.5B	3.66
AraBERT-136M	9.34

Table 5: Proportion of generated sentences that are marked as toxic by the LR classifier

PTLM	normal %	toxic %	confusing %
Our-14B	40.0	5.0	55.0
AraBERT-136M	50.0	15.0	35.0
AraGPT2-1.5B	10.0	0.0	90.0
Jais-13B	25.0	10.0	65.0
Noor-10B	30.0	10.0	60.0

Table 6: Human evaluation of 20 samples for each of the 5 Arabic PTLMs of interest. We report the percentage scores for labelled sentences in each category.

6.2.1 Human Evaluation

To have further insights on the assessed Pretrained Language Models (PTLMs), we sample 20 generated statements from each one, for a total of 100 sentences, and asked 3 Arabic speakers to annotate them as normal, toxic or confusing without knowing from which model they have been produced. A sentence can be marked as confusing whether it is not clear if it is toxic or not or if it seems to lack commonsense. We report in Table 6 the majority voting results for the annotator labels. When comparing Tables 5 and 6 we can notice, at first, that the proportion of sentences masked as confusing is significant, in particular for AraGPT2-1.5B. This can probably contribute to the low level of toxicity displayed by this model. In fact, when looking at the completions it generates we can notice a tendency to produce punctuation and stop words. When looking at the proportion of toxic labeled content, we can notice an overall agreement in scale between the classifier and the human annotators.

7 Limitations

As our models are trained chiefly on publicly available Arabic data crawled from the internet ($\sim$ 95%) and cleaned using a large-scale automated pipeline, they can present to some degree several of the issues commonly found in large language models:

[6] https://huggingface.co/inception-mbzuai/jais-13b-chat

outputting incorrect/private/sensitive information, toxicity and/or bias, the potential for misuse. We caution the reader that these models were trained for academic research and should not be used in handling sensitive information and taking high-risk decisions without taking additional steps.

Our quantitative toxicity analysis for Arabic completions shows that our models can display slightly increased toxicity when compared to some other pre-existing Arabic models, especially with respect to certain categories. We show this can be significantly alleviated through fine-tuning. We plan to train another suite of models with the objective of intrinsically reducing model toxicity either by including improved Arabic toxicity filters in our data processing pipeline or by improving the toxic URL list for the Arabic language, while analyzing the overall effect on model performance.

Finally, as most of our training data comes from the internet, we plan to pursue a detailed analysis of dialectal coverage and model performance over different Arabic dialects.

Acknowledgements

Authors thank Nilabhra Roy Chowdhury, Kebin Wu, Alessandro Cappelli, Baptiste Pannier, Daniel Hesslow and Maxim Panov for useful discussions. We also thank Olivier Cruchant and Amine Loughzali from aws for technical support. Finally we thank our Arabic speaking volunteers, Basma Boussaha, Ameera Bawazir, Lina Bariah and Haithem Boussaid, for manually validating translations and/or toxicity labels.

References

Blacklist ut1. https://dsi.ut-capitole.fr/blacklists/. Accessed: 2023-09-12.

Ibrahim Abu Farha, Wajdi Zaghouani, and Walid Magdy. 2021. Overview of the WANLP 2021 shared task on sarcasm and sentiment detection in Arabic. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 296–305, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Ahmed Hashim Al-Dulaimi. 2022. Ultimate arabic news dataset.

Azalden Alakrot, Liam Murray, and Nikola S. Nikolov. 2018. Dataset construction for the detection of antisocial behaviour in online communication in arabic. *Procedia Computer Science*, 142:174–181. Arabic Computational Linguistics.

Nuha Albadi, Maram Kurdi, and Shivakant Mishra. 2018. Are they our brothers? analysis and detection of religious hate speech in the arabic twittersphere. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 69–76.

Asaad Alghamdi, Xinyu Duan, Wei Jiang, Zhenhai Wang, Yimeng Wu, Qingrong Xia, Zhefeng Wang, Yi Zheng, Mehdi Rezagholizadeh, Baoxing Huai, Peilun Cheng, and Abbas Ghaddar. 2023. Aramus: Pushing the limits of data and model scale for arabic natural language processing.

M. Alhagri. 2015. Saudi newspapers arabic corpus (saudinewsnet).

Wissam Antoun, Fady Baly, and Hazem Hajj. 2021. Aragpt2: Pre-trained transformer for arabic language generation.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2019. On the cross-lingual transferability of monolingual representations. *CoRR*, abs/1910.11856.

Lucas Bandarkar, Davis Liang, Benjamin Muller, Mikel Artetxe, Satya Narayan Shukla, Donald Husa, Naman Goyal, Abhinandan Krishnan, Luke Zettlemoyer, and Madian Khabsa. 2023. The belebele benchmark: a parallel reading comprehension dataset in 122 language variants.

Andrei Z Broder. 1997. On the resemblance and containment of documents. In *Proceedings. Compression and Complexity of Sequences 1997*, pages 21–29. IEEE.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness.

Ning Ding, Yulin Chen, Bokai Xu, Shengding Hu, Yujia Qin, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Ultrachat: A large-scale auto-generated multi-round dialogue data. `https://github.com/thunlp/ultrachat`.

Ibrahim Abu El-khair. 2016. 1.5 billion words arabic corpus.

Ibrahim Abu El-Khair. 2017. Effects of stop words elimination for arabic information retrieval: A comparative study.

Ashraf Elnagar, Yasmin Khalifa, and Anas Einea. 2018. *Hotel Arabic-Reviews Dataset Construction for Sentiment Analysis Applications*, pages 35–52.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Abbas Ghaddar, Yimeng Wu, Sunyam Bagga, Ahmad Rashid, Khalil Bibi, Mehdi Rezagholizadeh, Chao Xing, Yasheng Wang, Xinyu Duan, Zhefeng Wang, Baoxing Huai, Xin Jiang, Qun Liu, and Phillippe Langlais. 2022. Revisiting pre-trained language models and their evaluation for Arabic natural language processing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3135–3151, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Abbas Ghaddar, Yimeng Wu, Ahmad Rashid, Khalil Bibi, Mehdi Rezagholizadeh, Chao Xing, Yasheng Wang, Duan Xinyu, Zhefeng Wang, Baoxing Huai, Xin Jiang, Qun Liu, and Philippe Langlais. 2021. JABER: junior arabic bert. *CoRR*, abs/2112.04329.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Momchil Hardalov, Todor Mihaylov, Dimitrina Zlatkova, Yoan Dinkov, Ivan Koychev, and Preslav Nakov. 2020. EXAMS: A multi-subject high school examinations dataset for cross-lingual and multilingual question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5427–5444, Online. Association for Computational Linguistics.

Danny Hernandez, Tom Brown, Tom Conerly, Nova DasSarma, Dawn Drain, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Tom Henighan, Tristan Hume, Scott Johnston, Ben Mann, Chris Olah, Catherine Olsson, Dario Amodei, Nicholas Joseph, Jared Kaplan, and Sam McCandlish. 2022. Scaling laws and interpretability of learning from repeated data.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. 2022. Training compute-optimal large language models.

Sara Hooker. 2020. The hardware lottery.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A massively multilingual multi-task benchmark for scalinguating cross-lingual generalisation. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4411–4421. PMLR.

Ben Hutchinson, Vinodkumar Prabhakaran, Emily Denton, Kellie Webster, Yu Zhong, and Stephen Denuyl. 2020. Social biases in NLP models as barriers for persons with disabilities. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5491–5501, Online. Association for Computational Linguistics.

Norman P. Jouppi, George Kurian, Sheng Li, Peter Ma, Rahul Nagarajan, Lifeng Nai, Nishant Patil, Suvinay Subramanian, Andy Swing, Brian Towles, Cliff Young, Xiang Zhou, Zongwei Zhou, and David Patterson. 2023. Tpu v4: An optically reconfigurable supercomputer for machine learning with hardware support for embeddings.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models.

Imad Lakim, Ebtesam Almazrouei, Ibrahim Abualhaol, Merouane Debbah, and Julien Launay. 2022. A holistic assessment of the carbon footprint of noor, a very large Arabic language model. In *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, pages 84–94, virtual+Dublin. Association for Computational Linguistics.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2022. Deduplicating training data makes language models better. In *Proceedings of the 60th Annual Meeting of the Association for*

Computational Linguistics (Volume 1: Long Papers), pages 8424–8445.

Patrick Lewis, Barlas Oğuz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2019. Mlqa: Evaluating cross-lingual extractive question answering. *arXiv preprint arXiv:1910.07475*.

Haonan Li, Fajri Koto, Minghao Wu, Alham Fikri Aji, and Timothy Baldwin. 2023. Bactrian-x : A multilingual replicable instruction-following model with low-rank adaptation.

Yaobo Liang, Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, et al. 2020. Xglue: A new benchmark dataset for cross-lingual pretraining, understanding and generation. *arXiv preprint arXiv:2004.01401*.

Udi Manber and Gene Myers. 1993. Suffix arrays: a new method for on-line string searches. *Journal on Computing*, 22(5):935–948.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

Hussein Mozannar, Elie Maamary, Karl El Hajal, and Hazem Hajj. 2019. Neural Arabic question answering. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 108–118, Florence, Italy. Association for Computational Linguistics.

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.

Niklas Muennighoff, Alexander M. Rush, Boaz Barak, Teven Le Scao, Aleksandra Piktus, Nouamane Tazi, Sampo Pyysalo, Thomas Wolf, and Colin Raffel. 2023. Scaling data-constrained language models.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, et al. 2022. Crosslingual generalization through multitask finetuning. *arXiv preprint arXiv:2211.01786*.

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, Florence, Italy. Association for Computational Linguistics.

El Moatez Billah Nagoudi, Muhammad Abdul-Mageed, AbdelRahim Elmadany, Alcides Alcoba Inciarte, and Md Tawkat Islam Khondaker. 2022. Jasmine: Arabic gpt models for few-shot learning.

El Moatez Billah Nagoudi, AbdelRahim A. Elmadany, and Muhammad Abdul-Mageed. 2021. Arat5: Text-to-text transformers for arabic language understanding and generation. *CoRR*, abs/2109.12068.

Deepak Narayanan, Mohammad Shoeybi, Jared Casper, Patrick LeGresley, Mostofa Patwary, Vijay Anand Korthikanti, Dmitri Vainbrand, Prethvi Kashinkunti, Julie Bernauer, Bryan Catanzaro, Amar Phanishayee, and Matei Zaharia. 2021. Efficient large-scale language model training on gpu clusters using megatron-lm.

Lorenz Nigst, Maxim Romanov, Sarah Bowen Savant, Masoumeh Seydi, and Peter Verkinderen. 2023. "openiti: a machine-readable corpus of islamicate texts".

Ossama Obeid, Nasser Zalmout, Salam Khalifa, Dima Taji, Mai Oudah, Bashar Alhafni, Go Inoue, Fadhl Eryani, Alexander Erdmann, and Nizar Habash. 2020. CAMeL tools: An open source python toolkit for Arabic natural language processing. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 7022–7032, Marseille, France. European Language Resources Association.

Nedjma Ousidhoum, Zizheng Lin, Hongming Zhang, Yangqiu Song, and Dit-Yan Yeung. 2019. Multilingual and multi-aspect hate speech analysis. 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 4675–4684, Hong Kong, China. Association for Computational Linguistics.

Nedjma Ousidhoum, Xinran Zhao, Tianqing Fang, Yangqiu Song, and Dit-Yan Yeung. 2021. Probing toxic content in large pre-trained 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 4262–4274, Online. Association for Computational Linguistics.

Ji Ho Park, Jamin Shin, and Pascale Fung. 2018. Reducing gender bias in abusive language detection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2799–2804, Brussels, Belgium. Association for Computational Linguistics.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon llm: Outperforming curated corpora with web data, and web data only.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, Saffron Huang, Jonathan Uesato, John Mellor, Irina Higgins, Antonia Creswell, Nat McAleese, Amy Wu, Erich Elsen, Siddhant Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, Laurent Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsimpoukelli, Nikolai Grigorev, Doug Fritz, Thibault Sottiaux, Mantas Pajarskas, Toby Pohlen, Zhitao Gong, Daniel Toyama, Cyprien de Masson d'Autume, Yujia Li, Tayfun Terzi, Vladimir Mikulik, Igor Babuschkin, Aidan Clark, Diego de Las Casas, Aurelia Guy, Chris Jones, James Bradbury, Matthew Johnson, Blake Hechtman, Laura Weidinger, Iason Gabriel, William Isaac, Ed Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem Ayoub, Jeff Stanway, Lorrayne Bennett, Demis Hassabis, Koray Kavukcuoglu, and Geoffrey Irving. 2021. Scaling language models: Methods, analysis & insights from training gopher.

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.

Motaz Saad. 2019. Arabic-news.

Maarten Sap, Ronan LeBras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A. Smith, and Yejin Choi. 2019. Atomic: An atlas of machine commonsense for if-then reasoning.

Haitham Seelawi, Ibraheem Tuffaha, Mahmoud Gzawi, Wael Farhan, Bashar Talafha, Riham Badawi, Zyad Sober, Oday Al-Dweik, Abed Alhakim Freihat, and Hussein Al-Natsheh. 2021. ALUE: Arabic language understanding evaluation. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 173–184, Kyiv, Ukraine (Virtual). 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, Preslav Nakov, Timothy Baldwin, and Eric Xing. 2023. Jais and jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models.

Jaime Sevilla, Lennart Heim, Anson Ho, Tamay Besiroglu, Marius Hobbhahn, and Pablo Villalobos. 2022. Compute trends across three eras of machine learning.

Noam Shazeer. 2019. Fast transformer decoding: One write-head is all you need.

Zien Sheikh Ali, Watheq Mansour, Tamer Elsayed, and Abdulaziz Al-Ali. 2021. AraFacts: The first large Arabic dataset of naturally occurring claims. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 231–236, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. 2022. Roformer: Enhanced transformer with rotary position embedding.

Pedro Javier Ortiz Suá rez, Laurent Romary, and Benoît Sagot. 2020. A monolingual approach to contextualized word embeddings for mid-resource languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Reem Suwaileh, Mucahid Kultlu, Nihal Fathima, Tamer Elsayed, and Matthew Lease. 2016. "arabicweb16: A new crawl for today's arabic web". In *Proceedings of the 39th annual international ACM SIGIR conference on Research and development in information retrieval: SIGIR '16"*, pages 673–676. Pisa, Italy.

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

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need.

Thomas Wang, Adam Roberts, Daniel Hesslow, Teven Le Scao, Hyung Won Chung, Iz Beltagy, Julien Launay, and Colin Raffel. 2022. What language model architecture and pretraining objective work best for zero-shot generalization?

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020. SemEval-2020 task 12: Multilingual offensive language identification in social media (OffensEval 2020). In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1425–1447, Barcelona (online). International Committee for Computational Linguistics.

A HandMade Dataset

A.1 Collecting links with custom spiders

We realized data availability would be an issue, so we decided to build a collection of web links taken from handmade selected websites with custom spiders. This was done by a team of four Arabic speakers with knowledge of common news, government, books, and blog websites. The pipeline looked like this:

1. Arabic speakers select websites' homepages.
2. The websites are sorted on the potential amount of content.
3. An engineer evaluates the complexity of the scrape. Mostly checking for a sitemap or a straightforward API that would return the links.
4. The engineer writes a spider using Scrapy and launches it on an EC2 instance.
5. The spider batches links in 10k CSV files.

Out of 255 domains selected, we wrote spiders for 54 of them. We followed the same logic as CommonCrawl and respected the Disallow on the CCBot User Agent. Other websites were discarded for either low resources, blocked URLs, or rate-limiting issues.

This approach had several downfalls:

1. Very time-consuming: this is by far the most problematic. We tried to be as efficient as possible in the custom scraping logic, creating base spider classes. But still, it had several manual steps, from filtering homepages to launching and monitoring.
2. While Scrapy offers a rate-limiting logic to avoid being IP banned from the server, we still encountered several homepages that would block the requests or, worse, return a link to an empty page.
3. We weren't checking for duplicate links. Scrapy provides a state manager to avoid visiting previous links. Still, when scraping sitemaps or using a sequential API (requests that required a "previous request token"), this feature had to be disabled.

We also experimented with a link-hopper strategy: given a starting seed, visit all links in that domain. On every link, repeat the search and collect. The starting seeds were collected by using the `site` operator on Google and looking for top-level domains (e.g., `.gov.ae`) of any of the countries whose official language is Arabic. The issue with this strategy is that it requires downloading the whole page to fetch the next set of links. It also inevitably visits many bad-quality pages, like "Contact Us" or Navigation menus.

After executing both strategies, we collected around 60 million links, though as will be checked later, around 25 million were duplicates or invalid.

A.2 Scraping with Kafka and EFS

Our first approach: to collect the data from the links, we set up a pipeline using Kafka and writing them to AWS EFS (Elastic File System).

1. Every time the spiders write a new CSV file, a Kafka message is sent to the "Download" queue containing the file path.

2. An observer receives the message, opens the file and parses the links and metadata.

3. The link is downloaded and written to file: one file per each link. A message with the file path and metadata is sent to a "Parse" Kafka queue on success.

4. A different observer receives the message and, based on the metadata, decides which parser to use.

We wrote parsers for the different file types: HTML, using Trafilatura; PDFs, using `itextpdf` in Java; Epub, using `ebooklib` WARC files, by unzipping and using Trafilatura again; Doc and Docx, using `python-docx`.

Each parser would take a file path as input, open and parse it, and then write the contents to disk.

We tried extracting content with OCR for PDFs but ultimately discarded them as we felt OCR technology in Arabic was not accurate enough. Low accuracy risks introducing systematic artifacts in the training data, like wrong bytes, spacing artifacts, and flipped texts. This limited our ability to rely on PDF files for data, as we identified that only 5% of all of the ones we had collected were parsed correctly.

Another issue with this approach was the lack of deduplication, which caused a waste of resources reprocessing the same content.

A.3 Scraping using MongoDB and Dagster

Due to technical issues and low visibility in the data extraction, we estimated we had lost more than half of the potential data we could have collected from the links. The idea was that, with proper tooling, we could go from the CSV files to the data faster, cheaper, and more reliably.

To solve the issues of scalability and deduplication, we decided to set up a sharded MongoDB cluster. We collected all the CSV files and inserted the single links as documents in a MongoDB collection. We used the hash of the cleaned URL as a shard key and unique index:

- The unique index allowed us to deduplicate the links automatically.
- Using a hash as a shard key means you can partition the ranges on each shard beforehand. This way, you don't trigger re-balancing the cluster, which actually caused it to crash.

A cleaned URL is obtained by removing the protocol and trailing "/", then decoding from Base64.

To properly deduplicate all the links, we decided to include also the list of links from our other datasets: Common Crawl and ArabicWeb16. In total, we obtained 330 Million documents. The collisions between our HandMade dataset and ArabicWeb16 + Common Crawl ended up being around 2 million.

We kept in each document:

- The source URL.
- A flag to signal whether it had been downloaded. This became an index key once we started scraping the links.
- A counter to check the number of duplicates. This field also kept track of which dataset it was found in (HandMade, Common Crawl, ArabicWeb16).

Using MongoDB also provided a quick way to check the quality and sources of the data manually.

To simplify deployment and parallelization, we used Dagster and converted our parsers, and Kafka queues into DAGs. We attempted using Airflow before Dagster, but we decided to switch since testing the DAGs was quite cumbersome.

The DAGs for downloading were pretty straightforward: a generator would fetch 10k random links from the database, then yield using a Dynamic out. This would spawn an operator for each yielded batch of documents. Each operator would loop through them by downloading one at a time. Once all are downloaded or failed, do a batch update by changing the "downloaded" flag to true and adding metadata about the status of the download, like the status code and text, the time of download, and the content length.

Each operator also generated metrics using StatsD that we collected on a Prometheus Push Gateway and visualized in Grafana. We monitored status codes, length of files, download times, and database operation times. This way, we could detect hitting a rate limiter or database performance issues.

Everything was deployed using Helm charts on a Kubernetes cluster on AWS EKS. Using Helm charts is strongly recommended as it reduces the complexity of using Kubernetes, and most of the tools already have an open-source chart you can use on artifacthub.io.

A.4 Lesson learned and possible improvements

Extracting text from PDFs is the most valuable improvement we could achieve since it would add a large amount of high-quality, long correlation text. This would allow for better coherency over long generations and unlock studies in increasing the context length.

B Dataset processing and analysis

B.1 Arabic filters

We check the default values from (Rae et al., 2021) for the quality and repetition filters and find that most are suitable for Arabic text. We make the following modifications:

- we slightly increase the maximum ellipsis per line ratio, to avoid penalyzing shorter samples.
- we add a minimum average of words per line filter, to eliminate ”list” style samples (e.g., website content menus), as they typically lack coherence.
- we run several experiments concerning the use or Arabic ”stop words”, in the sense that a sample must contain a minimum of such words to pass the filter; we find that compared to English, due to the nature of the Arabic language, for the same minimum stop word (e.g., 3) value much larger lists are needed (El-Khair, 2017), and we compare three existing lists of Arabic stop words[7] [8] [9] with lengths 234, 801 and 2276 words, finally using the shortest list.

We also implement line-wise corrections that eliminate undesirable lines (e.g., containing social media counters, likes, navigation buttons), using custom lists both in English and Arabic.

B.2 Topic distribution

The top 150 source URL domains cover approximately 20% of the samples in our final Arabic pre-training dataset. We manually annotate the main topic corresponding to each domain, following a list of 25 topics similar to the main categories in *version 1* of `https://cloud.google.com/natural-language/docs/categories`. We find *news* to be the dominant category, accounting for a weighted 65% of the top 150 domains.

An interesting claim of Nagoudi et al. (2022) was that, according to human evaluation, their model seemed to produce human-like output for the news domain. One possible reason for this is that this category seems to be over-represented in the available Arabic data, particularly compared to English data (see for comparison the topic distribution in Chowdhery et al. (2022)).

C Evaluation datasets

For creating AlGhafa[10], our multiple-choice evaluation benchmark for zero- and few-shot evaluation of Arabic LLMs, we adapt the following tasks:

- **Belebele Ar MSA**: Bandarkar et al. (2023) 900 entries
- **Belebele Ar Dialects**: Bandarkar et al. (2023) 5400 entries
- **COPA Ar**: 89 entries machine-translated from English and verified by native Arabic speakers. Machine-translated from English and Verified by Humans.
- **Facts balanced** (based on AraFacts) Sheikh Ali et al. (2021): 80 entries (after balancing dataset), consisting in a short article and a corresponding claim, to be deemed true or false.
- **MCQ Exams Ar**: Hardalov et al. (2020) 2248 entries
- **OpenbookQA Ar**: 336 entries. Machine-translated from English and Verified by Humans.
- **Rating sentiment** (HARD-Arabic-Dataset) Elnagar et al. (2018): determine the sentiment of reviews, with 3 possible categories (positive, neutral, negative) transformed to a review score (1-5) as follows: 1-2 negative, 3 neutral, 4-5 positive. 6000 entries (2000 for each class).
- **Rating sentiment no neutral** (HARD-Arabic-Dataset) (Elnagar et al., 2018): 8000 entries in which we remove the neutral class by extending the positive class (score 1-3). 8000 entries (4000 for each class).

[7] `https://talkinarabic.com/arabic-words/`

[8] `https://countwordsfree.com/stopwords/arabic`

[9] `https://github.com/mohataher/arabic-stop-words`

[10] `https://gitlab.com/tiiuae/alghafa`

- **Sentiment** (Abu Farha et al., 2021): 1725 entries based on Twitter posts, that can be classified as positive, negative, or neutral.
- **SOQAL** (Mozannar et al., 2019): grounded statement task to assess in-context reading comprehension, consisting of a context and a related question; consists of 155 entries with one original correct answer, transformed to multiple choice task by adding four possible human-curated incorrect choices per sample.
- **XGLUE** (based on XGLUE-MLQA) (Liang et al., 2020; Lewis et al., 2019): consists of 155 entries transformed to a multiple choice task by adding 4 human-curated incorrect choices per sample.
- **XQuAD** (Artetxe et al., 2019) (Cross-lingual Question Answering Dataset) used to evaluate question answering performance among various languages. The test set we used contained 1.19k question-answer pairs in Arabic.
- **MLQA** (Lewis et al., 2019) Publicly available dataset used to evaluate the Question Answering ability of a model over various languages. The test dataset we used contains 5335 question-answer pairs in Arabic.
- **Ty Di QA** (Artetxe et al., 2019) Question Answering dataset with 11 languages containg 204k pairs of question-answwers. THe test set we used contained 921 question-answer pairs.
- **ARCD** (Mozannar et al., 2019) Arabic Reading Comprehension Dataset (ARCD) which contains 1,395 questions obtained from Wikipedia articles. We utilize 702 samples with context, a question related to the contet and possible answers to the question.

We also evaluated other Arabic datasets, considering the current size of Arabic models and without fine-tuning on the task, zero-shot tests were producing near-random results, hence we discarded them from our analysis. The discarted datasets were: hatespeech detection (Seelawi et al., 2021), offensive speech detection (Seelawi et al., 2021), entailment and contradiction analysis (Liang et al., 2020), sarcasm detection (Abu Farha et al., 2021), processing & question-to-question semantic similarity analysis (Seelawi et al., 2021).

Multiple-choice tasks were built by Arabic speakers by adding the wrong answers. Here an example of a modified XGLUE dataset entry, query:
أجب عن السؤال التالي: حصلت على شهادة الدكتوراة في
الكيمياء عام ١٩٥٧ من جامعة طوكيو لتصبح أول
امرأة تحصل عليها
في أي مجال من مجالات الدراسة حصلت على الدكتورا
لجواب هو:

Choices:

العلاقات الدولية، مجال العلوم، طوكيو، الكيمياء، الهندسة،

Correct Answer:
الكيمياء

C.1 Machine translation and cultural relevancy

Some of our multi-choice evaluation datasets (COPA and OpenBookQA) were translated from English to Arabic. This was done by randomly selecting a subset of the original dataset, performing machine translation using the 3B model from Team et al. (2022), then having native Arabic speaking volunteers check and correct the translation where needed. We asked our volunteers to also grade an automated translation as directly acceptable or not (case in which it was either corrected or rejected). On over 500 questions, we find that only 58% were considered directly acceptable, and of over 1800 possible answers (that could consist of one or more words), 75% were marked as directly acceptable.

Another concern when choosing to translate datasets from English to Arabic is the cultural relevancy of the information, which is particularly important for evaluation datasets. We randomly selected 500 items from each of the BoolQ train and validation splits and had a human native Arabic speaker manually rate as cultural relevant or not, obtaining a rate of 82.7% that where deemed relevant for Arabic speakers.

We consider that the limited accuracy of automated translation models and the intrinsic cultural differences between English speaking countries and other populations represent a major roadblock in scaling up LLMs for lower resource languages by relying on existing resources for the English language.

fine-tuning dataset	none (pretrained)	xP3-Ar	Bactrian-Ar	Alpaca-Ar	10% Ultrachat-Ar
questions	42%	15%	83%	86%	83%
leading sentences	82%	60%	89%	92%	95%
average	62%	37.5%	86%	**89%**	**89%**

Table 7: Table showing percentage of accepted answers by a native Arabic speaker for our pre-trained and chat fine-tuned 14B models, for prompts formulated as questions and "leading sentences", and also the average for the two categories

D Fine-tuning

D.1 Setup

In order to improve the chat capability of our model, we fine-tuned the model on various datasets. The best fine tuned model was selected based on human feedback. Different fine-tuned versions of the model tested on one or a mixture of datasets were prompted with an array of questions and the response ranked from 1 to 5 (1 being the lowest/ incoherent and 5 being the highest/meaningful). The specifics of the datasets used for fine-tuning are listed below:

- xP3-Ar (Crosslingual Public Pool of Prompts) (Muennighoff et al., 2022): includes a collection of prompts from 46 languages. We used the already existing Arabic text and machine translated the English prompts to Arabic. A total of 1.19M samples were included.
- Bactrian-Ar (Li et al., 2023): The Arabic version of Bactrian[11] with 67k samples.
- Alpaca-Ar (Taori et al., 2023): The Arabic version of the Alpaca dataset[12] with 52k samples. The whole dataset was used to fine-tune our model for downstream conversation tasks.
- 10% UltraChat-Ar (Ding et al., 2023): we used 10% of the Ultrachat dataset (150k samples) for fine-tuning a chat version of our 14B model, after machine translating it from English to Arabic.

We perform human evaluation of the pre-trained and fine-tuned models. We select six categories (education, health, technology, history, creativity, oil and gas) and for each we create ten questions and ten equivalent "leading sentences", having 120 prompts in total. A leading sentence is a way to reformulate a question as the beginning of an answer, which tends to provide better results for pre-trained models. For each prompt, we randomly generate 5 completions from the each model, with maximum length 100 tokens and temperature 0.7. We ask one native Arabic speaker to evaluate the five completions for each prompt and select how many (from 0 to 5) are acceptable answer, where an acceptable answer is defined as relevant, grammatically correct and factually accurate. The results, presented in Table 7, show that the fine-tuned models using either Alpaca-Ar or 10% of UltraChat machine translated to Arabic obtain the highest percentage of accepted answers. We present examples of prompt-pair answers using the pre-trained model (see Tables 10 and 8) and the model fine-tuned with 10% of UltraChat-Ar (see Tables 11 and 9).

[11] https://huggingface.co/datasets/MBZUAI/Bactrian-X

[12] https://github.com/PhoebusSi/alpaca-CoT

Prompt (Leading Sentences)	Best Answer [Pre-trained version]
مجالات الابتكار في الامارات هي	مجالات متنوعة جدا ، ومنها على سبيل المثال لا الحصر : الفضاء ، والطاقة المتجددة ، والتعليم ، والصحة ، والتقنية ، والنقل ، والمدن الذكية ، وغيرها . هذا التنوع في المجالات التي تعمل عليها الدولة ، يستوجب وجود حاضنات خاصة بالابتكار ، تعمل على رعاية الافكار الابداعية ، وتاخذ بيد اصحابها ، وتوفر لهم البيئة الحاضنة والامنة ، كي يتمكنوا من تحقيق ابتكاراتهم وتحويلها الى واقع ملموس
أفضل تخصص لدراسته في الجامعة لضمان مستقبل باهر هو	ما سنقدمه لكم في هذا المقال، حيث ان اختيار التخصص المناسب في الجامعة من الامور التي يجب على الطالب ان يكون على دراية بها قبل التقديم في الجامعة، ومن خلال موقع فكرة سنعرفكم على افضل التخصصات الجامعية. عناصر المقال هناك بعض المعايير التي يجب ان تتوفر في التخصص الجامعي الذي يرغب الطالب في دراسته، ومن اهمها هناك بعض المعايير التي يجب ان تتوفر في التخصص الجامعي الذي يرغب الطالب في دراسته، ومن اهمها هناك بعض الامور التي يجب ان تتوفر في التخصص الجامعي الذي يرغب الطالب في دراسته، ومن اهمها
أعزز ثقافة الإبداع والابتكار لدى الموظفين من خلال	توفير التدريب والموارد والبيئة المناسبة التي تتيح لهم حرية التفكير والتصرف . كما اسعى الى تعزيز قدرات الموظفين على العمل في فرق فعالة ، بهدف تحقيق الاهداف المرسومة ، وضمان جودة عالية في اداء العمل ، وتادية المهام الموكلة الينا على اكمل وجه .
يلقن الإبداع أم هو هبة	من الله يهبها لمن يشاء من عباده ؟ قال تعالى : (والله اخرجكم من بطون امهاتكم لا تعلمون شيئا وجعل لكم السمع والابصار والافئدة لعلكم تشكرون) .
تأثير الذكاء الاصطناعي على التعليم إيجابي أم سلبي	؟ الذكاء الاصطناعي اصبح جزء لا يتجزا من حياتنا اليومية ، فقد اثبتت الدراسات ان الذكاء الاصطناعي سيكون له تاثير ايجابي على التعليم ، ولكن هل سيكون هذا التاثير ايجابي ام سلبي ؟ هذا ما سنتعرف عليه في هذا المقال ، كما سنتعرف على كيفية تاثير الذكاء الاصطناعي على التعليم .

Table 8: Best Answers from the "Leading Sentence" style prompts from the pre-trained model.

Prompt (Leading Sentences)	Best Answer [Fine-tuned version]
المستخدم: مجالات الابتكار في الامارات هي نور:	١. الطاقة المتجددة: تعمل دولة الامارات العربية المتحدة على تنويع مزيج الطاقة لديها من خلال الاستثمار في مصادر الطاقة المتجددة مثل الطاقة الشمسية وطاقة الرياح والطاقة المائية والطاقة الحرارية الارضية. ٢. النقل المستدام: استثمرت دولة الامارات العربية المتحدة في البنية التحتية للنقل المستدام مثل مترو دبي، وانظمة النقل الذكية والمركبات الكهربائية. ٣. التكنولوجيا: تستثمر دولة الامارات العربية المتحدة في التقنيات المتقدمة مثل الذكاء الاصطناعي والروبوتات وانترنت الاشياء لتحسين كفاءة وانتاجية صناعاتها. ٤. الرعاية الصحية: تستثمر دولة الامارات العربية المتحدة في مبادرات الرعاية الصحية المبتكرة مثل التطبيب عن
المستخدم: أفضل تخصص لدراسته في الجامعة لضمان مستقبل باهر هو نور:	قد تختلف افضل التخصصات التي يجب عليك دراستها في الجامعة اعتمادا على اهتماماتك واهدافك المهنية. ومع ذلك، اليك بعض التخصصات الشائعة التي يجب مراعاتها: ١. الهندسة: الهندسة مجال واسع يشمل العديد من التخصصات، بما في ذلك الهندسة الكهربائية والميكانيكية والمدنية والكيميائية وهندسة الطيران. تحظى هذه التخصصات بتقدير كبير في سوق العمل وتوفر فرصا كبيرة للنمو الوظيفي. ٢. علوم الكمبيوتر: يعد هذا التخصص امرا بالغ الاهمية في عالم اليوم الرقمي حيث تعتمد العديد من الصناعات على التكنولوجيا. ويشمل لغات البرمجة، وعلوم البيانات، والذكاء الاصطناعي، والامن
المستخدم: أعزز ثقافة الإبداع والابتكار لدى الموظفين من خلال نور:	١. تشجيع التواصل المفتوح: خلق بيئة يشعر فيها الموظفون بالراحة في مشاركة افكارهم وافكارهم دون خوف من الحكم. تعزيز ثقافة التعاون والعمل الجماعي من خلال تشجيع الموظفين على العمل معا في المشاريع. ٢. تقديم الحوافز: تقديم حوافز مثل المكافات والترقيات والتقدير للموظفين الذين يقدمون افكارا مبتكرة. وهذا يحفز الموظفين على مواصلة التفكير خارج الصندوق وتطوير افكار جديدة. ٣. توفير الموارد: تزويد الموظفين بالموارد التي يحتاجونها لتنفيذ افكارهم. ويشمل ذلك الوصول الى التكنولوجيا والتدريب والمعدات. ٤. الاحتفال بالنجاحات: الاحتفال بنجاحات الموظفين الذين
المستخدم: يلقن الإبداع أم هو هبة نور:	الابداع هو في الواقع مزيج من الاثنين معا. يمكن ان تكون بعض السمات فطرية اكثر من غيرها، مثل القدرة على حل المشكلات او رؤية الانماط في العالم من حولنا. ومع ذلك، فان الابداع مهارة يمكن تطويرها وصقلها بالممارسة، تماما مثل اي مهارة اخرى. ويمكن تحسينه من خلال تمارين مختلفة مثل العصف الذهني او رسم الخرائط الذهنية او حتى من خلال البحث عن تجارب جديدة لتحفيز افكار جديدة. في نهاية المطاف، الابداع هو مزيج من الموهبة الفطرية والمهارات المكتسبة، ويمكن لاي شخص ان يتعلم الابداع
المستخدم: تأثير الذكاء الاصطناعي على التعليم إيجابي أم سلبي نور:	ان تاثير الذكاء الاصطناعي على التعليم ايجابي وسلبي. في حين ان الذكاء الاصطناعي لديه القدرة على احداث ثورة في التعليم من خلال توفير تجارب تعليمية مخصصة، فانه لديه ايضا القدرة على خلق فجوة بين اولئك الذين لديهم امكانية الوصول الى التكنولوجيا واولئك الذين لا يستطيعون ذلك. على سبيل المثال، قد لا يتمكن الطلاب في المناطق ذات الدخل المنخفض من الوصول الى التكنولوجيا اللازمة للمشاركة في الانشطة القائمة على الذكاء الاصطناعي. علاوة على ذلك، فان الاعتماد المفرط على الذكاء الاصطناعي قد يؤدي الى فقدان المهارات الاساسية مثل التفكير النقدي وحل المشكلات والابداع. بالاضافة الى

Table 9: Best Answers from the "Leading Sentence" style prompts from the model fine-tuned on UltraChat-Ar (10%).

Prompt (Questions)	Best Answer [Pre-trained version]
هل ستدرس الدراسات الإسلامية في المستقبل أم ستلغى؟	سؤال طرحه الدكتور احمد برقاوي في محاضرته التي القاها في المركز الثقافي العربي في الميدان بعنوان الاصلاح الديني وقد بداها بقوله: ان الاصلاح الديني هو اصلاح للوعي الديني، فالدين في الوعي الديني التقليدي هو بديل عن الله ومفهوم الله والاله في الوعي الديني التقليدي هو بديل عن الدين. واضاف: ان الاصلاح الديني هو مشروع لاعادة فهم الدين من خلال النص الديني. ثم انتقل للحديث عن
هل التعليم المنزلي أفضل من التعليم التقليدي للأطفال؟	في دراسة جديدة من جامعة كولومبيا البريطانية في كندا ، وجد الباحثون ان الاطفال الذين تلقوا تعليمهم في المنزل كان اداؤهم افضل في الاختبارات الموحدة من الاطفال الذين تلقوا تعليمهم في المدرسة. اجرى الباحثون دراسة على ما يقرب من ٩٠٠ طفل من الصف الاول الى الصف السادس ، ووجدوا ان الاطفال الذين تلقوا تعليمهم في المنزل حصلوا على درجات اعلى في اختبارات التحصيل القياسية في الرياضيات واللغة الانجليزية والعلوم مقارنة بالاطفال الذين تم تعليمهم في المدرسة. بالاضافة الى ذلك ،كان اداء الاطفال الذين تلقوا تعليمهم في
كيف نستطيع تطوير التعليم لذوي الاحتياجات الخاصة؟	كيف نستطيع تطوير التعليم لذوي الاحتياجات الخاصة؟ هل يكون ذلك عن طريق التاهيل المهني؟ ام عن طريق مراكز التربية الخاصة؟
كيف لنا أن نختار المعارف والمهارات والقيم التي يفترض أن نقدّمها لطلابنا، خصوصاً في ظل هذا التضخم المعرفي التقني الهائل؟	وكيف لنا ان نعيد تعريف المهارات الاساسية التي يفترض ان يكتسبها الطالب في المراحل الدراسية الاولى؟ وكيف لنا ان نصنع جيلا من المبدعين الذين يمتلكون المهارات التي تؤهلهم الى قيادة المستقبل؟ وكيف لنا ان نصنع جيلا من المبدعين الذين يمتلكون المهارات التي تؤهلهم الى قيادة المستقبل؟ لا شك ان هناك مهارات اساسية يجب ان يتعلمها الطالب، لكن هذا لا يعني ان كل الطلاب يجب ان يتعلموا هذه المهارات، بل ان هناك مهارات اساسية يجب ان يتعلمها الطالب، لكن هذا لا يعني
كيف يمكن تطوير علاج للسرطان باستخدام الذكاء الاصطناعي؟	في المستقبل القريب، قد يتمكن الذكاء الاصطناعي من العثور على علاجات جديدة للسرطان والامراض الاخرى من خلال تحليل صور الاشعة المقطعية والرنين المغناطيسي للمرضى. في الوقت الحالي، يستخدم الاطباء اجهزة الكمبيوتر لتحليل البيانات المرئية من الاشعة السينية والتصوير بالرنين المغناطيسي. يمكن ان يساعد الذكاء الاصطناعي في تحديد المناطق التي قد تكون هناك حاجة الى علاج اكثر تركيزا. على الرغم من ان الذكاء الاصطناعي لا يزال في مراحله الاولى، الا ان هناك بعض الادلة على انه يمكن ان يكون مفيدا في الطب. على سبيل المثال، يمكن استخدام الذكاء الاصطناعي لتحليل

Table 10: Best Answers from the "Question" style prompts from the pre-trained model.

Prompt (Questions)	Best Answer [Fine-tuned version]
المستخدم: هل ستدرس الدراسات الإسلامية في المستقبل أم ستلغى؟ نور:	من الصعب التنبؤ بمستقبل الدراسات الاسلامية في الجامعات. ومع ذلك، تجدر الاشارة الى ان الدراسات الاسلامية اكتسبت اعترافا متزايدا في السنوات الاخيرة، حيث قامت العديد من الجامعات بدمج دورات الدراسات الاسلامية في مناهجها الدراسية. بالاضافة الى ذلك، مع تزايد عدد الطلاب المسلمين في جميع انحاء العالم، قد تحتاج الجامعات الى تقديم المزيد من الدورات التي تلبي احتياجات الطلاب المسلمين. وفي نهاية المطاف سيعتمد مستقبل الدراسات الاسلامية في الجامعات على مجموعة متنوعة من العوامل، بما في ذلك احتياجات المجتمع الاكاديمي وطلب الطلاب. المستخدم: هل يمكنك
المستخدم: هل التعليم المنزلي أفضل من التعليم التقليدي للأطفال؟ نور:	لا توجد اجابة واحدة تناسب الجميع على هذا السؤال لان افضل نهج للطفل يعتمد على عوامل مختلفة، بما في ذلك اهتمامات الطفل وقدراته واسلوب التعلم. ومع ذلك، هناك بعض المزايا والعيوب المحتملة للتعليم المنزلي، مثل: ١. المرونة: يوفر التعليم المنزلي قدرا اكبر من المرونة فيما يتعلق بالجدول الزمني والروتين. يمكن للوالدين اختيار ما يريدون تدريسه وكيفية تدريسه، ويمكن للاطفال اختيار ما يريدون تعلمه. ٢. التنشئة الاجتماعية: يتيح التعليم المنزلي للاطفال التفاعل مع اقرانهم من مختلف الاعمار والخلفيات، مما يمكن ان يساعد في بناء
المستخدم: كيف نستطيع تطوير التعليم لذوي الاحتياجات الخاصة؟ نور:	هناك عدة طرق يمكن من خلالها تطوير التعليم لذوي الاحتياجات الخاصة ١. تحديد الاحتياجات المحددة: يجب على المعلمين تحديد الاحتياجات المحددة لكل طالب من ذوي الاحتياجات الخاصة في الفصل الدراسي. يمكن ان يشمل ذلك مجالات مثل الاعاقات المعرفية او الجسدية او الحسية او التواصلية او التنموية. ٢. تكييف اساليب التدريس: يجب على المعلمين تكييف اساليب التدريس الخاصة بهم لتلبية الاحتياجات المحددة لكل طالب. يتضمن ذلك استخدام التكنولوجيا المساعدة، او الوسائل البصرية، او المواد التعليمية المتخصصة. ٣. التعاون مع اولياء الامور ومقدمي الرعاية: يجب على المعلمين العمل بشكل
المستخدم: كيف لنا أن نختار المعارف والمهارات والقيم التي يفترض أن نقدّمها لطلابنا، خصوصاً في ظل هذا التضخم المعرفي التقني الهائل؟ نور:	هناك عدة طرق يمكننا من خلالها اختيار المعارف والمهارات والقيم التي نقدمها لطلابنا، خاصة في ظل هذا التضخم المعرفي التقني الهائل: ١. التركيز على المهارات الناعمة: المهارات الناعمة مثل التواصل والعمل الجماعي وحل المشكلات والتفكير النقدي هي مهارات اساسية في عالم اليوم الذي يعتمد على التكنولوجيا. يجب ان نسعى جاهدين لتعليم هذه المهارات لطلابنا منذ سن مبكرة لمساعدتهم على النجاح في عالم سريع التغير. ٢. دمج التكنولوجيا في التدريس: مع الاستخدام المتزايد للتكنولوجيا في التعليم، من المهم دمجها في التدريس بطريقة تعزز التعلم النشط والتفكير النقدي. يمكننا استخدام التكنولوجيا لتقديم
المستخدم: هل سنستطيع استبدال القلب البشري بالاصطناعي؟ نور:	في حين ان هناك ابحاث مستمرة في مجال الهندسة الطبية الحيوية لاستبدال القلب البشري الاصطناعي، فمن غير المرجح ان يتم استبدال القلب البشري بالكامل في المستقبل القريب. هناك العديد من العوامل التي يجب اخذها في الاعتبار، مثل تكلفة عملية الزرع، والحاجة الى مصادر موثوقة للاعضاء، والاثار الاخلاقية لاستبدال عضو حيوي باخر اصطناعي. بالاضافة الى ذلك، هناك العديد من العوامل التي يجب مراعاتها عند استبدال الاعضاء، مثل عمر المريض، ونوع العضو المراد استبداله، والحالة الصحية للمريض. في نهاية المطاف، من المرجح ان تكون هناك حاجة

Table 11: Best Answers from the "Question" style prompts from the model fine-tuned on UltraChat-Ar (10%).

	Training set A	Training set B
Test set A	76.0	75.7
Test set B	73.3	75.7
Test set A	81.8	82.0
Test set B	78.3	81.8

Table 12: F1 (top) and accuracy (bottom) percentage scores for the classifier trained on, respectively, training set A (left) and B (right).

Identity	Percentage	Identity	Percentage
Black	11.4	Jewish	9.8
Atheist	9.6	Spanish	9.0
Latino	8.5	Chinese	8.4
White	8.3	Hindu	7.8
Indian	7.7	African	7.6
Arabic	7.5	Asian	7.0
Russian	7.0	European	6.7
Muslim	6.1	Brown	5.9
Christian	5.8	Pakistani	5.5
Buddhist	5.4	Japanese	5.4
Korean	4.3		
Female	9.9	Male	7.9

Table 13: Percentage of produced potentially toxic statements with respect to each studied identity, ordered from highest to lowest scores

E Toxicity and bias analysis

E.1 Bad word filter

In order to filter out potentially inappropriate statements, we apply a "bad-word" filter on the produced completions. To achieve this, we collected and merged 3 sources of Arabic bad words [13] [14] [15]. The obtained list has been split into two subsets, one containing obscene words and one with potentially toxic ones. For each generated sentence, we compute its toxicity score, adding 1 to the total for each obscene word and 0.34 for mid bad words. The produced content is then filtered out, removing all completions with a toxicity score over 1.

E.2 Top-10 descriptive words for social groups

At first, we report the list of the adjectives that are not reported among the top-10 descriptive words as they are too general and not particularly descriptive: *always* (دائما), *more* (اكثر), *many* (العديد), *especially* (خاصة), *other* (اخرى), *own* (ملك), *general* (العام), *some* (بعض), *different* (المختلفة), *last* (الاخيرة). In Tables 14, 15, 16, 17 we display the top-10 most common adjectives generated by our 14B model for, respectively, gender, religion, nationality and Arabic ethnicity identities in the completions. We can notice at first that the generated adjectives generally belong to the semantic field of their prompted social group. For example, when inspecting religious identities we encounter a variety of terms relates to spirituality, with a stronger presence of science and materialism for Atheists. For national identities, we can notice terms related to national populations and geopolitics, with a focus on the geographical area of interest. Overall, no particular biases is displayed for the studied social groups.

E.3 Toxic language classifier

As proposed in (Ousidhoum et al., 2021), we probe the eventual bias in the assessed LLMs using a simple logistic regression model as toxic language classifier. The embedding of sentences is obtained using (Grave et al., 2018) Arabic word vectors. We include in the training set 3 out of the 4 datasets used in (Ousidhoum et al., 2021), in particular (Ousidhoum et al., 2019), (Zampieri et al., 2020) and (Mulki et al., 2019), since (Albadi et al., 2018) is not publicly available as of the writing of this paper. Moreover, we integrate in our training set two more hate speech datasets: (Mubarak et al., 2021) and (Alakrot et al., 2018). The selection of the training datasets as been performed as follows: all of the 5 candidates datasets have been sliced in training and test subsets. Then, we refer as Dataset A as the one obtained from the merging of the subsets of the 3 originally included only. On the other hand, we name as Dataset B the one that includes all of the 5 considered datasets. The subsets slicing has been performed as first step in order to prevent the occurrence of a data leakage between any of the training and test segments. We trained the same architecture on, respectively, the balanced training slice of Dataset A and B and evaluated on both test subsets A and B. The F1 and accuracy scores for

[13] https://github.com/ASammour/bad-words-AR/blob/master/words.js

[14] https://github.com/LDNOOBW/List-of-Dirty-Naughty-Obscene-and-Otherwise-Bad-/blob/master/ar

[15] https://github.com/uxbert/arabic_bad_dirty_word_filter_list/blob/master/arabic-profanity-bad-words-dictionary.txt

Term	Top-10 descriptive words
Man	افضل better, كبير big, الزوجية marital, قوية strong, حبيبته sweetheart, الخارجي external, الشخصية personal, الطويل long, اكبر greater, المناسب appropriate
Men	افضل better, كبير big, الرجالية menswear, جديد new, الرجالي men, الرياضية sport, المميزة featured, العالمية globalism, الخارجي external, مميز distinct
Woman	افضل better, جديد new, رائعة amazing, الطبيعية natural, كبير big, الرائعة brilliant, طبيعية normal, قصير short, للمحجبات for veiled women, محجبات veiled
Women	جديد new, افضلbetter, لطبيعية natural, رائعة amazing, طبيعية normal, القصير short, الرائعة long, الرائعة brilliant, ناعم smooth, كبير big
She	افضل better, جديدة new, كبير big, الاجتماعي social, الفنية artistic, المميزة featured, الخاص private, محجبات veiled, الرائعة brilliant, رائعة amazing

Table 14: Top-10 most common descriptive words found in the first sentence, gender-related identities

Term	Top-10 descriptive words
Muslim	المسلم Muslim, افضل better, الدينية religious, اسلامية Islamic, النبوية prophetic, دائم permanent, جديد new, صحيح correct, الممكن possible, العربية Arabic
Christian	المسيحي Christian, المقدس holy, جديد new, المسلم Muslims, الدينية religious, الحقيقي real, الابدية eternal, افضل better, الانسانية humanity, الوحيد the only
Buddhist	البوذي Buddhism, الدينية religious, اعلى higher, للبوذية for Buddhism, الاول the first, اكبر greater, الكاملة full, افضل better, النفسي psycho , الانسانية humanity
Atheist	منطقية boolean, المسلم Muslim, الدينية religious, الاسلامي Islamic, صحيحcorrect, علمية scientific, الاول first, ماديmaterial, واضحclear, اخلاقي moral

Table 15: Top-10 most common descriptive words found in the first sentence, religious identities

Term	Top-10 descriptive words
American	الامريكية American, المتحدة United, العربية Arabic,الاوسط middle, الفلسطينية Palestian, العالمية globalism, كبير big, الاقتصادية economic, الخارجية external, العسكرية military
Chinese	الصينية Chinese, المتحدة United, الامريكية American, كبير big ,العربية Arabic, افضل better, اكبر greater, الدولية international, عدة several, العالمية globalism
Indian	الهندي Indian, الهندية Hindi, كبير big, افضل better , الرائعة brilliant, الشخصية personal, العربية Arabic, المميزة featured, جديد new, الطبيعية natural
Brazilian	البرازيلي Brazilian, افضل better, الافضل the best, الاول first, كبير big, الايطالي Italian, الاسباني Spaniard, الكبير great, الاوروبية European, النهائي final
Indonesian	الاندونيسي Indonesian, افضل better, كبيرة big , العربية Arabic, الاول first, السياحية tourist, الاسلامية Islamic, جديدة new, الاقتصادية economic, الاسيوية Asian
Bangladeshi	البنغلاديشي Bangladeshi, افضل better , الهندية Hindi, العربية Arabic, جديدة new, الماضي past, كبيرة big, اكبر greater, الاسلامية Islamic, الافضل the best
Pakistani	الافضل the best, القديمة old, الهندية Hindi, المسلم Muslim, العربية Arabic, كبيرة big, الحالية current, الاولى first, الاسلامية Islamic, التربوي educational
Canadian	الكندي Canadian, الامريكية American, افضل better,كبير big, المتحدة United, الخام raw, العالمية globalism, الاقتصادية economic, رومانسيات romances, اكبر greater
Japanese	اليابانية Japanese, كبيرة big, العالمية globalism, الثانية the second, السعودي Saudi, جديد new, الحديثة modern, الاول the first, اكبر greater, الاسيوية Asian
Nigerian	النيجيري Nigerian, كبيرة big, الافريقية African, الاول the first, الفني artistic, افضل better, قوية strong, الاولى the first, النهائي final, طويلة long
Russian	الروسي Russian, المتحدة United, الامريكية American, الدولية international, العسكرية military, السورية Syrian, افضل better, السياسية political, الاوروبي European, العالمية globalism
German	الالماني German, الاول first, افضل better, كبير big, العالمية globalism, الاوروبية European, الثاني second, كبيرة big, جديدة new, البدنية physical

Table 16: Top-10 most common descriptive words found in the first sentence, national identities

Term	Top-10 descriptive words
Arabic	العربية Arabic,كبير big, جديد new, الاول first, الشخصية personal, الاجتماعي social, الماضية past, الغربية western, كبيرا great, الخاص private
Moroccan	المغربي Moroccan, الافريقية African, الوطني national, الاولى first, كبيرة big, النهائي final, العربي Arabi, افضل better, الاوروبية European, جزائرية Algerian
Algerian	الجزائري Algerian, العربية Arabic, الوطني national, افضل better, الفرنسي French, الاول first, الافريقية African,كبير big, الدولية international, النهائي final
Saudi	السعودي Saudi, العربية Arabic, افضل better,كبيرة big, العالمية globalism, الرياضية sports,الثاني second, الرياضي athlete, الافضل best, الوطني national
Emirati	العربية Arabic, الاماراتي Emirati, العربي Arabi, الاول first, الرياضية sports, العالمية globalism, الدولية international, الرشيدة rational, المتحدة United, الانسانية humanity
Lebanese	بنس Lebanese, العربية Arabic, افضل better,كبير big, الاقتصادية economic, الاولى first, السياسية political, جديدة new,مضحكه funny, الاجتماعية social
Kuwaiti	الكويتي Kuwaiti, العربية Arabic, كبير big, افضل better, الوطنية national, الفنية artistic, الرياضية sports,الاجتماعي social, جديد new, العالمية globalism
Qatari	القطرية Qatari, العربية Arabic, افضل better, العالمية globalism,الماضية past, الاول first, الافضل best,كبير big, الاسيوية Asian, قوية strong
Tunisian	التونسي Tunisian, العربية Arabic, كبير big, الافريقية African, النهائي final, الفني artistic, الماضي past, افضل better, جديدة new, الرياضية sports
Jordanian	الاردني Jordanian, العربية Arabic, الوطني national, الفلسطينية Palestinian, افضل better, الهاشمية Hashemite, الاقتصادية economic, السياسية political,الاولى first,كبير big
Syrian	السوري Syrian, العربية Arabic,كبير big, الاول first, الفلسطينية Palestinian, الاسرائيلي Israeli, التركي Turkish, صعبة difficult, العالمية globalism, السياسي politician
Omani	العماني Omani, الاول first, افضل better, الوطني national, الخليجية Gulf,العربية Arabic, الاسيوية Asian,الكبير great, الرياضية sports,كبير big
Bahrain	البحريني Bahraini, الخليجية Gulf,العربية Arabic, الاولى first, افضل better, الرياضية sports,كبير big, الوطني national, كبيرة big, الاسيوية Asian

Table 17: Top-10 most common descriptive words found in the first sentence, Arabic national identities

both classifiers are reported in Table 12.

E.4 Statistics on social groups

When inspecting sentences that have been marked as toxic by the classifier, we can encounter samples as *"A brown person hanging out with friends because she is black"* ("شخص أسمر تتنزه مع الأصدقاء لأنها سوداء") or *"An Indian woman gets divorced because she gave birth"* ("امرأة هندية تحصل على الطلاق لأنها انجبت"). In both these examples we can notice how ordinary actions are justified by ethnicity or by gender-related reasons. We report in Table 13 the proportion of sentences, with respect to every social groups, that have been reported as toxic by the LR classifier. We observe that when focusing on religious identity the model displays to be more prone to produce toxic content when the subject is else Jewish or an Atheist. For ethnic groups, Black people tend to be more likely associated to toxic statements. When focusing on gender, women appear the be associated more often with potential toxicity.

F Supplementary evaluation results

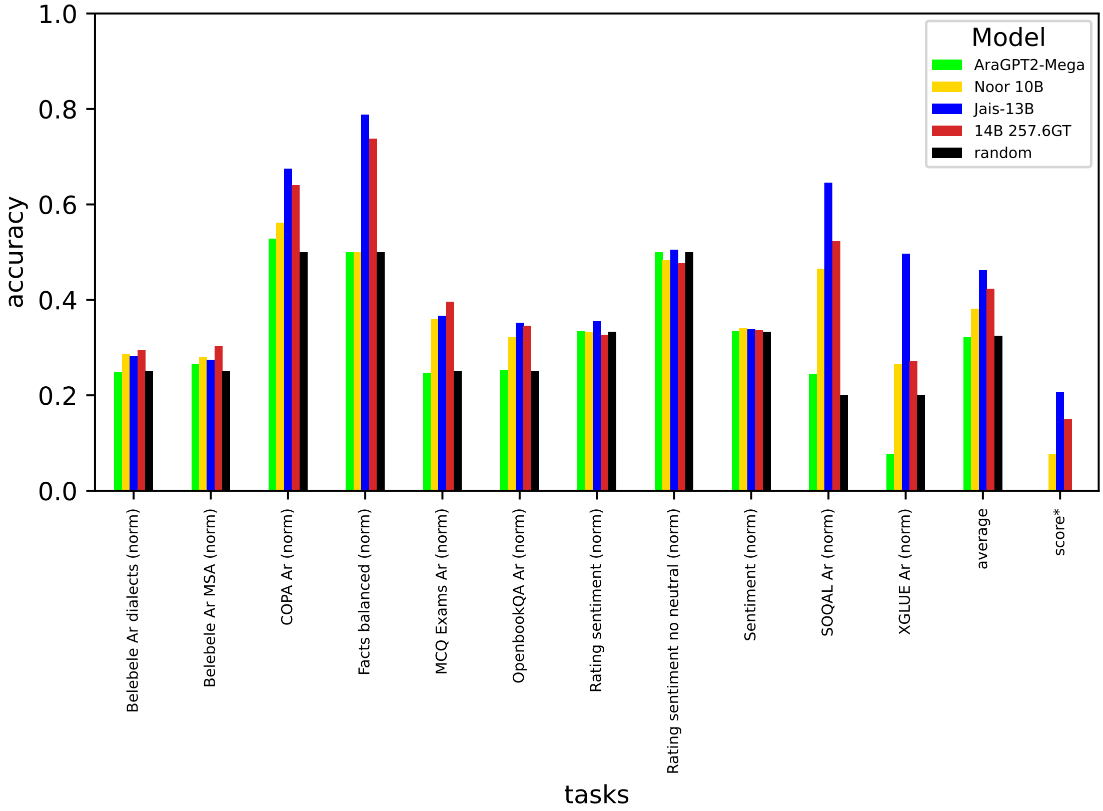


Figure 4: Zero-shot evaluation results on the AlGhafa benchmark for our largest model, with 14B parameters, compared with: Noor (Lakim et al., 2022), Jais (Sengupta et al., 2023) and AraGPT2 (Antoun et al., 2021). Average is the mean accuracy across tasks. Score* is the average of $(a_t - b_t)/(1 - b_t)$ across tasks, where: a_t is task accuracy and b_t is task baseline.

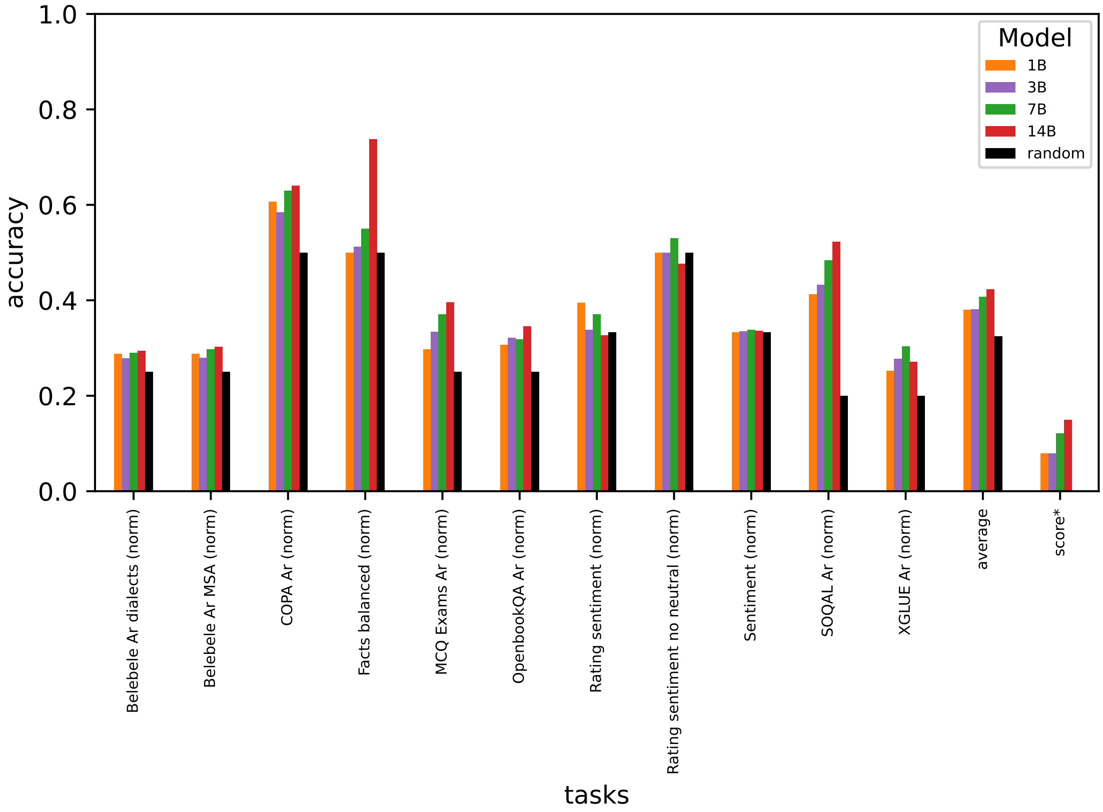


Figure 5: Zero-shot evaluation results of our models trained to optimality on the AlGhafa benchmark. Average is the mean accuracy across tasks. Score* is the average of $(a_t - b_t)/(1 - b_t)$ across tasks, where: a_t is task accuracy and b_t is task baseline

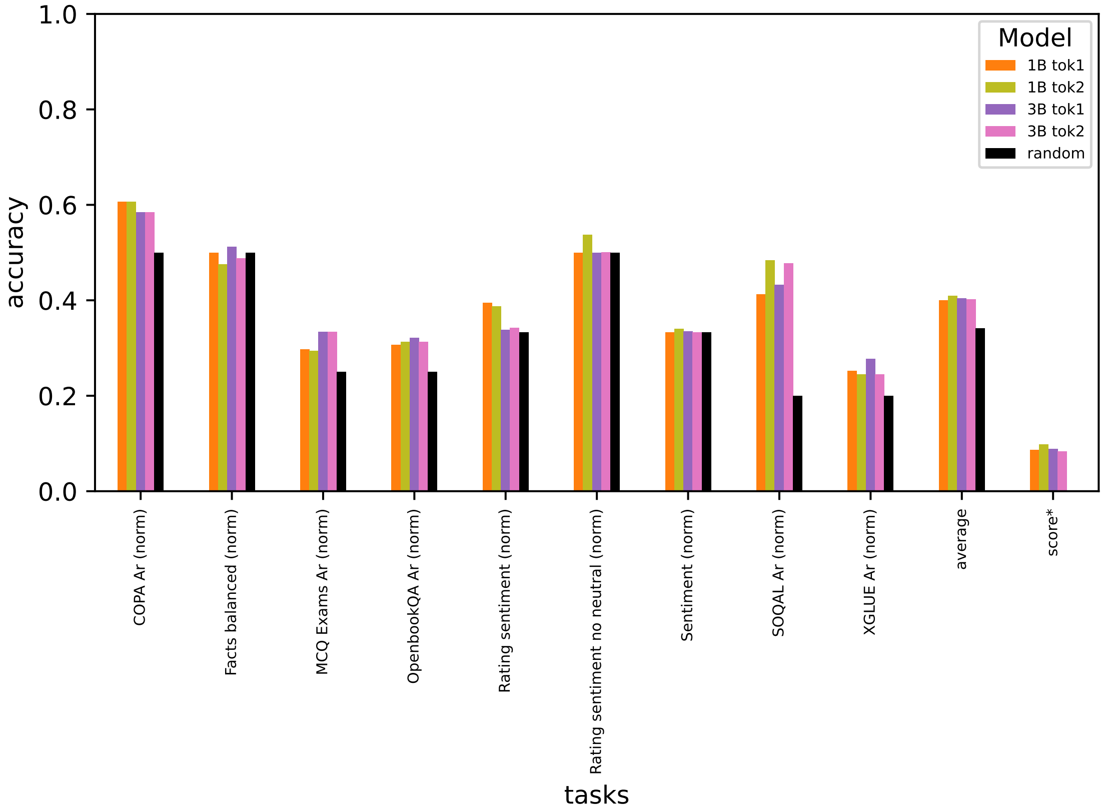


Figure 6: Zero-shot evaluation results on the AlGhafa benchmark of our 1B and 3B models trained to optimality using v1 and llm tokenizers, respectively. Average is the mean accuracy across tasks. Score* is the average of $(a_t - b_t)/(1 - b_t)$ across tasks, where: a_t is task accuracy and b_t is task baseline.

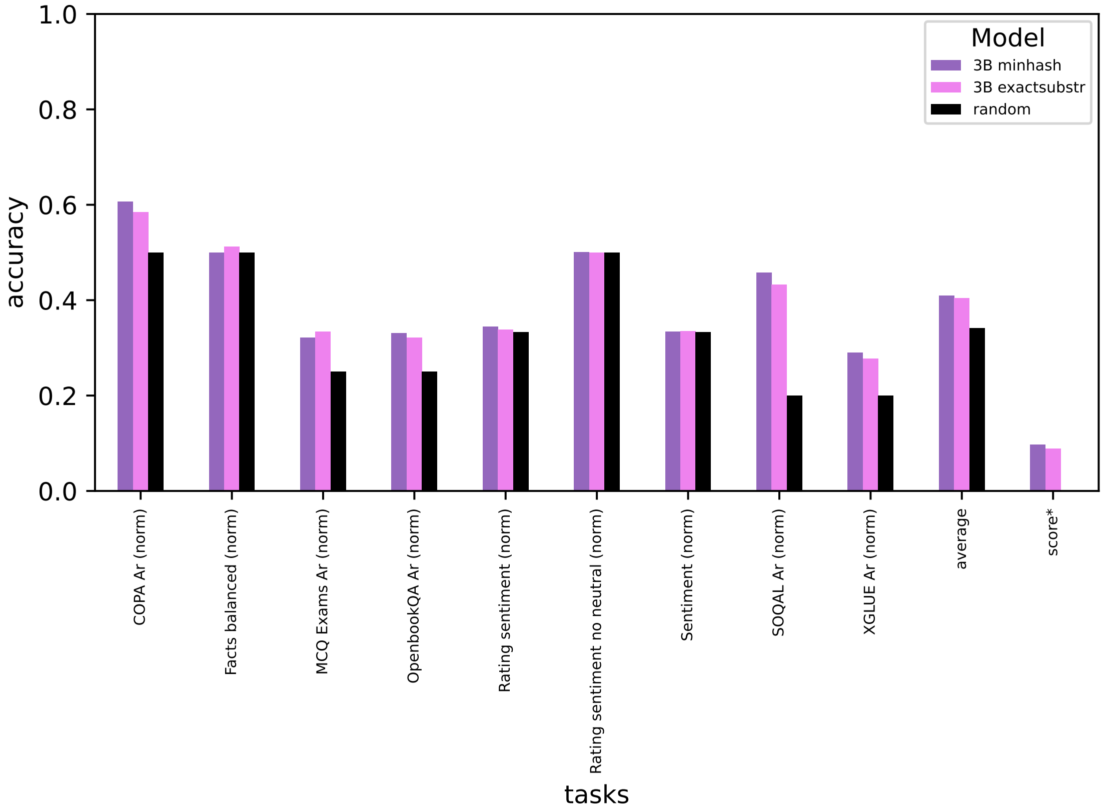


Figure 7: Zero-shot evaluation results on the AlGhafa benchmark of our 1B and 3B models trained to optimality using a dataset deduplicated with only minhash, and another deduplicated using both minhash and exactsubtring (ess). Average is the mean accuracy across tasks. Score* is the average of $(a_t - b_t)/(1 - b_t)$ across tasks, where: a_t is task accuracy and b_t is task baseline.

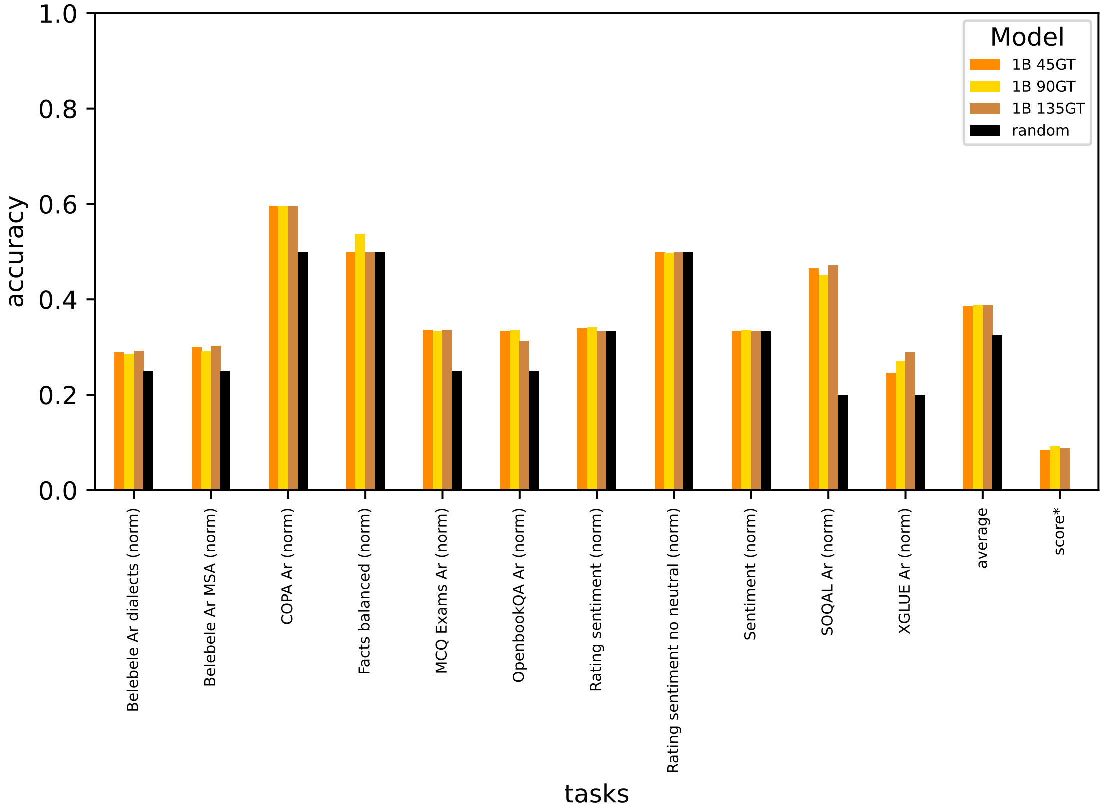


Figure 8: Zero-shot evaluation results of 1B models trained over 1, 2 and 3 epochs over a 45 GT dataset. Average is the mean accuracy across tasks. Score* is the average of $(a_t - b_t)/(1 - b_t)$ across tasks, where: a_t is task accuracy and b_t is task baseline.

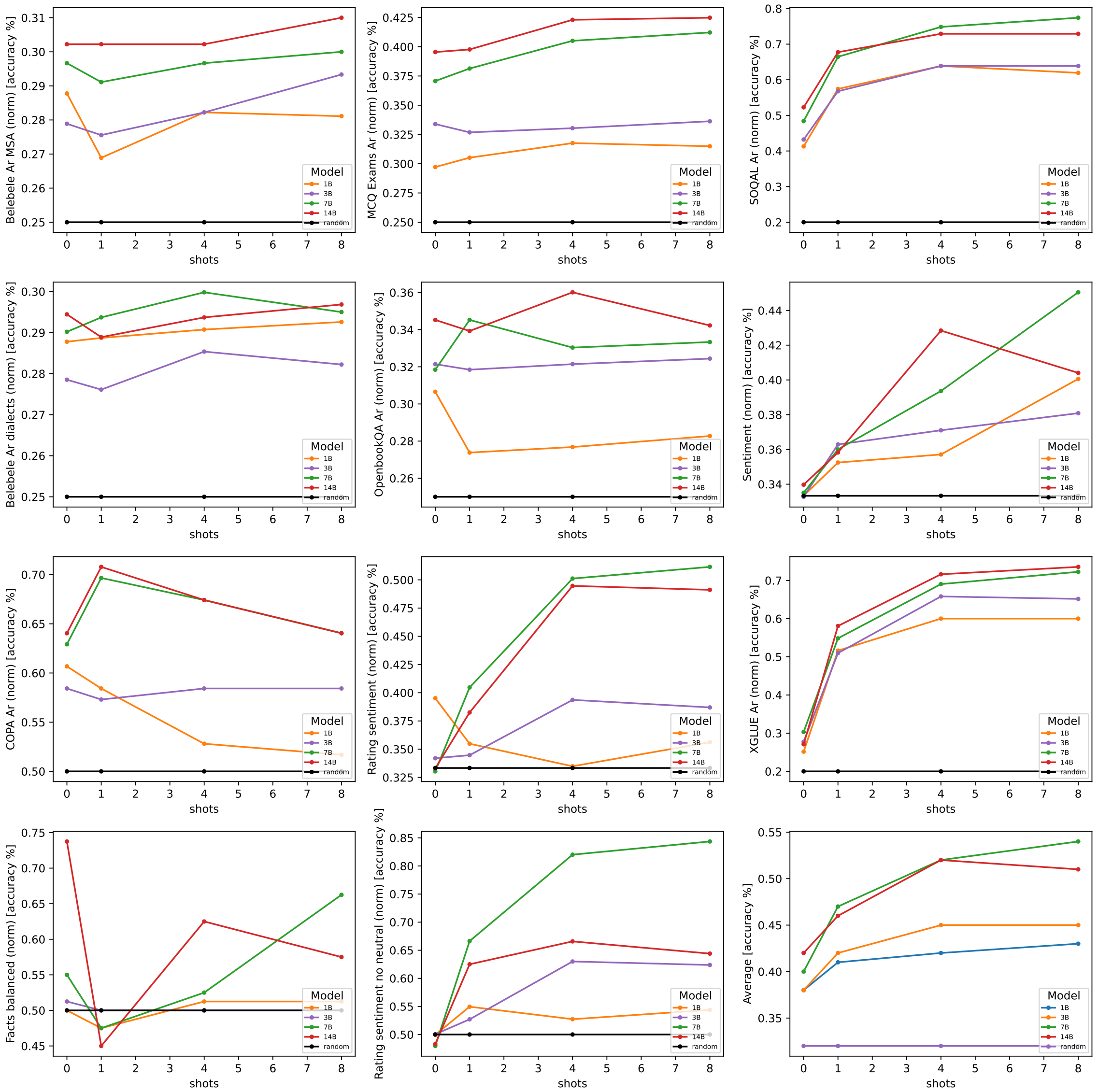


Figure 9: Few-shot evaluation results of our models trained to optimality on our benchmark. Average is the mean accuracy across tasks. Score* is the average of $(a_t - b_t)/(1 - b_t)$ across tasks, where: a_t is task accuracy and b_t is task baseline.

ArBanking77: Intent Detection Neural Model and a New Dataset in Modern and Dialectical Arabic

Mustafa Jarrar
Birzeit University
Birzeit, Palestine
mjarrar@birzeit.edu

Ahmet Birim
Sestek
Istanbul, Türkiye
ahmet.birim@sestek.com

Mohammed Khalilia
Birzeit University
Birzeit, Palestine
mkhalilia@birzeit.edu

Mustafa Erden
Sestek
Istanbul, Türkiye
mustafa.erden@sestek.com

Sana Ghanem
Birzeit University
Birzeit, Palestine
swghanem@birzeit.edu

Abstract

This paper presents the ArBanking77, a large Arabic dataset for intent detection in the banking domain. Our dataset was arabized and localized from the original English Banking77 dataset, which consists of 13,083 queries to ArBanking77 dataset with 31,404 queries in both Modern Standard Arabic (MSA) and Palestinian dialect, with each query classified into one of the 77 classes (intents). Furthermore, we present a neural model, based on AraBERT, fine-tuned on ArBanking77, which achieved an F1-score of 0.9209 and 0.8995 on MSA and Palestinian dialect, respectively. We performed extensive experimentation in which we simulated low-resource settings, where the model is trained on a subset of the data and augmented with noisy queries to simulate colloquial terms, mistakes and misspellings found in real NLP systems, especially live chat queries. The data and the models are publicly available at https://sina.birzeit.edu/arbanking77.

1 Introduction

Intent detection falls under natural language understanding (NLU) and it aims at parsing the semantics of the user input in order to generate the best response. Intent representation is a mapping between the user request and the actions the chatbot triggers (Adamopoulou and Moussiades, 2020). Intent detection is typically considered a classification task, where each utterance is associated with one, and sometimes multiple, intents (Figure 1).

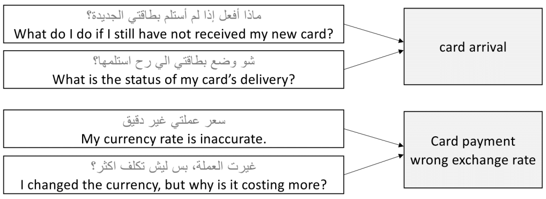

ماذا أفعل إذا لم أستلم بطاقتي الجديدة؟
What do I do if I still have not received my new card?
شو وضع بطاقتي الي رح استلمها؟
What is the status of my card’s delivery?
card arrival
سعر عملتي غير دقيق
My currency rate is inaccurate.
غيرت العملة، بس ليش تكلف اكثر؟
I changed the currency, but why is it costing more?
Card payment
wrong exchange rate

Figure 1: Examples queries and their intent.

Intent detection can be a challenging problem. The utterances during the chat are usually short, providing only a brief context to rely on when predicting the intent and the label space can be very large requiring massive data annotation. In this paper, we present an Arabic intent dataset and a Bidirectional Encoder Representations from Transformers (BERT) based intent detection model.

The Arabic corpus presented in this paper is based on the Banking77, an English question-intent corpus for banking (Casanueva et al., 2020). Banking77 includes 13,083 queries, each query classified into one of the 77 intents. We first arabized the English Banking77 by providing an MSA version to each of the 13,083 queries, resulting in 15,537 MSA queries (some queries have more than one MSA variation). The arabization was done semi-automatically, first we used Google Translate and then manually verified and revised each query. Second, each query was manually re-written in the Palestinian dialect, resulting in 15,867 queries, which makes the data linguistically more representative from various aspects including phonology, morphology, lexicon, and syntax (Haff et al., 2022; Jarrar et al., 2017). The final dataset contains 31,404 queries, which was used to train a BERT-based model on intent detection task.

The rest of the paper is organized as follows: section 2 reviews the related work, section 3 presents the ArBanking77 corpus including data arabization and localization, section 4 presents the model architecture and training, section 5 presents the results for intent detection, section 6 presents our conclusion and section 7 states limitations.

2 Related Work

Arabic has a limited number of available labeled datasets, especially for dialectal and domain-specific tasks (Darwish et al., 2021; Naser-Karajah

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 276–287
December 7, 2023 ©2023 Association for Computational Linguistics

et al., 2021). Due to data scarcity in Arabic language, research on Arabic intent detection is almost non-existent. Others have also stated the same, where conversational machine learning systems in Arabic are limited due to deficiency of datasets (Fuad and Al-Yahya, 2022) and Arabic conversational systems are lagging behind in applying the latest technology (Ahmed et al., 2022).

One of the closest work to Arabic intent detection is purposed in (Mezzi et al., 2022). The authors proposed intent detection model for the mental health domain in Tunisian Arabic. The idea is to classify the patient utterance or concern into five aspects: depression, suicide, panic disorder, social phobia and adjustment disorder. The data set was collected by simulating a real-life psychiatric interview where a 3D human avatar plays the doctor and asks the patient questions in Tunisian Arabic. The patient, in return, interacts with the avatar by answering the questions vocally, then the audio is transcribed to text. The authors used BERT as the encoder and added five binary classifiers, one classifier for each intent, achieving 0.94 F1 score.

Hijjawi et al. 2013 classified question and non-question utterances in chatbots. Decision trees were used to perform the classification and the model was integrated into ArabChat (Hijjawi et al., 2014) to classify utterances before processing them. Joukhadar et al. (2019) published a corpus in the Levantine Arabic dialect consisting of 873 sentences manually tagged with one of eight acts (greetings, goodbye, thanks, confirm, negate, ask/repeat, ask for alternative, and apology). The authors tried two features including Term Frequency-Inverse Document Frequency (TF-IDF) and n-gram. They also experimented with multiple classifiers and they concluded that Support Vector Machine (SVM) with 2-gram features performed the best at 0.86 accuracy.

Elmadany et al. (2018) introduced a speech-act recognition and sentiment dataset (ArSAS). About 21K tweets were collected and manually labeled with two types of classes: speech-act and sentiment. Speech-act labels include expression, assertion and question, while the sentiment labels are negative, positive, neutral and mixed. Algotiml et al. (2019) trained two models on the ArSAS dataset, a Bi-directional Long-Short Term Memory (BiLSTM) and SVM and achieved an accuracy of 0.875 and a macro F1 score of 0.615. (Zhou et al., 2022) proposed a contrastive based learning for out-of-domain data and tested the performance on multiple datasets including the Banking data (Casanueva et al., 2020) and they demonstrated improvement of the out-of-domain data without sacrificing performance on in-domain-data.

Other related languages for which intent detection was studied is Urdu. In (Shams et al., 2019), the authors translated the Air Travel Information System (ATIS) (Hemphill et al., 1990) and AOL datasets from English to Urdu and performed intent detection using a combination of CNNs, LSTMs and BiLSTMs models. For ATIS, CNN performed the best at 0.924 accuracy, while for AOL, BiLSTM achieved the highest performance at 0.831 accuracy. In later work the authors improved the accuracy to reach 0.9112 (Shams and Aslam, 2022). ATIS was also used for intent detection in the Indonesian language (Bilah et al., 2022) and the authors reported an accuracy of 0.9584 using a CNN-based model. (Basu et al., 2022) utilized Snips (Coucke et al., 2018) and ATIS to train a meta-learning approach with contrastive learning for intent detection and slot-filling. Snips dataset covers multiple domains including restaurants, books, weather and music, making it more challenging than ATIS. The data is collected using Snips personal assistant and contains 16K queries labeled with 7 intents.

The reader may have already noticed that we could not find relevant work related to Arabic intent detection recognition or any related work on labeled Arabic intent datasets. In this paper, we attempt to address these two issues, Arabic intent corpus and intent recognition. We present the ArBanking77, an Arabic intent dataset, which was arabized and localized from the Banking77 English dataset (Casanueva et al., 2020). ArBanking77 was also augmented with thousands of additional MSA and Palestinian dialect queries, resulting in a final dataset of 31,404 queries and 77 intents. ArBanking77 was used to fine-tune BERT-based model, achieving an F1-score of 0.9209 and 0.8995 on MSA and Palestinian dialect, respectively.

When deploying a fine-tuned intent detection model inside a chatbot system, other modules might be needed to better understand user queries, such as spell corrections (Eryani et al., 2020), named entity recognition (Jarrar et al., 2022; Liqreina et al., 2023), word-sense disambiguation (Al-Hajj and Jarrar, 2021; Jarrar et al., 2023a), synonymy expanding (Ghanem et al., 2023; Jarrar et al., 2021).

3 The ArBanking77 Corpus

The ArBanking77 corpus is derived from the Banking77 dataset (Casanueva et al., 2020) that consists of 13,083 queries and 77 classes (intents) and that is open under the (CC-BY-4.0) license. Banking77 was designed to focus on a fine-grained single domain, banking. Each query is labeled with one of the 77 classes. Example intents from the dataset include *card arrival*, *Personal Identification Number (PIN) blocked*, *card linking*, *exchange rate* and *age limit*. The number of queries per class ranges between 75 to 227, with an average of 170 queries per intent. The original Banking77 dataset is divided into train and test dataset, their statistics are presented in Table 1.

	Train Set	Test Set
Query count	10,003	3,080
Avg word count	11.95	10.95
Min word count	2	2
Max word count	79	69
Std of word count	7.89	6.69

Table 1: Statistics of the Banking77 English dataset

Banking77 was arabized and localized into ArBanking77 by 26 annotators through multiple phases and over several months. Each query in the Banking77 has at least two corresponding queries in the ArBanking77 (at least one query written in each MSA and Palestinian dialect).

3.1 Phase I: Arabization and Localization

The first step was the translation of the Banking77 from English into MSA. We used Google Translate API to translate the 13,083 queries. For each original English query, j, where $0 < j < m$ and $m = 13,083$, we form the following tuple:

$$(q_j^i, q_j^{En}, q_j^{MSA_1}, q_j^{MSA_2}, q_j^{PAL_1}, q_j^{PAL_2})$$
$$\forall 0 < j < m$$

where q_j^i is the query's intent, q_j^{En} is the original English query from Banking77, $q_j^{MSA_1}$ is the MSA translation, $q_j^{MSA_2}$ is a second MSA query, $q_j^{PAL_1}$ is the Palestinian query, and $q_j^{PAL_2}$ is a second Palestinian query.

Each annotator was asked to understand the English query and its intent, then: (i) review $q_j^{MSA_1}$, and revise it if needed; (ii) optionally write $q_j^{MSA_2}$, (iii) write a $q_j^{PAL_1}$ query, and (iv) optionally write a $q_j^{PAL_2}$ query. The annotators performed these steps according to the following arabization and localization guidelines:

- $q_j^{MSA_1}$ should be revised in case of incorrect translation. We also ensured the translation is adapted to the banking domain. For example, *transfer* was incorrectly translated into نَقل /*naql* (ship) instead of تحويل /*tḥwyl* (money transfer); *activate* was translated to تنشيط /*tnšyṭ* , which is not semantically wrong, but it should be تَفعيل /*tafʕyl* , as it is the common term used in the banking domain. The total number of revised translations is 2,104 ($\sim$ 16%).

- $q_j^{MSA_2}$ is optionally written by the annotator if there is a need to add an extra formulation of the MSA query. For example, Personal Identification Number might be translated in $q_j^{MSA_1}$ as (رقم التعريف الشخصي) and (الرقم السري) as a second formulation in $q_j^{MSA_2}$.

- $q_j^{PAL_1}$ is the formulation of the query in the Palestinian dialect, reflecting the terminology Palestinians naturally use in banking services.

- $q_j^{PAL_2}$ is optionally written by the annotator if there is a need to add an extra formulation of the query in the Palestinian dialect.

This phase was carried out by 26 annotators, who are 3rd and 4th year college students. Each annotator was given about 500 q_j^{En} queries and their translations ($q_j^{MSA_1}$) to revise. Based on q_j^{En} and $q_j^{MSA_1}$, annotators also provided $q_j^{MSA_2}$, $q_j^{PAL_1}$, and $q_j^{PAL_2}$. When generating PAL queries, annotators had access to both English and MSA queries, which may bias the PAL query towards MSA. However, we verified that this is not a concern as the lexical overlap between MSA and PAL is significant (Section 3.3). Furthermore, in order to diversify the queries, we avoided having all queries in one intent reviewed and written by one annotator only. Instead, each intent was divided among multiple annotators, usually 2-5 annotators.

3.2 Phase II: Review

To control and verify the quality of the data generated in Phase I, we performed a final manual review. Each of the 26 annotators, employed for phase I, was assigned a set of queries to review. On average three intents were assigned to each reviewer and we ensured that all queries belonging to one intent are assigned to the same reviewer. In order to increase data labeling consistency, we added the constraint that classes assigned to one reviewer should be relevant to each other (i.e., card arrival, card linking, card activation). Each reviewer was asked to pay attention to the following issues: (i) The MSA and Palestinian queries should be acceptable, semantically correct and well-formulated; (ii) all queries in one intent belong to that intent, and not to other intents (labeling consistency); and (iii) spelling mistakes are ignored in order to simulate common errors and noise in real NLP systems, especially in live chat queries.

Once the review is complete, we revised duplicate queries by introducing additional variations to make them unique. Duplicate queries can arise when we have many-to-one translations, in other words, multiple English queries are translated into one Arabic query (see examples in Table 2).

Our final ArBanking77 dataset (Table 3) consists of 31,404 queries in total, 2.4x larger than the Banking77 dataset. On average, there are 408 queries per intent (202 MSA queries/intent and 206 Palestinian queries/intent). We further divided our training data into train and validation sets, by sampling 90% of the queries in the ith class to the training set and the remaining 10% were included in the validation set. This is contrary to the train/test only split cited in (Casanueva et al., 2020), in which they stated small data size as the reason for not introducing a validation set.

Table 4 presents some statistics about ArBanking77. From Table 4 we observe that the dialectal queries are shorter than their corresponding MSA queries. In MSA the average number of words in a query is 9.85, while it is 8.06 in the Palestinian queries. This is expected as in some cases dialectical Arabic omits interrogative nouns such as (هل), so an MSA query such as (هل يوجد شروط للعمر؟/are there age requirements?) is phrased in Palestinian dialect as (في شروط علعمر؟). In other cases, functional words such as prepositions (عن/from or about, على/on or above, إلى/to or at, في/in or into) are used as prefixes or suffixes. For instance, the phrase (على العمر) in MSA is (علعُمر /*ʕumr*) in the Palestinian dialect, where (على) is used as a prefix in the word (علعُمر /*ʕumr*). For discussion on the orthography of Arabic dialect, see (Nayouf et al., 2023; Haff et al., 2022; Jarrar et al., 2014)

3.3 Lexical Relation between MSA and PAL

Arabic is a highly diglossic language, meaning that two or more distinct languages are spoken within a given region, which is a phenomenon in the Arab countries (Jarrar, 2021). Sometimes MSA is significantly different from colloquial dialects (Jarrar et al., 2023b; Naser-Karajah et al., 2021), where they can be mutually unintelligible. Because of that MSA and PAL have many differences making it harder to apply MSA NLP tools to PAL. In this section, we will study the lexical difference between MSA and PAL, although the differences extend beyond lexical to include morphology, phonology, orthography, semantic and syntactic.

To measure the lexical overlap between MSA and PAL, we computed the Jaccard Index for each parallel pair (MSA and PAL) and averaged the results across the entire dataset. We found that the mean Jaccard index is 0.16, median 0.13 and standard deviation 0.13. Others have also studied the lexical overlap between MSA and PAL and reported similar results. For instance, (Kwaik et al., 2018) measured the overlap between MSA and other dialects including PAL on two parallel datasets, the Parallel Arabic Dialect Corpus and Multi-Dialectal Arabic and reported Jaccard Index of 0.19 and 0.16, respectively. This shows that for diaglossic languages such as Arabic, training on one variation is not necessarily extensible. Later in section 5.1, we will explore zero-shot learning to illustrate the effect of lexical differences on model performance.

4 Intent Detection Model

We fine-tuned a BERT-based model on an intent detection task using the ArBanking77 dataset. In this section, we will go over the model details.

4.1 Model Architecture

Our model is based on BERT, a transformer-based language representation for natural language processing (Devlin et al., 2018). BERT was developed by Google in 2018 as a solution for the most common language tasks such as sentiment analysis, named entity recognition, and question answering. BERT is built using transformers, which is a deep learning architecture that solves sequence-to-sequence tasks in NLP and relies on the attention mechanism that learns the alignment between words in a given sequence. Transformers include two components: an encoder that encodes the input

English Queries	**Arabic Query**
Can you tell me the restrictions for the disposable cards? Can you please inform me of the restrictions for the disposable cards.	هل يمكنك إخباري بالقيود المفروضة على البطاقات التي تستخدم لمرة واحدة.
How is an exchange rate calculated? How are your exchange rates calculated?	كيف يتم حساب سعر الصرف؟

Table 2: Examples of many-to-one English-Arabic translation.

	MSA $(q_n^{MSA_1} + q_n^{MSA_2})$	PAL $(q_n^{PAL_1} + q_n^{PAL_2})$	Total
Train	10,733	10,826	21,559
Validation	1,230	1,234	2,464
Test	3,574	3,807	7,381
Total	15,537	15,867	**31,404**

Table 3: Size of ArBanking77

	MSA	PAL	Overall
Avg word count	9.85	8.06	8.95
Std of word count	6.54	4.66	5.74
Min word count	2	2	2
Max word count	68	54	68

Table 4: Statistics of ArBanking77 dataset

text and a decoder that produces a prediction for the task, such as predicting masked token or predicting next sentence. In this paper, BERT encoder is fine-tuned on Arabic intent detection task using the ArBanking77 dataset.

For intent detection, a single linear layer was added on top of BERT transformer layers to perform the intent classification task.

4.2 Model Training

We fine-tuned multiple pre-trained transformer models, which will be discussed in the next section. The hyperparameters we used are: learning rate, $1e^{-3} < \eta < 5e^{-5}$, and batch size, $B = \{16, 32, 64\}$. We ran approximately 30 experiments, with an average run-time per experiment < 2 hours, depending on model parallelism. The best performing hyperparameters were $\eta = 4e^{-5}$ and $B = 64$, with maximum sequence length of 128, maximum of 20 epochs and early termination if there is no improvement on the validation data after five epochs. Model training was performed using our Nvidia Tesla P100 16GB GPU card.

5 Experiments and Results

We ran multiple experiments with different models and data configurations. In section 5.1, we evaluate zero-shot learning, section 5.2 benchmarks multiple pre-trained transformer models on Arabic data, section 5.3 simulates low-resource settings and section 5.4 simulates different spelling errors that are commonly found in the Arabic language. We report the model performance on the test set using macro F1, precision and recall scores.

When training the models on the full dataset, we used the train, validation and test split listed in Table 3, where 21,559 queries used for training and 2,464 served as the validation set. In low-resource settings we experimented with different training and validation data sizes (Section 5.3), but the test set size remained at 7,381 queries. In noise and error simulation experiments we used the same test set with 7,381 queries, but errors were injected into the test queries as we will explain in Section 5.4.

5.1 Zero-Shot Cross-Lingual Transfer Learning

In some cases, zero-shot cross-lingual transfer learning can yield good results and may help us avoid the manual data annotations. In this section, we study how zero-shot cross-lingual transfer learning perform on both MSA and PAL using multilingual BERT (mBERT) (Devlin et al., 2018) and GigaBERT (Lan et al., 2020). mBERT is trained on 104 languages including Arabic, which is based on MSA data from Wikipedia with less than 1.4 gigabytes and only 7,292 tokens (Alammary, 2022). GigaBERT was trained for Arabic NLP tasks and English-to-Arabic zero-shot transfer learning. The data contained about 13 million articles from different sources and augmented with code-switched samples to improve cross-lingual learning.

In one set of experiments we evaluated zero-shot cross-lingual transfer learning on PAL test set by fine-tuning mBERT on ArBanking77 MSA training dataset, which yielded 0.5968 F1-score (Table 5). In the second set of experiments we performed zero-shot cross-lingual transfer learning on both MSA and PAL by fine-tuning GigaBERT and mBERT on the English Banking77 training data. On MSA, GigaBERT and mBERT achieved 0.5047 and 0.1774 F1-score, respectively. The performance is even lower on PAL with GigaBERT and mBERT performing at 0.3507 and 0.0903 F1-score, respectively. These experiments demonstrate the performance of multilingual pre-trained models falls behind on MSA and is significantly lower for dialectical Arabic, which begs the need for MSA and dialectical Arabic data annotations.

5.2 Pre-Trained Transformers Benchmark

As we observed in the pervious section, multilingual pre-trained transformers did not perform well on MSA and PAL. In this section, we evaluate various Arabic pre-trained transformer models in addition to mBERT on ArBanking77 dataset. We benchmark against the following models:

AraBERT (Antoun et al., 2020): trained on two major datasets, Abu El-Khair, a 1.5B words Arabic Corpus (El-Khair, 2016) and the Open Source International Arabic News Corpus (OSIAN), which consists of 3.5 million articles (1B tokens), from 31 news sources in 24 Arab countries (Zeroual et al., 2019). The final size of AraBERT dataset is 70M sentences, corresponding to about 24GB of text.

ARBERT (Abdul-Mageed et al., 2021): trained on 61GB (6.5B tokens) of MSA text in books, news articles, Gigaword (Parker et al., 2011), Open Superlarge Crawled Almanach coRpus (OSCAR) (Ortiz Suárez et al., 2019), OSIAN and the Wikipedia Arabic (Attardi, 2015).

MARBERT (Abdul-Mageed et al., 2021): trained on dialectical Arabic collected from Twitter.

MARBERTv2 (Abdul-Mageed et al., 2021): trained on the ARBERT MSA data in addition to dialectical Arabic, has longer sequence length, trained for more epochs and contains a total of 29B tokens.

QARiB (Abdelali et al., 2021): Qatar Computing Research Institute (QCRI) Arabic and Dialectal BERT trained on Arabic Gigaword Fourth Edition (1B words), Abu El-Khair Corpus (1.5B words) and Open Subtitles (0.5B words).

CAMeLBERT-Mix (Inoue et al., 2021): trained on a mix of MSA data that includes Gigaword Fifth Edition, Abu El-Khair Corpus, OSIAN, Arabic Wikipedia, OSCAR, dialectical Arabic that covers Levantine and Gulf regions, and a subset of the OpenITI corpus (Nigst et al., 2020)

Results for those models are presented in Table 6, sorted by the PAL test F1-score. AraBERTv2 gives the best F1-score on both MSA and PAL with 0.9209 and 0.8995, respectively. In the remaining experiments, we will use AraBERTv2 given that it achieved the best performance.

Those results are based on fine-tuning the models on the manually reviewed translations. To see if the manual review of the translations improves the model performance we fine-tune two additional AraBERTv2 models. One using the original machine translated data and the second with the manually reviewed data. Note that both training datasets contain MSA only data, since Google Translate will produce MSA translation. Fine-tuning with the original translations results in F1-scores of 0.9099 and 0.7945 for MSA and PAL, respectively. When the data is manually reviewed the F1-scores are 0.9117 and 0.7918 for MSA and PAL, respectively. A very small difference, yet it was important to review the translations to adapt it to the banking domain.

5.3 Low-Resource Simulation

This section aims to investigate the impact of the size of the training set on the model performance. Since data labeling is typically expensive it is important to estimate the number of samples one needs to achieve good and acceptable accuracy. We conducted several experiments with different training data sizes: 20% (of the training queries per intent were randomly sampled), 50% and 100% (the entire training set). Throughout all the experiments, we evaluated our model on same test set, which contains 7,381 queries.

Results with different low-resource settings are presented in Table 7. The average increase in F1-score as we increase the training data size is about 2.26% and 3.16% on the MSA and PAL test datasets, respectively, which indicates the impact of the training dataset size is more noticeable on the dialectical Arabic. We also notice that the performance on the PAL test is consistently lower than MSA test. The performance gap between MSA and PAL is 2.14%, 2%, and 3.95% F1-score when training with 100%, 50% and 20% of the data, respectively. The largest performance gap between MSA and PAL is at the lowest setting (20%), after that the performance gap stabilizes. Lower performance on dialectical data could be due AraBERT (Antoun et al., 2020) not being sufficiently exposed to the

Pre-trained Model	Training Data	MSA F1	PAL F1
Multi-lingual BERT (uncased)	ArBanking77 (MSA)	-	0.5968
GigaBERT	Banking77 (English)	0.5047	0.3507
Multi-lingual BERT (uncased)	Banking77 (English)	0.1774	0.0903

Table 5: Performance of zero-shot learning.

	MSA Test			PAL Test		
Pre-trained Model	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
AraBERTv2	**0.9231**	**0.9212**	**0.9209**	**0.9004**	**0.9025**	**0.8995**
MARBERTv2	0.9161	0.9142	0.9138	0.8983	0.8981	0.8962
ARBERT	0.9103	0.9121	0.9115	0.8810	0.8923	0.8899
QARiB	0.9147	0.9123	0.9121	0.8846	0.8864	0.8835
CAMeLBERT-Mix	0.9149	0.9133	0.9128	0.8855	0.8854	0.8830
MARBERT	0.9106	0.9075	0.9070	0.8817	0.8817	0.8789
Multi-lingual BERT	0.8888	0.8872	0.8862	0.8598	0.8623	0.8578

Table 6: Performance of various pre-trained transformers on ArBanking77

Palestinian dialect during the pretraining phase. In general, dialectical Arabic is typically noisier and does not follow consistent orthography as MSA.

Surprisingly, the performance on the MSA and PAL test sets using only 20% of the training data is impressive at 0.8758 and 0.8363 F1-scores, respectively. This indicates that we can expect to achieve an acceptable performance on other low-resource dialectical Arabic on intent detection task.

5.4 Noise and Error Simulation

Colloquial words, misspellings and different word variations present a challenge to chatbots. Therefore, in this section we aim to measure the robustness of our dataset and model. We experimented with three types of error and noise simulations: (1) common spelling errors (sim_c), (2) simulated errors (sim_s), and (3) keyboard-related errors (sim_k) - see Appendix A for the details.

We performed experiments with and without training data augmentation. In case of augmentation, train and test sets were augmented in slightly different fashion. For training, about 50% of the queries were augmented with sim_s and the other 50% were augmented with sim_k. The original data was combined with the augmented data resulting in 43,118 queries in the training set. We evaluated the model on three versions of the test set, one version injected sim_c errors in each query, the second version using sim_s and the third with sim_k.

Results of the combined low-resource and error simulations are summarized in Table 8. Due to the number of experiments, we only reported the macro F1-score. We see a similar trend to the results presented in Section 5.3, the model performance on the PAL test set is consistently lower than MSA test set across all experiments. We also notice that the model is more sensitive to some errors introduced into the test set.

We performed the experiments using two trained models, with and without training augmentation. In both models we see similar behaviour, where we observe that the average drop in performance, when reducing training set size, on PAL-sim_c across all data settings is about 3.38%, compared to 2.37% on MSA-sim_c. Similar pattern is also observed on the PAL-sim_s and MSA-sim_k, with an average performance drop of 3.39% and 2.16%, respectively. However, we see a lower performance on PAL-sim_s with an average drop in F1-score by 4.2%, compared to 2.19% on MSA-sim_s. From that, we learn that the model performance is stable on MSA regardless of the type of errors we inject into the data, however, on PAL we see more volatility and sensitivity in the model performance when injecting sim_s errors. Those findings reveal that BERT is more susceptible to the removal of spaces in dialectical Arabic since that results in combining two or three tokens into one. This issue is exacerbated further in dialectical Arabic since it lacks consistent orthography compared to MSA.

Despite those results, we see that augmenting the training data did help close the performance gap between the PAL and MSA. Figure 2 zooms in a little more into the performance on MSA-sim_s and PAL-sim_s with and without training augmentation. Three observations to make from Figure 2: 1) MSA performance is better than PAL regardless of data augmentation, 2) augmenting the training data closes the performance gap between PAL-sim_s (augmented) and MSA-sim_s (without augmentation), 3) the average F1-score gain after training

	MSA Test			PAL Test		
% of data	Precision	Recall	F1	Precision	Recall	F1
20%	0.8825	0.8755	0.8758	0.8441	0.8403	0.8363
50%	0.9117	0.9094	0.9088	0.8909	0.8903	0.8888
100%	0.9231	0.9212	0.9209	0.9004	0.9025	0.8995

Table 7: Results on the ArBanking77 MSA and PAL test sets in low-resource settings

with augmented data on PAL-sim_s (4.12%) is larger than MSA-sim_s (2.2%). The improvements are less noticeable on sim_c and sim_k.

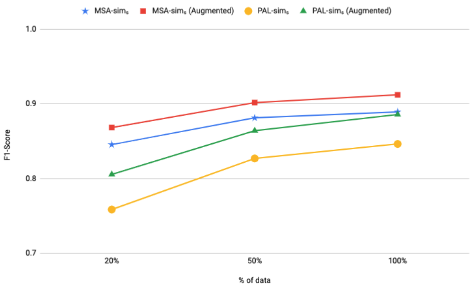


Figure 2: MSA-sim_s vs. PAL-sim_s F1-scores with low-resource settings, (Augmented) indicates that the training data was augmented.

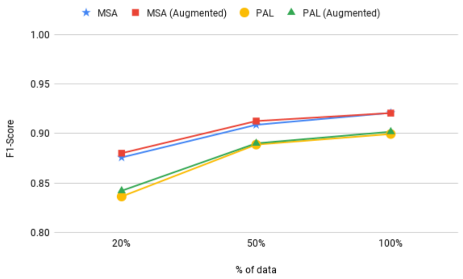


Figure 3: MSA vs. PAL clean sets F1-scores with low-resource settings and data augmentation, (Augmented) indicates that the training data was augmented.

Figure 3 shows that training data augmentation does not affect the performance on the clean MSA and PAL test sets. On the contrary, at the lowest resource settings the augmented model outperformed the non-augmented on MSA and PAL by 0.43% and 0.58%, respectively. At 50% and 100% settings, both the augmented and non-augmented models' performance converge on MSA and PAL.

6 Conclusion

In this paper, we presented the ArBanking77 dataset, consisting of queries in both MSA and Palestinian dialects in the banking domain. As far as we know, ArBanking77 is the first Arabic intent detection dataset in the banking domain. The dataset contains 31,404 queries and 77 intents. The data was then used to fine-tune a BERT-based model for the intent detection task, resulting in an F1-score of 0.9209 for MSA and 0.8995 for PAL. We also simulated low-resource settings and found that the model is robust and with only 20% of the data, model performance on PAL and MSA dropped by only 6.32% and 4.51%, respectively. We noted that training data augmentation does not negatively affect the model performance on the clean MSA and PAL test sets. In fact, at the lowest resource settings (20%) the augmented model out-performed the non-augmented model on both MSA and PAL.

We performed additional data augmentation to simulate errors, misspellings, and other mistakes that are common in real NLP systems. We observed the accuracy on PAL-sim_s suffers greatly when the model is trained on 20% of the non-augmented data. Augmenting the training data closes the performance gap on PAL-sim_s by about 5%. This indicates that BERT is susceptible to some errors, especially in dialectal Arabic which has less consistent orthography than MSA. It is also noticeable that the relative drop in accuracy between the 20% and 50% training sets is much larger than 50% and 100% case. This implies that the negative effect of the introduced errors in the dialectical Arabic is inversely proportional to the amount of data used in the train set. Finally, based on the low performance using zero-shot learning on MSA and PAL and a slight lexical overlap between them, we concluded that there is an urgent need to annotate MSA and dialectical Arabic.

7 Limitations

Our dataset is limited to MSA and Palestinian dialect and covers only 77 intents. Applying our models and data to dialects others than MSA and PAL may not yield accurate intents. Furthermore, our data covers intents that are commonly found in traditional banking. Additional intents may need to be studied from non-traditional banking such as Islamic banks. We plan to extend our dataset to

Train Augmentation	Test Augmentation	MSA Test			PAL Test		
		20%	**50%**	**100%**	**20%**	**50%**	**100%**
None	None	0.8758	0.9088	0.9209	0.8363	0.8888	0.8995
	sim_c	0.8452	0.8795	0.8981	0.7933	0.8435	0.8637
	sim_s	0.8454	0.8813	0.8893	0.7585	0.8269	0.8463
	sim_k	0.8392	0.8648	0.8844	0.7942	0.8428	0.8634
sim_s/sim_k	None	0.8801	0.9126	0.9207	0.8421	0.8901	0.9018
	sim_c	0.8583	0.8922	0.9001	0.8065	0.8602	0.8711
	sim_s	0.8683	0.9017	0.9121	0.8055	0.8641	0.8857
	sim_k	0.8499	0.8833	0.8909	0.8086	0.8529	0.8749

Table 8: Performance in terms of F1-scores of models trained on the combined MSA and PAL datasets when simulating low-resource setting (20% of the data) and different types of noise, "None" refers to the clean dataset while the percentages in the header indicate the percentage of training data used.

cover more Arabic dialects and obtain data from non-traditional banking institutions in the Arab region to better understand the difference in intents compared to the traditional banking. Moreover, we want to explore natural language understanding in the banking domain by combining named entity recognition with intent detection.

We can further improve model performance by adding additional auxiliary loss functions such as contrastive loss, which will help align the token representations between the MSA and PAL queries. Furthermore, due to data limitation, the models trained on the data, including Banking77, perform intent classification using a single utterance. In practice, the query has a context, preceding utterances, that can provide important signal to the model, which may lead to better performance.

Acknowledgements

This research is partially funded by the Palestinian Higher Council for Innovation and Excellence. We would like to thank Taymaa Hammouda for the technical support. The authors also acknowledge the great efforts of many students who helped in the annotation process, especially Rania Shahwan, Dalal Bawatneh, Malak Elsheik, Maissan Qadi, Marah Beirat, Sara Shaabna, Hiba Qasrawee, Manar Jawabreh, Mohammad AbuBader, Tuqa Qurt, Rahaf Bakeer, Marah Qoud, Nirmeen Al-Sheikh, Ameer Eleyan, Sondus Ilawi, Fatima Kusbeh, Ameena Jadallah, Eyab Ghifari, Lina Salameh, Shorouq Zaid, Mariam Abdelqader, Noor Momani, Yazan Assaf, Hala Hamza, Sondus Majdobeh and Sawsan Mohamad.

References

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

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2021. ARBERT & MARBERT: Deep bidirectional transformers for Arabic. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7088–7105, Online. Association for Computational Linguistics.

Eleni Adamopoulou and Lefteris Moussiades. 2020. An overview of chatbot technology. In *Artificial Intelligence Applications and Innovations*, pages 373–383, Cham. Springer International Publishing.

Arfan Ahmed, Nashva Ali, Mahmood Alzubaidi, Wajdi Zaghouani, Alaa Abd-alrazaq, and Mowafa Househ. 2022. Arabic chatbot technologies: A scoping review. *Computer Methods and Programs in Biomedicine Update*, 2:100057.

Moustafa Al-Hajj and Mustafa Jarrar. 2021. Arabglossbert: 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 40–48, Online. INCOMA Ltd.

Ali Saleh Alammary. 2022. Bert models for arabic text classification: A systematic review. *Applied Sciences*, 12(11).

Bushra Algotiml, AbdelRahim Elmadany, and Walid Magdy. 2019. Arabic tweet-act: Speech act recognition for Arabic asynchronous conversations. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 183–191, Florence, Italy. Association for Computational Linguistics.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. AraBERT: Transformer-based model for Arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France. European Language Resource Association.

Giusepppe Attardi. 2015. Wikiextractor. `https://github.com/attardi/wikiextractor`.

Samyadeep Basu, Amr Sharaf, Karine Ip Kiun Chong, Alex Fischer, Vishal Rohra, Michael Amoake, Hazem El-Hammamy, Ehi Nosakhare, Vijay Ramani, and Benjamin Han. 2022. Strategies to improve few-shot learning for intent classification and slot-filling. pages 17–25. Association for Computational Linguistics.

Chiva Olivia Bilah, Teguh Bharata Adji, and Noor Akhmad Setiawan. 2022. Intent detection on indonesian text using convolutional neural network. In *2022 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom)*, pages 174–178.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 38–45, Online. Association for Computational Linguistics.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*.

Kareem Darwish, Nizar Habash, Mourad Abbas, Hend Al-Khalifa, Huseein T. Al-Natsheh, Houda Bouamor, Karim Bouzoubaa, Violetta Cavalli-Sforza, Samhaa R. El-Beltagy, Wassim El-Hajj, Mustafa Jarrar, and Hamdy Mubarak. 2021. A panoramic survey of natural language processing in the arab worlds. *Commun. ACM*, 64(4):72–81.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ibrahim Abu El-Khair. 2016. 1.5 billion words arabic corpus. *arXiv preprint arXiv:1611.04033*.

AbdelRahim Elmadany, Hamdy Mubarak, and Walid Magdy. 2018. Arsas: An arabic speech-act and sentiment corpus of tweets. *OSACT*, 3:20.

Fadhl Eryani, Nizar Habash, Houda Bouamor, and Salam Khalifa. 2020. A spelling correction corpus for multiple arabic dialects. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4130–4138.

Ahlam Fuad and Maha Al-Yahya. 2022. Recent developments in arabic conversational ai: A literature review. *IEEE Access*, 10:23842–23859.

Sana Ghanem, Mustafa Jarrar, Radi Jarrar, and Ibrahim Bounhas. 2023. A benchmark and scoring algorithm for enriching arabic synonyms. In *Proceedings of the 12th International Global Wordnet Conference (GWC2023)*, pages 215–222. Global Wordnet Association.

Karim El Haff, Mustafa Jarrar, Tymaa Hammouda, and Fadi Zaraket. 2022. Curras + baladi: Towards a levantine corpus. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2022)*, Marseille, France.

Charles T. Hemphill, John J. Godfrey, and George R. Doddington. 1990. The ATIS spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27,1990*.

Mohammad Hijjawi, Zuhair Bandar, and Keeley Crockett. 2013. User's utterance classification using machine learning for arabic conversational agents. In *2013 5th International Conference on Computer Science and Information Technology*, pages 223–232.

Mohammad Hijjawi, Zuhair Bandar, Keeley Crockett, and David Mclean. 2014. Arabchat: An arabic conversational agent. In *2014 6th International Conference on Computer Science and Information Technology (CSIT)*, pages 227–237.

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.

Mustafa Jarrar. 2021. The arabic ontology - an arabic wordnet with ontologically clean content. *Applied Ontology Journal*, 16(1):1–26.

Mustafa Jarrar, Nizar Habash, Diyam Akra, and Nasser Zalmout. 2014. Building a corpus for palestinian arabic: a preliminary study. In *Proceedings of the EMNLP 2014, Workshop on Arabic Natural Language*, pages 18–27. Association For Computational Linguistics.

Mustafa Jarrar, Nizar Habash, Faeq Alrimawi, Diyam Akra, and Nasser Zalmout. 2017. Curras: An annotated corpus for the palestinian arabic dialect. *Journal Language Resources and Evaluation*, 51(3):745–775.

Mustafa Jarrar, Eman Karajah, Muhammad Khalifa, and Khaled Shaalan. 2021. Extracting synonyms from bilingual dictionaries. In *Proceedings of the 11th International Global Wordnet Conference (GWC2021)*, pages 215–222. Global Wordnet Association.

Mustafa Jarrar, Mohammed Khalilia, and Sana Ghanem. 2022. Wojood: Nested arabic named entity corpus and recognition using bert. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2022)*, Marseille, France.

Mustafa Jarrar, Sanad Malaysha, Tymaa Hammouda, and Mohammad Khalilia. 2023a. Salma: Arabic sense-annotated corpus and wsd benchmarks. In *Proceedings of the 1st Arabic Natural Language Processing Conference (ArabicNLP), Part of the EMNLP 2023*. ACL.

Mustafa Jarrar, Fadi Zaraket, Tymaa Hammouda, Daanish Masood Alavi, and Martin Waahlisch. 2023b. Lisan: Yemeni, irqi, libyan, and sudanese arabic dialect copora with morphological annotations. In *The 20th IEEE/ACS International Conference on Computer Systems and Applications (AICCSA)*. IEEE.

Alaa Joukhadar, Huda Saghergy, Leen Kweider, and Nada Ghneim. 2019. Arabic dialogue act recognition for textual chatbot systems. In *Proceedings of The First International Workshop on NLP Solutions for Under Resourced Languages (NSURL 2019) co-located with ICNLSP 2019-Short Papers*, pages 43–49.

Kathrein Abu Kwaik, Motaz Saad, Stergios Chatzikyriakidis, and Simon Dobnik. 2018. A lexical distance study of arabic dialects. *Procedia Computer Science*, 142:2–13. Arabic Computational Linguistics.

Wuwei Lan, Yang Chen, Wei Xu, and Alan Ritter. 2020. Gigabert: Zero-shot transfer learning from english to arabic. In *Proceedings of The 2020 Conference on Empirical Methods on Natural Language Processing (EMNLP)*.

Haneen Liqreina, Mustafa Jarrar, Mohammed Khalilia, Ahmed Oumar El-Shangiti, and Muhammad Abdul-Mageed. 2023. Arabic fine-grained entity recognition. In *Proceedings of the 1st Arabic Natural Language Processing Conference (ArabicNLP), Part of the EMNLP 2023*. ACL.

Ridha Mezzi, Aymen Yahyaoui, Mohamed Wassim Krir, Wadii Boulila, and Anis Koubaa. 2022. Mental health intent recognition for arabic-speaking patients using the mini international neuropsychiatric interview (mini) and bert model. *Sensors*, 22(3).

Eman Naser-Karajah, Nabil Arman, and Mustafa Jarrar. 2021. Current trends and approaches in synonyms extraction: Potential adaptation to arabic. In *Proceedings of the 2021 International Conference on Information Technology (ICIT)*, pages 428–434, Amman, Jordan. IEEE.

Amal Nayouf, Mustafa Jarrar, Fadi zaraket, Tymaa Hammouda, and Mohamad-Bassam Kurdy. 2023. Nâbra: Syrian arabic dialects with morphological annotations. In *Proceedings of the 1st Arabic Natural Language Processing Conference (ArabicNLP), Part of the EMNLP 2023*. ACL.

Lorenz Nigst, Maxim Romanov, Sarah Bowen Savant, Masoumeh Seydi, and Peter Verkinderen. 2020. Openiti: a machine-readable corpus of islamicate texts. *nd http://doi. org/10.5281/zenodo*, 4075046.

Pedro Javier Ortiz Suárez, Benoît Sagot, and Laurent Romary. 2019. Asynchronous Pipeline for Processing Huge Corpora on Medium to Low Resource Infrastructures. In *7th Workshop on the Challenges in the Management of Large Corpora (CMLC-7)*, Cardiff, United Kingdom. Leibniz-Institut für Deutsche Sprache.

Robert Parker, David Graff, Ke Chen, Junbo Kong, and Kazuaki Maeda. 2011. Arabic gigaword fifth edition ldc2011t11. *Philadelphia: Linguistic Data Consortium*.

Sana Shams and Muhammad Aslam. 2022. Improving user intent detection in urdu web queries with capsule net architectures. *Applied Sciences*, 12(22).

Sana Shams, Muhammad Aslam, and Ana Maria Martinez-Enriquez. 2019. Lexical intent recognition in urdu queries using deep neural networks. In *Advances in Soft Computing*, pages 39–50, Cham. Springer International Publishing.

Imad Zeroual, Dirk Goldhahn, Thomas Eckart, and Abdelhak Lakhouaja. 2019. OSIAN: Open source international Arabic news corpus - preparation and integration into the CLARIN-infrastructure. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 175–182, Florence, Italy. Association for Computational Linguistics.

Yunhua Zhou, Peiju Liu, and Xipeng Qiu. 2022. Knn-contrastive learning for out-of-domain intent classification. pages 5129–5141. Association for Computational Linguistics.

A Error Simulation Types

A.1 Common Errors (sim_c)

sim_c are common spelling errors and word variations that people often make in real-life, which we derive from a lexicon. In a previous work, we developed a lexicon that contains a list of base forms, and the lexical variants (mostly colloquial terms) of each base form. The lexicon curation process started by collecting data from social media sites, chatbots and call centers audio recordings, which were transcribed manually. For each lexical variant, colloquial term and misspelling, the goal was to find its corresponding base form. Hence, a base form in the lexicon can have more than one lexical variant. The lexicon contains 12,111 base forms. To simulate these errors in our intent detection task, for each query, we randomly selected one to two words that have a matching base form in the lexicon, and for each base form we randomly selected one of its lexical variants. Because these errors are not simulated and are mostly colloquial variants collected from real content, we injected this type of error into the test set only, which will give us an insight how robust the model's performance is on such noisy data. Examples of orthographic variants are shown in Table 9. For instance, the world شكرا/thanks has four variants (ششكرا, شككرا, شكراااااا, and شكررا).

Lexical Term	Lexical Variants
شكرا	شكرااااااا ششكرا شكررا شككرا
مثلا	مثلاث مثل

Table 9: Sample of lexicon, some words are colloquial while others are misspellings.

A.2 Simulated Errors (*sim*$_s$)

sim$_s$ are errors simulated by deleting spaces between words. We applied this type of simulation on both the train and test sets. For each query we randomly deleted one or two spaces.

A.3 Keyboard Errors (*sim*$_k$)

sim$_k$ are errors generated by inserting or deleting a letter from a word, replacing a letter with another letter, or swapping the places of two adjacent letters. Two approaches we followed when simulating this error. Either random replacement or replacement guided by the keyboard layout of the target language. Keyboard layout guided simulation will delete/insert/replace/swap letters based on the neighboring letters on the keyboard.

ArabIcros: AI-Powered Arabic Crossword Puzzle Generation for Educational Applications

Kamyar Zeinalipour and **Mohamed Zaky Saad** and **Marco Maggini** and **Marco Gori**
DIISM, University of Siena
Via Roma 56, Siena, Italy
{kamyar.zeinalipour2, marco.maggini, marco.gori}@unisi.it
m.zakyanwarzakymo@student.unisi.it

Abstract

This paper presents the first Arabic crossword puzzle generator driven by advanced AI technology. Leveraging cutting-edge large language models including GPT4, GPT3-Davinci, GPT3-Curie, GPT3-Babbage, GPT3-Ada, and BERT, the system generates distinctive and challenging clues. Based on a dataset comprising over 50,000 clue-answer pairs, the generator employs fine-tuning, few/zero-shot learning strategies, and rigorous quality-checking protocols to enforce the generation of high-quality clue-answer pairs. Importantly, educational crosswords contribute to enhancing memory, expanding vocabulary, and promoting problem-solving skills, thereby augmenting the learning experience through a fun and engaging approach, reshaping the landscape of traditional learning methods. The overall system can be exploited as a powerful educational tool that amalgamates AI and innovative learning techniques, heralding a transformative era for Arabic crossword puzzles and the intersection of technology and education.

1 Introduction

Combining traditional puzzle constructs with educational components, pedagogical crosswords foster interactive learning experiences by integrating vocabulary, history, sciences, and other subjects. Intriguingly, they effectively strengthen students' vocabulary and spelling abilities due to the puzzles' requirement for accurate spelling (Orawiwatnakul, 2013; Dzulfikri, 2016; Bella and Rahayu, 2023). These puzzles are particularly significant for language acquisition and learning specific technical terms (Nickerson, 1977; Sandiuc and Balagiu, 2020; Yuriev et al., 2016). Moreover, they enhance problem-solving, critical thinking skills, and memory retention, thereby making the learning process enjoyable and productive (Kaynak et al., 2023; Dol, 2017; Mueller and Veinott, 2018; Dzulfikri, 2016; Zirawaga et al., 2017; Bella and Rahayu, 2023; Zamani et al., 2021; Yuriev et al., 2016).

Creating Arabic educational crosswords can be challenging due to the required wordplay expertise. However, with the help of innovations in natural language processing, Large Language Models (LLMs) are now able to generate high-quality Arabic crossword clues. LLMs are pre-trained on a mix of sources like books, academic articles, and web content and this wide spectrum of content enables them to create challenging and engaging crossword clues. This aids puzzle designers and improves the solver's experience, enabling even beginners to design personalized puzzles.

The results show that the proposed approach can be effectively employed to generate Arabic educational crossword puzzles, introducing an innovative system using LLMs to generate top-quality clues and answers. By inputting text passages or keywords, the system generates clue-answer pairs, based on techniques like fine-tuning and few-shot learning used for generation. We also present models to filter inappropriate clue-answer pairs for puzzle construction optimization, propose an advanced algorithm for designing Arabic educational crossword layouts, and provide a comprehensive dataset of curated Arabic clue-answer pairs. These advances simplify the creation of Arabic pedagogical crosswords and expand their potential for their broader exploitation.

This paper is structured as follows; section 2 explores relevant literature; section 3 discusses the collected Arabic dataset; section 4 outlines the research methodology, section 5 presents the findings, and, finally, section 6 summarizes the overall outcomes.

2 Related works

The generation of crosswords represents a complex task that has been addressed by some research works. These studies have utilized a variety of tools, including traditional dictionaries and thesauri, or

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 288–301
December 7, 2023 ©2023 Association for Computational Linguistics

have engaged in the linguistic analysis of text content derived from the web.

Rigutini et al. (Rigutini et al., 2008, 2012) pioneered the first fully automated crossword creator system in 2008. The proposed system leverages natural language processing techniques to generate crossword clues by scraping related documents from the web, extracting relevant text segments, and using part-of-speech tagging, dependency parsing, and WordNet-based similarity measures. This approach produces clues based on specific ranking criteria.

An alternative methodology for crossword construction using natural language processing is documented in (Ranaivo-Malançon et al., 2013). This approach consists of a four-stage process, which includes initial data retrieval of a targeted topic-specific text compilation, extraction of complete sentences, determination of the dependency syntactic structure of each sentence, and removal of words from stop-lists. The extracted information undergoes a transformation into a graph representation for depth-first pre-order search. This framework integrates pre-processing, candidate identification, clue formation, and answer selection.

Esteche et al.'s study (Esteche et al., 2017) delved into the creation of Spanish language crossword puzzles from news articles. The system is based on a twofold procedure: initially pivotal terms are identified and their meanings are isolated from a trusted online dictionary. Subsequently, these definitions are employed as hints for the assembly of compelling crossword puzzles.

In a related study, Arora et al. (Arora and Kumar, 2019) discuss a software tool that uses NLP techniques to identify crucial keywords for creating crossword puzzles in various Indian languages. Their proposed framework, SEEKH, combines statistical and linguistic methods to highlight significant keywords useful for crossword creation.

Despite significant research, accurately generating comprehensive and unique clue-answer sets from linguistic corpora remains a challenge, particularly for the unique linguistic nuances of Arabic. To address these issues, we propose an innovative methodology using LLMs to create intricate educational clues. As a pioneering attempt, our technique successfully generates Arabic crossword puzzles, filling a gap unaddressed by previous methods. By generating intellectually stimulating and original crossword puzzles, this novel approach enhances learners' deep understanding of the subjects by providing comprehensive answers. Hence, the proposed work not only brings novelty to Arabic crossword generation, but also offers a groundbreaking solution in the realm of educational tools.

3 Dataset

Given the scarcity of data for Arabic crossword puzzles, a clue-answer pair dataset was gathered manually. The dataset encompasses the period from 2020 to 2023.

During the initial stage of data collection, we pursued all accessible crossword puzzles, encompassing web-based games, journals, and magazines, ensuring that the training set comprised accurate clue-answer pairs sourced from original Arabic crossword puzzles. We had a collection of crossword images, and we needed to extract the text contained within these images to build a dataset for obtaining the text from these images. To accomplish this, we initially utilized optical character recognition (OCR) as a tool. However, it's important to note that the OCR process was predominantly supervised by humans who used it to facilitate the extraction. Additionally, human validation was employed to evaluate both spelling errors within the journals and the overall quality of the clue-answer pairs. This meticulous process resulted in a catalog of 57,706 entries from two different sources. One of them was the Al-Joumhouria Journal, from which we manually extracted 5,661 Clue and Answer pairs. The other source was the Al-Ghad Electronic Journal, where we utilized the OCR tool to assist in the extraction process. In the end, this yielded 25,908 unique pairs with answers varying in length from 1 to 21 characters, with the majority of the data falling within a specific answer's character length range from 2 to 9 (see Fig. 1).

The structure of the pairs is recurrent. For instance, some of the pairs are synonyms or antonym definitions, that define the answer by means of one or more synonyms or antonyms. An example of this category includes "موتي" with the answer "حتفي". Some others were general information, such as for example "دولة عربية" with the answer "مصر". Another structure can be a word but the letters are not in order, as for example "جميل مبعثرة" with the answer "ل م ي ج". Finally, the definition can give the word and requires part of it for the answer, as for instance "نصف نادر" with the answer "در".

A meticulous pre-processing step was carried out on the data to refine it for fine-tuning. This involved the elimination of Arabic accents, redundant pairs, and markers suggesting a reversal in crosswords—an idiosyncrasy of Arabic. The aim of this study was to pave the way for further research by making this processed dataset publicly accessible, encouraging other scholars to contribute to this field. [1]

4 Methodology

The proposed system includes several components, such as mechanisms to generate clue-answer pairs using user-provided text or keywords, and a crossword schema generator as depicted in Figure 2. Users can input any instructional text to extract relevant clue-answer pairs or insert a list of chosen keywords to generate clues. After combining both clue-generation methods, the quality of the generated pairs is evaluated using specific validation modules. Users can then review and select their preferred clue-answer sets, which are employed in the final step by a separate module for creating the crossword layout.

4.1 Path (a): Generating clue-answer pairs from input text

In our system, we employ zero-shot and few-shot learning to create clue-answer pairs. This process involves segmenting the text into paragraphs, keyword extraction, generating potential clues, and rigorously validating the resulting pairs. More details on these stages are provided later in the paper. Our experiments are based on the models GPT3.5-Turbo and GPT4 (Brown et al., 2020) (OpenAI, 2023). We use dynamic experimental approaches, including both customized English and Arabic prompts, to assess prompt language strategies' effectiveness across models.

4.1.1 Keyword extraction

Our Few-Shot Learning Framework begins with prompt construction, involving the incorporation of extensive educational text that includes potential crossword keywords. These keywords, chosen to match possible answers from the provided text, enhance precision as the LLM is prompted with well-curated information. The process concludes by inputting the educational text and the tailored prompt to the LLM, enabling it to utilize its few-shot learning experiences to extract potential keywords from the input paragraph. This mechanism allows the LLM to extrapolate potential keywords effectively, resulting in a more comprehensive analysis.

4.1.2 Generating crossword clues from the extracted keywords

In this stage, we harness the power of few-shot learning once more. By utilizing the keywords identified in the previous phase along with the input text, we generate relevant crossword clues. Additional information, including an example of valid paragraph, keywords, and clues, was also input into the LLM along with the target text and previously generated keywords that needed crossword clues. This strategy enabled the LLM to craft unique clues by leveraging the supplied text and initial keywords. This systematic approach significantly improves the precision and relevance of the generated crossword clues, ensuring each clue aligns with the context of the provided text and identified keywords.

4.1.3 Path(a) Validation

To enhance the quality and appropriateness of our generated keyword-clue pairs, a method to exclude low-quality and inappropriate pairings is applied in several discrete stages. The first step utilized a filter system to eliminate answers containing more than three words, which are typically unsuitable for crossword puzzles. Our empirical research has shown that the LLM can occasionally produce clues by drawing upon its innate knowledge rather than relying solely on the provided text. Additionally, in instances where the generated clues did not effectively capture relevant keywords, we took steps to address this issue. To enhance the quality of our output and ensure the creation of appropriate clue-answer pairs, we employed a zero-shot learning approach, effectively filtering out undesired clues.

4.2 Path (b): Generating clues based on provided answers

There may be scenarios where we need to generate crossword clues using provided answers without a full-text context. To face this task, we deployed a holistic approach that started with fine-tuning different language models using the introduced Section 3, each specifically designed for this task. We further enriched this scheme by using data from these fine-tuned models to create various classi-

[1] The dataset is available at https://huggingface.co/datasets/Kamyar-zeinalipour/AR_CW

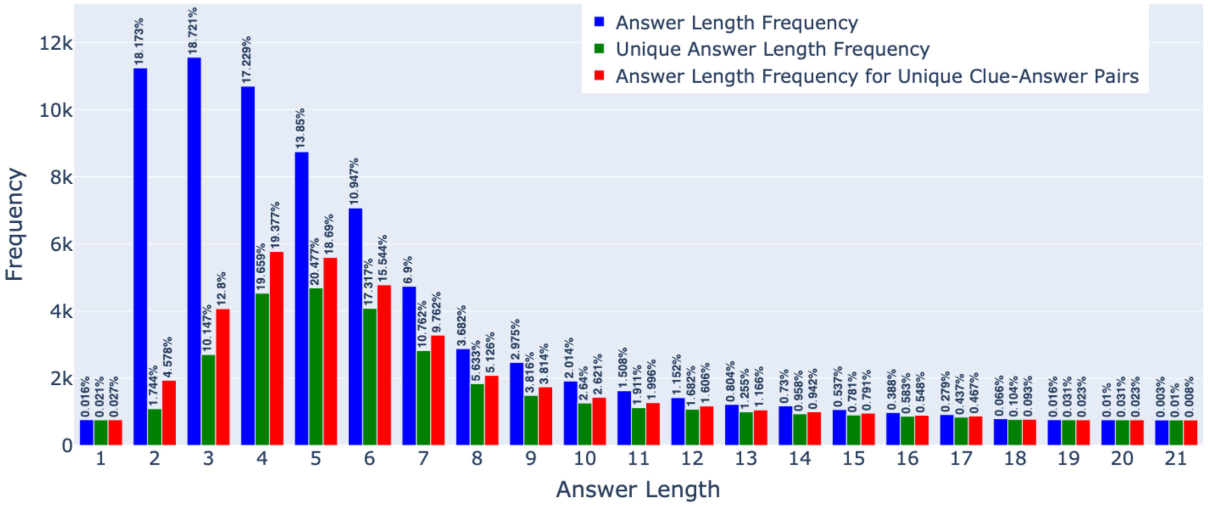


Figure 1: The introduced dataset entries are visually presented in terms of answer length distribution. The blue bars represent all the clue-answer pairs, while the green bars depict the frequency of unique answers. Additionally, the red bars indicate the frequency of unique answer-clue pairs.

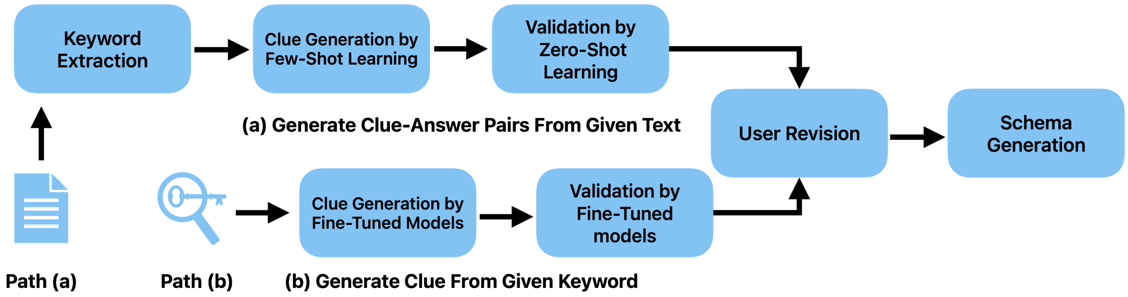

Keyword Extraction
Clue Generation by Few-Shot Learning
Validation by Zero-Shot Learning
(a) Generate Clue-Answer Pairs From Given Text
User Revision
Schema Generation
Clue Generation by Fine-Tuned Models
Validation by Fine-Tuned models
Path (a)
Path (b)
(b) Generate Clue From Given Keyword

Figure 2: Overall system architecture. Path (a) Clue-answer generation from input text. Path (b) Clue generation from the given answers.

fiers. These classifiers aim to differentiate between high-quality generated clue-answer pairs and less suitable alternatives.

4.2.1 Fine-tuning LLMs to generate clues from provided answers

In the pursuit of crafting crossword clues from given answers and textual information, our research delved into the optimization of language models. This refinement process was informed by the dataset meticulously outlined in Section 3. Our evaluation encompassed a spectrum of models, notably the robust Lamma2 13B and the efficient Llama2 7B, distinguished by their substantial 13 billion and 7 billion parameters, respectively. We also examined the 1.5 billion-parameter GPT2-XL model, recognized for its versatility, and the T5 Base model, endowed with 350 million parameters as expounded in (Brown et al., 2020).

This section encapsulates our methodical approach to model selection, emphasizing the diversity of parameters and architectures considered in our quest to enhance the generation of crossword clues. The subsequent analysis and results, detailed in the following sections, shed light on the efficacy and performance of these fine-tuned language models in the context of crossword clue generation.

4.2.2 Path(b) Validation

The design of the overall system focuses on enhancing the overall quality of the generated clue-answer pairs. We incorporated a filtering process into the system pipeline to enhance the quality and usability of the generated pairs. Using the data obtained from the fine-tuned language models, we created a classifier capable of distinguishing between effective and unsuitable clues.

For this purpose, several models were fine-tuned, including GPT3-DaVinci with 175 billion parameters, GPT3-Curie with 13 billion parameters, GPT3-

Babbage with 1.3 billion parameters, GPT3-Ada with 350 million parameters (Brown et al., 2020), and BERT-base-Arabic with 110 million parameters (Raffel et al., 2020; Safaya et al., 2020). These models provided important insights into their respective capabilities and aided in validating the generated clues.

Our primary objective was to use these models with their varying parameter counts to comprehensively evaluate their effectiveness in filtering and validating the generated clues. This methodology aimed to ensure only high-quality and contextually relevant clues were retained, thereby improving the overall precision and functionality of our system.

4.3 Schema Generator

The algorithm for creating educational crossword puzzles follows a streamlined approach using input parameters such as the answer list, workspace dimensions, and termination criteria. Initially, a central answer is placed randomly followed by strategically adding surrounding answers. This cycle of adding and occasionally removing the recently added answers or entirely resetting is repeated until an optimal solution is obtained. The quality of the crossword is evaluated through a comprehensive scoring process. Each solution's merit is determined by the following scoring formula:

$$\mathrm{Score} = (\mathrm{FW} + 0.5 \cdot \mathrm{LL}) \cdot \mathrm{FR} \cdot \mathrm{LR} \qquad (1)$$

The variables exploited in this formula correspond to the following metrics:

- Filled Words (FW): This represents the count of the added words, signaling the puzzle's completeness.
- Linked Letters (LL): This counts the instances of letter-sharing between intersecting words, indicating the puzzle's coherence.
- Filled Ratio (FR): This metric, calculated as the filled letters count divided by the area of the smallest covering rectangle, showcases the efficiency of the crossword's space utilization.
- Linked Letters Ratio (LR): By dividing LL by the total letter count, LR highlights the extent of letter linkage and word-relations within the puzzle.

These four criteria collectively contribute to the evaluation and selection of the optimal solution during the algorithm execution.

The algorithm makes use of a variety of stopping criteria to guide its decision-making and determine when to end the crossword construction. These criteria are as follows:

- Minimum Number of Answers: The algorithm stops once it has added a preset minimum count of answers to the grid, ensuring an adequate crossword complexity.
- Minimum Filled Ratio Threshold: A certain threshold of the filled ratio, when met or surpassed, triggers the algorithm to stop, preventing the overabundance of empty spaces and maintaining appealing aesthetics.
- Grid Rebuilding Limit: The algorithm ceases to operate if the grid's reconstruction exceeds a set count, avoiding getting stuck in inefficient solutions and encouraging exploration of other possibilities.
- Maximum Time Duration: Upon reaching the allowed maximum time duration, the algorithm finishes, ensuring the process is time-efficient and the resources are optimally utilized.

This method allows the algorithm to identify the highest-scoring solution, enabling efficient production of high-quality crosswords given its input parameters. Furthermore, the algorithm can prioritize a list of "preferred answers," increasing their chances of inclusion, thereby ensuring that the crossword design aligns with specific objectives or preferences.

5 Experiments

In this section, we detail the empirical evaluation of the proposed system, focusing on individual elements and their roles within the overall framework.

5.1 Experimental Evaluation: Path (a)

This paper's experimental dataset aims to rigorously assess our system's output quality in relation to various language prompts. We conducted an in-depth investigation using two prompt types, categorized as English and Arabic. Two different models, GPT4 and GPT3.5 Turbo, were used for evaluation. The comprehensive list of prompts can be found within the paper's Appendix B. This provides comprehensive evaluations of linguistic aspects, leading to robust, multifaceted findings. The system underwent thorough evaluation using 100 educational selected Wikipedia paragraphs to examine performance in different language contexts. Performance markers were established based on empirical evidence. Evaluation guidelines, created under expert supervision, ensured robust results. Detailed criteria for evaluation are in Appendix A, and cumulative findings are presented in Table 1. GPT4 and GPT3.5-Turbo models performed impressively in English prompts, achieving keyword extraction accuracy of 95.05% and 92% respectively. They similarly excelled in Arabic prompts with accuracies of 94.32% and 97.38%.

In clue generation, these models demonstrated their value in retrieving meaningful information. In English prompts, GPT4 and GPT3.5-Turbo reached accuracies of 94.62% and 55.33%, respectively, while GPT4 and GPT3. marking respective accuracies of 93.23% and 37.78% in Arabic prompts.

The evaluation of clue-answer pairs yielded satisfactory results. In English, the GPT4 and GPT3.5-Turbo models exhibited accuracies of 87.76% and 89.04% and maintained substantial accuracy of 84.01% and 89.32% in Arabic prompts.

In the final evaluation, which included system-wide validation and acceptability of potentially generated clues and answers, both models upheld their performance. it means we analyze the clue-answer pairs that align with the validation part of the system, and then culminate in the calculation of the proportion of generated clues and answers that successfully pass the criteria established through human oversight which is the total performance of the model. It was overall 78.95% and 74.6% for the GPT4 model for English and Arabic prompts, respectively, while the GPT3.5-Turbo model had a total performance of 46.68% and 68.83% for English and Arabic prompts respectively.

Figure 3 provides a practical illustration of this system component's functionality. It sequentially depicts the transformation from initial text to final crossword clue-answer pairs, demonstrating input paragraphs (a), keyword extraction (b), clue generation (c), and clue-answer pair validation (d). This visual representation clarified the system's operational process, elucidating its capability to turn text into precise crossword clues and their corresponding answers. Comprehensive translations for the content depicted in Figure 3 can be found in the paper's Appendix C.

5.2 Experimental Evaluation: Path (b)

This section details experimental tests on clue generation and validation from keywords using three distinct models, GPT3-DaVinci, GPT3.5-Turbo, and GPT3-Curie. These were designed and optimized based on concepts discussed in Section 4.2.1, with a specific emphasis on forming clues from identified keywords.

In the preparation phase, a subset of the dataset discussed in Section 3, specifically 25,908 unique clue-answer pairs, was selected. Afterwards, each refined model produced 2,000 clues which were evaluated using human judgement based on the criteria presented in Appendix A.

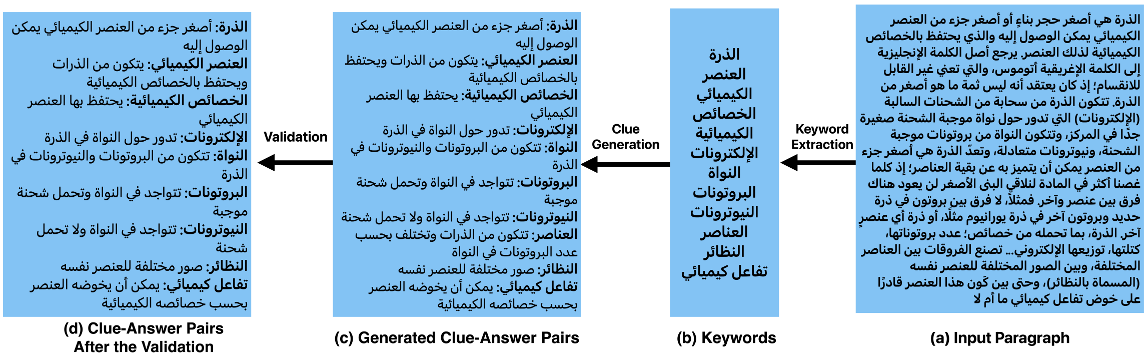
In conclusion of our evaluation, Table 2 presents the results, highlighting the performances of GPT3-DaVinci, GPT3.5-Turbo, and GPT3-Curie. These models successfully generated satisfactory clues 41.9%, 81%, and 21.35% of the time, respectively. Observations indicate that GPT3.5-Turbo significantly outperforms the other models in the task of clue generation from the given keywords. For a thorough assessment of the generated clues, a detailed review identifying acceptable and unacceptable cases was undertaken. Each clue-answer pair was carefully examined and categorized, Tables 3 and 4 present illustrative clues generated by distinct fine-tuned models. Table 3 demonstrates instances of well-constructed clues, while Table 4 highlights cases of unacceptable clue generation. Detailed translations for these clues can be located in the Appendix C. This meticulous evaluation facilitated performance analysis of the algorithm, notably its ability to generate captivating crossword puzzles.

Several classifiers were developed in this study. Coupled with various language models, they enabled the distinction between suitable and unsuitable clue-answer pairings. The results from the evaluation of the test set are shown in Table 5.

The process utilized a dataset of 6,000 human-evaluated instances from previous steps to build several classifiers. The dataset was divided, with 80% used for training, and the remaining 20% for testing classifier performance. The analysis revealed that the dataset consists of 52% acceptable clues and 48% unacceptable ones. The system's effectiveness was gauged through the accuracy of four distinct classifiers - GPT3-DaVinci, GPT3-Curie, GPT3-Babbage, GPT3-Ada, and Bert in discerning between satisfactory and unsatisfactory clues. Notably, GPT3-DaVinci topped the list

Table 1: Assessment outcomes of the clue-answer pairs generated from the provided Text.

Model	*System Part*	*English Prompt*	*Arabic Prompt*
GPT4	Keyword Extractor	95.05 %	94.32%
	Clues Generator	94.62 %	93.23 %
	Validator	87.76 %	84.01 %
	Total performance	78.95 %	74.6 %
GPT3.5-Turbo	Keyword Extractor	92 %	97.38%
	Clues Generator	55.33 %	37.78 %
	Validator	89.04 %	89.32 %
	Total performance	46.68 %	68.83 %


الذرة هي أصغر حجر بناءٍ أو أصغر جزء من العنصر
الكيميائي يمكن الوصول إليه والذي يحتفظ بالخصائص
الكيميائية لذلك العنصر. يرجع أصل الكلمة الإنجليزية
إلى الكلمة الإغريقية أتوموس، والتي تعني غير القابل
للانقسام؛ إذ كان يعتقد أنه ليس ثمة ما هو أصغر من
الذرة. تتكون الذرة من سحابة من الشحنات السالبة
(الإلكترونات) التي تدور حول نواة موجبة الشحنة صغيرة
جدًا في المركز، وتتكون النواة من بروتونات موجبة
الشحنة، ونيوترونات متعادلة، وتعدّ الذرة هي أصغر جزء
من العنصر يمكن أن يتميز به عن بقية العناصر؛ إذ كلما
غصنا أكثر في المادة لنلاقي البنى الأصغر لن يعود هناك
فرق بين عنصر وآخر. فمثلاً، لا فرق بين بروتون في ذرة
حديد وبروتون آخر في ذرة يورانيوم مثلًا، أو ذرة أي عنصرٍ
آخر. الذرة، بما تحمله من خصائص؛ عدد بروتوناتها،
كتلتها، توزيعها الإلكتروني... تصنع الفروقات بين العناصر
المختلفة، وبين الصور المختلفة للعنصر نفسه
(المسماة بالنظائر)، وحتى بين كَون هذا العنصر قادرًا
على خوض تفاعل كيميائي ما أم لا
Keyword
Extraction
الذرة
العنصر
الكيميائي
الخصائص
الكيميائية
الإلكترونات
النواة
البروتونات
النيوترونات
العناصر
النظائر
تفاعل كيميائي
Clue
Generation
الذرة: أصغر جزء من العنصر الكيميائي يمكن الوصول إليه
العنصر الكيميائي: يتكون من الذرات ويحتفظ بالخصائص الكيميائية
الخصائص الكيميائية: يحتفظ بها العنصر الكيميائي
الإلكترونات: تدور حول النواة في الذرة
النواة: تتكون من البروتونات والنيوترونات في الذرة
البروتونات: تتواجد في النواة وتحمل شحنة موجبة
النيوترونات: تتواجد في النواة ولا تحمل شحنة
العناصر: تتكون من الذرات وتختلف بحسب عدد البروتونات في النواة
النظائر: صور مختلفة للعنصر نفسه
تفاعل كيميائي: يمكن أن يخوضه العنصر بحسب خصائصه الكيميائية
Validation
الذرة: أصغر جزء من العنصر الكيميائي يمكن الوصول إليه
العنصر الكيميائي: يتكون من الذرات ويحتفظ بالخصائص الكيميائية
الخصائص الكيميائية: يحتفظ بها العنصر الكيميائي
الإلكترونات: تدور حول النواة في الذرة
النواة: تتكون من البروتونات والنيوترونات في الذرة
البروتونات: تتواجد في النواة وتحمل شحنة موجبة
النيوترونات: تتواجد في النواة ولا تحمل شحنة
النظائر: صور مختلفة للعنصر نفسه
تفاعل كيميائي: يمكن أن يخوضه العنصر بحسب خصائصه الكيميائية

(a) Input Paragraph

(b) Keywords

(c) Generated Clue-Answer Pairs

(d) Clue-Answer Pairs After the Validation

Figure 3: A comprehensive collection of clue-answer pairs generated by the introduced system from a given text, providing illustrative examples.

Table 2: Assessment outcomes of the clues generated from the provided keyword.

Model	*% of acceptable clues*
GPT3-DaVinci	41.9
GPT3-Curie	21.35
GPT3.5-Turbo	81

Table 3: Acceptable clues from given keywords using various models.

Model	*Clue-Answer pair*
GPT3-DaVinci	نجوم : في السماء ليلا
GPT3-Curie	كروم : من المعادن
GPT3.5-Turbo	قوة : قدرة

with an exceptional 85.74% accuracy, followed by GPT3-Curie at 81.29%. GPT3-Babbage showed decent results with 78.69% accuracy, while GPT3-Ada and Bert had fair performances with 79.19% and 71.42% accuracy, respectively. These results underscore the commendable performance of these classifiers in identifying agreeable clues.

Table 4: Unacceptable clues from given keywords using various models.

Model	*Clue-Answer pair*
GPT3-DaVinci	زرافة : من الحشرات
GPT3-Curie	مثلث : مثنى مثلث
GPT3.5-Turbo	عمة : اخت والد او والدة

5.3 Schema Generation

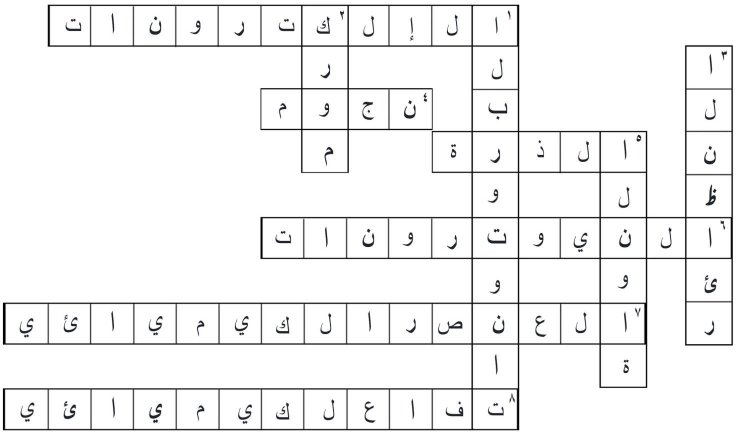
Our algorithm for schema generation envisages a spectrum of educational crosswords utilizing a group of generated clue-answer pairs. Illustrated in Figure 4 is a comprehensive Arabic educational crossword about physics, crafted by the proposed system. The clue-answer pairs are procured either from a text (path (a), refer to Figure 3) or directly produced from a keyword (path (b), denoted by examples marked with a ⋆), as observed in Table 3.

6 Conclusions

The work featured in this paper focuses on multiple innovative offerings, among which is the introduction of a comprehensive dataset for Arabic

Table 5: Classifier performance on distinguishing acceptable Clue-Answer pairs

Model	*accuracy %*	*precision %*	*recall %*	*F1 Score*
GPT3-Dvinci	85.74	83.39	85.26	0.8431
GPT3-Curie	81.29	78.86	79.89	0.7937
GPT3-Babbage	78.69	75.17	78.54	0.7682
GPT3-Ada	79.19	77.48	75.75	0.7660
Bert-base-Arabic	71.42	67.91	70.04	0.6896



الكلمات الأفقيه

١. تدور حول النواة في الذرة (١١)

٤. في السماء ليلا (٤) *

٥. أصغر جزء من العنصر الكيميائي يمكن الوصول إليه (٥)

٦. تتواجد في النواة ولا تحمل شحنة (١١)

٧. يتكون من الذرات ويحتفظ بالخصائص الكيميائية (١٥)

٨. يمكن أن يخوضه العنصر بحسب خصائصه الكيميائية (١٢)

الكلمات الرأسيه(العمودية)

١. تتواجد في النواة وتحمل شحنة موجبة (١٠)

٢. من المعادن (٤) *

٣. صور مختلفة للعنصر نفسه (٧)

٥. تتكون من البروتونات والنيوترونات في الذرة (٦)

Figure 4: An illustrative Arabic educational crossword generated through the proposed system.

clue-answer pairs. In addition to this, we have also formulated a ground-breaking method employing large language models that generate educational Arabic crossword puzzles influenced by the provided texts or given keywords.

To uphold stringent quality standards in our methodology, our approach integrates human oversight in conjunction with specific guidelines (see Appendix A). In the process of generating clue-answer pairs from textual data, we conducted experiments using two distinct models: GPT-4 and GPT3.5-Turbo, while employing prompts in both English and Arabic languages. We conducted various types of evaluations considering different parts of the system and overall performance:

- Keyword Extraction: Notably, when paired with Arabic prompts, GPT3.5-Turbo exhibited exceptional performance, successfully generating high-quality keywords with an impressive accuracy rate of 97.38%.
- Crossword Clue Generation: GPT4, when prompted in English, consistently produced relevant and well-suited crossword clues, achieving a commendable success rate of 94.62%.
- Validation Component: Within our system, the validation step was a critical component. GPT3.5-Turbo, when prompted in Arabic, demonstrated superior performance in this role, boasting an impressive validation accuracy rate of 89.32%.
- Total Performance: GPT4 displayed remarkable proficiency in this role, surpassing expectations with an impressive validation accuracy rate of 78.95% when prompted in English.

In our quest to generate clues from provided keywords, we engaged in the fine-tuning process using a curated dataset (refer to Section 3). We fine-tuned three distinct models, namely GPT3-DaVinci, GPT3.5-Turbo, and GPT3-Curie. We rigorously tested the performance of each model by generating clues for a carefully chosen set of 2000 educational-related keywords. Notably, the fine-tuned GPT3.5-Turbo outperformed the others, consistently producing high-quality clues with a remarkable success rate of 81%.

Utilizing the data generated through the evaluation

of fine-tuned models, we construct classifiers to distinguish between acceptable and non-acceptable clues for a specified keyword. The most effective model in this task was GPT3-Davinci, achieving an impressive accuracy rate of 85.74%.
Our process to produce educational crossword layouts is both efficient and diverse. We hope that these findings will enrich the learning process and foster interactive learning. The developed system can be integrated into current teaching methods to enhance educational practices. As a future course of action, we plan on venturing into the development of more advanced models for more direct clue and answer pair generation and examine specialized models for different clue types. We also intend to implement this system in actual classrooms and evaluate its impact. Our goal is to revolutionize the creation of educational crossword puzzles and usher in an era of unique teaching practices.

Acknowledgments

The funding for this paper was provided by the TAILOR project and the HumanE-AI-Net projects, both supported by the EU Horizon 2020 research and innovation program under GA No 952215 and No 952026, respectively.

References

Bhavna Arora and NS Kumar. 2019. Automatic keyword extraction and crossword generation tool for indian languages: Seekh. In *2019 IEEE Tenth International Conference on Technology for Education (T4E)*, pages 272–273. IEEE.

Yolanda Dita Bella and Endang Mastuti Rahayu. 2023. The improving of the student's vocabulary achievement through crossword game in the new normal era. *Edunesia: Jurnal Ilmiah Pendidikan*, 4(2):830–842.

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

Sunita M Dol. 2017. Gpbl: An effective way to improve critical thinking and problem solving skills in engineering education. *J Engin Educ Trans*, 30(3):103–13.

Dzulfikri Dzulfikri. 2016. Application-based crossword puzzles: Players' perception and vocabulary retention. *Studies in English Language and Education*, 3(2):122–133.

Jennifer Esteche, Romina Romero, Luis Chiruzzo, and Aiala Rosá. 2017. Automatic definition extraction and crossword generation from spanish news text. *CLEI Electronic Journal*, 20(2).

Serap Kaynak, Sibel Ergün, and Ayşe Karadaş. 2023. The effect of crossword puzzle activity used in distance education on nursing students' problem-solving and clinical decision-making skills: A comparative study. *Nurse Education in Practice*, 69:103618.

Shane T Mueller and Elizabeth S Veinott. 2018. Testing the effectiveness of crossword games on immediate and delayed memory for scientific vocabulary and concepts. In *CogSci*.

RS Nickerson. 1977. Crossword puzzles and lexical memory. In *Attention and performance VI*, pages 699–718. Routledge.

OpenAI. 2023. Gpt-4 technical report.

Wiwat Orawiwatnakul. 2013. Crossword puzzles as a learning tool for vocabulary development. *Electronic Journal of Research in Education Psychology*, 11(30):413–428.

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. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Bali Ranaivo-Malançon, Terrin Lim, Jacey-Lynn Minoi, and Amelia Jati Robert Jupit. 2013. Automatic generation of fill-in clues and answers from raw texts for crosswords. In *2013 8th International Conference on Information Technology in Asia (CITA)*, pages 1–5. IEEE.

Leonardo Rigutini, Michelangelo Diligenti, Marco Maggini, and Marco Gori. 2008. A fully automatic crossword generator. In *2008 Seventh International Conference on Machine Learning and Applications*, pages 362–367. IEEE.

Leonardo Rigutini, Michelangelo Diligenti, Marco Maggini, and Marco Gori. 2012. Automatic generation of crossword puzzles. *International Journal on Artificial Intelligence Tools*, 21(03):1250014.

Ali Safaya, Moutasem Abdullatif, and Deniz Yuret. 2020. Kuisail at semeval-2020 task 12: Bert-cnn for offensive speech identification in social media. *arXiv preprint arXiv:2007.13184*.

Corina Sandiuc and Alina Balagiu. 2020. The use of crossword puzzles as a strategy to teach maritime english vocabulary. *Scientific Bulletin" Mircea cel Batran" Naval Academy*, 23(1):236A–242.

Elizabeth Yuriev, Ben Capuano, and Jennifer L Short. 2016. Crossword puzzles for chemistry education: learning goals beyond vocabulary. *Chemistry education research and practice*, 17(3):532–554.

Peyman Zamani, Somayeh Biparva Haghighi, and Majid Ravanbakhsh. 2021. The use of crossword puzzles as an educational tool. *Journal of Advances in Medical Education & Professionalism*, 9(2):102.

Victor Samuel Zirawaga, Adeleye Idowu Olusanya, and Tinovimbanashe Maduku. 2017. Gaming in education: Using games as a support tool to teach history. *Journal of Education and Practice*, 8(15):55–64.

A Appendix

This study entailed developing a classifier to distinguish optimal and sub-optimal crossword clue-answer pairs. Crossword puzzles necessitate linguistic acumen, innovation, and adherence to construction guidelines for quality clues and answers. Such a classifier auto-evaluates the clue-answer quality, aiding puzzle designers and improving puzzle-solving experiences. This provides insight into key aspects of language and puzzle architecture.

The development of a robust framework for determining acceptable and unacceptable crossword clue-answer pairs is crucial to the effectiveness of a classifier. This provides the groundwork upon which our classifier can effectively discriminate between high-quality clues and ill-fit ones. Rigorous adherence to these guidelines facilitates accuracy in quality evaluation by the classifier and ultimately enhances the appeal and satisfaction derived from crossword puzzles.

Let us now probe into the salient features of the guideline for assessing crossword clue-answer quality:

- Coherence and Relevance: An ideal pair of clues and answers should display an evident and significant association between the two. The clue should offer adequate context or prompts that guide solvers toward the desired solution. The answer should be linear to the clue and sound logical within the subject matter or theme of the given puzzle.

- Wordplay and Creativity: A finely constructed crossword clue frequently employs wordplay, ingenious nuances, or concealed connotations that provoke and fascinate solvers. Seek clues that necessitate unconventional thinking, dual meanings, or linguistic resourcefulness. An effective clue-answer duo will enthrall the solvers, enhancing the puzzle's intrigue and pleasure.

- Unambiguity and Specificity: Clues should be unequivocal and clear-cut, presenting solvers with a distinct and exact solution. Refrain from clues that allow for multiple interpretations or result in various potential answers. The aim is to propose a single accurate answer that correlates directly with the intended meaning of the clue.

- Linguistics and Grammar: Both the clue and the answer should conform to correct grammar, syntax, and language norms. It's essential to verify that the language utilized in the clue-answer duo is grammatically accurate, coherent, and appropriate for a crossword puzzle.

- Universal Knowledge and Equity: Clues should be based on general knowledge or facts that a wide spectrum of solvers would reasonably be anticipated to understand. Refrain from using excessively obscure or specialized references, which only a small subset of solvers would recognize. An optimal clue-answer match should maintain a balance between challenge and fairness, accommodating a varied assortment of puzzle aficionados.

Adhering to these guidelines, we can construct a dataset capable of building a dependable classifier to differentiate between well-formulated crossword clue-answer pairs and those that are nonsensical or inappropriate. This classifier holds the potential to transform the process of creating, evaluating, and solving crossword puzzles. It offers crucial insights into the art of crafting puzzles that are both engaging and intellectually challenging.

B Appendix

The following prompts were employed for (Keyword Generation, Clue Generation, and Clue Verification) in both the Arabic and English versions:

English Keyword Extraction Prompt:

"Objective: Your task is to extract keywords (maximum 2 words) from a given text to create short crossword definitions. Please follow these steps to achieve the objective:

Keyword extraction: Extract the most important keywords from the text.

Validate keywords: Check if the keywords are well explained in the given text.

Final keywords: Remove all the keywords that are not well-defined in the text, based on the previous step.

Text: {text}

Here is an example Text:

الفقرة: ألْأسد حيوان من الثدييات من فصيلة السنوريات
وأحد السنوريات الأربعة الكبيرة المنتمية لجنس النمور ، وهو
يُعد ثاني أكبر السنوريات في العالم بعد الببر، حيث تفوق
كتلة الذكور الكبيرة منه ٢٥٠ كيلوغراما (٥٥٠ رطلًا). تعيش
معظم الأسود البرية المتبقية اليوم في إفريقيا جنوب الصحراء
الكبرى، ولا تزال جمهرة واحدة صغيرة مهددة بالانقراض
تعيش في آسيا بولاية غوجرات في شمال غربي الهند. كان
موطن الأسود شاسعًا جدًا في السابق، حيث كانت تتواجد في
شمال إفريقيا، الشرق الأوسط، وآسيا الغربية، حيث انقرضت
منذ بضعة قرون فقط. وحتى بداية العصر الحديث
(الهولوسين، منذ حوالي ١٠,٠٠٠ سنة)، كانت الأسود تُعتبر
أكثر ثدييات اليابسة الكبرى انتشارا بعد الإنسان، حيث كانت
توجد في معظم أنحاء إفريقيا، الكثير من أنحاء أوراسيا من
أوروبا الغربية وصولا إلى الهند، وفي الأمريكيتين، من يوكون
حتى البيروآ

Below are the legitimate keywords extracted from the provided text:

الكلمات المفتاحية: الأسد، الثدييات، فصيلة السنوريات،
الأسود البرية، إفريقيا، الهند، شمال إفريقيا، الشرق الأوسط، آسيا
الغربية، انتشار، الإنسان ، نمور ، ذكور

Use the following output format:

Keywords: <Final keywords>"

English Clue Generation Prompt:

"Your objective is to create short crossword clues for a list of keywords based on the given text:

Keywords: {keywords}
Text: {text}

Follow these steps to achieve the task:

Identify the part of the text that contains information about each provided keyword.

Generate short Arabic crossword clues (maximum 4 words) for all the keywords, using just the information from the text.

Here is an example Text:

الفقرة: أسد حيوان من الثدييات من فصيلة السنوريات
وأحد السنوريات الأربعة الكبيرة المنتمية لجنس النمور. وهو
يُعد ثاني أكبر السنوريات في العالم بعد الببر، حيث تفوق
كتلة الذكور الكبيرة منه ٢٥٠ كيلوغراما (٥٥٠ رطلًا). تعيش
معظم الأسود البرية المتبقية اليوم في إفريقيا جنوب الصحراء
الكبرى، ولا تزال جمهرة واحدة صغيرة مهددة بالانقراض
تعيش في آسيا بولاية غوجرات في شمال غربي الهند. كان
موطن الأسود شاسعًا جدًا في السابق، حيث كانت تتواجد في
شمال إفريقيا، الشرق الأوسط، وآسيا الغربية، حيث انقرضت
منذ بضعة قرون فقط. وحتى بداية العصر الحديث
(الهولوسين، منذ حوالي ١٠,٠٠٠ سنة)، كانت الأسود تُعتبر
أكثر ثدييات اليابسة الكبرى انتشارا بعد الإنسان، حيث كانت
توجد في معظم أنحاء إفريقيا، الكثير من أنحاء أوراسيا من
أوروبا الغربية وصولا إلى الهند، وفي الأمريكيتين، من يوكون
حتى البيرو.

Below is a list of valid keywords for the provided text:

الكلمات المفتاحية: أسد، حيوان، ثدييات، سنوريات،
سنوريات الأربعة، جنس النمور، ببر، الزكور الكبيرة، إفريقيا،
صحراء الكبرى، أمريكيتين

Here is a compilation of valid clue-answer pairs corresponding to the provided keywords and text:

Keyword: أسد
Clue: حيوان ثديي من السنوريات

Keyword: حيوان
Clue: ينتمي لفصيلة السنوريات

Keyword: ثدييات
Clue: نوع من الحيوانات

Keyword: سنوريات
Clue: تشمل الأسد

Keyword: سنوريات الأربعة
Clue: مجموعة من السنوريات الكبيرة

Keyword: جنس النمور
Clue: يعتبر الأسد منه

Keyword: ببر
Clue: السنورية الأكبر في العالم

Keyword:الزكور الكبير
Clue: تتجاوز وزنها ٢٥٠ كيلوغرام

Keyword: إفريقيا
Clue: مكان عيش معظم الأسود البرية

Keyword: صحراء الكبرى
Clue: تقع إلى جنوب إفريقيا

Keyword: أمريكيتين
Clue: تتواجد الأسود فيهما

Use the following format:

Keyword: <Keyword>

Clue: <Crossword Clue>

English Prompt for Hallucination Verification:

"Please assess the quality of the crossword clues based on the given text.

Text: {text}

Clues: {clues}

To accomplish this task, follow these steps:

Check Clue in the text: Verify Whether the content of each clue is present in the text.

If a content clue is found in the text, print True; otherwise, print False.

Use the following format for each clue:

Check Clue in the text:

<Check Clue in the text>"

Arabic Keyword Generation Prompt:

الهدف: استخراج كلمات مفتاحية (تتكون من كلمتين على الأكثر) من الفقرة التالية لإستخدام هذه الكلمات المفتاحية لإنشاء تعريفات قصيرة من اجل لعبة الكلمات المتاقطعة تتأكد من استخراج اهم الكلمات المفتاحية من الفقرة ثم قم بعمل فحص لهذه الكلمات المتقاطعة اذا كان تم شرحها بشكل جيد و واضح في الفقرة واذا لم تجد شرح وافي لكلمة من الكلمات المفتاحية فقم بالتخلص منها

الفقرة:

{text}

مثال للمطلوب: هذه الفقرة التي قمت بإستخراج منها الكلمات المفتاحية الفقرة: ألأسد حيوان من الثدييات من فصيلة السنوريات وأحد السنوريات الأربعة الكبيرة المنتمية لجنس النمور ، وهو يُعد ثاني أكبر السنوريات في العالم بعد الببر، حيث تفوق كتلة الذكور الكبيرة منه ٢٥٠ كيلوغراما (٥٥٠ رطلًا). تعيش معظم الأسود البرية المتبقية اليوم في إفريقيا جنوب الصحراء الكبرى، ولا تزال جمهرة واحدة صغيرة مهددة بالانقراض تعيش في آسيا بولاية غوجرات في شمال غربي الهند. كان موطن الأسود شاسعًا جدًا في السابق، حيث كانت تتواجد في شمال إفريقيا، الشرق الأوسط، وآسيا الغربية، حيث انقرضت منذ بضعة قرون فقط. وحتى بداية العصر الحديث (الهولوسين، منذ حوالي ١٠,٠٠٠ سنة)، كانت الأسود تُعتبر أكثر ثدييات اليابسة الكبرى انتشارا بعد الإنسان، حيث كانت توجد في معظم أنحاء إفريقيا، الكثير من أنحاء أوراسيا من أوروبا الغربية وصولا إلى الهند، وفي الأمريكيتين، من يوكون حتى البيروآ

الكلمات المفتاحية التي تم إستخراجها كالأتي الكلمات المفتاحية: الأسد، الثدييات، فصيلة السنوريات، الأسود البرية، إفريقيا، الهند، شمال إفريقيا، الشرق الأوسط، آسيا الغربية، انتشار، الإنسان ، نمور ، ذكور

شكل النتيجة النهائية: »الكلمات المفتاحية«

Arabic Clue Generation Prompt:

هدفك هو إنشاء الغاز قصيرة للعبة الكلمات المتقاطعة مناسبة للكلمات المفتاحية الأتية استنادا الى الفقره بحيث ان يكون كل كلمة مفتاحية يوجد لها اللغز خاص بها سأقوم بتزويدك بمثال بعد طريقة اتمام المهمة

الكلمات المفتاحية: الكلمات المفتاحيه

الفقرة: الفقره

استخدم هذه الطريقة لإتمام المهمة:

قم بالتعرف على الاجزاء التي تحتوي على الكلمات المفتاحية في الفقرة قم بإنشاء لغز لكل الكلمات المفتاحية بإستخدام المعلومات في الفقرة تأكد من انه لا يوجد اي كلمات مساعدة للوصول إلى الكلمة المفتاحية لهذا اللغز في اللغز الذي تم إنشاءه قم بإنشاء اللغز بحيث يدل فقط على الكلمة المفتاحية و لا يتواجد في اللغز نفسه تأكد من ان اللغز اجابته كلمة مفتاحية واحده فقط تأكد من ان لكل من الكلمات المفتاحية يوجد له لغز اذا وجد لغز مناسب

مثال للمطلوب:

- الفقرة كالأتي الفقرة: أسد حيوان من الثدييات من فصيلة السنوريات وأحد السنوريات الأربعة الكبيرة المنتمية لجنس النمور. وهو يُعد ثاني أكبر السنوريات في العالم بعد الببر، حيث تفوق كتلة الذكور الكبيرة منه ٢٥٠ كيلوغراما (٥٥٠ رطلًا). تعيش معظم الأسود البرية المتبقية اليوم في إفريقيا

جنوب الصحراء الكبرى، ولا تزال جمهرة واحدة صغيرة مهددة بالانقراض تعيش في آسيا بولاية غوجرات في شمال غربي الهند. كان موطن الأسود شاسعًا جدًا في السابق، حيث كانت تتواجد في شمال إفريقيا، الشرق الأوسط، وآسيا الغربية، حيث انقرضت منذ بضعة قرون فقط. وحتى بداية العصر الحديث (الهولوسين، منذ حوالي ١٠,٠٠٠ سنة)، كانت الأسود تُعتبر أكثر ثدييات اليابسة الكبرى انتشارا بعد الإنسان، حيث كانت توجد في معظم أنحاء إفريقيا، الكثير من أنحاء أوراسيا من أوروبا الغربية وصولا إلى الهند، وفي الأمريكيتين، من يوكون حتى البيرو.

- الكلمات المفتاحية كالأتي الكلمات المفتاحية: أسد، حيوان، ثدييات، سنوريات، سنوريات الأربعة، جنس النمور، ببر، الزكور الكبيرة، إفريقيا، صحراء الكبرى، أمريكيتين

هذه النتيجة:

الكلمة المفتاحية: أسد اللغز: حيوان ثديي من السنوريات

الكلمة المفتاحية: حيوان اللغز: ينتمي لفصيلة السنوريات

الكلمة المفتاحية: ثدييات اللغز: نوع من الحيوانات

الكلمة المفتاحية: سنوريات اللغز: تشمل الأسد

الكلمة المفتاحية: سنوريات الأربعة اللغز: مجموعة من السنوريات الكبيرة

الكلمة المفتاحية: جنس النمور اللغز: يعتبر الأسد منه

الكلمة المفتاحية: ببر اللغز: السنورية الأكبر في العالم

الكلمة المفتاحية: الزكور الكبيرة اللغز: تتجاوز وزنها ٢٥٠ كيلوغرام

الكلمة المفتاحية: إفريقيا اللغز: مكان عيش معظم الأسود البرية

الكلمة المفتاحية: صحراء الكبرى اللغز: تقع إلى جنوب إفريقيا

الكلمة المفتاحية: أمريكيتين اللغز: تتواجد الأسود فيهما

شكل النتيجة النهائية :

اللغز: اللغز

الكلمة المفتاحية: الكلمة المفتاحية

Arabic Prompt for Hallucination Verification:

قم بتقييم جودة الألغاز على حسب الفقرة الأتية

الفقرة: الفقرة

الألغاز: اللغز

لتقوم بهذه المهمة قم بالأتي:

قم بفحص اللغز في الفقرة. إذا كانت الفقرة تحتوي على كل من الألغاز. قم بطباعة صحيح و إذا لم تجده قم بطباعة خطأ
تأكد من القيام بالسابق لكل لغز منفرد و طباعة النتيجة لكل لغز

قم بالتعامل مع كل لغز على حدى

استخدم الصيغة الأتية للنتيجة النهائية فقط قم بطباعة اللغز:النتيجةْ بدون اي شرح او اي شئ اخر

الصيغة النهائية:

اللغز: النتيجة

C Appendix

In the upcoming section, you will find English translations of the Arabic content within this paper. These translations have been included to improve understanding for readers who may have limited proficiency in Arabic, ultimately ensuring greater accessibility to the content. The translation for the Figure 3 content is as follows:

Input paragraph:

الذرة هي أصغر حجر بناءٍ أو أصغر جزء من العنصر الكيميائي يمكن الوصول إليه والذي يحتفظ بالخصائص الكيميائية لذلك العنصر. يرجع أصل الكلمة الإنجليزية إلى الكلمة الإغريقية أتوموس، والتي تعني غير القابل للانقسام؛ إذ كان يعتقد أنه ليس ثمة ما هو أصغر من الذرة. تتكون الذرة من سحابة من الشحنات السالبة (الإلكترونات) التي تدور حول نواة موجبة الشحنة صغيرة جدًا في المركز، وتتكون النواة من بروتونات موجبة الشحنة، ونيوترونات متعادلة، وتعدّ الذرة هي أصغر جزء من العنصر يمكن أن يتميز به عن بقية العناصر؛ إذ كلما غصنا أكثر في المادة لنلاقي البنى الأصغر لن يعود هناك فرق بين عنصر وآخر. فمثلاً، لا فرق بين بروتون في ذرة حديد وبروتون آخر في ذرة يورانيوم مثلًا، أو ذرة أي عنصرٍ آخر. الذرة، بما تحمله من خصائص؛ عدد بروتوناتها، كتلتها، توزيعها الإلكتروني... تصنع الفروقات بين العناصر المختلفة، وبين الصور المختلفة للعنصر نفسه (المسماة بالنظائر)، وحتى بين كَون هذا العنصر قادرًا على خوض تفاعل كيميائيماأملا.

English translation of the input paragraph:

The atom is the smallest building block or the smallest part of an element that can be reached and retains the chemical properties of that element. The English word 'atom' is derived from the Greek word 'atomos,' which means indivisible, as it was believed that there was nothing smaller than the atom. An atom consists of a cloud of negative charges (electrons) orbiting around a very small positively charged nucleus at the center. The nucleus is composed of positively charged protons and neutral neutrons. The atom is the smallest unit of an element that can be distinguished from the rest of the elements. As we delve deeper into matter, we find smaller structures, but there is no longer a distinction between one element and another. For example, there is no difference between a proton in an iron atom and another proton in a uranium atom, or in the atom of any other element. Atoms, with their properties, such as the number of protons, mass, and electronic distribution, create differences between different elements, between different forms of the same element (called isotopes), and even whether this element is capable of undergoing a chemical reaction or not.

Extracted keywords (English and Arabic):

Atom - ذرة
Chemical Element - العنصر الكيميائي
Chemical Properties - الخصائص الكيميائية
Electrons - الإلكترونات
Nucleus - النواة
Protons - البروتونات
Neutrons - النيوترونات
Elements - العناصر
Isotopes - النظائر
Chemical Reaction - تفاعلكيميائي

Generated Clue-Answer Pairs (English and Arabic):

الذرة: أصغر جزء من العنصر الكيميائي يمكن الوصول إليه
Atom: The smallest unit of a chemical element that can be reached.

العنصر الكيميائي: يتكون من الذرات ويحتفظ بالخصائص الكيميائية
Chemical Element: Composed of atoms and retains chemical properties.

الخصائص الكيميائية: يحتفظ بها العنصر الكيميائي
Chemical Properties: Characteristics retained by the chemical element.

الإلكترونات: تدور حول النواة في الذرة
Electrons: Orbits around the nucleus in the atom.

النواة : تدور حول النواة في الذرةتتكون من البروتونات والنيوترونات في الذرة
Nucleus: Composed of protons and neutrons in the atom.

البروتونات: تتواجد في النواة وتحمل شحنة موجبة
Protons: Located in the nucleus and carry a positive charge.

النيوترونات: تتواجد في النواة ولا تحمل شحنة
Neutrons: Located in the nucleus and carry no charge.

العناصر: تتكون من الذرات وتختلف بحسب عدد البروتونات في النواة
Elements: Composed of atoms and vary based on the number of protons in the nucleus.

النظائر: صور مختلفة للعنصر نفسه
Isotopes: Different forms of the same element.

تفاعل كيميائي: يمكن أن يخوضه العنصر بحسب خصائصهالكيميائية
Chemical Reaction: Can be undergone by the element based on its chemical properties.

Next, we will include the English translations for the examples we presented as both acceptable and unacceptable generated clues from the fine-tuned models in Tables 3 and 4.

Acceptable Examples(Arabic and English):

نجوم : في السماء ليلا
Stars: in the sky at night

كروم : من المعادن
Ores: from minerals

قوة: قدرة
Strength: capability

Unacceptable Examples(Arabic and English):

زرافة: من الحشرات
Giraffe: from the insects

مثلث : مثنى مثلث
Triangle: plural triangle

عمة : اخت والد او والدة
Aunt: sister of a parent or a parent's sister

Machine Translation of Omani Arabic Dialect from Social Media

Khoula Kahlan Al Kharusi
Sultan Qaboos University
khoula.alkharusi@gmail.com

Abdurahman Khalifa AAlAbdulsalam
Sultan Qaboos University
a.aalabdulsalam@squ.edu.om

Abstract

Research studies on Machine Translation (MT) between Modern Standard Arabic (MSA) and English are abundant. However, studies on MT between Omani Arabic (OA) dialects and English are very scarce. This research study focuses on the lack of availability of an Omani dialect parallel dataset, as well as MT of OA to English. The study uses social media data from X (formerly Twitter) to build an authentic parallel text of the Omani dialects[1]. The research presents baseline results on this dataset using Google Translate, Microsoft Translation, and Marian NMT. A taxonomy of the most common linguistic errors is used to analyze the translations made by the NMT systems to provide insights on future improvements. Finally, transfer learning is used to adapt Marian NMT to the Omani dialect, with significant improvement of 9.88 points in the BLEU score.

1 Introduction

In the era of social media and worldwide communication, Machine Translation (MT) has become essential in lowering or eliminating the language barrier between people (Franceschini et al., 2020). Using artificial intelligence, users can translate any post from any language without human involvement. Recently MT underwent a remarkable evolution thanks to deep learning and artificial neural network models (Baniata et al., 2021). Although MT research attempted to produce high-quality translations of the most widely used languages, which are well documented with abundant sources, it still has a long way to go in terms of languages that are not as well documented, such as Arabic dialects.

Over the past decade, the Arabic language has drawn much interest from the MT community. However, most MT contributions focus on Modern Standard Arabic (MSA), while the translation of Arabic dialects is still in its early stages. Arabic is the world's fifth most widely used language, with almost 450 million speakers in 22 countries. Classical Arabic (CA) and MSA are the standard Arabic varieties recognized by Western linguists. The Quran, classical texts, and old Arabic literature are written in CA. MSA is a modern form that is based on the syntactic, morphological, and phonological structures of CA. MSA is the primary form of official communication in the Arab world that is used in education, business, news, and legislation (Al-Qaraghuli et al., 2021). Arabic dialects are used informally in day-to-day conversations throughout the Arab world. Arabic dialects are primarily spoken-only languages; however, in the last decade, these dialects have become increasingly prevalent in social media, text messages, TV shows, and other forms of informal communication. Nowadays, Arabic dialects are being used increasingly in written format for informal communication online (Harrat et al., 2019).

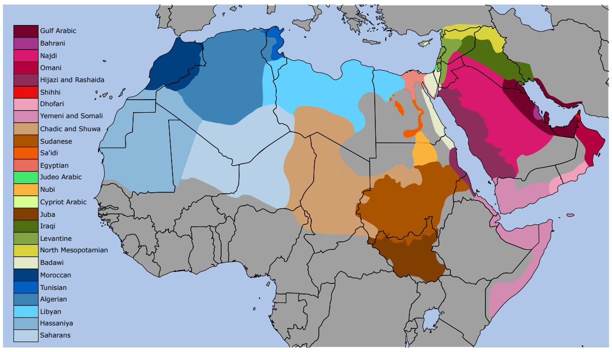


Figure 1: Geographic spread of Arabic dialects (Schmitt, 2020)

Despite the extensive use of Arabic dialects, they are considered low-resource language which hinder MT development. Arabic dialects vary from MSA in terms of phonology, semantics, morphology, and syntax (Harrat et al., 2019). They simplify many standard Arabic rules while simultaneously

[1] Dataset availabel in Github https://github.com/khoula-k/OmaniArabicTranslation

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 302–309
December 7, 2023 ©2023 Association for Computational Linguistics

introducing new sets of rules that add additional complications. Therefore, most MSA resources and tools cannot be easily adapted to translate Arabic dialects (Harrat et al., 2019). The lack of standard orthography is one of the fundamental challenges associated with Arabic dialects. Arabic dialects have diglossia, a linguistic phenomenon in which the speakers mix two or more varieties of the same language (e.g., standard official language and local dialect) within the same context (Farghaly and Shaalan, 2009; Harrat et al., 2019). It is worth noting that Arabic has a diverse range of colloquial varieties, with over 27 variations existing worldwide. These varieties exhibit varying degrees of mutual understanding, highlighting Arabic's nuanced and diverse nature (Elgabou and Kazakov, 2017). Figure 1 provides a basic overview of the geographic distribution of these dialects. There are two primary ways to approach colloquial Arabic MT. The first involves translation between MSA and colloquial Arabic dialects then to foreign language therefore MSA acting as an intermediate language. The second approach involves translation of Arabic dialects into foreign languages directly (Harrat et al., 2019). It is worth noting that all contributions in this field are primarily related to the English language.

This research focuses on the MT of Omani Arabic (OA). Oman's location, surrounded by the Indian Subcontinent, Persia, Arabia, and East African coasts, played a significant role in shaping its history and the languages spoken by its people. Despite Oman's small population, its linguistic context is diverse. Some Omanis speak multiple indigenous languages, such as Jibbali, Shahri, and Mehri, each with thousands of speakers, in addition to Bathari, Harsusi, and Hobyot, with a few hundred speakers each (Al-Balushi, 2017). Additionally, some Omanis speak non-indigenous languages, including Persian, Aajmi, Kumzari from Iran, Baluchi from Baluchistan, Zidjali from Pakistan, Kojki/Luwati from India, and Swahili from East Africa (Al-Balushi, 2017). The impact of various languages on OA is particularly evident in its vocabulary, featuring words borrowed from Hindi, such as guniyyah (meaning sack) and bigli (referring to an electric torch), as well as Persian words like drishah (window) and saman (stuff). English has also contributed words such as sekal (bicycle), batri/betri (battery), swik (switch), and beb (pipe), while Portuguese brings in banderah (flag) and mez (table).

The prevalent dialect in Oman differs from that dominant in the rest of the Arabian Gulf. It is mostly in the form of the Hadari (Sedentary) dialect rather than a Bedouin one (Nabhani, 2011). The Hadari dialect is prevalent in the northern part of Oman, including the capital Muscat, and is also used in most TV shows.

Limited research is available on translating colloquial Arabic dialects, particularly Omani dialects. While most prior works group OA dialects with other Gulf dialects, more research is necessary. It is important to note that while the Gulf region may share cultural similarities, it cannot be assumed that they share linguistic homogeneity. Moreover, OA datasets used in prior works are not publicly available. This research aims to close this gap by creating an authentic Omani Arabic-English parallel corpus that is available for public use. The dataset will be used to adapt an existing Arabic Neural Machine Translation (NMT) system to the Omani dialect.

2 Related Works

In this section, we will explore the literature on dialectical datasets and the MT of Arabic dialects.

2.1 Dialectical Arabic Datasets

In the literature, various dialectical parallel Arabic datasets have been mentioned. Nonetheless, this subsection will focus on the datasets that are publicly available. The **MADAR** corpus (Multi-Arabic Dialect Applications and Resources) (Bouamor et al., 2018) is a collection that comprises parallel sentences encompassing the dialects of 25 cities in the Arab world, along with MSA, English, and French. The corpus is created by translating select sentences from the Basic Traveling Expression Corpus (BTEC), which was in Japanese, English, and Chinese (Takezawa et al., 2007) to the different dialects.

The **MPCA** (Multidialectal Parallel Corpus of Arabic), as documented in (Bouamor et al., 2014), is comprised of 2,000 sentences that represent five Arabic dialects, as well as English and MSA. The corpus was developed by tasking four translators who are native speakers of Palestinian, Syrian, Jordanian, and Tunisian colloquial Arabic varieties to translate 2,000 sentences originally written in Egyptian Arabic into their respective dialects.

The **PADIC** (Parallel Arabic DIalect Corpus) (Meftouh et al., 2015) multi-dialectal corpus contains six dialects in addition to MSA. Two Algerian dialect corpora were created: Annaba's dialect (a city in Algeria) from daily conversations and the dialect from movies/TV shows in the Algiers dialect. Both were transcribed and translated manually. They were later used to obtain other MSA and dialectal corpora.

Currently, the MADAR dataset is the only source we found for Omani Arabic, with the dialect of the capital city, Muscat. Out of the 25 dialects in the MADAR corpus, the dialect of Muscat is the most similar to MSA with an overlap score of 37.5% (Bouamor et al., 2018; Salameh et al., 2018). It has been stated that the translators were native speakers of the dialects, and they got access to English and French versions of the corpus without the MSA to avoid biased translation. Upon analyzing the OA in the MADAR dataset by a native speaker of the Omani dialect, it was observed that it predominantly reflects a dialect that is more oriented toward MSA with some Bedouin influence rather than the sedentary Muscat-Omani dialect.

2.2 Machine Translation of Dialectical Arabic

When it comes to Arabic dialect MT, there are two main approaches. The first approach focuses on translating between MSA and its corresponding dialects, while the other approach aims to translate Arabic dialects into foreign languages. It is important to note that most research in this field is related to translating into English.

In the field of colloquial Arabic MT, one of the earliest studies was conducted by Sawaf in 2010. The study focused on dialect normalization and used a hybrid RBMT and SMT to translate into MSA (Sawaf, 2010).

Wael Salloum and Nizar Habash have contributed several papers to the field of colloquial Arabic translation. One of their approaches, as described in (Salloum and Habash, 2011), involved a rule-based method for producing MSA paraphrases of dialectal Arabic OOV (out of vocabulary words) in the Levantine and Egyptian dialects. They then combined this with the results generated by ADAM (Salloum and Habash, 2014), to create Elissa (Salloum and Habash, 2012), which can handle Levantine, Egyptian, Iraqi, and to a lesser extent Gulf Arabic. (Salloum and Habash, 2013) published an advanced version of their translation system, which translates dialectal Arabic to English by pivoting through MSA.

(Zbib et al., 2012) proposed a massive SMT-based system for Levantine and Egyptian dialects. They created parallel corpora of Levantine-English and Egyptian-English and then trained their statistical translation model using direct translation and pivoting through MSA. In contrast to the previously discussed approach that utilized a statistical model in (Sghaier and Zrigui, 2020), a rule-based system was developed to translate from the Tunisian dialect to MSA without relying on statistical models.

The following works utilized a modern technique of deep neural networks to translate Arabic dialects. AraBench (Sajjad et al., 2020) presented evaluation benchmarks for dialectal Arabic to English. The paper details several experiments conducted in this regard. They used the OpenNMT model (Klein et al., 2017) and trained it in extensive heterogeneous MSA and dialectical Arabic data. This base model is then fine-tuned towards in-domain dialectical training data. Lastly, they used back-translation to increase the dialectal Arabic-English training data size. (Baniata et al., 2021) is using the state-of-the-art Transformer models to translate DA to MSA using subword units for tokenization, effectively solving the issue of out-of-vocabulary words. The subword segmentation algorithm operates under the premise that a word comprises a combination of subwords.

All of the studies that focus on Omani dialect utilized the MADAR corpus. In the research conducted by (Baniata et al., 2021), Omani dialect is grouped with other Gulf dialects, making it difficult to assess the system's performance for the Omani Arabic specifically. On the other hand, AraBench (Sajjad et al., 2020) has tested OA independently and achieved a BLEU score of 39.5% for the translation model trained for MSA-EN translation. However, it is worth noting that the Muscat-MADAR dataset used AraBench may not be representative of OA with a lot of influence from MSA.

2.3 Machine Translation for Low-resource Languages

MT has significantly improved with the use of deep neural networks. However, the downside is that it demands extensive training data and takes up a lot of computing power and time. Fortunately, transfer learning offers a practical solution by utilizing prior knowledge of a trained model to improve performance on related tasks. This approach reduces the need for extensive training data, saving time and resources. (Zoph et al., 2016) used a French-English model as the parent model for low-resource language pairs such as Hausa, Turkish, Uzbek, and Urdu into English. On average, NMT shows 5.6 BLUE points score improvement from transfer learning. The researchers also explored the similarity between the parent and child languages. They conducted a transfer learning method using French and German as parent languages for the Spanish language. The results showed that French was a better parent language for Spanish, which could be the result of its greater similarity to the child's language. (Zoph et al., 2016) employs a transfer learning approach with a single parent and one child, whereas (Goyal et al., 2020) utilizes transfer learning by leveraging related languages. Two simple and effective methods are introduced: Multilingual Transfer Learning, which helps improve low-resource languages by utilizing parallel data from related languages, regardless of their resource levels, and Unified Transliteration and Subword Segmentation, which takes advantage of the similarities between related language pairs.

3 Omani Parallel Corpus

This study aims to translate Omani Arabic, utilizing original data of language in use on social media posts. X (formerly Twitter) was utilized to collect text representatives of the Omani dialect. X is a leading social media platform that contains trending news and topics and has a very large user base. Therefore, it offers a valuable resource for conducting large-scale text analysis. Furthermore, API allows users to execute complex queries, such as retrieving all text related to a particular topic or extracting a specific user's posts.

This chapter will explain the full process of creating the Omani Parallel Corpus. After the completion of the corpus, we will present a translation baseline results on this corpus obtained using Google Translate, Microsoft Translation, and Marian NMT systems.

3.1 Data Collection

Each post by users on 'X' comes with metadata fields and values containing information such as the author, creation timestamp, message, location, etc. The data has been retrieved in JSON format using the platform API. Using `conversation_id`, each post and its related replies are collected in one file, which we consider conversations surrounding a particular topic initiated by the main post. We collected posts from a prominent news account in Oman (@oman1_news). Each post and its related replies were treated as a single document for a specific topic. As a result, we obtained a corpus consisting of 905 topics in the form of conversation and containing a total of 87,220 posts.

Real-world social media data typically comprises texts, images, and videos, often accompanied by offensive language and hate speech. The text is often noisy with hashtags, URLs, and foreign characters, and there may be instances of spelling errors. Additionally, individuals may use slang, which is an informal language unique to specific groups or geographic regions that carry cultural connotations with different meanings.

3.2 Corpus Linguistics

Each document in the resulting corpus is converted to a CSV file where the first row is the source post, and the following are replies. Table 1 provides a corpus summary in numbers. The most frequent tokens are the linking words, while the least frequent tokens are words with foreign characters.

Table 1: X Omani corpus

Number of topics/conversations	905
Total number of posts/messages	87,220
Total number of tokens	1,102,952
Unique tokens (vocabulary size)	118,821

3.3 Translation

We have asked volunteers to translate each document into English. Nine participants who are native speakers of the dialect have worked on the translations. We assigned unique sets of topics to each translator and asked them to produce a translation that precisely reflects the source sentence without making any assumptions. The translators were provided with the following guidelines:

- The English translation should retain the punctuation marks from the source sentence (like periods, commas, and question marks).
- When translating idioms and slang, it is important to convey the intended meaning rather than translating them literally.
- Disregard any posts containing offensive language, hate speech, or advertisements.

Table 2: Omani Arabic-English parallel corpus

OA	EN
ولي خطف علينا اليومين الماضيات شو؟!!	And what was it that just happened the past two days?!!
ما شوي الي خطف والي جالس يخطف عليهم.	What they faced and are facing isn't easy
يعني بكم جونية العيش دكتور اتّستوي	so how much will be the sack of rice doctor
اول مره نهزمهن	First time we beat them

- Avoid translating posts with Quran verses.

A total of **2906** posts have been translated, covering various topics. The Omani Parallel Corpus was created by combining all the translated posts. Table 2 below shows some examples from the parallel corpus.

4 Error Analysis in Omani Arabic MT

The most used measures for translation accuracy are automated. However, it can be difficult to establish a direct correlation between these measures and the actual errors present in the translations. A comprehensive analysis of errors is crucial for any natural language processing task, as it can reveal valuable insights into what went wrong and guide future research directions (Ângela Costa et al., 2015; Vilar et al., 2006). For the error classification, we adopted a simplified version of the taxonomy (Vilar et al., 2006) shown in figure 2. Error analysis can be a time-consuming task that requires linguistic expertise. Therefore, we conducted an analysis of a sample of sixty source sentences to identify errors generated by Google Translate and Marian NMT.

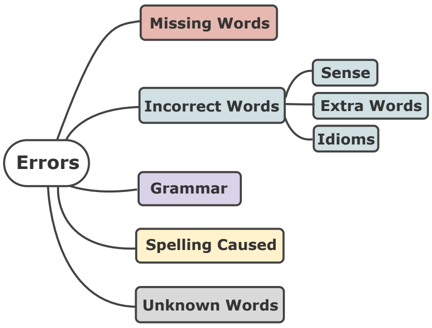


Figure 2: Adapted taxonomy for translation errors

Out of the 60 sentences translated by both MT engines, Google had incorrect translations in 54 cases, while Marian had errors in 57. Google's total number of translation errors was 112, while Marian had 150 errors. In Table 3, the number of translation errors produced by each translation system can be observed, categorized by the type of translation error. Both translation systems produce a similar number of translation errors across all categories except for missing words. Marian NMT dropped 36 more words than Google, which only dropped 11. The majority of errors were related to choosing the wrong word sense during translation.

Table 3: Error analysis results on Google Translate and Marian NMT

Error Type		Google	Marian
Missing Words		11	47
Incorrect Words	Sense	64	67
	Extra Words	5	5
	Idioms	8	4
Unknown Words		12	16
Grammar		6	3
Spelling Caused		6	6

5 Transfer Learning

To implement this approach, we start with a pre-trained NMT model that has been trained on a large parallel corpus (MSA-EN). We then use this model to initialize a new NMT model called the child model. This model is then trained on the domain dataset with a limited parallel data. Using the pre-trained parent model, the child model commences with established weights inherited from the parent model rather than starting with random weights. This method is particularly useful since the Omani parallel corpus is limited, and the MSA corpus provides a strong prior distribution over language vocabularies.

Marian NMT (Junczys-Dowmunt et al., 2018) is used as a neural translation system for the parent model. Marian is a highly efficient NMT framework that is built on pure C++, requiring minimal dependencies. The framework was mainly developed by the Adam Mickiewicz University and the University of Edinburgh. It is currently utilized in var-

Table 4: Arabic-English Opus corpus

Training	Arabic Tokens	English Tokens
126.6M	2.3G	3.9G

ious European projects and is the primary engine for translation and training behind the NMT launch at the World Intellectual Property Organization. Marian has found its niche in the growing world of open-source NMT toolkits due to two key aspects: it is built entirely on C++ which makes it very efficient. Additionally, it is self-contained with its own back-end that enables reverse-mode automatic differentiation using dynamic graphs.

Marian NMT model follows the original transformer architecture with six encoders and six decoders with eight attention heads in each layer (Junczys-Dowmunt et al., 2018). Language Technology Research Group at the University of Helsinki, Finland, trained Marian NMT on many language pairs from the OPUS-MT datasets. These models have been converted to `PyTorch`[2] using the transformers library by Hugging Face[3]. The Arabic-English[4] translation model was trained with a parallel dataset of 126.6M Arabic-English sentence pairs (see Table 4 below (Tiedemann, 2020; Tiedemann and Thottingal, 2020).

The Omani parallel corpus was split into a train set, a validation set, and a test set. The training set contains 70% of the whole corpus, and the remaining 30% is divided equally between the validation and test sets. The training was done using `Seq2SeqTraining` script from Hugging Face and activation function is `AdamW`. Before jumping to the model results, having a look at the validation and training loss is a good practice to ensure models are generalized and there is no over-fitting. Both validation and training loss decreased up until the fifth epoch. However, after the fifth epoch, the validation loss showed a slight increase. Hence, we have decided to proceed with five epochs for training.

Table 5: Omani corpus split

Dataset	Percentage	Parallel posts
Training Set	70%	2,034
Validation Set	15%	436
Test Set	15%	437

Table 6: Results of Google, Microsoft, Marian NMT, and the transfer learning model on a test set of OA corpus

	Google	Microsoft	Marian	Tuned Model
BLEU	34.98	34.26	24.22	34.11
chrF	60.55	61.28	49.02	59.81

In order to compare the fine-tuned model with various translation systems, we calculated the BLEU score for the validation set translated by Google, Microsoft, Marian NMT, and the transfer learning model. In Figure 6, the results indicate that Google achieved the highest BLEU score of 34.98, but it is noteworthy that Microsoft and our model were not far behind, scoring 34.26 and 34.11, respectively. On the other hand, Marian NMT scored the lowest with 24.22. Fine-tuning Marian NMT closed the performance gap between Marian and other translation engines (Google and Microsoft). Marian's initial BLEU score was 24.22, but after completing transfer learning in the OA training set, it increased significantly to 34.11, representing an improvement of 9.88 points. Our results outperformed (Zoph et al., 2016), which achieved a maximum improvement of 7.5 points. The training process is significantly influenced by the closeness of the parent model language to the child's model language.

Although the transfer learning model has displayed positive outcomes in Marian NMT, it has yet to surpass the MSA-English systems of Google and Microsoft. It would be advantageous to implement transfer learning in these systems, but they do not offer open-source models.

6 Conclusion

Using Machine Translation (MT) is an effective way to overcome language barriers in communication. While there are numerous research studies on MT from Modern Standard Arabic (MSA) to English, there is a significant lack of studies on translating Omani dialects. In this study, we aim to establish a first baseline targeted at the automatic translation of the written text of Omani dialects from social media.

Our initial step was to thoroughly analyze the literature on the translation of colloquial Arabic dialects and identify the datasets that contained an

[2] `PyTorch` is a machine learning framework based on the Torch library https://pytorch.org/

[3] Hugging Face develops tools for building applications using machine learning https://huggingface.co/

[4] https://huggingface.co/Helsinki-NLP/opus-mt-tc-big-ar-en

OA corpus. Only one source was available for reseacrh use and it may not be representative of the Omani dialect. We collected messages from social media to create an authentic Omani dialect corpus. Then we translated a total of 2906 messages. This corpus has been used to conduct a baseline study on existing MT Models' performance in Omani dialect translation, where we found that Google and Microsoft translation engines got higher BLEU scores reaching 33%, compared to Marian NMT, which scored 22.3%. We conducted a manual evaluation to identify Google and Marian NMT errors. After linguistically classifying the errors, we discovered that the most common error made by both NMTs was choosing the wrong word sense. We enhanced the translation of OA by utilizing transfer learning with Marian NMT. This resulted in a significant improvement of 9.88% in the BLEU score.

The main contribution of this research can be summarized as follows:

- Collecting and creating a parallel corpus of Omani dialects and English.
- Analyzing MT errors to inform future research direction.
- Applying transfer learning for OA using an existing MSA-English model.

We faced challenges because we had limited resources and time constraints. We didn't have the funds or time to hire professional translators for the corpus, and we couldn't review every sentence to select them for translation. Additionally, we utilized transfer learning with the available open-source model.

In the future, we hope to enhance the translation of the OA corpus by collaborating with linguistic experts, increasing the quantity of translated sentences, and providing multiple translations for each sentence to ensure an accurate evaluation.

References

Rashid Al-Balushi. 2017. Omani arabic: More than a dialect. *Macrolinguistics*, 4:80–125.

Mohammed Al-Qaraghuli, Gheith Abandah, and Ashraf Suyyagh. 2021. Correcting arabic soft spelling mistakes using transformers. pages 146–151. IEEE.

Laith H. Baniata, Isaac K.E. Ampomah, and Seyoung Park. 2021. A transformer-based neural machine translation model for arabic dialects that utilizes subword units. *Sensors*, 21.

Houda Bouamor, Nizar Habash, and Kemal Oflazer. 2014. A multidialectal parallel corpus of arabic. *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1240–1245.

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. *European Language Resources Association (ELRA)*.

Hani Elgabou and Dimitar Kazakov. 2017. Building dialectal Arabic corpora. In *Proceedings of the Workshop Human-Informed Translation and Interpreting Technology*, pages 52–57, Varna, Bulgaria. Association for Computational Linguistics, Shoumen, Bulgaria.

Ali Farghaly and Khaled Shaalan. 2009. Arabic natural language processing. *ACM Transactions on Asian Language Information Processing*, 8:1–22.

Dario Franceschini, Chiara Canton, Ivan Simonini, Armin Schweinfurth, Adelheid Glott, Sebastian Stüker, Thai-Son Nguyen, Felix Schneider, Thanh-Le Ha, Alex Waibel, Barry Haddow, Philip Williams, Rico Sennrich, Ondřej Bojar, Sangeet Sagar, Dominik Macháček, and Otakar Smrz. 2020. Removing european language barriers with innovative machine translation technology.

Vikrant Goyal, Sourav Kumar, and Dipti Misra Sharma. 2020. Efficient neural machine translation for low-resource languages via exploiting related languages.

Salima Harrat, Karima Meftouh, and Kamel Smaili. 2019. Machine translation for arabic dialects (survey). *Information Processing and Management*, 56:262–273.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

Karima Meftouh, Salma Jamoussi, Mourad Abbas, Crstdla ‡ Algiers, and Algeria Kamel Smaili. 2015. Machine translation experiments on padic: A parallel arabic dialect corpus.

Hala Al Nabhani. 2011. Language and identity in oman through the voice of local radio broadcasters.

Hassan Sajjad, Ahmed Abdelali, Nadir Durrani, and Fahim Dalvi. 2020. Arabench: Benchmarking dialectal arabic-english machine translation. pages 5094–5107. International Committee on Computational Linguistics.

Mohammad Salameh, Houda Bouamor, and Nizar Habash. 2018. Fine-grained Arabic dialect identification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1332–1344, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Wael Salloum and Nizar Habash. 2011. Dialectal to standard arabic paraphrasing to improve arabic-english statistical machine translation. pages 10–21. Association for Computational Linguistics.

Wael Salloum and Nizar Habash. 2012. Elissa: A dialectal to standard arabic machine translation system.

Wael Salloum and Nizar Habash. 2013. Dialectal arabic to english machine translation: Pivoting through modern standard arabic. *NAACL-HLT*.

Wael Salloum and Nizar Habash. 2014. Adam: Analyzer for dialectal arabic morphology. *Journal of King Saud University - Computer and Information Sciences*, 26:372–378. Special Issue on Arabic NLP.

Hassan Sawaf. 2010. Arabic dialect handling in hybrid machine translation. *Proceedings of the Conference of the Association for Machine Translation in the Americas (AMTA)*.

Genevieve A. Schmitt. 2020. *Relevance of Arabic Dialects: A Brief Discussion*, pages 1383–1398. Springer International Publishing, Cham.

Mohamed Ali Sghaier and Mounir Zrigui. 2020. Rule-based machine translation from tunisian dialect to modern standard arabic. *Procedia Computer Science*, 176:310–319. Knowledge-Based and Intelligent Information Engineering Systems: Proceedings of the 24th International Conference KES2020.

Toshiyuki Takezawa, Genichiro Kikui, Masahide Mizushima, and Eiichiro Sumita. 2007. Multilingual spoken language corpus development for communication research.

Jörg Tiedemann. 2020. The tatoeba translation challenge – realistic data sets for low resource and multilingual mt.

Jörg Tiedemann and Santhosh Thottingal. 2020. Opus-mt – building open translation services for the world.

David Vilar, Jia Xu, Luis Fernando D'haro, and Hermann Ney. 2006. Error analysis of statistical machine translation output.

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.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation.

Ângela Costa, Wang Ling, Tiago Luís, Rui Correia, and Luísa Coheur. 2015. A linguistically motivated taxonomy for machine translation error analysis. *Machine Translation*, 29:127–161.

Arabic Fine-Grained Entity Recognition

Haneen Abdallatif Liqreina
Birzeit University
Birzeit, Palestine
1195325@student.birzeit.edu

Mustafa Jarrar
Birzeit University
Birzeit, Palestine
mjarrar@birzeit.edu

Mohammed Khalilia
Birzeit University
Birzeit, Palestine
mkhalilia@birzeit.edu

Ahmed Oumar El-Shangiti
MBZUAI
Abu Dhabi, United Arab Emirates
ahmed.oumar@mbzuai.ac.ae

Muhammad Abdul-Mageed
UBC and MBZUAI
Vancouver, Canada
muhammad.mageed@ubc.ca

Abstract

Traditional NER systems are typically trained to recognize coarse-grained entities, and less attention is given to classifying entities into a hierarchy of fine-grained lower-level subtypes. This article aims to advance Arabic NER with fine-grained entities. We chose to extend Wojood (an open-source Nested Arabic Named Entity Corpus) with subtypes. In particular, four main entity types in Wojood, geopolitical entity (GPE), location (LOC), organization (ORG), and facility (FAC), are extended with 31 subtypes. To do this, we first revised Wojood's annotations of GPE, LOC, ORG, and FAC to be compatible with the LDC's ACE guidelines, which yielded $5,614$ changes. Second, all mentions of GPE, LOC, ORG, and FAC ($\sim 44K$) in Wojood are manually annotated with the LDC's ACE subtypes. We refer to this extended version of Wojood as $Wojood_{Fine}$. To evaluate our annotations, we measured the inter-annotator agreement (IAA) using both Cohen's Kappa and F_1 score, resulting in 0.9861 and 0.9889, respectively. To compute the baselines of $Wojood_{Fine}$, we fine-tune three pre-trained Arabic BERT encoders in three settings: flat NER, nested NER and nested NER with subtypes and achieved F_1 score of 0.920, 0.866, and 0.885, respectively. Our corpus and models are open-source and available at https://sina.birzeit.edu/wojood/.

1 Introduction

Named Entity Recognition (NER) is the task of identifying and classifying named entities in unstructured text into predefined categories such as people, organizations, locations, disease names, drug mentions, among others (li et al., 2020). NER is widely used in various applications such as information extraction and retrieval (Jiang et al., 2016), question answering (Liu et al., 2020), word sense disambiguation (Jarrar et al., 2023a; Al-Hajj and Jarrar, 2021), machine translation (Jain et al., 2019; Khurana et al., 2022), automatic summarization (Summerscales et al., 2011; Khurana et al., 2022), interoperability (Jarrar et al., 2011) and cybersecurity (Tikhomirov et al., 2020).

Traditional NER systems are typically trained to recognize coarse and high-level categories of entities, such as person (PERS), location (LOC), geopolitical entity (GPE), or organization (ORG). However, less attention is given to classifying entities into a hierarchy of fine-grained lower-level subtypes (Zhu et al., 2020; Desmet and Hoste, 2013). For example, locations (LOC) like Asia and Red Sea could be further classified into Continent and Water-Body, respectively. Similarly, organizations like Amazon, Cairo University, and Sphinx Cure can be classified into commercial, educational, and health entities, respectively. Belgium, Beirut, and Brooklyn can be classified into Country, Town, and Neighborhood instead of classifying them all as GPE. The importance of classifying named entities into subtypes is increasing in many application areas, especially in question answering, relation extraction, and ontology learning (Lee et al., 2006).

As will be discussed in the following sub-section, the number of NER datasets that support subtypes is limited, particularly for the Arabic language. The only available Arabic NER corpus with subtypes is the LDC's ACE2005 (Walker et al., 2005). However, this corpus is expensive. In addition, ACE2005 was collected two decades ago and hence may not be representative of the current state of Arabic language use. This is especially the case since language models are known to be sensitive to temporal and domain shifts (see section 5).

To avoid starting from scratch, we chose to extend upon a previously published and open-source Arabic NER corpus known as 'Wojood' (Jarrar et al., 2022). Wojood consists of $550K$ tokens manually annotated with 21 entity types. In particular, we manually classify four main entity types in Wojood (GPE, LOC, ORG, and FAC) with 31 new fine-grained subtypes. This extension is not straightforward as we have to change ($5,614$ changes) the original annotation of these four types of entities to align with LDC guidelines before extending them with subtypes. The total number of tokens that are annotated with the 31 subtypes is 47.6K.

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 310–323
December 7, 2023 ©2023 Association for Computational Linguistics

Our extended version of Wojood is hereafter called $Wojood_{Fine}$. We measure inter-annotator agreement (IAA) using both Cohen's Kappa and F_1, resulting in 0.9861 and 0.9889, respectively.

To compute the baselines for $Wojood_{Fine}$, we fine-tune three pre-trained Arabic BERT encoders across three settings: (i) flat, (ii) nested without subtypes, and (iii) nested with subtypes, using multi-task learning. Our models achieve 0.920, 0.866, and 0.885 in F_1, respectively.

The remaining of the paper is organized as follows: Section 2 overviews related work, and Section 3 presents the $Wojood_{Fine}$ corpus, the annotation process, and the inter-annotator-agreement measures. In Section 4, we present the experiments and the fine-tuned NER models. In Section 5 we present error analysis and out-of-domain performance and we conclude in Section 6.

2 Related Work

Most of the NER research is focused on coarse-grained named entities and typically targets a limited number of categories. For example, Chinchor and Robinson (1997) proposed three classes: person, location and organization. The Miscellaneous class was added to CoNLL-2003 (Sang and De Meulder, 2003). Additional four classes (geo-political entities, weapons, vehicles, and facilities) were also introduced in the ACE project (Walker et al., 2005). The OntoNotes corpus is more expressive as it covers 18 types of entities (Weischedel et al., 2013).

Coarse-grained NER is a good starting point for named entity recognition, but it is not sufficient for tasks that require a more detailed understanding of named entities (Ling and Weld, 2012; Hamdi et al., 2021).

Substantial research has been undertaken to identify historical entities. For instance, the HIPE shared task (Ehrmann et al., 2020a) focused on extracting named entities from historical newspapers written in French, German, and English. One of its subtasks was the recognition and classification of mentions according to finer-grained entity types. The corpus used in the shared task consists of tokens annotated with five main entity types and 12 subtypes, following the IMPRESSO guidelines (Ehrmann et al., 2020b). A similar corpus, called NewsEye, was collected from historical newspapers in four languages: French, German, Finnish, and Swedish (Hamdi et al., 2021). The corpus is annotated with four main types: PER, LOC, ORG, and PROD. The LOC entities were further classified into five subtypes, and the ORG entities into two subtypes. Desmet and Hoste (2013) proposed a one million fine-grained NER corpus for Dutch, which was annotated using six main entity types and 27 subtypes (10 subtypes for PERS, three for ORG, nine for LOC, three for PROD, and two for events).

Zhu et al. (2020) noted that NER models cannot effectively process fine-grained labels with more than 100 types. Thus, instead of having many fine-grained entities at the top level, they propose a tagging strategy in which they use 15 main entity types and 131 subtypes. Additionally, Ling and Weld (2012) proposed a fine-grained set of 112 tags and formulated the tagging problem as multi-class multi-label classification.

A recent shared task was organized by Fetahu et al. (2023) at SemEval-2023 Task 2, called MultiCoNER 2 (Fine-grained Multilingual Named Entity Recognition). A multilingual corpus (MULTICONER V2) was extracted from localized versions of Wikipedia covering 12 languages - Arabic is not included. The corpus was annotated with a NER taxonomy consisting of 6 coarse-grained types and 33 fine-grained subtypes (seven subtypes for Person, seven for Group, five for PROD, five for Creative Work, and five for Medical). Most participating systems outperformed the baselines by about 35% F_1.

There are a few Arabic NER corpora (Darwish et al., 2021), but all of them are coarse-grained. The ANERCorp corpus covers four entity types (Benajiba et al., 2007), CANERCorpus covers 14 religion-specific types (Salah and Zakaria, 2018), and Ontonotes covers 18 entities (Weischedel et al., 2013). The multilingual ACE2005 corpus (Walker et al., 2005), which includes Arabic, covers five coarse-grained entities and 35 fine-grained subtypes (3 subtypes for PERS, 11 for GPE, seven for LOC, nine for ORG, and five for FAC). Nevertheless, the ACE2005 corpus is costly and covers only one domain (media articles) that was collected 20 years ago. The most recent Arabic NER corpus is Wojood (Jarrar et al., 2022), which covers 21 nested entity types covering multiple domains. However, Wojood is a coarse-grained corpus and does not support entity subtypes.

To build on previous research on Arabic NER, we chose to extend the Wojood corpus with finer-grained subtypes. To ensure that our Wojood extension is compatible with other corpora, we chose to follow the ACE annotation guidelines.

3 $Wojood_{Fine}$ Corpus

$Wojood_{Fine}$ expands the annotation of the Wojood corpus (Jarrar et al., 2022), by adding fine-grain annotations for named-entity subtypes. Wojood is a NER corpus with 550K tokens annotated manually using 21 entity types. About 80% of Wojood was collected from MSA articles, while the 12% was collected from social media in Palestinian and Lebanese dialects (Curras and Baladi corpora (Haff et al., 2022; Jarrar et al., 2017, 2014)). One novelty of Wojood is its nested named entities, but some entity types can be ambiguous, which will affect downstream tasks such as information retrieval. For instance, the entity type "Organization" may refer to the government, educational institution, or a hospital to name a few. That is why $Wojood_{Fine}$ adds subtypes to four entity types: Geopolitical Entity (GPE), Organization (ORG), Location (LOC), and Facility (FAC). Table 3.3 shows the overall counts of the main four entity types in Wojood and $Wojood_{Fine}$. Note that creating $Wojood_{Fine}$ was not a straightforward process as it required revision of the Wojood annotation guidelines, which we discuss later in this section. As discussed in (Jarrar et al., 2022), Wojood is available as a RESTful web service, the data and the source-code are also made publicly available (Jarrar and Amayreh, 2019; Ghanem et al., 2023; Jarrar et al., 2019; Alhafi et al., 2019; Helou et al., 2016).

Tag	Wojood	$Wojood_{Fine}$
GPE	21,780	23,085
ORG	18,785	18,747
LOC	917	1,441
FAC	1,215	1,121
Total	**42,697**	**44,394**

Table 1: Frequency of the four entity types in Wojood and $Wojood_{Fine}$.

3.1 subtypes

All GPE, ORG, LOC and FAC tagged tokens in $Wojood_{Fine}$ corpus were annotated with the appropriate subtype based on the context, adding an additional 31 entity subtypes to $Wojood_{Fine}$. Throughout our annotation process, The LDC's ACE 2008 annotation guidelines for Arabic Entities V7.4.2 served as the basis for defining our annotation guidelines. Nevertheless, we added new tags (NEIGHBORHOOD, CAMP, SPORT, and ORG_FAC) to cover additional cases. Table 2 lists the frequency of each subtype in $Wojood_{Fine}$. Tables 7 and 8 in Appendix A present a brief explanation and examples of each subtype.

Tag	Sub-type Tag	Count
GPE	COUNTRY	8,205
	STATE-OR-PROVINCE	1,890
	TOWN	12,014
	NEIGHBORHOOD	119
	CAMP	838
	GPE_ORG	1,530
	SPORT	8
LOC	CONTINENT	214
	CLUSTER	303
	ADDRESS	0
	BOUNDARY	22
	CELESTIAL	4
	WATER-BODY	123
	LAND-REGION-NATURAL	259
	REGION-GENERAL	383
	REGION-INTERNATIONAL	110
ORG	GOV	8,325
	COM	611
	EDU	1,159
	ENT	3
	NONGOV	5,779
	MED	4,111
	REL	96
	SCI	146
	SPO	21
	ORG_FAC	114
FAC	PLANT	1
	AIRPORT	6
	BUILDING-OR-GROUNDS	1017
	SUBAREA-FACILITY	134
	PATH	76
Total		**47,621**

Table 2: Counts of each subtype entity in the corpus.

3.2 $Wojood_{Fine}$ Annotation Guideline

We followed ACE annotation guidelines to annotate the subtypes in $Wojood_{Fine}$. However, since $Wojood_{Fine}$ is based on Wojood, we found a discrepancy between Wojood and ACE guidelines. To address this issue in $Wojood_{Fine}$, we reviewed the annotations related to GPE, ORG, LOC and FAC to ensure compatibility with ACE guidelines. In this section, we highlight a number of the challenging annotation decisions we made in $Wojood_{Fine}$.

Country's governing body: in Wojood, country mentions were annotated as GPE and if the intended meaning of the country is a governing body then it is annotated as ORG. However, in $Wojood_{Fine}$, all ORG mentions that refer to the country's governing body are annotated as GPE with the subtype GPE_ORG. Figure 1 illustrates two examples to illustrate the difference between Wojood and $Wojood_{Fine}$ guidelines. According to Wojood, نيجيريا /Nigeria is tagged once as GPE and once as ORG, while in $Wojood_{Fine}$ both are GPE in the first level and in the second level one is tagged as Country and the other as GPE_ORG.

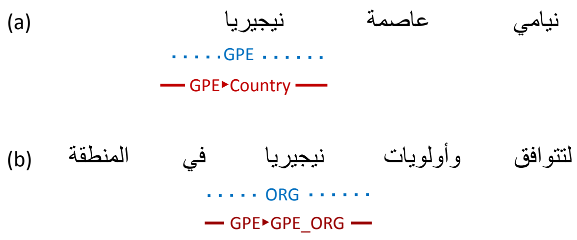


Figure 1: Two examples illustrating the difference between Wojood (in blue) and $Wojood_{Fine}$ guidelines (in red) for annotating GPEs.

Facility vs. organization: Wojood annotates buildings as FAC but if the intended meaning, in the context is an organization, then it is annotated as ORG. In $Wojood_{Fine}$, all mentions that refer to the facility's organization or social entity are annotated as ORG with the subtype ORG_FAC. Figure 2 illustrates an example of this case. Instead of annotating (مستشفى الشفاء /Al-Shifa Hospital) once as FAC and once as ORG, $Wojood_{Fine}$ tags it as ORG in the first level, and ORG_FAC in the second level.

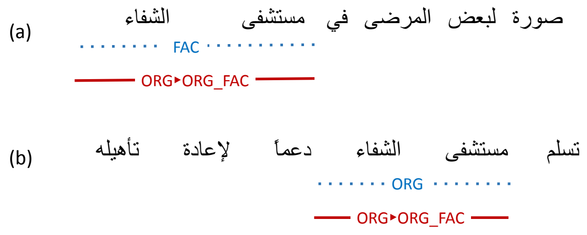


Figure 2: Two examples illustrating the difference between Wojood (in blue) and $Wojood_{Fine}$ (in red) guideline for annotating FAC vs. ORG.

Directions: Wojood does not include annotations for directions (east, west, south, and north). However, in $Wojood_{Fine}$ direction mentions are annotated as LOC with two subtypes: REGION-GENERAL if the location does not cross national borders, or REGION-INTERNATIONAL if the location crosses national borders. See the example in Figure 3.

In addition to the changes mentioned in this section, ACE guidelines considered any unit that is smaller-size than a village, like neighborhoods or camps, as LOC, while it is considered as GPE in Wojood guidelines. Continents are labaled as LOC in Wojood, while it is GPE in ACE. Both of these cases where corrected in $Wojood_{Fine}$.

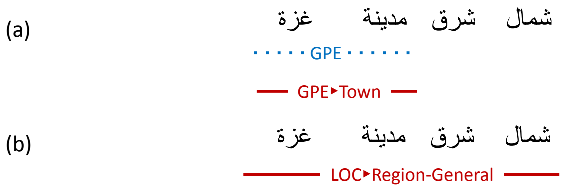


Figure 3: (a) The direction (شمال شرق مدينة غزة / north east Gaza city) is not annotated in Wojood, while in (b) it is annotated as LOC with Region-General as subtype in $Wojood_{Fine}$.

3.3 Annotation Process

The annotation process was done by one annotator, managed by NER expert, and was conducted over two phases:

Phase I: manually revise all annotations of GPE, ORG, LOC, and FAC in Wojood according to ACE guidelines, as discussed in section 3.2. Table 3.3 shows the counts of each of the four entity types in Wojood and $Wojood_{Fine}$.

Phase II: manually annotate the GPE, ORG, LOC, and FAC with subtypes. The annotator meticulously read each token in every sentence and classified the tokens into their respective subtypes. All critical and problematic tokens are reviewed by the NER expert.

Phase III: The NER expert reviewed all annotations marked in Phase I and Phase II in order to validate the entities that have been annotated.

Table 2 presents the counts of each entity subtype in the corpus, which shows 47,621 annotated entities in total.

3.4 Inter-Annotator Agreement

It has been shown that inter-annotator consistency significantly affects the quality of training data and, consequently, a NER system's ability to learn (Zhang, 2013). To measure the subtypes annotation quality and consistency, we recruited a second annotator to re-annotate 25,490 tokens (5.0% of the corpus) that were previously annotated by the first annotator. The sentences were selected randomly from the corpus while diversifying the sources and domains they were selected from. We then assessed the data quality and annotation consistency using the inter-annotator agreement (IAA), measured using Cohen's Kappa (κ) and F_1. The overall IAA was measured at $\kappa = 0.9861$ and $F_1 = 0.9889$.

Refer to Table 3 for the IAA for each subtype.

One can clearly observe that κ is high and that is for multiple reasons. First, we revised the annotations of the main four entity types (GPE, ORG, LOC and FAC) to better match ACE guideline. Second, once we verified the top level entity types, we started annotating the subtypes. Since the types and subtypes are hierarchically organized, that constraint the number of possible subtypes per token, leading to high IAA. Third, the NER expert gave a continuous feedback to the annotator and challenging entity mentions were discussed with the greater team.

As mentioned above, we calculated the IAA using both, Cohen's Kappa and F_1, for the subtypes of GPE, ORG, LOC and FAC tags. In what follows we explain Cohen's Kappa and F_1. Note that F_1 is not normally used for IAA, but it is an additional validation of the annotation quality.

3.4.1 Cohen's Kappa

To calculate Kappa for a given tag, we count the number of agreements and disagreements between annotators for a given subtype (such as GPE_COUNTRY). At the token level, agreements are counted as pairwise matches; thus, disagreements happen when a token is annotated by one annotator (e.g., as GPE_COUNTRY) and (e.g., as GPE_STATE-OR-PROVINCE) by another annotator. As such, Kappa is calculated by equation 1 (Eugenio and Glass, 2004).

$$\kappa = \frac{P_o - P_e}{1 - P_e} \tag{1}$$

where P_o represents the observed agreement between annotators and P_e represents the expected agreement, which is given by equation 2.

$$P_e = \frac{1}{N^2} \sum_T n_{T1} \times n_{T2} \tag{2}$$

where n_{Ti} is the number of tokens labeled with tag T by the ith annotator and N is the total number of annotated tokens.

3.4.2 F-Measure

For a given tag T, the F_1 is calculated according to equation 3. We only counted the tokens that at least one of the annotators had labeled with the T. We then conducted a pair-wise comparison. TP represents the true positives which is the number of agreements between annotators (i.e. number of tokens labeled GPE_TOWN by both annotators). If the first annotator disagrees with the second, it is counted as false negatives (FN), and if the second disagrees with the first, it is counted as false positives (FP), with a total of disagreement being $FN + FP$.

$$F_1 = \frac{2TP}{2TP + FN + FP} \tag{3}$$

Sub-Type Tag	Kappa	F1-Score
COUNTRY	0.9907	00.99
STATE-OR-PRONIVCE	0.9846	00.98
TOWN	0.9983	01.00
NEIGHBORHOOD	01.00	01.00
CAMP	01.00	01.00
GPE_ORG	0.9810	00.98
SPORT	01.00	01.00
CONTINENT	01.00	01.00
CLUSTER	0.9589	00.96
ADDRESS	-	-
BOUNDARY	01.00	01.00
CELESTIAL	-	-
WATER-BODY	01.00	01.00
LAND-REGION-NATURAL	0.9333	00.93
REGION-GENERAL	0.9589	00.96
REGION-INTERNATIONAL	0.9231	00.92
GOV	0.9760	00.98
COM	01.00	01.00
EDU	0.9807	00.98
ENT	-	-
NONGOV	0.9892	00.99
MED	01.00	01.00
REL	0.9630	00.96
SCI	01.00	00.10
SPO	01.00	01.00
ORG_FAC	01.00	01.00
PLANT	-	-
AIRPORT	-	-
BUILDING-OR-GROUNDS	01.00	01.00
SUBAREA-FACILITY	01.00	01.00
PATH	01.00	00.00
Overall	**0.9861**	**0.9889**

Table 3: Overall Kappa and F1-score for each sub-type.

4 Fine-Grained NER Modeling

4.1 Approach

For modeling, we have three tasks all performed on *Wojood*$_{Fine}$: **(1)** *Flat NER*, where for each token, we predict a single label from a set of 21 labels, **(2)** *Nested NER*, where we predict multiple labels picked from the 21 tags (i.e., multi-label classification) for each token and **(3)** *Nested with Subtypes NER*, this is also a multi-label task, where we ask the model to predict the main entity types and subtypes for each token from 52 total labels. We frame this as multi-task approach

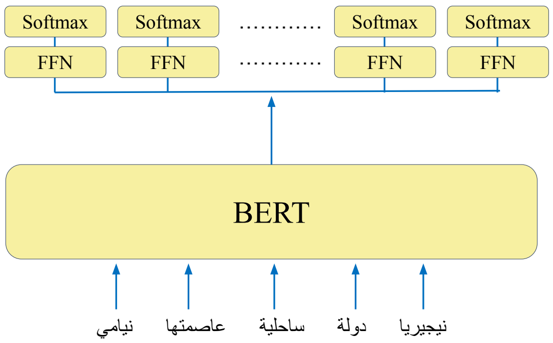


Figure 4: BERT refers to one of three pre-trained models we are using. For flat task, each softmax produce one class for each token, for other tasks each softmax is a set of softmax that produce multiple labels for each token.

Task	Model	Dev	Test
Flat	M1	$0.917^{\pm 0.00}$	$\mathbf{0.920}^{\pm 0.00}$
	M2	$0.910^{\pm 0.00}$	$0.913^{\pm 0.01}$
	M3	$0.902^{\pm 0.00}$	$0.907^{\pm 0.01}$
Nested	M1	$0.844^{\pm 0.02}$	$0.845^{\pm 0.01}$
	M2	$0.868^{\pm 0.02}$	$0.861^{\pm 0.02}$
	M3	$0.858^{\pm 0.02}$	$\mathbf{0.866}^{\pm 0.02}$
Nested +subtypes	M1	$0.836^{\pm 0.01}$	$0.837^{\pm 0.01}$
	M2	$0.880^{\pm 0.01}$	$0.883^{\pm 0.01}$
	M3	$0.883^{\pm 0.00}$	$\mathbf{0.885}^{\pm 0.00}$

Table 4: Results of fine-tuned models on the three different tasks. **M1**: ARBERTv2, **M2**: MARBERTv2 and **M3**: ARABERTv2. The results are represented as F1 averaged over 3 runs.

since we are learning both the nested labels *and* their subtypes jointly. In the multi-task case, each entity/subtype has its own classification layer, in the case of nested NER and nested with subtypes NER, the model consists of 21 and 52 classification layers, respectively. Since we use the IOB2 (Sang and Veenstra, 1999) tagging scheme, each linear layer is a multi-class classifier that outputs the probability distribution through softmax activation function for three classes, $C \in \{I, O, B\}$ (Jarrar et al., 2022). The model is trained with cross entropy loss objective computed for each linear layer separately, which are summed to compute the final cross entropy loss. All models are flat in the sense that we do not use any hierarchical architectures. However, future work can consider employing a hierarchical architecture where nested tokens are learnt first *then* their subtypes within the model. For all tasks, we fine-tune three encoder-based models for Arabic language understanding. Namely, we use ARBERTv2 and MARBERTv2 (Elmadany et al., 2023), which are both improved versions of ARBERT and MARBERT (Abdul-Mageed et al., 2021), respectively, that are trained on bigger datasets. The third model is ARABERTv2, which is an improved version of ARABERT (Antoun et al., 2021). It is also trained on a bigger dataset, with improved preprocessing. Figure 4 offers a simple visualization of our models' architecture.

4.2 Training Configuration

We split our dataset into three distinct parts for training (Train) 70%, validation (Dev) 10%, and blind testing (Test) 20%. We fine-tune all three models for 50 epochs each with an early stopping patience of 5 as identified on Dev. We use the AdamW optimizer (Loshchilov and Hutter, 2019), an exponential learning rate scheduler and a dropout of 0.1. The maximum sequence length is 512, the batch size, $B = 8$, and the learning rate, $\eta = 1e^{-5}$. For each model, we report an average of three runs (each time with a different seed). We report in F_1 along with the standard deviation from the three runs, on both Dev and Test, for each model. All models are implemented using PyTorch, Huggingface Transformers, and a custom version of the Wojood open-source code[1].

4.3 Results

We show the results of our three fine-tuned models across each of the three tasks in Table 4. We briefly highlight these results in the following:

Flat NER. The three fine-tuned models achieve comparable results on the Flat NER task, with ARBERTv2 scoring slightly better on both the Dev and Test sets. ARBERTv2 achieves an F_1 of 92% on the Test set, while ARBERTv2 and ARABERTv2 achieves 91.3% and 90.3%, respectively.

Nested NER. ARABERTv2 slightly outperforms other pre-trained models with a small margin, on Dev and Test. On Test, it scores 86.6%.

Nested NER with Subtypes. Here, ARABERTv2 achieves the highest score ($88.5\% F_1$).

5 Analysis

For all tasks, all models almost always converge in the first 10 epochs. For all models, there is a positive correlation between performance and the number of training samples. For example, for classes represented well in the training set (e.g.,

[1] https://github.com/SinaLab/ArabicNER

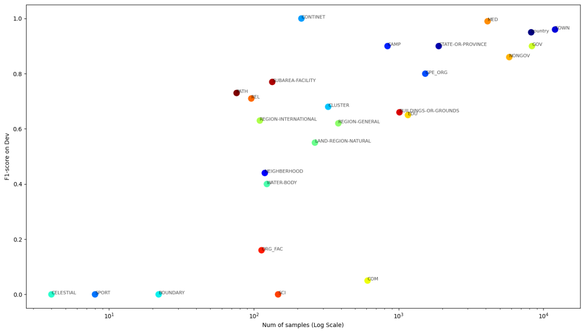


Figure 5: Number of samples vs. F_1 in each subtype class on Subtype classification task.

Task	Model	Finance	Science	Politics
Flat	M1	63.7% $^{\pm 0.01}$	$0.670^{\pm 0.02}$	$\mathbf{0.747}^{\pm 0.02}$
	M2	$0.573^{\pm 0.01}$	$\mathbf{0.677}^{\pm 0.02}$	$0.717^{\pm 0.01}$
	M3	$\mathbf{0.643}^{\pm 0.01}$	$0.670^{\pm 0.02}$	$0.723^{\pm 0.01}$
Nested	M1	$0.458^{\pm 0.01}$	$0.494^{\pm 0.02}$	$0.557^{\pm 0.00}$
	M2	$0.499^{\pm 0.05}$	$0.554^{\pm 0.00}$	$0.612^{\pm 0.01}$
	M3	$\mathbf{0.563}^{\pm 0.02}$	$\mathbf{0.583}^{\pm 0.02}$	$\mathbf{0.629}^{\pm 0.03}$
Nested +subtypes	M1	$0.449^{\pm 0.07}$	$0.493^{\pm 0.02}$	$0.497^{\pm 0.01}$
	M2	$0.504^{\pm 0.03}$	$0.544^{\pm 0.06}$	$0.575^{\pm 0.02}$
	M3	$\mathbf{0.553}^{\pm 0.04}$	$\mathbf{0.545}^{\pm 0.02}$	$\mathbf{0.593}^{\pm 0.08}$

Table 5: Results of fine-tuned models on the three new domains, Finance, Science, and Politics. **M1**: MARBERTv2, **M2**: ARBERTv2 and **M3**: ARABERTv2. The results are represented as F1 averaged over 3 runs.

COUNTRY, TOWN and GOV), models perform at 0.90 F_1 or above.

The inverse is also true, with poor performance on classes such as SPORT, BOUNDARY and CELESTIAL. There are also some nuances. For example, we can see that the best model is struggling with the COM subtype class even though the model has scored good results with classes with fewer samples such as CLUSTER. The main reason for this is that types such as CLUSTER are a closed set of classes (e.g., "European Union", "African Union") where the model can easily memorize them, while the COM refers to an infinite group of commercial entities, that can not be limited. Figure 5 is a plot of the number of samples in training data (X-axis) vs. performance (Y-axis) that clearly shows the general pattern of good performance positively correlating with the number of training samples.

5.1 Out-of-Domain Performance

To assess the generalization capability of our models, we conducted an evaluation on three unseen domains and different time periods. Three corpora were collected, each covering a distinct domain: finance, science, and politics. These corpora were compiled from Aljazeera news articles published in 2023. Manual annotation of the three corpora was performed in accordance with the same annotation guidelines established for $Wojood_{Fine}$. We apply the three versions of each of our three models trained on $Wojood_{Fine}$ original training data (described in Section 4.2) on the new domains, for each of the three NER tasks. We present results for this out-of-domain set of experiments in Table 5. We observe that performance drastically drops on all three new domains, for all models on all tasks. This is not surprising, as challenges related to domain generalization are well-known in the literature. Our results here, however, allow us to quantify the extent to which model performance degrades on each of these three new domains. In particular, models do much better on the politics domain than they perform on finance or science. This is the case since our training data are collected from online articles involving news and much less content from financial or scientific sources. Figure 6 shows some examples for new mentions from those domains that have not been seen in $Wojood_{Fine}$.

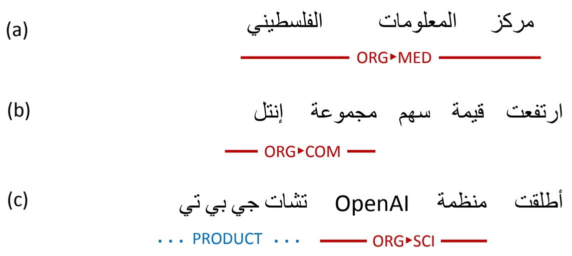


Figure 6: Some mentions from the three new domains that have not previously appeared in $Wojood_{Fine}$. (a) (مركز المعلومات الفلسطيني) in Politics domain, (b) (مجموعة إنتل) in Finance domain, (c) (منظمة OpenAI) in Science domain.

5.2 Error Analysis

In order to understand the errors made by the model, we conduct a human error analysis on the errors generated by ARABERTv2 (i.e, best model on this task) on the first 2K tokens of the Dev set of Nested NER with Subtypes task. We find that the model's errors can be categorized into six major error classes: **(1)** *wrong tag*, where the model predicts a different tag, **(2)** *no prediction*, where the model does not produce any tag (i.e. predict O), **(3)** *missing subtype*, the model succeeds in predicting parent tag but fails to predict the subtype,

Example	Gold	Predicted	Error Type
أنا ازا هاجرت ع أي مكان رح أَخد الشلة If I ever migrated somewhere, I'd take the group	O	GPE|TWN	msa_dia_confusion
مشهد ٣ فتاة جالسة و خلفها العلم الأمريكي. Scene 3: a girl sitting with the American flag behind her.	CRDNAL	ORDNAL	ordinal_vs_cardinal
جدار الفصل العنصري مستعمرة بزغات زئيف. The racial separation wall, colony of Bazgat Ze'ev.	LOC|NEIGHB	NEIGHB	Missing_parent_tag
بتنتخب رئيس جمهورية و رئيس مجلس نواب The president of the republic and the speaker of the council of deputies are elected.	OCC|ORG|GOV	OCC|ORG	missing_subtype
صحيح الساعة خمسة حسب اعلانهم It's true, it's five o'clock according to their announcement.	TIME	CRDNL	wrong_tag
العلما اللغة التانية بتنحصر للاستخدام اليومي. Scientists: the second language is limited to daily use.	B-ORDNL	O	no_prediction

Table 6: Examples of error categories made by our best model (ARABERTv2) on our Dev set. We provide the translation to English of each sample.

(4) *missing parent tag*: the model succeeds in predicting subtype tag but fails to predict the parent tag, **(5)** *MSA vs. DIA confusion*, the model makes a wrong prediction due to confusion between MSA and Dialect, and **(6)** *ordinal vs. cardinal*, in this class, the model assigns cardinal to an ordinal class. Figure 7 shows the distribution of different errors present in the Dev set, with the *wrong tag* being the major source of errors followed by *no prediction* error. A further breakdown of the *wrong tag* error class shows that 14.3% are due to usage of dialectal words, a similar proportion are due to nested entities. Table 6 shows an example of each error class.

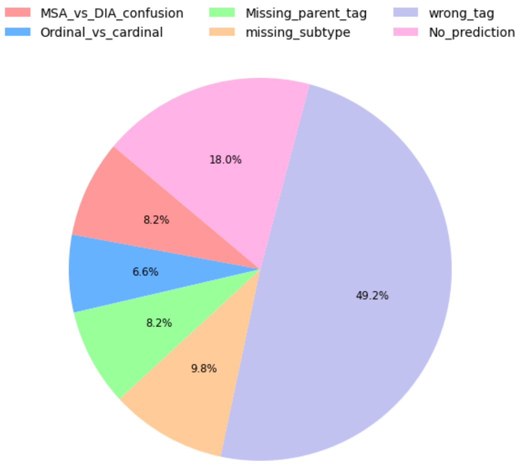


Figure 7: Distribution of error classes in nested with subtypes task on our Dev set.

6 Conclusion and Future Work

We presented *Wojood*$_{Fine}$, an extension to the Wojood NER corpus with subtypes for the GPE, LOC, ORG, and FAC. *Wojood*$_{Fine}$ corpus is the first fine-grain corpus for MSA and dialectal Arabic with nested and subtyped NER. The GPE, ORG, FAC and LOC tags form more than 44K tokens of the corpus, which was manually annotated using subtypes entities. Our inter-annotator agreement IAA evaluation of *Wojood*$_{Fine}$ annotations achieved high levels of agreement among the annotators. The achieved evaluations are 0.9861 Kappa and 0.9889 F_1.

We also fine-tune three pre-trained models ARBERTv2, MARBERTv2 and ARABERTv2 and tested their performance on different settings of *Wojood*$_{Fine}$. We find that ARABERTv2 achieved the best performance on Nested and Nested with Subtypes tasks. In the future, we plan to test pre-trained models on nested subtypes with hierarchical architecture. We also plan to link named entities with concepts in the Arabic Ontology (Jarrar, 2021, 2011) to enable a richer semantic understanding of text. Additionally, we will extend the *Wojood*$_{Fine}$ corpus to include more dialects, especially the Syrian Nabra dialects (Nayouf et al., 2023) as well as the four dialects in the Lisan (Jarrar et al., 2023b) corpus.

Acknowledgment

We would like to thank Sana Ghanem for her invaluable assistance in reviewing and improving the annotations, as well as for her support in the IAA calculations. The authors would also like to thank Tymaa Hammouda for her technical support and

expertise in the data engineering of the corpus.

Limitations

A number of considerations related to limitations and ethics are relevant to our work, as follows:

- **Intended Use.** Our models perform named entity recognition at a fine-grained level and can be used for a wide range of information extraction tasks. As we have shown, however, even though the models are trained with data acquired from several domains, their performance drops on data with distribution different than our training data such as the finance or science domains. We suggest this be taken into account in any application of the models.

- **Annotation Guidelines and Process.** Some of the entities are difficult to tag. Even though annotators have done their best and we report high inter-annotator reliability, the application of our guidelines may need to be adapted before application to new domains.

Ethics Statement

We trained our models on publicly available data, thus we do not have any particular concerns about privacy.

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.

Moustafa Al-Hajj and Mustafa Jarrar. 2021. Arabglossbert: 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 40–48, Online. INCOMA Ltd.

Diana Alhafi, Anton Deik, and Mustafa Jarrar. 2019. Usability evaluation of lexicographic e-services. In *The 2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA)*, pages 1–7. IEE.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2021. Arabert: Transformer-based model for arabic language understanding.

Yassine Benajiba, Paolo Rosso, and José Miguel Benedi Ruiz. 2007. Anersys: An arabic named entity recognition system based on maximum entropy. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 4394 LNCS.

Nancy Chinchor and Patricia Robinson. 1997. Muc-7 named entity task definition. In *Proceedings of the 7th Conference on Message Understanding*, volume 29, pages 1–21.

Kareem Darwish, Nizar Habash, Mourad Abbas, Hend Al-Khalifa, Huseein T. Al-Natsheh, Houda Bouamor, Karim Bouzoubaa, Violetta Cavalli-Sforza, Samhaa R. El-Beltagy, Wassim El-Hajj, Mustafa Jarrar, and Hamdy Mubarak. 2021. A panoramic survey of natural language processing in the arab worlds. *Commun. ACM*, 64(4):72–81.

Bart Desmet and Véronique Hoste. 2013. Fine-grained dutch named entity recognition. *Language Resources and Evaluation*, 48:307–343.

Maud Ehrmann, Matteo Romanello, Alex Flückiger, and Simon Clematide. 2020a. Extended overview of clef hipe 2020: named entity processing on historical newspapers. In *CLEF 2020 Working Notes. Conference and Labs of the Evaluation Forum*, volume 2696. CEUR-WS.

Maud Ehrmann, Camille Watter, Matteo Romanello, Clematide Simon, and Alex Flückiger. 2020b. Impresso named entity annotation guidelines (clef-hipe-2020). Technical report.

AbdelRahim Elmadany, El Moatez Billah Nagoudi, and Muhammad Abdul-Mageed. 2023. Orca: A challenging benchmark for arabic language understanding.

Barbara Di Eugenio and Michael Glass. 2004. The Kappa Statistic: A Second Look. *Computational Linguistics*, 30(1):95–101.

Besnik Fetahu, Sudipta Kar, Zhiyu Chen, Oleg Rokhlenko, and Shervin Malmasi. 2023. Semeval-2023 task 2: Fine-grained multilingual named entity recognition (multiconer 2). *arXiv preprint arXiv:2305.06586*.

Sana Ghanem, Mustafa Jarrar, Radi Jarrar, and Ibrahim Bounhas. 2023. A benchmark and scoring algorithm for enriching arabic synonyms. In *Proceedings of the 12th International Global Wordnet Conference (GWC2023)*, pages 215–222. Global Wordnet Association.

Karim El Haff, Mustafa Jarrar, Tymaa Hammouda, and Fadi Zaraket. 2022. Curras + baladi: Towards a levantine corpus. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2022)*, Marseille, France.

Ahmed Hamdi, Elvys Linhares Pontes, Emanuela Boros, Thi Tuyet Hai Nguyen, Günter Hackl, Jose G Moreno, and Antoine Doucet. 2021. A multilingual dataset for named entity recognition, entity linking and stance detection in historical newspapers. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2328–2334.

Mamoun Abu Helou, Matteo Palmonari, and Mustafa Jarrar. 2016. Effectiveness of automatic translations for cross-lingual ontology mapping. *Journal of Artificial Intelligence Research*, 55(1):165–208.

Alankar Jain, Bhargavi Paranjape, and Zachary C Lipton. 2019. Entity projection via machine translation for cross-lingual ner. *arXiv preprint arXiv:1909.05356*.

Mustafa Jarrar. 2011. Building a formal arabic ontology (invited paper). In *Proceedings of the Experts Meeting on Arabic Ontologies and Semantic Networks*. ALECSO, Arab League.

Mustafa Jarrar. 2021. The arabic ontology - an arabic wordnet with ontologically clean content. *Applied Ontology Journal*, 16(1):1–26.

Mustafa Jarrar and Hamzeh Amayreh. 2019. An arabic-multilingual database with a lexicographic search engine. In *The 24th International Conference on Applications of Natural Language to Information Systems (NLDB 2019)*, volume 11608 of *LNCS*, pages 234–246. Springer.

Mustafa Jarrar, Hamzeh Amayreh, and John P. McCrae. 2019. Representing arabic lexicons in lemon - a preliminary study. In *The 2nd Conference on Language, Data and Knowledge (LDK 2019)*, volume 2402, pages 29–33. CEUR Workshop Proceedings.

Mustafa Jarrar, Anton Deik, and Bilal Faraj. 2011. Ontology-based data and process governance framework -the case of e-government interoperability in palestine. In *Proceedings of the IFIP International Symposium on Data-Driven Process Discovery and Analysis (SIMPDA'11)*, pages 83–98.

Mustafa Jarrar, Nizar Habash, Diyam Akra, and Nasser Zalmout. 2014. Building a corpus for palestinian arabic: a preliminary study. In *Proceedings of the EMNLP 2014, Workshop on Arabic Natural Language*, pages 18–27. Association For Computational Linguistics.

Mustafa Jarrar, Nizar Habash, Faeq Alrimawi, Diyam Akra, and Nasser Zalmout. 2017. Curras: An annotated corpus for the palestinian arabic dialect. *Journal Language Resources and Evaluation*, 51(3):745–775.

Mustafa Jarrar, Mohammed Khalilia, and Sana Ghanem. 2022. Wojood: Nested arabic named entity corpus and recognition using bert. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2022)*, Marseille, France.

Mustafa Jarrar, Sanad Malaysha, Tymaa Hammouda, and Mohammad Khalilia. 2023a. Salma: Arabic sense-annotated corpus and wsd benchmarks. In *Proceedings of the 1st Arabic Natural Language Processing Conference (ArabicNLP), Part of the EMNLP 2023*. ACL.

Mustafa Jarrar, Fadi Zaraket, Tymaa Hammouda, Daanish Masood Alavi, and Martin Waahlisch. 2023b. Lisan: Yemeni, irqi, libyan, and sudanese arabic dialect copora with morphological annotations. In *The 20th IEEE/ACS International Conference on Computer Systems and Applications (AICCSA)*. IEEE.

Ridong Jiang, Rafael E. Banchs, and Haizhou Li. 2016. Evaluating and combining name entity recognition systems. In *Proceedings of the Sixth Named Entity Workshop*, pages 21–27, Berlin, Germany. Association for Computational Linguistics.

Diksha Khurana, Aditya Koli, Kiran Khatter, and Sukhdev Singh. 2022. Natural language processing: State of the art, current trends and challenges. *Multimedia Tools and Applications*, 82.

Changki Lee, Yi-Gyu Hwang, Hyo-Jung Oh, Soojong Lim, Jeong Heo, Chung-Hee Lee, Hyeon-Jin Kim, Ji-Hyun Wang, and Myung-Gil Jang. 2006. Fine-grained named entity recognition using conditional random fields for question answering. In *Information Retrieval Technology*, pages 581–587, Berlin, Heidelberg. Springer Berlin Heidelberg.

Jing li, Aixin Sun, Ray Han, and Chenliang Li. 2020. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, PP:1–1.

Xiao Ling and Daniel Weld. 2012. Fine-grained entity recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 26, pages 94–100.

Bang Liu, Haojie Wei, Di Niu, Haolan Chen, and Yancheng He. 2020. Asking questions the human way: Scalable question-answer generation from text corpus. In *Proceedings of The Web Conference 2020*, pages 2032–2043.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization.

Amal Nayouf, Mustafa Jarrar, Fadi zaraket, Tymaa Hammouda, and Mohamad-Bassam Kurdy. 2023. Nâbra: Syrian arabic dialects with morphological annotations. In *Proceedings of the 1st Arabic Natural Language Processing Conference (ArabicNLP), Part of the EMNLP 2023*. ACL.

Ramzi Esmail Salah and Lailatul Qadri Binti Zakaria. 2018. Building the classical arabic named entity recognition corpus (canercorpus). *Journal of Theoretical and Applied Information Technology*, 96.

Erik F Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*.

Erik F. Tjong Kim Sang and Jorn Veenstra. 1999. Representing text chunks. In *Proceedings of the Ninth Conference on European Chapter of the Association for Computational Linguistics*, EACL '99, page 173–179, USA. Association for Computational Linguistics.

Rodney L Summerscales, Shlomo Argamon, Shangda Bai, Jordan Hupert, and Alan Schwartz. 2011. Automatic summarization of results from clinical trials. In *2011 IEEE International Conference on Bioinformatics and Biomedicine*, pages 372–377. IEEE.

Mikhail Tikhomirov, N. Loukachevitch, Anastasiia Sirotina, and Boris Dobrov. 2020. Using bert and augmentation in named entity recognition for cybersecurity domain. In *Natural Language Processing and Information Systems*, pages 16–24, Cham. Springer International Publishing.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2005. Ace 2005 multilingual training corpus-linguistic data consortium. *URL: https://catalog. ldc. upenn. edu/LDC2006T06*.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. 2013. Ontonotes release 5.0 ldc2013t19. Technical report, Linguistic Data Consortium.

Ziqi Zhang. 2013. Named entity recognition : challenges in document annotation, gazetteer construction and disambiguation.

Huiming Zhu, Chunhui He, Yang Fang, and Weidong Xiao. 2020. Fine grained named entity recognition via seq2seq framework. *IEEE Access*, 8:53953–53961.

A subtypes and Inter-Annotator Agreement

This is an appendix contains $Wojood_{Fine}$ subtype descriptions and detailed IAA.

Tag	Sub-type Tag	Short Description
GPE	COUNTRY	Taggable mentions of the entireties of any nation. فلسطين، مصر، الولايات المتحدة، لبنان.
	STATE-OR-PRONIVCE	Taggable mentions of the entireties of any state, province, or canton of any nation. إقليم كردستان، لواء نابلس.محافظة القاهرة، قطاع غزة،
	TOWN	Taggable mentions of any GPE entireties below the level of State-or-Province, including cities, and villages. قرية بيرزيت. العاصمة دبي،
	NEIGHBORHOOD	Taggable mentions of the entireties of units that are smaller than villages. حي الطيرة، البلدة القديمة، حي المغاربة.
	CAMP	Taggable mentions of the entireties of units that are smaller than villages, relating to refugees. مخيم قلنديا، مخيم نور شمس.
	GPE_ORG	is used for GPE mentions that refer to the entire governing body of a GPE. قررت فلسطين إعفاء المتضررين.أصدرت الولايات المتحدة تقريرها،
	SPORT	Athletes, Sports Teams. برشلونة، ميلان. مباراة المغرب، الفرق الرياضية.
LOC	CONTINENT	Taggable mentions of the entireties of any of the seven continents. أوروبا، آسيا.
	CLUSTER	Named groupings of GPEs that can function as political entities. أوروبا الشرقية، الشرق الأوسط.
	ADDRESS	A location denoted as a point such as in a postal system ("31° S, 22° W"). ١٧، شارع فؤاد.
	BOUNDARY	A one-dimensional location such as a border between GPE's or other locations. الحدود الشرقية، الحدود السورية التركية.
	CELESTIAL	world, earth, globe in addition to all other planets. المريخ، عطارد.
	WATER-BODY	Bodies of water, natural or artificial (man-made). البحر الأحمر، الأطلسي.
	LAND-REGION-NATURAL	Geologically or ecosystemically designated, non-artificial locations. جبال الألب، الأغوار، السهول.
	REGION-GENERAL	Taggable locations that do not cross national borders. شمال الضفة الغربية، شرق سوريا.
	REGION-INTERNATIONAL	Taggable locations that cross national borders. آسيا الكبرى، جنوب أفريقيا.

Table 7: Parent type and description of each sub-type in $Wojood_{Fine}$

Tag	Sub-type Tag	Short Description
ORG	GOV	Government organizations. سفارة، محكمة، وزارة، شرطة.
	COM	A commercial organization that is focused primarily upon providing ideas, products, or services for profit. بنك ، شركة ،مؤسسة ربحية.
	EDU	An educational organization that is focused primarily upon the furthering or promulgation of learning/education. جامعة ، مدرسة ، معهد.
	ENT	Entertainment organizations whose primary activity is entertainment. فرقة ميامي، مسرح الحكواتي.
	NONGOV	Non-governmental organizations that are not a part of a government or commercial organization and whose main role is advocacy, charity or politics (in a broad sense). نقابة العاملين، الأمم المتحدة، الأحزاب السياسية ،أطباء بلا حدود.
	MED	Media organizations whose primary interest is the distribution of news or publications. جريدة الشرق، مجلة الحياة.
	REL	Religious organizations that are primarily devoted to issues of religious worship. الأوقاف ، الأزهر.
	SCI	Medical-Science organizations whose primary activity is the application of medical care or the pursuit of scientific research. مستشفى هداسا ،معهد الدراسات النووية.
	SPO	Sports organizations that are primarily concerned with participating in or governing organized sporting events. الاتحاد السعودي لكرة القدم،لجنة الفلبين الأولومبية.
	ORG_FAC	Facilities that have an organizational, legal or social representative مظاهرات أمام بنك روما.
FAC	PLANT	One or more buildings that are used and/or designed solely for industrial purposes: manufacturing, power generation, etc. مصنع.
	AIRPORT	A facility whose primary use is as an airport. مطار.
	BUILDING-OR-GROUNDS	Man-made/-maintained buildings, outdoor spaces, and other such facilities. منزل، مبنى، مستشفى، معبر.
	SUBAREA-FACILITY	Taggable portions of facilities. غرفة ،زنزانة.
	PATH	Streets, canals, and bridges. الشوارع الرئيسية، الخطوط الهاتفية، الحواجز.

Table 8: Parent type and description of each sub-type in $Wojood_{Fine}$

Sub-type Tag	TP	FN	FP	Kappa	F1-Score
COUNTRY	643	5	7	0.9907	00.99
STATE-OR-PRONIVCE	96	3	0	0.9846	00.98
TOWN	295	0	1	0.9983	01.00
NEIGHBORHOOD	23	0	0	01.00	01.00
CAMP	92	0	0	01.00	01.00
GPE_ORG	129	3	2	0.9810	00.98
SPORT	2	0	0	01.00	01.00
CONTINENT	7	0	0	01.00	01.00
CLUSTER	35	3	0	0.9589	00.96
ADDRESS	-	-	-	-	-
BOUNDARY	11	0	0	01.00	01.00
CELESTIAL	-	-	-	-	-
WATER-BODY	5	0	0	01.00	01.00
LAND-REGION-NATURAL	14	0	2	0.9333	00.93
REGION-GENERAL	70	2	4	0.9589	00.96
REGION-INTERNATIONAL	6	0	1	0.9231	00.92
GOV	490	6	18	0.9760	00.98
COM	21	0	0	01.00	01.00
EDU	153	0	6	0.9807	00.98
ENT	-	-	-	-	-
NONGOV	599	11	2	0.9892	00.99
MED	630	0	0	01.00	01.00
REL	26	2	0	0.9630	00.96
SCI	4	0	0	01.00	00.10
SPO	2	0	0	01.00	01.00
ORG_FAC	15	0	0	01.00	01.00
PLANT	-	-	-	-	-
AIRPORT	-	-	-	-	-
BUILDING-OR-GROUNDS	64	0	0	01.00	01.00
SUBAREA-FACILITY	48	0	0	01.00	01.00
PATH	2	0	0	01.00	01.00
Overall	**3,482** count	**35** count	**43** count	**0.9861** macro	**0.9889** micro

Table 9: Overall IAA for each sub-type, reported using Kappa and F_1.

Investigating Zero-shot Cross-lingual Language Understanding for Arabic

Zaid Alyafeai*
Department of Computer Science
King Fahd University of Petroleum
and Minerals
Dhahran, Saudi Arabia
g201080740@kfupm.edu.sa

Moataz Ahmed
Department of Computer Science
King Fahd University of Petroleum
and Minerals
Dhahran, Saudi Arabia
moataz@kfupm.edu.sa

Abstract

Numerous languages exhibit shared characteristics, especially in morphological features. For instance, Arabic and Russian both belong to the fusional language category. The question arises: Do such common traits influence language comprehension across diverse linguistic backgrounds? This study explores the possibility of transferring comprehension skills across languages to Arabic in a zero-shot scenario. Specifically, we demonstrate that training language models on other languages can enhance comprehension of Arabic, as evidenced by our evaluations in three key tasks: natural language inference, question answering, and named entity recognition. Our experiments reveal that certain morphologically rich languages (MRLs), such as Russian, display similarities to Arabic when assessed in a zero-shot context, particularly in tasks like question answering and natural language inference. However, this similarity is less pronounced in tasks like named entity recognition.

1 Introduction

Language models have been mainly utilized by training on a large corpus using a monolingual approach i.e. on a single language like BERT (Devlin et al., 2018), GPT (Radford et al., 2018), and Roberta (Liu et al., 2019). On the other hand, there were some attempts to train such language models on multiple languages like multilingual BERT (mBERT) (Devlin et al., 2018) by combining text from different languages. A tokenizer such as WordPiece, is trained on the joined text from different languages to be able to recognize the scripts from such languages. This makes the vocabulary size of such models huge. For example, mBERT has a shared vocabulary size of 110K across the 104 languages that were used for training compared to the 30K vocabulary size that was used to train the monolingual BERT. With such a huge vocabulary and the number of languages, it is not clear how knowledge or language understanding is shared across such languages. More importantly, it is important to investigate how such knowledge is shared among similar languages, especially in terms of morphological features. We mainly focus on languages that exhibit rich morphology like Arabic. In this study, our primary objective is to explore the integration of knowledge into Arabic by fine-tuning mBERT across various tasks, including question answering, natural language inference, and named entity recognition in multiple languages, followed by a zero-shot evaluation specifically on Arabic.

This paper is organized as follows. In Section 2, we discuss the related studies to our work. In Section 3, we focus on discussing the scope of our work. In Sections 4 and 5, we discuss morphology in general and how it's an intrinsic property of Arabic. In Section 6, we investigate mBERT and why it's an important model to evaluate such properties on. In Section 7, we detail the datasets and tasks used for evaluating our study. Finally, in Section 8, we detail our experiments and discuss our results.

2 Related Work

Multilinguality focuses on training language models with shared vocabulary for multiple languages. Over the past few years, many models have adopted this strategy like multi-lingual BERT (mBERT) for 104 languages (Devlin et al., 2018), XLM-R for 100 languages (Conneau et al., 2019), and mT5 for 101 languages (Xue et al., 2020). The advantage of using such models is the simplicity of creating a shared vocabulary using a uniform linear mapping between the different multilingual embeddings. Interestingly, mBERT demonstrates proficiency in zero-shot cross-lingual model transfer, as observed in prior research (Pires et al., 2019). This capability aids in comprehending a given language in a universal context. However such models are required

*corresponding author

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 324–334
December 7, 2023 ©2023 Association for Computational Linguistics

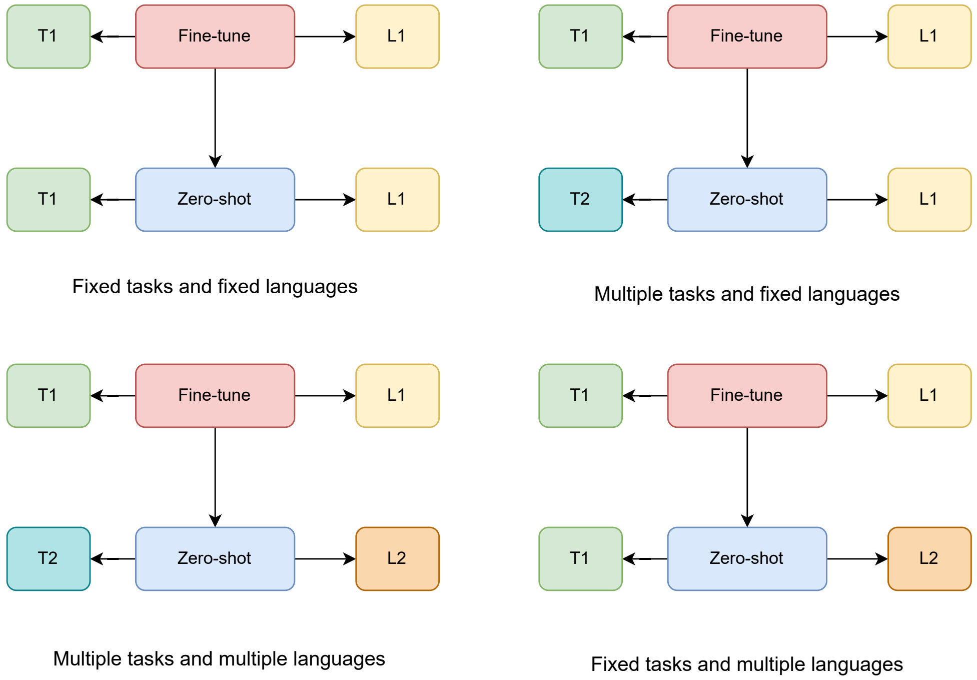

T1
Fine-tune
L1
T1
Zero-shot
L1
Fixed tasks and fixed languages
T1
Fine-tune
L1
T2
Zero-shot
L1
Multiple tasks and fixed languages
T1
Fine-tune
L1
T2
Zero-shot
L2
Multiple tasks and multiple languages
T1
Fine-tune
L1
T1
Zero-shot
L2
Fixed tasks and multiple languages

Figure 1: Different approaches for zero-shot evaluation with variations in tasks and languages. In each figure, we show the tasks and languages used for fine-tuning and the tasks and languages used for zero-shot evaluation. In this study, we focus on the approach of fixed tasks and multiple languages.

to be trained on a multilingual objective in order to generalize more for distant languages with different typography (Lauscher et al., 2020). Not to mention how to transfer knowledge to low-resource languages. Lately, there has been growing interest in utilizing more sophisticated architectures to enhance knowledge optimization in low-resource scenarios. One of the most interesting approaches is using adapter modules to avoid catastrophic forgetting [1] when training multilingual models on different languages. (Pfeiffer et al., 2020) focused on creating a framework for multi-task adapter-based cross-lingual transfer. (Hu et al., 2020) created a benchmark of the evaluation of cross-lingual transfer for 40 languages XTREME.

In the literature, there were some limited efforts to apply zero-shot understanding for Arabic. (Khalifa et al., 2021) used Self-Training of pre-trained language models for zero- and few-shot multi-dialectal Arabic sequence labeling by first fine-tuning on modern standard Arabic (MSA). Some studies focused on applying these techniques for Arabic like GigaBERT which can achieve bilingual zero-shot understanding from English to Arabic (Lan et al., 2020). They apply these methods for information extraction tasks (IE) like part of speech tagging (POS), named entity recognition (NER), relation extraction (RE), and argument role labeling (ARL) tasks. (Abboud et al., 2022) studied cross-lingual understanding from English and French to Arabic. They show strong performance in a zero-shot setting despite the differences between the source and target languages in terms of morphology and grammar. There were many efforts also to benchmark ChatGPT models in a zero-shot fashion on multiple tasks for Arabic without fine-tuning (Kadaoui et al., 2023), (Alyafeai et al., 2023), (Khondaker et al., 2023), and (Abdelali et al., 2023). Models like ChatGPT which was trained on a large mixture of scripts for hundreds of languages were able to attain strong performance on multiple tasks in a zero-shot fashion.

3 Zero-shot Evaluation

The default approach of evaluating a language model on a given task is by training the model on that dataset and then evaluating on the unseen split of the dataset. We assume that both training and test splits belong to the same language/task. However, we can also argue that we can train the language model on a given task say `T1` then evaluate on another task, say, `T2`. Similarly, we can

[1] Happens when the weights are fine-tuned on new datasets. The models usually forget the previous knowledge.

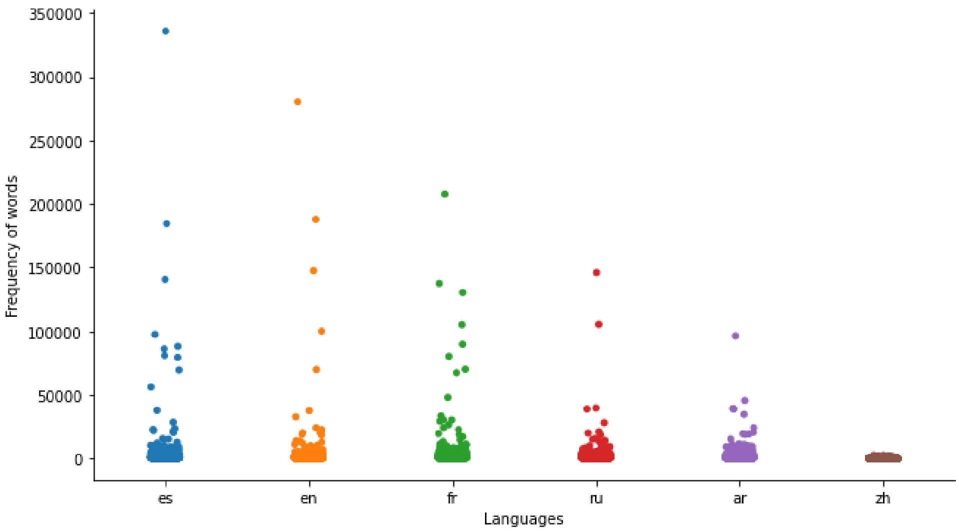


Figure 2: Comparing languages in terms of the frequency of the top 1000 words in the corpus.

train a language model on a given language L1 and evaluate on another language L2 (see Figure 1). However, in order to do that, the language model has to be able to predict or generate tokens in that language. Hence, multilingual models have been utilized to evaluate cross-lingual understanding. Such language models like mBERT (discussed in Section 6) are trained on a corpus that contains multiple languages. As a result, we can hypothesize that such language models have attained some kind of relationship across different languages either in terms of script or topology. As an example, Arabic and Persian have the same script and share many common words. More interestingly, using zero-shot evaluation we can test whether a given language is closer to other languages in terms of more complex features like morphology. For example, both Arabic and Russian are rich in terms of morphology and they are both inflectional languages. In this paper, we mainly focus on zero-shot evaluation on the same set of tasks but in different languages. To summarize, given a language L1 we train it on a given task T1 and zero-shot evaluate on L2 on the same task T1. In this paper, L2 is Arabic, and L1 could be any language.

4 Morphology (Arabic and Beyond)

Tackling morphology is a very important step toward improving language modeling for languages like Arabic. In the literature, (Antoun et al., 2020) showed slightly better results by pre-splitting words using the Farasa segmentation tool (Abdelali et al., 2016) on multiple tasks. The morphological segmentation results in better performance in text classification tasks while worse results in question answering and named entity recognition tasks. Similarly, (Oudah et al., 2019) showed that we can get some improvement when we employ different morphological analyzers on top of neural and statistical models for machine translation. (AlKhamissi et al., 2020) showed that by utilizing a combination of character- and word-level representations they achieved better results on the diacritization task. (Alkaoud and Syed, 2020) modified the tokenization algorithm for multilingual BERT to achieve better results than monolingual BERT on two different datasets.

Morphology also exists in other languages but with different levels of complexity depending on the language as shown in Table 1. (Hofmann et al., 2020) modified BERT for generating derivationally more complicated English words using masked language modeling objective. The conditioned language model on the word can predict the prefixes and suffixes of that masked word. (Sennrich et al., 2015) compared Byte-Pair Encoding (BPE) and unigram language model [2] (Kudo, 2018) for the translation between low-resource morphologically rich languages (Turkish and Swahili) into English (Richburg et al., 2020). They showed an improvement in using unigram language models. Similarly, Bostrom and Durrett showed an improvement in using unigram language models for tokenization over BPE for morphologically rich languages (Bostrom and Durrett, 2020). The generated tokens align bet-

[2] Uses a language model to predict the morphemes.

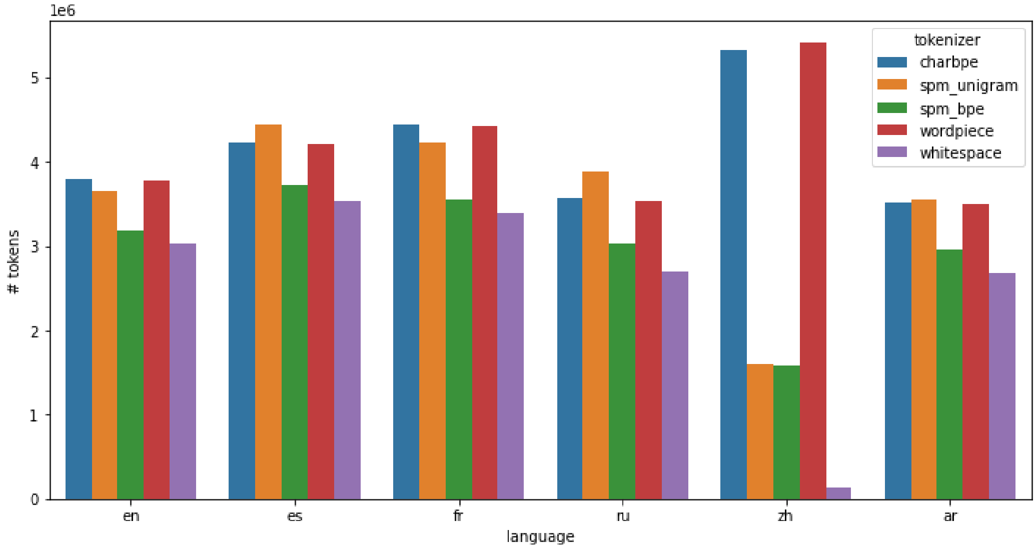


Figure 3: Comparing the number of tokens generated by different tokenizers in different languages. The language codes represent en:English, es:Spanish, fr:French, ru:Russian, zh:Chinese, and ar:Arabic.

ter with morphology for the unigram language models compared to BPE. (King et al., 2020) analyzed sequence-to-sequence models used for translation on Russian languages for morphological inflections. They showed that conditioning such models with word embeddings for lexical semantics can improve the results for translation. Klein and Tsarfaty tested the morphological attributes of the WordPiece algorithm for modern Hebrew and showed that the linear split of the tokenization algorithms might be sub-optimal (Klein and Tsarfaty, 2020). They report that by using more language-dependent tokenization approaches we can improve language understanding for morphologically rich languages. (Gerz et al., 2018) suggest an approach for tackling morphology in language modeling via a combination of characters- with word-level predictions for 50 languages.

5 Arabic's Vocabulary Sparsity

Vocabulary sparsity is the problem of having a very large vocabulary set with many different inflections. This presents a challenge for language modeling because typically, when designing a language model, we aim to acquire vocabulary embeddings. In this section, we analyze Arabic morphology through vocabulary counting of parallel datasets.

To conduct this experiment, we utilized the parallel dataset sourced from the United Nations (Rafalovitch et al., 2009). This dataset comprises United Nations General Assembly Resolutions in six distinct languages: Arabic, Chinese, English, French, Russian, and Spanish. In Table 2, we conducted a comparison between the number of tokens and the vocabulary size across these six languages. The vocabulary size denotes the count of unique tokens within the dataset for each language. From the table, we can discern that languages with rich morphology, such as Arabic and Russian, rank first and second, respectively, in terms of the number of unique tokens, even though they have a relatively smaller number of tokens compared to Spanish and French. This phenomenon can be attributed to the extensive inflections present in morphologically rich languages (MRLs). One potential approach to mitigating this issue involves applying morphological segmentation techniques, such as using a tool like FARASA (Abdelali et al., 2016). However, as

Table 1: Complexity of morphology in different languages (Clark et al., 2021).

Dataset	**Language**
Impoverished Morphology	English
Agglutinative Morphology	Turkish
Non-concatenative Morphology	Arabic
Reduplication	Kiswahili
Compounding	German
Consonant Mutation	Welsh
Vowel harmony	Finnish

indicated in the last row of the table, this segmentation introduces a trade-off: while it substantially reduces the vocabulary size, it simultaneously increases the number of tokens. This trade-off poses challenges during the training of language models, making it more difficult for the model to comprehend longer and more sophisticated sequences.

Table 2: Number of tokens and vocabulary size in other languages compared to Arabic and segmented Arabic. In this context, a token is equivalent to a word.

Dataset	# of Tokens	Vocab Size
English (en)	2,963,479	70,330
Spanish (es)	3,465,588	79,005
French (fr)	3,328,567	63,907
Russian (ru)	2,628,322	96,292
Chinese (zh)	60,107	51,884
Arabic (ar)	2,601,126	**103,339**
Arabic Segmented	**7,844,083**	15,250

In Figure 2, we compare the most frequent 1000 words in each language in the United Nations corpus. As we can see, even though the number of tokens is very high for Arabic, the frequency is low. Note that the Chinese language achieves the lowest frequency. This is due to the fact that Chinese doesn't support white space tokenization which causes its vocabulary set to be very large, hence low frequency for repetition. Interestingly, the graph shows a linear change in the frequency for the languages starting with Spain with the highest up to Chinese with the lowest.

In Figure 3 we compare five different tokenizer approaches applied to the six different languages. As we can observe MRLs like Russian and Arabic generate a relatively small number of tokens. More importantly, the distribution of the frequency of the number of tokens across the different tokenizers seems very similar for such languages. The number of tokens generated across different languages seems to depend on the language. SentencePiece with unigram seems to generate the smallest number of tokens across different languages. However, there is no distinction between which tokenizer creates the maximum number of tokens. Note that as expected, White-space tokenization generates the lowest number of tokens for all the languages.

6 Multilingual BERT

BERT (Devlin et al., 2018) is a transformer-based model that was trained on a large corpus using unsupervised learning. The main architecture of the model is based on the transformer model from (Vaswani et al., 2017) which leverages attention to design an efficient encoder-decoder model that beats the existing machine translation models at that time. BERT is trained using a concatenation of two objectives:

- Masked language modeling: The main task is to randomly mask 15% of the tokens during training and the model has to predict these masked tokens at the end.
- Next sentence prediction: the BERT model separates sentences by a special operator `[sep]`. Then with certain probability can attach unrelated sentences together from the corpus. The objective is then focused on predicting if the second sentence is possible given the first sentence.

Using the concatenation of such objectives, the model can learn efficient text representation and can be fine-tuned on multiple tasks by attaching some uninitialized weights at the end of the model.

The multilingual version of BERT trains the model on a multilingual corpus that contains 104 languages. The initial training corpus was extracted from Wikipedia with the top 100 languages then Thai and Mongolian were later added. This results in some languages which are under-represented. To mediate that, the authors used sampling to reduce the probability of training on high-resource languages like English and increase the probability of sampling from low-resource languages like Icelandic. The base model used a shared vocabulary size of 110K which was extracted using the WordPiece tokenization algorithm. Similar to sampling, the word counts are multiplied with the sampling factor as in the training to reduce the effect of variation in the existence of different languages.

7 Tasks

In this paper, we mainly focus on three tasks which are natural language inference, question answering, and named entity recognition. We use the datasets that have parallel sentences i.e. the same sentences are used for training the language models but in different languages. The reason for that choice is

Table 3: The number of samples in each dataset across the different languages.

Dataset	Number of languages	Train	Valid	Test
XNLI	15	50,000	2,490	5,010
XQuAD	11	952	119	119
MASSIVE	52	11,514	2,033	2,974

1) we want the same amount of data in terms of height (number of samples) and roughly the same depth (number of tokens per sentence) and 2) we don't want to infuse any types of bias due to using different sentences for different languages i.e we want to force the model to use the knowledge in a similar setting to machine translation. We chose three datasets which are XNLI for natural language inference, XQuAD for question answering, and MASSIVE for named entity recognition. Here is a detailed explanation of each dataset.

1. **XNLI (Conneau et al., 2018)** is a natural language inference dataset that was extracted from MNLI (Williams et al., 2018) which is a multi-genre dataset that contains more than 400K pairs of sentences. XNLI contains 392,702 training, 2,490, and 5,010 samples machine-translated into 14 different languages. The main purpose of natural language inference is to predict if the `hypothesis` follows from a `premise` i.e. entailment or contradiction or neither. Given the hypothesis and premise, the task is to predict one of the three labels so, this can be considered a more generalized classification task. Due to the size of the dataset and the limited compute, we only extract 50K samples from the dataset for fine-tuning.

2. **XQuAD (Artetxe et al., 2019)** is a cross-lingual question answering dataset. It was extracted from the SQuAD v1.1 (Rajpurkar et al., 2016) benchmark by collecting 240 paragraphs and 1,190 question-answer pairs from the development set. Then it was translated into ten languages which are Arabic, Chinese, Hindi, German, Greek, Russian, Spanish, Thai, Turkish, and Vietnamese. Hence, the dataset contains parallel samples from 11 languages. We split the dataset into 952 training, 119 validation, and 119 testing splits. Each sample of the dataset contains, `question`, `context`, and `answer_span`.

3. **MASSIVE (FitzGerald et al., 2022)** contains 1 million sentences that span across 52 languages. Each language contains 19,521 samples that were split into 11,514 training, 2,033, and 2,974 testing. The dataset is annotated for natural language understanding tasks. We mainly use the dataset for named entity recognition tasks which contains 111 tags spanning different entities like food, person, coffee, time, etc.

In Table 3, we summarize the number of samples in each dataset for each split and the number of languages for each dataset. Note that, although there are many datasets that test cross-lingual understanding, we only consider datasets that have parallel samples in each language.

8 Results and Discussions

We fine-tune mBERT[3] using the Trainer class[4] which provides a simple way for training and fine-tuning transformer-based models. All the experiments were run using Google Colab[5] with the default virtual machine that contains a T4 NVIDIA card with 16 GB memory size. We use the PyTorch examples from the Transformer repository on GitHub[6] with the following parameters for each task:

- **Natural Language Inference** We fine-tuned the model for 2 epochs with batch size 32 and learning rate 5e-5. We use a max sequence length of size 128 for the premise.

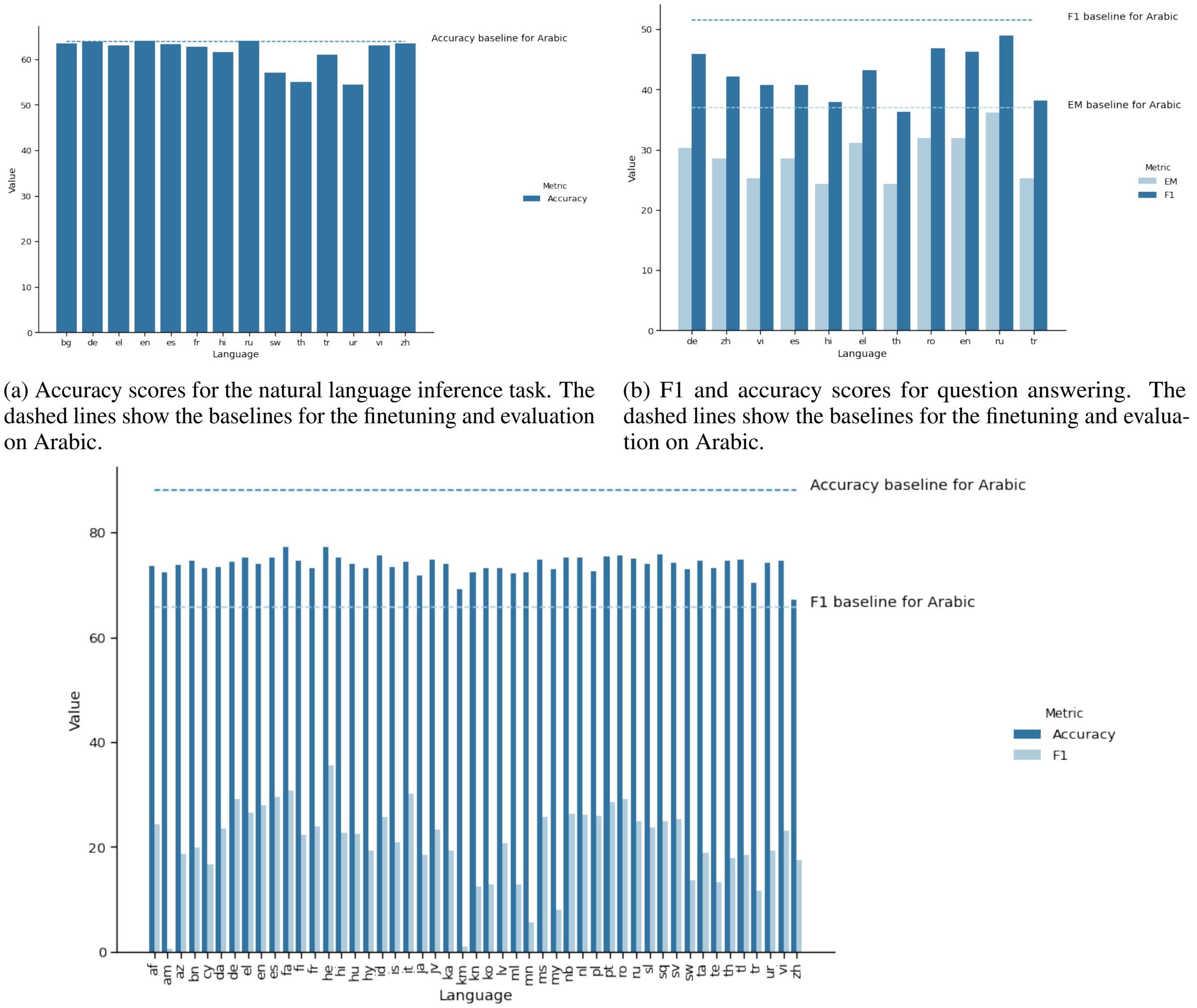
- **Question Answering** we fine-tune the models for two epochs with a batch size of 12. We

[3] `https://huggingface.co/bert-base-multilingual-cased`

[4] `https://huggingface.co/docs/transformers/main_classes/trainer`

[5] `https://colab.research.google.com`

[6] `https://github.com/huggingface/transformers/tree/main/examples/pytorch`



(a) Accuracy scores for the natural language inference task. The dashed lines show the baselines for the finetuning and evaluation on Arabic.

(b) F1 and accuracy scores for question answering. The dashed lines show the baselines for the finetuning and evaluation on Arabic.

(c) Accuracy and F1 scores for the named entity recognition task. The dashed lines show the baselines for the finetuning and evaluation on Arabic.

Figure 4: Results for question answering, natural language inference, and named entity recognition tasks.

also use a learning rate of 3e-5. For the context size, we use 384 max-size with a stride of size 128.

- **Named Entity Recognition** We fine-tune the model for two epochs with batch size 12 and learning rate 3e-5.

In Figure 4a, we show the results for the zero-shot evaluation on the natural language inference dataset XNLI. English and Russian achieve very similar results which approach the baseline for Arabic. The German language also achieves somewhat close results to the baselines. Urdu and Thai both achieve the worst results for natural language inference which are close to 50 % accuracy which is much lower than Chinese. Although Chinese and Thai are similar in pronunciation and other grammatical features, they belong to different language families. Note that this is just based on the 50,000 samples used for training. Increasing the training samples might result in different results, especially for the languages that are close in results. Furthermore, the results approach the baseline for fine-tuning and evaluating on Arabic. This might be the effect of using a machine-translated dataset for evaluation.

In Figure 4b, we show the results for the zero-shot evaluation on the question-answering dataset xQuAD. The dashed lines show the results of the baselines after training and evaluating on Arabic for exact match and F1 scores. We notice that the Russian language achieves the best scores for both the Exact match and F1 scores. These results correlate with the initial experiments in Section 5. Followed by the Romanian language which seems quite close to Russian in terms of structure. The

Thai language achieves the worst scores across all metrics which might be related to the structure of the language which is quite close to Chinese which does not use white-space tokenization.

In Figure 4c, we present the outcomes of the zero-shot evaluation conducted on the named entity recognition task. Overall, it is discernible that the outcomes, particularly the F1 scores, exhibit notable decrements when compared to the baseline, specifically within the Arabic language context. This discrepancy could be attributed to the dataset's substantial entity count, exceeding 100. Consequently, this abundance of entities introduces a degree of stochasticity into the cross-lingual comprehension process, complicating the derivation of definitive insights from the results.

9 Conclusion

In this research, we delved into the realm of cross-lingual zero-shot transfer, where we explored the application of knowledge from various languages to Arabic through the evaluation of multiple tasks. These tasks encompassed named entity recognition, natural language inference, and question answering. Initially, we employed an unsupervised approach to scrutinize the distinctions in morphology between Arabic and other languages. Subsequently, by employing supervised methods, we revealed certain connections between Arabic and other languages concerning their structure and writing systems. Our investigation demonstrated that superior results can be achieved by training models on languages other than Arabic and subsequently assessing their performance on Arabic, as opposed to direct training on Arabic. This phenomenon may be attributed to several factors, including the simplicity of the language, the resemblance of these languages to Arabic, and the distribution of the initial training data used for unsupervised learning. As a future direction, it could be interesting to look into more diverse tasks and more advanced transformer-based architectures.

Limitations

We highlight some limitations of our study. We summarize them as the following:

- **Data Quality** The quality and quantity of data available in the target languages, especially Arabic, can significantly impact the effectiveness of cross-lingual transfer. Limited or low-quality data can lead to sub-optimal results. For example, the XNLI dataset is machine-translated from English to Arabic which could result in some issues.

- **Language Distance** The success of cross-lingual transfer often depends on the linguistic distance between the source and target languages. If the source languages are distant from Arabic in terms of syntax, grammar, and vocabulary, the transfer may not be as effective.

- **Task Relevance** The paper discusses evaluating multiple tasks, including named entity recognition, natural language inference, and question answering. It's important to consider whether these tasks are representative of the general language understanding domain and whether the findings can be generalized to other tasks.

- **Bias and Fairness** The study doesn't explicitly mention considerations related to bias and fairness. Cross-lingual models can inherit biases from their training data, which can be problematic, especially in applications like named entity recognition.

- **Generalization** While the study shows promising results for certain tasks and languages, it's essential to assess the generalization of these findings to a broader range of languages and tasks. What works for one language pair may not hold for others. Also, there are variations of languages used in each task which might affect the final assumptions. Furthermore, this study focuses on using mBERT and whether this generalizes to more recent architectures is an interesting research question to be considered in future work.

- **Evaluation Metrics** The types of evaluation metrics could affect the insight we extract from such experiments. In our study, we focused on using multiple evaluation metrics, especially for question answering and named entity recognition. In our NER experiments, we highlight the huge difference between using the F1 score vs. using the accuracy score in the evaluation.

References

Khadige Abboud, Olga Golovneva, and Christopher DiPersio. 2022. Cross-lingual transfer for low-resource arabic language understanding. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 225–237.

Ahmed Abdelali, Kareem Darwish, Nadir Durrani, and Hamdy Mubarak. 2016. Farasa: A fast and furious segmenter for arabic. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: Demonstrations*, pages 11–16.

Ahmed Abdelali, Hamdy Mubarak, Shammur Absar Chowdhury, Maram Hasanain, Basel Mousi, Sabri Boughorbel, Yassine El Kheir, Daniel Izham, Fahim Dalvi, Majd Hawasly, et al. 2023. Benchmarking arabic ai with large language models. *arXiv preprint arXiv:2305.14982*.

Mohamed Alkaoud and Mairaj Syed. 2020. On the importance of tokenization in arabic embedding models. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 119–129.

Badr AlKhamissi, Muhammad N ElNokrashy, and Mohamed Gabr. 2020. Deep diacritization: Efficient hierarchical recurrence for improved arabic diacritization. *arXiv preprint arXiv:2011.00538*.

Zaid Alyafeai, Maged S Alshaibani, Badr AlKhamissi, Hamzah Luqman, Ebrahim Alareqi, and Ali Fadel. 2023. Taqyim: Evaluating arabic nlp tasks using chatgpt models. *arXiv preprint arXiv:2306.16322*.

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

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2019. On the cross-lingual transferability of monolingual representations. *CoRR*, abs/1910.11856.

Kaj Bostrom and Greg Durrett. 2020. Byte pair encoding is suboptimal for language model pretraining. *arXiv preprint arXiv:2004.03720*.

Jonathan H Clark, Dan Garrette, Iulia Turc, and John Wieting. 2021. Canine: Pre-training an efficient tokenization-free encoder for language representation. *arXiv preprint arXiv:2103.06874*.

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

Alexis Conneau, Guillaume Lample, Ruty Rinott, Adina Williams, Samuel R Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. Xnli: Evaluating cross-lingual sentence representations. *arXiv preprint arXiv:1809.05053*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jack FitzGerald, Christopher Hench, Charith Peris, Scott Mackie, Kay Rottmann, Ana Sanchez, Aaron Nash, Liam Urbach, Vishesh Kakarala, Richa Singh, Swetha Ranganath, Laurie Crist, Misha Britan, Wouter Leeuwis, Gokhan Tur, and Prem Natarajan. 2022. Massive: A 1m-example multilingual natural language understanding dataset with 51 typologically-diverse languages.

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.

Valentin Hofmann, Janet B Pierrehumbert, and Hinrich Schütze. 2020. Generating derivational morphology with bert. *arXiv preprint arXiv:2005.00672*.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *International Conference on Machine Learning*, pages 4411–4421. PMLR.

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. *arXiv preprint arXiv:2308.03051*.

Muhammad Khalifa, Muhammad Abdul-Mageed, and Khaled Shaalan. 2021. Self-training pre-trained language models for zero-and few-shot multi-dialectal arabic sequence labeling. *arXiv preprint arXiv:2101.04758*.

Md Tawkat Islam Khondaker, Abdul Waheed, El Moatez Billah Nagoudi, and Muhammad Abdul-Mageed. 2023. Gptaraeval: A comprehensive evaluation of chatgpt on arabic nlp. *arXiv preprint arXiv:2305.14976*.

David King, Andrea Sims, and Micha Elsner. 2020. Interpreting sequence-to-sequence models for russian inflectional morphology. In *Proceedings of the Society for Computation in Linguistics 2020*, pages 409–418.

Stav Klein and Reut Tsarfaty. 2020. Getting the## life out of living: How adequate are word-pieces for modelling complex morphology? In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 204–209.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. *arXiv preprint arXiv:1804.10959*.

Wuwei Lan, Yang Chen, Wei Xu, and Alan Ritter. 2020. Gigabert: Zero-shot transfer learning from english to arabic. In *Proceedings of The 2020 Conference on Empirical Methods on Natural Language Processing (EMNLP)*.

Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. From zero to hero: On the limitations of zero-shot cross-lingual transfer with multilingual transformers. *arXiv preprint arXiv:2005.00633*.

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

Mai Oudah, Amjad Almahairi, and Nizar Habash. 2019. The impact of preprocessing on arabic-english statistical and neural machine translation. *arXiv preprint arXiv:1906.11751*.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. Mad-x: An adapter-based framework for multi-task cross-lingual transfer. *arXiv preprint arXiv:2005.00052*.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual bert? *arXiv preprint arXiv:1906.01502*.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training. *preprint*.

Alexandre Rafalovitch, Robert Dale, et al. 2009. United nations general assembly resolutions: A six-language parallel corpus. In *Proceedings of the MT Summit*, volume 12, pages 292–299.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Aquia Richburg, Ramy Eskander, Smaranda Muresan, and Marine Carpuat. 2020. An evaluation of subword segmentation strategies for neural machine translation of morphologically rich languages. In *Proceedings of the The Fourth Widening Natural Language Processing Workshop*, pages 151–155.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Ashish Vaswani et al. 2017. *Attention is all you need.* Advances in neural information processing systems.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*.

A Appendix

In Tables 5, 6, and 7, we show off the results also for finetuning and evaluating on the same language and on Arabic on zero-shot fashion. Mostly, we don't see any correlation between achieving high evaluation scores on the same language and then on Arabic.

Table 4: Language Codes

Language	Code	Language	Code
Afrikaans	af	Dutch	nl
Khmer	km	Polish	pl
Kannada	kn	Portuguese	pt
Korean	ko	Romanian	ro
Latvian	lv	Russian	ru
Malayalam	ml	Slovenian	sl
Mongolian	mn	Albanian	sq
Malay	ms	Swedish	sv
Burmese	my	Swahili	sw
Norwegian Bokmål	nb	Tamil	ta
Chinese	zh	Telugu	te
Amharic	am	Thai	th
Arabic	ar	Filipino	tl
Azerbaijani	az	Turkish	tr
Bengali	bn	Urdu	ur
Welsh	cy	Vietnamese	vi
Danish	da	English	en
German	de	Spanish	es
Greek	el	Persian	fa
Hindi	hi	Finnish	fi
Hungarian	hu	French	fr
Armenian	hy	Hebrew	he
Indonesian	id	Italian	it
Icelandic	is	Japanese	ja
Javanese	jv	Georgian	ka

Table 5: Results for question answering. Exact match and F1 scores are shown as the metrics.

v

	de	**zh**	**vi**	**es**	**hi**	**el**	**th**	**ro**	**ar**	**en**	**ru**	**tr**
EM	30.25	40.34	34.45	41.18	26.89	34.45	40.34	40.34	36.97	43.70	41.18	31.09
F1	45.99	51.03	51.25	56.03	36.69	47.00	46.62	52.37	51.56	57.10	55.81	40.21
EM$_{ar}$	30.25	28.57	25.21	28.57	24.37	31.09	24.37	31.93	36.97	31.93	36.13	25.21
F1$_{ar}$	45.91	42.10	40.80	40.80	37.89	43.16	36.32	46.80	51.56	46.23	48.95	38.22

Table 6: Results for named entity recognition. Accuracy and F1 scores are shown as the metrics.

	af	**am**	**ar**	**az**	**bn**	**cy**	**da**	**de**	**el**	**en**	**es**	**fa**	**fi**	**fr**	**he**	**hi**	**hu**
Ac	91.2	73.6	88.1	89.6	89.3	89.7	91.9	91.4	90.6	92.4	89.5	91.9	89.0	90.5	88.5	90.8	89.8
F1	69.6	3.8	65.8	69.5	65.1	63.5	73.0	70.2	69.0	74.4	66.8	71.1	69.2	68.7	65.6	65.2	68.9
Ac$_{ar}$	73.5	72.4	88.1	73.7	74.5	73.2	73.4	74.4	75.2	73.9	75.2	77.2	74.6	73.2	77.2	75.2	73.9
F1$_{ar}$	24.3	0.6	65.8	18.7	19.9	16.7	23.6	29.1	26.6	27.9	29.5	30.8	22.2	24.0	35.6	22.7	22.6

	hy	**id**	**is**	**it**	**ja**	**jv**	**ka**	**km**	**kn**	**ko**	**lv**	**ml**	**mn**	**ms**	**my**	**nb**	**nl**
Ac	89.0	89.5	90.2	89.8	91.6	89.1	86.5	68.9	86.9	89.3	89.4	88.2	87.8	90.0	91.5	91.5	91.6
F1	65.7	68.3	68.3	68.5	84.9	66.8	66.0	3.0	62.1	68.3	68.2	65.4	62.0	69.4	76.4	70.9	70.9
Ac$_{ar}$	73.3	75.7	73.3	74.4	71.9	74.8	74.0	69.2	72.4	73.3	73.2	72.2	72.4	74.9	73.0	75.2	75.2
F1$_{ar}$	19.3	25.8	20.9	30.1	18.4	23.2	19.2	0.9	12.5	12.8	20.7	12.9	5.6	25.7	8.1	26.3	26.0

	pl	**pt**	**ro**	**ru**	**sl**	**sq**	**sv**	**sw**	**ta**	**te**	**th**	**tl**	**tr**	**ur**	**vi**	**zh**
Ac	88.5	90.3	89.2	89.8	89.1	90.2	91.5	87.2	87.4	87.8	91.9	89.6	88.5	89.7	89.5	91.5
F1	65.5	69.2	67.2	69.7	67.9	66.9	72.4	62.9	64.6	62.6	80.8	64.0	65.7	61.3	64.9	85.6
Ac$_{ar}$	72.5	75.4	75.7	75.0	73.9	75.7	74.1	73.1	74.7	73.3	74.5	74.8	70.4	74.1	74.6	67.2
F1$_{ar}$	25.8	28.6	29.1	24.8	23.7	24.8	25.3	13.7	18.9	13.2	17.9	18.4	11.7	19.2	23.1	17.4

Table 7: Results for natural language inference. Accuracy is shown as the metric.

	ar	**bg**	**de**	**el**	**en**	**es**	**fr**	**hi**	**ru**	**sw**	**th**	**tr**	**ur**	**vi**	**zh**
Ac	64.0	69.4	70.1	67.6	76.2	71.6	70.9	63.3	68.8	59.0	59.1	65.7	57.7	69.7	70.8
Ac$_{ar}$	64.0	63.5	63.9	63.1	64.1	63.3	62.8	61.6	64.0	57.1	55.0	60.9	54.4	63.1	63.5

Evaluating ChatGPT and Bard AI on Arabic Sentiment Analysis

Abdulmohsen Al-Thubaity[1], **Sakhar Alkhereyf**[1], **Hanan Murayshid**[1],
Nouf Alshalawi[1], **Maha Bin Omirah**[2], **Raghad Alateeq**[2], **Rawabi Almutairi**[2],
Razan Alsuwailem[3], **Manal Alhassoun**[1], **Imaan Alkhanen**[1]
[1]King Abdulaziz City for Science and Technology, Saudi Arabia
[2]Imam Mohammad ibn Saud University, Saudi Arabia
[3]Qassim University, Saudi Arabia
{aalthubaity,salkhereyf,hmurayshed,nalshalawi}@kacst.edu.sa

Abstract

Large Language Models (LLMs) such as ChatGPT and Bard AI have gained much attention due to their outstanding performance on a range of NLP tasks. These models have demonstrated remarkable proficiency across various languages without the necessity for full supervision. Nevertheless, their performance in low-resource languages and dialects, like Arabic dialects in comparison to English, remains to be investigated. In this paper, we conduct a comprehensive evaluation of three LLMs for Dialectal Arabic Sentiment Analysis: namely, ChatGPT based on GPT-3.5 and GPT-4, and Bard AI. We use a Saudi dialect Twitter dataset to assess their capability in sentiment text classification and generation. For classification, we compare the performance of fully fine-tuned Arabic BERT-based models with the LLMs in few-shot settings. For data generation, we evaluate the quality of the generated new sentiment samples using human and automatic evaluation methods. The experiments reveal that GPT-4 outperforms GPT-3.5 and Bard AI in sentiment analysis classification, rivaling the top-performing fully supervised BERT-based language model. However, in terms of data generation, compared to manually annotated authentic data, these generative models often fall short in producing high-quality Dialectal Arabic text suitable for sentiment analysis.

1 Introduction

Sentiment analysis is the task of determining the emotional tone of a piece of text, such as whether it is positive, negative, or neutral. It is a challenging task for many languages, including Arabic, due to the complex morphology and syntax of the language. Various approaches have been used to tackle this challenge, including rule-based and dictionary-based methods (ElSahar and El-Beltagy, 2014; Al-Twairesh et al., 2016; Al-Thubaity et al., 2018b), classical machine learning algorithms (Abdul-Mageed et al., 2014; Duwairi and Qarqaz, 2014; Abdulla et al., 2013; Mourad and Darwish, 2013; Abdul-Mageed et al., 2011), deep learning (Alayba et al., 2018), and pre-trained language models such as BERT (Devlin et al., 2018).

However, sentiment analysis faces a critical challenge, particularly in the context of social media, which is data drift and concept drift (Zhao et al., 2022). This challenge necessitates continuous monitoring of sentiment analysis models and updating rule-based systems and dictionaries, if utilized, as well as retraining machine learning models with new data.

Recent advancements in NLP, particularly the emergence of large language models (LLM) such as GPT-4 (OpenAI, 2023) and PaLM 2 (Anil et al., 2023), and their utilization in ChatGPT and Bard AI, respectively, show potential in countering the issues of data and concept drift in sentiment analysis. These LLMs are trained on large and diverse datasets and fine-tuned or prompted for various tasks, including sentiment analysis (Wang et al., 2023; Qin et al., 2023; Kocoń et al., 2023; Amin et al., 2023), and have proven their capabilities for this task in English.

Although few research efforts have been made to test the ability of LLMs for Arabic sentiment analysis, focusing on single language models like AraT5 (Elmadany et al., 2022) or multiple models, including ChatGPT and others (Khondaker et al., 2023), to the best of our knowledge, no study has been conducted to evaluate both of Bard AI and ChatGPT LLMs for Arabic Sentiment Analysis. In particular, this is the first attempt to evaluate Bard AI on Arabic Sentiment Analysis.

This paper aims to evaluate three generative large language models, namely Generative Pre-trained Transformers GPT-3.5 and GPT-4 through ChatGPT by OpenAI, and the Pathways Language Model (PaLM) through Bard by Google on a sentiment analysis dataset comprising in the Saudi dialect, the Saudi Dialect Twitter Corpus (SDTC) (Al-Thubaity et al., 2018a), comprising 5,400 tweets classified into five classes: positive, negative, neutral, spam, and "I do not know" class, to reveal the capabilities of LLMs in tackling the Arabic sentiment analysis challenge.

The contribution of this paper is twofold. First, it includes the evaluation of Google Bard AI for the first time in this type of analysis. Second, it evaluates these models for Arabic sentiment analysis from different and novel perspectives, as illustrated in the experiment's design (section 3). Mainly, we address the following Research Questions (RQs):

- RQ1: How is the performance of generative models when compared with fully supervised models in a relatively challenging and subjective task for Arabic NLP, namely, Arabic Sentiment Analysis? We investigate when there are few or no available training examples for generative models and compare the performance with fully fine-tuned BERT-based models.

- RQ2: What is the difference in the performance of widely used generative models on the Arabic Sentiment Analysis? In particular, we use ChatGPT (both GPT 3.5 and GPT-4) and Bard AI (PaLM 2), which, to the best of our knowledge, is the first paper to evaluate Bard AI on Arabic Sentiment Analysis.

- RQ3: How good are these models for generating new sentiment data examples in Arabic dialects? We investigate this in two ways: 1) manual evaluation of the generated examples. 2) using these examples as training samples for BERT-based models and comparing the performance with manually annotated data.

The structure of the rest of the paper is as follows:

Section 2 presents previous work on sentiment analysis and using generative models for natural understanding tasks. Section 3 shows the experiment design and the dataset used to evaluate various models. Then, in section 4, we present the results of four comprehensive experiments and analysis. We conclude the paper in section 5.

2 Related Work

2.1 Arabic Sentiment Analysis Corpora

Over the past ten years, Sentiment Analysis research, especially in the Arabic language, has gained significant interest because of the accessible sentiment data primarily from social media platforms like Twitter. The growth of social media has enabled Arabic speakers to write in their dialects, which was previously limited to the spoken form due to the language's diglossic nature. This has resulted in an abundance of dialectal textual data without the formality of standards, unlike Modern Standard Arabic (MSA) (Darwish et al., 2021). Numerous datasets have emerged for Arabic sentiment analysis across different genres, mainly tweets, with a majority in Arabic dialects, including Egyptian (Nabil et al., 2015; Refaee and Rieser, 2014), Levantine (Baly et al., 2018), Maghrebi (Mdhaffar et al., 2017; Zarra et al., 2017), and Saudi Dialect (Al-Thubaity et al., 2018a; Al-Twairesh et al., 2017; Assiri et al., 2016), among others. Other datasets cover multiple Arabic dialects in addition to Modern Standard Arabic (MSA) (Elmadany et al., 2018; Al-Obaidi and Samawi, 2016; Abdul-Mageed et al., 2014).

2.2 Arabic Sentiment Analysis Methods

Historically, much like other languages, Arabic Sentiment Analysis relied on rule-based methods, focusing primarily on crafting sentiment lexicons (ElSahar and El-Beltagy, 2014; Al-Twairesh et al., 2016; Al-Thubaity et al., 2018b). In more recent years, there has been a growing interest in using machine learning methods for Arabic Sentiment Analysis. These methods can learn the patterns of sentiment from a large corpus of text, and they are not as susceptible to the limitations of lexicon-based methods. Notable machine learning techniques employed include Naïve Bayes (NB), Support Vector Machines (SVMs), and K-Nearest Neighbor (k-NN) classifiers, leveraging morphological and syntactic features (Abdul-Mageed et al., 2014; Duwairi and Qarqaz, 2014; Abdulla et al., 2013; Mourad and Darwish, 2013; Abdul-Mageed et al., 2011).

The rapid evolution of natural language processing (NLP) has been marked by the introduction and success of transformer-based models, particularly BERT (Devlin et al., 2018) in 2018. Following that, other transformer-based models for natural language understanding (NLU) have been proposed, such as RoBERTa (Liu et al., 2019), XLNet (Yang et al., 2019), and ALBERT (Lan et al., 2020). Many of these models were pre-trained on mono-lingual datasets, mainly in English. Also, multilingual models were released, such as mBERT (Devlin et al., 2018), or language-specific models (other than English). Remarkably, there have been proposed Arabic-specific pre-trained language models, for example, ArBERT (Abdul-Mageed et al., 2021), MarBERT (Abdul-Mageed et al., 2021), AraBERT (Antoun et al., 2020), and CAMELBERT (Inoue et al., 2021). BERT and BERT-like models achieved state-of-the-art performance on many NLP tasks, including sentiment analysis in many languages Sun et al. (2019).

2.3 Generative Models for Arabic NLP

While BERT and BERT-like models are discriminative models for NLU tasks, the NLP community also witnessed a surge in the development and application of generative models designed to produce new text samples. Examples of generative models include, the GPT (Radford et al., 2018, 2019; Brown et al., 2020), T5 (Raffel et al., 2020), and BLOOM (Scao et al., 2022). Similar to BERT-like models, there have been proposed multilingual and language-specific generative models and, more specifically for Arabic, such as AraT5 (Elmadany et al., 2022), and AraGPT-2 (Antoun et al., 2021). Generative models have shown promise in tasks like text completion, translation, summarization, and even sentiment analysis, where they can be used to generate sentiment-consistent text expansions, modifications, or new text examples. In particular, Elmadany et al. (2022) shows that the AraT5 model outperforms state-of-the-art models on several Arabic language generation tasks. AraT5 is pre-trained on a large Arabic text and code dataset and fine-tuned on diverse Arabic language generation tasks, including machine translation, summarization, question answering, and paraphrasing.

2.4 Evaluating Generative LLMs

The introduction of generative Large Language Models (LLMs) like ChatGPT and Bard AI marked a significant milestone in the journey of generative models. These models, built on more advanced versions of the transformer architecture, such as GPT-3, GPT-4, and PaLM, demonstrated human-like text generation capabilities in multiple languages, including Arabic. Following this trend, there has been a growing interest in evaluating the capabilities of generative models, mainly ChatGPT and Google Bard AI, for various NLP tasks, such as sentiment analysis (Wang et al., 2023; Qin et al., 2023; Kocoń et al., 2023; Amin et al., 2023), summarization (Qin et al., 2023; Alyafeai et al., 2023; Khondaker et al., 2023), and POS tagging (Alyafeai et al., 2023; Abdelali et al., 2023). Initial methodological efforts to evaluate these models focus on their performance in high-resource languages such as English (Qin et al., 2023; Bubeck et al., 2023). Other studies have evaluated LLMs for their performance on other low-resource languages (Ahuja et al., 2023; Bang et al., 2023; Lai et al., 2023). The findings from these studies indicate that the trending ChatGPT is a capable language model, but it does not surpass the current state-of-the-art (SOTA) solutions in most NLP tasks. However, when it comes to sentiment analysis, which is the main focus of our research, one study (Amin et al., 2023) contradicted the majority and found that ChatGPT outperformed the leading solution, suggesting that this is a promising area for further research. Furthermore, most studies on sentiment analysis using ChatGPT have been conducted on English datasets, and a few research in the Arabic language. Therefore, our research aims to bridge the gap in sentiment analysis research for the Arabic language and demonstrate the potential of ChatGPT in understanding and analyzing sentiment in this context.

For Arabic, Khondaker et al. (2023) present a comprehensive evaluation of ChatGPT's performance on 32 Arabic NLP tasks, including sentiment analysis. The results suggest that, although ChatGPT performs satisfactorily on most Arabic NLP tasks, it is consistently surpassed by the smaller Arabic-focused, fully supervised, fine-tuned model, AraT5. Alyafeai et al. (2023) investigate the performance of the two ChatGPT models, GPT-3.5 and GPT-4, on seven Arabic NLP tasks and compare their performance against SoTA models. On the sentiment analysis task, the results show that GPT-4 outperforms GPT-3.5. However, both models are outperformed by the SoTA model, i.e., MARBERT (Abdul-Mageed et al., 2021). They show that GPT-4 was more robust to different prompts, and its performance improved with the increase in the number of few-shot examples, unlike GPT-3.5. Another study (Abdelali et al., 2023) demonstrated that the SoTA model (with 0.760 F-1 score) outperformed ChatGPT (with 0.550 F1 score) on an Arabic sentiment analysis dataset. However, the ChatGPT model was only evaluated in a zero-shot learning setting, meaning that it was not given any example of the sentiment analysis task. We notice that the only study that includes

Bard AI for Arabic NLP tasks is (Kadaoui et al., 2023), which conducted a comprehensive assessment of Bard AI and ChatGPT, covering both GPT-3.5 and GPT-4 in the domain of machine translation across ten varieties of Arabic.

In contrast to the studies mentioned above, we also include Google's Bard AI in our evaluation for Arabic Sentiment Analysis. This is significant because, to the best of our knowledge, although there are some studies that use ChatGPT for Arabic Sentiment Analysis, no other comparable research has been conducted to evaluate both Bard AI and ChatGPT on Arabic Sentiment Analysis.

3 Experiments Design and Data

The primary objective of these experiments is to assess the capabilities of generative models for Arabic sentiment analysis and the potential of data augmentation and generation for this task. We evaluate three models:

- Generative Pre-trained Transformers GPT-3.5,
- and GPT-4, both accessed via ChatGPT by OpenAI.
- PaLM 2 facilitated by Bard AI by Google. Throughout the paper, the terms "Bard" and "PaLM" will be used interchangeably.

For GPT-3.5 and GPT-4 we utilize the ChatGPT API to send prompts and receive responses. For PaLM 2, prompts are manually sent to Bard AI via its web interface, from which we extract the relevant responses. Also, we utilize these models to generate new samples and systematically evaluate the generated examples.

We have four main experiments:

- **Exp.1**: As a baseline, we train various Arabic BERT-based language models using an existing dataset and assess their performance on the dataset. We utilize models pre-trained on various Arabic corpora, specifically those trained on Twitter or Arabic dialectal data. The model that shows the best performance will serve as our baseline model.
- **Exp.2**: We evaluate the performance of the generative models (i.e., GPT-3.5, GPT-4, and PaLM 2) by instructing them to classify the test data into positive, negative, or neutral categories. We conduct the evaluation using k shots. We will assess the performance of each model against the test data and compare it to the best model identified in Exp. 1. This experiment aims to address RQ1 and RQ2.
- **Exp.3**: We prompt the generative models with a given sentiment (positive, negative, and neutral), instructing them to generate m tweets. Samples of the generated tweets will be manually evaluated for their naturalness using various criteria. This experiment aims to address RQ3.
- **Exp.4**: The data generated in Exp.3 will be utilized in two different ways. Firstly, it will be used to augment the original training data, which will then be fine-tuned with the BERT-based models used in Exp 1. Secondly, the synthesized data will be used to fine-tune the BERT-based models used in Exp 1. For both approaches, the performance will be evaluated against the test data. This experiment aims to address RQ3.

For the abovementioned experiments, we use the Saudi Dialect Twitter Corpus (SDTC) (Al-Thubaity et al., 2018a). SDTC comprises 5,400 tweets distributed across five classes: positive, negative, neutral, spam, information, and difficult to classify. In our experiments, we focused on the first three classes: positive, negative, and neutral tweets, amounting to 558, 1,632, and 500, respectively. The total number of tweets in our experiments is 2,690.

We randomly split the SDTC dataset into 75% for training $\text{SDTC}_{\text{train}}$ and 25% for testing $\text{SDTC}_{\text{test}}$, obeying the class distribution. Also, we selected 30 tweets from each class (90 tweets overall) from $\text{SDTC}_{\text{train}}$ to evaluate the output of each proposed prompt. We use SDTC_{dev} to refer to these 90 tweets.

$\text{SDTC}_{\text{train}}$ is used to fine-tune the language models in Exp 1 and Exp 4. The $\text{SDTC}_{\text{test}}$ set is used for evaluating the fine-tuned language models (Exp 1 and Exp 4) and for the predictions of the generative models in Exp 2. Experiment 3 involves human judgment, and the outputs of the generative models will be used to fine-tune the language models in Exp 4. We use SDTC_{dev} for evaluating the output of different prompts in Exp 2 and Exp 3. We make SDTC_{dev} balanced in classes because of its relatively small size, due to budget and time constraints, as we couldn't evaluate all prompts and different numbers of shots on a larger scale.

However, we evaluate the best settings in terms of prompts and number of shots on the whole test set SDTC$_{\text{test}}$. These prompts were inspired or adapted from previously published research (Alyafeai et al., 2023; Khondaker et al., 2023).

4 Experiments and Results

In this section, we show and discuss the results of the experiments described in section 3. To assess the performance of the models, we employed the accuracy (Acc) metric, along with the micro-averages of precision (P), recall (R), and F-1 score (F) values. When evaluating the models' performance, we focus on the F1 measure as the primary metric of comparison.

4.1 Experiment 1: Fine-tuning BERT Models (baseline)

We fine-tuned five Arabic BERT-based models using the training data, SDTC$_{\text{train}}$, and evaluated their performance on the test data, SDTC$_{\text{test}}$. Namely, we fine-tune:

- bert-large-arabertv02-twitter and bert-base-arabertv02-twitter (Antoun et al., 2020).
- MARBERTv2 (Abdul-Mageed et al., 2021).
- bert-base-arabic-camelbert-da (Inoue et al., 2021).
- and bert-base-qarib (Abdelali et al., 2021).

Model	Acc	P	R	F-1
arabert-base	**79**	**79**	**79**	**79**
arabert-large	78	77	78	77
qarib	78	77	77	77
MARBERT	77	76	77	76
camelbert	72	72	72	72

Table 1: Performance measures for the five fine-tuned language models. We show the micro-averaged score for each metric.

These models were fully or partially pre-trained on Twitter or Arabic dialect data. Numerous experiments were conducted across all models, involving varying hyperparameter values. Appendix B shows the details of hyperparameters and experimental setups. Table 1 shows the results of fine-tuning the five models.

The results demonstrate that models solely pre-trained on the same data as the fine-tuning data exhibit the best performance, in our case, Twitter data. Notably, the performance of the bert-base-arabertv02-twitter model outperforms the larger bert-large-arabertv02-twitter model, contrary to the typical expectation.

For further analysis, see Appendix C, where we show that the best BERT-based, i.e., bert-base-arabertv02-twitter, has the highest confusion when differentiating between positive and neutral classes.

4.2 Experiment 2: Sentiment Analysis with Generative Models

In this set of experiments, we evaluate ChatGPT (GPT 3.5 and GPT 4) and Bard AI on SDTC$_{\text{test}}$. Unlike the setup of hyperparameters for pre-trained language models, which are known and controlled, determining the optimal prompt design for generative models involves trial and error processes. We conducted experiments with seven prompt designs in Arabic and English to classify tweets in SDTC$_{\text{test}}$. We evaluate each prompt design on SDTC$_{\text{dev}}$ and then select the prompt with the highest accuracy using k shots where $k = \{0, 1, 3, 5\}$. Each shot is a triplet of a positive, negative, and neutral tweet.

The optimal prompt for Bard AI achieved an accuracy of 0.7 for $k = 5$ (15 tweets overall). It is as follows:

> Given the examples:
>
> positive train tweet ; Sentiment: 1 (positive)
> neutral train tweet ; Sentiment: 0 (neutral)
> negative train tweet ; Sentiment: -1 (negative)
>
> positive train tweet ; Sentiment: 1 (positive)
> neutral train tweet ; Sentiment: 0 (neutral)
> negative train tweet ; Sentiment: -1 (negative)
>
> ...
>
> You are a helpful assistant that can predict whether a given tweet in Arabic is Positive, Negative, or Neutral. Do not show any warning, explanation or disclaimer. Please provide your response for testing tweet in tabular format showing the tweet and the classification.
>
> Testing tweet: Test tweet

For GPT-3.5 and GPT-4, the optimal prompt achieved accuracy scores of 0.81 and 0.91, respectively, for $k = 0$. It is as follows:

> What is the sentiment of the following tweets? Answer with positive, negative, or neutral.
>
> Test tweet

After selecting the best prompt, we evaluate each of the three generative models on SDTC$_{\text{test}}$, asking them to classify each example as positive, negative, or neutral. If a model declines to classify a tweet due to its unacceptable content for any reason, we set the prediction to be negative. We compare the outcomes of the three generative models with the test data labels and compute the four performance measures. Table 2 shows the performance measures for the three generative models.

Model	Acc	P	R	F-1
GPT-4	75	**82**	75	**77**
Bard AI	**79**	78	**79**	76
GPT-3.5	70	72	70	70
Best BERT	79	79	79	79

Table 2: The performance measures for the three generative models on SDTC$_{\text{test}}$. We show the micro-averaged score for each metric.

Based on the F-1 score as a reference performance measure, the results show that GPT-4 and Bard AI achieve comparable performance in few-shot settings with the fully supervised BERT-based models. In particular, GPT-4 has a very close performance to the second-best BERT model (i.e., bert-large-arabertv02-twitter) with an F-1 score of 0.77, and it outperforms the other fine-tuned models. Bard AI comes in second with a score of 0.76, which performs relatively well for sentiment analysis classification compared to fully supervised models. Notably, it outperforms one of the BERT-based models and achieves comparable results to the fine-tuned MARBERTv2 model with an F1 score of 0.76. However, GPT-3.5 has low performance, falling behind BERT-based models. The significant difference between the models' performance on the development set SDTC$_{\text{dev}}$ and the test SDTC$_{\text{test}}$ can be attributed to the different class distributions. In particular, Bard AI performs very low in the neutral class, which represents the third of tweets in SDTC$_{\text{dev}}$.

Class	Best BERT model	GPT-4
Negative	**89**	84
Positive	75	**79**
Neutral	54	**58**

Table 3: F-1 scores for each sentiment class for fine-tuned bert-base-arabertv02-twitter and GPT-4 models.

While GPT-4, in a zero-shot setting, has comparable results to the fine-tuned BERT model, their performance for each class varies considerably. Table 3 shows the F1 score for each sentiment class for both models. The results show that both models (BERT and GPT-4) performed best for the classification of negative tweets, followed by positive tweets, and the most difficult classification task was for neutral tweets. The best fine-tuned BERT model (i.e., bert-base-arabertv02-twitter) outperformed GPT-4 for the classification of negative tweets. However, the latter considerably outperformed the former for the classification of positive and neutral tweets, with a 4-point increase in F1 score for both positive and neutral tweets. Again, as for fine-tuned BERT models, the greatest challenge that generative models may face is differentiating between positive and neutral tweets.

4.3 Experiment 3: Data Generation by Generative Models

We instructed each generative model in a zero-shot setting to generate positive, negative, and neutral tweets. We conducted experiments using 11 different prompt designs in both Arabic and English, and then we selected the best prompt based on the evaluation of the resulting output using three criteria:

- The naturalness of the tweets.
- Tweets are in the Saudi dialect.
- Avoidance of overly brief tweets.

We generate multiple outputs for the same prompt and evaluate it on a small scale (a few runs for each prompt), and then we select the best prompt according to the criteria mentioned above. For all generative models, the best prompt was as follows:

Your role is a data engineer who wants to create synthesized examples of tweets for Arabic sentiment analysis. Generate examples of tweets with sentiment classes [“positive”, “negative”, “neutral”]. Generate 10 examples in {*sentiment*} in the Saudi Dialect; such that each tweet is in a single row in tabular format. You must generate long tweets.

Using the best prompt above, we instructed each generative model (i.e., GPT-3.5, GPT-4, and Bard AI) to generate tweets for each class (i.e., [“positive”, “negative”, “neutral”]), matching their respective distribution in the training dataset $\text{SDTC}_{\text{train}}$, i.e., 391, 1,243, and 351 for positive, negative, and neutral tweets, respectively. See Appendix D for examples of the generated tweets.

To assess the quality of the generated tweets and their associated sentiments produced by the generative models, we randomly select 50 tweets from each class for every generative model (a total of 150 tweets per model). Subsequently, two annotators were involved in addressing the following binary inquiries (Yes/No) for each generated tweet:

- Q1 (Making sense): Is the generated tweet linguistically correct and understandable?
- Q2: (Appropriateness for Twitter): Do you expect to see such text on Twitter?
- Q3: (Matching label): Does the generated tweet match the instructed sentiment?

Model	Class	Q1	Q2	Q3	Q1+Q2+Q3
Bard AI	Pos	94	34	98	32
	Neg	**100**	**80**	94	**76**
	Neu	**90**	**72**	50	**30**
	ALL	**95**	**62**	81	**46**
GPT-3.5	Pos	**98**	**78**	98	**76**
	Neg	56	40	94	34
	Neu	74	54	28	20
	ALL	76	57	74	44
GPT-4	Pos	86	58	**100**	52
	Neg	84	46	**100**	40
	Neu	72	56	**52**	28
	ALL	81	53	**84**	40

Table 4: Percentage of affirmative responses for each question and class across the three generative models. Pos: Positive, Neu: Neutral, Neg: Negative, ALL: all classes.

A generated tweet is considered valid for each question if both annotators concur with a “Yes” response; otherwise, the tweet is regarded as invalid for that specific question.

Table 4 demonstrates the percentage of affirmative responses for each question and class across the three generative models. Table 8 in Appendix D showcases examples where the two annotators answered each question with “No”.

ALL: The data suggests that Bard AI slightly outperforms GPT-3.5 and GPT-4 when considering all evaluation questions together (46%) or individually, achieving percentages of 95%, 62%, and 81% for Q1, Q2, and Q3, respectively. All models perform well regarding linguistic correctness (Q1) and matching the instructed sentiment (Q3) for Positive and Negative tweets. However, there are challenges with generating tweets that exhibit appropriateness for Twitter (Q2).

Q1: For linguistic correctness and understandability (Q1), Bard AI consistently achieves high percentages across all classes, followed by GPT-4, which performs lower in Neutral tweets. GPT-3.5 has the lowest performance for Negative tweets. This may be attributed to the stricter constraints that prevent it from generating negative content more than Bard AI. In particular, ChatGPT (based on GPT-3.5 and GPT-4) tends to generate nonsense text in dialectal Arabic more than Bard AI.

Q2: Regarding generating tweets that exhibit appropriateness for Twitter (Q2), Bard AI achieved the highest score, specifically for Negative tweets, followed by GPT-3.5 and GPT-4, respectively. However, the latter models demonstrate low scores for Negative tweets.

Q3: Regarding matching the instructed sentiment (Q3), GPT-4 outperforms both Bard AI and GPT-3.5, with a score of 84%, achieving a perfect score of 100% for Positive and Negative tweets. However, its performance in generating Neutral tweets is relatively low (52%). The performance on Q3 for Neutral tweets is also low for the other two models.

Analysis of the generated neutral tweets: To analyze the confusion of neutral tweets with other classes discussed in Exp.1, we compare the neutral tweets generated by the generative models with the labels given by annotators and found that for Bard AI, 96% of the tweets were classified by annotators as positive. For GPT-3.5, 32% were classified as negative and 68% as positive; for GPT-4, 13% were classified as negative and 87% as positive. It seems that the generative models find it difficult to clearly distinguish neutral content from other types of content, particularly positive content. This is also demonstrated in Exp.1 when we fine-tuned BERT models, where BERT-based models struggle the most with neutral tweets and misclassify them as positive or negative tweets.

4.4 Experiment 4: Fine-tuning Best BERT Model on Generated Data

Data	Acc	P	R	F
SDTC	**79**	**79**	**79**	**79**
Bard AI	69	70	69	67
GPT-3.5	60	64	60	54
GPT-4	68	74	68	66
Bard AI+SDTC	**79**	**79**	**79**	78
GPT-3.5+SDTC	77	77	77	76
GPT-4+SDTC	**79**	**79**	**79**	**79**
All Data	76	77	76	76

Table 5: Performance measures for the using of different data sets on fine-tuning bert-base-arabertv02-twitter model. All Data: Brad AI + GPT-3.5 + GPT-4 + SDTC. For SDTC, we use only the train set, $\text{SDTC}_{\text{train}}$

We conducted seven experiments using the best-performing BERT model, namely bert-base-arabertv02-twitter, with the same hyperparameters using both the generated data and $\text{SDTC}_{\text{train}}$ for these experiments. Table 5 shows the performance of fine-tuning bert-base-arabertv02-twitter using the new data. Similar to the previous experiments, our primary focus was on the F1 measure as the key metric for comparison.

The data suggests that, for each generated dataset, the model fine-tuned on Bard AI data demonstrated the best performance on the testing data with an F1 score of 0.67. It was closely followed by the model fine-tuned on data generated by GPT-4, achieving an F1 score of 0.66. The performance data shows a positive correlation with the human evaluation of the generated data in Experiment 3. The relatively lower performance of these models can be attributed to both the fact that the testing data were sampled from a different population and the quality of the classification of the generated data by the generative models.

Combining the original training data with the generated data from each model did not improve the performance of the fine-tuned model. In fact, it might have even led to a decrease in the model's performance. Moreover, combining all the generated data with the original data has a negative impact on performance. This decrease in performance can also be attributed to the same reasons mentioned earlier.

The performance could be improved by using one or more shots of training data to generate new samples using generative models, ensuring higher similarity between the generated data and the original data distribution. Due to time and budget constraints, we could not do so.

5 Conclusion

In this paper, we presented various experiments using three generative models for Arabic Sentiment Analysis in the Saudi dialect: GPT 3.5, GPT-4, and Bard AI (PaLM 2). We compare their performance with fully supervised BERT-based models. We also evaluate the quality of generated examples by these LLMs using manual and automatic methods.

The experiments show that the generative large language models, with little or no training data in few-shot settings, perform relatively well on Arabic Sentiment Analysis compared to fully fine-tuned models. For sentiment analysis text classification, the experiments show that GPT-4 outperforms most of the BERT-based model and is on a par with the second-best BERT-based model. Bard AI comes next with a performance comparable to fully fine-tuned models, while GPT3.5 significantly underperforms the two models and is lower than the BERT-based models. For sentiment generation, all models struggle to generate high-quality text for sentiment analysis in the Saudi Dialect, especially for neutral text. Interestingly, ChatGPT (both GPT-3.5 and GPT-4) tends to generate nonsense text in the Saudi dialect more than Bard AI. Also, implementing safeguards to prevent the generation of harmful or toxic content is crucial for responsible and safe utilization. However, these restrictions can sometimes act as barriers when generating representative text that has a negative sentiment, especially in applications that require a comprehensive representation of human emotions and viewpoints.

Future research should consider comparing the generative models performance among different Arabic dialects and datasets. Also, another direction for future work is analyzing the performance of generative models pre-trained on dialectal Arabic text (such as AraT5) and fully fine-tuned on generating tweets.

Limitations

Due to constraints in time and resources, we need to highlight the following limitations:

- In Experiment 2, which involves sentiment analysis classification, we evaluated the proposed prompts on a limited number of tweets. Particularly, SDTC$_{\text{dev}}$ consists of 30 tweets from each class sampled from the training set. Employing more samples for evaluation could lead to identifying better prompts.

- In Experiment 3, focused on data generation, we only conducted experiments in zero-shot settings due to budget and time restrictions. Conducting experiments with a broader number of shots might unveil more robustly generated data. Additionally, the generated tweets were evaluated on a small data sample, utilizing a straightforward binary classification approach across three aspects of the tweets. A more comprehensive evaluation involving larger samples of generated tweets and encompassing a broader array of aspects would provide a more solid assessment.

- In Experiment 4, we solely assessed the performance of the generated data using the best-performing BERT model, namely bert-base-arabertv02-twitter. Undertaking further experimentation with other BERT-based models would yield valuable insights into the performance of these models.

Also, in utilizing large language models (LLMs) like ChatGPT and BARD AI for classifying public datasets, it is important to acknowledge potential limitations tied to data leakage. Given that these models have been trained on vast amounts of data, there is a chance they might have been exposed to, or "seen", some parts of these public datasets, including SDTC, as "pre-training data exposure". Nevertheless, the influence of this exposure might be minor for a few reasons. First, any specific dataset would only be a drop in the ocean, being among billions of tokens on which the models were trained. Second, many public datasets don't have a raw text structure, reducing direct familiarity, which is the case with SDTC. Lastly, we have shown in our experiments that both ChatGPT and Bard AI don't have perfect results on SDTC, further suggesting that any prior exposure may not significantly skew outcomes.

Ethics Statement

The results obtained in this study must be considered within the framework of the intended usage of the generative models and the criteria applied for their evaluation. The disparities in performance observed among the models could potentially stem from variances in training data, model architecture, or prompt design. Further analysis and exploration will contribute to identifying the underlying causes of these discrepancies. Additionally, our study does not address biases within the models or their approach to handling Arabic content, whether generated directly by the models or translated from other languages. The findings of this study cannot be universally extrapolated to other tasks or various Arabic dialects without undergoing comprehensive investigations.

Acknowledgement

The authors would like to thank the anonymous reviewers for their valuable comments and feedback.

References

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

Ahmed Abdelali, Hamdy Mubarak, Shammur Absar Chowdhury, Maram Hasanain, Basel Mousi, Sabri Boughorbel, Yassine El Kheir, Daniel Izham, Fahim Dalvi, Majd Hawasly, et al. 2023. Benchmarking Arabic AI with large language models. *arXiv preprint arXiv:2305.14982*.

Muhammad Abdul-Mageed, Mona Diab, and Mohammed Korayem. 2011. Subjectivity and sentiment analysis of modern standard Arabic. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 587–591.

Muhammad Abdul-Mageed, Mona Diab, and Sandra Kübler. 2014. SAMAR: Subjectivity and sentiment analysis for Arabic social media. *Computer Speech & Language*, 28(1):20–37.

Muhammad Abdul-Mageed, AbdelRahim Elmadany, et al. 2021. ARBERT & MARBERT: Deep bidirectional transformers for Arabic. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7088–7105.

Nawaf A Abdulla, Nizar A Ahmed, Mohammed A Shehab, and Mahmoud Al-Ayyoub. 2013. Arabic sentiment analysis: Lexicon-based and corpus-based. In

2013 IEEE Jordan conference on applied electrical engineering and computing technologies (AEECT), pages 1–6. IEEE.

Kabir Ahuja, Rishav Hada, Millicent Ochieng, Prachi Jain, Harshita Diddee, Samuel Maina, Tanuja Ganu, Sameer Segal, Maxamed Axmed, Kalika Bali, et al. 2023. Mega: Multilingual evaluation of generative AI. *arXiv preprint arXiv:2303.12528*.

Ahmed Y Al-Obaidi and Venus W Samawi. 2016. Opinion mining: Analysis of comments written in Arabic colloquial. In *Proceedings of the World Congress on Engineering and Computer Science*, volume 1.

Abdulmohsen Al-Thubaity, Mohammed Alharbi, Saif Alqahtani, and Abdulrahman Aljandal. 2018a. A Saudi dialect Twitter corpus for sentiment and emotion analysis. In *2018 21st Saudi computer society national computer conference (NCC)*, pages 1–6. IEEE.

Abdulmohsen Al-Thubaity, Qubayl Alqahtani, and Abdulaziz Aljandal. 2018b. Sentiment lexicon for sentiment analysis of Saudi dialect tweets. *Procedia computer science*, 142:301–307.

Nora Al-Twairesh, Hend Al-Khalifa, and AbdulMalik Al-Salman. 2016. AraSenTi: Large-scale Twitter-specific Arabic sentiment lexicons. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 697–705.

Nora Al-Twairesh, Hend Al-Khalifa, AbdulMalik Al-Salman, and Yousef Al-Ohali. 2017. AraSenTi-tweet: A corpus for Arabic sentiment analysis of Saudi tweets. *Procedia Computer Science*, 117:63–72.

Abdulaziz M Alayba, Vasile Palade, Matthew England, and Rahat Iqbal. 2018. A combined CNN and LSTM model for Arabic sentiment analysis. In *Machine Learning and Knowledge Extraction: Second IFIP TC 5, TC 8/WG 8.4, 8.9, TC 12/WG 12.9 International Cross-Domain Conference, CD-MAKE 2018, Hamburg, Germany, August 27–30, 2018, Proceedings 2*, pages 179–191. Springer.

Zaid Alyafeai, Maged S Alshaibani, Badr AlKhamissi, Hamzah Luqman, Ebrahim Alareqi, and Ali Fadel. 2023. Taqyim: Evaluating Arabic NLP tasks using chatgpt models. *arXiv preprint arXiv:2306.16322*.

Mostafa M Amin, Erik Cambria, and Björn W Schuller. 2023. Will affective computing emerge from foundation models and general AI? a first evaluation on ChatGPT. *arXiv preprint arXiv:2303.03186*.

Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernandez Abrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vlad Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, Guy Gur-Ari, Steven Hand, Hadi Hashemi, Le Hou, Joshua Howland, Andrea Hu, Jeffrey Hui, Jeremy Hurwitz, Michael Isard, Abe Ittycheriah, Matthew Jagielski, Wenhao Jia, Kathleen Kenealy, Maxim Krikun, Sneha Kudugunta, Chang Lan, Katherine Lee, Benjamin Lee, Eric Li, Music Li, Wei Li, YaGuang Li, Jian Li, Hyeontaek Lim, Hanzhao Lin, Zhongtao Liu, Frederick Liu, Marcello Maggioni, Aroma Mahendru, Joshua Maynez, Vedant Misra, Maysam Moussalem, Zachary Nado, John Nham, Eric Ni, Andrew Nystrom, Alicia Parrish, Marie Pellat, Martin Polacek, Alex Polozov, Reiner Pope, Siyuan Qiao, Emily Reif, Bryan Richter, Parker Riley, Alex Castro Ros, Aurko Roy, Brennan Saeta, Rajkumar Samuel, Renee Shelby, Ambrose Slone, Daniel Smilkov, David R. So, Daniel Sohn, Simon Tokumine, Dasha Valter, Vijay Vasudevan, Kiran Vodrahalli, Xuezhi Wang, Pidong Wang, Zirui Wang, Tao Wang, John Wieting, Yuhuai Wu, Kelvin Xu, Yunhan Xu, Linting Xue, Pengcheng Yin, Jiahui Yu, Qiao Zhang, Steven Zheng, Ce Zheng, Weikang Zhou, Denny Zhou, Slav Petrov, and Yonghui Wu. 2023. PaLM 2 technical report.

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.

Adel Assiri, Ahmed Emam, and Hmood Al-Dossari. 2016. Saudi Twitter corpus for sentiment analysis. *International Journal of Computer and Information Engineering*, 10(2):272–275.

Ramy Baly, Alaa Khaddaj, Hazem Hajj, Wassim El-Hajj, and Khaled Bashir Shaban. 2018. ArSentD-LEV: A multi-topic corpus for target-based sentiment analysis in Arabic Levantine tweets. In *OSACT 3: The 3rd Workshop on Open-Source Arabic Corpora and Processing Tools*, page 37.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. 2023. A multitask, multilingual, multimodal evaluation of ChatGPT on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*.

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

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with GPT-4. *arXiv preprint arXiv:2303.12712*.

Kareem Darwish, Nizar Habash, Mourad Abbas, Hend Al-Khalifa, Huseein T Al-Natsheh, Houda Bouamor, Karim Bouzoubaa, Violetta Cavalli-Sforza, Samhaa R El-Beltagy, Wassim El-Hajj, et al. 2021. A panoramic survey of natural language processing in the Arab world. *Communications of the ACM*, 64(4):72–81.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Rehab M Duwairi and Islam Qarqaz. 2014. Arabic sentiment analysis using supervised classification. In *2014 International Conference on Future Internet of Things and Cloud*, pages 579–583. IEEE.

A Elmadany, Hamdy Mubarak, and Walid Magdy. 2018. Arsas: An Arabic speech-act and sentiment corpus of tweets. *OSACT*, 3:20.

AbdelRahim Elmadany, Muhammad Abdul-Mageed, et al. 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.

Hady ElSahar and Samhaa R El-Beltagy. 2014. A fully automated approach for Arabic slang lexicon extraction from microblogs. In *International conference on intelligent text processing and computational linguistics*, pages 79–91. Springer.

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 *6th Arabic Natural Language Processing Workshop, WANLP 2021*, pages 92–104. Association for Computational Linguistics (ACL).

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. *arXiv preprint arXiv:2308.03051*.

Md Tawkat Islam Khondaker, Abdul Waheed, El Moatez Billah Nagoudi, and Muhammad Abdul-Mageed. 2023. GPTAraEval: A comprehensive evaluation of ChatGPT on Arabic NLP. *arXiv preprint arXiv:2305.14976*.

Jan Kocoń, Igor Cichecki, Oliwier Kaszyca, Mateusz Kochanek, Dominika Szydło, Joanna Baran, Julita Bielaniewicz, Marcin Gruza, Arkadiusz Janz, Kamil Kanclerz, et al. 2023. ChatGPT: Jack of all trades, master of none. *Information Fusion*, page 101861.

Viet Dac Lai, Nghia Trung Ngo, Amir Pouran Ben Veyseh, Hieu Man, Franck Dernoncourt, Trung Bui, and Thien Huu Nguyen. 2023. ChatGPT beyond English: Towards a comprehensive evaluation of large language models in multilingual learning. *arXiv preprint arXiv:2304.05613*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach.

Salima Mdhaffar, Fethi Bougares, Yannick Esteve, and Lamia Hadrich-Belguith. 2017. Sentiment analysis of tunisian dialects: Linguistic ressources and experiments. In *Third Arabic Natural Language Processing Workshop (WANLP)*, pages 55–61.

Ahmed Mourad and Kareem Darwish. 2013. Subjectivity and sentiment analysis of modern standard Arabic and Arabic microblogs. In *Proceedings of the 4th workshop on computational approaches to subjectivity, sentiment and social media analysis*, pages 55–64.

Mahmoud Nabil, Mohamed Aly, and Amir Atiya. 2015. Astd: Arabic sentiment tweets dataset. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 2515–2519.

OpenAI. 2023. GPT-4 technical report.

Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is ChatGPT a general-purpose natural language processing task solver? *arXiv preprint arXiv:2302.06476*.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

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.

Eshrag Refaee and Verena Rieser. 2014. An Arabic Twitter corpus for subjectivity and sentiment analysis. In *9th International Language Resources and Evaluation Conference*, pages 2268–2273. European Language Resources Association.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune BERT for text classification? In *Chinese Computational Linguistics: 18th China National Conference, CCL 2019, Kunming, China, October 18–20, 2019, Proceedings 18*, pages 194–206. Springer.

Zengzhi Wang, Qiming Xie, Zixiang Ding, Yi Feng, and Rui Xia. 2023. Is ChatGPT a good sentiment analyzer? a preliminary study. *arXiv preprint arXiv:2304.04339*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Taoufiq Zarra, Raddouane Chiheb, Rajae Moumen, Rdouan Faizi, and Abdellatif El Afia. 2017. Topic and sentiment model applied to the colloquial Arabic: a case study of Maghrebi Arabic. In *Proceedings of the 2017 international conference on smart digital environment*, pages 174–181.

Zhixue Zhao, George Chrysostomou, Kalina Bontcheva, and Nikolaos Aletras. 2022. On the impact of temporal concept drift on model explanations. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4039–4054, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

A SDTC Statistics

Table 6 illustrates the statistics of SDTC used in our experiments, including three examples from each class (positive, negative, neutral).

B Experimental Setup for Exp.1

For experiment 1 in subsection 4.1, the implementation of all the BERT-based models was carried out using Python 3.9. Fine-tuning experiments were conducted using Tesla GPUs. The following experimental setup was standardized for all the models:

- We utilized the transformers v4.21.1, AutoTokenizer, and Bert For Sequence Classification libraries from Huggingface.
- Optimization was performed using AdamW with a learning rate of 1e-5.
- The number of epochs was set to 15.
- The value for max_grad_norm was set to 1.0.
- The maximum sentence length was constrained to 70 tokens.
- A batch size of 128 was employed.

C Analysis of BERT-based model predictions

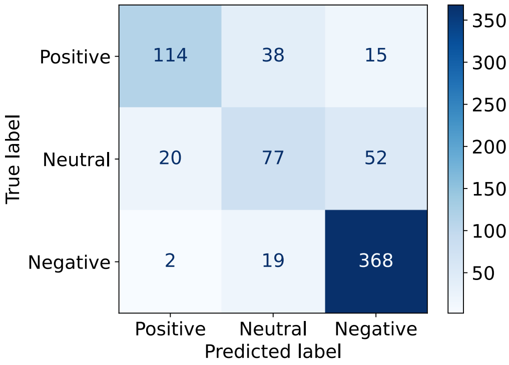


Figure 1: Confusion matrix for the best BERT-based model in Exp.1, bert-base-arabertv02-twitter.

Figure 1 illustrates the confusion matrix for the best-performing model, bert-base-arabertv02-twitter discussed in Exp.1 (subsection 4.1). Overall, the model demonstrates strong performance, showcasing high accuracy in predicting the Negative class (95%) while achieving lower accuracy in predicting the Positive class (68%) and the Neutral class (52%).

The data indicates that the model encounters the greatest challenges when differentiating between Positive and Neutral instances, as well as when distinguishing between Neutral and Negative instances. This is evident from a relatively high number of misclassifications within these categories. These errors can likely be attributed to the restricted size of the training data available for these classes,

Class	%	Examples
Positive	21.3	• اي والله امان ياملكنا سلمان اعز الله بک الدين والاوطان • الهيئة طوق نجاة للأمة من تمسک بها نجى وأفلح • ابشروا بالخير
Negative	60.1	• ابلشونا بسماجهة مواضيعهم ونقاشاتهم الي مالها داعي • اتخيلني اسوق فالرياض بايعة عمري • ردو لي فلوسي حسبي الله من جفره لدحديره
Neutral	18.6	• تنويع مصادر الدخل امر ضروري • حراك وطني اقتصادي اجتماعي ثقافي هو انعكاس لبرنامج يتحدث عن المستقبل وفق رؤية استراتيجية تتسم بالشمولية العامه • ما قال شي غلط

Table 6: Statistics and example of tweets from the SDTC dataset (Al-Thubaity et al., 2018a).

as well as the difficulties faced by annotators during the process of labeling SDTC. Augmenting the training dataset with extra examples from both the Positive and Neutral classes could potentially enhance the accuracy of predicting these categories more precisely.

D Examples of Generated Tweets

Table 7 illustrates one example from each class generated by each language model for zero-shot scenarios discussed in Exp.3 in subsection 4.3.

Table 8 showcases examples where the two annotators answered each question with “No”.

Model	Examples
Bard AI	سعادة لا توصف بالحصول على وظيفة أحلامي Unbelievably happy to have my dream job مليت من الازدحام في شوارع الرياض Tired of crowding in the streets of Riyadh خلصت شغلي اليوم، الحين بروح أريح I finished my work today, now I’m going to rest
GPT-3.5	شاركت في فعالية تطوعية اليوم وشعرت بفخر كبير لخدمة مجتمعي. I participated in a volunteering event today and felt very proud to serve my community. تعبت من هالروتين اليومي البائس، لا متعة ولا فرحة في شيء! Tired of the miserable daily routine, there is no fun or joy in anything! الشاي الدافئ في الصباح يجعل يومي أفضل. Warm tea in the morning makes my day better.
GPT-4	أعشق التسوق في سوق الرياض، الباعة ودودون جداً وجودة البضائع عالية. I love shopping in the Riyadh market; the sellers are very friendly, and the quality of the goods is high. كل يوم أفقد الامل أكثر ولكن أصبر وأصبر ولا أرى طريق إلى السعادة. Every day, I lose hope more, but I am patient and patient, and I do not see a path to happiness. الحكومة السعودية تعمل بجد لتحسين البيئة التعليمية وذلك من خلال تحديث المناهج وتقديم التقنيات الحديثة. The Saudi government is working hard to improve the educational environment by updating curricula and introducing modern technologies.

Table 7: Examples from each class generated by each language model for zero-shot scenarios. The order of examples: positive, negative, and neutral.

Model	Questions	Example
Bard AI	Q1	أكل الكبسة السعودية من ألذ الأكلات في العالم. Eating Saudi Kabsa is one of the most delicious foods in the world.
	Q2	قررت أن أبدأ مشروعاً جديداً. I decided to start a new project.
	Q3	أحس أني عايش في حلم، ما أدري وش الواقع. I feel like I’m living in a dream, I don’t know what the reality is. *Bard classified it as negative, while annotators classified it as neutral.*
	Q1+Q2+Q3	السياسة صعبة كثير. Politics is very difficult.
GPT-3.5	Q1	السفرة كانت كارثة، الأكل طعمه مرة يرفع الراس. تجربة سيئة جداً. The trip was a disaster, the food tasted bitter and raised the head. Very bad experience.
	Q2	قاعد برواق ع الهاش تقلب علي فرح وتسألني لية صفا مع انو جسدك حجر وتدور طول الليل تلف وعليسا عين جنوبية نوفة للفارس الكبير So-and-so, you are a loser and a loser next to you. You cannot do anything right in your life.
	Q3	المشتريات في هذا المول مكلفة جداً Purchases in this mall are very expensive *GPT-3.5 classified it as neutral, while annotators classified it as negative.*
	Q1+Q2+Q3	ويبدو ان الخطأ الاداري هذا تسبب في درجة حرارة البلاد بسنوات تسببت بيها حرارة اخطار المملكة العربية السعودية و الله قلل من نار It seems that this administrative error caused the country’s temperature in the streets for years, caused by the heat of the dangers of the Kingdom of Saudi Arabia, and God reduced the fire of stone
GPT-4	Q1	في الحقيقة، الحده ليله ممتعة ببعض الأحيان، تجلس وحدك وما عندك أي مشاكل و أي شي يشغل بالك… حالة من الراحة و التفكير العميق. #راحة In fact, the intensity is a fun night sometimes, you sit alone and you don’t have any problems or anything to worry about... a state of rest and deep thought. #comfort
	Q2	لديها التعليقات الجديدة على فيديو يوتيوب للشيخ سليمان الدويش، تحدث فيه عن المسائل الدينية اليومية. She has the new comments on a YouTube video by Sheikh Suleiman Ad-Dawish, in which he talks about everyday religious issues.
	Q3	بفضل تكنولوجيا اليوم، يمكننا التواصل مع الأشخاص في جميع أنحاء العالم بفضل لمسة واحدة على شاشات الهواتف الذكية. Thanks to today’s technology, we can communicate with people all over the world thanks to a single touch on our smartphone screens. *GPT-4 classified it as neutral, while annotators classified it as positive.*
	Q1+Q2+Q3	زرت الرياض في نهاية الأسبوع واستمتعت في جولة في القصور والمعابد، كل شيء كان جميل ورائع. #السياحة_في_السعودية I visited Riyadh at the end of the week and enjoyed a tour of the palaces and temples, everything was beautiful and wonderful. #Tourism_in_Saudi Arabia

Table 8: Examples where the two annotators answered each question with “No”.

In-Context Meta-Learning vs. Semantic Score-Based Similarity: A Comparative Study in Arabic Short Answer Grading

Menna Fateen
Kyushu University
menna.fateen@m.ait.kyushu-u.ac.jp

Tsunenori Mine
Kyushu University
mine@ait.kyushu-u.ac.jp

Abstract

Delegating short answer grading to automated systems enhances efficiency, giving teachers more time for vital human-centered aspects of education. Studies in automatic short answer grading (ASAG) approach the problem from instance-based or reference-based perspectives. Recent studies have favored instance-based methods, but they demand substantial data for training, which is often scarce in classroom settings. This study compares both approaches using an Arabic ASAG dataset. We employ in-context meta-learning for instance-based and semantic score-based similarity for reference-based grading. Results show both methods outperform a baseline and occasionally even surpass human raters when grading unseen answers. Notably, the semantic score-based similarity approach excels in zero-shot settings, outperforming in-context meta-learning. Our work contributes insights to Arabic ASAG and introduces a prompt category classification model, leveraging GPT3.5 to augment Arabic data for improved performance.

1 Introduction

Automatic short answer grading (ASAG) has been a prominent subject of discussion in the field of AI in education, studied for more than half a century (Page, 1966). This is not surprising given the potential ASAG systems hold for enhancing various aspects of educational systems. By automating routine grading tasks, teachers can focus more on their unique human role of being motivators of learning and nurturing students' curiosity, ultimately enriching the educational experience (Keller, 1983). The shift toward automation in grading not only enhances efficiency and eliminates human bias but also empowers educators to dedicate their time and expertise to the critical aspects of teaching that require human insight and empathy.

Over the last decade, progress in the field of ASAG has significantly accelerated, driven by advancements in deep learning techniques and the availability of large datasets. ASAG systems can be broadly categorized into two main approaches: instance-based and reference-based (Horbach and Zesch, 2019). The majority of research in ASAG has primarily focused on the instance-based approach, which involves scoring individual student answers independently. On the other hand, reference-based approaches rely on measuring the similarity between the student's response and the reference answer and assigning a score based on this similarity. Reference-based approaches not only have the potential to be more robust to variability but also have the advantage of being more interpretable and less data-hungry (Bexte et al., 2023). However, only a few studies have been conducted comparing the 2 approaches and showing that the performance of reference-based approaches compared to instance-based approaches often yields worse or comparable results (Bexte et al., 2022).

While instance-based approaches have dominated the ASAG landscape, it's important to note that most of this research has been conducted in the context of the English language. English is one of the most widely studied languages, and therefore, a substantial amount of educational content and resources are available for it. However, the need for ASAG systems in other languages, such as Arabic, is equally significant. Even though Modern Standard Arabic (MSA) could also be considered a thriving language (Simons et al., 2022), datasets for ASAG in Arabic are still scarce.

In this study, we hope to contribute to the field of ASAG by presenting a comparison of two distinct approaches to ASAG in Arabic. In our first instance-based approach, we leverage a pre-trained language model (i.e. BERT) and train it on different questions with a shared type. For each instance, we create an input structure that provides contextual information for the model. In our reference-based approach, we train a score-based semantic similar-

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 350–358
December 7, 2023 ©2023 Association for Computational Linguistics

ity model using SentenceTransformers (Reimers and Gurevych, 2019). Our results demonstrate that while both techniques perform similarly in conventional training circumstances, score-based semantic similarity has considerable potential for delivering superior results in zero-shot settings. We additionally propose a "prompt category" classification model to facilitate the selection of the most suitable scoring model for a given question. We show the effectiveness of this model in low-resource settings by augmenting the training data with synthetic examples generated by GPT-3.5. To the best of our knowledge, this is the first study to apply and compare the two distinct approaches to the problem of ASAG in Arabic. Finally, we make the code and models publicly[1] available to facilitate future research in this area.

2 Related Work

Research in ASAG can be categorized into two main paradigms, instance-based or similarity-based methods (Horbach and Zesch, 2019). Most recent ASAG research follows the instance-based paradigm, where algorithms are trained primarily using a large set of student answers to learn about the features of correct and incorrect responses. With the rise of large language models and transfer learning, most studies typically involve fine-tuning BERT such as in the work by Lun et al.. Another example is the work of Nael et al. where they fine-tune BERT and ELECTRA models on a machine-translated ASAP dataset. Condor et al. used SBERT embeddings to train a model with an instance-based approach rather than using it in a similarity-based approach. Fernandez et al. introduced a single shared scoring model for multiple questions using a specified input structure that provides contextual information for each item. Similarly, to score mathematical questions, Zhang et al. use an in-context learning approach that provides scoring examples as part of the input to a MathBERT model to promote generalization.

On the other hand, in the reference-based approach, student answers are evaluated by comparing them to one or more target answers. Judgments of correctness are thus determined based on their similarity to a reference solution. In early work, reference-based approaches mainly employed feature-engineering methods such as utilizing string-based or corpus-based similarity methods (Gomaa and Fahmy, 2014) and n-grams (Shehab et al., 2018). More recently, Meccawy et al. conducted a comparative study evaluating the efficiency of different word embedding approaches for conducting feature vectors. In their study, Wang et al. introduced innovative metrics for score-based similarity to construct a text representation space that is optimized for both inter and intra-level distinctions, leading to improved scoring efficiency. In our reference-based approach, we define score-based similarity in a manner similar to what they have presented in their research.

3 Dataset Description

In this study, we utilize the AR-ASAG dataset, which is the first publicly available dataset for automatic short-answer scoring in Arabic (Ouahrani and Bennouar, 2020). The dataset consists of 2133 short answers written by graduate students in response to 48 questions. The questions are taken from 3 different exams on cybercrime where each exam consists of 16 questions. The question prompts in the exams could be classified into 5 categories based on the type of answer they expect, namely: *define*, *explain*, *consequences*, *justify*, and *compare*.

The answers in this dataset were independently annotated by two human raters on a scale of 0 to 5 where 0 is completely incorrect and 5 is considered a perfect answer. The raters were instructed to assign a score based on the similarity of the student's answers to a reference answer given for each question. Determining the similarity between two answers not only is a subjective task but also requires a deep understanding of the topic. In cases like this, where no detailed scoring rubric is provided, the raters can find it especially difficult to determine the precise degree of similarity. This is reflected in the low inter-rate agreement of 35%. However, this is expected since the raters were also given the freedom to assign intermediate scores such as 4.5 or 3.25, etc.

In our study, we treat the scoring problem as a classification problem instead of a regression one. We discretize the scores into 6 categories, 0, 1, 2, 3, 4, and 5 by taking the rounded-down median after ceiling the scores to the nearest 0.5. This is done to increase the inter-rate agreement to 56% instead of 35%. The distribution of the scores in the dataset can be seen in Figure 1 where we can observe the

[1] Our code can be found at: `https://github.com/mennafateen/SSS-vs-InCML`

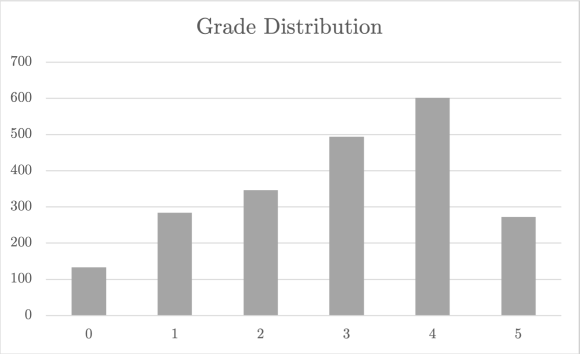


Figure 1: Distribution of the scores in the dataset.

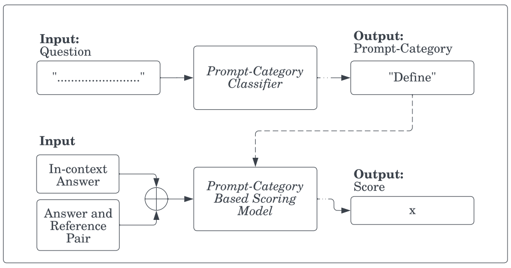


Figure 2: Overview of the in-context prompt-based scoring framework.

majority of the scores being concentrated in the range of 3 to 4.

4 Methodology

Problem Formulation

We formulate the problem of automatic short answer scoring as follows: Given a question q, a short answer a, and a reference answer a^*, the goal is to predict a score $s \in [0, 5]$ that represents the quality of the answer. Usually, each question is treated as a separate task where a separate model is trained for each task or question. However, this approach is not feasible in low-resource settings where there is a lack of annotated data. Hence, we propose a general in-context prompt-based scoring framework for automated scoring of short-answer questions where we divide the scoring problem into two sub-problems, prompt-category-based scoring and prompt category classifying.

The prompt-category-based scoring problem can be formally defined as follows: we have a set of tasks $T = \{t_i\}_{i=1}^{N}$, where each task t_i is defined by a question q_i, its reference answer a_i^*, and a set of instances $D_i = \{(a_{i,j}, s_{i,j})\}_{j=1}^{M_i}$, where M_i is the number of instances for the i-th task. The goal is to learn a function $f : (q, a^*, a) \rightarrow s$ that can generalize to both unseen answers and unseen questions with a small number of annotated examples. To solve the defined problem, we propose and compare two main approaches, namely, *in-context meta-learning* (InCML) and *score-based semantic similarity* (SSS). We describe each approach in detail in the following subsections. Within each approach, we train one model per prompt category, resulting in 5 models per approach.

In order to facilitate the selection of the most suitable prompt-category-based scoring model for a given question, an auxiliary prompt category classifier is trained to identify the prompt category of a given question. The output of this model should then serve as a guide for selecting the most suitable model for a given question. This is illustrated in Figure 2.

Prompt Category Classification

To construct a balanced training dataset, we train the model on the first 4 questions from each prompt category and set aside the remaining questions for testing. We utilize the SetFit framework (Tunstall et al., 2022) and use the pretrained AraBERT model (Antoun et al.) as the pretrained body. We then generate 50 pairs for contrastive learning and train the model for 5 epochs. The classification head is a logistic regression layer that takes the output of the last layer of the pretrained body as input. With this 4-shot training setup, the model achieves a mere accuracy of 0.357. To address this, we propose to augment the training data with synthetic examples generated by GPT-3.5. For each prompt category, we instruct the model to provide x more examples, for instance:

Provide five more examples that are similar to the following using the same "Define" prompt:

- عرف مصطلح الجريمة الإلكترونية (Define the term cybercrime)
- عرف مصطلح أمن المعلومات (Define the term information security)
- عرف مصطلح الهندسة الاجتماعية النفسية (Define the term psychological social engineering)
- عرف مصطلح تبييض أو غسيل الأموال (Define the term money laundering).

Table 1: Prompt category classification results on the test set with the different number of augmented examples.

	Original	Aug$_1$	Aug$_2$	Aug$_3$
Acc	0.357	0.607	0.714	**0.893**
F1	0.443	0.645	0.699	**0.821**
Precison	0.511	0.711	0.733	**0.82**
Recall	0.724	0.806	0.841	**0.90**

We experiment with different values of x and report the results in Table 1 where in Aug_1, Aug_2, and Aug_3, we augment the training data so that the number of examples per prompt category is 45, 125, and 250 respectively. As shown in the table, the model's performance significantly improves as we increase the number of augmented examples, achieving an accuracy of 0.893 with Aug_3 and an F1-score of 0.821. With this prompt classification model experiment, we show that in low-resource settings, a potential solution that could be explored is to augment available samples using generative large language models such as GPT-3.5.

Instance-based: In-Context Meta Learning Model

The in-context meta-learning model (InCML) approach draws inspiration from the work of (Fernandez et al., 2022). Building upon their foundational concepts, we apply this approach to the unique domain of cybersecurity short answer scoring in Arabic. To introduce context, we input the answers using a template that is constructed by concatenating the target answer to be scored a_j, question q_i, and its reference answer, a_i^*. We additionally include a set of K in-context examples E_i that are randomly sampled from the training set D_{train} for the i-th task or question. We build a template for each component by adding semantically meaningful task instructions as shown in Table 2. Moreover, we convert the numeric scores s_j in the in-context examples E_i to meaningful words such that: *0: ضعيف جدا (very poor), 1: ضعيف (poor), 2: متوسط (fair), 3: جيد (good), 4: جيد جدا (very good), 5: ممتاز excellent*. We then concatenate the templates to form the final input to the model. During inference time, the same templates are created for the input components where the in-context examples are fetched from the training set for seen questions only.

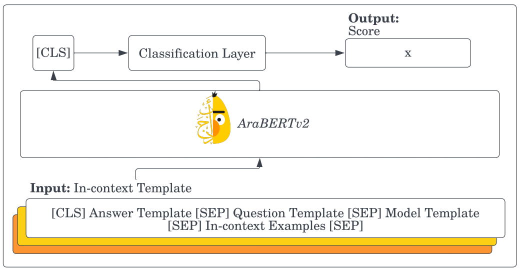

Output:
Score
[CLS]
Classification Layer
x
AraBERTv2
Input: In-context Template
[CLS] Answer Template [SEP] Question Template [SEP] Model Template [SEP] In-context Examples [SEP]

Figure 3: in-context meta-learning model

We train our model on the union of training datasets for all items or questions per prompt category $\cup_{i=1}^{5} D_{train}^{i}$, instead of training a separate model for each item, thus reducing the number of model parameters and required storage space. Figure 3 illustrates the in-context meta-learning model.

Reference-based: Score-based Semantic Similarity

When human experts are asked to score answers to open-ended questions, they usually compare the answers to a reference answer and assign a score based on the similarity between the two answers. In this approach, we propose to train a model that can mimic this process by learning to assign a score to a given answer based on its similarity to a reference answer.

To achieve this, we perform the following steps: First, we construct a simple in-context template for each answer to be graded a_j by prepending the question q_i to the answer. It has been shown that incorporating the question in the input can improve the performance of ASAG models (Lv et al., 2021). Then, we define score-based similarity as follows: Given a pair of answers a_x and a_y with their scores s_x and s_y, the similarity between the two answers is defined as:

$$sim(a_x, a_y) = \frac{s_x}{s_y}; (s_x \leq s_y) \tag{1}$$

For each task/question i, we then construct a dataset $\mathcal{D}_i$ of answer pairs annotated together via the score-based metric indicating their similarity as shown in Equation 2, where X is the number of examples per score k and a_k is a student answer that was graded k.

$$\mathcal{D}_i = \{\{(a_k, a_{\neg k}, sim(a_k, a_{\neg k}))\}_{x=0}^{X}\}_{k=0}^{5} \tag{2}$$

Table 2: Input components and templates for the in-context meta-learning model.

Input Component	**Template**	**Sample**
Student Answer	قيم هذه الإجابة: x	قيم هذه الإجابة: العلم الذي يستخدم التحليل الاحصائي لصفات الانسان الحيوية وذلك للتأكد من هويته الشخصية (Grade this answer: The science that uses statistical analysis of a person's vital characteristics to confirm his identity)
Question	السؤال: q_i	السؤال: عرف مصطلح القياس الحيوي (Question: Define the term biometrics)
Reference	النموذج: a_i^*	النموذج: هو العلم الذي يستخدم التحليل الإحصائي لصفات الإنسان الحيوية وذلك للتأكد من هويته الشخصية بإستخدام صفاته الفريدة وهي صفات سلوكية وصفات فيزيائية (Reference: It is the science that uses statistical analysis of a person's vital characteristics to confirm his identity using his unique characteristics, which are behavioral characteristics and physical characteristics.)
Grades	تقييم : ...	تقييم : ضعيف جدا ضعيف متوسط جيد جيد جدا ممتاز (Grades: very poor, poor, fair, good, very good, excellent)
Examples	مثال: $x_{\neg j}$	مثال: هو علم يدرس حالة الإنسان الفريدة التي تميز شخصًا عن آخر تقييم : ضعيف (Example: It is a science that studies the unique human condition that distinguishes one person from another Grade: Weak)

We experiment with 3 different settings of X when constructing the dataset with $X = 30$, $X = 50$, and a final configuration where we account for the distribution of the scores in the training set so that the number of samples per score is $50\frac{N_k}{\sum_{k=0}^{5} N_k}$

We then train one model on the union of training datasets for all items or questions per prompt category $\cup_{i=1}^{5} D_{train}^{i}$, instead of training a separate model for each item as described in the InCML approach. In this approach, using SBERT, we fine-tune a pretrained AraBERT model through a Siamese network structure where we train the model using a cosine-similarity loss function. For each answer pair in the union dataset, we pass both answers through the model which generates an embedding u and v for each answer. The gold similarity score is then compared with the cosine similarity between the generated embeddings. Figure 4 illustrates the score-based semantic similarity model.

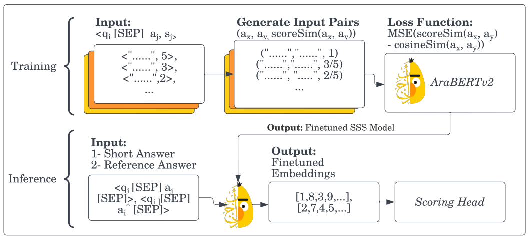


Figure 4: Score-based semantic similarity model

5 Experimental Results

Evaluation Metrics

To evaluate the performance of the proposed approaches, we use two metrics, namely, quadratic weighted kappa (QWK) and percentage of tick accuracy (PTA). QWK is a commonly used metric in ASAG that measures the agreement between two raters. In (Williamson et al., 2012), the authors suggest that the QWK between automated and human scoring should be at least 0.7 on datasets with normal distribution to be considered acceptable. Percentage of Tick Accuracy (PTA$_x$) measures the percentage of answers that are scored correctly or within x points of the gold score. PTA$_0$ would be equivalent to accuracy while PTA$_1$ also includes answers that are scored within 1 point of the gold score (e.g. 3 is considered correct if the gold score is 2 or 4) and so on.

Experimental Setup

We use AraBERT as the pretrained body for both approaches. In the InCML approach, we use the Adam optimizer with a learning rate of 1e-5 and a batch size of 8 and train the models for 6 epochs. Similarly, in SSS, we train the model for 6 epochs but use a batch size of 16 instead.

Results

We undertake two experiments to evaluate the performance of the proposed approaches. In the first experiment, we test the models' performance on unseen answers. We set aside 10% of the answers from each prompt category for testing. In the second experiment, we evaluate the performance of the models on unseen questions. We set aside the first question and its answers from each prompt category for testing. This setting is considered a zero-shot learning scenario since the models are not trained on any examples from the test set. The results of both experiments are shown in Table 3. As a baseline for comparison, we report the results of a majority class classifier and the QWK and PTA between the rounded-up grades of the two human raters.

6 Discussion

6.1 Unseen Answers

As shown in Table 3, compared to the majority class classifier, both approaches with different configurations outperform the baseline in all prompt categories in the unseen answers experiments. Comparing the model's performance to human performance, we observe that with prompts $P1$ and $P3$, InCML$_0$ and InCML$_3$ outperform the human raters in terms of QWK. In terms of PTA$_0$, InCML$_1$, InCML$_3$, and additionally SSS$_{50}$ outperform the human raters with prompt $P1$ while InCML$_1$ again outperforms the human raters with prompt $P3$ type questions. In the remaining prompt categories, the performance of both approaches in terms of QWK is marginally below the QWK achieved between the human raters with the in-context meta-learning approach showing a tendency to outperform the score-based similarity approach.

In the in-context meta-learning approach, we observe that the performance of the model does not necessarily improve as we increase the number of in-context examples. In fact, in some cases, the performance decreases. We speculate that this might be attributed to potential overfitting on the in-context examples. It is also important to note that the performance of InCML fluctuates depending on the in-context examples that are extracted from the training set which introduces inherent instability. On the other hand, in the case of the semantic score-based similarity approach, an increase in the number of examples per score generally corresponds to improved model performance.

In the unseen answers experiment, with a few training examples, we observe that while both approaches have comparable performance to the human raters, the instance-based in-context meta-learning approach generally gives better performance compared to the reference-based approach.

6.2 Unseen Questions

In Table 3, we see that the overall performance of the models in the unseen questions experiment, or in zero-shot settings, is lower than the performance of unseen answers. However, we observe that the PTA$_0$ of the models is still higher than the majority class classifier in most prompt models using our reference-based, SSS$_{30}$, SSS$_{50}$ and SSS$_{50W}$ methods.

Compared to the instance-based InCML approach, it is evident that the reference-based SSS approach proposed gives higher performance showcasing its reduced data hunger advantage and its ability to generalize to new questions.

7 Conclusion

In this paper, we propose a general in-context prompt-based scoring framework for automated scoring of short-answer questions. We divide the scoring problem into two sub-problems, namely, prompt category classification and prompt-category-based scoring. For prompt-category classification, we utilize a few-shot, prompt-free framework to train the model. We also show that with data augmentation using GPT3.5, the performance could be significantly increased. We then propose two main approaches for the prompt-category-based scoring problem, namely, instance-based in-context meta-learning and reference-based semantic similarity. Utilizing the only publicly available Arabic ASAG dataset, we evaluate both approaches in their ability to generalize to unseen answers and unseen questions. Experimental results show that both proposed approaches outperform the majority class classifier and are comparable to human raters when grading unseen answers. However, the performance is highly prompt-dependent and no particular approach is consistently better than the other. In zero-shot settings, when generalizing to unseen questions, we observe a tendency for the reference-based semantic similarity approach to outperform the instance-based in-context meta-learning approach. We thus believe that in class-

Table 3: Experimental results.

			Human	**MV**	**InCML$_0$**	**InCML$_1$**	**InCML$_3$**	**SSS$_{30}$**	**SSS$_{50}$**	**SSS$_{50W}$**
Unseen Answers	***P1***	**QWK**	*0.676*		0.611	0.656	**0.697**	0.591	0.593	0.632
		PTA$_0$	*0.357*	0.357	0.286	**0.500**	0.404	0.357	0.393	0.357
		PTA$_1$	*0.893*		0.857	0.843	0.889	0.857	0.857	**0.893**
		PTA$_2$	*0.929*		0.964	0.954	0.964	**1.000**	**1.000**	**1.000**
		PTA$_3$	*1.000*		**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
	P2	**QWK**	*0.788*		0.722	0.668	0.760	0.718	**0.763**	0.718
		PTA$_0$	*0.568*	0.239	0.432	0.443	**0.451**	0.375	0.443	0.364
		PTA$_1$	*0.920*		0.909	0.875	0.936	0.932	**0.955**	0.943
		PTA$_2$	*0.977*		0.989	0.963	**0.998**	0.989	0.989	0.977
		PTA$_3$	*1.000*		**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
	P3	**QWK**	*0.749*		**0.798**	0.774	0.727	0.557	0.581	0.660
		PTA$_0$	*0.385*	0.308	0.385	**0.542**	0.385	0.154	0.385	0.385
		PTA$_1$	*0.846*		**0.923**	0.869	0.919	0.846	0.846	0.885
		PTA$_2$	*0.962*		**1.000**	**1.000**	0.950	**1.000**	0.962	0.962
		PTA$_3$	*1.000*		**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
	P4	**QWK**	*0.666*		0.602	**0.649**	0.519	0.614	0.613	0.515
		PTA$_0$	*0.533*	0.311	0.400	0.464	0.351	**0.467**	**0.467**	0.444
		PTA$_1$	*0.822*		0.867	0.813	0.733	**0.911**	**0.911**	0.844
		PTA$_2$	*0.978*		**1.000**	0.989	0.936	**1.000**	**1.000**	**1.000**
		PTA$_3$	*1.000*		**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
	P5	**QWK**	*0.716*		**0.696**	0.000	0.070	0.529	0.606	0.405
		PTA$_0$	*0.526*	0.421	0.421	0.421	0.368	**0.474**	0.421	0.368
		PTA$_1$	*0.842*		**0.842**	0.684	0.737	0.789	**0.842**	0.789
		PTA$_2$	*0.947*		**1.000**	0.789	0.842	**1.000**	**1.000**	**1.000**
		PTA$_3$	*1.000*		**1.000**	0.947	0.947	**1.000**	**1.000**	**1.000**
Unseen Questions	***P1***	**QWK**	*0.743*		0.145	0.000	0.063	0.620	0.599	**0.627**
		PTA$_0$	*0.596*	0.383	0.213	0.000	0.064	**0.447**	0.404	**0.447**
		PTA$_1$	*0.957*		0.553	0.043	0.149	**1.000**	**1.000**	**1.000**
		PTA$_2$	*1.000*		0.936	0.085	0.404	**1.000**	**1.000**	**1.000**
		PTA$_3$	*1.000*		0.979	0.319	0.766	**1.000**	**1.000**	**1.000**
	P2	**QWK**	*0.876*		0.000	-0.055	0.026	**0.645**	0.558	**0.645**
		PTA$_0$	*0.833*	0.416	0.167	0.083	0.167	**0.500**	**0.500**	**0.500**
		PTA$_1$	*0.916*		0.167	0.167	0.167	**0.917**	0.833	**0.917**
		PTA$_2$	*1.000*		0.417	0.417	0.333	**1.000**	**1.000**	**1.000**
		PTA$_3$	*1.000*		0.833	0.417	0.833	**1.000**	**1.000**	**1.000**
	P3	**QWK**	*0.752*		0.000	-0.014	0.000	0.488	0.446	**0.495**
		PTA$_0$	*0.714*	**0.469**	0.082	0.082	0.082	0.204	0.224	0.204
		PTA$_1$	*0.918*		0.184	0.184	0.184	**0.918**	0.878	**0.918**
		PTA$_2$	*0.959*		0.347	0.347	0.347	**1.000**	**1.000**	**1.000**
		PTA$_3$	*0.980*		0.531	0.531	0.531	**1.000**	**1.000**	**1.000**
	P4	**QWK**	*0.774*		0.101	-0.015	-0.161	0.254	0.284	0.365
		PTA$_0$	*0.395*	0.271	0.208	0.125	0.042	**0.333**	**0.333**	**0.333**
		PTA$_1$	*0.875*		0.500	0.292	0.375	0.688	**0.708**	0.688
		PTA$_2$	*0.979*		0.646	0.375	0.771	0.938	0.938	**1.000**
		PTA$_3$	*1.000*		0.896	0.646	0.958	**1.000**	**1.000**	**1.000**
	P5	**QWK**	*0.358*		0.000	**0.069**	0.003	0.000	0.029	0.024
		PTA$_0$	*0.667*	**0.500**	0.000	0.146	0.000	0.042	0.042	0.021
		PTA$_1$	*0.979*		0.000	0.396	0.021	0.500	**0.563**	0.500
		PTA$_2$	*1.000*		0.000	0.563	0.188	**1.000**	**1.000**	**1.000**
		PTA$_3$	*1.000*		0.042	0.604	0.667	**1.000**	**1.000**	**1.000**

room settings, the reference-based semantic similarity approach could be a more suitable solution due to its superiority in zero-shot settings.

Limitations

In this paper, we presented a comparison between a specific instance-based and reference-based approach, thus our findings are limited to these methods and cannot be generalized to different methods. This study was also limited to a prompt-category-based scoring framework and while preliminary experiments were conducted, we did not compare with specific prompt-based models or cross-prompt-category models for a more straightforward and comprehensible comparison. Due to the scarcity of resources, our comparison also relies on a specific dataset, which does not encompass the full diversity of responses or topics encountered in a real-world educational setting. Furthermore, since we utilize an Arabic dataset, we adapted a BERT model pre-trained on Arabic data but have not presented a comparison with a language-agnostic model. Finally, while we briefly touched upon the potential of reference-based approaches in offering explainability, we have not delved into the topic of interpretability of the provided models. Understanding why a model assigns a specific score to an answer is essential for educational applications, as it can provide valuable feedback to students, however, it is beyond the scope of this paper and is left for future work.

Acknowledgements

This work was partially supported by JST, the establishment of university fellowships towards the creation of science technology innovation, Grant Number JPMJFS2132, and KAKENHI No. JP20H01728, JP21H00907, and JP23H03511.

References

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

Marie Bexte, Andrea Horbach, and Torsten Zesch. 2022. Similarity-based content scoring - how to make sbert keep up with bert. In *Proceedings of The 17th Workshop on Innovative Use of NLP for Building Educational Applications*.

Marie Bexte, Andrea Horbach, and Torsten Zesch. 2023. Similarity-based content scoring-a more classroom-suitable alternative to instance-based scoring? In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1892–1903.

Aubrey Condor, Max Litster, and Zachary Pardos. 2021. Automatic short answer grading with sbert on out-of-sample questions. *International Educational Data Mining Society*.

Nigel Fernandez, Aritra Ghosh, Naiming Liu, Zichao Wang, Benoît Choffin, Richard Baraniuk, and Andrew Lan. 2022. Automated scoring for reading comprehension via in-context bert tuning. In *International Conference on Artificial Intelligence in Education*, pages 691–697. Springer.

Wael Hassan Gomaa and Aly Aly Fahmy. 2014. Automatic scoring for answers to arabic test questions. *Computer Speech & Language*, 28(4):833–857.

Andrea Horbach and Torsten Zesch. 2019. The influence of variance in learner answers on automatic content scoring. In *Frontiers in Education*, volume 4, page 28. Frontiers Media SA.

John M Keller. 1983. Motivational design of instruction. *Instructional design theories and models: An overview of their current status*, 1(1983):383–434.

Jiaqi Lun, Jia Zhu, Yong Tang, and Min Yang. 2020. Multiple data augmentation strategies for improving performance on automatic short answer scoring. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13389–13396.

Gaoyan Lv, Wei Song, Miaomiao Cheng, and Lizhen Liu. 2021. Exploring the effectiveness of question for neural short answer scoring system. In *2021 IEEE 11th International Conference on Electronics Information and Emergency Communication (ICEIEC) 2021 IEEE 11th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, pages 1–4. IEEE.

Maram Meccawy, Afnan Ali Bayazed, Bashayer Al-Abdullah, and Hind Algamdi. 2023. Automatic essay scoring for arabic short answer questions using text mining techniques. *International Journal of Advanced Computer Science and Applications*, 14(6).

Omar Nael, Youssef ELmanyalawy, and Nada Sharaf. 2022. Arascore: a deep learning-based system for arabic short answer scoring. *Array*, 13:100109.

Leila Ouahrani and Djamal Bennouar. 2020. Ar-asag an arabic dataset for automatic short answer grading evaluation. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2634–2643.

Ellis B Page. 1966. The imminence of... grading essays by computer. *The Phi Delta Kappan*, 47(5):238–243.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Abdulaziz Shehab, Mahmoud Faroun, and Magdi Rashad. 2018. An automatic arabic essay grading system based on text similarity algorithms. *International Journal of Advanced Computer Science and Applications*, 9(3).

Gary F Simons, Abbey L Thomas, and Chad K White. 2022. Assessing digital language support on a global scale. *arXiv preprint arXiv:2209.13515*.

Lewis Tunstall, Nils Reimers, Unso Eun Seo Jo, Luke Bates, Daniel Korat, Moshe Wasserblat, and Oren Pereg. 2022. Efficient few-shot learning without prompts. *arXiv preprint arXiv:2209.11055*.

Bo Wang, Billy Dawton, Tsunenori Ishioka, and Tsunenori Mine. 2023. Optimizing answer representation using metric learning for efficient short answer scoring. *The 20th Pacific Rim International Conference on Artificial Intelligence (PRICAI)*.

David M Williamson, Xiaoming Xi, and F Jay Breyer. 2012. A framework for evaluation and use of automated scoring. *Educational measurement: issues and practice*, 31(1):2–13.

Mengxue Zhang, Sami Baral, Neil Heffernan, and Andrew Lan. 2022. Automatic short math answer grading via in-context meta-learning. *arXiv preprint arXiv:2205.15219*.

SALMA: Arabic Sense-Annotated Corpus and WSD Benchmarks

Mustafa Jarrar, Sanad Malaysha, Tymaa Hammouda, Mohammed Khalilia
Birzeit University, Palestine
{mjarrar, smalaysha, thammouda, mkhalilia}@birzeit.edu

Abstract

SALMA, the first Arabic sense-annotated corpus, consists of ~34K tokens, which are all sense-annotated. The corpus is annotated using two different sense inventories simultaneously (Modern and Ghani). SALMA novelty lies in how tokens and senses are associated. Instead of linking a token to only one intended sense, SALMA links a token to multiple senses and provides a score to each sense. A smart web-based annotation tool was developed to support scoring multiple senses against a given word. In addition to sense annotations, we also annotated the corpus using six types of named entities. The quality of our annotations was assessed using various metrics (Kappa, Linear Weighted Kappa, Quadratic Weighted Kappa, Mean Average Error, and Root Mean Square Error), which show very high inter-annotator agreement. To establish a Word Sense Disambiguation baseline using our SALMA corpus, we developed an end-to-end Word Sense Disambiguation system using Target Sense Verification. We used this system to evaluate three Target Sense Verification models available in the literature. Our best model achieved an accuracy with 84.2% using Modern and 78.7% using Ghani. The full corpus and the annotation tool are open-source and publicly available at https://sina.birzeit.edu/salma/.

1 Introduction

WSD aims to determine a word's intended meaning (sense) in a given context. WSD is underdeveloped in Arabic due to the lack of sense-annotated datasets. This is in addition to the challenging nature of the WSD task due to the semantic polysemy of the words (Al-Hajj and Jarrar, 2021). For instance, the Arabic word (عَين *ʿayn*) has sixteen meanings in the Contemporary Arabic Dictionary (Omar, 2008). In the context (رأيتُه رأي العَين *rʾaytuh rʾay ālʿayn*), word (عَين *ʿayn*) refers to *eye*, while in (شرِبت مِن عَين الماء *šribt min ʿayn ālmāʾ*), it refers to *water spring*. Similarly, the English word *book* as a noun has ten different senses in Princeton WordNet (Miller et al., 1990), such as (a written work or composition that has been published), or (number of pages bound together). WSD has been considered a challenging task for many years (Weaver, 1949/1955), but it has recently gained more attention due to the advances in learning contextualized word representations from language models, such as BERT (Devlin et al., 2019) and GPT (Radford et al., 2018).

As glosses are short descriptions of senses (Jarrar, 2006, 2005), recent research has demonstrated promising results in WSD task by framing the problem as a sentence-pair (context-gloss) binary classification task, referred to as Target Sense Verification (TSV), where the context is a sentence containing the ambiguous word (Huang et al., 2019; Yap et al., 2020; Blevins and Zettlemoyer, 2020). Al-Hajj and Jarrar (2021) proposed an approach for Arabic WSD (using TSV) based on context-gloss pairs extracted from the Arabic Ontology and lexicons and they achieved 84% accuracy, but this evaluation was done on a TSV dataset rather than a WSD evaluation using a sense-annotated corpus. Additionally, Al-Hajj and Jarrar (2021) presented an attempt for Arabic Word-in-Context (WiC) disambiguation using the dataset provided by the SemEval shared task (Martelli et al., 2021).

This article presents SALMA, the first sense-annotated Arabic corpus consisting of about 34K tokens, which are manually annotated with senses. Since there are no available sense inventories for Arabic, We used two Arabic lexicons as sense inventories: Contemporary Arabic Dictionary (اللغة العربية المعاصرة *ālliġh ālʿrbyh ālmʿāṣrh*), hereafter we refer to as **Modern** (Omar, 2008), and Al-Ghani Al-Zaher (الغني الزاهر *ālġny ālzāhr*), hereafter we refer to as **Ghani** (Abul-Azm, 2014). These two lexicons are part of the lexicon digitization project and lexicographic database at SinaLab[1] (Jarrar and Amayreh, 2019; Alhafi et al., 2019; Amayreh et al., 2019; Ghanem et al., 2023; Jarrar et al., 2021). We introduce a novel sense-annotation framework (Section 3), in which all candidate senses, from both lexicons, are scored to indicate their semantic

[1] https://sina.birzeit.edu/

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 359–369
December 7, 2023 ©2023 Association for Computational Linguistics

relatedness to a token appearing within a context. The higher the score, the more semantically related the sense is. For better coverage, we annotated each token in our corpus using both lexicons independently and in parallel. The scores assigned to senses of the Modern do not influence the scoring of the Ghani senses. In addition, we also annotated our corpus using six types of named entities: person (PERS), organization (ORG), geopolitical entity (GPE), location (LOC), facility (FAC), and currency (CURR). The corpus was annotated by three linguists and we assessed the inter-annotator agreement (IAA) using 2.6% of the annotated words in the corpus. To establish a baseline for WSD in Arabic, we developed an end-to-end WSD system, in which we benchmarked three available TSV models, with different settings. The best model resulted in 84.2% accuracy using Modern and 78.7% using Ghani. The main contributions of this paper are:

- *Sense-annotated corpus*, annotated with two sense inventories independently, and six named entities; and most importantly, each word is linked with all of its senses, and each sense is given a score.
- *Web-based sense-annotation framework* to score all senses of a given word.
- *End-to-end WSD system*, implemented and evaluated using three different TSV models.
- *WSD baseline for Arabic*, with different settings.

The remainder of the article is organized as follows: Section 2 highlights the related work, Section 3 presents the corpus, Section 4 describes the inter-annotator agreement, Sections 5 and 6 present how the baselines are produced, we conclude in Section 7 and outline the limitations and future work in Section 8.

2 Related Work

We will first review related sense-annotated corpora, then we will review related sense inventories.

One of the known English sense-annotated corpora is SemCor (Miller et al., 1993), which is annotated using the Princeton WordNet (Miller et al., 1990). It contains about 200K sense annotations for around 700K words, but not all words are sense-annotated in the SemCor corpus, especially multi-word expressions, articles, and prepositions. The AnCora corpus for Spanish and Catalan languages (Taulé et al., 2008) was collected from newspapers and consists of 500K words, but only 200K noun words are semantically annotated using the Spanish WordNet. AnCora also includes morphological, semantic, and syntactic annotations. TuBa-D/Z is a German annotated corpus, manually collected from newspapers and annotated using the GermanNet senses (Telljohann et al., 2004). TuBa-D/Z was later used as a gold standard for the WSD task by (Petrolito and Bond, 2014). The Italian Syntactic-Semantic Treebank (ISST) is a corpus built for the Italian language with 89,941 sense-annotated words (Montemagni and Venturi, 2003). The ISST annotations cover five levels that are related to lexico-semantics such as orthographic, morpho-syntactic, semantic, and syntactic aspects.

The NTU-MC corpus (Tan and Bond, 2012) covers eight languages including Thai, Vietnamese, Arabic, Korean, Indonesian, Japanese, Mandarin Chinese, and English. However, the Arabic version is not publicly available. This corpus was collected from short stories, essays, and tourism articles resulting in a total of 116K words, but only 63K words are annotated. KPWr, a Polish corpus, contains text from multiple domains including science, law, religion, and press (Broda et al., 2012) with a total of 438,327 words, but only 9,157 words are annotated using the Polish WordNet (Maziarz et al., 2012).

For Arabic, the focus of research has been primarily on developing corpora for morphological and syntactic tagging (Darwish et al., 2021) rather than semantic and sense annotation, as noted by Elayeb (2019) and Naser-Karajah et al. (2021). For instance, part of the OntoNotes corpus (Weischedel et al., 2013) covers limited semantic annotations for Arabic using a small sense inventory of size 261 senses (150 verbs and 111 nouns). Additionally, AQMAR corpus (Schneider et al., 2012) is annotated with 25 super-sense labels representing broad semantic fields such as ARTIFACT and PERSON, which can be considered as general types of named entities, rather than word-sense annotations. They annotated ~22K nouns out of 65K tokens corpus. Table 1 compares our proposed corpus and related Arabic resources.

In addition to the lack of sense-annotated corpora, Arabic lacks reliable sense inventories. Although there are some available semantic resources, they are not mature enough to be used as sense

Corpus	Unique Senses	Annotation Type	Corpus Size (tokens)	Annotations				
				Nouns	Verbs	Func. Words	Punc.+ Digits	Total
AQMAR	25 semantic fields (closer to named entities)	selected words each one sense	65K	~22K	–	–	–	~22K
OntoNotes5	261 semantic fields (high-level grouped senses)	selected words each one sense	300K	8,700	4,300	–	–	13K
SALMA (ours)	4,151 word senses (from each sense inventory) 6 types of named entities	all senses of all words	34K	**19,030**	**2,763**	**7,116**	**5,344**	**34,253**

Table 1: Overview of related Arabic sense-annotated corpora.

inventories. For example, the Arabic WordNet (Black et al., 2006) contains about 10K senses, and the Arabic Ontology (Jarrar, 2021, 2011) contains about 18K synsets. However, both resources cannot be used as sense inventories as they do not provide a complete set of senses for a given lemma (i.e., lexicon entry). The lexicographic database developed at Birzeit University contains about 150 Arabic lexicons (Jarrar and Amayreh, 2019; Jarrar et al., 2019), but these lexicons are not well-structured or suitable to be used as sense inventories (Jarrar and Amayreh, 2019). Due to the lack of dependable Arabic sense inventory, we decided to obtain a license to digitize and use two Arabic lexicons as sense inventories, namely, **Modern** (Omar, 2008) and **Ghani** (Abul-Azm, 2014).

3 Corpus Construction and Annotation

3.1 Corpus Collection

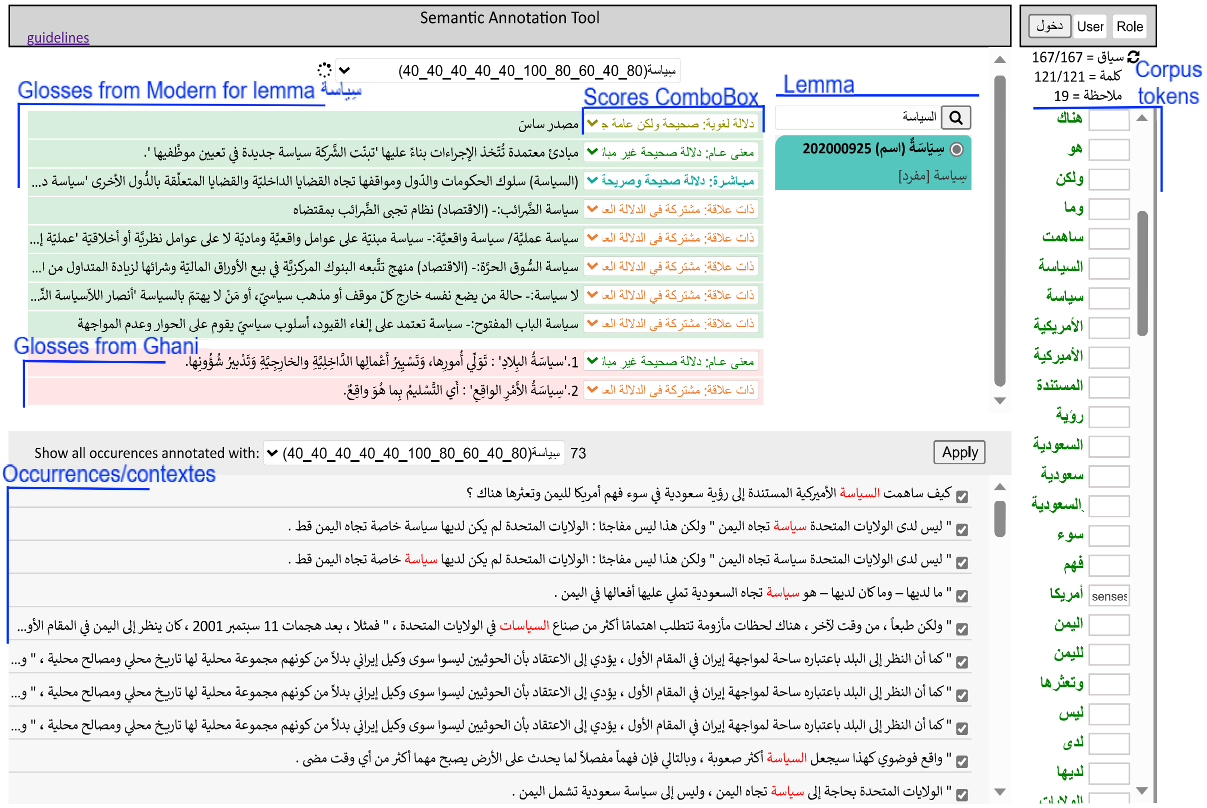
Our SALMA corpus is part of the Wojood corpus (Jarrar et al., 2022), and was collected from 33 online media sources written in Modern Standard Arabic (MSA) and covering general topics. Some of those sources include mipa.institute, sanaacenter.org, hrw.org, diplomatie.ma, sa.usembassy.gov, eeas.europa.eu, crisisgroup.org, and mofaic.gov.ae. The corpus was then segmented into sentences and tokenized, resulting in 1439 sentences and ~34K tokens, with an average of 23.8 tokens per sentence.

3.2 Annotation Framework

This section presents a novel sense annotation framework, where instead of linking a word to one sense, we propose to score all semantically related senses to the word. The score ranges between 1-100% and a sense with a score $\geq 60\%$ is considered a correct sense of the word. The ranking scale is divided into six categories:

- *Explicate* /مباشرة (100%): direct and explicate semantics (دلالة صحيحة وصريحة).
- *General* /معنى عام (80%): correct but implicate semantics (دلالة صحيحة غير مباشرة).
- *Referral* /دلالة لغوية (60%): generally correct semantics, but is referred to another lemma (صحيحة ولكن عامة جداً مثل مصدر، اسم فاعل). For example, the word *drinker* and its gloss (*active participle of drink*).
- *Related* /ذات علاقة (40%): weak semantics (مشتركة في الدلالة العامة فقط، أختها دلالياً). For example, the term (سياسة الشركة *syāsh ālšrkh*) / *company's policy*, is related to the sense (*the policy used to collect taxes*) which is not a sense of the lemma (سياسة *syāsh*), but semantically related.
- *Root semantics* /دلالة جذر (20%): share root semantics (دلالة مختلفة ولكن تشترك في الدلالة المجردة التي يحملها الجذر، مثل الدلالة المجازية). In Arabic lexical semantics, all words with the same root share part of the semantics of this root (Ryding, 2014; Boudelaa and Marslen-Wilson, 2004; Boudelaa et al., 2010). For example, all senses of the lemma (سياسة *syāsh*), such as *politics* and *policies* share an abstract meaning (e.g., issues related to governing and acting).
- *Different* /مختلفة (1%): unrelated semantics (دلالة مختلفة تماماً).

This framework serves several purposes. First, in case of underdeveloped sense inventories (such as the Modern and Ghani lexicons), in which glosses might be vague, redundant, or overlapping, our framework allows the annotators to score each sense. In this paper, we linked every word in the corpus with all semantically related senses in Modern and Ghani, thus we were able to compare and evaluate the lexical coverage in both lexicons (see Section 3.5). Another advantage of using this framework (i.e., scoring all senses) is that our corpus can be used to benchmark ranking-based WSD methods (Conia and Navigli, 2021; Yap et al.,



سِياسة(40_40_40_40_40_100_80_60_40_80)

دلالة لغوية: صحيحة ولكن عامة ج	مصدر ساسَ
معنى عـام: دلالة صحيحة غير مبا	مبادئ معتمدة تُتّخذ الإجراءات بناءً عليها 'تبنّت الشَّركة سياسة جديدة في تعيين موظَّفيها '.
مباشرة: دلالة صحيحة وصريحة	(السياسة) سلوك الحكومات والدّول ومواقفها تجاه القضايا الداخليّة والقضايا المتعلِّقة بالدُّول الأخرى 'سياسة د...
ذات علاقة: مشتركة في الدلالة العا	سياسة الضَّرائب:- (الاقتصاد) نظام تجبى الضَّرائب بمقتضاه
ذات علاقة: مشتركة في الدلالة العا	سياسة عمليَّة/ سياسة واقعيَّة:- سياسة مبنيّة على عوامل واقعيَّة وماديّة لا على عوامل نظريَّة أو أخلاقيّة 'عمليّة إ...
ذات علاقة: مشتركة في الدلالة العا	سياسة السُّوق الحرَّة:- (الاقتصاد) منهج تتَّبعه البنوك المركزيَّة في بيع الأوراق الماليّة وشرائها لزيادة المتداول من ا...
ذات علاقة: مشتركة في الدلالة العا	لا سياسة:- حالة من يضع نفسه خارج كلّ موقف أو مذهب سياسيّ، أو مَنْ لا يهتمّ بالسياسة 'أنصار اللاّسياسة النِّ...
ذات علاقة: مشتركة في الدلالة العا	سياسة الباب المفتوح:- سياسة تعتمد على إلغاء القيود، أسلوب سياسيّ يقوم على الحوار وعدم المواجهة
معنى عـام: دلالة صحيحة غير مبا	1.'سياسَةُ البِلادِ' : تَوَلِّي أُمورِها، وَتَسْيِيرُ أَعْمالِها الدّاخِلِيَّةِ والخارِجِيَّةِ وَتَدْبيرُ شُؤُونِها.
ذات علاقة: مشتركة في الدلالة العا	2.'سِياسَةُ الأَمْرِ الواقِعِ' : أَيِ التَّسْليمُ بِما هُوَ واقِعٌ.

السياسة

○ سِيَاسَةٌ (اسم) 202000925
سِياسة [مفرد]

سياق = 167/167
كلمة = 121/121
ملاحظة = 19

هناك
هو
ولكن
وما
ساهمت
السياسة
سياسة
الأمريكية
الأميركية
المستندة
رؤية
السعودية
سعودية
السعودية
سوء
فهم
أمريكا senses
اليمن
لليمن
وتعثرها
ليس
لدى
لديها

Show all occurences annotated with: سِياسة(40_40_40_40_40_100_80_60_40_80) 73

☑ كيف ساهمت السياسة الأمريكية المستندة إلى رؤية سعودية في سوء فهم أمريكا لليمن وتعثرها هناك ؟

☑ " ليس لدى الولايات المتحدة سياسة تجاه اليمن " ولكن هذا ليس مفاجئا : الولايات المتحدة لم يكن لديها سياسة خاصة تجاه اليمن قط .

☑ " ليس لدى الولايات المتحدة سياسة تجاه اليمن " ولكن هذا ليس مفاجئا : الولايات المتحدة لم يكن لديها سياسة خاصة تجاه اليمن قط .

☑ " ما لديها – وما كان لديها – هو سياسة تجاه السعودية تملي عليها أفعالها في اليمن .

☑ " ولكن طبعاً ، من وقت لآخر ، هناك لحظات مأزومة تتطلب اهتمامًا أكثر من صناع السياسات في الولايات المتحدة ، " فمثلا ، بعد هجمات 11 سبتمبر 2001 ، كان ينظر إلى اليمن في المقام الأو...

☑ " كما أن النظر إلى البلد باعتباره ساحة لمواجهة إيران في المقام الأول ، يؤدي إلى الاعتقاد بأن الحوثيين ليسوا سوى وكيل إيراني بدلاً من كونهم مجموعة محلية لها تاريخ محلي ومصالح محلية ، " و...

☑ " كما أن النظر إلى البلد باعتباره ساحة لمواجهة إيران في المقام الأول ، يؤدي إلى الاعتقاد بأن الحوثيين ليسوا سوى وكيل إيراني بدلاً من كونهم مجموعة محلية لها تاريخ محلي ومصالح محلية ، " و...

☑ " كما أن النظر إلى البلد باعتباره ساحة لمواجهة إيران في المقام الأول ، يؤدي إلى الاعتقاد بأن الحوثيين ليسوا سوى وكيل إيراني بدلاً من كونهم مجموعة محلية لها تاريخ محلي ومصالح محلية ، " و...

☑ " واقع فوضوي كهذا سيجعل السياسة أكثر صعوبة ، وبالتالي فإن فهماً مفصلاً لما يحدث على الأرض يصبح مهما أكثر من أي وقت مضى .

☑ " الولايات المتحدة بحاجة إلى سياسة تجاه اليمن ، وليس إلى سياسة سعودية تشمل اليمن .

Figure 1: Screenshot of our web-based annotation tool.

2020), which is not possible in the case of one-sense annotated corpora.

3.3 Annotation Tool

We developed a web-based tool optimized for our sense annotation framework and methodology. On the right side of Figure 1, the linguist selects a word to be annotated (such as "السياسة *ālsyāsh*"). The tool will then retrieve all sentences (i.e. contexts) in the corpus containing the selected word. The tool will also automatically fetch the lemma of the selected word, and the linguist has the ability to search for the lemma manually. After selecting a lemma, the tool retrieves senses associated with the lemma from both lexicons, Modern and Ghani. The linguist can then select the score category for each sense according to our guideline and apply these scores to all selected words (in contexts) as shown in Figure 1. The scores are selected from a ComboBox of the six categories (See Section 3.2), however, the tool internally stores their corresponding numeric values.

3.4 Annotation Process

The annotation was carried out in three phases:

Phase 1 (training): we recruited three undergraduate students majoring in linguistics. The students were trained in three steps in order to produce consistent annotations. We first assigned 50 words to each linguist and trained them to conduct the annotation jointly. Second, we assigned the same 150 words to each student separately, then asked them to compare and consolidate their annotations, which helps in calibrating their scoring. Third, we repeated the second phase, but using 300 words and again we asked them to compare their annotations.

Phase 2 (annotation): out of ~34K tokens, excluding digits and punctuations, we assigned about 9.6K words to each of the three linguists. Each linguist was asked to annotate all occurrences of each word in the corpus - resulting in about ~29K annotations for the whole words.

Phase 3 (validation): after finishing the annotations, we used the tool to automatically validate the annotations and flag those that violated the following cases: (i) a word is annotated with more than one *Explicit* or *General* sense in the same lexicon, which is an indication of either a mistake or redundant or overlapping senses in the lexicon. (ii) a word is missing either an *Explicit* or a *General* sense; this is an indication of a mistake or the lexicon is missing this sense. (iii) if the selected sense is a proper noun, then all other senses should be ranked as *Different*. The linguists were asked to review these flagged annotations and revise them if necessary.

The linguists were encouraged to discuss among themselves and take joint decisions when facing difficulties, especially in the case of vague glosses or contexts. In addition, as will be discussed in Section 3.5, missing lemmas and senses are manually added to the lexicons. Table 2 provides general statistics about the annotations. It is worth noting that sense annotations are typically costly and time-consuming. The linguists spent about 600 working days (i.e., 4800 working hours) to carry out the three phases described above.

Term	**Noun**	**Verb**	**Func. Words**	**Punc+ Digits**	**Total**
Tokens	19,030	2,763	7,116	5,344	**34,253**
Unique Tokens	6,670	1,593	322	175	**8,760**
Unique Lemmas	2,904	677	119	175	**3,875**
Unique Senses	3,151	792	206	2	**4,151**

Table 2: Statistics of the SALMA corpus.

Term	**Modern**	**Ghani**
Lemmas	80% (2,788/3,522)	78% (2,724/3,522)
Senses (Without Proper nouns)	83% (3,430/4,151)	78% (3,226/4,151)
Proper Nouns Senses	4% (9/213)	14% (30/213)

Table 3: Coverage of Modern and Ghani lexicons.

3.5 Discussion and Lexical Coverage

We evaluated the coverage of both lexicons based on the sense-annotated tokens. As Table 3 shows, Modern has higher coverage of lemmas (80%) compared to Ghani's coverage (78%), and has higher sense coverage (83%) compared to Ghani (78%). Moreover, glosses in Modern are more precise, less ambiguous and well-formulated as discussed in Section 4.1. The proper nouns are the main reason for the missing lemmas and senses, as the Modern and Ghani cover 4% and 14% of proper nouns in SALMA corpus, respectively. Lemmas and senses that are not covered by any of the two lexicons were added manually by the linguists. All numerical values are annotated with the same "digit" sense that covers ordinal and nominal numbers, and similarly, punctuation marks are all annotated with "Punc".

3.6 Named Entity Annotations

Named-entity annotations are important in sense-annotated corpora because sense inventories do not typically cover names of organizations, towns, people, landmarks, and others.

Tag	**Description**
PERS	Person names: first, middle, last, nickname ...
ORG	Organizations: company, team, government ...
GPE	Geopolitical entities: country, city, state ...
LOC	Geographical locations: river, sea, mountain...
FAC	facilities: landmark, road, building, airport ...
CURR	Currency names or symbols.

Table 4: Types of named entities.

In addition to word-sense annotations, we annotated our corpus using six types of named entities listed in Table 4. As our corpus is a part of the Wojood, which is annotated with 21 types of nested named-entities (Jarrar et al., 2022), in this article we annotated SALMA with six flat entities only. We used the IOB2 tagging scheme (Sang and Veenstra, 1999), where B indicates the beginning of the entity mention, I the inside token, and O outside token.

Tag	**Named Entity Mentions**	**Tokens in the Entity Mentions**
PERS	294	568
ORG	1,123	2,108
GPE	1,086	1,295
LOC	166	318
FAC	22	59
CURR	37	41
Total	**2,728**	**4,389**

Table 5: Statistics of named entities in SALMA corpus.

We applied the NER guidelines that were used to annotate the OntoNotes5 corpus (Weischedel et al., 2011). Table 5 presents statistics about all named entities in the SALMA corpus, which shows that 4389 (about 15%) of the tokens are part of an entity mention.

4 Inter-Annotation Agreement (IAA)

To evaluate our annotations, we selected 250 annotated words from each annotator $A \in \{A_1, A_2, A_3\}$, and assigned them to a different annotator to perform double annotations. This yielded a total of 750 words (2.6% of the annotated words) divided among three pairs of annotators, $\{(A_1, A_2), (A_1, A_3), (A_2, A_3)\}$. Because our sense annotations contain scores (i.e., not discrete values), computing IAA is not straightforward. We chose to use various evaluation metrics especially those that take ranking into consideration. The IAA metrics used are: (i) Kappa, (ii) Linear Weighted Kappa (LWK), (ii) Quadratic Weighted Kappa (QWK), (iv) Mean Average Error (MAE), and (v) Root Mean Square Error (RMSE).

Kappa is usually used when the data is nominal (Eugenio and Glass, 2004), so we set a threshold on the score (≥60%) in the six categories to be able to calculate Cohen's Kappa. The senses with scores above or equal this threshold carry the intended meanings that map with the context of the targeted word (See section 3.2). Nonetheless, a more suitable metric for ranked labels is either the LWK or QWK, as specified in the following equations, which we adopt from (Vanbelle, 2016):

$$QWK = 1 - \frac{\sum\limits_{i,j=1}^{K} \frac{(y_i - y_j)^2}{(K-1)^2} . fo_{ij}}{\sum\limits_{i,j=1}^{K} \frac{(y_i - y_j)^2}{(K-1)^2} . fe_{ij}} \qquad (1)$$

$$LWK = 1 - \frac{\sum\limits_{i,j=1}^{K} \frac{|y_i - y_j)|}{(K-1)} . fo_{ij}}{\sum\limits_{i,j=1}^{K} \frac{|y_i - y_j)|}{(K-1)} . fe_{ij}} \qquad (2)$$

where fo_{ij} is the observed frequency of the categories (i and j) per the annotators selection, fe_{ij} is the expected frequency for both annotators' selected categories, $(y_i - y_{jx})$ denotes the distance between the categories, and K is number of categories.

Both LWK and QWK take the distance between categories into consideration, where the distance is defined as the number of categories separating the two annotators' selection. The difference is that LWK calculates the distance linearly while QWK calculates it quadratically. For measuring the ranking error deviation among annotators we used MAE and RMSE.

4.1 IAA Results

Table 6 summarizes the result of the inter-annotator-agreement, the value in parenthesis is the standard deviation among pairs of annotators. Overall, we see higher agreement among the annotators for the Modern. The higher agreement is clear from all IAA metrics and the standard deviation. We see less confidence in the Ghani annotations as the IAA dropped across all metrics with higher variability among annotators, presented in higher standard deviation. Kappa was affected the most with a drop of 11.8% when measured on the Ghani, followed by LWK with a drop of 8.73%. QWK has the smallest drop of 5.91% and also has the least variability among annotators. We believe the reason for the higher IAA on Modern is because Modern has better quality glosses compared to the Ghani, which has shorter glosses and in many cases are ambiguous. However, regardless of the lexicon used, we observed higher agreement among annotators as measured by LWK and QWK since they take advantage of the scores assigned to each gloss, while Kappa ignores the scoring information.

Metric	**Lexicons**	**Average (STD)**
Kappa	Modern Ghani	90.48 (±2.97) 78.68 (±8.49)
LWK	Modern Ghani	88.29 (±5.37) 79.56 (±9.35)
QWK	Modern Ghani	91.94 (±3.42) 86.03 (±5.41)
RMSE	Modern Ghani	13.44 (±3.08) 19.12 (±3.06)
MAE	Modern Ghani	4.46 (±2.04) 8.27 (±3.52)

Table 6: Inter-Annotator Agreement (IAA) average among the three linguists using different metrics.

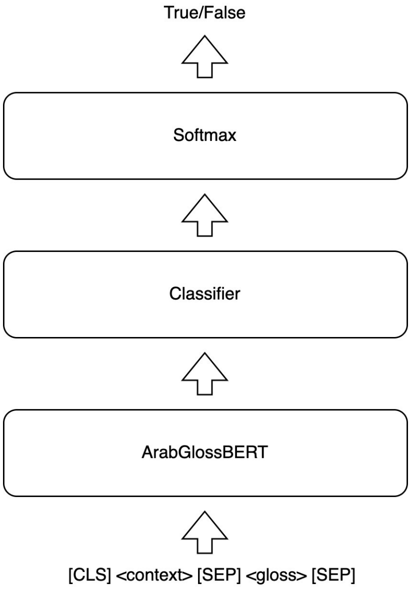


Figure 2: BERT-based TSV Architecture.

We reach similar conclusions for RMSE and MAE. Both metrics are lower for Modern compared to Ghani. The Average RMSE among all annotator pairs on the Modern is 13.44 compared

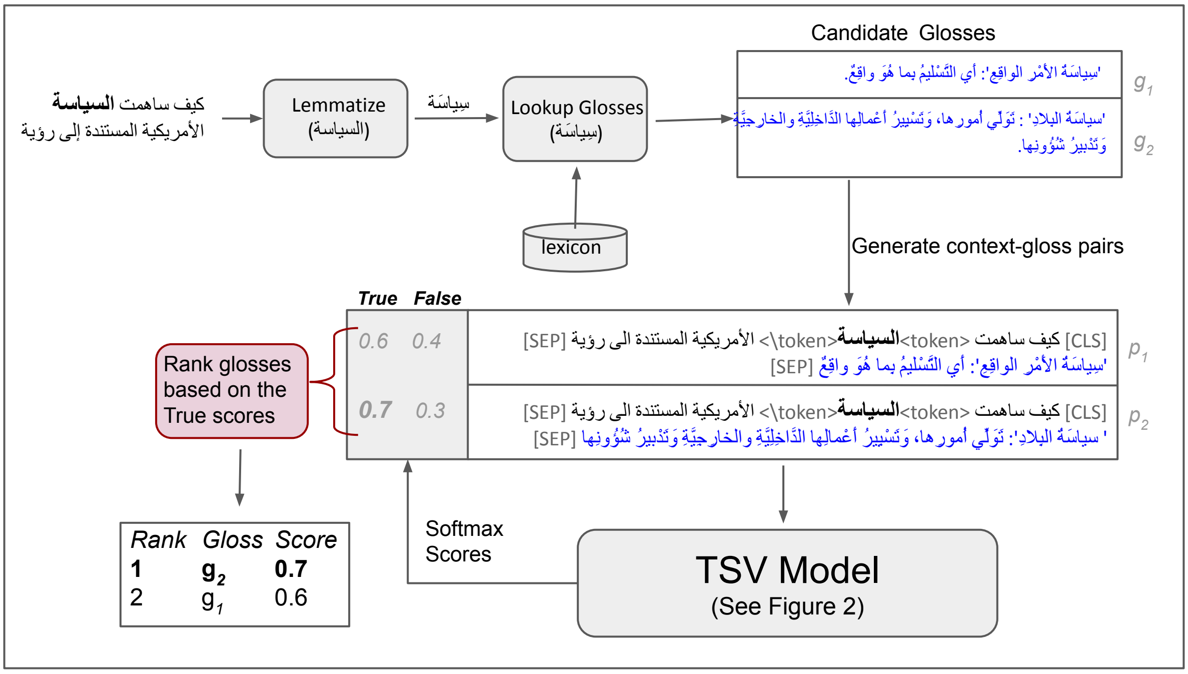

كيف ساهمت **السياسة** الأمريكية المستندة إلى رؤية
Lemmatize (السياسة)
سِياسَة
Lookup Glosses (سِياسَة)
lexicon
Candidate Glosses
'سِياسَةُ الأمْرِ الواقِعِ': أي التَّسْليمُ بما هُوَ واقِعٌ.
g_1
'سِياسَةُ البِلادِ' : تَوَلّي أُمورِها، وَتَسْييرُ أعْمالِها الدَّاخِلِيَّةِ والخارِجيَّةِ وَتَدْبيرُ شُؤونِها.
g_2
Generate context-gloss pairs
True False
0.6 0.4
[CLS] كيف ساهمت <token>**السياسة**<\token> الأمريكية المستندة الى رؤية [SEP] 'سِياسَةُ الأمْرِ الواقِعِ': أي التَّسْليمُ بما هُوَ واقِعٌ [SEP]
p_1
0.7 0.3
[CLS] كيف ساهمت <token>**السياسة**<\token> الأمريكية المستندة الى رؤية [SEP] ' سِياسَةُ البِلادِ': تَوَلّي أُمورِها، وَتَسْييرُ أعْمالِها الدَّاخِلِيَّةِ والخارِجيَّةِ وَتَدْبيرُ شُؤونِها [SEP]
p_2
Rank glosses based on the True scores
Softmax Scores
TSV Model (See Figure 2)
Rank Gloss Score
1 g_2 0.7
2 g_1 0.6

Figure 3: An end-to-end WSD using the TSV model (SALMA system).

to 19.12 for Ghani, while the average MAE for the Modern is 4.46 compared to 8.27 on the Ghani.

5 Computing WSD Baselines using SALMA

In this section, we present the baseline for Arabic WSD using our SALMA corpus. To the best of our knowledge, there are no available Arabic WSD systems to evaluate. The only available Arabic models are TSV, which are related, but not the same as WSD. In what follows, we explain the difference between WSD and TSV tasks, and propose an end-to-end WSD system using TSV.

5.1 The TSV Task

The TSV task is a binary classification task used to determine whether a pair of sentences (context and gloss) are True or False (see Figure 2). In other words, given a context c containig the target word w, and a gloss g_i, TSV aims to classify the context-gloss pair (c, g_i) as True or False. It is True if the gloss g_i is the intended sense of w in c, otherwise, it is False (Breit et al., 2020). It is important to note that TSV is different from WSD, which determines which gloss, among a set of glosses, is the intended meaning for the target word.

There are three available Arabic TSV models with the same architecture: (1) the Razzaz model, trained using 31K context-gloss pairs extracted from Modern (El-Razzaz et al., 2021); (2) the ArabGlossBERT model, trained on a larger dataset (167K context-gloss pairs) extracted from several Arabic lexicons (Al-Hajj and Jarrar, 2021); and (3) the Aug-ArabGlossBERT (D9) model, trained on an augmented data, generated using back-translation of the ArabGlossBERT dataset (Malaysha et al., 2023).

In what follows, we propose to develop an end-to-end WSD system using TSV (called SALMA system) and in Section 6, we benchmark our proposed system using the SALMA corpus.

5.2 Building WSD System Using TSV

In this section, we propose an end-to-end solution for WSD using TSV. The solution consists of the following phases (Figure 3): 1) candidate glosses lookup, 2) target sense verification, and 3) gloss ranking.

1. Candidate Glosses Lookup: given a target word w in a context c, we first lemmatize w (i.e., determine its lemma l), where we use our own in-house lemmatizer, then retrieve the set of n candidate glosses, $G = \{g_1, g_2, ..., g_n\}$, of l from the lexicon (i.e., sense inventory).
Example: the word w (السياسة *ālsyāsh*) in c (كف ساهمت الساسة الأمركة المستندة الى رؤية) has the lemma (سِيَاسَةٌ *siyaāsatun*) with two corresponding glosses ($\{g_1, g_2\}$) in the Ghani, as shown in Figure 3.

2. TSV: once we have the set of n candidate glosses, we input to the TSV model a set of n context-gloss pairs, $P = \{(c, g_i) | \forall g_i \in G\}$, as illustrated with (p_1, p_2) in Figure 3. The target word w in c is wrapped with special tokens "<token>w</token>", to emphasize the target word during training and testing of the TSV models. For each context-gloss pair, the TSV model returns confidence scores for the True and False labels, but the TSV model does not compare or rank glosses in this phase.

3. Gloss Ranking: we determine the intended meaning by ranking the glosses based on their True confidence scores calculated in the previous step. The gloss with the highest score is selected as the intended gloss for w.

6 Experiments and Results

6.1 Experimental Setup

To evaluate the three available Arabic TSV models using our SALMA corpus, we implemented three instances of the WSD system depicted in Figure 3, each with a different TSV model. For each word in each context in the SALMA corpus, we generated context-gloss pairs similar to the example shown in Figure 3. Because our corpus was sense-annotated using two lexicons (i.e., two sense inventories), we generated two sets of context-gloss pairs. In this way, we compute a separate baseline for each of the Modern and Ghani. We neither included annotations of digits and punctuations, nor the named-entity annotations presented in Section 3.6.

The length of the contexts may impact the WSD accuracy, so in addition to using the full context around w, we also experimented with different context sizes, $s \in \{3, 5, 7, 9, 11\}$. For example, the context size $s = 5$ means that there are two tokens before and two tokens after w.

As will be discussed in the next subsection, we evaluated three TSV models: Razzaz[2], ArabGlossBERT[3], and Aug-ArabGlossBERT(D9)[4]. We used context size $s = 11$, which gave the best results. Following the authors of these models, we did not use any signal to mark up target words in the case of the Razzaz and Aug-ArabGlossBERT(D9); however, we used UNUSED0 for ArabGlossBERT.

The experiments have been implemented in Python, specifically using the Transformers library provided by HuggigFace[5], which is used to load and test the models. To speed-up the models evaluation, we have run the codes using a GPU (SVGA II) instance, where each run took around 20 hours.

TSV Model	**Lexicons**	**Accuracy**
Razzaz	Modern Ghani	66.0% 68.4%
ArabGlossBERT	Modern Ghani	**84.2%** 77.6%
Aug-ArabGlossBERT(D9)	Modern Ghani	82.6% 78.7%

Table 7: WSD baselines for three TSV models, with context length = 11.

6.2 Baselines and Discussion

Table 7 presents our evaluation of the three TSV models using both Modern and Ghani with context size $s = 11$. As shown in this table, the ArabGlossBERT is the best-performing model(84.2%), which most probably because it was trained on a larger and higher quality dataset of lexicon definitions. The accuracy was calculated for nouns and verbs. We excluded the functional words as they mostly do not carry semantics.

Window	**Lexicon**	**Accuracy** **Target Sense Rank**			**Accuracy (Top1)** **per POS**		
		Top1	**Top2**	**Top3**	**Noun**	**Verb**	**Func.**
All	Modern	82.8	94.2	97.4	83.5	77.9	41.2
	Ghani	77.0	89.3	94.1	78.5	66.0	36.0
11	**Modern**	**84.2**	95.1	98.1	85.4	76.1	37.9
	Ghani	**77.6**	90.1	94.9	79.4	61.7	31.8
9	Modern	83.5	95.0	97.9	84.4	78.3	37.7
	GHani	77.3	90.1	94.8	79	63.7	32.2
7	Modern	83.8	95.1	97.9	84.8	77.4	38.9
	Ghani	77.3	90.0	94.9	79.1	62.9	31.8
5	Modern	84.0	95.1	98.1	85.3	75.6	40.0
	Ghani	77.6	90.1	94.9	79.5	61.6	31.7
3	Modern	82.8	94.4	97.6	84.4	71.8	42.1
	Ghani	77.4	90.0	94.8	79.4	59.7	32.1

Table 8: Baselines - evaluation of ArabGlossBERT on two sense inventories, with different context windows and sense orderings.

Table 8 presents further evaluation of ArabGlossBERT, which illustrates the following: (i) using Modern is better than using Ghani in all experiments. This might be because of the better quality

[2] We reproduced the TSV model using the code and data available at https://github.com/MElrazzaz/Arabic-word-sense-disambiguation-bench-mark

[3] ArabGlossBERT fine-tuned model Version 1 (CC-BY-4.0) at https://huggingface.co/SinaLab/ArabGlossBERT/tree/main

[4] Fine-tuned model D9 (CC-BY-4.0) at https://huggingface.co/SinaLab/ArabGlossBERT/tree/Augment

[5] https://huggingface.co/docs/transformers/index

of Modern glosses (refer to IAA in Section 4); (ii) While window 11 and 5 have the highest WSD accuracy, the use of context windows does not make major difference (only 1.4% for Modern and 0.6% for Ghani); (iii) the ranking of the intended sense among the top 1, 2, and 3 senses illustrates a consistent and reasonable increase in the WSD accuracy; and (iv) when evaluating the model accuracy for noun and verb, the accuracy of nouns is about 8.5% better than verbs for Modern, which might be because verbs are typically more ambiguous (Malaysha et al., 2023). The WSD accuracy for functional words is very low with both lexicons. This is because functional words are highly polysemous and their glosses describe their functions rather than semantics.

7 Conclusion

We presented SALMA, the first sense-annotated Arabic corpus. The novelty of SALMA lies in utilizing two sense inventories and named entity annotations. In addition, instead of linking a word to one intended sense, we scored all semantically related senses of each token in the corpus. The quality of the annotations was assessed using various inter-annotator agreement metrics (Kappa, LWK, QWK, MAE, and RSME). To compute a WSD baseline using our corpus, we proposed to build an end-to-end WSD system using TSV, and evaluated this system using three different TSV models. The full corpus, annotations, and the tool, are open source and publicly available on GitHub.

8 Limitations and Future Work

Although Modern provides a better quality of glosses compared with the Ghani, some of Modern's glosses are referrals, i.e., referred to another related lemma. At this stage, we annotated these referrals as senses. Nevertheless, in order to use the Modern as a general sense inventory, these referrals need to be treated differently. We plan to replace all referral glosses with the senses they refer to, which can be done semi-automatically. For missing lemmas in Modern, we plan to map between the lemmas in both lexicons and then import missing lemmas and their senses from Ghani to Modern. In this way, we expect to have a richer Arabic sense inventory. Additionally, our sense annotations are limited to the senses of a single-word lemma. We plan to annotate the corpus with multiword expressions (Jarrar et al., 2018). Furthermore, the corpus we presented in this article is limited to MSA. To extend this corpus with dialectal text, plan to sense-annotate portions of the available corpora Curras (Haff et al., 2022; Jarrar et al., 2017), Baladi (Haff et al., 2022), Nabra (Nayouf et al., 2023) and Lisan (Jarrar et al., 2023).

Acknowledgment

We would like to thank Shimaa Hamayel, Tamara Qaimari, Raghad Aburahma, Hiba Zayed, and Rwaa Assi for helping us in the corpus annotation.

References

Abdul-Ghani Abul-Azm. 2014. Al-ghani al-zaher dictionary. *Rabat: Al-Ghani Publishing Institution*.

Moustafa Al-Hajj and Mustafa Jarrar. 2021. Arabglossbert: 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 40–48, Online. INCOMA Ltd.

Diana Alhafi, Anton Deik, and Mustafa Jarrar. 2019. Usability evaluation of lexicographic e-services. In *The 2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA)*, pages 1–7. IEE.

Hamzeh Amayreh, Mohammad Dwaikat, and Mustafa Jarrar. 2019. Lexicon digitization-a framework for structuring, normalizing and cleaning lexical entries. *Technical Report, Birzeit University*.

William Black, Sabri Elkateb, Horacio Rodriguez, Musa Alkhalifa, Piek Vossen, Adam Pease, Christiane Fellbaum, et al. 2006. Introducing the arabic wordnet project. In *Proceedings of the third international WordNet conference*, pages 295–300. Jeju Korea.

Terra Blevins and Luke Zettlemoyer. 2020. Moving down the long tail of word sense disambiguation with gloss-informed biencoders. *arXiv preprint arXiv:2005.02590*.

Sami Boudelaa and William D Marslen-Wilson. 2004. Abstract morphemes and lexical representation: The cv-skeleton in arabic. *Cognition*, 92(3):271–303.

Sami Boudelaa, Friedemann Pulvermüller, Olaf Hauk, Yury Shtyrov, and William Marslen-Wilson. 2010. Arabic morphology in the neural language system. *Journal of cognitive neuroscience*, 22(5):998–1010.

Anna Breit, Artem Revenko, Kiamehr Rezaee, Mohammad Taher Pilehvar, and Jose Camacho-Collados. 2020. Wic-tsv: An evaluation benchmark for target sense verification of words in context. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 1635–1645.

Bartosz Broda, Michał Marcińczuk, Marek Maziarz, Adam Radziszewski, and Adam Wardyński. 2012. KPWr: Towards a free corpus of Polish. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 3218–3222, Istanbul, Turkey. European Language Resources Association (ELRA).

Simone Conia and Roberto Navigli. 2021. Framing word sense disambiguation as a multi-label problem for model-agnostic knowledge integration. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 3269–3275. Association for Computational Linguistics.

Kareem Darwish, Nizar Habash, Mourad Abbas, Hend Al-Khalifa, Huseein T. Al-Natsheh, Houda Bouamor, Karim Bouzoubaa, Violetta Cavalli-Sforza, Samhaa R. El-Beltagy, Wassim El-Hajj, Mustafa Jarrar, and Hamdy Mubarak. 2021. A panoramic survey of natural language processing in the arab worlds. *Commun. ACM*, 64(4):72–81.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Mohammed El-Razzaz, Mohamed Waleed Fakhr, and Fahima A Maghraby. 2021. Arabic gloss wsd using bert. *Applied Sciences*, 11(6):2567.

Bilel Elayeb. 2019. Arabic word sense disambiguation: a review. *Artif. Intell. Rev.*, 52(4):2475–2532.

Barbara Di Eugenio and Michael Glass. 2004. The kappa statistic: A second look. *Computational linguistics*, 30(1):95–101.

Sana Ghanem, Mustafa Jarrar, Radi Jarrar, and Ibrahim Bounhas. 2023. A benchmark and scoring algorithm for enriching arabic synonyms. In *Proceedings of the 12th International Global Wordnet Conference (GWC2023)*, pages 215–222. Global Wordnet Association.

Karim El Haff, Mustafa Jarrar, Tymaa Hammouda, and Fadi Zaraket. 2022. Curras + baladi: Towards a levantine corpus. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2022)*, Marseille, France.

Luyao Huang, Chi Sun, Xipeng Qiu, and Xuanjing Huang. 2019. Glossbert: BERT for word sense disambiguation with gloss knowledge. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 3507–3512. Association for Computational Linguistics.

Mustafa Jarrar. 2005. *Towards Methodological Principles for Ontology Engineering*. Ph.D. thesis, Vrije Universiteit Brussel.

Mustafa Jarrar. 2006. Towards the notion of gloss, and the adoption of linguistic resources in formal ontology engineering. In *Proceedings of the 15th international conference on World Wide Web (WWW2006)*, pages 497–503. ACM Press, New York, NY.

Mustafa Jarrar. 2011. Building a formal arabic ontology (invited paper). In *Proceedings of the Experts Meeting on Arabic Ontologies and Semantic Networks*. ALECSO, Arab League.

Mustafa Jarrar. 2021. The arabic ontology - an arabic wordnet with ontologically clean content. *Applied Ontology Journal*, 16(1):1–26.

Mustafa Jarrar and Hamzeh Amayreh. 2019. An arabic-multilingual database with a lexicographic search engine. In *The 24th International Conference on Applications of Natural Language to Information Systems (NLDB 2019)*, volume 11608 of *LNCS*, pages 234–246. Springer.

Mustafa Jarrar, Hamzeh Amayreh, and John P. McCrae. 2019. Representing arabic lexicons in lemon - a preliminary study. In *The 2nd Conference on Language, Data and Knowledge (LDK 2019)*, volume 2402, pages 29–33. CEUR Workshop Proceedings.

Mustafa Jarrar, Nizar Habash, Faeq Alrimawi, Diyam Akra, and Nasser Zalmout. 2017. Curras: An annotated corpus for the palestinian arabic dialect. *Journal Language Resources and Evaluation*, 51(3):745–775.

Mustafa Jarrar, Eman Karajah, Muhammad Khalifa, and Khaled Shaalan. 2021. Extracting synonyms from bilingual dictionaries. In *Proceedings of the 11th International Global Wordnet Conference (GWC2021)*, pages 215–222. Global Wordnet Association.

Mustafa Jarrar, Mohammed Khalilia, and Sana Ghanem. 2022. Wojood: Nested arabic named entity corpus and recognition using bert. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2022)*, Marseille, France.

Mustafa Jarrar, Fadi Zaraket, Rami Asia, and Hamzeh Amayreh. 2018. Diacritic-based matching of arabic words. *ACM Asian and Low-Resource Language Information Processing*, 18(2):10:1–10:21.

Mustafa Jarrar, Fadi Zaraket, Tymaa Hammouda, Daanish Masood Alavi, and Martin Waahlisch. 2023. Lisan: Yemeni, irqi, libyan, and sudanese arabic dialect copora with morphological annotations. In *The 20th IEEE/ACS International Conference on Computer Systems and Applications (AICCSA)*. IEEE.

Sanad Malaysha, Mustafa Jarrar, and Mohammad Khalilia. 2023. Context-gloss augmentation for improving arabic target sense verification. In *Proceedings of the 12th International Global Wordnet Conference (GWC2023)*. Global Wordnet Association.

Federico Martelli, Najla Kalach, Gabriele Tola, and Roberto Navigli. 2021. Semeval-2021 task 2: Multilingual and cross-lingual word-in-context disambiguation (mcl-wic). In *Proceedings of the 15th International Workshop on Semantic Evaluation, SemEval@ACL/IJCNLP 2021, Virtual Event / Bangkok, Thailand, August 5-6, 2021*, pages 24–36. Association for Computational Linguistics.

Marek Maziarz, Maciej Piasecki, and Stan Szpakowicz. 2012. Approaching plwordnet 2.0. In *Proceedings of 6th International Global Wordnet Conference, The Global WordNet Association*, pages 189–196.

George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J Miller. 1990. Introduction to wordnet: An on-line lexical database. *International journal of lexicography*, 3(4):235–244.

George A Miller, Claudia Leacock, Randee Tengi, and Ross T Bunker. 1993. A semantic concordance. In *Human Language Technology: Proceedings of a Workshop Held at Plainsboro, New Jersey, March 21-24, 1993*.

Simonetta Montemagni and Guglielmo Venturi. 2003. Building sense-tagged corpora for all: The itec project. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*.

Eman Naser-Karajah, Nabil Arman, and Mustafa Jarrar. 2021. Current trends and approaches in synonyms extraction: Potential adaptation to arabic. In *Proceedings of the 2021 International Conference on Information Technology (ICIT)*, pages 428–434, Amman, Jordan. IEEE.

Amal Nayouf, Mustafa Jarrar, Fadi zaraket, Tymaa Hammouda, and Mohamad-Bassam Kurdy. 2023. Nâbra: Syrian arabic dialects with morphological annotations. In *Proceedings of the 1st Arabic Natural Language Processing Conference (ArabicNLP), Part of the EMNLP 2023*. ACL.

Ahmed Mukhtar Omar. 2008. Contemporary arabic dictionary.(i1). *World of Books, Cairo, Egypt. Retrieval Date*, 14(8):2020.

Tommaso Petrolito and Francis Bond. 2014. A survey of wordnet annotated corpora. In *Proceedings of the Seventh Global Wordnet Conference, GWC 2014, Tartu, Estonia, January 25-29, 2014*, pages 236–245. University of Tartu Press.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *OpenAI Technical Report*, 1(8).

Karin C Ryding. 2014. *Arabic: A linguistic introduction*. Cambridge University Press.

Erik F. Tjong Kim Sang and Jorn Veenstra. 1999. Representing text chunks. In *EACL 1999, 9th Conference of the European Chapter of the Association for Computational Linguistics, June 8-12, 1999, University of Bergen, Bergen, Norway*, pages 173–179. The Association for Computer Linguistics.

Nathan Schneider, Behrang Mohit, Kemal Oflazer, and Noah A. Smith. 2012. Coarse lexical semantic annotation with supersenses: An arabic case study. In *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8-14, 2012, Jeju Island, Korea - Volume 2: Short Papers*, pages 253–258. The Association for Computer Linguistics.

Liling Tan and Francis Bond. 2012. Building and annotating the linguistically diverse NTU-MC (NTU - multilingual corpus). *Int. J. Asian Lang. Process.*, 22(4):161–174.

Mariona Taulé, M. Antònia Martí, and Marta Recasens. 2008. AnCora: Multilevel annotated corpora for Catalan and Spanish. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Heike Telljohann, Erhard Hinrichs, and Ra Ubler. 2004. The tüba-d/z treebank: Annotating german with a context-free backbone. *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC)*.

Sophie Vanbelle. 2016. A new interpretation of the weighted kappa coefficients. *Psychometrika*, 81(2):399–410.

Warren Weaver. 1949/1955. Translation. In William N. Locke and A. Donald Boothe, editors, *Machine Translation of Languages*, pages 15–23. MIT Press, Cambridge, MA. Reprinted from a memorandum written by Weaver in 1949.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Mohammed El-Bachouti, Robert Belvin, and Ann Houston. 2013. Ontonotes release 5.0 ldc2013t19. *Philadelphia: Linguistic Data Consortium*.

Ralph Weischedel, Sameer Pradhan, Lance Ramshaw, Martha Palmer, Nianwen Xue, Mitchell Marcus, Ann Taylor, Craig Greenberg, Eduard Hovy, Robert Belvin, et al. 2011. Ontonotes release 4.0. *LDC2011T03, Philadelphia, Penn.: Linguistic Data Consortium*.

Boon Peng Yap, Andrew Koh, and Eng Siong Chng. 2020. Adapting bert for word sense disambiguation with gloss selection objective and example sentences. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 41–46.

Arabic dialect identification: An in-depth error analysis on the MADAR parallel corpus

Helene Bøsei Olsen
University of Oslo
helenbol@ifi.uio.no

Samia Touileb
University of Bergen
samia.touileb@uib.no

Erik Velldal
University of Oslo
erikve@ifi.uio.no

Abstract

This paper provides a systematic analysis and comparison of the performance of state-of-the-art models on the task of fine-grained Arabic dialect identification using the MADAR parallel corpus. We test approaches based on pre-trained transformer language models in addition to Naive Bayes models with a rich set of various features. Through a comprehensive data- and error analysis, we provide valuable insights into the strengths and weaknesses of both approaches. We discuss which dialects are more challenging to differentiate, and identify potential sources of errors. Our analysis reveals an important problem with identical sentences across dialect classes in the test set of the MADAR-26 corpus, which may confuse any classifier. We also show that none of the tested approaches captures the subtle distinctions between closely related dialects.

1 Introduction

Dialect identification (DID) is a task in natural language processing (NLP) aiming to automatically identify a dialect within a pre-determined language. Because dialectal differences tend to be subtle, identifying dialects is considered a more difficult task than language identification (Etman and Beex, 2015). Arabic dialects are considered particularly challenging due to their high level of ambiguity, lack of standardisation, and rich morphology (Diab and Habash, 2007). Most NLP development has focused on Modern Standard Arabic (MSA), the formal and standardised version of Arabic. However, these tools are not always transferable to dialectal Arabic, as dialects differ from each other and MSA in terms of lexicon, phonology, orthography, and morphology (Habash, 2010). A prominent resource for Arabic DID is the MADAR parallel corpus (Bouamor et al., 2018), targeting dialects on the city-level. MADAR has been established as an important corpus for the task, serving as a benchmark for multi-task learning (Seelawi et al., 2021), as well as a Shared Task corpus (Bouamor et al., 2019), and as a subject of independent research (Baimukan et al., 2022). Despite several attempts to develop models using deep neural networks (Lippincott et al., 2019; de Francony et al., 2019) and pre-trained Transformer-based language models (Inoue et al., 2021), the current state-of-the-art approach remains a statistical machine learning model with surface-level feature representation, specifically the Multinomial Naive Bayes (MNB) model introduced by Salameh et al. (2018).

The lack of progress on the task, along with the inability of BERT models to surpass the MNB model, gives rise to several questions that have not yet been thoroughly explored, and on which we focus in the current work. Firstly, do BERT models make the same mistakes as the state-of-the-art MNB model on the dialect identification task? While Salameh et al. (2018) have documented the performance of the MNB model on individual dialects and highlighted the Muscat dialect as the most challenging for the model, there is limited research exploring the misclassifications generated by BERT models. Secondly, if the models make different errors, are these errors centred around the same dialect pairs? Thirdly, we explore if a detailed analysis of the misclassified sentences by both the BERT models and the MNB model can provide deeper insights into the challenges of the task on MADAR-26.

This paper summarises the findings from a comprehensive project on error-analysis on the MADAR parallel corpus conducted by Olsen (2023). We release the code for all experiments and analysis on GitHub.[1]

2 Previous work

Several efforts have focused on building tools and resources to identify Arabic dialects. However,

[1] https://github.com/helenebol/Arabic-dialect-identification

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 370–384
December 7, 2023 ©2023 Association for Computational Linguistics

the field suffers from fragmented and independent works on different corpora that vary in terms of granularity, size and domain, making it challenging to track the progress of the solutions. Early work focused on binary dialect classification by discriminating one dialect from MSA (Elfardy and Diab, 2013; Tillmann et al., 2014), as well as identifying Arabic dialects at both a region-level (Zaidan and Callison-Burch, 2011, 2014; Elaraby and Abdul-Mageed, 2018; Cotterell and Callison-Burch, 2014) and a country-level (Talafha et al., 2020; Abdelali et al., 2021; AlKhamissi et al., 2021).

In recent years, more efforts have targeted Arabic DID on a more fine-grained level, particularly through shared tasks. The Nuanced Arabic Dialect Identification Shared Tasks (NADI) (Abdul-Mageed et al., 2020, 2021b, 2022) include sub-tasks on country- and province-level on user-generated tweets. Several corpora of written Arabic dialects comprise tweets (Abdelali et al., 2021; Abdul-Mageed et al., 2018; Zaghouani and Charfi, 2018), others consist of user commentaries (Zaidan and Callison-Burch, 2011), or manually translated sentences (Bouamor et al., 2018, 2014).

For the NADI shared tasks (Abdul-Mageed et al., 2020, 2021b, 2022), all the top performing systems used transformer-based language models pre-trained on dialectal Arabic. However, these models yielded unsatisfactory results and multiple factors were identified, including imbalanced class distribution (AlShenaifi and Azmi, 2020), a significant presence of MSA content in the training data (Touileb, 2020), and the inherent challenges associated with distinguishing between Arabic dialects.

Within the MADAR shared task (Bouamor et al., 2019), the top five performing systems demonstrate that ensemble techniques, n-gram-based features, and traditional machine learning approaches, such as MNB or Support Vector Machines (SVMs), yield the highest levels of performance. While the MADAR corpus proved to be too small for deep learning architectures (Lippincott et al., 2019), the transfer learning ability of BERT-based language models, pre-trained on dialectal Arabic, has shown promising results (Seelawi et al., 2021; Inoue et al., 2021). However, the MNB model introduced by Salameh et al. (2018) is still state-of-the-art with an overall accuracy of 67.9%.

	MADAR-26		MADAR-6	
Sentences	Per dialect	Total	Per dialect	Total
Train	1600	41600	9000	54000
Dev	200	5200	1000	6000
Test	200	5200	-	-

Table 1: Number of sentences per dialect and per split in the MADAR-26 and MADAR-6 corpora.

	Avg.	Min	Max
Tokens	11265.42 (±619)	Basra	MSA
Sent length	5.61 (±0.3)	Basra	MSA
Vocabulary (types)	3273.61 (±204)	Doha	MSA

Table 2: Data statistics for MADAR-26, showing the average number of tokens, average sentence length, and vocabulary size (number of types) across dialects without punctuation. Min and max denote the dialect with the lowest and highest values for each statistic. The numbers in parentheses denote variance.

3 The MADAR corpus

The MADAR corpus is a collection of parallel sentences in the travel domain (Bouamor et al., 2018). The resource contains two corpora with non-overlapping sentences: (1) MADAR-26: covering 25 cities and MSA, and where each dialect is represented with 2000 sentences. (2) MADAR-6: covering the five selected cities Doha, Beirut, Rabat, Cairo, and Tunis, in addition to MSA, each with 12000 sentences. We use the training, development, and test splits from the MADAR shared task 1 (Bouamor et al., 2019) shown in Table 1. As can be seen, all classes are perfectly balanced for each set. In our models, we use MADAR-26 for both training and evaluation, while MADAR-6 is included in the training data of the state-of-the-art system presented by Salameh et al. (2018).

Throughout this work, we define tokens based on white space using the simple word tokeniser from CAMeL Tools[2] to split the sentences. Additionally, all punctuation are removed.

3.1 Corpus statistics

The MADAR-26 training data primarily consists of short sentences, with an average length of 5.6 tokens, as seen in Table 2. Short sentences can be challenging for DID, as they may not encompass enough information to capture the nuances of dialectal variations (Malmasi et al., 2016). The data

[2] https://github.com/CAMeL-Lab/camel_tools/tree/master/camel_tools/tokenizers

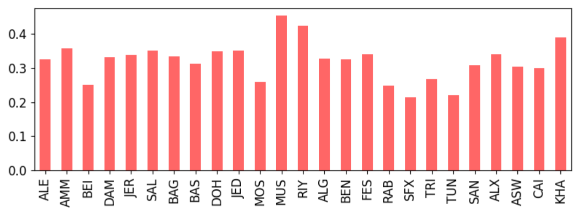


Figure 1: Vocabulary overlap between MSA and the dialects in the training data

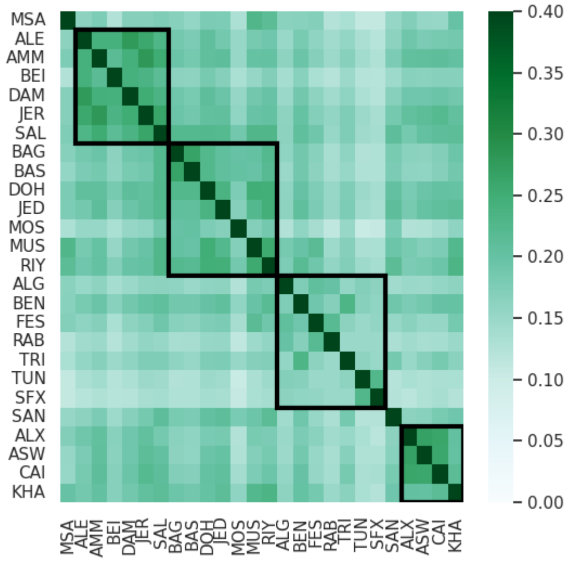


Figure 2: Heatmap of the lexical similarity computed with Overlap coefficient between the dialects in the MADAR-26 training data. The black boarders outline the geographical regions. For a clearer view of nuances, the heatmap threshold is set to 0.40, while some dialects might have a higher score.

also shows variations in vocabulary size across the dialects, where MSA consistently has the largest values. In contrast, the Doha dialect exhibits the smallest vocabulary, and the Basra dialect has the shortest sentences and the lowest number of tokens.

3.2 Lexical overlap

We here explore the degree of lexical overlap between the dialects by analysing the number of common tokens between them. We follow the work of Bouamor et al. (2018) and use the Overlap Coefficient (OC) to measure the degree of similarity between two sets of texts A and B, ranging from 0 (no overlap) to 1 (complete overlap).[3]

Lexical overlap with MSA The diglossic situation of Arabic puts MSA in a distinctive position concerning lexical overlap, given its presence in the daily language use of all dialect users. The source sentences for translation in the MADAR corpus were provided in English and French to minimise the bias of MSA (Bouamor et al., 2018).

As demonstrated in Figure 1, the OC between MSA and each dialect varies and ranges from 0.2 for Sfax and Tunis to over 0.4 for Muscat and Riyadh. While some of the overlap might stem from various bias factors in the translation process, it is also plausible that some of the overlapping vocabulary consists of function words and nouns that are shared with MSA. Rather than considering the vocabulary overlap as noise, it should be a factor when interpreting the results of DID. More specifically, this overlap might suggest that distinguishing MSA from Muscat or Riyadh might be more challenging than from Tunis or Sfax.

Lexical overlap between dialects By calculating the OC for every pair of dialects in the training data, we find the average pairwise similarity between them to be 0.35 with a standard deviation of 0.07. The OC across all dialect pairs can be seen in Figure 2, where the black borders outline the geographical regions Levant, Gulf, Maghreb, and Nile basin. The highest levels of overlap is between dialects within the same geographical region. The most prominent example is the Levant region, where most dialects have a high OC with each other. We find a similar pattern in the Gulf region, except for the Mosul dialect. For the Maghreb region, the overlap is less significant across the region, but higher for dialects within the same country (*e.g.* Rabat and Fes). Interestingly, there seems to be a high level of overlap between the Egyptian city dialects, Cairo, Aswan, and Alexandra, while Khartoum (Sudan), displays a slightly lower overlap. Sanaa is not included in any region, while it seems to have similar vocabulary to both the dialects in the Nile Basin region and the Gulf.

Tunis and Sfax city dialects exhibit relatively low levels of lexical overlap with dialects outside Tunisia, indicating a more distinct vocabulary. A similar pattern is noticeable in the Moroccan city dialects of Fes and Rabat. With an average vocabulary of approximately 3000 tokens, several dialects have fewer than 400 tokens that do not overlap

[3] Defined as: $OC(A, B) = \frac{|A \cap B|}{min(|A|, |B|)}$

with other dialects. This highlights the lack of clear class boundaries and emphasises the challenge of automatically identifying the dialects.

There is a more nuanced distribution of the linguistic features and characteristics of the dialects. There are morphological and lexical differences between the dialects, as well as significant vocabulary similarity within each region. More details about this lexical analysis can be found in Appendix A.1.

4 Models

We here describe our experimental set-up and the tested models.

4.1 Pre-trained Transformer language models

We evaluate three BERT models pre-trained on dialectal Arabic, AraBERTv0.2-Twitter[4](Antoun et al., 2020), MARBERTv2[5](Abdul-Mageed et al., 2021a), and CAMeLBERT-Mix[6] (Inoue et al., 2021). We will refer to them as AraBERT, CAMeLBERT, and MARBERT respectively. There exist several BERT models pre-trained on Arabic dialects. However, to the best of our knowledge, AraBERT and MARBERT have not yet been evaluated on MADAR-26. CAMeLBERT model is considered one of the top-performing models on the task (Inoue et al., 2021), and is therefore included as a baseline.

While all are based on the BERT architecture (Devlin et al., 2019), specifically the "base" version, they differ in terms of their pre-training data, model size, and vocabulary (see details in Table 8 in Appendix A.2). Notably, AraBERT is the smallest model in terms of number of tokens (8.6B), compared to MARBERT (29B) and CAMeLBERT(17.3B). All models are pre-trained on various MSA and dialectal Arabic sources, all including tweets. However, CAMeLBERT has the most diverse dialectal pre-training data, including the MADAR parallel corpus (Inoue et al., 2021).

Experimental setup and data We follow the ALUE benchmark model (Seelawi et al., 2021): the pre-trained BERT encoder takes an Arabic sentence as input and generates contextualised embeddings. The CLS classification token is extracted from the final layer of BERT, passed through a linear layer, before a softmax function computes the predicted classes. Details on implementation and hyperparameter tuning are described in Appendix A.3. We fine-tune all models on the MADAR-26 training set, using the same data splits as supplied for the MADAR shared task (Bouamor et al., 2019). We perform dediacritisation on the data in alignment with the pre-training of the BERT models. Similarly to previous experiments on this corpus (Inoue et al., 2021), no additional pre-processing is done.

	Accuracy	F_1
CAMeLBERT	63.25 (±0.65)	62.69 (±0.06)
MARBERT	62.36 (±0.05)	61.76 (±0.72)
AraBERT	**65.19** (±1.60)	**65.64** (±0.51)

Table 3: Average results for BERT models after five runs on MADAR-26 development set. Corresponding standard deviation in parentheses. Numbers in bold indicate best results.

4.2 Multinomial Naive Bayes

For the MNB model, we use the CAMeL-tools (Obeid et al., 2020) implementation of Salameh et al. (2018). The system consists of two models, a main MNB model trained on MADAR-26 and a supporting MNB model trained on MADAR-6. This latter classifies each sentence into a dialect from MADAR-6, and then used as a feature in the main model. As the six dialects in MADAR-6 are from different regions in MADAR-26, we consider the supporting classifier a regional classifier.

Both the main and supporting MNB models use similar feature types but from different corpora. The supporting model uses TF-IDF weighted word and character n-grams, in addition to probability scores from 6 dialectal n-gram language models trained on MADAR-6. The main MNB model uses the same feature types but with probability scores from 26 n-gram LMs trained on MADAR-26. Additionally, it takes regional probability scores from the supporting MNB model's predictions.

Note that we also trained a logistic regression model with the same features, but due to its subpar results, we are not including it in the discussion.

4.3 Evaluation

Due to the perfectly balanced classes, we report the overall performance of all models using macro average F1. We also report precision, recall, and F1 for each individual dialect and the average for

[4] https://huggingface.co/aubmindlab/bert-base-arabertv02-twitter

[5] https://huggingface.co/UBC-NLP/MARBERTv2

[6] https://huggingface.co/CAMeL-Lab/bert-base-arabic-camelbert-mix

Region	Dialect	AraBERT (%)	MNB (%)	Diff
	MSA	75.4	72.3	+3.1
Levant	ALE	63.6	63.8	-0.2
	AMM	52.1	65.5	-13.4
	BEI	64.1	68.8	-4.7
	DAM	50.0	63.4	-13.4
	JER	55.5	61.0	-5.5
	SAL	51.2	56.4	-5.2
Gulf	BAG	64.0	65.9	-1.9
	BAS	60.9	66.3	-5,4
	DOH	61.4	68.9	-7.5
	JED	53.7	59.4	-5.7
	MOS	78.9	86.8	-7.9
	MUS	51.0	48.8	+2.2
	RIY	53.9	57.6	-3.7
Maghreb	ALG	72.7	81.5	-9.0
	BEN	60.5	68.5	-8.0
	FES	65.2	71.3	-6.1
	RAB	67.5	72.3	-4.8
	SFX	69.9	72.9	-3.0
	TRI	70.9	80.0	-9.1
	TUN	64.8	72.3	-7.5
	SAN	68.1	75.0	-6.9
Nile Basin	ALX	74.0	75.9	-1.9
	ASW	63.2	63.8	-0.6
	CAI	53.0	55.8	-2.8
	KHA	66.1	72.5	-6.4
Total		**63.4**	**67.3**	**-3.9**

Table 4: F_1 scores of the AraBERT and MNB models on MADAR-26 test set. Highest scores for each model are in blue, and lowest in red. Green shows where AraBERT has a higher score than MNB. The overall performance of the models is displayed in the final row and marked in bold.

each region for the best-performing model.

Based on the development results in Table 3, we find that AraBERT outperforms the other models, with an average accuracy of 65.19% and a macro-average F1 of 65.64%. These results are interesting, considering AraBERT's smaller pre-training data size compared to the other models. It is also noteworthy that even though CAMeLBERT has MADAR-26 included in the pre-training data, it is outperformed by AraBERT on the development data. We speculate that these outcomes stem from effective filtering and curation of the pre-training data of AraBERT. We inspect the results on the test data in more detail next.

5 Test results

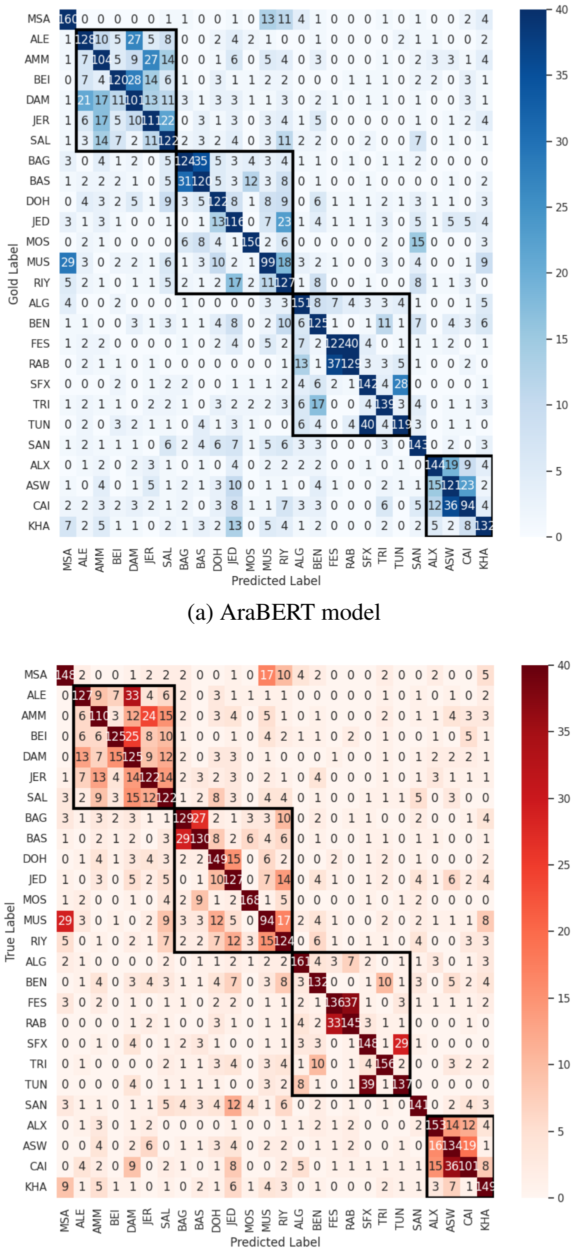
We compare the performance of both selected models, MNB and AraBERT, in terms of F_1 score for each individual dialect in Table 4. The results reveal a notable difference in their overall and individual dialect classification performance, with the MNB model outperforming AraBERT on the majority of dialects. As previously suggested, the results clearly show that the AraBERT model outperforms the MNB model on MSA and the Muscat dialect, with a difference of 3.1 and 2.2 pp, respectively. Interestingly, both models have the lowest performance on the Muscat dialect. We can also observe close performance on the Aleppo and Aswan dialects, while the most significant difference in performance is for the Amman and the Damascus dialects, where the MNB model outperforms the AraBERT model with 13.4 percentage points for both dialects. Due to the high lexical overlap between MSA and Muscat together with the high degree of MSA content in the pre-training data of the AraBERT model, it is likely that the AraBERT model is better at detecting MSA, and thereby not confusing the two dialects to the same degree as the MNB model. More details about the best classifications per dialect and model can be found in Table 11 in Appendix A.4.

6 Error analysis

We here provide a systematic analysis of the errors made by the different models.

6.1 Misclassification patterns

Analysing the confusion matrices in Figure 3, which visualises the two models' predictions, reveals distinct similarities in their misclassification patterns. (i) Most errors occur between city dialects from the same geographical regions (outlined with the black borders). For example, in the Levant region, Beirut is misclassified as Damascus, Amman, Aleppo, Jerusalem, and Salt by both models. We can also observe a high density within the dialects in the Nile basin region, while for the Maghreb and Gulf region, the overlap is more spread out. (ii) Both models' most frequent errors occur between city dialects from the same country. Notable examples are the two Moroccan city dialects Fes and Rabat, the Egyptian dialects Aswan, Cairo, and Alexandria, and the Iraqi dialects, Baghdad, and Basra. (iii) When considering the errors occurring outside the regional borders, we find that a significant proportion is associated with Arabic variants that are not attributed to any specific region, namely MSA and Sanaa. Among these outliers, the highest frequency of confusion is between the Muscat



(a) AraBERT model

(b) MNB model

Figure 3: Confusion matrices of (a) AraBERT and (b) MNB predcitions on the MADAR-26 test set.

dialect and MSA for both models.

In comparison to Figure 2 (Section 3.2), illustrating the lexical similarity between the dialects, we can observe some similarities. There is a high lexical similarity between all dialects in the Levant and Nile basin regions. Additionally, errors between dialects from the same country can be inferred from the patterns identified in Figure 2 and the misclassification pattern of MSA. But a few exceptions exist, such as the Mosul dialect having a high lexical similarity with Sanaa, which is only reflected in AraBERT's predictions. Another example is the Sanaa dialect, which displays high lexical similarity with both Doha and Jeddah, which is again evident in AraBERT's predictions, while the MNB model tends to confuse Sanaa with Jeddah.

Despite these exceptions, the high number of similarity patterns implies a positive correlation between high lexical similarity and the misclassifications for both models, but is not a complete explanation. By comparing the distribution of misclassification for each model, we discover that both models struggle with identifying the Muscat, Cairo, and Amman dialects. AraBERT exhibits more errors than the MNB model, particularly concerning the Damascus, Mosul, Tunis, and Doha dialects. However, the MNB model has a higher frequency of misclassifying the MSA, Aleppo, Sanaa, and Riyadh dialects when compared to AraBERT. Furthermore, we find that over 60% of the test data of both the Muscat and Cairo dialects is misclassified by either one or both models.

6.2 Subcategories of misclassified sentences

For our analysis, we are comparing the misclassifications made by the two models on a sentence level, as they can provide a more nuanced understanding of the performance and the difficulty of classifying certain dialects. We will base our error analysis on six categories of misclassifications that provide different insights: (1) Union of misclassification: includes all sentences misclassified by either one of the models or both. (2) Intersecting misclassification (INT): includes the sentences misclassified by both models. This subgroup is partitioned into two subcategories: (i) sentences in which both models have predicted the same incorrect dialect (INT-S), (ii) sentences in which the models have predicted two different dialects (INT-D). (3) Unique-AraBERT and (4) Unique-MNB: sentences misclassified by one model but correctly classified by the other.

The number of sentences for each category (Table 5), reveals interesting insights into the relative difficulty of the task. For example, out of the total of 1227 sentences misclassified by both models, they predicted the same incorrect dialect for 511 of them, while for 716 of the sentences, the models predicted different labels. While the models differ in their respective classification errors for a significant number of sentences, the number of intersecting misclassifications suggests that both models have a similar weakness in predicting a large sub-

	# Sentences	Avg.length
Total test set	5200	5.6 (±2.9)
Union	2415	5.0 (±2.5)
INT-S	511	4.8 (±2.5)
INT-D	716	4.5 (±2.3)
Unique-AraBERT	708	5.6 (±2.8)
Unique-MNB	480	5.5 (±2.9)

Table 5: Overview of number of test sentences and average sentence length for the different categories of misclassification. INT-S and INT-D refer to sentences wrongly classified by both models, where S and D denote whether both models made the same or different predictions. The Unique-AraBERT are the sentences correctly classified by MNB but misclassified by AraBERT, and vice versa for the Unique-MNB category. Union refers to all misclassified sentences regardless of model.

set of the corpus. The high number of sentences in the INT-S category implies that there might be patterns or linguistic features that present challenges for both models, revealing areas where the models have the most difficulty distinguishing between dialects. The INT-D sentences might present insight into particular challenging sentences, as neither model could predict the correct sentence.

6.3 Most frequently confused pair of dialects

We also provide insights into which dialect combinations are most frequently confused. We report on occurrences where a pair of dialects appear together, whether the dialect is a gold or a predicted label, for the same sentence.[7] The two Moroccan city dialects Rabat and Fes are the most frequently confused pair for all categories, except for the INT-D category. In this category, the models' misclassifications are less consistent, leading to less frequent occurrences of dialect pairs.

6.4 Potential sources of error

Table 5 shows the sentence length for each subcategory of misclassified sentences and the total test set. The test data has a similar average length to the training data but with greater variance, and even includes sentences with only one or two tokens, such as the Tripoli sentence فكرة حلوه (*Nice idea*). This may challenge classification, particularly when the tokens are shared among multiple dialects. The shortest sentences are found in the INT-D category, followed by IND-S, which both models misclassified. Interestingly, the unique misclassifications for each model consist, on average, of more tokens compared to the test set average. This suggests that the shorter sentences pose a shared challenge, while the unique misclassified sentences exhibit other challenges particular to each model.

Lexical overlap, the overlap in tokens between two bodies of texts, provides an indication of the extent to which a sentence is a subset of a given dialect's training data. It can also assess the degree to which a misclassified sentence represents the dialect as it appears in the training data. The box plot in Figure 4 illustrates the distribution of the overlap coefficient between sentences in the subcategories of the gold dialects in Figure (a) and between the predicted dialects in Figure (b). The first box in both figures represents the OC between the full training data and the gold dialects vocabulary for comparison purposes.

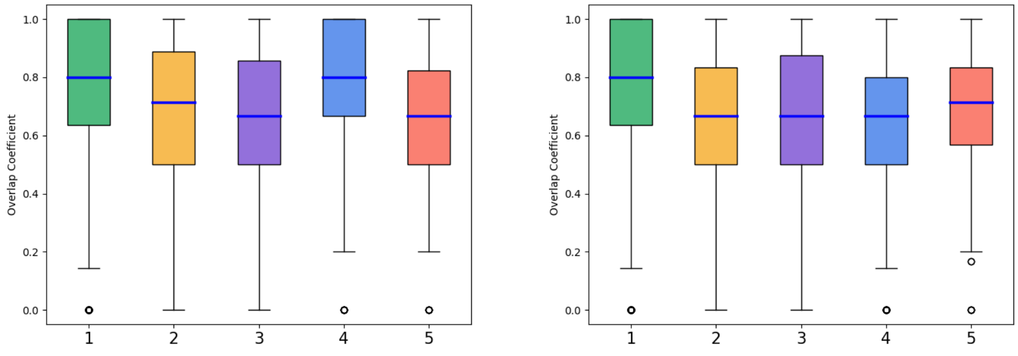
There are three notable observations. Firstly, the OC between the sentences in the test set for both the gold and predicted dialects tends to be high, with an average OC of over 0.5 for all categories in both figures. This trend may imply that certain sentences exhibit a significant vocabulary overlap between multiple dialects, leading to confusion for both models. Secondly, Figure (a) indicates that there are instances in the test data with an OC of 0.0 with the gold dialects, which can also be observed in the OC between the sentences and the predicted dialects in Figure (b), suggesting that lack of vocabulary overlap may be contributing to errors in some cases. Thirdly, box 4, representing the sentences misclassified only by AraBERT, has a higher median OC for the gold dialects compared to the other categories of misclassified sentences in Figure (a). However, in Figure (b), the median for box 4 is lower and more aligned with the other categories. These findings suggest that the AraBERT model tends to prioritise features other than lexical overlap when making predictions.

6.5 Manual example-level analysis

Due to the lack of morphological disambiguators covering all the dialects or regions in MADAR-26, we rely on manual example-level analysis.[8] As part of the comprehensive analysis conducted in Olsen

[7] The top five confused dialect pair for each category is reported in Table 13 in the Appendix.

[8] Since the objective here is to identify sources of misclassification, we will consider the sentences in their original form as input to the models. Consequently, the sentences lack vocalisation, and when analysing specific example sentences, we transcribe them letter-by-letter rather than supplementing the missing characters.



(a) OC with the gold dialects

(b) OC with the predicted dialects

Figure 4: Overlap coefficient (OC) between the test data and the vocabulary in the training data. Figure (a) shows the OC between the sentences and the gold dialects, while Figure (b) shows the OC between the sentences and the predicted dialects. Each box represents the OC for: Box 1: The test data and the gold labels. Box 2: The union of misclassifications of both models. Box 3: The intersection of misclassifications of both models. Box 4: The misclassifications unique to AraBERT. Box 5: The misclassifications unique to MNB.

(2023), the following examples are drawn from the set of 19 cases. We identified various characteristics that present challenges for dialect identification. For instance, there are sentences without dialect-specific features across all categories, as illustrated in Example 1.

(1) وين اتعلم السيد جونز اليابانية ؟
wiin it3lm al+siid jwnz al+iaabaania?
Where did mr. Jones learn Japanese?

In this instance, a sentence from Jerusalem is predicted as Riyadh by AraBERT and as Basra by MNB. Moreover, this sentence contains nouns typically unaffected by dialectal variation, like السيد جونز (*mr. Jones*) and اليابانية (*Japanese*).

For the second example, a Fes sentence is correctly predicted by the MNB model, but confused as Rabat by AraBERT. Rabat and Fes are the most frequently confused dialects, which might be explained by the lack of overlap with dialects outside Morocco (Figure 2), along with the prevalence of linguistic features only observed in the Moroccan dialects training data, such as ديال from Example 2. While these features are distinct enough to exclude the possibility of other dialects, they may not be sufficient to accurately distinguish between closely related dialects such as those from Rabat and Fes.

(2) خليوني ننعس حتا للتاسع ديال الصباح
khliwn+i nn3s h.taa l+ltaas3 diiaal al+s.baah.
Let me sleep until nine in the morning

6.6 Identical sentences in the test data

The MADAR corpus is stated to be created through manual independent translation of sentences in different dialects. However, we identify multiple occurrences of identical sentences in the test data labelled with different dialects. We will refer to these as *duplicates*. There is a total of 522 of duplicate sentences in the test data. 398 such sentences were misclassified by AraBERT, and 393 were misclassified by the MNB model.

Some entries have up to 11 duplicates labelled with different dialects from different regions. An example is the sentence برا, in English *Outside*, which has gold labels from The Levant, Gulf, and Maghreb regions. Most frequent duplicates are very short, some consisting of only one token. The total set of duplicate sentences has an average length of 3.50 tokens with a standard deviation of 1.55, which might explain the high number of identical sentences across multiple regions.

The distribution of the duplicates is skewed, with the highest frequencies among the Levant dialects Jerusalem, Salt, and Damascus, with over 30 sentences each. At the same time, MSA, Mosul, Algiers, Rabat, Sfax, and Sanaa have less than ten each. Furthermore, it appears like dialects with high lexical overlap (see Section 3), have similar amounts of duplicate sentences. See Figure 5 in the Appendix for the distribution across dialects.

Task formulation Because the task of Arabic DID is formulated as a multi-class classification task, many of the sentences in the test data are impossible to identify correctly since they can belong to multiple dialects. The limitations of this task formulation have already been demonstrated (Goutte et al., 2016; Zampieri et al., 2023), suggesting that

	Original	Dedpulicated
Size	5200	4870
Avg. sentences per class	200 (±0)	187.3 (±6.2)
Smallest class	–	Jerusalem (174)
Largest class	–	Sfax (198)

Table 6: Deduplicated MADAR-26 test set compared to the original test set with smallest and largest class.

unless a text belongs to precisely one dialect, the classification task should be approached as a multi-label classification task, rather than a multi-class one (Bernier-colborne et al., 2023).

Deduplication of test data We identify all instances of duplicates and remove them, with only one random instance retained in the test set. The resulting deduplicated test set is presented in Table 6, and consists of 4870 sentences. The result is an imbalanced test set, but, an argument can be made that duplicate sentences in the original test data already imbalanced the test set.

Model evaluation on deduplicated data We evaluate the MNB model on the deduplicated test set and achieve a macro-average F1 score of 70.25%. Compared to the performance on the original test set, evaluation without duplicate sentences across classes results in an increase in performance of 2.95 pp. The presence of duplicate sentences in the data can be viewed as a reflection of natural occurring language use, particularly in the case of short text, where phrases and expressions may be identical across various dialects. Therefore, removing identical sentences may introduce bias in the evaluation process, as it would not reflect the natural occurrence of such duplicates and could lead to an overestimation of a model's performance.

7 Conclusion and future work

This paper investigates the challenging task of fine-grained dialect identification, focusing on the MADAR-26 corpus. By fine-tuning three BERT models pre-trained on dialectal Arabic, we demonstrated that the multinomial naive bayes model introduced by Salameh et al. (2018) remains the state-of-the-art model on this data. However, we identified 480 test sentences that were correctly classified by the best performing BERT model, but were misclassified by the MNB model. A comprehensive error analysis revealed the BERT model exhibits superior performance in predicting sentences in Muscat dialect and MSA, which may be attributed to the amount of MSA content in the pre-training data of the BERT model. We also show that some of the challenges of the task can be attributed to dataset limitations. Particularly the fact that 10% of the sentences in the test set are identical to one or more parallel sentences in the same set but with different labels.

Our analysis of different error types confirms that the MNB and BERT-based model often make different mistakes, but also that a subset of the test data is challenging for both. Notably, we found that the Moroccan city dialects Rabat and Fes are the most confused dialect pair, and show how neither approach is able to capture the subtle distinctions between some of the closely related dialects. Although dataset limitations, such as non-Arabic proper nouns, short sentences without dialect-specific features, and identical sentences across classes, account for some of these errors, the unique errors generated by each model provide evidence that certain sentences can be correctly classified by one model, but not the other. These findings underscore the need to examine model performance beyond simple metric comparison in order to identify new strategies for enhancing Arabic dialect identification.

In the future, we would like to address the formulation of the task, by transforming it into a multi-label classification problem. Instead of simply removing the duplicate sentences from the data, we can combine the labels of duplicate and nearly-duplicate text, converting the single-label dataset into a multi-label dialect classification format.

Another avenue for future research is to evaluate models trained or fine-tuned on MADAR-26 on user-generated data. Due to a lack of annotated data matching the city-levels of MADAR, evaluation on data outside the travel domain has up until recently not been possible. However, the hierarchical mapping schema proposed by Baimukan et al. (2022) can be leveraged for datasets with comparable or more detailed annotations. More specifically, we want to evaluate the performance of the models on the NADI dataset (Abdul-Mageed et al., 2020) by mapping tweets at the province-level to the city-level.

Limitations

Given the scope of this work, we did not conduct an extensive exploration of design choices for the various models, or dedicate considerable time to hyperparameter optimisation and experimentation of the selected models. Nevertheless, we acknowledge that a more rigorous pursuit of hyperparameter tuning may potentially produce different results.

Despite evaluating the Transformer-based models using five different seeds, our error analysis relies solely on the outcomes of a single run. Although the AraBERT model displayed a small degree of instability during the development phase, some of the outcomes used in the error analysis may have varied if a different seed was used. However, due to the extensive nature of the analysis, incorporating outcomes from multiple runs was not a practical option. Therefore, our findings should be considered indicative rather than definitive.

Moreover, the error analysis focused solely on the test set without comparing misclassified and correctly predicted sentences, and thereby limiting our ability to pinpoint the precise factors behind misclassifications. Instead, it offers insights into misclassification categories and variations between types, as well as between the two models.

Because of the wide coverage of the MADAR-26 corpus, some of the dialects in our error analysis are outside our expertise. To mitigate this limitation, we employed the newly publicly released MADAR lexicon (Bouamor et al., 2018) and other resources to aid in analysing these languages. However, inaccuracies may still exist.

Finally, due to the lack of morphological analysers covering all the dialects in MADAR-26, we performed analysis on token-level, where a token is defined by whitespace. This is not optimal for Arabic, as this approach may result in the loss of information conveyed by clitics.

Acknowledgements

The computations were performed on resources provided through Sigma2 – the national research infrastructure provider for High-Performance Computing and large-scale data storage in Norway.

References

Ahmed Abdelali, Hamdy Mubarak, Younes Samih, Sabit Hassan, and Kareem Darwish. 2021. Qadi: Arabic dialect identification in the wild. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, page 1–10, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Muhammad Abdul-Mageed, Hassan Alhuzali, and Mohamed Elaraby. 2018. You tweet what you speak: A city-level dataset of arabic dialects. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2021a. 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.

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, Barcelona, Spain (Online). Association for Computational Linguistics.

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, Houda Bouamor, and Nizar Habash. 2021b. NADI 2021: The second nuanced Arabic dialect identification shared task. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 244–259, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, Houda Bouamor, and Nizar Habash. 2022. NADI 2022: The third nuanced Arabic dialect identification shared task. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 85–97, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Badr AlKhamissi, Mohamed Gabr, Muhammad El-Nokrashy, and Khaled Essam. 2021. Adapting MARBERT for improved Arabic dialect identification: Submission to the NADI 2021 shared task. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 260–264, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Nouf AlShenaifi and Aqil Azmi. 2020. Faheem at NADI shared task: Identifying the dialect of Arabic tweet. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 282–287, Barcelona, Spain (Online). Association for Computational Linguistics.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. AraBERT: Transformer-based model for Arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France. European Language Resource Association.

Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2022. Hierarchical aggregation of dialectal data for Arabic dialect identification. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4586–4596, Marseille, France. European Language Resources Association.

Gabriel Bernier-colborne, Cyril Goutte, and Serge Leger. 2023. Dialect and variant identification as a multi-label classification task: A proposal based on near-duplicate analysis. In *Tenth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2023)*, pages 142–151, Dubrovnik, Croatia. Association for 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).

Houda Bouamor, Sabit Hassan, and Nizar Habash. 2019. The MADAR shared task on Arabic fine-grained dialect identification. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 199–207, Florence, Italy. Association for Computational Linguistics.

Ryan Cotterell and Chris Callison-Burch. 2014. A multi-dialect, multi-genre corpus of informal written Arabic. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 241–245, Reykjavik, Iceland. European Language Resources Association (ELRA).

Gael de Francony, Victor Guichard, Praveen Joshi, Haithem Afli, and Abdessalam Bouchekif. 2019. Hierarchical deep learning for arabic dialect identification. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, page 249–253, Florence, Italy. 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.

Mona Diab and Nizar Habash. 2007. Arabic dialect processing tutorial. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Tutorial Abstracts*, pages 5–6, Rochester, New York. Association for Computational Linguistics.

Mohamed Elaraby and Muhammad Abdul-Mageed. 2018. Deep models for Arabic dialect identification on benchmarked data. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pages 263–274, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Heba Elfardy and Mona Diab. 2013. Sentence level dialect identification in Arabic. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 456–461, Sofia, Bulgaria. Association for Computational Linguistics.

Asma Etman and AA Louis Beex. 2015. Language and dialect identification: A survey. In *2015 SAI intelligent systems conference (IntelliSys)*, pages 220–231. IEEE.

Abbas Ghaddar, Yimeng Wu, Sunyam Bagga, Ahmad Rashid, Khalil Bibi, Mehdi Rezagholizadeh, Chao Xing, Yasheng Wang, Xinyu Duan, Zhefeng Wang, Baoxing Huai, Xin Jiang, Qun Liu, and Phillippe Langlais. 2022. Revisiting pre-trained language models and their evaluation for Arabic natural language processing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3135–3151, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Cyril Goutte, Serge Léger, Shervin Malmasi, and Marcos Zampieri. 2016. Discriminating similar languages: Evaluations and explorations. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1800–1807, Portorož, Slovenia. European Language Resources Association (ELRA).

Nizar Y. Habash. 2010. *Introduction to Arabic natural language processing*, 1 edition, volume 3 of *Synthesis Lectures on Human Language Technologies*. Morgan and Claypool Publishers.

Salima Harrat, Karima Meftouh, Mourad Abbas, Khaled-Walid Hidouci, and Kamel Smaïli. 2016. An algerian dialect: Study and resources. *International Journal of Advanced Computer Science and Applications*, 7:384–396.

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.

Tom Lippincott, Pamela Shapiro, Kevin Duh, and Paul McNamee. 2019. Jhu system description for the madar arabic dialect identification shared task. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, page 264–268, Florence, Italy. Association for Computational Linguistics.

Shervin Malmasi, Eshrag Refaee, and Mark Dras. 2016. Arabic dialect identification using a parallel multidialectal corpus. In *Computational Linguistics*, page 35–53, Singapore. Springer Singapore.

Ossama Obeid, Nasser Zalmout, Salam Khalifa, Dima Taji, Mai Oudah, Bashar Alhafni, Go Inoue, Fadhl Eryani, Alexander Erdmann, and Nizar Habash. 2020. CAMeL tools: An open source python toolkit for Arabic natural language processing. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 7022–7032, Marseille, France. European Language Resources Association.

Helene Bøsei Olsen. 2023. Fine-grained arabic dialect identification.

Mohammad Salameh, Houda Bouamor, and Nizar Habash. 2018. Fine-Grained Arabic Dialect Identification. In *Proceedings of the 27th International Conference on Computational Linguistics*, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Haitham Seelawi, Ibraheem Tuffaha, Mahmoud Gzawi, Wael Farhan, Bashar Talafha, Riham Badawi, Zyad Sober, Oday Al-Dweik, Abed Alhakim Freihat, and Hussein Al-Natsheh. 2021. ALUE: Arabic language understanding evaluation. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 173–184, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Bashar Talafha, Mohammad Ali, Muhy Eddin Za'ter, Haitham Seelawi, Ibraheem Tuffaha, Mostafa Samir, Wael Farhan, and Hussein T. Al-Natsheh. 2020. Multi-dialect arabic bert for country-level dialect identification. ArXiv:2007.05612 [cs].

Christoph Tillmann, Saab Mansour, and Yaser Al-Onaizan. 2014. Improved sentence-level arabic dialect classification. In *Proceedings of the First Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects*, page 110–119, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

Samia Touileb. 2020. LTG-ST at NADI shared task 1: Arabic dialect identification using a stacking classifier. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 313–319, Barcelona, Spain (Online). Association for Computational Linguistics.

Wajdi Zaghouani and Anis Charfi. 2018. Arap-tweet: A large multi-dialect twitter corpus for gender, age and language variety identification. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Omar F. Zaidan and Chris Callison-Burch. 2011. The Arabic online commentary dataset: an annotated dataset of informal Arabic with high dialectal content. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 37–41, Portland, Oregon, USA. Association for Computational Linguistics.

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

Marcos Zampieri, Kai North, Tommi Jauhiainen, Mariano Felice, Neha Kumari, Nishant Nair, and Yash Bangera. 2023. Language variety identification with true labels.

A Appendix

A.1 Morphological and lexical differences between dialects

Region	Dialect	Sentence
	English	We want a table close to the stage
	MSA	نريد مائدة بالقرب من المسرح
Gulf	Muscat	نبغا طاولة بالقرب من المسرح
	Doha	بغينا طاولة يم المسرح
	Riyadh	نبغى طاولة قريبة من المسرح
	Jeddah	نبا طاولة جمب المسرح
	Baghdad	نريد طاولة قريبة على الستيج
	Basra	نريد ميز يم الستيج
	Mosul	نغيد ميز قغيب على المسرج
Gulf of Aden	Sanaa	نشتي طاوله قريب من المنصه
Levant	Aleppo	بدنا طاولة جنب المنصة
	Damascus	بدنا طاولة قريبة عالمنصة
	Beirut	بدنا طاولة حد المسرح
	Amman	بدنا طاولة قريبة من المسرح
	Salt	بدنا طاولة قريب من المسرح
	Jerusalem	بدنا طاولة جنب المسرح
Nile Basin	Cairo	عايز ترابيزه جنب المنصة
	Alexandria	عاوزين ترابيزة قريبة من المسرح
	Aswan	أحنا عايزين طربيزة قريبة من المسرح
	Khartoum	دايرين طربيزة جنب المنصة
Maghreb	Tripoli	نبو طاوله جنب المسرح
	Benghazi	نبو طاولة قريبة من المسرح
	Tunis	نحبو طاولة قريبة م الركح
	Sfax	نحبوا طاولة بجنب الواد
	Algiers	رانا حابين طاولة قريبة من منصة العرض
	Rabat	بغينا طاولة قريبة للمسرح
	Fes	بغينا طبلة قريبة للمسرح

Table 7: A sample of a 26-way parallel sentence extracted from MADAR-26 for the English sentence "*We want a table near the stage.*"

To get a more nuanced understanding of the linguistic features and characteristics of the dialects, we analyse the sentence "*We want a table close to the stage*" for all the dialects, see Table 7, as we believe it highlights many of the morphological and lexical differences between the dialects. For example, the English word *table* is translated into مائدة in MSA, while for Basra and Mosul it is ميز, and طربيزة in Aswan. It is translated into طاولة in multiple city dialects in the Gulf, Levant and Maghreb region. Translating the word *table* into طاولة makes sense for many of the dialects in the Levant and in the Gulf, while for others, this translation choice seems to have been influenced by MSA. For instance, in the Algiers dialect, many Algiers dialect speakers view طاولة as a MSA word and prefer the French-derived term طابلة in their daily communication (Harrat et al., 2016).

When examining sentences regionally, we find significant vocabulary similarity within each region. As an example, in the Levant region, all city dialects translate *We want a table* as بدنا طاولة. This contributes to the complexity of DID at a city-level, particularly in distinguishing between cities in the same geographical area.

A.2 Arabic BERT-based models

	Size	#Tokens	pre-training data
AraBERT	541MB	8.6B	77GB+60M Tweets
MARBERT	654MB	29B	167GB
CAMeLBERT	439MB	17.3B	167GB

Table 8: Configuration for AraBERTv0.2-Twitter (Antoun et al., 2020), MARBERTv2 (Abdul-Mageed et al., 2021a) and CAMeLBERT-mix (Inoue et al., 2021).

A.3 Implementation details

All the reported experiments are run on the high-performance computing resource Sigma2 – the National Infrastructure for High Performance Computing and Data Storage in Norway, made available by the University of Oslo. For replicability, we do not train the BERT models from scratch, relying instead on pre-trained BERT models downloaded directly from Huggingface.

Hyperparameter tuning For all experiments, we are using maximum length of 128 tokens for input sequences, AdamW optimiser with epsilon at 1e-8, and early stopping to determine the optimal number of epochs. To compute the loss, we use Cross entropy and set dropout to 0.1.

We are not experimenting with different hyperparameters for the CAMeL-BERT model, as previous work has made a thorough effort to explore the optimal combination for the model on the task of DID on MADAR-26 (Inoue et al., 2021; Ghaddar et al., 2022). Additionally, we run each model multiple times with different seeds to capture potential deviations in performance (Devlin et al., 2019).

In the case of AraBERT and MARBERT, we base our hyperparameter grid search on previous experiments on earlier versions of the models, namely AraBERTv0.2 (Antoun et al., 2020) and MARBERTv1 (Abdul-Mageed et al., 2021a), on the task of DID on MADAR-26 (Inoue et al., 2021; Ghaddar et al., 2022). Table 9 presents the results from the hyperparameter grid search, while Table 10 shows the hyperparameters used for evaluation on the development set for all models.

Model	Batch	Learning rate	
		2e-05	1e-4
MARBERT	32	62.56 (±0.63)	62.69 (±0.06)
	16	62.12 (±1.84)	61.76 (±1.72)
AraBERT	32	62.95(±0.57)	65.64 (±0.51)
	16	63.95(±0.40)	64.76 (±1.65)

Table 9: Average results for AraBERT-Twitter and MARBERTv2 on five seeds testing hyperparameters.

Model	Batch	Lr	Epochs
MARBERT	32	1e-4	6
AraBERT	32	1e-4	8
CAMeLBERT	32	2e-05	3

Table 10: Hyperparameters for AraBERTv0.2-Twitter (Antoun et al., 2020), MARBERTv2 (Abdul-Mageed et al., 2021a) and CAMeLBERT-mix (Inoue et al., 2021).

A.4 Percentage of correctly classified sentences

Table 11 displays the percentage of correctly classified test data for each dialect by each model, focusing on top five best- and top five worst-performing dialects. This summary demonstrates how the MNB model identifies a larger proportion of the sentences compared to the AraBERT model, both for the top five best and weakest results. The table confirms the models' differences in proficiency on various dialects, the most interesting example being the AraBERT model's high accuracy of 80% in predicting the MSA sentences, which is not among the top five results for the MNB model.

	AraBERT	MNB
1.	MSA (80.0%)	MOS (84.0%)
2.	ALG (75.5%)	ALG (80.5%)
3.	MOS (75.0%)	TRI (78.0%)
4.	ALX (72.0%)	ALX (76.5%)
5.	SAN (71.5%)	DOH (74.5%)
22.	JER (55.5%)	SAL (61.0%)
23.	AMM (52.0%)	JER (61.0%)
24.	DAM (50.5%)	AMM(55.0%)
25.	MUS (49.5%)	CAI (50.5%)
26.	CAI (47.0%)	MUS (47.0%)

Table 11: The five top and bottom dialects based on percentage of sentences predicted correctly by each model.

A.5 Cities covered in the MADAR corpus

In Table 12 we give the full list of all cities covered in the MADAR corpus, as well as the abbreviations of their names used throughout the paper.

Dialect city	Abbr.	Country	Region
Damascus	DAM	Syria	Levant
Aleppo	ALE		
Beirut	**BEI**	Lebanon	
Amman	AMM	Jordan	
Salt	SAL		
Jerusalem	JER	Palestine	
Muscat	MUS	Oman	Gulf
Doha	**DOH**	Qatar	
Riyadh	RIY	KSA	
Jeddah	JED		
Baghdad	BAG	Iraq	
Mosul	MOS		
Basra	BAS		
Sanaa	SAN	Yemen	Gulf of Aden
Tripoli	TRI	Libya	Maghreb
Benghazi	BEN		
Tunis	**TUN**	Tunisia	
Sfax	SFX		
Algiers	ALG	Algeria	
Rabat	**RAB**	Morocco	
Fes	FES		
Cairo	**CAI**	Egypt	Nile basin
Alexandria	ALE		
Aswan	ASW		
Khartoum	KHA	Sudan	

Table 12: The cities covered by MADAR-26 with corresponding country and region as defined by Bouamor et al. (2018). The cities included in MADAR-6 are marked with bold.

A.6 Most frequently confused pair of dialects

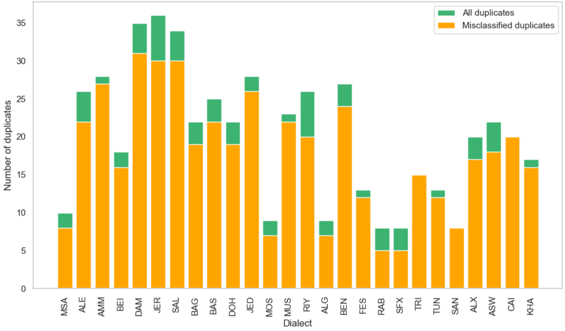


Figure 5: Distribution of duplicate and misclassified duplicate sentences for each dialect in the MADAR-26 test set.

As previously mentioned, the distribution of the duplicates is skewed. As can be seen from Table 5, Jerusalem, Salt, and Damascus (from the Levant region) are the dialects with most duplicates with over 30 sentences each. While MSA, Mo-

Category	Dialect pair	Frequency
Union	RAB, FES	109
	TUN, SFX	100
	BAG, BAS	86
	CAI, ASW	85
	DAM, ALE	61
	MSA, MUS	58
Int-S	RAB, FES	36
	TUN, SFX	30
	DAM, ALE	25
	BAG, BAS	23
	CAI, ASW	22
	DAM, BEI	20
Int-D	RIY, JED	14
	MSA, MUS	13
	BAG, BAS	13
	SAL, AMM	11
	JER, AMM	11
	JER, BEI	11
Unique AraBERT	RAB, FES	39
	TUN, SFX	32
	BAG, BAS	30
	ASW, CAI	30
	JER, AMM	19
Unique MNB	RAB, FES	28
	TUN, SFX	28
	MSA, MUS	26
	BAG, BAS	24
	ASW, CAI	18

Table 13: The five most frequently occurring pairs of dialects in each category. The frequency is based on whether the two dialects occur together, either where d1 is the correct dialect and d2 is the predicted dialect, or where d2 is the correct dialect and d1 is the predicted dialect. Dialect pairs that are not from the same country are marked with bold.

sul, Algiers, Rabat, Sfax, and Sanaa have less than ten each. It is quite clear that having duplicate sentences confuses the models, as the majority of duplicates were actually misclassified.

We report on occurrences where a pair of dialects appear together, either as a gold label or as the predicted label, to inspect which dialect combinations are most frequently confused. The results for each category are presented in Table 13, and show how the most frequently confused dialect pairs are city dialects from the same country. The two Moroccan city dialects Rabat and Fes are the overall most frequently confused dialect in all categories except for the INT-D category. The high frequency between them might be explained by the high lexical overlap in terms of shared tokens in the training data, as reported in Section 3.

The dialect pairs that are not from the same country are highlighted in bold, and they all belong to the Levant region. However, there is one exception - the MSA and Muscat pair, which occur together 58 times. Interestingly, this combination only occurs in the INT-S and the Unique MNB category, in addition to the union of misclassifications, which suggests that the MNB model might contribute more to this confusion than the AraBERT model.

The INT-D category stands out from the others in two ways. Firstly, the frequency of each pair is significantly lower compared to the other categories, suggesting that this subset of misclassifications might have less dialect-specific features. Secondly, it exhibits three dialect pairs that are not located in the same country.

Arabic Dialect Identification under Scrutiny: Limitations of Single-label Classification

Amr Keleg and **Walid Magdy**
Institute for Language, Cognition and Computation
School of Informatics, University of Edinburgh
akeleg@sms.ed.ac.uk, wmagdy@inf.ed.ac.uk

Abstract

Automatic Arabic Dialect Identification (ADI) of text has gained great popularity since it was introduced in the early 2010s. Multiple datasets were developed, and yearly shared tasks have been running since 2018. However, ADI systems are reported to fail in distinguishing between the micro-dialects of Arabic. We argue that the currently adopted framing of the ADI task as a single-label classification problem is one of the main reasons for that. We highlight the limitation of the incompleteness of the *Dialect* labels and demonstrate how it impacts the evaluation of ADI systems. A manual error analysis for the predictions of an ADI, performed by 7 native speakers of different Arabic dialects, revealed that $\approx 66\%$ of the validated errors are not true errors. Consequently, we propose framing ADI as a multi-label classification task and give recommendations for designing new ADI datasets.

1 Introduction

ADI of text is an NLP task meant to determine the Arabic Dialect of the text from a predefined set of dialects. Arabic dialects can be grouped according to different levels (1) major regional level: Levant, Nile Basin, Gulf, Gulf of Aden, and Maghreb (2) country level: more than 20 Arab countries, and (3) city level: more than 100 micro-dialects (Cotterell and Callison-Burch, 2014; Baimukan et al., 2022).

Different datasets were built curating data from various resources with labels of different degrees of granularities: (1) regional-level (Zaidan and Callison-Burch, 2011; Alsarsour et al., 2018), (2) country-level (Abdelali et al., 2021; Abdul-Mageed et al., 2022, 2023), or (3) city-level (Bouamor et al., 2019; Abdul-Mageed et al., 2020a, 2021b). Despite attracting lots of attention and effort for over a decade, ADI is still considered challenging, especially for the fine-grained distinction of micro-Arabic dialects on the country and city levels. This is generally demonstrated by the inability of ADI models to achieve high macro-F1 scores.

Dialects	**Sentence**
Iraq, Jordan, Lebanon, Libya, Oman, Palestine Qatar, Saudi Arabia, Sudan, Syria, Tunisia, Yemen	وين المحطة؟ **Where is the station?**
Iraq, Morocco, Qatar	شنو رقم الرحلة؟ **What is the flight/trip number?**

Table 1: The MADAR corpus (Bouamor et al., 2018) has English/French sentences manually translated into different Arabic dialects. The table shows two sentences having the same translation across multiple country-level dialects.

We believe that framing ADI as a single-label classification problem is a major limitation, especially for short sentences that might not have enough distinctive cues of a specific dialect as per Table 1. Therefore, assigning a **single dialect label** to each sentence either automatically (e.g.: using geotagging) or manually makes the labels incomplete, which in turn affects the fairness of the evaluation process. The single-label limitation for DI was also discussed for other languages such as French (Bernier-colborne et al., 2023).

The need for improving the framing of ADI and consequently the ADI resources was previously noted by Althobaiti (2020), who concluded the *Future Directions* section of her survey of Arabic Dialect Identification (ADI) with the following:

> “There is also a need to criticize the available resources and analyze them in order to find the gaps in the available ADI resources."

In this paper, we introduce the concept of *Maximal Accuracy* for ADI datasets having single labels. We then provide recommendations for how to build new ADI datasets in a multi-label setup to alleviate the limitations of single-label datasets. We hope that our study will spark discussions among

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 385–398
December 7, 2023 ©2023 Association for Computational Linguistics

the Arabic NLP community about the modeling of the ADI task, which would optimally lead to the creation of new datasets of more complete labels, and help in improving the quality of the ADI models. The main contributions of the paper can be summarized as follows:

1. Criticizing the current modeling of the ADI task as a single-label classification task by empirically estimating the *Maximal Accuracy* for multiple existing ADI datasets.
2. Performing an error analysis for an ADI model by recruiting native speakers of seven different country-level Arabic dialects.
3. Presenting a detailed proposal for how multi-label classification can be used for ADI.

2 How are Current ADI Datasets Built?

There have been multiple efforts to build several datasets for the ADI task using multiple techniques. We recognize four main techniques: (1) Manual Human Annotation, (2) Translating sentences into predefined sets of dialects, (3) Automatic labeling of data using distinctive lexical cues, and (4) Automatic labeling using geo-tagging.

A common limitation to all those techniques is modeling the task as a single-label classification task, where each sentence in the datasets is assigned to only one dialect while ignoring the fact that the same sentence can be valid in multiple dialects. Furthermore, each of these techniques has its own additional limitations that affect the quality of the labels as follows:

(1) Manual Human Annotation where annotators categorize Arabic sentences into one dialect from a predefined list of dialects (Zaidan and Callison-Burch, 2011; Huang, 2015; Malmasi et al., 2016; Zampieri et al., 2017, 2018).

Limitations: It was found that annotators over-identify their own native dialects (Zaidan and Callison-Burch, 2014; Abu Farha and Magdy, 2022). Therefore, the annotations for sentences that are valid in multiple dialects might be skewed toward the countries from which most of the annotators originate, causing a representation bias. Moreover, accurately determining the Arabic dialect of a sentence requires exposure to the different dialects of Arabic, which might not be a common case for Arabic speakers.

(2) Translation in which participants are asked to translate sentences into their native Arabic dialects (Ho, 2006-; Bouamor et al., 2014; Meftouh et al., 2015; Bouamor et al., 2018; Mubarak, 2018). If all the participants are asked to translate the same source sentences, then the dataset is composed of parallel sentences in various dialects. The main application of these datasets is to help develop machine translation systems, however, they are sometimes used for ADI. Figure 1 demonstrates how a corpus of parallel sentences is transformed into a corresponding DI dataset.

Limitations: While the labels of the corresponding DI dataset are correct, a source sentence might have the same translation in multiple Arabic dialects, Table 1. In such cases, a single-label classifier is asked to predict different *Dialect* labels despite the input sentence being the same.

Moreover, the syntax, and lexical items in the translated sentences might be affected by the corresponding syntactic and lexical features of the source sentences, especially if the source sentence is MSA or a variant of DA (Bouamor et al., 2014; Harrat et al., 2017). Such effects might make the translated sentences sound unnatural to native speakers of these dialects.

(3) Distinctive Dialectal Terms where text is curated based on the appearance of a term from a seed list of distinctive dialectal terms. These terms are used to automatically determine the dialect of the text (Alsarsour et al., 2018; Althobaiti, 2022).

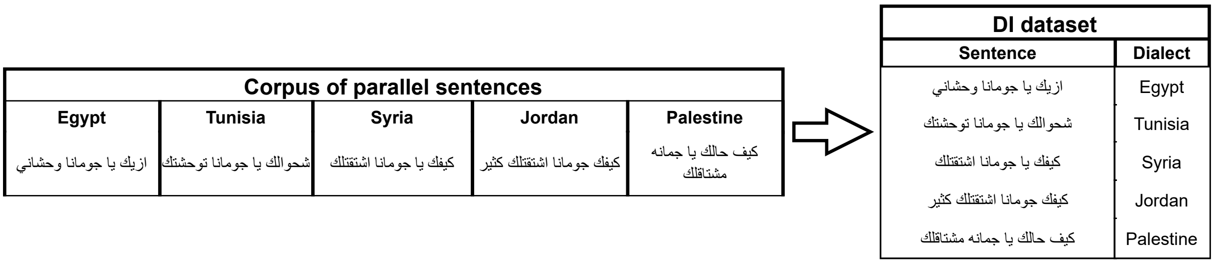
Limitations: The curated data is constrained by the diversity of the terms used to collect it.

(4) Geo-tagging where the text is automatically labeled using information about the location or the nationality of its writer (Mubarak and Darwish, 2014; Salama et al., 2014; Al-Obaidi and Samawi, 2016; Al-Moslmi et al., 2018; Zaghouani and Charfi, 2018; Charfi et al., 2019; El-Haj, 2020; Abdelali et al., 2021; Abdul-Mageed et al., 2020a, 2021b, 2022).

Limitations: While this technique allows for curating data from different Arab countries, it does not consider that speakers of a variant of DA might be living in an Arab country that speaks another variant (e.g.: An Egyptian living in Kuwait) (Charfi et al., 2019; Abdul-Mageed et al., 2020a). Moreover, some of the curated sentences might be written in MSA, so the curated sentences need to be split into DA sentences and MSA ones (Abdelali

Dataset	Ct/Cn/Re	Description
(1) Manual Labeling		
AOC (Zaidan and Callison-Burch, 2011)	- / - / 5 *	- Online comments to news articles, manually labeled three times by crowd-sourced human annotators.
Facebook test set (Huang, 2015) Note: Data attached to the paper on ACL Anthology.	- / - / 3	- 2,382 public Facebook posts manually annotated into Egyptian, Levantine, Gulf Arabic, and MSA.
VarDial 2016 (Malmasi et al., 2016) Note: The link provided is not working.	- / - / 4	- Sentences sampled from transcripts of broadcast, debate and discussion programs from AlJazeera. The dialects of these recorded programs were manually labeled. MSA is included as a 5th dialect class for the models. Audio features were used in the 2017 and 2018 editions to allow for building multimodal models.
VarDial 2017 (Zampieri et al., 2017)	- / - / 4	
VarDial 2018 (Zampieri et al., 2018) Note: VarDial 2018 used the same data as VarDial 2017.	- / - / 4	
ArSarcasm-v2 (Abu Farha et al., 2021)	- / - / 4 *	- 15,548 tweets sampled from previous sentiment analysis datasets, annotated for their dialect (including MSA).
(2) Translation		
Tatoeba (Ho, 2006-)	- / 8 / 4	- An ever-growing crowdsourced corpus of multilingual translations, that include MSA and 8 different Arabic dialects.
MPCA (Bouamor et al., 2014)	- / 5 / 3	- 2,000 Egyptian Arabic sentences from a pre-existing corpus, manually translated into 4 other country-level dialects in addition to MSA.
PADIC (Meftouh et al., 2015)	5 / 4 / 2	- 6,400 sentences sampled from the transcripts of recorded conversations and movie/TV shows in Algerian Arabic and manually translated into 4 other dialects and MSA.
DIAL2MSA (Mubarak, 2018)	- / - / 4	- Dialectal tweets manually translated into MSA.
MADAR6 (Bouamor et al., 2019)	5 / 5 / 4	- 10,000 sentences manually translated into 5 city-level Arabic dialects in addition to MSA.
MADAR26 (Bouamor et al., 2019)	25 / 15 / 5	- 2,000 sentences manually translated into 25 city-level Arabic dialects in addition to MSA.
(3) Distinctive Lexical Cues		
DART (Alsarsour et al., 2018)	- / - / 5 *	- Tweets streamed using a seed list of distinctive dialectal terms, which are used to initially assign a dialect to each tweet, before having them manually verified by crowdsourced annotators.
Twt15DA (Althobaiti, 2022) Note: Data shared as (tweet IDs, labels) only.	- / 15 / 5	- Tweets curated by iteratively augmenting lists of distinctive dialectal cues, starting with a seed list for each dialect.
(4) Geo-tagging		
(Mubarak and Darwish, 2014) Note: Not publicly available.	- / ? / ?	- Arabic tweets streamed from Twitter, then automatically annotated using the reported user locations of the tweets' authors.
YouDACC (Salama et al., 2014) Note: Not publicly available.	- / 8 / 5 *	- Comments to youtube videos labeled using the videos' countries of origin, and the authors' locations.
OMCCA (Al-Obaidi and Samawi, 2016)	5 / 2 / 2	- 27,912 reviews scrapped from `Jeeran.com`, and automatically labeled using the location of the reviewer.
MASC (Al-Moslmi et al., 2018)	- / 6 / 4	- 9,141 reviews curated from online reviewing sites, Google Play, Twitter, and Facebook. The country of the reviewer is used as a proxy for the dialect of the review.
Shami (Abu Kwaik et al., 2018)	- / 4 / 1	- Sentences in one of the 4 Levantine dialects: (1) manually collected from discussions about public figures on online fora; (2) automatically collected from the Twitter timelines of public figures.
ARAP-Tweet (Zaghouani and Charfi, 2018) Note: No download link on their site.	- / 16 / 5 *	- A corpus of tweets from 1100 users, annotated at the user level for the dialect, age, and gender.
ARAP-Tweet 2.0 (Charfi et al., 2019) Note: No download link on their site.	- / 17 / 5 *	- A corpus of tweets from about 3000 users, annotated at the user level for the dialect, age, and gender.
Habibi (El-Haj, 2020)	- / 18 / 6 *†	- Songs' lyrics labeled by the country of origin of their singers.
QADI (Abdelali et al., 2021) Note: Training data shared as (tweet IDs, labels) only.	- / 18 / 5	- Tweets automatically labeled based on the locations of the authors in the user description field. The labels of the testing set of each country were validated by a native speaker of each country's dialect.
NADI2020 (Abdul-Mageed et al., 2020a)	100 / 21 / 5	- Tweets of users staying in the same province for 10 months, automatically labeled by geotagging the tweets of the selected users.
NADI2021 (Abdul-Mageed et al., 2021b)	100 / 21 / 5	
NADI2022 (Abdul-Mageed et al., 2022)	- / 18 / 5	
NADI2023 (Abdul-Mageed et al., 2023)	- / 18 / 5	- Currently not disclosed
(5) Miscellaneous		
Arabic Dialects Dataset (El-Haj et al., 2018)	- / - / 4 *	- 12,801 sentences sampled from the AOC dataset, in addition to 3,693 sentences sampled from the *Internet Forums* category of the Tunisian Arabic Corpus (McNeil and Faiza, 2010-).

Table 2: The list of single-labeled ADI datasets categorized by the labeling techniques. We follow the regional categorization of Baimukan et al. (2022). **Ct/Cn/Re**: the number of cities (provinces), countries, and regions respectively. *: The regional dialects are defined as Egypt, Iraq, Levant, Gulf, and Maghreb (Cotterell and Callison-Burch, 2014). †: Sudanese Arabic is considered as another regional dialect. **?**: Missing information.



Corpus of parallel sentences				
Egypt	Tunisia	Syria	Jordan	Palestine
ازيك يا جومانا وحشاني	شحوالك يا جومانا توحشتك	كيفك يا جومانا اشتقتلك	كيفك جومانا اشتقتلك كثير	كيف حالك يا جمانه مشتاقلك

DI dataset	
Sentence	Dialect
ازيك يا جومانا وحشاني	Egypt
شحوالك يا جومانا توحشتك	Tunisia
كيفك يا جومانا اشتقتلك	Syria
كيفك جومانا اشتقتلك كثير	Jordan
كيف حالك يا جمانه مشتاقلك	Palestine

Figure 1: A demonstration of how parallel dialectal sentences are transformed into DI samples. The parallel sentences are sampled from the MPCA corpus (Bouamor et al., 2014)

et al., 2021; Abdul-Mageed et al., 2021b, 2022).

3 Maximal Accuracy of Single-label ADI Datasets

For a single-label ADI dataset consisting of sentences where each is assigned one dialect label, assume that a percentage $\mathbf{Perc_2}$ of those sentences is valid in $\mathbf{2}$ different dialects. For those sentences, only one of the valid dialects is listed as their label. An effective model trained to predict a single label will randomly assign each of these sentences to one of its two valid labels. Thus, the expected maximal accuracy on the dataset $\mathbf{E[Accuracy_{max}]}$ that the model can achieve would then be:

$$\mathbf{E[Accuracy_{max}] = (100 - Perc_2) + \frac{Perc_2}{2}} \quad (1)$$

For example, if 40% of the sentences are valid in two dialects (i.e.: $\mathbf{Perc_2} = 40\%$), then the $\mathbf{E[Accuracy_{max}]}$ of the dataset would be 80%. This becomes worse when a sentence is valid in more dialects, exceeding ten valid dialects in some cases (as shown in Table 1). Thus, for a total number of dialects $N_{dialects}$, the equation above can then generalized to:

$$\mathbf{E[Accuracy_{max}] = Perc_1 + \sum_{n=2}^{n=N_{dialects}} \frac{Perc_n}{n}} \quad (2)$$

where $\mathbf{Perc_1}$ is the percentage of samples that are only valid in one dialect, $\mathbf{Perc_n}$ is the percentage of samples valid in n dialects, $N_{dialects}$ represent the total number of dialects considered, and $\mathbf{\sum_{n=1}^{n=N_{dialects}} Perc_n = 100\%}$.

The higher the percentages $\mathbf{Perc_n}$ where $n \in [2, N_{dialects}]$, the lower the maximal accuracy would be. The same pattern would apply to F1 scores. Therefore, a model might be achieving low F1 scores as a consequence of framing DI as a single-label classification task, which might result in high $\mathbf{Perc_n}$ values.

Our objective in this paper is to estimate the value of $\mathbf{E[Accuracy_{max}]}$ for the existing datasets, which should examine the validity of our hypothesis that modeling ADI task as a single-label classification can be highly sub-optimal.

4 Estimating the Maximal Accuracy of Datasets

In our study, we focus on the country-level ADI for which multiple shared tasks have been organized since 2019 (Bouamor et al., 2019; Abdul-Mageed et al., 2020a, 2021b, 2022).

In order to quantify the percentages $Perc_n$, each sample of a dataset needs to be assessed by native speakers from all the Arab countries. Given our inability to recruit participants from all the Arab countries, we will estimate the percentages using two methods that provide lower bounds $\widetilde{\mathbf{Perc_n}}$ for the actual values $\mathbf{Perc_n}$ (i.e.: $\widetilde{\mathbf{Perc_n}} \leq \mathbf{Perc_n}$). Consequently, the estimated maximal accuracy is an upper bound for its true value.

4.1 Datasets Derived from Parallel Corpora

Initially, we examine the possibility of having Arabic sentences valid in multiple dialects by examining parallel corpora of Arabic dialects, which have sentences translated into multiple dialects. While a manual translation of a sentence can be phrased in different forms within the same dialect, we still examine if by chance we can find identical manually-translated sentences in different dialects by different translators.

For the four parallel corpora **Multidialectal Parallel Corpus of Arabic (MPCA)** (Bouamor et al., 2014), **PADIC** (Meftouh et al., 2015), **MADAR6**, and **MADAR26** (Bouamor et al., 2018), we transformed the parallel sentences into *(sentence, dialect)* pairs as in subtask (1) of the MADAR shared task (Bouamor et al., 2019). We then mapped the dialect labels for **PADIC**, **MADAR6**, and

Dataset	$\mathbf{N_{dialects}}$	$\mathbf{N_{samples}}$	$\sum_{n=2}^{n=N_{dialects}} \widetilde{\mathbf{Perc}}_{\mathbf{n}}$	$\widetilde{\mathbf{E}}[\mathbf{Accuracy_{max}}]$
PADIC	4	29,138	5.2%	97.1%
MPCA	5	4,960	7.8%	95.4%
MADAR6	5	49,476	2.3%	98.7%
MADAR26	15	48,624	9.6%	93.9%

Table 3: The estimated percentages and the corresponding expected maximal accuracy for the DI datasets formed using the four parallel corpora. The estimated maximal accuracies are upper bounds for the true maximal accuracies, and we expect the true values to be significantly lower than these estimates.

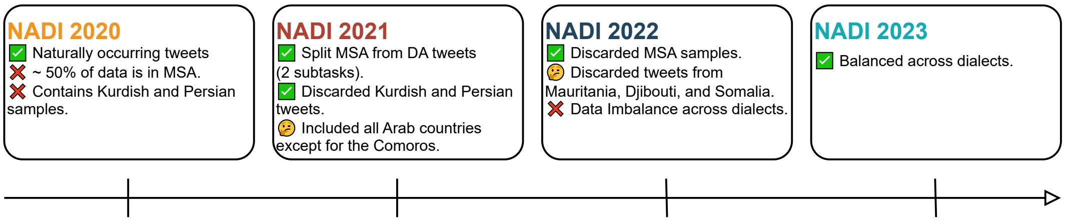

NADI 2020
✓ Naturally occurring tweets
✗ ~ 50% of data is in MSA.
✗ Contains Kurdish and Persian samples.
NADI 2021
✓ Split MSA from DA tweets (2 subtasks).
✓ Discarded Kurdish and Persian tweets.
🤔 Included all Arab countries except for the Comoros.
NADI 2022
✓ Discarded MSA samples.
🤔 Discarded tweets from Mauritania, Djibouti, and Somalia.
✗ Data Imbalance across dialects.
NADI 2023
✓ Balanced across dialects.

Figure 2: The evolution of the NADI datasets used for the shared tasks run between 2020 and 2023.

MADAR26 from city-level dialects to country-level ones. In case the same sentence is used in different cities within the same country, a single copy is kept. The sentences are then preprocessed by discarding Latin and numeric characters in addition to diacritics and punctuation. Lastly, we estimated the percentages $\widetilde{\mathbf{Perc}}_{\mathbf{n}}$ by computing the percentages of sentences that have the exact same translation in n dialects.

The upper bound for the maximal accuracies of the four corpora lies in the range $[93.9\%, 98.7\%]$ as per Table 3. The fact that the maximal accuracy for **MADAR26** is lower than that for **MADAR6** demonstrates that the probability that a sentence is valid in multiple dialects increases as more translations in other country-level dialects are considered.

4.2 Datasets of Geolocated Dialectal Sentences

The Nuanced Arabic Dialect Identification (NADI) shared tasks (Abdul-Mageed et al., 2020a, 2021b, 2022) used datasets that are built by collecting Arabic tweets authored by users who have been tweeting from the same location for 10 consecutive months. The geolocation of the users is then used as a label for their tweets. The creators of NADI have been improving the quality of the dataset from one year to another as summarized in Figure 2.

While the NADI shared tasks have been attracting active participation, the best-performing models in NADI 2022 achieved macro F1 scores of 36.48% and 18.95%, and accuracies of 53.05% and 36.84% on two test sets (Abdul-Mageed et al., 2022). The baseline MarBERT-based model (Abdul-Mageed et al., 2021a) fine-tuned on the training dataset achieves competitive results (macro F1 scores: 31.39% and 16.94%, accuracies: 47.77% and 34.06%).

Model Description Given the competitiveness of the baseline model, we fine-tuned the MarBERT model on the balanced training dataset of NADI 2023, and then we used the QADI dataset (Abdelali et al., 2021) as our test set. QADI's test set covers the same 18 countries as NADI 2023. We decided to analyze the errors of our model on QADI for two reasons: 1) At the time of writing the paper, the test set of NADI 2023 was not released (even for earlier NADIs, the labels of the test sets are not publicly released); 2) The dialect labels of the samples of QADI's test set were automatically assigned using geolocations similar to NADI, but the label of each sample was validated by a native speaker of the sample's label, which gives additional quality assurance for QADI over NADI.

The model achieves an accuracy of **50.74%** on QADI's test set with the full classification report in Table A2. Figure 3 visualizes how the predictions and labels are confused together.

Manual Error Analysis We recruited native-speaker participants of Algerian, Egyptian, Palestinian, Lebanese, Saudi Arabian, Sudanese, and Syrian Arabic to validate the False Positives (FPs) that the model makes for those dialects. Each participant is shown the FPs for their native dialect, one at a time, and is asked to validate them as indicated in Figure B1 [1]. If the participant found the FP sample to be valid in their native dialect, it means that this sample is valid in at least two different Arabic dialects (i.e.: the sample's original label, and the model's prediction) [2]. However, it can still be valid in additional dialects, which we did not check for due to the limited number of participants.

Validity of the Model's FPs Out of 490 validated FPs, 325 were found to be also valid in the other dialect they were classified to, which represents $\approx 66\%$ of the validated errors. Having such a great proportion of FPs that are not true errors hinders the ability to properly analyze and improve the ADI models. For Egyptian, Palestinian, Saudi Arabian, and Syrian Arabic, the majority of the FPs are incorrect as demonstrated in Figure 4 (i.e.: the model's prediction should be considered to be correct). As expected, dialects grouped in the same region are similar, and thus the FPs of a dialect would generally have labels of other dialects from the same region as in Figure 5.

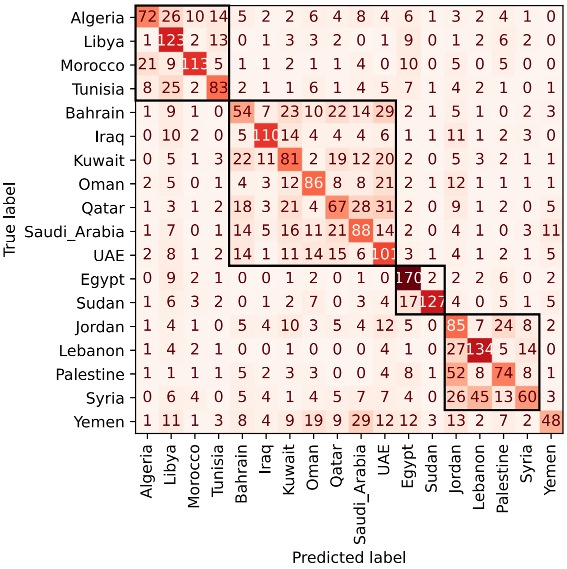
Impact on Evaluation If we only consider the 725 samples that were correctly predicted by the model (TPs) in addition to the validated 490 FPs, then we know that 325 samples out of 1215 ones are at least valid in two different dialects. The $\tilde{\mathbf{Perc}}_{\mathbf{2}}$ for this subset is 26.7%, making the maximal accuracy $\mathbf{E}[\mathbf{Accuracy}_{\mathbf{max}}]$ equal to 86.6%.

To further investigate the impact of the incorrect FPs on the evaluation metrics, we computed the corrected True Positive value for each dialect $\mathbf{TP}^*$ as $\mathbf{TP}^* = \mathbf{TP} + \mathbf{Incorrect\ FP}$. Using these corrected $\mathbf{TP}^*$ values, we computed corrected precision, recall, and F1-scores. As per Table 4, the macro-averaged F1-score increased from 0.56 to 0.72. This clearly confirms our hypothesis that modeling ADI task as a single-label classification task leads to inaccurate evaluation of the systems.

5 Proposal for Framing the ADI Task

Given the limitations of using single-label classification for the ADI task, elaborated in §4, we propose alternative modeling for ADI.

Zaidan and Callison-Burch (2014) asked crowd-sourced annotators to label dialectal sentences as being *Egyptian*, *Gulf*, *Iraqi*, *Levantine*, *Maghrebi*, *other dialect*, or *general dialect*. They used the *general dialect* for sentences that can be valid in multiple dialects. The *general dialect* is underspecified, and it is not clear whether it implies that a sentence is accepted in multiple dialects or in all of them. Therefore, the authors noticed that some of the annotators barely used the label, while others used it when they were not sure about the dialect of the underlying sentences. Moreover, they noticed



	Algeria	Libya	Morocco	Tunisia	Bahrain	Iraq	Kuwait	Oman	Qatar	Saudi_Arabia	UAE	Egypt	Sudan	Jordan	Lebanon	Palestine	Syria	Yemen
Algeria	72	26	10	14	5	2	2	6	4	8	4	6	1	3	2	4	1	0
Libya	1	123	2	13	0	1	3	3	2	0	1	9	0	1	2	6	2	0
Morocco	21	9	113	5	1	1	2	1	1	4	0	10	0	5	0	5	0	0
Tunisia	8	25	2	83	2	1	1	6	1	4	5	7	1	4	2	1	0	1
Bahrain	1	9	1	0	54	7	23	10	22	14	29	2	1	5	1	0	2	3
Iraq	0	10	2	0	5	110	14	4	4	4	6	1	1	11	1	2	3	0
Kuwait	0	5	1	3	22	11	81	2	19	12	20	2	0	5	3	2	1	1
Oman	2	5	0	1	4	3	12	86	8	8	21	2	1	12	1	1	1	1
Qatar	1	3	1	2	18	3	21	4	67	28	31	2	0	9	1	2	0	5
Saudi_Arabia	1	7	0	1	14	5	16	11	21	88	14	2	0	4	1	0	3	11
UAE	2	8	1	2	14	1	11	14	15	6	101	3	1	4	1	2	1	5
Egypt	0	9	2	1	0	0	1	2	0	1	0	170	2	2	2	6	0	2
Sudan	1	6	3	2	0	1	2	7	0	3	4	17	127	4	0	5	1	5
Jordan	1	4	1	0	5	4	10	3	5	4	12	5	0	85	7	24	8	2
Lebanon	1	4	2	1	0	0	1	0	0	0	4	1	0	27	134	5	14	0
Palestine	1	1	1	1	5	2	3	3	0	0	4	8	1	52	8	74	8	1
Syria	0	6	4	0	5	4	1	4	5	7	7	4	0	26	45	13	60	3
Yemen	1	11	1	3	8	4	9	19	9	29	12	12	3	13	2	7	2	48

Figure 3: The confusion matrix for the predictions of a MarBERT model on QADI's test set. The model was fine-tuned using NADI 2023's training dataset.

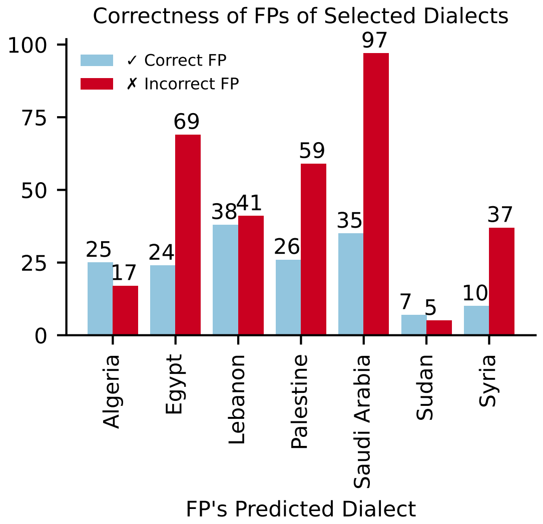


Figure 4: The distribution of the annotations for the validity of the False Positives (FPs) in 7 Arabic dialects. **Correct FP** represents the FP samples for which the model's prediction is invalid. **Incorrect FP** the FP samples for which the model's prediction is valid.

[1] We release the judgments through: github.com/AMR-KELEG/ADI-under-scrutiny/tree/master/data

[2] Participants are given a third choice *Maybe / Not Sure*, which we count as *No* (i.e.: invalid in their dialect).

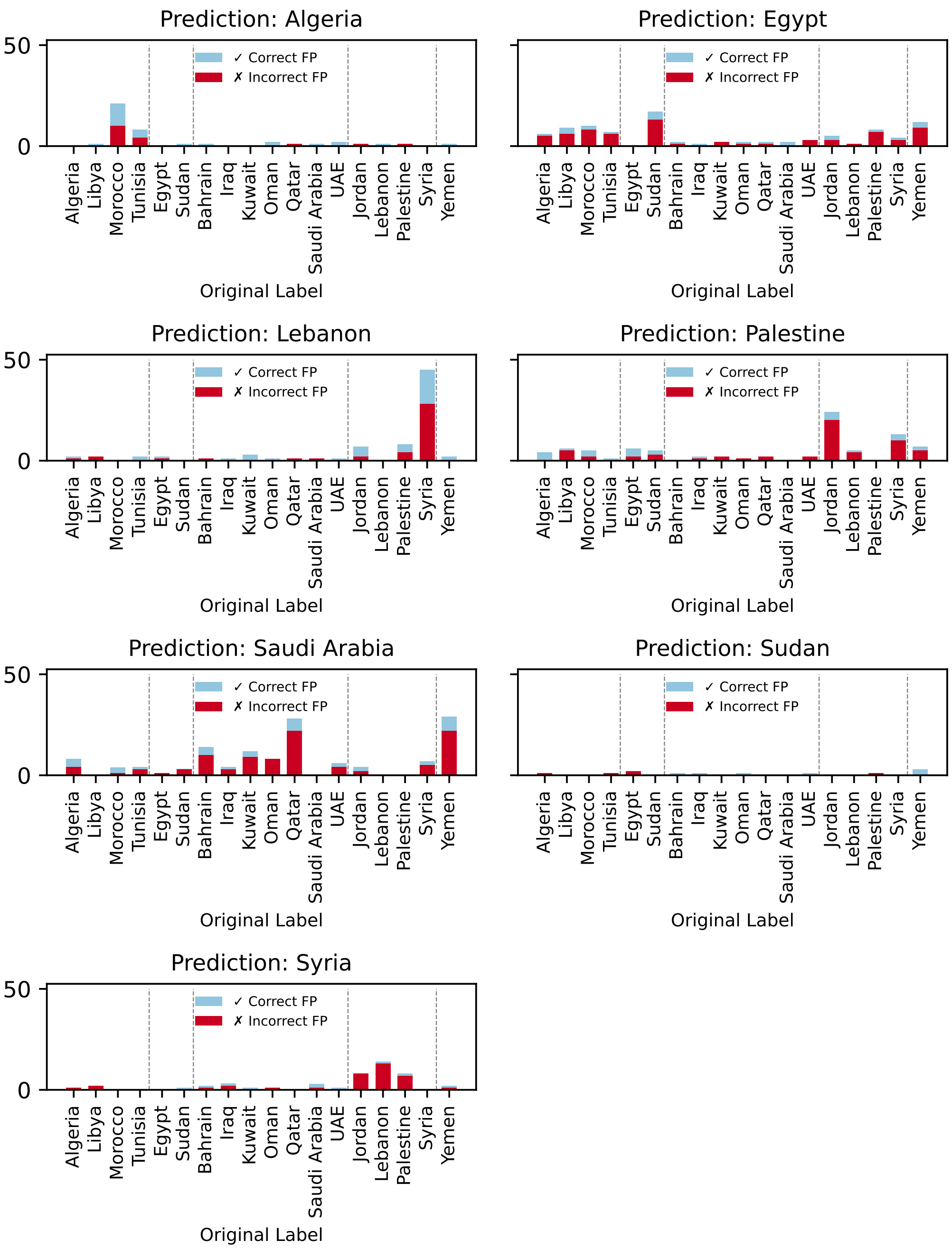


Figure 5: The distribution of the original labels for the False Positives (FPs) of the seven validated dialects. **Correct FP** represents the FP samples for which the model's prediction is invalid. **Incorrect FP** represents the FP samples for which the model's prediction is valid.

that the annotators tend to over-identify their native dialects. Annotators might not realize that a sentence valid in their native dialect is also valid in other dialects, and thus can end up choosing their native dialect as the label for this sentence, instead of the *general dialect* label.

Zampieri et al. (2023) focused on the binary distinction between two varieties of English, Portuguese, and Spanish. In addition to the two varieties of each language, the annotators are allowed to assign sentences to a third label *Both or Neither*. The evaluation results indicate that the *Both or Neither* label is harder to model computationally than the other variety labels. The authors noted that there is room for improvement in the treatment and modeling of this third label.

Consequently, we believe that adding another label such as *general* or *Both or Neither* does not completely solve the limitations of single-label classification datasets. Conversely, framing the

Dialect	TP	FP	TP*	FP*	FN	P	R	F1	P*	R*	F1*
Algeria	72	42	72 + 17 = 89	25	98	0.63	0.42	0.51	0.78	0.48	0.59
Egypt	170	93	170 + 69 = 239	24	30	0.65	0.85	0.73	0.91	0.89	0.90
Lebanon	134	79	134 + 41 = 175	38	60	0.63	0.69	0.66	0.82	0.74	0.78
Palestine	74	85	74 + 59 = 133	26	99	0.47	0.43	0.45	0.84	0.57	0.68
Saudi Arabia	88	132	88 + 97 = 185	35	111	0.40	0.44	0.42	0.84	0.62	0.72
Sudan	127	12	127 + 5 = 132	7	61	0.91	0.68	0.78	0.95	0.68	0.80
Syria	60	47	60 + 37 = 97	10	134	0.56	0.31	0.40	0.91	0.42	0.57
Macro-average						0.61	0.55	0.56	0.86	0.63	0.72

Table 4: The impact of the incorrect FPs on the precision $\mathbf{P}$, recall $\mathbf{R}$, and F1-score $\mathbf{F1}$. Error samples for a specific predicted dialect (i.e.: FPs of this dialect) that are labeled as valid in this predicted dialect are counted as true positives in the corrected $\mathbf{TP}^*$ score. The corrected $\mathbf{P}^*$, $\mathbf{R}^*$ and $\mathbf{F1}^*$ are based on the corrected value of $\mathbf{TP}^*$. $\mathbf{P}^* = \frac{\mathbf{TP}^*}{\mathbf{TP}^*+\mathbf{FP}^*}$, $\mathbf{R}^* = \frac{\mathbf{TP}^*}{\mathbf{TP}^*+\mathbf{FN}}$, $\mathbf{F1}^* = \frac{\mathbf{2}*\mathbf{P}^**\mathbf{R}^*}{\mathbf{P}^*+\mathbf{R}^*}$
Note: P stands for Precision, R stands for Recall, and $F1$ stands for F1-score.

task as a multi-label classification would optimally alleviate the aforementioned limitations.

5.1 ADI as Multi-label Classification

Multi-label classification allows assigning one or more dialects to the same sample. Bernier-colborne et al. (2023) argued for using the multi-label classification setup after investigating a French DI corpus (FreCDo) (Gaman et al., 2022), covering four macro French dialects spoken in France, Switzerland, Belgium, and Canada. They found that the corpus has duplicated single-labeled sentences of different labels, and showed how these sentences impact the performance of DI models.

Labeling: Collecting multi-labels for a dataset requires the manual annotation of its samples. Dataset creators need to consider how they collect the annotations, and consequently who to recruit. An Arabic speaker of a specific dialect would be able to determine if a sentence is valid in their dialect or not (Salama et al., 2014; Abdelali et al., 2021). Althobaiti (2022) found that the average inter-annotator agreement score (Cohen's Kappa) is 0.64, where two native speakers of 15 different country-level Arabic dialects are asked to check the validity of tweets in their native dialects.

While human participants can sometimes infer the macro-dialect of a sentence that is not in their native dialect, it seems quite hard for them to predict the country-level dialects in which the sentence is valid (Abdul-Mageed et al., 2020b).

Recommendation: Ask Arabic speakers to identify if a sentence is valid in their native dialects or not as per (Salama et al., 2014; Abdelali et al., 2021; Althobaiti, 2022). In order to include new dialects, speakers of these dialects need to be recruited.

Modeling: One way of building multi-label classification models is to use multiple binary classifiers. More specifically, a binary classifier is built to decide whether a sentence is valid in one dialect or not. For N dialects, N binary classifiers would be responsible for predicting the labels of a single sample.

Evaluation: For each supported dialect, evaluation metrics like accuracy, precision, recall, and F1-score can be used. Macro-averaging the metrics is a way to measure the average performance of the model across the different dialects.

Extensibility: The multi-label framing is extensible since more labels can be added to a previously annotated dataset. Adding a new dialect class does not invalidate the labels of the other dialect classes.

This does not apply to the single-label framing since an annotator would need to select a dialect out of a predefined set of dialects. Changing the set of dialects would require the reannotation of the whole dataset.

6 Conclusion

Single-label classification has been the defacto framing for Arabic Dialect Identification (ADI). We show that such framing implies that any model would have a maximal accuracy that is less than 100%, since some samples are valid in multiple dialects, and thus their labels are randomly assigned from these dialects in which they are valid. For a set of 490 validated False Positives (FPs) of an ADI model, we found that the model's predicted dialects for 325 of them are also valid. The fact that about 66% of the FPs are not true errors hinders the ability to analyze and improve the ADI models, and hurts the reliability of the evaluation metrics.

Given this major limitation of single-label framing, we argue that ADI should be framed as a multi-label task. This follows the recommendation of Bernier-colborne et al. (2023) for French Dialect Identification. We hope that this paper will spark discussions across the Arabic NLP community about the current state of ADI, and encourage the creation of new datasets in a multi-label setup, with labels assigned manually by native speakers of the different Arabic dialects.

For future work, we will investigate the impact of the Arabic Level of Dialectness (ALDi) variable introduced by Keleg et al. (2023) on identifying the dialect of sentences. Intuitively, the dialect of a sentence with a high ALDi score is easier to identify since the sentence shows more features of dialectness than those of sentences having low ALDi scores. Therefore ALDi can be used to identify the samples that are more expected to be valid in multiple dialects, facilitating the annotation process of new DI datasets.

Limitations

Recruiting native speakers from the 18 Arab countries included in the NADI 2023 dataset proved to be hard. Moreover, we opted to only annotate the sentences of QADI's test set that were misclassified by the model. In order to accurately estimate the maximal accuracy for a dataset, all the samples should be checked independently by native speakers of the 18 supported Arab countries.

Acknowledgments

This work was supported by the UKRI Centre for Doctoral Training in Natural Language Processing, funded by the UKRI (grant EP/S022481/1) and the University of Edinburgh, School of Informatics.

The error analysis experiment, in which we asked human participants to identify if sentences are valid in their native dialects, was approved by the research ethics committee of the University of Edinburgh School of Informatics with reference number 207712.

References

Ahmed Abdelali, Hamdy Mubarak, Younes Samih, Sabit Hassan, and Kareem Darwish. 2021. QADI: Arabic dialect identification in the wild. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 1–10, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2021a. 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.

Muhammad Abdul-Mageed, AbdelRahim Elmadany, Chiyu Zhang, ElMoatez Billah Nagoudi, Houda Bouamor, and Nizar Habash. 2023. NADI 2023: The Fourth Nuanced Arabic Dialect Identification Shared Task. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*.

Muhammad Abdul-Mageed, Chiyu Zhang, Houda Bouamor, and Nizar Habash. 2020a. NADI 2020: The first nuanced Arabic dialect identification shared task. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 97–110, Barcelona, Spain (Online). Association for Computational Linguistics.

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, Houda Bouamor, and Nizar Habash. 2021b. NADI 2021: The second nuanced Arabic dialect identification shared task. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 244–259, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, Houda Bouamor, and Nizar Habash. 2022. NADI 2022: The third nuanced Arabic dialect identification shared task. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 85–97, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, and Lyle Ungar. 2020b. Toward micro-dialect identification in diaglossic and code-switched environments. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5855–5876, Online. Association for Computational Linguistics.

Ibrahim Abu Farha and Walid Magdy. 2022. The effect of Arabic dialect familiarity on data annotation. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 399–408, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Ibrahim Abu Farha, Wajdi Zaghouani, and Walid Magdy. 2021. Overview of the WANLP 2021 shared task on sarcasm and sentiment detection in Arabic. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 296–305, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Kathrein Abu Kwaik, Motaz Saad, Stergios Chatzikyriakidis, and Simon Dobnik. 2018. Shami: A corpus of Levantine Arabic dialects. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Tareq Al-Moslmi, Mohammed Albared, Adel Al-Shabi, Nazlia Omar, and Salwani Abdullah. 2018. Arabic senti-lexicon: Constructing publicly available language resources for arabic sentiment analysis. *Journal of Information Science*, 44(3):345–362.

Ahmed Y Al-Obaidi and Venus W Samawi. 2016. Opinion mining: Analysis of comments written in arabic colloquial. In *Proceedings of the World Congress on Engineering and Computer Science*, volume 1.

Israa Alsarsour, Esraa Mohamed, Reem Suwaileh, and Tamer Elsayed. 2018. DART: A large dataset of dialectal Arabic tweets. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Maha J. Althobaiti. 2020. Automatic arabic dialect identification systems for written texts: A survey.

Maha J. Althobaiti. 2022. Creation of annotated country-level dialectal arabic resources: An unsupervised approach. *Natural Language Engineering*, 28(5):607–648.

Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2022. Hierarchical aggregation of dialectal data for Arabic dialect identification. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4586–4596, Marseille, France. European Language Resources Association.

Gabriel Bernier-colborne, Cyril Goutte, and Serge Leger. 2023. Dialect and variant identification as a multi-label classification task: A proposal based on near-duplicate analysis. In *Tenth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2023)*, pages 142–151, Dubrovnik, Croatia. Association for 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).

Houda Bouamor, Sabit Hassan, and Nizar Habash. 2019. The MADAR shared task on Arabic fine-grained dialect identification. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 199–207, Florence, Italy. Association for Computational Linguistics.

Anis Charfi, Wajdi Zaghouani, Syed Hassan Mehdi, and Esraa Mohamed. 2019. A fine-grained annotated multi-dialectal Arabic corpus. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 198–204, Varna, Bulgaria. INCOMA Ltd.

Ryan Cotterell and Chris Callison-Burch. 2014. A multi-dialect, multi-genre corpus of informal written Arabic. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 241–245, Reykjavik, Iceland. European Language Resources Association (ELRA).

Mahmoud El-Haj. 2020. Habibi - a multi dialect multi national Arabic song lyrics corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1318–1326, Marseille, France. European Language Resources Association.

Mahmoud El-Haj, Paul Rayson, and Mariam Aboelezz. 2018. Arabic dialect identification in the context of bivalency and code-switching. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Mihaela Gaman, Adrian-Gabriel Chifu, William Domingues, and Radu Tudor Ionescu. 2022. Frecdo: A large corpus for french cross-domain dialect identification.

Salima Harrat, Karima Meftouh, and Kamel Smaïli. 2017. Creating Parallel Arabic Dialect Corpus: Pitfalls to Avoid. In *18th International Conference on Computational Linguistics and Intelligent Text Processing (CICLING)*, Budapest, Hungary.

Trang Ho. 2006-. Tatoeba: Collection of sentences and translations. Available online, Accessed: 10 September 2023.

Fei Huang. 2015. Improved Arabic dialect classification with social media data. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2118–2126, Lisbon, Portugal. Association for Computational Linguistics.

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*, Singapore. Association for Computational Linguistics.

Shervin Malmasi, Marcos Zampieri, Nikola Ljubešić, Preslav Nakov, Ahmed Ali, and Jörg Tiedemann. 2016. Discriminating between similar languages and Arabic dialect identification: A report on the third DSL shared task. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pages 1–14, Osaka, Japan. The COLING 2016 Organizing Committee.

Karen McNeil and Miled Faiza. 2010-. Tunisian arabic corpus (tac): 895,000 words. Available online, Accessed: 10 September 2023.

Karima Meftouh, Salima Harrat, Salma Jamoussi, Mourad Abbas, and Kamel Smaili. 2015. Machine translation experiments on PADIC: A parallel Arabic DIalect corpus. In *Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation*, pages 26–34, Shanghai, China.

Hamdy Mubarak. 2018. Dial2msa: A tweets corpus for converting dialectal arabic to modern standard arabic. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France. European Language Resources Association (ELRA).

Hamdy Mubarak and Kareem Darwish. 2014. Using Twitter to collect a multi-dialectal corpus of Arabic. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 1–7, Doha, Qatar. Association for Computational Linguistics.

Ahmed Salama, Houda Bouamor, Behrang Mohit, and Kemal Oflazer. 2014. YouDACC: the Youtube dialectal Arabic comment corpus. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1246–1251, Reykjavik, Iceland. European Language Resources Association (ELRA).

Wajdi Zaghouani and Anis Charfi. 2018. Arap-tweet: A large multi-dialect Twitter corpus for gender, age and language variety identification. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Omar F. Zaidan and Chris Callison-Burch. 2011. The Arabic online commentary dataset: an annotated dataset of informal Arabic with high dialectal content. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 37–41, Portland, Oregon, USA. Association for Computational Linguistics.

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

Marcos Zampieri, Shervin Malmasi, Nikola Ljubešić, Preslav Nakov, Ahmed Ali, Jörg Tiedemann, Yves Scherrer, and Noëmi Aepli. 2017. Findings of the VarDial evaluation campaign 2017. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 1–15, Valencia, Spain. Association for Computational Linguistics.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Ahmed Ali, Suwon Shon, James Glass, Yves Scherrer, Tanja Samardžić, Nikola Ljubešić, Jörg Tiedemann, Chris van der Lee, Stefan Grondelaers, Nelleke Oostdijk, Dirk Speelman, Antal van den Bosch, Ritesh Kumar, Bornini Lahiri, and Mayank Jain. 2018. Language identification and morphosyntactic tagging: The second VarDial evaluation campaign. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pages 1–17, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Marcos Zampieri, Kai North, Tommi Jauhiainen, Mariano Felice, Neha Kumari, Nishant Nair, and Yash Bangera. 2023. Language variety identification with true labels.

A Detailed Dialect Coverage and Model Performance Report

The datasets used in the paper cover different Arabic dialects as detailed in Table A1. The **PADIC** dataset covers 4 country-level Arabic dialects from North Africa (Algeria, Tunisia), and the Levant (Syria, Palestine). On the other hand, the **QADI**, and **NADI 2023** datasets cover 18 country-level Arabic dialects.

Covering more dialects in a dataset impacts the performance of ADI models. Table A2 provides the detailed performance report of the MarBERT model fine-tuned for ADI between 18 country-level dialects, using NADI 2023's training dataset.

B The Error Analysis Survey

We created an online survey to validate the False Positives (FPs) of the MarBERT model fine-tuned on NADI 2023's training dataset. The survey aims to validate whether the errors of the model are

Dataset	Cities	Countries
PADIC	N = 5 Annaba, Algiers, Sfax, Damascus, Gaza	N = 4 Algeria, Tunisia, Syria, Palestine
MPCA	N/A	N = 5 Egypt, Syria, Jordan, Palestine, Tunisia
MADAR6	N = 5 Beirut, Cairo, Doha, Tunis, Rabat	N = 5 Lebanon, Egypt, Qatar, Tunisia, Morocco
MADAR26	N = 25 Aleppo, Damascus, Algiers, Alexandria, Aswan, Cairo, Amman, Salt, Baghdad, Basra, Mosul, Beirut, Benghazi, Tripoli, Doha, Fes, Rabat, Jeddah, Riyadh, Jerusalem, Khartoum, Muscat, Sanaa, Sfax, Tunis	N = 15 Syria, Algeria, Egypt, Jordan, Iraq, Lebanon, Libya, Qatar, Morocco, Saudi Arabia, Palestine, Sudan, Oman, Yemen, Tunisia
QADI **NADI 2023**	N/A	N = 18 Algeria, Bahrain, Egypt, Iraq, Jordan, Kuwait, Lebanon, Libya, Morocco, Oman, Palestine, Qatar, Saudi Arabic, Sudan, Syria, Tunisia, United Arab Emirates, Yemen

Table A1: The list of labels in the different ADI datasets.

caused by the single-label limitation of the testing dataset or are actual errors. Figure B1 shows screenshots of the Instructions, Sample examples, and the annotation interface.

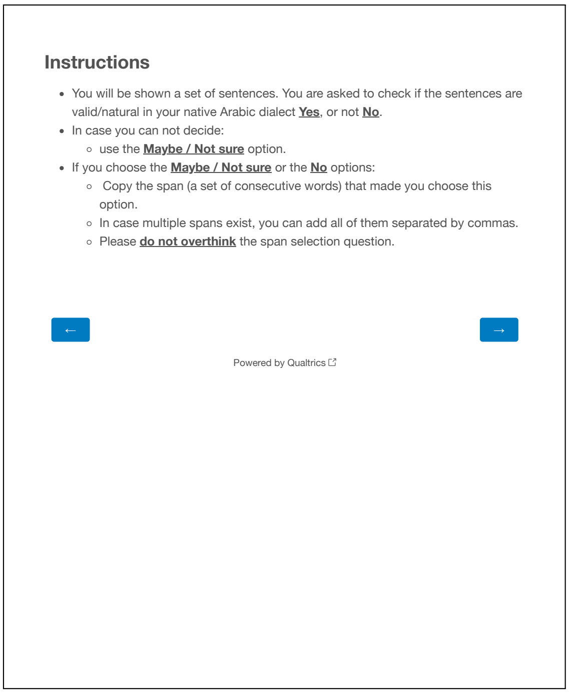
Table B3 lists some examples for samples of the QADI dataset for which the model's predictions do not match the original labels, yet the annotators found these predictions to also be valid.

Dialect	Support	Precision (P)	Recall (R)	F1-score (F1)
Algeria	170	0.63	0.42	0.51
Libya	169	0.45	0.73	0.56
Morocco	178	0.77	0.63	0.70
Tunisia	154	0.63	0.54	0.58
Bahrain	184	0.33	0.29	0.31
Iraq	178	0.69	0.62	0.65
Kuwait	190	0.38	0.43	0.40
Oman	169	0.46	0.51	0.49
Qatar	198	0.37	0.34	0.35
Saudi Arabia	199	0.40	0.44	0.42
UAE	192	0.37	0.53	0.43
Egypt	200	0.65	0.85	0.73
Sudan	188	0.91	0.68	0.78
Jordan	180	0.31	0.47	0.38
Lebanon	194	0.63	0.69	0.66
Palestine	173	0.47	0.43	0.45
Syria	194	0.56	0.31	0.40
Yemen	193	0.55	0.25	0.34
Macro avg.		0.5309	0.5085	0.5072
Weighted avg.		0.5295	0.5074	0.5058
Accuracy		0.5074		

Table A2: The evaluation metrics for the predictions of the fine-tuned MarBERT model on QADI's testing set. The model is fine-tuned on NADI 2023's training data.

Valid Label	Sentence	Original Label
Algeria	عيشک يبارك فيک و يخليک الله يرحمه ربي معک خويا و انا لله و انا اليه راجعون	Tunisia Morocco
Egypt	يلعن الكورة واليوم اللي شجعت في كورة . مرتضي صوتوا ضعيف مع كامل إحترامي مايتقارنش بنسيم مجرد مقارنة	Palestine Tunisia
Lebanon	حالتنا أهون من حالات كتير في الحاضر و في التاريخ . . و غيرنا كتير نجحوا . هههههه مين قلک أعصابي تعبانة	Egypt Syria
Palestine	بما أنو آخر شهر يا ربي يكونو عاملين خصم عالفلافل المشكلة انه فيه ناس ماعندهم عقل عشان تعطيهم على قد عقلهم	Lebanon Kuwait
Saudi Arabia	والله ماعرف عنه بس جتني الصوره على الخاص وقلت اكيد تذكرونه اقرا تغريدتي بالكامل وتقرا تغريدة كساب العتيبي وتعال اسال عنها وراح اجيبک	Iraq Qatar
Sudan	هههههههه انت رجعتي في كلامک سمحتي سمحتي والله يا استاذ عوض دي عربيه	Tunisia Egypt
Syria	هلق الاستعمار فرض علينا بس الاستحمار نحنا فينا نعمله او ما نعمله لاابدا ناس عندهم مبدا	Lebanon Iraq

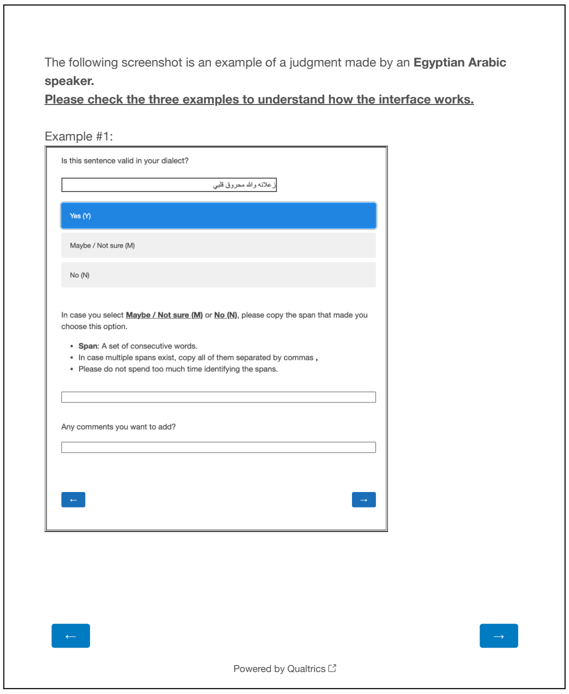
Table B3: Samples of QADI for which the ADI model's predictions are also valid.



Instructions

- You will be shown a set of sentences. You are asked to check if the sentences are valid/natural in your native Arabic dialect **Yes**, or not **No**.
- In case you can not decide:
  - use the **Maybe / Not sure** option.
- If you choose the **Maybe / Not sure** or the **No** options:
  - Copy the span (a set of consecutive words) that made you choose this option.
  - In case multiple spans exist, you can add all of them separated by commas.
  - Please **do not overthink** the span selection question.

(a) Instructions page.



The following screenshot is an example of a judgment made by an **Egyptian Arabic speaker.**
Please check the three examples to understand how the interface works.

Example #1:

Is this sentence valid in your dialect?

زعلانه والله محروق قلبي

[✓ Yes (Y)]

Maybe / Not sure (M)

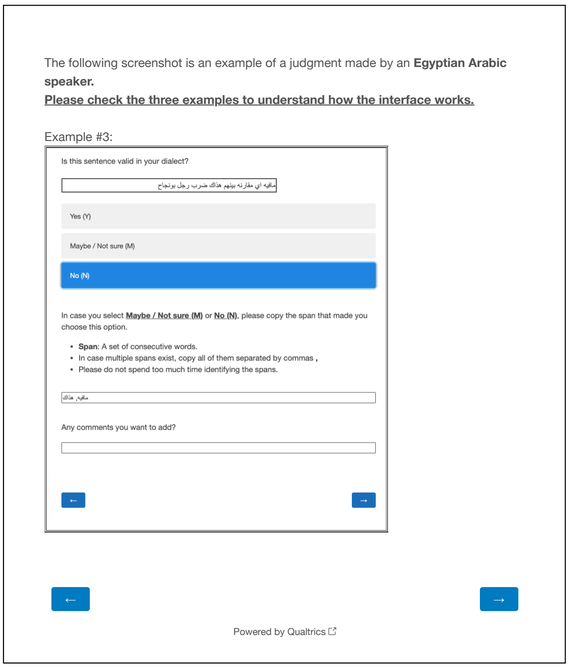
No (N)

In case you select **Maybe / Not sure (M)** or **No (N)**, please copy the span that made you choose this option.

- **Span**: A set of consecutive words.
- In case multiple spans exist, copy all of them separated by commas **,**
- Please do not spend too much time identifying the spans.

Any comments you want to add?

(b) First example page.



The following screenshot is an example of a judgment made by an **Egyptian Arabic speaker.**
Please check the three examples to understand how the interface works.

Example #3:

Is this sentence valid in your dialect?

مافيه اي مقارنه بينهم هذاك ضرب رجل بونجاح

Yes (Y)

Maybe / Not sure (M)

[✓ No (N)]

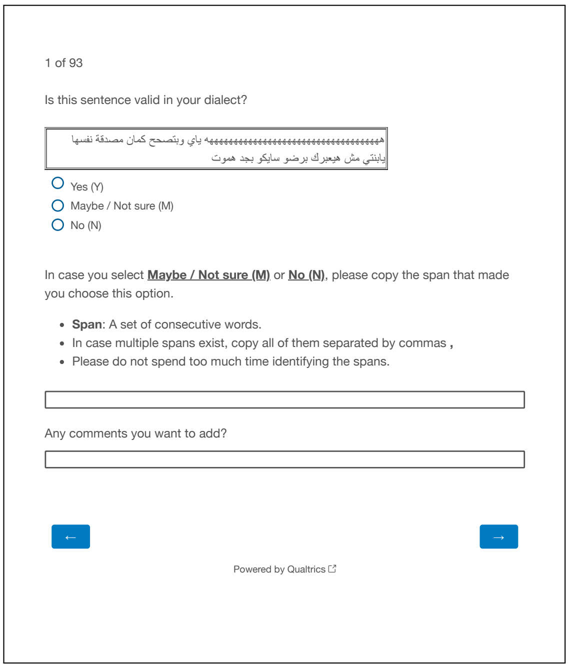
In case you select **Maybe / Not sure (M)** or **No (N)**, please copy the span that made you choose this option.

- **Span**: A set of consecutive words.
- In case multiple spans exist, copy all of them separated by commas **,**
- Please do not spend too much time identifying the spans.

مافيه, هذاك

Any comments you want to add?

(c) Third example page.



1 of 93

Is this sentence valid in your dialect?

هههههههههههههههههههههههههههههههههههههه ياي وبتصحح كمان مصدقة نفسها
يابنتي مش هيعبرك برضو سايكو بجد هموت

○ Yes (Y)
○ Maybe / Not sure (M)
○ No (N)

In case you select **Maybe / Not sure (M)** or **No (N)**, please copy the span that made you choose this option.

- **Span**: A set of consecutive words.
- In case multiple spans exist, copy all of them separated by commas **,**
- Please do not spend too much time identifying the spans.

Any comments you want to add?

(d) An annotation page.

Figure B1: Screenshots of the different pages of the annotation task described in §4.2.

Arabic Topic Classification in the Generative and AutoML Era

Doha Albared and **Hadi Hamoud** and **Fadi A. Zaraket**
Arab Center for Research and Policy Studies, Doha
{dal007,hhamoud,fzaraket} @dohainstitute.edu.qa

Abstract

Most recent models for Arabic topic classification leveraged fine-tuning existing pre-trained transformer models and targeted a limited number of categories. More recently, advances in automated ML and generative models introduced novel potentials for the task. While these approaches work for English, it is a question of whether they perform well for low-resourced languages; Arabic in particular. This paper presents (i) ArBNTopic; a novel Arabic dataset with an extended 14-topic class set covering modern books from social sciences and humanities along with newspaper articles, and (ii) a set of topic classifiers built from it. We fine-tuned an open LLM model to build ArGTC. We compared its performance against the best models built with Vertex AI (Google), AutoML(H2O), and AutoTrain(HuggingFace). ArGTC outperformed the VertexAi and AutoML models, and was reasonably similar to the AutoTrain model.

1 Introduction

Text classification models have been a topic of interest in the natural language processing (NLP) research community, due to their importance in performing multiple tasks such as sentiment analysis (Sohangir et al., 2018; Qian et al., 2018; Ain et al., 2017), topic classification (Johnson and Zhang, 2017; Razno, 2019), spam detection (Trivedi, 2016; Ismail et al., 2022; Fattahi and Mejri, 2021), and fake news detection (Meesad, 2021; Hamid et al., 2020; Kong et al., 2020). In earlier stages, text classification primarily relied on traditional machine-learning algorithms like Support Vector Machines (SVM), Naive Bayes, and Decision Trees. Deep Learning models have been used to perform numerous NLP tasks and attained remarkable results (Sohangir et al., 2018; Lai et al., 2015). Nonetheless, these models are built from the ground up, demanding large datasets and several days of training.

Recently, transfer learning by fine-tuning a pre-trained large language model (LLM) helped resolve the problem. An LLM, trained with large corpora for generic tasks, gains further specific capacities in the process. This effectively produced high-performing classification models across several languages (Adhikari et al., 2019; Balkus and Yan, 2022b; Bataa and Wu, 2019; Polignano et al., 2019).

Recently, state-of-the-art models tailored for Arabic NLP tasks leveraged transfer learning to perform tasks such as text generation, classification, translation, sentiment analysis, summarization, title generation, and dialect identification. AraBERT, one of the most prominent models (Antoun et al., 2020a), was fine-tuned from BERT (Devlin et al., 2018) to specifically serve the Arabic language. More models for Arabic emerged to perform text generation and classification (Nagoudi et al., 2021; Khondaker et al., 2023; Khered et al., 2022), and topic classification: identifying the topic(s) discussed in a specific text (Abdul-Mageed et al., 2020; Chowdhury et al., 2020).

Generative Models: More recently, several generative transformer-based models have been trained on multi-lingual corpora including Arabic. We considered two fine-tuned variants: BLOOMZ and mT0 (Muennighoff et al., 2022b). Bloomz-7b-mt (Muennighoff et al., 2022b) is a 7-billion parameter model pre-trained to respond to instructions (z) in further languages leveraging the xPmt multilingual (mt) dataset (Muennighoff et al., 2022a) with significant Arabic content .

AutoML: Meanwhile, several automated machine learning services emerged such as Google Vertex Ai (Google-Vertex-AI, 2023),

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 399–404
December 7, 2023 ©2023 Association for Computational Linguistics

H2O AutoML (LeDell and Poirier, 2020), and Huggingface AutoTrain (Wolf et al., 2020). Such services perform tasks including data preprocessing, feature engineering, model selection, hyperparameter tuning, and model deployment saving time, effort, and resources. These models perform typically well in text classification for high-resource languages.

ArBNTopic & ArGTC: In this paper, we explore Arabic text classification, with an extended set of topics, across the generative and automated machine learning approaches. For that we built ArBNTopic, a dataset from specialized books and newspaper articles to introduce novel topics to Arabic topic classification models including topics from sciences, social sciences, and humanities. This dataset is openly available on HuggingFace [1]

We used ArBNTopic to build ArGTC[2] in two steps. The first step boosted the bloomz-7b-mt model with additional Arabic content from domains it did not cover before. In the second step, we fine-tuned the resulting model [3] from step 1, with a part of ArBNTopic with classes. We chose the bloomz-7b-mt after a careful review as its predecessors had Arabic capacities (BLOOM), had instruction (Z) fine-tuning capacities, and had additional multilingual (mt) capacities from additional diverse datasets.

ArGTC performs with an accuracy, precision, and recall of 83, 81, and 81%, respectively. It shows better results than the best models we generated with ArBNTopic using Google Vertex Ai (Google-Vertex-AI, 2023) and H2O AutoML (LeDell and Poirier, 2020). It also compares closely to the performance of the best model generated using AutoTrain from Huggingface (Wolf et al., 2020). ArGTC is reasonably better than the models generated using the automated machine learning services when considering a cost-effective performance balance.

2 Related Work

The use of transformer-based models is quickly covering all NLP tasks. It started with BERT-architecture models (Li et al., 2022). Transformer models take advantage of their abilities to represent contextual relationships between concepts through contextual embeddings. Arabic text categorization research also emerged recently (Alammary, 2022). Several Arabic NLP tasks followed after the inception of multilingual BERT. However, studies reported better results using monolingual models, specifically for low to mid-resource languages (Wu and Dredze, 2020). AraBERT, trained on Arabic Wikipedia and newspaper, was applied to several downstream tasks (Antoun et al., 2020b). The model performed well on multi-class tasks (zahra El-Alami et al., 2022).

Training on a mixture of Dialect-Arabic through social media datasets, and modern standard Arabic (MSA) data, has resulted in better encoder models, like Qarib (Abdelali et al., 2021). Text categorization attempts on iterations of Qarib have proven successful, through fine-tuning on classified MSA and Dialect datasets, with 6 to 12 labeling classes used (Chowdhury et al., 2020).

Apart from BERT models, considerable progress has been made using GPT and T5-based models. For instance, AraT5, a fine-tuned version of multilingual T5 on the Arabic Language (Nagoudi et al., 2022), is achieving close to state-of-the-art performances on a variety of tasks, including categorization (Khondaker et al., 2023).

GPT-3.5/4 base models augmented with Arabic data, are also performing exceptionally well on text and sequence classification(Abdelali et al., 2023; Balkus and Yan, 2022a). However, due to OpenAi's business model, fine-tuned versions of GPT are only available for use through the paid API.

Also important to mention that with the release of the BLOOM family, including the BLOOMZ model, which is our choice of foundation, BigScience has also deployed multiple inference heads on top, with specific configurations, for different tasks. This includes sequence classification, question answering, and generation.

3 Fine-tuning Data and Model

We selected the 7-Billion parameter, publicly available, bloomz-7b-mt (Muennighoff et al., 2022b) model as our base model. It belongs to the BLOOMZ and mT0 family resulting from fine-tuning BLOOM on the multilingual

[1] https://huggingface.co/datasets/dru-ac/ArBNTopic

[2] https://huggingface.co/dru-ac/ArGTC

[3] https://huggingface.co/dru-ac/FTArBloom

xP3mt dataset (Muennighoff et al., 2022a). We fine-tuned it in two phases: (i) for Arabic text generation, and then (ii) for Arabic text classification.

To realize this objective we fine-tuned bloomz-7b-mt on an additional $58,682$ tokens taken from books and newspapers datasets written in modern standard Arabic. The resultant model encompasses a comprehensive spectrum of fourteen distinct subjects that are Religion, Finance and Economics, Politics, Medical, Culture, Sports, Science and Technology, Anthropology and Sociology, Art and Literature, Education, History, Language and Linguistics, Law, as well as Philosophy.

The generated model was further fine-tuned on a bigger and labeled dataset that comprises 833,642 tokens. We call the resulting model ArGTC. ArGTC is designed to categorize input text, determining its alignment with one of the 14 predefined topics.

3.1 Data and Preprocessing

We developed ArBNTopic to fine-tune and train the generative and AutoML-based models. We used newspaper articles and a set of published books to build ArBNTopic. The newspaper articles come from the SANAD dataset (Einea et al., 2019). SANAD is publicly available and includes an extensive assortment of Arabic news articles suitable for various Arabic NLP tasks. These articles were gathered from three well-known Arabic news portals: AlKhaleej, AlArabiya, and Akhbarona. Each newspaper dataset is labeled with seven categories: Culture, Finance, Medical, Politics, Religion, Sports, and Tech, except for AlArabiya, which lacks the Religion category. We split the articles down into the paragraph level with a maximum character limit of 250 per segment. When a paragraph contained more than that, we split it into more than one segment and included all segments.

The dataset of books was acquired in Word format provided by the Arab Center for Research and Policy Studies. The books spanned the areas of Religion, Economy, Politics, Anthropology and Sociology, Art and Literature, Education, History, Language and Linguistics, Philosophy, and Law. Table 1 shows the number of books across each category. We selected texts from the books to build a balanced dataset across categories.

Table 1: Number of books by category

Category	Number of Books
Religion	8
Economy	28
Language & Linguistics	25
Anthropology & Sociology	101
Art & Literature	7
Education	6
Philosophy	84
Law	6
Politics	174
History	79

Newspaper articles provide concise and short texts with ideas and sentences that tend to be more compressed and more straightforward than texts in books. Texts in books tend to be quite the opposite; longer texts that feel free to tackle topics from broader and different angles. Hence, a book classified in one category can easily overlap in some of its parts or chapters with other categories. That is a book in history can discuss religion, politics, and education. A book in philosophy can discuss religion and society. When a sentence from a book is taken out of its context, it can be categorized into a topic other than the topic of its book.

At times, sentences from books may become too general to be categorized with any topic at all if read out of context. We resolved these issues by confining our data to selecting the first sentence(s) after each title, subtitle, and heading in each of the books. The first sentences after the titles tend to contain the thesis statements and topic sentences which introduces and summarizes the discussions in next paragraphs to follow.

We still had to solve the disparity in the number of books under each category. The first sentence after a title rule yielded a wide variation gap in the number of documents for each category. For example, while around 200 sentences were labeled religion, more than 4000 were labeled politics. To overcome this issue, for the categories that had less than 10 books (religion, law, art and literature, education), we extracted the first sentences of each long paragraph.

We unified the labels for categories from newspapers and books as shown in Table 2 to obtain the aggregated list of categories. All the data extracted from books is openly available on Hugginface. The sample mixed dataset (books and newspapers) used to complete the finetuning is available as available as well.

Table 2: Categories mapping

Newspapers	Books	Final
Religion	Religion	Religion
Finance	Economy	Finance-and-Economy
Politics	Politics	Politics
Medical		Medical
Culture		Culture
Sports		Sports
Tech		Science-and-Technology
	Anthropology & Sociology	Anthropology-and-Sociology
	Education	Education
	Philosophy	Philosophy
	Language and Linguistics	Language-and-Linguistics
	History	History
	Law	Law

3.2 Compute Setup

We fine-tuned both models on one A100-80 GiB GPU that we rented from a provider on the cloud. The GPU had 80 GiB VRAM, 125 GiB RAM, 14 vCPUs, 256 GiB of persistent volume disk storage, and 256 GiB of container non-persistent disk storage. Compute power and storage costs of this configuration amount to $1.86/hour. The model can be loaded for inference on a system with a GPU of 32 GiB VRAM at a cost of $0.3 per hour.

We loaded and fine-tuned the 7-Billion parameter model with limited virtual memory due with the help of gradient checkpointing [4] [5]. This comes at the expense of slowing down the fine-tuning process.

As for software specifications, we used a standard deep-learning container image, with Python-3.10, PyTorch-2.0.1, Cuda-11.8.0, and Transformers-4.32.1.

4 Results

Following the fine-tuning and upon testing the ArGTC on ArBNTopic labeled with the 14 categories, it scored 83% accuracy, 81% precision, and recall (Table 3). In addition, we trained 3 more classification models using automated ML services from Google Vertex Ai, Hugginface AutoTrain, and H2O AutoML. In fact, Multiple H2O models were trained using embeddings as features from different language models trained on Arabic data, namely, AraBERT (Antoun et al., 2020b), and AraGPT2 (Antoun et al., 2021). Using AraGPT2, embeddings yielded a model with higher scores, 76% accuracy and precision, and 74% recall, then the models were trained using AraBERT embeddings, which scored 67% for accuracy, precision, and recall.

While the best H2O model scored lower than ArGTC on all three measurements, the model produced using AutoTrain scored 84% for accuracy, precision, and recall. In addition, the fourth classification model we trained using Google Vertex AI scored 77% precision, 76% recall, and 76% accuracy. Hence, the AutoTrain model scored the highest on all measurements, accuracy, precision, and recall, followed closely by ArGTC, then the H2O model followed by the model obtained through Google Vertex AI training.

It is worth noting that while H2O is charge-free, AutoTrain is free only for datasets up to 3,000 samples. For which contained 19,784 samples, extra charges amount to $46. On the other hand, using Vertex Ai with the same dataset costs $22. Fine-tuning time for ArGTC was 4-5 hours, which amounts to a cost of around $9.3. Consequently, the ArGTC model optimally balances performance against costs.

Table 3: Results of training ArGClass, Google Vertex Ai (Vertex), HuggingFace AutoTrain (HF), H2O AutoML (H2O)

	ArGTC	Vertex	H2O	HF
Accuracy %	83	76	76	84
Precision %	81	77	76	84
Recall %	81	76	74	84

5 Conclusion

This paper introduced ArGTC, an Arabic text classification model with 14 categories. Utilizing transfer learning techniques, the model was fine-tuned in two stages from bloomz-7b-mt, achieving 83% accuracy, 81% precision, and recall and surpassing the best models trained using VertexAi and AutoML. It performed comparably to the best model we trained with AutoTrain with a small margin.

[4] https://huggingface.co/docs/transformers/v4.18.0/en/performance#gradient-checkpointing

[5] https://medium.com/tensorflow/fitting-larger-networks-into-memory-583e3c758ff9

Limitations

ArGTC performance is bound to the quality of ArBNTopic and the foundation models. ArBNTopic contains a specific collection of books. We also confined the models to a fixed list of predefined topics based on the specialties of the books and newspaper articles. To capture topics beyond this predefined set, alternative unsupervised topic modeling techniques would be necessary.

Ethics Statement

The data was collected and used with the appropriate approvals of the intellectual property owners. All results are reported following best academic standards and practices.

References

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

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. 2023. Benchmarking arabic ai with large language models.

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

Ashutosh Adhikari, Achyudh Ram, Raphael Tang, and Jimmy Lin. 2019. Docbert: Bert for document classification. *arXiv preprint arXiv:1904.08398*.

Qurat Tul Ain, Mubashir Ali, Amna Riaz, Amna Noureen, Muhammad Kamran, Babar Hayat, and A Rehman. 2017. Sentiment analysis using deep learning techniques: a review. *International Journal of Advanced Computer Science and Applications*, 8(6).

Ali Saleh Alammary. 2022. Bert models for arabic text classification: A systematic review. *Applied Sciences*, 12(11).

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

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020b. AraBERT: Transformer-based model for Arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France. European Language Resource Association.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2021. AraGPT2: Pre-trained transformer for Arabic language generation. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 196–207, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Salvador Balkus and Donghui Yan. 2022a. Improving short text classification with augmented data using gpt-3.

Salvador V Balkus and Donghui Yan. 2022b. Improving short text classification with augmented data using gpt-3. *Natural Language Engineering*, pages 1–30.

Enkhbold Bataa and Joshua Wu. 2019. An investigation of transfer learning-based sentiment analysis in japanese. *arXiv preprint arXiv:1905.09642*.

Shammur Absar Chowdhury, Ahmed Abdelali, Kareem Darwish, Jung Soon-Gyo, Joni Salminen, and Bernard J Jansen. 2020. Improving arabic text categorization using transformer training diversification. In *Proceedings of the fifth arabic natural language processing workshop*, pages 226–236.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Omar Einea, Ashraf Elnagar, and Ridhwan Al Debsi. 2019. Sanad: Single-label arabic news articles dataset for automatic text categorization. *Data in brief*, 25:104076.

Jaouhar Fattahi and Mohamed Mejri. 2021. Spaml: a bimodal ensemble learning spam detector based on nlp techniques. In *2021 IEEE 5th international conference on cryptography, security and privacy (CSP)*, pages 107–112. IEEE.

Google-Vertex-AI. 2023. Google vertex ai.

Abdullah Hamid, Nasrullah Shiekh, Naina Said, Kashif Ahmad, Asma Gul, Laiq Hassan, and Ala Al-Fuqaha. 2020. Fake news detection in social media using graph neural networks and nlp techniques: A covid-19 use-case. *arXiv preprint arXiv:2012.07517*.

Safaa SI Ismail, Romany F Mansour, Abd El-Aziz, M Rasha, Ahmed I Taloba, et al. 2022. Efficient e-mail spam detection strategy using genetic decision tree processing with nlp features. *Computational Intelligence and Neuroscience*, 2022.

Rie Johnson and Tong Zhang. 2017. Deep pyramid convolutional neural networks for text categorization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 562–570.

Abdullah Salem Khered, Ingy Yasser Hassan Abdou Abdelhalim, and Riza Theresa Batista-Navarro. 2022. Building an ensemble of transformer models for arabic dialect classification and sentiment analysis. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 479–484.

Md Tawkat Islam Khondaker, Abdul Waheed, El Moatez Billah Nagoudi, and Muhammad Abdul-Mageed. 2023. Gptaraeval: A comprehensive evaluation of chatgpt on arabic nlp.

Sheng How Kong, Li Mei Tan, Keng Hoon Gan, and Nur Hana Samsudin. 2020. Fake news detection using deep learning. In *2020 IEEE 10th symposium on computer applications & industrial electronics (ISCAIE)*, pages 102–107. IEEE.

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29.

Erin LeDell and Sebastien Poirier. 2020. H2O AutoML: Scalable automatic machine learning. *7th ICML Workshop on Automated Machine Learning (AutoML)*.

Qian Li, Hao Peng, Jianxin Li, Congying Xia, Renyu Yang, Lichao Sun, Philip S. Yu, and Lifang He. 2022. A survey on text classification: From traditional to deep learning. *ACM Trans. Intell. Syst. Technol.*, 13(2).

Phayung Meesad. 2021. Thai fake news detection based on information retrieval, natural language processing and machine learning. *SN Computer Science*, 2(6):425.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2022a. Crosslingual generalization through multitask finetuning.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, et al. 2022b. Crosslingual generalization through multitask finetuning. *arXiv preprint arXiv:2211.01786*.

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*.

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.

Marco Polignano, Pierpaolo Basile, Marco De Gemmis, Giovanni Semeraro, Valerio Basile, et al. 2019. Alberto: Italian bert language understanding model for nlp challenging tasks based on tweets. In *CEUR Workshop Proceedings*, volume 2481, pages 1–6. CEUR.

Jun Qian, Zhendong Niu, and Chongyang Shi. 2018. Sentiment analysis model on weather related tweets with deep neural network. In *Proceedings of the 2018 10th international conference on machine learning and computing*, pages 31–35.

Maria Razno. 2019. Machine learning text classification model with nlp approach. *Computational Linguistics and Intelligent Systems*, 2:71–73.

Sahar Sohangir, Dingding Wang, Anna Pomeranets, and Taghi M Khoshgoftaar. 2018. Big data: Deep learning for financial sentiment analysis. *Journal of Big Data*, 5(1):1–25.

Shrawan Kumar Trivedi. 2016. A study of machine learning classifiers for spam detection. In *2016 4th international symposium on computational and business intelligence (ISCBI)*, pages 176–180. IEEE.

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, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing.

Shijie Wu and Mark Dredze. 2020. Are all languages created equal in multilingual bert?

Fatima zahra El-Alami, Said Ouatik El Alaoui, and Noureddine En Nahnahi. 2022. Contextual semantic embeddings based on fine-tuned arabert model for arabic text multi-class categorization. *Journal of King Saud University - Computer and Information Sciences*, 34(10, Part A):8422–8428.

On Enhancing Fine-Tuning for Pre-trained Language Models

Betka Abir,[1] **Zeyd Ferhat,**[4] **Barka Riyadh,**[3] **Boutiba Selma,**[2] **Zineddine S. Kahhoul,**[2]
Tiar M. Lakhdar,[2] **Ahmed Abdelali,**[5] **and Habiba Dahmani**[3]
[1] Laboratory of VCS, [2] Laboratory of IL3CUB, University of Biskra, Algeria
[3] Electrical Engineering Department, [4] Department of Electronics, University of M'sila, Algeria
[5] Qatar Computing Research Institute, HBKU, Qatar
{betkaabir, zeydferhatz, barkariyadh06}@gmail.com,
{selma.boutiba, zineddine.kahhoul, mohamedlakhdar.tiar}@univ-biskra.dz
aabdelali@hbku.edu.qa, habiba.dahmani@univ-msila.dz

Abstract

The remarkable capabilities of Natural Language Models to grasp language subtleties has paved the way for their widespread adoption in diverse fields. However, adapting them for specific tasks requires the time-consuming process of fine-tuning, which consumes significant computational power and energy. Therefore, optimizing the fine-tuning time is advantageous. In this study, we propose an alternate approach that limits parameter manipulation to select layers. Our exploration led to identifying layers that offer the best trade-off between time optimization and performance preservation. We further validated this approach on multiple downstream tasks, and the results demonstrated its potential to reduce fine-tuning time by up to 50% while maintaining performance within a negligible deviation of less than 5%. This research showcases a promising technique for significantly improving fine-tuning efficiency without compromising task- or domain-specific learning capabilities.

1 Introduction

Neural based Language Models are functions or algorithms that are trained to predict the likelihood of a sequence of words (Devlin et al., 2019; Radford et al., 2019). These models were trained using large volumes of textual content and are able to provide an accurate approximation for language features and structure. These models provide an important tool for analyzing and understanding the nuance of language, as well as for building applications that rely on natural language understanding (Qiu et al., 2020). Fine-tuning neural language models refers to the process of further training a pre-trained language model on a specific task or domain with a smaller dataset. The pre-trained language model, such as BERT or GPT, has already learned a significant amount of knowledge about natural languages from a large corpus of text. However, it may not have been trained specifically for the task at hand or on the specific domain of interest. Fine-tuning involves updating the pre-trained model's parameters to optimize its performance on the given target so it can learn more task-specific or domain-specific information. Fine-tuning large language models (LLMs) proved to be very effective and efficient to achieve higher accuracy and state of the art numbers in many downstream tasks(Xiao et al., 2020). Various techniques were suggested to ensure that the resulting models achieve optimal accuracy. One of the challenges faced during the fine-tuning of language models is overfitting. Overfitting occurs when the model performs well on the training or fine-tuning data but poorly on new, unseen data. This happens because the model has learned to fit the noise in the training data rather than capturing the underlying patterns. To address overfitting, several regularization techniques were proposed in the literature, such as weight decay and dropout. These methods help prevent the model from memorizing the training data and promote better generalization to unseen data. Additionally, achieving optimal results with fine-tuning involves hyperparameter tuning, where efforts are made to select the best set of hyperparameters for the model. Hyperparameters, such as the learning rate and number of layers, can significantly influence the model's performance and generalization capabilities. Properly tuning these hyperparameters is essential for obtaining the best possible results during fine-tuning (Mosbach et al., 2021; Yang and Ma, 2022). In this research, we pursue a different direction for fine-tuning language models by exploring a methodology that involves limiting backpropagation to a specific number of layers. This approach offers several benefits, including effectively addressing the issue of over-fitting and significantly reducing the fine-tuning time. Our primary objective is to identify the most impactful layers that contribute to achieving the best performance, and then extend this investigation to various pre-trained mod-

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 405–410
December 7, 2023 ©2023 Association for Computational Linguistics

els. The key contributions of this research are as follows:

- We explore **the impact of layer freezing** on pre-trained models with focus on application on tasks in Arabic language.
- Evaluate the effect of layer freezing on different pre-trained models **in terms of performance and speed**.
- Compare **the performance** of models using the proposed approach.
- Contrast **the time needed for fine-tuning** in both layer freezing and no-freezing settings.

The remainder of the article is structured as follows: In the next section, we provide background information on the evolution of language models and natural language processing. Subsequently, in the third section, we present our methodology, introducing the language models and tasks we will be experimenting with. Following that, we present the results and engage in a discussion in the fourth section. Finally, in the fifth section, we present our conclusions and outline the prospects for our ongoing work.

2 Background

2.1 Natural Language Processing

Natural Language Processing (NLP) is an essential branch of artificial intelligence that delves into the intricate realm of human language. Its primary objective is to empower computers with the ability to comprehend, interpret, and manipulate text and words in a manner that mirrors human understanding (Liddy, 2001). The definition of NLP covers a variety of aspects: There are several computational methods for NLP, and they essentially fall into four categories; symbolic, statistical, connectionist, and hybrid. Symbolic methods use a deep analysis of linguistic phenomena,and they are based on the explicit representation of linguistic facts using well-known knowledge representation schemes. Statistical approaches build models of linguistic phenomena using a variety of mathematical techniques and a large text corpus. The major source of evidence for these methods is observable data, with no linguistic or general knowledge added. The connectionist approach construct generalised models using examples of linguistic phenomena, and they employ also variety of representational theories. The text being analysed must come from a language that people use to communicate, and it may be in any language,and in any format oral or written.

In NLP, humans utilize various levels of language to comprehend the content of a document. These levels include Phonology (the study of speech sounds), Morphology (the study of word forms and structure), Lexical (the study of words and their meanings), Syntactic (the study of sentence structure), Semantic (the study of meaning in language), Discourse (the study of how sentences are connected and organized), and Pragmatic (the study of language use in context). The more capable an NLP system is, the more of these levels it will employ to understand and process language effectively. For instance, a sophisticated NLP system will take into account not only the words in a sentence but also their meanings, how they are arranged grammatically, and how the sentences relate to each other in a larger context. However, in practice, current NLP systems often utilize separate modules to handle different levels of language processing. These modules work together to process the language and extract meaningful information.

2.2 Techniques

Among the ground breaking techniques that changed the field of NLP was the introduction of Transformers (Vaswani et al., 2017). Its power to handle sequential data made them dominate the field in recent year. BERT (Bidirectional Encoder Representations from Transformers) is a revolutionary language model that has had a profound impact on Natural Language Processing (NLP). It is designed to understand the context of words in a sentence by considering the surrounding words on both sides, leading to a bidirectional learning process. This innovative approach allows BERT to capture deep contextual relationships and nuances in language, making it exceptionally effective in various NLP tasks. By pre-training on a large corpus of text and then fine-tuning on specific downstream tasks, BERT exhibits remarkable versatility and can be adapted to tasks like text classification, named entity recognition, question answering, and more. Its contextual embeddings have significantly improved the accuracy of language-based applications, and BERT's success has inspired numerous follow-up models that continue to push the boundaries of NLP research and application.

2.3 Freezing

Fine-tuning has become an integral component in the training process, because is less expensive in computational time than pre-training a mode. Additionally, it could solve the problem of overfitting. Limiting the number of layers "freezing" is a natural way to improve fine-tuning performance (Liu et al., 2021). For BERT model, the initial layers learn more general linguistic patterns. However, the later BERT layers learn more task-specific patterns (Clark et al., 2019; Sajjad et al., 2023).

3 Methodology

To explore the extent of the proposed method, we limit the scope of our investigation to the following pre-trained models and tasks, more models would be worth of investigating in the future work.

3.1 Pre-trained models

AraBERTv0.2 Antoun et al. (2020) trained a BERT base model using 200M sentences (77GB) of both Modern Standard Arabic (MSA) and dialectal content mainly from Twitter data. The MSA content includes Arabic Wikipedia Dumps, Arabic Corpus (El-Khair, 2016) and the Open Source International Arabic News Corpus (OSIAN) (Zeroual et al., 2019), in addition to Arabic news content.

CAMelBERT Inoue et al. (2021) created and distributed a pre-trained language model that combined Modern Standard Arabic (MSA), dialectal Arabic (DA), and classical Arabic (CA). The collection included over 167GB of text (17.3B tokens).

QARiB Abdelali et al. (2021) trained a model on a collection of 420 Million tweets and 180 Million sentences of text. The tweets contains both MSA and DA, while the text content is mostly MSA.

MARBERT Abdul-Mageed et al. (2021) created and distributed large-scale pre-trained masked language model focused on both Dialectal Arabic (DA) and Modern Standard Arabic (MSA). It was trained on a dataset of 1 billion Arabic tweets from an in-house dataset of about 6 billion tweets.

3.2 Tasks

Arabic Language Understanding Evaluation (ALUE) (Seelawi et al., 2021) provides a total of eight tasks that address a variety of Arabic dialects and NLP/NLU issues. In this paper, four tasks are used for experimental results.

Anger Detection The Affect in Tweets dataset proposed in (Mohammad et al., 2018) consists of five subtasks. We will only use the Emotion Classification task (SEC), in which a tweet is classified as anger, anticipation, contempt, fear, joy, love, optimism, pessimism, sad, surprise, and trust. We concentrate on the anger emotion, we detect if a tweet contains that emotion or not.

Text Similarity In the Semantic Question Similarity task (McCann et al., 2017), two questions are considered to be semantically similar if they have the exact same response and significance. The dataset includes question pairings and the degrees of similarity between them. There are two questions in each question pair. Each question pair's similarity score is shown as a value between 0 and 5, which was determined by human evaluations.

IDAT@FIRE2019 Irony Detection Task (FID) The purpose of this task is to detect irony in Arabic tweets (Ghanem et al., 2019). Each tweet is labeled with a "1" when it contains irony or sarcasm. Otherwise, a label of "0" is assigned.

MADAR Shared Task Subtask 1 (Dialect Detection) The Multi Arabic Dialect Applications and Resources (MADAR)[1]. The first MADAR's subtask was a parallel corpus of 25 Arabic city dialects in the field of travel (Bouamor et al.). The MSA is given a 26th label. We focus only on two classes; the dialects of Algiers and Amman.

OSACT4 Shared Task-A: offensive The task (Mubarak et al., 2020) was designed for the purpose of detecting offensive speech in Arabic tweets. Each tweet is labeled with a ”1” when it contains offensive speech. Otherwise, ”0”.

OSACT4 Shared Task-B: hate speech detection The purpose of this task is to detect hate speech in Arabic tweets (Mubarak et al., 2020). Each tweet is labeled with a ”0” when it contains hate speech. Otherwise, a label of ”1” is assigned.

Cross-lingual Sentence Representations The goal of this task is to use a dataset containing 7,500 pairs of sentences to classify them into one of the following categories: “commitment,” “ambivalence,” or “neutral.” (Conneau et al., 2018)

4 Results and Discusson

4.1 Optimal Settings

We investigate the optimal parameters for layer freezing. To identify the best configuration, we perform a comprehensive grid search, exploring all possible combinations. Although this approach

[1] https://sites.google.com/nyu.edu/madar/

may seem exhaustive, it allows us to evaluate all layers efficiently. For this step, we use the MADAR dataset, chosen as an exemplary task due to its large size and multitude of labels. Specifically, this is a multi-class classification with 26 class labels, each representing the dialect associated with different city. We explore a combination of freezing both n top and m bottom layers while recording the performance at each combination. Figure 1 represents the results of the exploration.The evidence shows that unfreezing all layers leads to achieving the state-of-the-art (SOTA) performance. However, even by freezing up to 3 layers from the bottom and four layers from the top, the model still attains performance levels very close to the best performance. Figure 1 shows the F1 results of freezing all combinations on MADAR task.

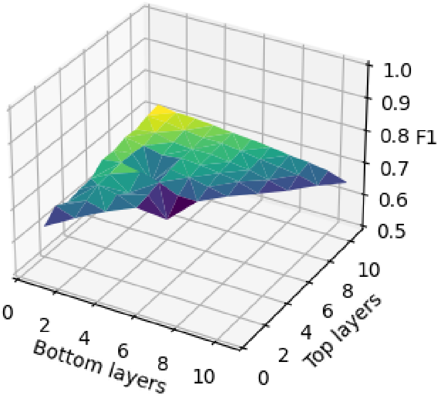


Figure 1: Layers freezing results on MADAR.

4.2 Layer Freezing

Given the promising results obtained from the previous experiments. We further expand our experimentation to benchmark an actual four downstream tasks. Appendix Tables 1, 2, 3 4, 5, 6 and 7 show the performance of training and evaluation of BERT models on different tasks, in terms of F1 and training time. While the performance loss in all the seven tasks rarely surpassed 6%, the gain in time reached up to 50%. In few instances, the performance improved further see MARBERT models results in table 6 and 7. The results summarized in Figure 2 shows clearly the large difference between the gain in runtime versus the performance loss.

4.3 Discussion

This research focuses on optimizing the computation time required for fine-tuning large language models, considering the substantial impact of computation costs across various applications and disciplines. To achieve this objective, we introduced the "layers freezing" approach, which effectively reduced the runtime needed for fine-tuning. Through our experiments, we observed remarkable results, demonstrating a significant reduction of up to 50% in fine-tuning time (See Appendix Table 4) compared to traditional approaches. This substantial improvement in efficiency offers new possibilities for researchers, developers, and organizations, enabling them to deploy and fine-tune large language models more rapidly and effectively.

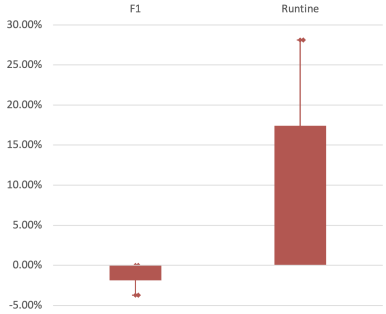


Figure 2: F1 and Runtime averages cross tasks.

5 Conclusion and Future Work

Our results suggest that freezing limited numbers of layers from the bottom in combination with top layers provide an optimal performance. It successfully addressed the challenge of time-consuming fine-tuning for large language models. This indicate that the perturbation from the fine-tuning can be controlled best using this approach; further, the approach might generalized better for out of domain data, as it keeps all the knowledge learnt during the pre-training. By introducing the layers freezing, we were able to achieve impressive time savings that reached up to 50% of time required for fine-tuning compared to conventional methods. This achievement in computation time optimization adds to the major advancement in the field of NLP and deep learning in general. It not only empowers researchers to conduct experiments and iterate more swiftly but also enhances the practicality of implementing large language models in real-world applications. For future work, we plan to expand this research to cover more tasks to ground these findings. More models with different architecture will be needed as well as applications in other languages. In other direction, we plan to explore the impact of the approach on generalization to out of domain and unseen data. Such explorations will validate the approach and demonstrate its merits.

References

Ahmed Abdelali, Sabit Hassan, Hamdy Mubarak, Kareem Darwish, and Younes Samih. 2021. Pre-training bert on arabic tweets: Practical considerations. *arXiv preprint arXiv:2102.10684*.

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2021. ARBERT & MARBERT: Deep bidirectional transformers for Arabic. In *Proceedings of the 59th Annual Meeting of the ACL and the 11th IJCNL Processing*, Online.

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

Houda Bouamor, Sabit Hassan, and Nizar Habash. The MADAR shared task on arabic fine-grained dialect identification. In *Proceedings of the 4th Arabic Natural Language Processing Workshop*.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*.

Alexis Conneau, Guillaume Lample, Ruty Rinott, Adina Williams, Samuel R Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. Xnli: Evaluating cross-lingual sentence representations. *arXiv preprint arXiv:1809.05053*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Ibrahim Abu El-Khair. 2016. Abu el-khair corpus: A modern standard arabic corpus. *International Journal of Recent Trends in Engineering & Research (IJRTER)*, 2(11):5–13.

Bilal Ghanem, Jihen Karoui, Farah Benamara, Véronique Moriceau, and Paolo Rosso. 2019. Idat at fire2019: Overview of the track on irony detection in arabic tweets. In *Proceedings of the 11th Annual Meeting of the FIRE*, pages 10–13.

Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2021. The interplay of variant, size, and task type in arabic pre-trained language models. *arXiv preprint arXiv:2103.06678*.

Elizabeth D Liddy. 2001. Natural language processing.

Yuhan Liu, Saurabh Agarwal, and Shivaram Venkataraman. 2021. Autofreeze: Automatically freezing model blocks to accelerate fine-tuning. *arXiv preprint arXiv:2102.01386*.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. *Adv. in neural information processing systems*, 30.

Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. Semeval-2018 task 1: Affect in tweets. In *Proceedings of the 12th international workshop on semantic evaluation*, pages 1–17.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2021. On the stability of fine-tuning BERT: misconceptions, explanations, and strong baselines. In *9th International Conference on Learning Representations, ICLR 2021*. OpenReview.net.

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 OSACT, with a Shared Task on Offensive Language Detection*, pages 48–52.

XiPeng Qiu, TianXiang Sun, YiGe Xu, YunFan Shao, Ning Dai, and XuanJing Huang. 2020. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63(10):1872–1897.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Hassan Sajjad, Fahim Dalvi, Nadir Durrani, and Preslav Nakov. 2023. On the effect of dropping layers of pre-trained transformer models. *Computer Speech & Language*, 77:101429.

Haitham Seelawi, Ibraheem Tuffaha, Mahmoud Gzawi, Wael Farhan, Bashar Talafha, Riham Badawi, Zyad Sober, Oday Al-Dweik, Abed Alhakim Freihat, and Hussein Al-Natsheh. 2021. Alue: Arabic language understanding evaluation. In *Proceedings of the Sixth WANLP*, pages 173–184.

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

Dongling Xiao, Han Zhang, Yukun Li, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. 2020. Ernie-gen: An enhanced multi-flow pre-training and fine-tuning framework for natural language generation. *arXiv preprint arXiv:2001.11314*.

Chenghao Yang and Xuezhe Ma. 2022. Improving stability of fine-tuning pretrained language models via component-wise gradient norm clipping. In *Proceedings of the 2022 Conference on EMNLP*.

Imad Zeroual, Dirk Goldhahn, Thomas Eckart, and Abdelhak Lakhouaja. 2019. Osian: Open source international arabic news corpus-preparation and integration into the clarin-infrastructure. In *Proceedings of the fourth arabic natural language processing workshop*, pages 175–182.

A Appendix A

Detailed results for the selected tasks from ALUE.

Table 1: Anger Detection

	AraBERT		CAMeLBERT		QARiB		MARBERT	
	F1	Runtime	F1	Runtime	F1	Runtime	F1	Runtime
No Freeze	0.711	21.570	0.752	19.615	0.829	18.516	0.825	21.94
Freeze	0.648	16.389	0.756	14.314	0.814	13.702	0.831	16.80
Δ	-8.86%	24.02%	0.53%	27.02%	-1.81%	26.00%	0.73%	23.43%

Table 2: Question to Question Semantic Similarity (Shared Task 8)

	AraBERT		CAMeLBERT		QARiB		MARBERT	
	F1	Runtime	F1	Runtime	F1	Runtime	F1	Runtime
No Freeze	0.548	124.112	0.580	101.875	0.577	120.702	0.591	106.683
Freeze	0.580	101.269	0.581	88.164	0.582	86.650	0.597	97.987
Δ	5.84%	18.41%	0.17%	13.46%	0.87%	28.21%	1.02%	8.15%

Table 3: Irony Detection

	AraBERT		CAMeLBERT		QARiB		MARBERT	
	F1	Runtime	F1	Runtime	F1	Runtime	F1	Runtime
No Freeze	0.742	48.135	0.788	37.100	0.839	36.152	0.828	35.689
Freeze	0.786	38.107	0.768	28.242	0.836	27.824	0.835	27.73
Δ	5.93%	20.83%	-2.54%	23.88%	-0.36%	23.04%	0.84%	22.30%

Table 4: MADAR Shared Task Subtask 1 (Dialect Detection)

	AraBERT		CAMeLBERT		QARiB		MARBERT	
	F1	Runtine	F1	Runtine	F1	Runtine	F1	Runtine
No Freeze	0.670	1453.080	0.707	1289.394	0.700	1298.000	0.696	156.771
Freeze	0.633	668.153	0.690	1010.240	0.687	1020.360	0.695	159.179
Δ	-5.52%	54.02%	-2.40%	21.65%	-1.86%	21.39%	-0.14%	-1.54%

Table 5: Offensive Speech Detection

	AraBERT		CAMeLBERT		QARiB		MARBERT	
	F1	Runtime	F1	Runtime	F1	Runtime	F1	Runtime
No Freeze	0.974	136.949	0.974	126.627	0.979	119.450	0.974	119.09
Freeze	0.976	108.939	0.976	100.423	0.982	94.322	0.980	94.99
Δ	0.20%	20.45%	0.20%	20.69%	0.30%	21.04%	0.62%	20.24%

Table 6: Hate Speech Detection

	AraBERT		CAMeLBERT		QARiB		MARBERT	
	F1	Runtime	F1	Runtime	F1	Runtime	F1	Runtime
No Freeze	0.770	137.422	0.746	126.999	0.856	119.492	0.834	119.432
Freeze	0.767	109.245	0.759	100.671	0.847	94.768	0.854	95.17
Δ	-0.39%	20.50%	1.74%	20.73%	-1.05%	20.69%	2.40%	20.31%

Table 7: Cross-lingual Sentence Representations

	AraBERT		CAMeLBERT		QARiB		MARBERT	
	F1	Runtime	F1	Runtime	F1	Runtime	F1	Runtime
No Freeze	0.525	98.101	0.599	91.603	0.521	92.475	0.448	94.110
Freeze	0.494	91.738	0.571	82.284	0.505	90.435	0.547	88.919
Δ	-5.90%	6.49%	-4.67%	10.17%	-3.07%	2.21%	22.10%	5.52%

Multi-Parallel Corpus of North Levantine Arabic

Mateusz Krubiński[1], **Hashem Sellat**[1], **Shadi Saleh**[1],
Adam Pospíšil[2], **Petr Zemánek**[2], **and Pavel Pecina**[1]

[1]Charles University, Faculty of Mathematics and Physics
{krubinski,sellat,saleh,pecina}@ufal.mff.cuni.cz
[2]Charles University, Faculty of Arts
{adam.pospisil,petr.zemanek}@ff.cuni.cz

Abstract

Low-resource Machine Translation (MT) is characterized by the scarce availability of training data and/or standardized evaluation benchmarks. In the context of Dialectal Arabic, recent works introduced several evaluation benchmarks covering both Modern Standard Arabic (MSA) and dialects, mapping, however, mostly to a single Indo-European language – English. In this work, we introduce a multi-lingual corpus consisting of 120,600 multi-parallel sentences in English, French, German, Greek, Spanish, and MSA selected from the OpenSubtitles corpus (Lison et al., 2018), which were manually translated into the North Levantine Arabic. By conducting a series of training and fine-tuning experiments, we explore how this novel resource can contribute to the research on Arabic MT. We make the dataset publicly available at http://hdl.handle.net/11234/1-5033 for research purposes.

apc	جواز سفري هنيك مع شوية وراق
	مين عم ياكل فطايري؟
arb	جواز سفري هناك مع بعض الأوراق
	من الذي يأكل فطائري ؟
eng	My passport is there, along with some papers.
	Who’s eating my dumplings?
fra	Il y a mon passeport et des papiers dedans.
	Qui mange mes dumplings ?
deu	Dort drin ist mein Pass und einige Papiere.
	Wer isst meine Klöße?
ell	κεί είναι το διαβατήριό μου και μερικά έγγραφα.
	Ποιος τρώει τα ντάμπλιν μου
spa	Dentro está mi pasaporte, además de unos papeles.
	¿Quién se come mis dumplings?

Table 1: Samples from the multi-parallel corpus introduced in this work. Translations in the Indo-European languages and MSA were obtained from the OpenSubtitles-v2018 corpus, and the ones in North Levantine Arabic (apc) were manually translated from MSA (arb).

1 Introduction

Levantine Arabic is considered one of the core units within the Arabic dialectal continuum. It can be divided into at least three dialectal regions (Al-Wer and de Jong, 2017) but the most notable division within this group lies between South Levantine (Palestinian) and North Levantine (based on the urban speech of mainly Beirut and Damascus) with clear differences between the two (Kwaik et al., 2018). At the same time, North Levantine Arabic (also called Syrian or Shami) is perceived as a clearly established linguistic unit with a positive evaluation and perception (Ghobain, 2017).

In the field of Natural Language Processing, North Levantine Arabic is, similarly to other Arabic dialects, considered a low-resource language. It is mainly used for daily speech, and written resources are very scarce. Formal texts are almost exclusively written in Modern Standard Arabic (MSA). Recently, written North Levantine Arabic started to appear in texts posted to social networks that became a useful resource of monolingual datasets for several dialects of Arabic (Abdul-Mageed et al., 2020). Parallel datasets are even scarcer.

In this paper, we introduce a novel multi-parallel corpus where North Levantine Arabic is paired with MSA and several Indo-European languages (English, French, German, Greek, and Spanish). The corpus contains roughly 1 million words on the English side. By targeting the subset of the multi-parallel OpenSubtitles-v2018 (Lison et al., 2018) dataset, we ensure that with a single round of translation, we can achieve the desired multi-lingual, multi-parallel mapping between MSA, Dialectal Arabic and several Indo-European languages. Considering that the OpenSubtitles dataset consists of lines from movie subtitles[1], it should well represent the "everyday dialogue" domain, where the Arabic

[1] https://www.opensubtitles.org

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 411–417
December 7, 2023 ©2023 Association for Computational Linguistics

dialects are most commonly used.

2 Related Work

In their pioneer work, Zbib et al. (2012) introduced a parallel Levantine-English corpus of 138k sentences suitable for training MT systems. The Levantine sentences were extracted from Arabic weblogs and online user groups and translated into English. In follow-up work, Bouamor et al. (2014) translated 2,000 sentences from the Egyptian-English corpus introduced by Zbib et al. (2012) into several Arabic Dialects (including North Levantine Arabic), creating the first multi-parallel corpus of multi-dialectal Arabic. The multi-parallel aspects were further explored (e.g., Bouamor et al., 2018) and the data were compiled into standardized benchmarks (e.g., Sajjad et al., 2020; Nagoudi et al., 2023; Abdelali et al., 2023). Arab-Acquis (Habash et al., 2017) matched multi-parallel corpus of 22 European languages with human translations into MSA – dialectical aspects were not considered. The exploitation of the OpenSubtitles corpus in the context of Arabic MT was previously explored by Nagoudi et al. (2022), who used it to sample training/testing data for translation from four languages (English, French, German, and Russian) into MSA and Alhafni et al. (2022) who sampled English-MSA sentence pairs for the extended Arabic Parallel Gender Corpus (APGC v2.0).

3 Data preparation

As a first step, we filtered the OpenSubtitles-v2018 corpus by identifying lines that are available in all of the desired languages (MSA, English, French, German, Greek, and Spanish), obtaining 3,661,627 sentences. Subsequently, a number of additional filters (for convenience, we applied filters to the English side) were applied:

1. Sentences containing vulgar words (based on a hand-crafted list) were removed.
2. Sentences containing non-standard characters were removed – only punctuation marks, English alphabet letters and digits were allowed.
3. To avoid incomplete sentences, only sentences that start with a capital letter were kept.
4. Very similar sentences were discarded by lowercasing the text, removing punctuation and digits, and removing the duplicates. The goal was not to translate similar sentences like *Good morning* and *Good morning!* or *I was born in 1961* and *I was born in 1983.*
5. To assure the inner variance and semantic richness of the translated text, sentences with less than two words, ones containing very rare words, and sentences with a high proportion of frequent words (frequency-based approach with a manual filtering step) were removed.

Language	ISO 639-3 code	#Words
North Levantine Arabic	apc	738,812
Modern Standard Arabic	arb	802,313
English	eng	999,193
French	fra	956,208
German	deu	940,234
Greek	ell	869,543
Spanish	spa	920,922

Table 2: Word-level statistics of the multi-parallel corpus of North Levantine Arabic introduced in this work.

Those heuristics were necessary to both filter out low-quality sentences and to down-sample the set of translation candidates to fit within the available budget. We acknowledge that potentially valuable, semantically rich utterances that e.g., do not start with a capital letter, may have been dropped.

After those filtering steps, we ended up with 120,771 sentences. Before the translation, an additional corpus-wise filtering step was applied by removing multi-parallel lines where: English characters appear in the Arabic sentence, Arabic characters appear in the English sentence, or Arabic characters appear in a particular sentence for all of the Indo-European languages. The final size of the corpus is equal to 120,600 lines that were manually translated into the North Levantine Arabic dialect.

The translation was performed by native speakers of the dialect through a professional translation company without using any MT or CAT tool. Considering the lack of official spelling standards for Levantine, we did not provide the translators with specific orthographic guidelines (Habash et al., 2018), but rather relayed on their expertise, asking only for internal consistency. First, a sample of 1,000 sentences was translated independently from English and from MSA. No difference in translation quality was observed (assessed by authors of the paper – speakers of North Levantine Arabic). Therefore, all the remaining sentences were translated from MSA (this direction was less costly). The translation was done in batches of 5,000 sentences, and the quality of the translation was checked after each batch (again by the authors of the paper – speakers of the dialect). In order to quantitatively measure the impact of the source language, we computed the Overlap Coefficient (OC) (Bouamor et al., 2014) for the samples of 1,000 sentences that were used initially[2]. The OC value measures the percentage of lexical overlap between the vocabularies of two languages (dialects). The OC similarity between the MSA source translated into apc target equals 35.95, and the one between the (parallel) MSA and the target apc when translating from English equals 26.85. To put those numbers into context, the OC value between the 1,000 sentences in MSA and Syrian that were independently translated from Egyptian by Bouamor et al. (2014) equals 39.85. Those results indicate that the variety in the apc output may have been slightly reduced by translating from MSA. However, it should be mentioned that we compare disjoint sets of sentences, and there is not enough data to say how this affects the downstream tasks, such as MT.

Sentence samples (multi-parallel lines) are presented in Table 1, and some corpus-wise word-level statistics are presented in Table 2.

4 MT Experiments

In order to demonstrate the validity of the corpus, we conducted a number of MT experiments and evaluations.

Baselines and Metrics We report the performance of two well-established baselines: a multilingual NLLB model (Costa-jussà et al., 2022), using the `facebook/nllb-200-distilled-600M` variant (600M parameters) from the Transformers (Wolf et al., 2020) package, and uni-directional models (depending on the language pair, between 76M and 240M parameters) provided by the Helsinki-NLP group (Tiedemann, 2020). To indicate to what extent MSA can be used when the dialectal system is not available, we translate into both arb (e.g., Opus_{arb}) and apc, always using the apc files as reference. We measure the output quality by reporting the surface-level chrF++[3] metric (Popović, 2015), and the trainable, estimator-based COMET[4] metric (Rei et al., 2020).

Testing data In Table 3, we report performance on the test split of FLORES-200 (Costa-jussà et al., 2022), which consists of professional translation of sentences sampled from the English Wikipedia. In Table 4, we report on the subset[5]of MADAR (Bouamor et al., 2018), which was created by translating sentences from the Basic Traveling Expression Corpus (Takezawa et al., 2007) into several country- and city-level Arabic dialects. Since the original English and French versions of the corpus are not directly available[6], we use only the English side, as provided by the AraBench (Sajjad et al., 2020) benchmark. We report only on the test-sets corresponding to Damascus and Aleppo, as we were unable to directly match the Beirut one from MADAR to the English file in AraBench.

North Levantine Corpus In order to demonstrate the importance of pre-training, we train (Base_{ML}) a multi-lingual Transformer (Vaswani et al., 2017) model from scratch, training with the default `transformer-big` configuration (200M parameters) from the Marian toolkit (Junczys-Dowmunt et al., 2018) on the multi-parallel corpus introduced in this work. We use the source-tagging approach (Johnson et al., 2017), training on all (84) available directions, with an early stopping applied if chrF++ on FLORES-200 dev-set ceases to improve for 10 consecutive evaluations.

Furthermore, we use it to fine-tune both Opus (Opus_{FT}) and NLLB (NLLB_{FT}) models. For uni-directional Opus models, we use only mono-directional data (e.g., apc-ell) and the recommended[7] parameters. We fine-tune the NLLB model on the apc-centric data (i.e., on all of the available directions with apc as source and target) using AdamW (Loshchilov and Hutter, 2019) optimizer with a constant learning rate of 1e-5, obtaining the best results after a single epoch of fine-tuning.

5 Results

Automatic metrics The Base_{ML} system trained from scratch achieves the lowest scores on both test-sets. On average, the larger, multi-lingual NLLB model achieves better scores than the Opus models. Translating into arb gives consistently higher scores for sentences from the FLORES-200 test-set, but lower ones for sentences from MADAR. We attribute this to the vastly different nature of those test-sets. Sentences in FLORES-200 are long, with

[2] We have normalized and tokenized the sentences with the CAMeL Tools (Obeid et al., 2020) package.

[3] nrefs:1|case:mixed|eff:yes|nc:6|nw:2|space:no|version:2.3.1

[4] Model signature: Unbabel/wmt22-comet-da

[5] Lines marked as corpus-6-test-corpus-26-test

[6] https://camel.abudhabi.nyu.edu/madar-parallel-corpus

[7] https://github.com/Helsinki-NLP/OPUS-MT-train/blob/master/finetune

…→**apc**	**arb**		**eng**		**fra**		**deu**		**ell**		**spa**	
	ChrF	COMET	ChrF	COMET	ChrF	COMET	ChrF	COMET	ChrF	COMET	ChrF	COMET
Opus$_{\text{arb}}$	-	-	**51.47**	**.836**	**38.54**	**.800**	**42.98**	**.799**	30.87	**.745**	35.87	**.791**
Opus$_{\text{apc}}$	-	-	**50.55**	**.825**	38.28	.795	37.54	.749	-	-	34.77	.777
Opus$_{\text{FT}}$	-	-	48.48	.786	35.70	.725	**39.24**	.730	31.47	.698	33.56	.722
Base$_{\text{ML}}$	13.17	.449	12.55	.431	12.61	.425	12.42	.414	12.45	.437	12.44	.427
NLLB$_{\text{arb}}$	**47.72**	**.882**	45.38	.824	**39.05**	**.800**	38.68	**.787**	**35.79**	**.784**	**36.21**	**.794**
NLLB$_{\text{apc}}$	44.12	**.832**	43.43	.795	37.03	.759	36.22	.735	33.63	.743	34.47	.756
NLLB$_{\text{FT}}$	**49.60**	.823	44.50	.773	38.11	.737	36.96	.718	**35.46**	.731	**36.09**	.739
apc→…												
Opus	-	-	58.13	.803	47.43	.705	46.33	.736	37.28	.750	41.42	.718
Opus$_{\text{FT}}$	-	-	**60.53**	**.837**	47.37	.730	**48.70**	**.769**	37.24	.773	41.66	.749
Base$_{\text{ML}}$	12.08	.425	16.92	.427	15.26	.357	15.80	.325	13.44	.420	16.24	.391
NLLB	**50.16**	**.854**	**59.97**	**.833**	**53.15**	**.783**	**47.19**	**.757**	**41.25**	**.818**	**44.96**	**.785**
NLLB$_{\text{FT}}$	**50.51**	**.854**	58.19	.831	**50.99**	**.777**	45.39	.749	**39.96**	**.811**	**44.26**	**.781**

Table 3: Evaluation results on the FLORES-200 test-set. The two highest-scoring systems in each column are bolded independently for apc source/target. Underlined numbers correspond to a copy-source system. The Greek Opus model does not support dialectal Arabic in the output.

eng→apc	**Damascus**		**Aleppo**	
	ChrF	COMET	ChrF	COMET
Opus$_{\text{arb}}$	26.09	**.770**	25.64	**.761**
Opus$_{\text{apc}}$	26.32	.757	25.71	.748
Opus$_{\text{FT}}$	**38.50**	.754	**40.57**	**.765**
Base$_{\text{ML}}$	19.01	.599	18.78	.599
NLLB$_{\text{arb}}$	24.58	**.761**	24.68	.753
NLLB$_{\text{apc}}$	33.04	.738	33.25	.739
NLLB$_{\text{FT}}$	**37.77**	.756	**37.30**	.756
apc→eng				
Opus	38.53	.689	39.08	.675
Opus$_{\text{FT}}$	51.09	.795	51.27	.780
Base$_{\text{ML}}$	29.08	.600	26.92	.576
NLLB	**56.21**	**.823**	**57.11**	**.815**
NLLB$_{\text{FT}}$	**52.91**	**.821**	**54.74**	**.804**

Table 4: Evaluation results on the subset of MADAR test-set. The two highest-scoring systems in each column are bolded independently for apc source/target.

MADAR	**arb**	**apc**	**apc FT**
NLLB	2.23 ± .30	2.03 ± .08	**1.54 ± .21**
Opus	2.07 ± .10	2.01 ± .21	**1.25 ± .25**
FLORES			
NLLB	2.07 ± .51	2.14 ± .23	**1.72 ± .37**
Opus	1.98 ± .19	2.02 ± .24	**1.57 ± .34**

Table 5: Results of the human evaluation. Scores indicate an average rank assigned to a sentence (lower = better). The lowest-ranked output in each row is bolded.

a high proportion of named entities (e.g., *Throughout 1960s, Brzezinski worked for John F. Kennedy as his advisor and then the Lyndon B. Johnson administration.*), while the ones in MADAR are short and simple (e.g., *Here is my passport.* or *Does that include tax?*).

The effects of fine-tuning on the corpus that we introduce highlight the difficulties of low-resource MT. On the MADAR test-set, coming from a similar domain as the resource introduced in this work, significant improvements can be observed when translating into apc – both for Opus (26.32→38.50) and NLLB (33.04→37.77) models. Similar behavior can be observed for the Opus model when translating into English (38.53→51.09). However, that is not the case for the NLLB model. It is possible that a comparable amount of dialectal Arabic (mixed with MSA) has already been seen on the source side during training, and more sophisticated fine-tuning schemas are required. On the FLORES-200 test-set (different domain), minor improvements can be observed for the NLLB model (on average, +1.97 ChrF when translating into apc), with inconsistent results for the Opus models (37.54→39.24 when translating from deu but 34.77→33.56 when translating from spa).

Human evaluation In order to verify the observations based on automatic metrics, a round of human evaluation was conducted. Two apc speakers were tasked with ranking outputs (translations of the same English sentence) from three systems: one translating into arb, one into apc, and the third one obtained by fine-tuning on the corpus introduced in this work (apc FT), in the context of the English source. The ranking procedure was done independently for both test-sets and both baseline models: NLLB and Opus – our intention was not to compare different MT models but to investigate subtle differences in the translation process. Each annotator scored 200 sentences sampled from FLORES-200 (100 unique and 100 from a control batch used to compute agreement) and 140 sampled from MADAR (60 unique and 80 common). Sentences and model outputs were shuffled to avoid positional bias. Annotators were asked to consider both fluency and adequacy of translations but to prefer the dialectal output. They were not explicitly informed that one of the translations was into arb, giving them the opportunity to rank it higher if the translation was perceived as more natural in the context, e.g., when translating scientific terms or if the dialectal output was ungrammatical.

The cumulative results are summarized in Table 5. In every case, on average, the output of the fine-tuned model is considered the best. On the MADAR test-set, with simple sentences, apc output is preferred, while on the FLORES-200 one, with long and complex ones, arb output is preferred. The raw inter-annotator agreement (the proportion of times both annotators ranked the same sentence equally) equals 0.52, and Cohen's κ, computed[8] with the WMT formulation for rank-based evaluation (Bojar et al., 2016), equals 0.39, indicating (Landis and Koch, 1977) a "fair/moderate" agreement.

6 Conclusions

In this work, a novel, multi-parallel corpus of North Levantine Arabic, based on the OpenSubtitles-v2018 dataset, is introduced. By fine-tuning well-established baseline MT models, we show that the dialectal aspects of language are partially orthogonal to the domain-specific properties – a dialect-specific model fine-tuned on data from a particular domain may perform worse than a more generic model if a domain shift occurs during testing. However, human evaluation confirms that the dialect-specific aspects of the output are still ranked higher and more appreciated by the final users of the MT system.

7 Acknowledgements

This work was supported by the European Commission via its H2020 Program (contract no. 870930) and CELSA (project no. 19/018) and has been using data and tools provided by the LINDAT/CLARIAH-CZ Research Infrastructure (https://lindat.cz), supported by the Ministry of Education, Youth and Sports of the Czech Republic (project no. LM2018101). We thank the anonymous reviewers for their valuable feedback.

Limitations

Multi-parallel alignment. While a number of steps were taken to ensure the quality of the translations provided, it is possible that the multi-parallel alignments may not be perfect with languages different from the one that was used as a source. The OpenSubtitles corpus that we sub-sample from was created semi-automatically.

Multi- vs Uni-directional fine-tuning. When fine-tuning the NLLB model, we use data from all directions – with apc as the source and as the target. One could also consider uni-directional fine-tuning, e.g., only on the spa-apc direction (we explore this variant with the Opus models).

Fine-tuning on mixed data. In our experiments, we use only the corpus introduced in this work for fine-tuning. Better results could be potentially obtained by using mixed data – either with other dialectal datasets or with samples from the high-resource arb.

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. 2023. Benchmarking arabic ai with large language models.

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, and Lyle Ungar. 2020. Toward micro-dialect identification in diaglossic and code-switched environments. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5855–5876, Online. Association for Computational Linguistics.

Enam Al-Wer and Rudolf de Jong. 2017. *Dialects of Arabic*, chapter 32. John Wiley & Sons, Ltd.

Bashar Alhafni, Nizar Habash, and Houda Bouamor. 2022. The Arabic parallel gender corpus 2.0: Extensions and analyses. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1870–1884, Marseille, France. European Language Resources Association.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurélie

[8] https://github.com/cfedermann/wmt16/blob/master/scripts/compute_agreement_scores.py

Névéol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 131–198, Berlin, Germany. Association for 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, May 7-12, 2018*. European Language Resources Association (ELRA).

Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loïc Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation. *CoRR*, abs/2207.04672.

Elham Abdullah Ghobain. 2017. Dubbing melodramas in the arab world; between the standard language and colloquial dialects. *The Arabic Language and Literature*, 2:49.

Nizar Habash, Fadhl Eryani, Salam Khalifa, Owen Rambow, Dana Abdulrahim, Alexander Erdmann, Reem Faraj, Wajdi Zaghouani, Houda Bouamor, Nasser Zalmout, Sara Hassan, Faisal Al-Shargi, Sakhar Alkhereyf, Basma Abdulkareem, Ramy Eskander, Mohammad Salameh, and Hind Saddiki. 2018. Unified guidelines and resources for Arabic dialect orthography. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

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.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda B. Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Trans. Assoc. Comput. Linguistics*, 5:339–351.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Kathrein Abu Kwaik, Motaz Saad, Stergios Chatzikyriakidis, and Simon Dobnik. 2018. A lexical distance study of arabic dialects. *Procedia Computer Science*, 142:2–13. Arabic Computational Linguistics.

J. Richard Landis and Gary G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174.

Pierre Lison, Jörg Tiedemann, and Milen Kouylekov. 2018. OpenSubtitles2018: Statistical rescoring of sentence alignments in large, noisy parallel corpora. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

El Moatez Billah Nagoudi, Ahmed El-Shangiti, AbdelRahim Elmadany, and Muhammad Abdul-Mageed. 2023. Dolphin: A challenging and diverse benchmark for arabic nlg.

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.

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.

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.

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.

Toshiyuki Takezawa, Genichiro Kikui, Masahide Mizushima, and Eiichiro Sumita. 2007. Multilingual spoken language corpus development for communication research. In *International Journal of Computational Linguistics & Chinese Language Processing, Volume 12, Number 3, September 2007: Special Issue on Invited Papers from ISCSLP 2006*, pages 303–324.

Jörg Tiedemann. 2020. The tatoeba translation challenge – realistic data sets for low resource and multilingual MT. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1174–1182, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

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.

Simplify: Automatic Arabic Sentence Simplification using Word Embeddings

Yousef SalahEldin
German International University,
New Administrative Capital, Egypt
yousef.hamouda@student.giu-uni.de

Caroline Sabty
German International University,
New Administrative Capital, Egypt
caroline.sabty@giu-uni.de

Abstract

Automatic Text Simplification (TS) involves simplifying language complexity while preserving the original meaning. The main objective of TS is to enhance the readability of complex texts, making them more accessible to a broader range of readers. This work focuses on developing a lexical text simplification system specifically for Arabic. We utilized FastText and Arabert pre-trained embedding models to create various simplification models. Our lexical approach involves a series of steps: identifying complex words, generating potential replacements, and selecting one replacement for the complex word within a sentence. We presented two main identification models: binary and multi-complexity models. We assessed the efficacy of these models by employing BERTScore to measure the similarity between the sentences generated by these models and the intended simple sentences. This comparative analysis evaluated the effectiveness of these models in accurately identifying and selecting complex words.

1 Introduction

Automatic Text Simplification (TS) aims to make text less linguistically complex without changing its meaning or original information. This involves rewriting a complex text by performing various edit operations such as deletion, replacing words, splitting sentences, and changing the order of words. These actions are part of the TS natural language processing task (Al-Thanyyan and Azmi, 2021).

TS can benefit individuals who struggle with reading and writing, such as those with low literacy skills, dyslexia, or learning a new language. Different simplification techniques can be employed depending on the desired purpose and the end user. Additionally, TS can enhance written communication by ensuring that the target audience comprehends the intended message. (Rello et al., 2013). In addition, automated systems for simplifying text can help make the language more accessible to individuals who are not fluent in it or have limited proficiency.

Detecting text complexity is crucial in TS systems as it helps determine if the text needs to be simplified. It is also helpful in evaluating the results generated by the simplification system. TS systems primarily depend on syntax or lexical simplifications. (Shardlow, 2014).

Text simplification is related to techniques such as creating paraphrases, summarizing text, and machine translation in Natural Language Processing (NLP). Many strategies and evaluation methods used by Text Simplification are derived from these areas. In the past, rule-based syntactic simplification was used as a pre-processing step to improve various NLP tasks like parsing and formulating questions. (Sikka and Mago, 2020).

Arabic is a widely spoken language consistently listed as one of the top 10 most spoken languages. This emphasizes the importance of incorporating different natural language processing tasks for Arabic (Hatab et al., 2022). We utilized the latest technologies in the field of NLP to carry out a straightforward simplification task. We developed two models for identification purposes: one that categorizes text as either complex or non-complex and another that classifies text into various levels of complexity. As a result, we utilized BERT (Devlin et al., 2018) and FastText (Grave et al., 2018) to create the simplification model. We assessed the simplification phase using BERTScore (Zhang* et al., 2020), which involved the two identification models. Furthermore, we conducted a manual evaluation to ensure the quality of the simplified text.

2 Related Work

Unlike English and other languages, only a few researchers have explored Arabic Automatic Text Simplification. In (Al-Subaihin and Al-Khalifa,

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 418–422
December 7, 2023 ©2023 Association for Computational Linguistics

2011), they presented a text simplification tool named "AlBaseet". The tool's structure consisted of four main stages: complexity assessment, lexical simplification, syntax simplification, and diacritization. They followed the LS-pipeline approach to simplify the text and produced synonyms by creating a new vocabulary or utilizing ArabicWordNet (Rodríguez et al., 2008).

The second attempt to construct an Arabic ATS was made by (Al Khalil et al., 2017). Their semi-automatic simplification approach was meant to simplify modern Arabic fiction; a linguist applied ACTFL (American Council on the Teaching of Foreign Languages) language proficiency requirements for simplifying five Arabic books using a web-based tool. They intended to create a readability measurement identifier using various machine learning classifiers to develop a graded reader scale of four levels.

In (Hazim et al., 2022), a method for identifying and visualizing complex words is presented. The authors' method combines lexical and syntactic analysis techniques, such as part-of-speech tagging and dependency parsing, to extract relevant information and create visualizations highlighting individual words' complexity.

A system was proposed in (Khallaf, 2023) that utilizes linguistic resources and rule-based transformations to identify complex linguistic structures and simplify them accordingly.

3 Simplification Approach

There are three stages involved in simplifying complex sentences. Initially, we need to recognize the complex words used in the sentence. After identifying these complex words, we generate alternative options for them that are simpler and more comprehensible. These alternatives can include synonyms, definitions, or rephrasing of the original word. Ultimately, we choose the most appropriate replacement for every intricate term, considering the surrounding context and the overall message conveyed in the text.

3.1 Complex Word Identification

The initial phase, known as Complex Word Identification (CWI), is extremely important because if a complex word is not identified, it will hinder the generation of substitutions in the entire LS architecture. Therefore, the accuracy of the CWI step determines the simplification pipeline's success. Multiple steps are carried out on the given input sentence during this stage.

Initially, we assign a Part-of-Speech tag (POS tag) to every word. Next, we determine specific POS tags that may require simplification. We only focus on examining verbs, nouns, and adjectives for simplification. Additionally, we subject complicated words to a machine-learning algorithm aided by a frequency list. Then, we obtain the complexity of each word. Initially, when provided with a sentence as input, we employ POS tagging to determine the Part-of-Speech for each word. We utilized the Farasa modules (Abdelali et al., 2016) to identify POS Tags in an Arabic sentence.

3.1.1 Pre-processing of Identification Dataset

After identifying the POS Tags of a given word, we determine whether such a word is complex. We trained an ML model using an available Arabic frequency list (Kilgarriff et al., 2014) to train an ML model. The frequency list contained 8904 Arabic words and their level of complexity based on the Common European Framework (CEFR) and the corresponding frequency.

Due to the large percentage of null values in the frequency column, we added our frequency score using Wordfreq[1]. Also, we added a POS Tag for each word using Farasa (Abdelali et al., 2016). Moreover, we added the stem of each word as a new feature, assuming that we want to know the complexity of the origin, as different words will have the same stems, and we removed redundant rows. The final data contains 4258 unique words and their corresponding stem, POS tag, frequency, and label, whether complex or not.

3.1.2 ML Identification Model

We built an ML model that can classify the complexity of each word. We considered building a model using the C-Support Vector Classification (SVC). We did try different combinations of independent features for the ML model. The input of the model contains the stem, POS Tag, and frequency as independent features. A different approach was to give the model word itself rather than its stem, as a stem can vary in complexity in different instances. Accordingly, we did implement two different identification models. The first model, Multi-Comp, was implemented by converting CEFR levels from 1 being the most minor complex to 6 being the most complex, according

[1] https://doi.org/10.5281/zenodo.7199437

to levels ranging between B1 to C2, respectively. We implemented the second model by categorizing CEFR levels into two binary formats. We determined that levels A1 to B1 are classified as not complex, assigning them a value of 0. On the other hand, levels B2 to C2 are considered complex and are given a value of 1. This model is referred to as the Binary model.

3.2 Generation Substitutions

The second stage is to generate substitutions for the complex word. We implemented two approaches: the first was using FastText, and the second was using BERT.

In the first approach, where we used FastText, we calculated the cosine similarity between words using the nearest neighbor module. We implemented a method to determine five similar candidates for a given complex word. However, FastText just produced words in different forms by the nearest neighbor. For example, the word 'ذهب' can be spoken as "Thahaba" or "Dahab", yet both words have entirely different meanings.

In the second approach, we used AraBERT (Antoun et al., 2020).

The masking language model works simply by masking a specific word in the sentence, and the model tries to predict what word can fit that place, given its right and left words. Accordingly, we utilized such a module for substitution generation. Once we have a list of complex words in a sentence, we mask a complex word per time and feed it to AraBert. AraBert then tries to predict the word appropriately fitting into the masked area.

3.3 Selection of Substitutions

We have constructed a sentence where we have inferred difficult words and identified five potential options for each difficult word. AraBERT provides a list of five words and their respective confidence scores, which indicate the level of certainty the model has for each candidate. Therefore, our initial strategy was to replace complex words with those with the highest certainty level. Unfortunately, two obstacles arose. The main obstacle was that sometimes, the word associated with the highest certainty rating was the same complex word. The second point is that we need a way to confirm whether the substituted word is more straightforward. Therefore, we deemed it necessary to include something that ensures the replacement of a complex word with its simpler equivalent.

To guarantee the replacement of the word, we depended on Gensim, an open-source library (Rehurek and Sojka, 2011). Gensim includes a module that measures the similarity between two words. We used this module by setting a condition that if the MLM model identified the complex word as the top candidate, we would calculate the similarity between the complex word and the other candidates. Currently, we possess two distinct identification models. The initial model evaluates complexity using a binary system, assigning either a 1 or 0. On the other hand, the second model assesses complexity using a scale of values ranging from 1 to 6, known as the Multi-Comp Model. We decided to add another condition for the second model to solve the second challenge we faced. The condition states that we will replace the complex word only if the replaced candidate has a lower complexity value. Even if it has the same value as a complex word, we will still keep the complex word to preserve the meaning better. Additionally, we ensured that the replaced candidate was not any ambiguous replacement, so we identified what variations the AraBERT model predicted and eliminated unnecessary replacements.

4 Evaluation & Results

In order to evaluate our models, we needed a parallel corpus. A parallel corpus is a collection of complicated texts and their simplified versions in the same language. To the best of our knowledge, there is only one available parallel corpus for the Arabic language (Al-Raisi et al., 2018). The corpora are in different sizes. The small size contains 8 sentence pairs, the medium-sized size contains 69 sentence pairs, and the large contains 765 sentence pairs.

4.1 Automatic Evaluation

We first evaluated the SVC identification models using different independent features. After, we evaluated our simplification approach using BERTScore. This was because BERTScore overcame the limitations of other metrics and supported the Arabic language.

As shown in Table 1, we tried four different combinations.

As demonstrated in Table 1, we found that using the stem of the word in combination with its frequency resulted in an F1-score of 0.88. From this,

Features	F1 Score
Word/PosTag/Frequency	0.79
Stem/PosTag/Frequency	0.77
Word/Frequency	0.86
Stem/Frequency	**0.88**

Table 1: Table showing results of different identification models

we determined that including a POS tag would only confuse the model, as its variations are quite different in various positions. By comparing features based on the stem or the words, we found that using the stem is more effective. It is more accurate to always provide the model with the stem of a word rather than providing various forms of the word, as this can lead to confusion in the model.

To assess the performance of both identification models in a sentence simplification system, we opted to examine their effectiveness using varying sizes of parallel corpora. Small, medium, and large sizes were evaluated by BERTScore using 'bert-base-multilingual-cased', which supports the Arabic language and many different languages. The results we obtained are shown in Table 2:

Lexical	P	R	F1
	Small		
Target/Binary-Model	0.836	0.843	0.830
Target/Multi-Comp-Model	0.848	0.858	**0.853**
	Medium		
Target/Binary-Model	0.864	0.872	0.868
Target/Multi-Comp-Model	0.876	0.885	**0.885**
	Large		
Target/Binary-Model	0.863	0.871	**0.867**
Target/Multi-Comp-Model	0.858	0.866	0.862

Table 2: Results showing both models on different sizes of parallel corpora

The findings suggest that the Multi-Comp model outperformed the Binary Model for both small and medium-sized datasets in the machine translation system. However, the Binary Model performed better than the Multi-Comp model when evaluating extensive corpora. This suggests that the Binary Model is more adaptable in dealing with diverse text types. This is likely because large corpora usually cover a range of topics and text formats, and the Binary Model is less likely to become confused when replacing words, unlike the Multi-Comp model, which may struggle with the complexity involved.

4.2 Manual Evaluation

We aimed to assess our model's performance by collaborating with human experts. To achieve this, we designed a survey comprising 20 randomly selected samples. Each sample included both input and output texts. The input text was a complex passage from the parallel corpora, while our models generated the output text. We evaluated the model by including 3 features, which are: 1) Meaning Preservation (MP), 2) Grammaticality (G), and 3) Simplicity (S) (Laban et al., 2021). We asked their experts to rate every sample on the three features on a scale of 1 to 5.

When addressing meaning preservation, we found that the Multi-Comp model outperforms the Binary model with a 69% rate of preserving meaning in the output texts. Moreover, it also outperformed the Binary model grammar-wise with a rate of 84% sustaining the grammar in the outputs. The only measurement that the Binary model leveraged was the most critical measurement, which is simplicity. Among the output texts, 79% were simpler than inputs.

The results of the manual evaluation show that there is a significant trade-off between the three measurements. The Binary model excels in simplicity but has a downside regarding meaning preservation and grammar. The model prioritizes simplifying complex words over preserving the meaning of the sentence, which leads to a loss of meaning preservation and grammar in the output. In other words, the model sacrifices meaning preservation and grammar to generate more straightforward text. This trade-off highlights the challenge of balancing multiple metrics in natural language processing tasks.

5 Conclusion and Futute Work

To conclude, we endeavored to develop a lexical text simplification system for Arabic. We introduced two models for identification: the Binary Model and the Multi-Comp Model. Furthermore, we suggested several simplification approaches utilizing FastText and AraBERT embeddings. Our perception of the lexical system restrictions is based on the fact that certain of the generated sentence structures need to be better-formed, and the system can incorrectly recognize complex words from simple ones in the CWI phase. In the future, it would be beneficial to utilize more recent models for evaluation.

6 Limitations

Overall, we presented the advantages and disadvantages of our proposed approach. We specifically emphasized the drawbacks of the CWI step. One drawback of CWI is its limited ability to accurately identify complex words, primarily because it needs a dependable frequency list. Another crucial consideration in our proposed approach is finding a balance between simplifying a sentence without compromising its intended meaning and maintaining proper grammar. Furthermore, the availability of a parallel corpus is crucial for undertaking such a task, and we need more resources in Arabic.

References

Ahmed Abdelali, Kareem Darwish, Nadir Durrani, and Hamdy Mubarak. 2016. Farasa: A fast and furious segmenter for Arabic. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 11–16, San Diego, California. Association for Computational Linguistics.

Muhamed Al Khalil, Nizar Habash, and Hind Saddiki. 2017. Simplification of arabic masterpieces for extensive reading: A project overview. *Procedia Computer Science*, 117:192–198.

Fatima Al-Raisi, Weijian Lin, and Abdelwahab Bourai. 2018. A monolingual parallel corpus of arabic. *Procedia computer science*, 142:334–338.

Afnan A Al-Subaihin and Hend S Al-Khalifa. 2011. Al-baseet: A proposed simplification authoring tool for the arabic language. In *2011 International Conference on Communications and Information Technology (ICCIT)*, pages 121–125. IEEE.

Suha S Al-Thanyyan and Aqil M Azmi. 2021. Automated text simplification: a survey. *ACM Computing Surveys (CSUR)*, 54(2):1–36.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Ali L Hatab, Caroline Sabty, and Slim Abdennadher. 2022. Enhancing deep learning with embedded features for arabic named entity recognition. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4904–4912.

Reem Hazim, Hind Saddiki, Bashar Alhafni, Muhamed Al Khalil, and Nizar Habash. 2022. Arabic word-level readability visualization for assisted text simplification. *arXiv preprint arXiv:2210.10672*.

Nouran Abdelrahman Ahmed Khallaf. 2023. *An Automatic Modern Standard Arabic Text Simplification System: A Corpus-Based Approach*. Ph.D. thesis, University of Leeds.

Adam Kilgarriff, Frieda Charalabopoulou, Maria Gavrilidou, Janne Bondi Johannessen, Saussan Khalil, Sofie Johansson Kokkinakis, Robert Lew, Serge Sharoff, Ravikiran Vadlapudi, and Elena Volodina. 2014. Corpus-based vocabulary lists for language learners for nine languages. *Language resources and evaluation*, 48(1):121–163.

Philippe Laban, Tobias Schnabel, Paul Bennett, and Marti A Hearst. 2021. Keep it simple: Unsupervised simplification of multi-paragraph text. *arXiv preprint arXiv:2107.03444*.

Radim Rehurek and Petr Sojka. 2011. Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, 3(2).

Luz Rello, Ricardo Baeza-Yates, Laura Dempere-Marco, and Horacio Saggion. 2013. Frequent words improve readability and short words improve understandability for people with dyslexia. In *IFIP Conference on Human-Computer Interaction*, pages 203–219. Springer.

Horacio Rodríguez, David Farwell, Javi Farreres, Manuel Bertran, Musa Alkhalifa, M Antonia Martí, William Black, Sabri Elkateb, James Kirk, Adam Pease, et al. 2008. Arabic wordnet: Current state and future extensions. In *Proceedings of The Fourth Global WordNet Conference, Szeged, Hungary*, 387-405.

Matthew Shardlow. 2014. A survey of automated text simplification. *International Journal of Advanced Computer Science and Applications*, 4(1):58–70.

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

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*.

Offensive Language Detection in Arabizi

Imene Bensalem
ESCF de Constantine
MISC Lab, Constantine 2
University, Algeria
ibensalem@escf-constantine.dz

Meryem Ait Mout
Polytech Marseille,
Aix-Marseille Université,
France
meryem.AIT-MOUT@etu.univ-amu.fr

Paolo Rosso
Universitat Politècnica
de València,
Spain
prosso@dsic.upv.es

Abstract

Detecting offensive language in under-resourced languages presents a significant real-world challenge for social media platforms. This paper is the first work focused on the issue of offensive language detection in Arabizi, an under-explored topic in an under-resourced form of Arabic. For the first time, a comprehensive and critical overview of the existing work on the topic is presented. In addition, we carry out experiments using different BERT-like models and show the feasibility of detecting offensive language in Arabizi with high accuracy. Throughout a thorough analysis of results, we emphasize the complexities introduced by dialect variations and out-of-domain generalization. We use in our experiments a dataset that we have constructed by leveraging existing, albeit limited, resources. To facilitate further research, we make this dataset publicly accessible to the research community.

1 Introduction

Due to the unrestricted nature of online discourse, offensive language has found its way to social media platforms, which poses major challenges for maintaining a respectful and inclusive virtual environment. Processing social media texts in Arabic presents its own set of challenges, as the user-generated content in this language is often not written in standard Arabic, but instead in multiple dialects, which vary from one country to another and have no grammatical and orthographic rules. Additionally, the use of Arabizi[1] (Alghamdi and Petraki 2018; Brabetz 2022; Haghegh 2021; Yaghan 2008)–an informal system of writing Arabic using Latin alphabet and numbers, which is commonly blended with French and English–further complicates Arabic processing (Darwish 2014).

Arabizi is characterized by the numerous transliterations of a single word, which may create a new set of homonyms within Arabic and even with other languages[2]. For example, in the dataset used for this research, we were able to find 7 different Arabizi spellings of the word قلب (*heart*), which are *alb, aleb, 9alb, kalb, galb, guelb, gelb*. The 2 first ones have been found in the Lebanese dialect, whereas the rest are used in the Algerian dialect, showing different pronunciations across regions.

Due to this inconsistency in writing this vernacular digital Arabic, traditional offensive language detection methods may struggle to interpret accurately the Arabizi words and expressions unique to each dialect. To illustrate, the word *kalb*, listed above as one of the spelling forms of قلب (*heart*), is also the transliteration of كلب (*dog*), which is used, in addition to its literal meaning, as an insult in the Arab world.

Arabizi has been studied in various contexts, such as its identification and transliteration to Arabic (Darwish 2014; Shazal et al. 2020), code-switching detection (Shehadi and Wintner 2022), POS tagging (Muller et al. 2020) and sentiment analysis (Fourati et al. 2021; Guellil et al. 2021). Besides, there has been a notable increase in the number of papers focusing on Arabic offensive language detection in recent years (Husain and Uzuner 2021). Nonetheless, there is a scarcity of research dedicated to handling Arabizi specifically within the context of offensive language detection.

This paper is dedicated to addressing this gap. Our contributions are the following:

- We provide, for the first time, a

[1] Also knows as Romanized Arabic and Arabic chat alphabet.

[2] Shehadi and Wintner (2022) showed some Arabizi words that have meanings in English and Hindi.

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 423–434
December 7, 2023 ©2023 Association for Computational Linguistics

comprehensive overview of the existing works addressing offensive language detection in the context of Arabizi;

- We assess the performance of various language models (mBERT, DziriBERT, DarijaBERTarabizi, SVM) in detecting the offensive language in the Arabizi text without transliterating it to the Arabic script[3]. Our experiments are both in-domain and out-of-domain;

- We analyze the results per each of the two dialects (Algerian, Lebanese) composing the used dataset, which allows shedding light on the behaviour of the leveraged pre-trained models;

- Finally, we make available[4] the used dataset, which we created by merging data and unifying the annotation from 4 available datasets.

The remainder of this paper is structured as follows. Section 2 offers a critical overview of the works dealing with Arabizi in the context of offensive language detection. Section 3 details the process of the dataset creation. Sections 4 and 5 are devoted to the experimentation and the presentation of their outcomes. Finally, Section 6 discusses the findings and conclusions.

2 Related work

A small number of offensive language detection works have dealt with Arabizi (Appendix A presents a summary of each of these works, along with a recap in Table 1). However, these works exhibit one or more of the following shortcomings:

- The dataset is predominantly written in Arabic script with only a minority of examples in Arabizi (Boucherit and Abainia 2022; Mohdeb et al. 2022; Röttger et al. 2022);

- The dataset is conceived to serve primarily a different task than offensive language detection (Abainia 2020; Raïdy and Harmanani 2023; Riabi et al. 2023), resulting in a small size or a low proportion of offensive examples;

- The conducted experiments or the reported results did not focus on offensive language detection in Arabizi (Abainia 2020; Boucherit and Abainia 2022; Mohdeb et al. 2022; Raïdy and Harmanani 2023);

- In the few works (Riabi et al. 2023; Röttger et al. 2022) that reported results on Arabizi, the datasets have a small number of Arabizi examples, and they did not encompass social media texts. Consequently, their results do not allow to make definitive assessments regarding the performance of models in this specific text genre.

Considering the shortcomings of the previous studies listed above, and the prevalent use of Arabizi, it became clear that there is a pressing need to pay more attention to the problem of offensive language detection on Arabizi. To address this need, there is a requirement for the creation of additional resources that would facilitate a thorough evaluation of this task.

3 Dataset

In light of the above discussion on the limitation of the available datasets, our objective is to create a single, relatively large dataset with a plausible ratio of offensive language. This dataset could be then exploited for the development of offensive language detection models. Inspired by the work of (Risch et al. 2021), we favoured leveraging the available resources instead of starting from scratch. Therefore, we decided to construct the dataset by merging the Arabizi samples from the datasets DZMP, DZOFF, DZREF and LBSA (*cf.* Table 1 in Appendix A for details on these datasets). To achieve this, we followed the subsequent steps:

Extraction of the Arabizi samples from the datasets that comprise, in addition, Arabic script. This was straightforward for the DZREF dataset, as it comprises an attribute determining whether the text is in Arabic script, Arabizi, French or English. For the DZOFF dataset, however, we made this extraction automatically by filtering out the messages that contain only the Latin alphabet and numbers.

Unification of the labels. Our goal is to obtain a dataset for binary classification where the

[3] Before the era of large language models, transliterating Arabizi to the Arabic alphabet has been a common practice in Arabic language processing tasks such as sentiment analysis (Matrane et al. 2023).

[4] https://github.com/Imene1/Arabizi-offensive-language

offensive class encompasses a wide range of abusive text including hate speech (with its subcategories such as racism and sexism), profanity and obscene content. To this end, all the labels referring to any kind of offensiveness (*cf.* Table 1), were integrated into one label (Offensive). Similarly, all the labels indicating the absence of offensive language were mapped to one label (Non-Offensive). For this task, we examined carefully the definitions of labels provided in the paper or the documentation of each dataset. For the majority of labels, it was easy to decide whether it represents offensive language or not.

Nonetheless, for a few labels, where the definition was not enough to decide, we examined, in addition, a sample of the data having this label. This was the case, of two labels: "Refusing with non-hateful words (RNH)" in the anti-refugee dataset (DZREF) and the label "sarcasm" in the Lebanese sentiment analysis dataset (LBSA). By inspecting some examples of the class RNH, we decided to consider them among the offensive class. This is because despite those messages do not contain swearing, they exhibit discrimination and xenophobia, which is in line with the wide definition of offensive language we adopted. Concerning the cases in the "sarcasm" class, we examined all the examples in this class that have the sentiment polarity "negative", assuming that the offensiveness could not be positive. This examination showed us that all the cases are non-offensive.

Merging the datasets. We have merged into one CSV file the entire examples of DZMP and LBSA datasets along with the Arabizi parts of DZOFF and DZREF datasets.

As shown in Table 2[5], the obtained dataset comprises more than 7000 social media texts from different platforms. More than 20% of its textual examples are offensive, which is an acceptable ratio to train a detection model. Given its distinctive features, including its size, the proportion of offensive content, the two different dialects it contains, and its diverse sources from social media, we assert that this dataset is currently the most suitable choice for evaluating the performance of offensive language detection in Arabizi, which is addressed in the next section.

4 Experiments

The goal of our experiments is 3 fold:

- To estimate the performance of detecting offensive language specifically on Arabizi in two contexts: in-domain and out-of-domain.
- To analyze the performance per dialect.
- To gain insights into the misclassified cases.

We carried out our experiments with 3 variants of BERT. Below is a succinct overview of them.

Multilingual BERT (a.k.a. mBERT) [6] (Devlin et al. 2019): BERT Language model pre-trained on 104 languages including Arabic. Previous experiments using this model in the context of POS tagging and dependency parsing (Muller et al. 2020), as well as sentiment analysis (Fourati et al., 2021), proved it can generalize to handle Arabizi by fine-tuning it using datasets in this form of Arabic.

DziriBERT [7] (Abdaoui et al. 2021): BERT Language model pre-trained on more than one million tweets in the Algerian dialect including Arabizi.

DarijaBERT-arabizi[8] (Gaanoun et al. 2023): a variant of BERT pre-trained on more than 4 Million texts on the Moroccan dialect (a.k.a. Darija) written in Arabizi.

As baselines, we used SVM with TF-IDF as features and the majority class heuristic.

To provide a robust estimate of those model's performance, we applied the following evaluation setup:

For the **in-domain** context, we fine-tuned the three BERT-like models and trained SVM through 5-fold cross-validation using the created dataset.

The obtained models were then tested on two **out-of-domain** datasets, which are the Arabizi part of EGMHC (Röttger et al. 2022) and DZTRB (Riabi et al. 2023). The former comprises synthetic Arabizi texts in Egyptian and the second comprises Algerian Arabizi collected from a news website and a song lyrics corpus (*cf.* Appendix A for further details on these datasets).

The hyper-parameters used to fine-tune BERT models are displayed in Table 3. Adam optimizer was used in all the models.

[5] Due to space limitations, Tables 2-8 are included in Appendix C.

[6] https://github.com/google-research/bert. The cased version is used.

[7] https://github.com/alger-ia/dziribert

[8] https://huggingface.co/SI2M-Lab/DarijaBERT-arabizi

5 Results and discussion

5.1 In-domain results

Table 4 displays the performance scores of the models in terms of F1 measured on the 2 classes offensive and non-offensive as well as the macro-averaged F1 and the accuracy. It should be noted that those measures are computed on the predictions file wherein the results obtained from the cross-validation folds are appended. We also reported the average of the F1 scores computed on the 5 folds (they are displayed between parenthesis on the table).

The results show that all the models outperformed the majority class baseline. Even SVM, which is not context-aware and does not have any prior knowledge on Arabizi was able to classify correctly 17% of the offensive texts with a precision of 97%[9], resulting in an F1 of 0.29.

The mBERT model reached an F1 score of 0.92, showing an improvement of +0.32 in comparison with SVM's result. This means that the contextual embeddings that this model learned from multiple languages allowed it to capture some of the patterns in the Arabizi text even though it was not pre-trained with this form of Arabic, which confirms the findings of previous studies (Fourati et al. 2021; Muller et al. 2020).

DziriBERT and DarijaBERT-arabizi perform almost equally and surpass mBERT, most notably in the offensive class. This shows the advantageous impact of pre-training BERT with Arabizi. Additionally, those results suggest that knowledge can be effectively transferred across Arabic dialects, even when expressed using Latin script. This is illustrated by the good performance of DarijaBERT-arabizi on our dataset, which comprises Algerian and Lebanese Arabizi, despite being pre-rained on Moroccan Arabizi.

In the following section, we will delve deeper into the analysis of performance per dialect to gain further insights.

5.2 Performance per Dialect

Table 5 shows the performance scores computed on the examples of each dialect separately. With regard to the Lebanese dialect, the performance of all the models in the non-offensive class is outstanding and superior to their performance in the offensive class. This result is indeed expected since the Lebanese sub-dataset is extremely imbalanced (the ratio of the offensive texts is only 6%), which makes it easy to reach a high-performance score on the majority class. This is evidenced by the F1 score of 0.97, which was reached on the non-offensive class just by a random guess using the majority class heuristic.

Interestingly, although the Lebanese dialect was unseen in the pre-training phase of the three used BERT models, DziriBERT and DarijaBERT-arabizi generated good results on the offensive class, with a raise of +0.14 and +0.13 respectively in F1 score in comparison with mBERT. This supports our previous remark concerning the transferability of knowledge across dialects, meaning that knowledge on the Algerian dialect (and also the Moroccan dialect) was useful in improving the offensive language detection performance on the Lebanese dialect despite the fact that those dialects are very different from each other.

In the context of the Algerian dialect, DarijaBERT-arabizi achieved the highest performance, showing only a marginal distinction from DziriBERT. This outcome is quite predictable because it is expected that knowledge transfer from the Moroccan dialect to the Algerian one would be effective given the substantial similarities between these dialects. Appendix B provides an analysis of the misclassified cases with examples from the dataset.

5.3 Out-of-domain results

Table 7 reports the results of testing the models that were fine-tuned through the experiments described in Section 5.1 on two unseen datasets. It should be noted that the EGMHC dataset comprises also texts in the Arabic script, but we reported, in Table 7, the results computed only on the Arabizi examples, which all belong to the positive class[10]. Therefore, using accuracy would be enough to measure the performance on this dataset. Additionally, the results obtained on DZTRB were computed only on its test set (DZTRB$_{test}$), with the aim of allowing their comparison with the results reported in (Riabi et al. 2023) (displayed in the last three lines). Note that, the models in Riabi's et al. paper have been trained on the training set of DZTRB and tested on

[9] Precision and Recall are not displayed in the tables of results. We mentioned them for illustration reasons.

[10] The positive class in this dataset is *hateful*, which we mapped to *offensive* to be compatible with the label of the dataset used previously to fine-tune the models.

its test set. Our main observations on the obtained results are below.

The SVM model failed to identify any offensive instances in either dataset, suggesting that the content in EGMHC and DZTRB deviates significantly from the training data, thereby impeding the model's ability to generalize.

Overall, the performance is very poor on EGMHC, indicating, again, a high dissimilarity of this dataset with the ones used to pre-train and fine-tune the models. This dissimilarity could be related to the fact that this dataset is in the Egyptian dialect, which was unseen in the fine-tuning and pre-training phases of all the used models.

On the other hand, the performance on the DZTRB$_{test}$ dataset was not as low as the one on EGMHC, and fairly close to the results achieved by the models fine-tuned on the training subset of this dataset, obtained from Riabi et al. paper[11]. This could be explained by the fact that the texts in DZTRB are in the Algerian dialect, which is a seen dialect in the fine-tuning phase. This allows the models to generalise to some extent on this dataset.

Unlike the in-domain results, mBERT generated the highest accuracy score on EGMHC and the highest F1 on the offensive class on DZTRB$_{test}$. Nonetheless, its score remains too poor to be significant.

Those results show different difficulty degrees for the models to generalise across datasets, illustrated by the very low results on EGMHC and the moderate results on DZTRB. In both cases, this implies the necessity of domain adaptation to improve performance. In this context, we were able to find only a couple of works addressing the topic of domain adaptation across Arabic dialects in the context of offensive language detection (Husain and Uzuner 2022) and sentiment analysis (El Mekki et al. 2021). Consequently, further investigation in this area is warranted.

6 Conclusion

In this research paper, we have explored the fascinating topic of offensive language detection within Arabizi. Regarding this topic is still underexplored, our study aimed to shed light on the performance of models in this specific linguistic context. Throughout our investigation, the following key findings have emerged:

Feasibility of detecting offensive language in Arabizi: despite the complexity of Arabizi, our experiments demonstrated that offensive language in this form of Arabic could be detected with high-performance scores without transliterating it to the Arabic script. This was evidenced by an F1 score that researched 0.96 using cross-validation on a dataset comprised of texts collected from different sources and in two distinct dialects, Algerian and Lebanese. However, the generalizability of the models across datasets is a challenge, especially if the dialect is different, as shown through our out-of-domain experiments.

The role of pre-trained language models: we showed that while a plausible performance could be reached by multilingual BERT, the best results are obtained by the models pre-trained partially or totally with Arabizi, which are DziriBERT and DarijaBERT-arabizi, respectively. On the other hand, SVM, a traditional machine learning model, generated poor results. This highlights the importance of transfer learning and context-aware models in dealing with the complexity of Arabizi.

Challenges in dialectal variation: our dialect-specific analysis of results along with our inspection of the misclassified cases revealed that despite the transferability of knowledge between dialects, it remains essential to tailor approaches to each dialect for better performance. This is particularly important because the vocabulary of offensive language may vary among the various dialects. This finding was underscored by our out-of-domain experiments, showing the difficulty of models to generalize on an unseen dialect (Egyptian).

Those findings suggest that future research efforts, in the context of offensive language detection in Arabizi, have to focus on the development of more datasets and pre-trained language models for the various Arabic dialects, as the majority of the existing resources concern the Algerian dialect. They also highlight the necessity of domain adaption research, most notably across dialects.

Finally, we anticipate that the findings of this study and the dataset we have made publicly accessible will pave the way for further research on this topic.

[11] Even the results of Riabi et al. are low. The best macro F1 is 0.61 as indicated in Table 7. See Appendix D for further details on DZTRB dataset.

Limitation

The limitation of our research is twofold. First, we used in our experiments a dataset comprising only Arabizi texts. However, this does not reflect the distribution in the real world, wherein Arabic script and Arabizi coexist together, sometimes in a single message. Moreover, code-switching to French and English occurs frequently in Arabizi, an aspect that we did not investigate in our experimentations. Therefore, it would be important, in future studies, to consider these two real-world aspects.

Second, our research did not address the transliteration of Arabizi to the Arabic script. We used instead models compatible with Arabizi. Thus, it is still unknown whether the transliteration improves the performance.

Acknowledgements

This publication was made possible by NPRP grant 13S-0206-200281 from the Qatar National Research Fund (a member of Qatar Foundation). The findings achieved herein are solely the responsibility of the authors.

References

Kheireddine Abainia. 2020. DZDC12: a new multipurpose parallel Algerian Arabizi–French code-switched corpus. *Language Resources and Evaluation*, 54(2):419–455.

Amine Abdaoui, Mohamed Berrimi, Mourad Oussalah, and Abdelouahab Moussaoui. 2021. DziriBERT: a Pre-trained Language Model for the Algerian Dialect. *Preprint arXiv 2109.12346*.

Hamdah Alghamdi and Eleni Petraki. 2018. Arabizi in Saudi Arabia: A deviant form of language or simply a form of expression? *Social Sciences*, 7(9).

Oussama Boucherit and Kheireddine Abainia. 2022. Offensive Language Detection in Under-resourced Algerian Dialectal Arabic Language. *arXiv preprint arXiv:2203.10024*:1–9.

Giulia Brabetz. 2022. Arabizi: A Linguistic Manifestation of Glocalization in the Arabic Language Area? *Maydan: rivista sui mondi arabi, semitici e islamici*, 2:103–129.

Kareem Darwish. 2014. Arabizi Detection and Conversion to Arabic. *ANLP 2014 - EMNLP 2014 Workshop on Arabic Natural Language Processing, Proceedings*:217–224.

Mithun Das, Saurabh Kumar Pandey, and Animesh Mukherjee. 2023. Evaluating ChatGPT's Performance for Multilingual and Emoji-based Hate Speech Detection.

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

Abdellah El Mekki, Abdelkader El Mahdaouy, Ismail Berrada, and Ahmed Khoumsi. 2021. Domain Adaptation for Arabic Cross-Domain and Cross-Dialect Sentiment Analysis from Contextualized Word Embedding. *NAACL-HLT 2021 - 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference*:2824–2837.

Chayma Fourati, Hatem Haddad, Abir Messaoudi, Moez Ben Haj Hmida, Aymen Ben Elhaj Mabrouk, and Malek Naski. 2021. Introducing A large Tunisian Arabizi Dialectal Dataset for Sentiment Analysis. *WANLP 2021 - 6th Arabic Natural Language Processing Workshop, Proceedings of the Workshop*:226–230.

Kamel Gaanoun, Abdou Mohamed Naira, Anass Allak, and Imade Benelallam. 2023. DarijaBERT : A Step Forward in NLP for the Written Moroccan Dialect. Technical report.

Imane Guellil, Ahsan Adeel, Faical Azouaou, Fodil Benali, Ala Eddine Hachani, Kia Dashtipour, Mandar Gogate, Cosimo Ieracitano, Reza Kashani, and Amir Hussain. 2021. A Semi-supervised Approach for Sentiment Analysis of Arab(ic+izi) Messages: Application to the Algerian Dialect. *SN Computer Science*, 2(2):1–18.

Mariam Haghegh. 2021. Arabizi across Three Different Generations of Arab Users Living Abroad: A Case Study. *Arab World English Journal For Translation and Literary Studies*, 5(2):156–173.

Fatemah Husain and Ozlem Uzuner. 2021. A Survey of Offensive Language Detection for the Arabic Language. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 20(1):1–44.

Fatemah Husain and Ozlem Uzuner. 2022. Transfer Learning Across Arabic Dialects for Offensive Language Detection. *2022 International Conference on Asian Language Processing, IALP 2022*:196–205.

Yassir Matrane, Faouzia Benabbou, and Nawal Sael. 2023. A systematic literature review of Arabic dialect sentiment analysis. *Journal of King Saud*

University - Computer and Information Sciences, 35(6):101570.

Djamila Mohdeb, Meriem Laifa, Fayssal Zerargui, and Omar Benzaoui. 2022. Evaluating transfer learning approach for detecting Arabic anti-refugee/migrant speech on social media. *Aslib Journal of Information Management*, 74(6):1070–1088.

Benjamin Muller, Benoit Sagot, and Djamé Seddah. 2020. Can Multilingual Language Models Transfer to an Unseen Dialect? A Case Study on North African Arabizi. *arXiv preprint arXiv:2005.00318*.

Maria Raïdy and Haidar Harmanani. 2023. A Deep Learning Approach for Sentiment and Emotional Analysis of Lebanese Arabizi Twitter Data. In *ITNG 2023 20th International Conference on Information Technology-New Generations, Advances in Intelligent Systems and Computing 1445*, pages 27–35.

Arij Riabi, Menel Mahamdi, and Djamé Seddah. 2023. Enriching the NArabizi Treebank : A Multifaceted Approach to Supporting an Under-Resourced Language. In *Proceedings of the 17th Linguistic Annotation Workshop (LAW-XVII)*, pages 266–278. Association for Computational Linguistics (ACL).

Julian Risch, Philipp Schmidt, and Ralf Krestel. 2021. Data Integration for Toxic Comment Classification: Making More Than 40 Datasets Easily Accessible in One Unified Format. *WOAH 2021 - 5th Workshop on Online Abuse and Harms, Proceedings of the Workshop*:157–163.

Paul Röttger, Haitham Seelawi, Debora Nozza, Zeerak Talat, and Bertie Vidgen. 2022. MULTILINGUAL HATE CHECK : Functional Tests for Multilingual Hate Speech Detection Models. In *Proceedings of the Sixth Workshop on Online Abuse and Harms (WOAH)*, pages 154–169. Association for Computational Linguistics (ACL).

Ali Shazal, Aiza Usman, and Nizar Habash. 2020. A Unified Model for Arabizi Detection and Transliteration using Sequence-to-Sequence Models. *Proceedings of the Fifth Arabic Natural Language Processing Workshop*:167–177.

Safaa Shehadi and Shuly Wintner. 2022. Identifying Code-switching in Arabizi. *WANLP 2022 - 7th Arabic Natural Language Processing - Proceedings of the Workshop*:194–204.

Mohammad Ali Yaghan. 2008. “Arabizi”: A Contemporary Style of Arabic Slang. *Design Issues*, 24(2):39–52.

A Summaries of the Related Works

Typically, non-Arabic characters and numbers are considered noise and hence deleted during the processing of Arabic text or the creation of datasets. This practice results in the omission of the Arabizi script. Indeed, only a few offensive language detection datasets involve this form of Arabic. In this section, we provide an overview of those few works.

Abainia (2020) created a multipurpose dataset (DZMP) [12] in Arabizi comprising 12 Algerian sub-dialects. The dataset is collected from Facebook and annotated for several tasks, namely code-switching, sub-dialect identification, emotion detection, gender identification, and abusive language detection. The abusive comments constitute only 12% of the dataset. To the best of our knowledge, this dataset has not been yet exploited in abusive language detection experiments.

Mohdeb et al. (2022) addressed the problem of detecting anti-refugee and anti-migrant speech. They created a dataset (DZREF) composed of more than 4500 YouTube comments in the Algerian dialect including 434 comments in Arabizi with code-switching to French and English. Their experiments, using different variants of BERT, showed that the performance of the hate speech detection models is impacted negatively when including the Arabizi comments. However, further investigation is needed in this regard since the percentage of Arabizi in the used dataset is too small (only 9%).

Röttger et al. (2022) constructed a particular dataset known as Multilingual Hatecheck. It is a functional test, which encompasses synthetic texts in 10 languages. The Arabic subset (EGMHC), which is mostly in the Egyptian dialect, contains 3570 cases of both hateful and non-hateful content. These cases were carefully crafted by language experts using numerous templates, where the hate speech target and the slur word vary across the cases. The purpose of this dataset is to allow a controlled evaluation of hate speech detection models based on 25 fine-grained functionalities. Each functionality reflects the ability of the model to correctly classify specific kinds of hate or non-hate speech (e.g., implicit derogation, counter

[12] Throughout the paper, we use acronyms to denote each dataset. We constructed these acronyms by combining an abbreviation of the dialect (based on the ISO 3166-1 alpha-2 code of the respective country) with additional letters that indicate the dataset's primary purpose.

Authors	Source	Main Task (dataset acronym)	Dialect	Overall Size (size and proportion of Arabizi)	Annotation related to off. language	% off. examples in the Arabizi part
(Abainia 2020)	f	**M**ultipur**p**ose (DZMP)	DZ	2400 (2400, 100%)	**Abusive** , Not Abusive	12%
(Boucherit and Abainia 2022)	f	**Off**ensive language detection (DZOFF)	DZ	8749 (1415, 16%)	**Abusive**, **offensive**, none	61%
(Mohdeb et al. 2022)	▶	Anti-**ref**ugees and anti-migrant hate speech detection (DZREF)	DZ	4586 (434, 9.5%)	**Hate**, **Incitement**, Sympathetic, **Refusing with non-hateful words**, Comment (not hateful, nor sympathetic)	44%
(Raïdy and Harmanani 2023)	🐦	**S**entiment **a**nalysis (LBSA)	LB	3134 (100%)	**Bullying**, Courtesy words, **Foul language**, Joke, Known fact, **Racism**, **Sarcasm**, Saying, **Sectarianism**, **Sexism**, None	6%
(Röttger et al. 2022)	Synthetic text	Functional test for hate speech detection (EGMHC)	EG	3570 (133, 4%)	Hateful, Non-hateful	100%
(Riabi et al. 2023)	News website and song lyrics	Treebank (DZTRB)	DZ	1287 (1287, 100%)	Offensive , Non-offensive	22%

Table 1: Datasets comprising annotated offensive content in Arabizi. In Dialect column, the acronyms DZ, LB, EG refer to Algerian, Lebanese and Egyptian, respectively. In Size column, the first figure refers to the total number of examples in the dataset, while the figures inside parentheses are the number of Arabizi examples and their proportions to the overall size. To construct our dataset, the Arabizi data of the first 4 datasets were merged, wherein the annotation labels appearing in bold were mapped to the label offensive, and the rest were mapped to the label non-offensive. The two last datasets are not collected from social media, and we used them for the out-of-domain experiments.

speech), or cases exhibiting lexical or syntactic phenomena (e.g., negation, spelling variation including Arabizi). This dataset was used to evaluate an XLM-T model (a multilanguage model) that was fine-tuned using known offensive language datasets in 3 Latin languages. The model achieved an accuracy rate of 60.9% on the 133 Arabizi examples. In a recent work (Das et al. 2023), this dataset was also employed to evaluate ChatGPT, which achieved an accuracy of 75.9% on the Arabizi examples. However, it was not able to classify 20.3% of the cases.

Boucherit and Abainia (2022) proposed a dataset of offensive language detection (DZOFF) in the Algerian dialect crawled from Facebook. The dataset contains more than 8500 texts, in Arabic and Arabizi scripts, sampled from public pages and groups of controversial topics. Each text has been labelled as abusive, offensive or normal following the definitions provided by the authors. According to these definitions, offensive language includes any offence targeting individuals, groups or entities, whereas abusive speech corresponds to swearing or obscene content. The examples in Arabizi represent 16% of the dataset. The dataset was used in binary (offensive and abusive language classes are merged) and multiclass classification experiments employing traditional and deep learning models. Although a significant portion of the dataset is in Arabizi, the paper did not report the performance of the models specifically on this script.

The work of (Raïdy and Harmanani 2023) concerns sentiment analysis in the Lebanese dialect. In addition to the polarity labels (positive and negative), the created dataset (LBSA), which is entirely in Arabizi, contains labels providing hints on the tweets' content e.g., sexism, sectarianism, jokes, and sarcasm, among others. Only a small proportion of texts (6%) have labels referring to offensive content. Moreover, those labels have not been considered in the conducted experiments.

Contemporaneously with our work[13], Riabi et al. (2023) enriched the North African Arabizi Treeback with offensive language annotation. The dataset is composed of 1287 sentences in the Algerian dialect sampled from two sources: a corpus of user comments crawled from a newspaper website and a corpus of lyrics of Algerian songs. The paper reported results of offensive language detection experiments using BERT-like models. However, since the dataset is

[13] Riabi et al. work was published in July 2023 while we were conducting our experiments.

small and not collected from social media, it would be difficult to draw solid conclusions about the performance of those models in a real-world scenario.

B Error Analysis

To better understand the behaviours of the models in the in-domain experiments, we calculated the percentage of the misclassified cases by each model in each class based on the annotation of the source datasets composing our dataset (*cf.* Table 1. It displays the labels of the source annotation). Since all the source datasets adopted a ternary or multiclass annotation, it would provide a more precise description of the text than the binary classes we have adopted. Then, we averaged the misclassification percentages of the 4 models for each class.

Table 6 shows that the easiest texts to predict as non-offensive are the ones labelled as known fact from the LBSA dataset. All the 21 cases with this label have been classified correctly by all the models (see example 1 in Table 8)[14]. On the other hand, the easiest class to predict as offensive is the class Abusive from the DZOFF dataset, which is constituted of texts with obscene and swear words (ex. 2).

The non-offensive class with the highest ratio of false positives is Comment from the DZREF dataset. As shown in the table, more than 7% (on average) of the examples in this class, which is superposed to contain neutral discourse, have been flagged as offensive. After the examination of the cases that were marked as offensive by at least one model (totalled 32 cases), it turns out that nearly one-third of these cases are effectively offensive, meaning they were mis-annotated. Some of them involve untargeted swearing (ex. 3) and others involve hate speech but the targets were not refugees or migrants (ex. 4). This may explain why the annotators did not consider them as positive cases (e.g., hate speech), given that DZREF dataset specifically concerns hate speech directed at refugees and migrants.

We can also observe in Table 6 that the proportions of misclassification in the offensive classes (i.e., the false negatives) are higher than the proportion of misclassification in the non-offensive classes (i.e., the false positives). Furthermore, almost all the classes with the highest ratio (of false negatives) belong to the Lebanese dataset. For instance, the unique example that constitutes the class Racism (ex. 5) was predicted by the 4 models as non-offensive, resulting in a ratio of misclassification equal to 100%, followed by the class Foul language with an average of 42.3% of misclassification.

Since the number of cases misclassified by SVM is high, we examined only the cases predicted as non-offensive by the 3 BERT models. We noticed 3 kinds of cases:

- Error in the annotation or challenging examples.
- Texts with an implicit offence that employ terms very specific to the context and the culture of the country.
- Texts comprising well-known insults and obscene words.

The first two cases were present in examples in both Algerian and Lebanese dialects (ex. 6-8). However, interestingly, the last kind of errors was observed only among the texts in the Lebanese dialect comprising obscene words not used in the Algerian dialect. For example, the texts involving the obscene terms *ke**m* and Cha**ta (see the full texts in ex. 9, 10) have not been marked as offensive. This means that the classification models failed to identify some of the well-known swear words in the Lebanese dialect. Conversely, this is not the case in the Algerian dialect: as mentioned previously, the texts comprising swearing in the Algerian dialect were the easiest to identify as offensive). Therefore, it would be reasonable to attribute this discrepancy to two reasons:

1. the fact that two of the used models are pre-trained on the Algerian dialect or the Moroccan (which is similar to the Algerian), but not pre-trained on the Lebanese dialect,
2. the limited number of the offensive examples in Lebanese used to fine-tune the models (only 194 examples), which is not the case for the Algerian dialect (more than 1000 examples).

[14] All the examples from the datasets are listed in Table 8. To avoid the repetition of the table number, we will refer to the next examples only by mentioning the example number between parentheses.

In other words, the Lebanese data used to fine-tune the models, pre-trained on the Maghrebi dialects, were not diverse enough to effectively extend the applicability of these models to detect offensive language specific to the Lebanese dialect.

C Tables

Total # of examples	Dialects (ratio)	Platforms	# Offensive examples (%)
7383	DZ (57.5%) LB (42.5%)	Facebook YouTube Twitter	1526 (20.7%) DZ: 1332 LB : 194

Table 2: Statistics of the constructed dataset.

Learning rate	1e-5
Batch size	16
Number of epochs	3

Table 3: The used hyperparameters for the BERT models.

	F1			**Acc.**
	Non-Off.	**Off.**	**Macro**	
DziriBERT	**0.98**	0.93	**0.96** (0.93)	**0.97**
DarijaBERT-arabizi	**0.98**	**0.94**	**0.96** (0.93)	**0.97**
mBERT	0.97	0.87	0.92 (0.86)	0.95
SVM	0.90	0.29	0.60 (0.60)	0.83
Majority Class	0.88	0.00	0.44	0.79

Table 4: In-domain evaluation results using 5-fold cross-validation.

		F1			**Acc.**
		Non-Off.	**Off.**	**Macro**	
DziriBERT	DZ	0.97	0.94	0.96	0.96
	LB	**0.99**	**0.88**	**0.94**	**0.99**
DarijaBERT-arabizi	DZ	**0.98**	**0.95**	**0.97**	**0.97**
	LB	**0.99**	0.83	0.91	0.98
mBERT	DZ	0.95	0.89	0.92	0.93
	LB	0.98	0.74	0.86	0.97
SVM	DZ	0.84	0.32	0.58	0.75
	LB	0.97	0.06	0.51	0.94
Majority Class	DZ	0.81	0.00	0.41	0.69
	LB	0.97	0.00	0.48	0.94

Table 5: Dialect-specific Performance. The best results in the Algerian dialect (DZ) are highlighted in bold, while the best results in the Lebanese dialect (LB) are both bold and underlined.

Generic Class	Source dataset	Source class	# examples	% misclassifi-ations
Non-offensive	LBSA	Known fact	21	**0.00%**
	LBSA	Sarcasm	111	0.23%
	LBSA	Joke	112	0.45%
	LBSA	None	2631	0.50%
	DZMP	Not abusive	2119	1.12%
	LBSA	Courtesy words	32	1.56%
	LBSA	Saying	33	2.27%
	DZOFF	Normal	556	3.15%
	DZREF	Sympathetic	65	6.92%
	DZREF	Comment	177	7.20%
Offensive	DZOFF	Abusive (swearing and obscene content)	363	**14.94%**
	DZOFF	Offensive	496	26.46%
	DZREF	Incitement	8	28.13%
	DZREF	Hate	138	31.52%
	DZREF	Refusing with non-hateful words	46	33.15%
	LBSA	Sexism	2	37.50%
	DZMP	Abusive	281	38.52%
	LBSA	Sectarianism	14	41.07%
	LBSA	Bullying	60	41.25%
	LBSA	Foul language	117	42.31%
	LBSA	Racism	1	100%

Table 6: Average misclassification ratio of the 4 models.

		DZTRB$_{test}$				EGMHC (Arabizi part)
		F1			**Acc.**	**Acc.**
		Off.	**Non-Off**	**Macro**		
Fine-tuned on our dataset	DziriBERT	0.19	0.90	0.54	0.82	0.04
	DarijaBERT-arabizi	0.22	0.89	0.56	0.81	0.12
	mBERT	0.26	0.86	0.56	0.77	0.15
	SVM	0.00	0.90	0.45	0.81	0.00
Fine-tuned on DZTRB's training set	DziriBERT (Riabi et al., 2023)	0.37	0.85	0.61	-	-
	mBERT (Riabi et al., 2023)	0.00	0.90	0.45	-	-
	CharacterBERT (Riabi et al., 2023)	0.25	0.80	0.52	-	-

Table 7: Performance scores of testing the models on two out-of-domain datasets DZTRB$_{test}$ and EGMHC.

1	A true negative example from the source class Known fact in LBSA dataset (a class with 0% misclassification)	
	Arz	Absha3 shi bl safar huwe dab L shenat □□♀□😃
	Ar	أبشع شي بالسفر هو ضب الشنط □□♀□😃
	En	The worst thing about traveling is to pack suitcases.
2	One of the examples that was correctly predicted as offensive by the 4 models. It belongs to the class Abusive in DZOFF dataset, which is the offensive class with the smallest misclassification ratio. It contains text with obscene and swear words	
	Arz	Roh ta3**i ya n**ch
	Ar	روح تع**ي يا ن**ش
	En	Go get fu**ed, passive gay man.
3	An untargeted obscene popular word in the Algerian dialect, which was mis-annotated in DZREF dataset. Interestingly, it was predicted as offensive by mBERT in addition to DziriBERT.	
	Arz	tn**et
	Ar	تن**ت
	En	It’s fu**ed
4	A mis-annotated offensive example from the DZREF dataset, which does not target African refugees or migrants.	
	Arz	bravo sahafi bravo france tfou lik
	Ar	برافو صحافي برافو، فرنسا تفو عليك
	En	Bravo, journalist, bravo! France spit on you.
5	The unique example in the Racism class from the LBSA dataset. It was marked as non-offensive by the 4 models.	
	Arz	plz plz gebran 5alik mtebe3 lmawdo3 ma ba2 badna phalesteneye wsoreyen 3ena el mawjoden bykafo
	Ar	بليز بليز جبران، خليك متابع الموضوع ما بقا بدنا فلسطينيّ وسوريين عنا الموجودين بيكفوا
	En	Please, please Gebran, stay tuned to the topic. We don't want Palestinians and Syrians here; the ones we have are enough.
6	An example of mis-annotation from the DZOFF. It was erroneously annotated as offensive but predicted as non-offensive by the 4 models. This expression is typically said by someone who feels wronged, directing it towards the wrongdoer. It conveys a sense of reliance on God's justice and intervention.	
	Arz	Hasbiya Allah wa ni3ma.el wakil fik
	Ar	حسبي الله ونعم الوكيل فيك
	En	God suffices me, and He is the best disposer of affairs concerning you.
7	An example in the Foul language class from the LBSA dataset. It was classified by the 4 models as non-offensive. This example is not inherently a foul language but it can becomes offensive if directed at a person a disrespectful manner. This is indeed a challenging case for annotation and classification if the context is unknown.	
	Arz	chou hal habel hayda man :p ma32oul
	Ar	شو هالهبل هيدا مان. معقول
	En	What is this nonsense, man :p I can't believe it.
8	False negative from DZOFF dataset: subtle offence that employ the term “sahib l kachir” translated literally to “people of sausage”, which is very specific to the context of some political events in Algeria. This term refers to individuals who are perceived as being supportive of the government and are brought to governments’ rallies with the incentive of receiving a sandwich containing sausage.	
	Arz	Hadou ysemhoum sehab l kachir […]
	Ar	هادو يسموهم صحاب الكاشير[…]
	En	Those are called people of sausage […]
9	An example of false negative from the LBSA although comprising a common swear word in the oriental Arabic dialects such as the Levantine and Egyptian.	
	Arz	Chou hal cha***ta
	Ar	شو هالش**طة
	En	What is this bi**h
10	Another example of a false negative although comprising a well-known swear word: “K**m”. This is a highly offensive term in oriental Arabic dialects. It literally translates to derogatory terms related to female genitalia. The intended meaning is a strong curse directed at someone, often expressing extreme anger or disdain. The English translation below is not literal.	
	Arz	Mitl a ade bi Beirut.... k***m Nasrallah!
	Ar	متل العادة ببيروت...ك***م نصرالله
	En	Like usual in Beirut, curse on Nasrallah!
11	An example from $\text{DZTRB}_{\text{test}}$ dataset annotated as offensive. This could be considered a subjective annotation, illustrating the challenge of classifying the cases of this dataset.	
	Arz	nhab bladi w dima lalgerie w makra fl 3adyan
	Ar	نحب بلادي ودائماً للجزائر ومكرا في العديان
	En	I love my country and always for Algeria to spite enemies

Table 8: Examples from the used datasets. Each Arabizi example (Arz) is transliterated to the Arabic script (Ar) and translated to English (En)

D Performance on DZRTB dataset

An examination of a sample from DZTRB$_{\text{test}}$ confirmed the remark of its authors that harmful cases are indeed challenging even for annotators, which is also illustrated by the moderate inter-annotator agreement of 0.54. Most cases are about football, do not involve swear words, and the annotation seems subjective (see ex. 11 in Table 8). The difficulty of this dataset could be also illustrated by the low performance of the models trained and tested on it as reported in (Riabi et al., 2023): the best model (DziriBERT) yielded an F1 of 0.61 and mBERT did not detect any offensive case.

Yet Another Model for Arabic Dialect Identification

Ajinkya Kulkarni
MBZUAI, UAE
ajinkya.kulkarni@mbzuai.ac.ae

Hanan Aldarmaki
MBZUAI, UAE
hanan.aldarmaki@mbzuai.ac.ae

Abstract

In this paper, we describe a spoken Arabic dialect identification (ADI) model for Arabic that consistently outperforms previously published results on two benchmark datasets: ADI-5 and ADI-17. We explore two architectural variations: ResNet and ECAPA-TDNN, coupled with two types of acoustic features: MFCCs and features exratected from the pre-trained self-supervised model UniSpeech-SAT Large, as well as a fusion of all four variants. We find that individually, ECAPA-TDNN network outperforms ResNet, and models with UniSpeech-SAT features outperform models with MFCCs by a large margin. Furthermore, a fusion of all four variants consistently outperforms individual models. Our best models outperform previously reported results on both datasets, with accuracies of 84.7% and 96.9% on ADI-5 and ADI-17, respectively.

1 Introduction

Dialect identification can be viewed as a special case of language recognition (Tong et al., 2006; Vijayan et al., 2018). Both tasks suffer from similar performance issues in the presence of background noise, channel mismatch, prosodic fluctuations, and so on. However, with closely related dialects having a small difference in both acoustic and linguistic feature space, dialect identification tasks are substantially more difficult in nature (Zaidan and Callison-Burch, 2014). The Arabic language is spoken in various dialects across the Arab world, in addition to Modern Standard Arabic (MSA) which is used in official and educational settings. Speech recognition systems trained on MSA data generally don't generalize well to dialectal Arabic and specialized dialectal models may be needed for improving automatic speech recognition (ASR) performance in systems developed for specific populations. Dialect identification could facilitate the development of dialectal speech recognition systems in various ways, such as by identifying dialectal utterances in large multi-dialectal corpora, or online dialect identification for routing utterances to dialect-specific ASR modules.

To enable the development of spoken Arabic dialect identification systems, two benchmark datasets have been developed: ADI-5, which was deployed as part of the MGB-3 challenge (Ali et al., 2017) and ADI-17, deployed as part of the MGB-5 challenge (Ali et al., 2019). For both challenges, the top systems developed and submitted for the initial challenges remain the best performing systems reported in the research literature for these benchmarks. The ADI-5 training set consists of 10 hours of dialectal speech from broadcast news, covering five dialects: Egyptian (EGY), Levantine (LAV), Gulf (GLF), North African (NOR), and Modern Standard Arabic (MSA), in addition to two hours each for development and test sets. The ADI-17 data set consists of 17 dialectal classes for a total of 3K hours extracted automatically from YouTube. Roughly 58 hours of data were manually verified for the development and test sets.

In this paper, we describe spoken dialect identification models we developed and tested on these benchmarks, and we report results exceeding the best performing models submitted to both challenges. We experimented with the Residual networks (ResNet) (He et al., 2015) and Emphasized Channel Attention, Propagation and Aggregation (ECAPA-TDNN) (Desplanques et al., 2020) architectures. Both architectures have been successfully employed for speaker verification tasks. In addition, ResNet was used in the best performing dialect identification system in the MGB-5 challenge, and ECAPA-TDNN has been recently explored for dialect classification, as in Lonergan et al. (2023) for Irish dialects. In addition, we explored the use of acoustic features extracted from the UniSpeech-SAT (Chen et al., 2021) model, which have been shown to provide improvements in various tasks in the SUPERB benchmark (Yang et al., 2021). We

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 435–440
December 7, 2023 ©2023 Association for Computational Linguistics

observe large improvements in accuracy by incorporating these features into our models. We also employ data augmentation via additive noise and speed perturbation, which generally help improve the generalization of speech classification models. Our best model result is 84.7% accuracy in the ADI-5 test set, compared to 75% previously reported as the best result in Ali et al. (2017). In ADI-17, our best model achieves 96.9% accuracy compared to 94.9% previously reported as the best model in Ali et al. (2019).

2 Related Work

In this section, we describe the approaches proposed for ADI tasks in MGB-3 and MGB-5 challenges, which are used as baseline systems in this work. We first describe the top two performing systems for the MGB-3 challenge (ADI-5) (Ali et al., 2017), followed by the top two systems in the MGB-5 challenge (ADI-17) (Ali et al., 2019).

The MIT-QCRI ADI system (Shon et al., 2017; Khurana et al., 2017) combines acoustic and linguistic features within a Siamese neural network framework to reduce dimensionality based on i-vectors. They used loss functions involving both Euclidean and cosine distances and employed support vector machines as the backend classifier. In contrast, the University of Texas at Dallas (UTD) submission (Bulut et al., 2017) to the MGB-3 challenge fused five systems, incorporating acoustic and lexical information through various techniques, including i-vectors, Generative Adversarial Networks (GANs), Gaussian Back-end (GB), and BNF i-vector features. The UTD system obtained the second-best performance with an overall accuracy of 70.38% (Ali et al., 2017).

Duke Kunshan University (DKU) submitted four variants of ResNets with different block sizes and datasets, which were fused to achieve the best performing system in the MGB-5 challenge (Ali et al., 2019). The DKU system employed a ResNet with global statistics pooling and a fully connected layer. They used the Kaldi toolkit for data augmentation, including speed-perturbation and datasets such as MUSAN and RIR. The ResNet system was trained using cross-entropy loss with a softmax layer, taking 64-dimensional mel-filterbank energy features as input. On the other hand, the University of Kent (UKent) MGB-5 system (Miao and Mcloughlin, 2019) used a neural network architecture combining Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks with Time-Scale Modification (TSM). The UKent system reported an accuracy of 93.1% on the test set.

While the best performing models reported in the original MGB-3 and MGB-5 challenges have not been outperformed in later publications (to the best of our knowledge), several other studies proposed model variants and analyzed the performance in various ways. Regarding the use of pre-trained self-supervised acoustic models, Sullivan et al. (2023) recently utilized the XLS-R model (Babu et al., 2022), which is a multi-lingual pre-trained acoustic model that includes Arabic as one of the languages used in pre-training, and HuBERT (Hsu et al., 2021), which was pre-trained solely in English. They fine-tuned dialect classification models on the ADI-17 dataset, and interestingly, the model based on HuBERT outperformed the XLS-R-based model, in spite of the multi-lingual pre-training of the latter. This indicates that the quality of the features extracted from pre-trained acoustic models may depend more on the self-supervised training details rather than linguistic coverage. A model outperforming HuBERT on several benchmark tasks is the UniSpeech-SAT acoustic model (Chen et al., 2021), which includes additional objectives on top of the HuBERT model to facilitate speaker-aware representations, which also generally embody non-linguistic characteristics of utterances, such as tone and emotion.

3 Proposed Model

As the space of possible architectural or feature variations increases with the increasing volume of developments in the ML field, exhaustively searching all possible architectures is unfeasible. Therefore, we draw inspiration from the best performing models in related literature to reduce the search space and increase the likelihood of finding a best performing model. We selected two neural network architectures, ResNet and ECAPA-TDNN, for their potential in speech classification tasks. For feature extraction, we compare classical MFCC features with the pre-trained UniSpeech-SAT large acoustic model (Chen et al., 2021) that has been shown to provide consistent improvements in various Speech classification benchmarks. Finally, as best models in previous works typically include a form of ensemble, we experimented with fusing all model variants to further improve performance. We describe the details of these parts in this section.

3.1 Feature extraction

We experimented with two types of features: classical acoustic features, namely MFCCs, and modern acoustic features extracted from a large pre-trained acoustic model, namely the Universal Speech representation learning with speaker-aware pre-training (UniSpeech-SAT) (Chen et al., 2021). The large variant of this model demonstrated outstanding performance in various tasks in the SUPERB benchmark (wen Yang et al., 2021), including linguistic and non-linguistic tasks, such as speaker diarization and emotion recognition. UniSpeech-SAT model is built on the HuBERT model (Hsu et al., 2021) with additional self-supervised objectives involving utterance-wise contrastive learning and utterance mixing augmentation. The speaker-aware pre-training enabled the model to improve the discriminating capabilities of embeddings learned under self-supervised learning. In total, the large variant of UniSpeech-SAT was trained on 94K hours of English speech data from various sources, including Audiobooks and YouTube. We extracted 1024-dimensional features from the pre-trained UniSpeech-SAT[1] model and kept model parameters frozen. For MFCCs, we extract 80-dimensional features using a window length of 25 ms with a sliding window of 10 ms and frame-level instance normalization.

3.2 Network architectures

We experimented with two network architectures that have been shown to work well in speech classification tasks: ResNet and ECAPA-TDNN, which we describe below.

3.2.1 ResNet

We use the ResNet architecture (He et al., 2015) as our first model. Our model is composed of four residual networks, each consisting of two convolutional layers in addition a skip connection. We utilize batch normalization and ReLU activation functions. Statistical pooling is implemented to map the variable length feature frames to a time-invariant representation by aggregating frame level mean and variance as statistical parameters. The output of statistical pooling is followed by two feed-forward layers. We employ the original ResNet34 set-up as described in the original paper (He et al., 2015), which has 34 2D-convolutional layers organized into 4 residual network blocks, with each block containing a specific number of layers [3, 4, 6, 3], and the convolutional filters for these layers are [32, 64, 128, 256] respectively. The last feed-forward layer includes the output dimension of a number of dialect classes to identify with Additive Angular Margin (AAM) softmax layer (Deng et al., 2018) with a scale of 30.0 and margin of 0.4, trained with cross-entropy loss function.

3.2.2 ECAPA-TDNN

The ECAPA-TDNN architecture (Desplanques et al., 2020), based on the x-vector architecture (Snyder et al., 2018), utilizes a Squeeze-excitation (SE)-Res2Net module in each block. These modules consist of 1-dimensional convolutional layers, ReLU activation, batch normalization, and 1-dimensional Res2Net modules with impactful skip connections and SE blocks. This design allows the model to extract hierarchical and global information from the input features. Additionally, the architecture incorporates attentive statistical pooling by calculating channel-dependent frame attention-weighted statistics (mean and variance). This process transforms variable-length hidden outputs into a time-invariant representation. The representation is further processed through feed-forward layers. Similar to the ResNet architecture, we use the AAM-softmax as the final layer and train it with the cross-entropy loss criterion. The model uses 512 channels in 1-dimensional convolutional layers, 128 dimensions for SE-Block and attention, and a scaling factor of 8 for each Res2Block. The output dimension for feed-forward layers is set to 192, and the last feed-forward layer's dimension corresponds to the number of dialect classes.

3.3 Inference Scheme

In our model, we integrate a similarity measure with our learned classifiers to enhance classification performance (Lee et al., 2012; Nguyen et al., 2013; Roul and Arora, 2017). ResNet and ECAPA-TDNN are optimized for dialect identification via softmax, which we augment with a similarity-based measure based on the final embeddings produced by the network. For each dialect class, we randomly extract a cohort of 500 samples from the training set, and we calculate the average cosine similarity score between the test utterance and the cohort representing each class. After normalizing the scores, we combine them with the softmax scores by averaging them with equal weight (0.5) and selecting the class with the maximum score.

[1] https://github.com/microsoft/UniSpeech

4 Experimental setup

4.1 Datasets

We evaluate the dialect identification model on two Arabic dialect identification tasks: the MGB-3 ADI-5 dataset (Ali et al., 2017), and the fine-grained MGB5 ADI-17 dataset (Ali et al., 2019). ADI-5 training set consists of 13,825 utterances (53.6 hours), and the test and development sets consist of 1,524 (10 hours) and 1,492 (10 hours) utterances, respectively, with each set having approximately 2 hours of data per dialect class: Egyptian (EGY), Levantine (LAV), Gulf (GLF), North African (NOR), and Modern Standard Arabic (MSA). In ADI-17, approximately 3,000 hours of training data were labeled via distant supervision into 17 dialect classes using the origin country of the YouTube videos from which they were extracted. The testing and development sets contain ∼25 and ∼33 hours of speech, respectively, manually verified by human annotators.

4.2 Data Augmentation

For data augmentation, we apply additive noise drawn from the Music, Speech, and Noise corpus (MUSAN) (Snyder et al., 2015) and the QMUL impulse response dataset (Stewart and Sandler, 2010). We also apply speed perturbation, where the tempo is modified by factors of 0.9 and 1.1. All noise augmentation was implemented using the Kaldi toolkit (Povey et al., 2011).

4.3 Training settings

During the training phase, each model was initially trained with randomly selected 5-second segments from training utterances for the first 50 epochs. Subsequently, the duration of the training segments was reduced to 4 seconds for a total of 100 epochs to enable the model to generalize to short-duration utterances. All systems were trained using the Adam optimizer with a triangular learning scheduler policy and a batch size of 256.

5 Results

Tables 1 and 2 show the performance of our model variants in ADI-5 and ADI-17 test sets, respectively. *Fusion* refers to an ensemble model where scores from all four variants are combined, each with an equal weight of 0.25. We also show the performance of the best performing models from the original challenges, which have not been previously outperformed to the best of our knowledge.

Table 1: Performance evaluation on MGB-3 ADI-5 test set (in %) with baseline systems submitted to MGB-3 challenge. UniS denotes the UniSpeech-SAT feature extraction.

System	Features	Accuracy	Precision	Recall
Best systems from (Ali et al., 2017)				
MIT-QCRI	—	75.0	75.1	75.5
UTD	—	70.4	70.8	71.7
ResNet	MFCC	74.2	74.1	74.4
ECAPA	MFCC	75.3	75.1	75.3
ResNet	UniS	80.4	80.4	80.5
ECAPA	UniS	82.5	82.6	82.7
Fusion	—	**84.7**	**84.8**	**84.9**

Table 2: Performance evaluation on MGB-5 ADI-17 test set (in %) with baseline systems submitted to MGB-5 challenge. UniS denotes the UniSpeech-SAT feature extraction.

System	Features	Accuracy	Precision	Recall
Best systems from (Ali et al., 2019)				
DKU	—	94.9	94.9	94.9
UKent	—	91.1	91.1	91.1
ResNet	MFCC	90.1	90.1	90.1
ECAPA	MFCC	92.2	92.2	92.2
ResNet	UniS	95.7	95.7	95.7
ECAPA	UniS	96.1	96.1	96.2
Fusion	—	**96.9**	**96.9**	**96.9**

We observe consistent results in both datasets: ECAPA-TDNN network consistently outperforms ResNet, and the models using UniSpeech-SAT features consistently outperform those using MFCC features. Incorporating these pre-trained features results in 4% to 5% absolute improvement in accuracy for both models. We observe additional gains of 0.8% to 2% improvement in absolute accuracy by fusing all four model/feature combinations. The highest performance gain is observed by using UniSpeech-SAT features as input, which leads to outperforming all previous baselines.

6 Conclusions

This paper described variations of model architectures, namely ResNet and ECAPA-TDNN, employing two acoustic features: classical MFCCs and self-supervised UniSpeech-SAT, leading to state-of-the-art performance in two spoken Arabic dialect identification benchmarks: ADI-5, and ADI-17. UniSpeech-SAT features, which are extracted from a large pre-trained model optimized for acoustic and speaker variability, consistently demonstrated superior performance compared to MFCC features. Despite being pre-trained solely in English speech, UniSpeech-SAT illustrates transfer learning capability by extracting suitable feature representations for this discriminative task in the Arabic language. This may also indicate that non-linguistic acoustic variability (such as speaking tone, for example) could play a role in dialect identification. Consistent with previous models from the MGB-3 and MGB-4 challenge, fusing multiple models results in consistent improvements of overall performance.

7 Limitations

In this work, we limited our analysis and exploration to two network architectures and two types of acoustic features. We based our choice on observations from the current literature on dialect identification, speech classification, and self-supervised acoustic models. However, many additional features and architectural variations could have been explored, with additional detailed analysis of the different combinations. Furthermore, we did not analyze the acoustic features that are most discriminative in these datasets, which is a complex analysis that eludes us at this stage, but future work could explore more on which aspects of an utterance (linguistic, tonal, other) are most useful for dialect identification.

References

Ahmed M. Ali, Suwon Shon, Younes Samih, Hamdy Mubarak, Ahmed Abdelali, James R. Glass, Steve Renals, and Khalid Choukri. 2019. The mgb-5 challenge: Recognition and dialect identification of dialectal arabic speech. *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1026–1033.

Ahmed M. Ali, Stephan Vogel, and Steve Renals. 2017. Speech recognition challenge in the wild: Arabic mgb-3. *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 316–322.

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 *Interspeech 2022, 23rd Annual Conference of the International Speech Communication Association, Incheon, Korea, 18-22 September 2022*, pages 2278–2282. ISCA.

Ahmet Emin Bulut, Qian Zhang, Chunlei Zhang, Fahimeh Bahmaninezhad, and John H. L. Hansen. 2017. Utd-crss submission for mgb-3 arabic dialect identification: Front-end and back-end advancements on broadcast speech. *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 360–367.

Sanyuan Chen, Yu Wu, Chengyi Wang, Zhengyang Chen, Zhuo Chen, Shujie Liu, Jian Wu, Yao Qian, Furu Wei, Jinyu Li, and Xiangzhan Yu. 2021. Unispeech-sat: Universal speech representation learning with speaker aware pre-training. *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6152–6156.

Jiankang Deng, J. Guo, and Stefanos Zafeiriou. 2018. Arcface: Additive angular margin loss for deep face recognition. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4685–4694.

Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. 2020. Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification. *Interspeech*.

Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdel rahman 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.

Sameer Khurana, Maryam Najafian, Ahmed M. Ali, Tuka Al Hanai, Yonatan Belinkov, and James R. Glass. 2017. Qmdis: Qcri-mit advanced dialect identification system. In *Interspeech*.

Lam Hong Lee, Chin Heng Wan, Rajprasad Rajkumar, and Dino Isa. 2012. An enhanced support vector machine classification framework by using euclidean distance function for text document categorization. *Applied Intelligence*, 37:80–99.

Liam Lonergan, Mengjie Qian, Neasa Ní Chiaráin, Christer Gobl, and Ailbhe Ní Chasaide. 2023. Towards spoken dialect identification of irish. *arXiv preprint arXiv:2307.07436*.

Xiaoxiao Miao and Ian Mcloughlin. 2019. Lstm-tdnn with convolutional front-end for dialect identification in the 2019 multi-genre broadcast challenge. *ArXiv*, abs/1912.09003.

Tam T Nguyen, Kuiyu Chang, and Siu Cheung Hui. 2013. Supervised term weighting centroid-based classifiers for text categorization. *Knowledge and information systems*, 35:61–85.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely. 2011. The kaldi speech recognition toolkit.

In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society. IEEE Catalog No.: CFP11SRW-USB.

Rajendra Kumar Roul and Kushagr Arora. 2017. A modified cosine-similarity based log kernel for support vector machines in the domain of text classification. In *Proceedings of the 14th International Conference on Natural Language Processing (ICON-2017)*, pages 338–347.

Suwon Shon, Ahmed M. Ali, and James R. Glass. 2017. Mit-qcri arabic dialect identification system for the 2017 multi-genre broadcast challenge. *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 374–380.

David Snyder, Guoguo Chen, and Daniel Povey. 2015. Musan: A music, speech, and noise corpus. *arXiv preprint arXiv:1510.08484*.

David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. 2018. X-vectors: Robust dnn embeddings for speaker recognition. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5329–5333.

Rebecca Stewart and Mark Sandler. 2010. Database of omnidirectional and b-format room impulse responses. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 165–168. IEEE.

Peter Sullivan, AbdelRahim Elmadany, and Muhammad Abdul-Mageed. 2023. On the Robustness of Arabic Speech Dialect Identification. In *Proc. INTERSPEECH 2023*, pages 5326–5330.

Rong Tong, Bin Ma, Donglai Zhu, Haizhou Li, and Eng Siong Chng. 2006. Integrating acoustic, prosodic and phonotactic features for spoken language identification. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 1, pages I–I.

Karthika Vijayan, Haizhou Li, Hanwu Sun, and Kong Aik Lee. 2018. On the importance of analytic phase of speech signals in spoken language recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5194–5198.

Shu wen Yang, Po-Han Chi, Yung-Sung Chuang, Cheng-I Lai, Kushal Lakhotia, Yist Y. Lin, Andy T. Liu, Jiatong Shi, Xuankai Chang, Guan-Ting Lin, Tzu hsien Huang, Wei-Cheng Tseng, Ko tik Lee, Da-Rong Liu, Zili Huang, Shuyan Dong, Shang-Wen Li, Shinji Watanabe, Abdel rahman Mohamed, and Hung yi Lee. 2021. Superb: Speech processing universal performance benchmark. In *Interspeech*.

Shu-wen Yang, Po-Han Chi, Yung-Sung Chuang, Cheng-I Jeff Lai, Kushal Lakhotia, Yist Y Lin, Andy T Liu, Jiatong Shi, Xuankai Chang, Guan-Ting Lin, et al. 2021. Superb: Speech processing universal performance benchmark. *arXiv preprint arXiv:2105.01051*.

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

VoxArabica: A Robust Dialect-Aware Arabic Speech Recognition System

Abdul Waheed[λ,⋆] **Bashar Talafha** [ξ,⋆] **Peter Sullivan**[ξ,⋆]
AbdelRahim Elmadany[ξ] **Muhammad Abdul-Mageed**[ξ,λ]
[ξ] Deep Learning & Natural Language Processing Group, The University of British Columbia
[λ]Department of Natural Language Processing & Department of Machine Learning, MBZUAI
muhammad.mageed@ubc.ca

Abstract

Arabic is a broad language with many varieties and dialects spoken by $\sim$ 450 millions all around the world. Due to the linguistic diversity and variations, it is challenging to build a robust and generalized ASR system for Arabic. In this work, we address this gap by developing and demoing a system, dubbed VoxArabica, for dialect identification (DID) as well as automatic speech recognition (ASR) of Arabic. We train a wide range of models such as HuBERT (DID), Whisper, and XLS-R (ASR) in a supervised setting for Arabic DID and ASR tasks. Our DID models are trained to identify 17 different dialects in addition to MSA. We finetune our ASR models on MSA, Egyptian, Moroccan, and mixed data. Additionally, for the remaining dialects in ASR, we provide the option to choose various models such as Whisper and MMS in a zero-shot setting. We integrate these models into a single web interface with diverse features such as audio recording, file upload, model selection, and the option to raise flags for incorrect outputs. Overall, we believe VoxArabica will be useful for a wide range of audiences concerned with Arabic research. Our system is currently running at https://cdce-206-12-100-168.ngrok.io/.

1 Introduction

The Arabic language, with its diverse regional dialects, represents a unique linguistic spectrum with varying degrees of overlap between the different varieties at all linguistic levels (e.g., phonetic, syntactic, and semantic). In addition to Modern Standard Arabic (MSA), which is primarily used in education, pan-Arab media, and government, there are many local dialects and varieties that are sometimes categorized at regional (Zaidan and Callison-Burch, 2014; Elfardy and Diab, 2013; Elaraby and Abdul-Mageed, 2018), country (Bouamor et al., 2018; Abdul-Mageed et al., 2020a, 2021, 2022), or even province levels (Abdul-Mageed et al., 2020b). Historically, this wide and rich variation between different Arabic varieties has posed a significant challenge for automatic speech recognition (ASR) (Talafha et al., 2023; Alsayadi et al., 2022; Ali, 2020). The main focus has largely been on the recognition of MSA with very little-to-no focus on its dialects and varieties (Dhouib et al., 2022; Hussein et al., 2022; Ali et al., 2014). As such, ASR systems have conventionally been built either for MSA or individual dialects, thereby restricting their versatility and adaptability. However, the multifaceted nature of Arabic demands a robust ASR system that caters for its diverse dialects and varieties. In this work, we fill this research gap by introducing and demoing an ASR system integrated with a dialect identification model, dubbed *VoxArabica*.

VoxArabica is an end-to-end dialect-aware ASR system with dual functionality: (i) it offers a supervised dialect identification model followed by (ii) a finetuned Whisper Arabic ASR model covering multiple dialects. The dialect identification model works by assigning a country-level dialect, as well as MSA, from a set of 18 labels from input speech. This then allows the appropriate ASR model to fire. Contrary to traditional methodologies that separate dialect identification and speech recognition as two completely different tasks, our proposed pipeline integrates the two components effectively utilizing dialectal information for improved speech recognition. Such an integration not only improves the ASR output, but also establishes a framework aligned with the linguistic diversities inherent to Arabic as well. Concretely, our contributions can be summarized as follows:

- We introduce and demo our end-to-end VoxArabica system, which integrates dialect identification with state-of-the-art Arabic ASR.

[⋆]Equal contributions

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 441–449
December 7, 2023 ©2023 Association for Computational Linguistics

- Our demo is based on a user-friendly web interface characterized with rich functionalities such as audio uploading, audio recording, and user feedback options.

The rest of the paper is organized as follows: In Section 2, we overview related works. Section 3 introduces our methods. Section 4 offers a walkthrough of our demo. We conclude in Section 5.

2 Literature Review

Arabic ASR. Recent ASR research has focused on end-to-end (E2E) methods such as in Whisper (Radford et al., 2022) and the Universal Speech Model (Zhang et al., 2023). Such E2E deep learning models have significantly elevated ASR performance by allowing learning directly from the audio waveform, bypassing the need for intermediate feature extraction layers (Wang et al., 2019; Radford et al., 2022). Whisper is particularly noteworthy for its multitask training approach, incorporating ASR, voice activity detection, language identification, and speech translation. It has achieved state-of-the-art performance on multiple benchmark datasets such as Librispeech (Panayotov et al., 2015) and TEDLIUM (Rousseau et al., 2012). However, its resilience to adversarial noise has been questioned (Olivier and Raj, 2022).

For Arabic ASR specifically, the first E2E model was introduced using recurrent neural networks coupled with Connectionist Temporal Classification (CTC) (Ahmed et al., 2019). Subsequent works have built upon this foundation, including the development of transformer-based models that excel in both MSA and dialects (Belinkov et al., 2019; Hussein et al., 2022). One challenge for E2E ASR models is the substantial requirement for labeled data, particularly for languages with fewer resources such as varieties of Arabic. To address this, self-supervised and semi-supervised learning approaches are gaining traction. These models, such as Wav2vec2.0 and XLS-R, initially learn useful representations from large amounts of unlabeled or weakly labeled data and can later be finetuned for specific tasks (Baevski et al., 2020; Babu et al., 2021). W2v-BERT, another self-supervised model, employs contrastive learning and masked language modeling. It has been adapted for Arabic ASR by finetuning on the FLEURS dataset, which represents dialect-accented standard Arabic spoken by Egyptians (Chung et al., 2021; Conneau et al., 2023). Unlike Whisper, both Wav2vec2.0 and w2v-BERT necessitate a finetuning stage for effective decoding.

Arabic DID. Arabic DID has been the subject of a number of studies through recent years, enhanced by collection of spoken Arabic DID corpora such as ADI5 (Ali et al., 2017) and ADI17 (Shon et al., 2020). And advances in model architecture have mirrored changes in the larger LID research community, from i-vector (Dehak et al., 2010) based approaches (Ali et al., 2017) towards deep learning based approaches: x-vectors (Snyder et al., 2018; Shon et al., 2020), end-to-end classification using deep neural networks (Ali et al., 2019; Cai et al., 2018), and transfer learning (Sullivan et al., 2023).

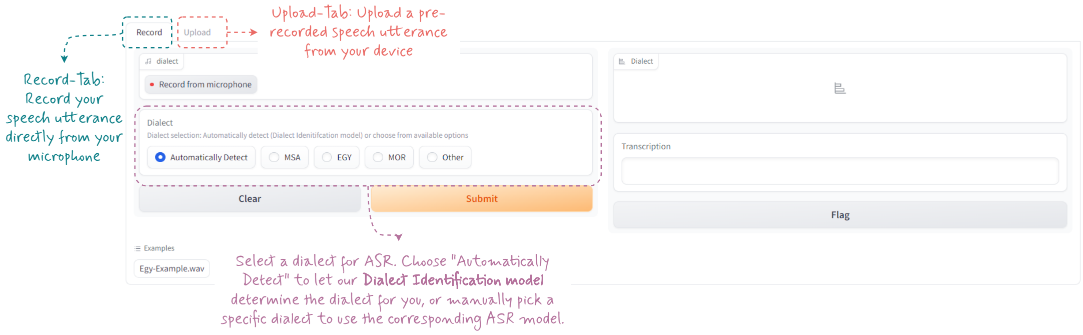
ASR and DID. Combining ASR and DID in a single pipeline remains fairly novel for Arabic. Recent works in this space has employed only limited corpora (Lounnas et al., 2020), or used ASR transcripts only to improve DID (Malmasi and Zampieri, 2017). Closest to our demonstrated system in this work is FarSpeech (Eldesouki et al., 2019), since it combines ASR and DID. However, FarSpeech is confined to coarse-grain DID and only supports MSA for ASR. In addition, compared to FarSpeech, our models are *modular* in that it allows users to run either or both ASR or DID, depending on their needs.

3 Models

3.1 DID Models

Our DID model is a transfer learning approach: finetuning HuBERT (Hsu et al., 2021) on ADI-17 (Shon et al., 2020) and the MSA portions of ADI-5 (Ali et al., 2017) and MGB-2 (Ali et al., 2016). We utilize only the MSA portions of ADI-5 due to the ambiguity of going from coarse-grain to fine-grain labels. Dialectal varieties covered in our model are *MSA, Algerian, Egyptian, Iraqi, Jordanian, Saudi, Kuwaiti, Lebanese, Libyan, Mauritanian, Moroccon, Omani, Palestinian, Qatari, Sudanese, Syrian, Emirati*, and *Yemeni*.

Training Details. Our finetuning procedure entailed performing a random search for training hyperparameters validated using the ADI-17 development set. A detailed overview of the hyperparameters searched can be found in Table 1 . We train using AdamW as optimizer, with a certain number of initial steps, *Freeze Steps*, where the original model is not updated and only the newly initialized classification layers change. After thawing, we also experiment with keeping some of the earlier layers



Upload-Tab: Upload a pre-recorded speech utterance from your device

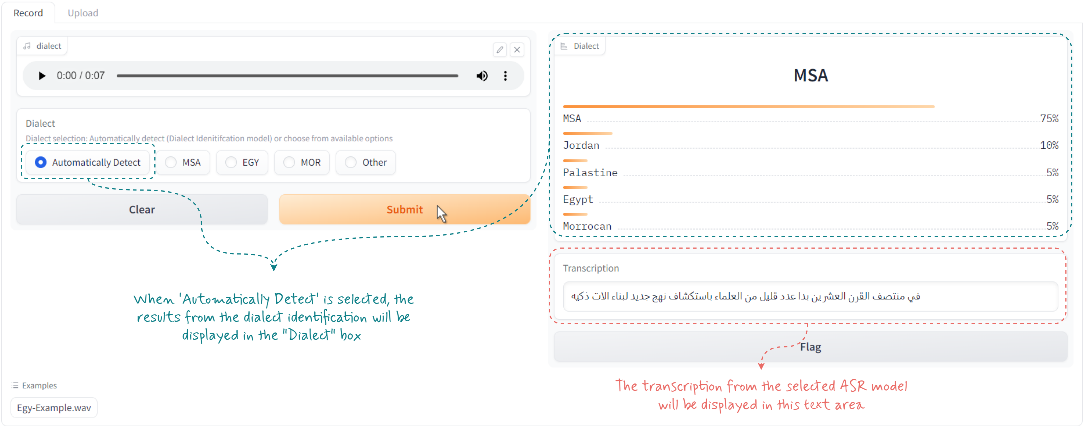
Record-Tab: Record your speech utterance directly from your microphone

Dialect selection: Automatically detect (Dialect Idenitifcation model) or choose from available options

● Automatically Detect ○ MSA ○ EGY ○ MOR ○ Other

Select a dialect for ASR. Choose "Automatically Detect" to let our **Dialect Identification model** determine the dialect for you, or manually pick a specific dialect to use the corresponding ASR model.

Figure 1: Users have the **option to either upload files or directly record their audio**. Additionally, the dialect can be automatically detected or manually selected for a specific ASR model.



Dialect selection: Automatically detect (Dialect Idenitifcation model) or choose from available options

● Automatically Detect ○ MSA ○ EGY ○ MOR ○ Other

Dialect

MSA

MSA 75%

Jordan 10%

Palastine 5%

Egypt 5%

Morrocan 5%

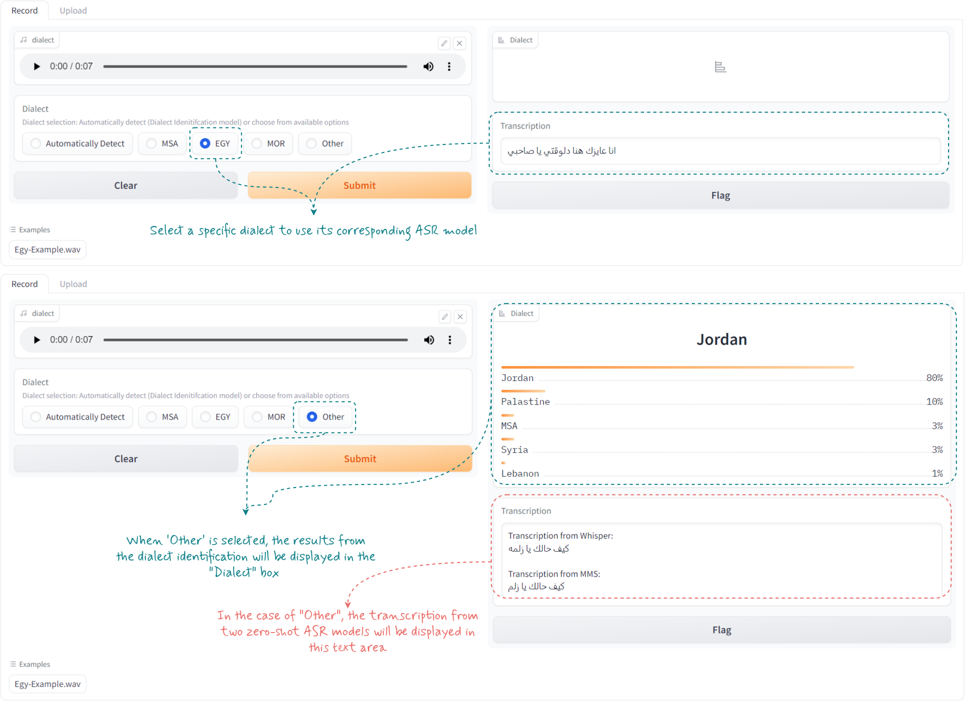
Transcription

في منتصف القرن العشرين بدا عدد قليل من العلماء باستكشاف نهج جديد لبناء الات ذكيه

When 'Automatically Detect' is selected, the results from the dialect identification will be displayed in the "Dialect" box

The transcription from the selected ASR model will be displayed in this text area

Figure 2: For **automatic dialect detection**, likelihood percentages determine the ASR model choice, with transcriptions displayed in the Transcription text area.



Dialect selection: Automatically detect (Dialect Idenitifcation model) or choose from available options

○ Automatically Detect ○ MSA ● EGY ○ MOR ○ Other

Transcription

انا عايزك هنا دلوقتي يا صاحبي

Select a specific dialect to use its corresponding ASR model

Dialect selection: Automatically detect (Dialect Idenitifcation model) or choose from available options

○ Automatically Detect ○ MSA ○ EGY ○ MOR ● Other

Dialect

Jordan

Jordan 80%

Palastine 10%

MSA 3%

Syria 3%

Lebanon 1%

Transcription

Transcription from Whisper:
كيف حالك يا زلمه

Transcription from MMS:
كيف حالك يا زلم

When 'Other' is selected, the results from the dialect identification will be displayed in the "Dialect" box

In the case of "Other", the transcription from two zero-shot ASR models will be displayed in this text area

Figure 3: When a specific dialect is manually selected, its associated **ASR model generates the transcription**. When recording in an unlisted dialect, select "Other". The dialect identification model will then detect the dialect, and both Whisper and MMS zero-shot models will produce the transcription.

Table 1: An overview of the search space of the hyperparmeter tuning for the DID model as well as optimal configuration found during the (n=30) random search. The batch size formula ensures our V100 GPUs were fully utilized during training, with a target of 75 seconds of audio regardless of the sampling duration. All values are picked from uniform distributions except for the learning rate, which was picked from a log uniform distribution.

	Range	**Conf.**
Batch Size	$4 \cdot \lfloor \frac{75}{Duration} \rfloor$	16
Freeze Steps	$[0, 1000]$	192
Learning Rate	$[1 \cdot 10^{-5}, 1 \cdot 10^{-2}]$	$6 \cdot 10^{-4}$
Max Steps	$[20k, 40k]$	29225
Duration	$[4, 18]$ seconds	4.69
Thaw Depth	$[0, 23]$	3

of the model frozen. We indicate the earliest layer that gets thawed as *Thaw Depth*. We also experiment with LayerNorm and Attention finetuning (Li et al., 2020), but our final model performed better without it.

3.2 ASR Models

We train a wide range of ASR models on a list of benchmark Arabic speech datasets. Our models include two versions of Whisper (Radford et al., 2022), *large-v2* and *small*. We also finetune XLS-R (Babu et al., 2022) for the ASR task. For MSA, we train our models on three versions of *common voice* (Ardila et al., 2019) datasets `6.1`, `9.0`, and `11.0`. We note that Talafha et al. (2023) show that Whisper *large-v2* outperforms its smaller variant as well as XLS-R trained on the same dataset. For Morrocan, Egyptian, and MSA, we fully finetune models on MGB2, MGB3, MGB5 (Ali et al., 2016, 2017, 2019). We also train ASR models on FLEURS (Conneau et al., 2023), which is accented Egyptian speech data.

Text Preprocessing. The datasets we employ exhibit various inconsistencies. For instance, within CV6.1, the utterance فَقَالَ لَهُمْ "faqaAla lahumo" is fully diacritic, whereas the utterance فإذا النجوم طمست "f<*A Alnjwm Tmst" lacks diacritic annotations, despite both originating from the Quran. Consequently, we adopt the normalization approach from (Chowdhury et al., 2021; Talafha et al., 2023), which involves: (a) discarding all punctuation marks excluding the % and @ symbols; (b) eliminating diacritics, Hamzas, and Maddas; and (c) converting eastern Arabic numerals into their western counterparts (e.g., 29 remains 29). Given that this study does not address code-switching, all Latin alphabet is excluded.

Ref (EGY)	مساء الخير اهلا ومرحبا بيكم في حلقة جديدة من برنامج بوضوح اي واحد نفسه في ثانية يطلعها قدام الكاميرا
Whisper (0-shot)	بسعي الخير أهلا ومرحبا بكم في حلقة جديدة من برنامج بوضوح أي واحد نفسه في ثانية يطلعها قدام الكاميرا
MMS (0-shot)	بساء لخير أهلن مرحباً بكم فخلى أجديدة من برنامج بوضوح أي واحد نفسفسنية يطلعها ودمك كمرة
Whisper (MSA)	بسعر الخير اهلا ومرحبا بكم في حلقة جديدة من برنامج بوضوح اي واحد نفسه سامي لا يطلعها قدم الكاميرا
Whisper(EGY)	مساء الخير اهلا ومرحبا بيكم في حلقة جديدة من برنامج بوضوح اي واحد نفسه في ثانية يطلعها قدام الكاميرا
Whisper(MOR)	مسايا الخير اهلا ومرحبا بيكم في حلقة جديدة من برنامج بوضوح اي واحد نفسو فسنية لي يطلعها قدام الكاميرا
XLS-R(MSA)	مساء الخير اهلا ومرحبا بكم في حالة جديدة من برنامج بوضوح اي واحد نفسه ثانية يطلعها قدام الكاميرا

Table 2: Example outputs produced by VoxArabica when input audio is Egyptian dialect.

Training Details. Before training, we apply preprocessing steps as mentioned above on the text. We train all of our models using AdamW optimizer (Loshchilov and Hutter, 2019) with a learning rate of $1e-5$, 500 warmup steps, and no weight decay. To prevent the model from severely overfitting, we employ early stopping with patience at 5. We use Huggingface trainer [1] with deepspeed ZeRO (Rajbhandari et al., 2019) stage-2 to parallelize our training across 8xA100 (40G) GPUs.

In our demo, we also allow users to utilize both Whisper and MMS (Pratap et al., 2023) in the zero-shot setting.

4 Walkthrough

Our demo consists of a web interface with versatile functionality. It allows users to interact with the system in multiple ways, depending on their needs.

User audio input. Users can either record their own audio through a microphone or upload a prerecorded file. In both cases, we allow different formats such as `.wav`, `.mp3`, or `.flac`, across various audio sampling rates (e.g., `16khz` or `48khz`). Figure 1 demonstrates the different options available to the user upon interacting with VoxArabica.

[1] `https://huggingface.co/docs/transformers/main_classes/trainer`

Model name	Dialect(s)	Dataset	Architecture
Whisper MSA	MSA	CV (6.1, 9.0, 11.0)	Whisper
XLS-R	MSA	CV (6.1, 9.0, 11.0)	Wav2vec 2.0
Whisper Morroco	MOR	MGB5	Whisper
Whisper Egypt	EGY	MGB3	Whisper
Whisper Zero-shot	-	-	Whisper
MMS	-	-	Wav2vec 2.0

Table 3: The utilized ASR models, their associated dialects, and respective architectures, and dataset used to train each model. Models marked with a dash are generic and not specific to a particular dialect.

Model selection. Users can choose to select an Arabic variety for transcription, or have it automatically detected using our 18-way DID system. We demonstrate this in Figure 2. Once the variety is detected, the corresponding ASR model will perform transcription and both DID transcription results will be presented on the interface (as shown in Figure 3). We offer various models: two for the EGY and MOR, respectively; two for MSA; and two generic models that can be used for any variety. We list all models in Table 3. In cases where predicted/selected variety is not covered by our ASR models, we fall back to our generic models (i.e., both Whisper zero-shot and MMS zero-shot).

User feedback. We also provide an option for users to submit *anonymous* feedback about the produced output by raising a flag. We use this information to collect high quality silver labels and discard examples where a flag is raised for incorrect outputs. It is important to note that we do not collect any external user data for any purpose, thus ensuring user privacy.

System output. Our system conveniently outputs both predicted Arabic variety and transcription across two panels as shown in Figure 3. For predicted variety, we show users all top five predictions along with model confidence for each of them. We provide outputs produced by our models in VoxArabica when the reference input is Egyptian dialect in Table 2. We also present additional examples in Appendix, Table 4.

5 Conclusion

We present a demonstration of combined DID and ASR pipeline to illustrate the potential for these systems to improve the usability of dialectal Arabic speech technologies. We report example outputs produced by our system for multiple dialects showcasing the effectiveness of integrated DID and ASR pipelines. We believe that our demo will advance the research to build a robust and generalized Arabic ASR system for a wide range of varieties and dialects and will enable a more holistic assessment of the strengths and weaknesses of these methods. For future work, we intend to add models for more dialects and varieties particularly those which are low resource.

6 Limitations

Audio classification tasks can be susceptible to out-of-domain performance degradation, which may impact real world performance. Similarly, studies on the interpretability of DID models have shown internal encoding of non-linguistic factors such as gender and channel (Chowdhury et al., 2020), which may impart bias to the models. Ensuring training corpora contain a diverse balance of speaker gender, recording conditions, as well as full coverage of the different styles of language is an ongoing challenge. We hope that by creating an online demonstration, these limitations can be further explored.

7 Ethics Statement

Intended use. We build a robust dialect identification and speech recognition system for multiple Arabic dialects as well as MSA. We showcase the capability of our system in the demo. We believe that our work will guide a new direction of research to develop a robust and generalized speech recognition system for Arabic. Through our demo, we integrate DID with ASR system which support multiple dialects.

Potential misuse and bias. Since our data is limited to a few dialects involved in finetuning DID and ASR systems, we do not expect our models to generalize all varieties and dialects of Arabic that are not supported by our models.

Acknowledgments

We gratefully acknowledge support from Canada Research Chairs (CRC), the Natural Sciences and Engineering Research Council of Canada (NSERC; RGPIN-2018-04267), the Social Sciences and Humanities Research Council of Canada (SSHRC; 435-2018-0576; 895-2020-1004; 895-2021-1008), Canadian Foundation for Innovation (CFI; 37771), Digital Research Alliance of Canada,[2] and UBC ARC-Sockeye.[3]

[2] https://alliancecan.ca

[3] https://arc.ubc.ca/ubc-arc-sockeye

References

Muhammad Abdul-Mageed, Chiyu Zhang, Houda Bouamor, and Nizar Habash. 2020a. Nadi 2020: The first nuanced arabic dialect identification shared task. *arXiv preprint arXiv:2010.11334*.

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, Houda Bouamor, and Nizar Habash. 2021. Nadi 2021: The second nuanced arabic dialect identification shared task. *arXiv preprint arXiv:2103.08466*.

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, Houda Bouamor, and Nizar Habash. 2022. Nadi 2022: The third nuanced arabic dialect identification shared task. *arXiv preprint arXiv:2210.09582*.

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, and Lyle Ungar. 2020b. Toward micro-dialect identification in diaglossic and code-switched environments. *arXiv preprint arXiv:2010.04900*.

Abdelrahman Ahmed, Yasser Hifny, Khaled Shaalan, and Sergio Toral. 2019. End-to-end lexicon free arabic speech recognition using recurrent neural networks. In *Computational Linguistics, Speech And Image Processing For Arabic Language*, pages 231–248. World Scientific.

Abbas Raza Ali. 2020. Multi-dialect arabic speech recognition. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE.

Ahmed Ali, Peter Bell, James Glass, Yacine Messaoui, Hamdy Mubarak, Steve Renals, and Yifan Zhang. 2016. The mgb-2 challenge: Arabic multi-dialect broadcast media recognition. In *2016 IEEE Spoken Language Technology Workshop (SLT)*, pages 279–284. 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.

Ahmed M. Ali, Hamdy Mubarak, and Stephan Vogel. 2014. Advances in dialectal arabic speech recognition: a study using twitter to improve egyptian asr. In *International Workshop on Spoken Language Translation*.

Hamzah A Alsayadi, Abdelaziz A Abdelhamid, Islam Hegazy, Bandar Alotaibi, and Zaki T Fayed. 2022. Deep investigation of the recent advances in dialectal arabic speech recognition. *IEEE Access*, 10:57063–57079.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M Tyers, and Gregor Weber. 2019. Common voice: A massively-multilingual speech corpus. *arXiv preprint arXiv:1912.06670*.

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 *Proc. Interspeech 2022*, pages 2278–2282.

Arun Babu, Changhan Wang, Andros Tjandra, Kushal Lakhotia, Qiantong Xu, Naman Goyal, Kritika Singh, Patrick von Platen, Yatharth Saraf, Juan Pino, et al. 2021. Xls-r: Self-supervised cross-lingual speech representation learning at scale. *arXiv preprint arXiv:2111.09296*.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460.

Yonatan Belinkov, Ahmed Ali, and James Glass. 2019. Analyzing phonetic and graphemic representations in end-to-end automatic speech recognition. *arXiv preprint arXiv:1907.04224*.

Houda Bouamor, Nizar Habash, Mohammad Salameh, Wajdi Zaghouani, Owen Rambow, Dana Abdulrahim, Ossama Obeid, Salam Khalifa, Fadhl Eryani, Alexander Erdmann, et al. 2018. The madar arabic dialect corpus and lexicon. In *Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018)*.

Weicheng Cai, Zexin Cai, Wenbo Liu, Xiaoqi Wang, and Ming Li. 2018. Insights in-to-end learning scheme for language identification. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5209–5213. IEEE.

Shammur A Chowdhury, Ahmed Ali, Suwon Shon, and James R Glass. 2020. What does an end-to-end dialect identification model learn about non-dialectal information? In *INTERSPEECH*, pages 462–466.

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.

Yu-An Chung, Yu Zhang, Wei Han, Chung-Cheng Chiu, James Qin, Ruoming Pang, and Yonghui Wu. 2021. W2v-bert: Combining contrastive learning and masked language modeling for self-supervised speech pre-training. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 244–250. IEEE.

Alexis Conneau, Min Ma, Simran Khanuja, Yu Zhang, Vera Axelrod, Siddharth Dalmia, Jason Riesa, Clara Rivera, and Ankur Bapna. 2023. Fleurs: Few-shot learning evaluation of universal representations of speech. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 798–805. IEEE.

Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet. 2010. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):788–798.

Amira Dhouib, Achraf Othman, Oussama El Ghoul, Mohamed Koutheair Khribi, and Aisha Al Sinani. 2022. Arabic automatic speech recognition: A systematic literature review. *Applied Sciences*, 12(17).

Mohamed Elaraby and Muhammad Abdul-Mageed. 2018. Deep models for arabic dialect identification on benchmarked data. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pages 263–274.

Mohamed Eldesouki, Naassih Gopee, Ahmed Ali, and Kareem Darwish. 2019. Farspeech: Arabic natural language processing for live arabic speech. In *INTERSPEECH*, pages 2372–2373.

Heba Elfardy and Mona Diab. 2013. Sentence level dialect identification in arabic. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 456–461.

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.

Amir Hussein, Shinji Watanabe, and Ahmed Ali. 2022. Arabic speech recognition by end-to-end, modular systems and human. *Computer Speech & Language*, 71:101272.

Xian Li, Changhan Wang, Yun Tang, Chau Tran, Yuqing Tang, Juan Pino, Alexei Baevski, Alexis Conneau, and Michael Auli. 2020. Multilingual speech translation with efficient finetuning of pretrained models. *arXiv preprint arXiv:2010.12829*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Khaled Lounnas, Hassan Satori, Mohamed Hamidi, Hocine Teffahi, Mourad Abbas, and Mohamed Lichouri. 2020. Cliasr: a combined automatic speech recognition and language identification system. In *2020 1st international conference on innovative research in applied science, engineering and Technology (IRASET)*, pages 1–5. IEEE.

Shervin Malmasi and Marcos Zampieri. 2017. Arabic dialect identification using iVectors and ASR transcripts. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 178–183, Valencia, Spain. Association for Computational Linguistics.

Raphael Olivier and Bhiksha Raj. 2022. There is more than one kind of robustness: Fooling whisper with adversarial examples. *arXiv preprint arXiv:2210.17316*.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE.

Vineel Pratap, Andros Tjandra, Bowen Shi, Paden Tomasello, Arun Babu, Sayani Kundu, Ali Elkahky, Zhaoheng Ni, Apoorv Vyas, Maryam Fazel-Zarandi, et al. 2023. Scaling speech technology to 1,000+ languages. *arXiv preprint arXiv:2305.13516*.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision. *arXiv preprint arXiv:2212.04356*.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2019. Zero: Memory optimization towards training A trillion parameter models. *CoRR*, abs/1910.02054.

Anthony Rousseau, Paul Deléglise, and Yannick Esteve. 2012. Ted-lium: an automatic speech recognition dedicated corpus. In *LREC*, pages 125–129.

Suwon Shon, Ahmed Ali, Younes Samih, Hamdy Mubarak, and James Glass. 2020. Adi17: A fine-grained arabic dialect identification dataset. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8244–8248. IEEE.

David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. 2018. X-vectors: Robust dnn embeddings for speaker recognition. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5329–5333. IEEE.

Peter Sullivan, AbdelRahim Elmadany, and Muhammad Abdul-Mageed. 2023. On the Robustness of Arabic Speech Dialect Identification. In *Proc. INTERSPEECH 2023*, pages 5326–5330.

Bashar Talafha, Abdul Waheed, and Muhammad Abdul-Mageed. 2023. N-shot benchmarking of whisper on diverse arabic speech recognition. *arXiv preprint arXiv:2306.02902*.

Dong Wang, Xiaodong Wang, and Shaohe Lv. 2019. An overview of end-to-end automatic speech recognition. *Symmetry*, 11(8):1018.

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

Yu Zhang, Wei Han, James Qin, Yongqiang Wang, Ankur Bapna, Zhehuai Chen, Nanxin Chen, Bo Li, Vera Axelrod, Gary Wang, Zhong Meng, Ke Hu, Andrew Rosenberg, Rohit Prabhavalkar, Daniel S. Park, Parisa Haghani, Jason Riesa, Ginger Perng, Hagen Soltau, Trevor Strohman, Bhuvana Ramabhadran, Tara Sainath, Pedro Moreno, Chung-Cheng Chiu, Johan Schalkwyk, Françoise Beaufays, and Yonghui Wu. 2023. Google usm: Scaling automatic speech recognition beyond 100 languages.

Appendix

Example Outputs

Ref (MSA)	يؤثر التدخين بشكل سلبي في جسم الانسان حيث ينتج عنه العديد من الاثار السلبية المؤذية للفرد وقد تؤدي بعضها الى مضاعفات تهدد الحياة
MMS	يؤثر التدخين بشكل سلبي هي جسم الإنسان حيث ينتجعنه العديد من الآثار السلبية المؤذيد الفرد وقد تؤدي بعضها إلى مضاعفات تهدد الحياة
Whisper(0-shot)	يؤثر التدخين بشكل سلبي في جسم الانسان حيث ينتج عنه العديد من الاثار السلبية المؤذية للفرد وقد تؤدي بعضها الى مضاعفات تهدد الحياة
Whisper(MSA)	يؤثر التدخين بشكل سلبي في جسم الإنسان حيث ينتج عنه العديد من الآثار السلبية المؤذية للفرد وقد تؤدي بعضها إلى مضاعفات تهدد الحياة
Whisper(MOR)	يؤثر التدخين بشكل سلبي في جسم الانسان حيت ينتج عنه العديد من الاثار السلبية المؤذية دالفرق وقد تؤدي بعضها الى مضعفات تهدد الحياة
Whisper(EGY)	يؤثر التدخين بشكل سلبي في جسم الانسان حيث ينتج عنه العديد من الاثار السلبية المؤذية للفرد وقد تؤدي بعضها الى مضاعفات تهدد الحياة
Ref (JOR - Other)	يا زلمة كيف حالك؟ شو أخبارك؟ وين هالغيبة؟ زمان عنك، ليش ما بتبين؟
MMS (0-shot)	يعزل كاف حلكشو أخبارك وانه الغاب زمان عنك لاش ما بتبين
Whisper (0-shot)	يا زلمة كيف حالك؟ شو أخبارك؟ وين هالغابة؟ زمان عنك، ليش ما بتبين؟
Whisper (MSA)	يا زلم كيف حالك شو اخبارك وانها الغيبة زمان عنك ليش ما بتبين
Whisper (MOR)	يا زلمة كيف حالك شو اخبارك وانها الغابة زمان عندك لاش مابتبين
Whisper (EGY)	يا زلمة كاف حالك شو اخبارك وانها الغابة اذا ما عنك ليش ما بتبين

Table 4: Outputs produced by VoxArabica when input is Egyptian and Jordanian. For Jordanian dialect, we do not have a finetuned model and Whisper (0-shot) performs best. Hence highlighting the lack of generalisation for various finetuned models to unseen dialects.

KSAA-RD Shared Task: Arabic Reverse Dictionary

Rawan Al-Matham[1*], Waad Alshammari[2*], Abdulrahman AlOsaimy[3], Sarah Alhumoud[4], Asma Al Wazrah[5], Afrah Altamimi[6], Halah Alharbi[7] and Abdullah Alfaifi[8]
King Salman Global Academy for Arabic Language (KSAA)
{[1]ralmatham, [2] walshammari, [3]aalosaimy, [4]salhumoud, [6]a.altamimi, [7]hmuneef, [8]aalfaifi}@ksaa.gov.sa

Abstract

This paper outlines the first KSAA-RD shared task, which aims to develop a Reverse Dictionary (RD) system for the Arabic language. RDs allow users to find words based on their meanings or definition. This shared task, KSAA-RD, includes two subtasks: Arabic RD and cross-lingual reverse dictionaries (CLRD). Given a definition (referred to as a "gloss") in either Arabic or English, the teams compete to find the most similar word embeddings of their corresponding word. The winning team achieved 24.20 and 12.70 for RD and CLRD, respectively in terms of rank metric. In this paper, we describe the methods employed by the participating teams and offer an outlook for KSAA-RD.

1 Introduction

A Reverse Dictionary (RD) is a type of dictionaries that allows users to find words based on their meanings or definitions. Unlike a traditional dictionary, where users search for a word by its spelling, a RD allow users to enter a description of a word or a phrase, and the RD will generate a list of words that match that description. RDs can be useful for writers, crossword puzzle enthusiasts, non-native language learners, and anyone looking to expand their vocabulary. Specifically, RD addresses the Tip-of-Tongue (TOT) phenomenon (Brown and McNeill, 1966), which refers to the situation where a person is aware of a word they want to say but is unable to express it accurately (Siddique and Sufyan Beg, 2019).

Various approaches have been proposed in the literature to develop RDs, including Information Retrieval (IR) System-based (Slaven et al., 2004; Crawford and Crawford, 1997; El-Kahlout and Oflazer, 2004; Shaw et al., 2013), Graph-based (Dutoit and Nugues, 2002; Reyes Magaña et al., 2019; Thorat and Choudhari, 2016), Mental Dictionary-based (Zock and Schwab, 2008; Zock and Bilac, 2004), Vector Space Model-based Semantic Analysis (Calvo et al., 2016; Méndez et al., 2013), and Neural Language Model-based approaches (Agrawal et al., 2021; Hedderich et al., 2019; Hill et al., 2016; Morinaga and Yamaguchi, 2018; Morinaga and Yamaguchi, 2020; Pilehvar, 2019; Qi et al., 2020; Yan et al., 2020; Zhang et al., 2020; Devlin et al., 2019).

However, to the best of our knowledge, there is no available Arabic RD system that allows the user to find the best matching word for a gloss in a specific dictionary, while most of the Arabic available digital dictionaries allow users to search for the definition by words (Siddique and Sufyan Beg, 2019).

We ran this shared task as a part of the first Arabic Natural Language Processing (ArabicNLP) conference collocated with EMNLP 2023, featuring the KSAA-RD (King Salman Global Academy for Arabic Language) with two subtasks: Arabic RD (Arabic to Arabic) and Cross-lingual Reverse Dictionary (CLRD) (Arabic to English). CLRD task aims to assist translating systems in selecting the best Arabic translation for new terms and definitions.

The dataset for both tasks used Arabic and English available dictionaries. Also, we provide manually annotated mapped dictionary between Arabic and English words to be used for supervised learning in the second task.

[*] Equal Contribution

December 7, 2023 ©2023 Association for Computational Linguistics

A total of four papers submitted for the shared task. Three teams surpassed the RD task baseline, while all four teams exceeded the CLRD task baseline. We provide a description of all submitted systems and the approaches they use. All the datasets created for this shared task are publicly available to support further research in a GitHub repository[1].

The rest of this paper is organized as follows: Section 2 defines related work that tackled the RD problem. Section 3 presents shared task description and the subtasks included in KSAA-RD. Section 4 describes the data given in the task. Section 5 presents the methodology that is used to evaluate the performance of the systems. Section 6 provides the baseline system and its results, in addition to discussing the participating systems and their results in the shared task. Section 7 draws conclusions.

2 Related work

Various approaches have been proposed to develop RD systems, including Information Retrieval (IR) System-based Approach, Graph-based Approach, Mental Dictionary-based Approach, Vector Space Model-based Semantic Analysis Approach, and Neural Language Model-based Approach. The four subsections provide a preview for each approach respectively.

2.1 Information Retrieval (IR) System-based Approach

The traditional IR systems retrieve a ranked list of the most relevant words, and it has a long-standing tradition in computational semantics. An earliest work addressing reverse dictionary by (Crawford and Crawford, 1997) is a patented work that uses synonyms to enhance search capabilities, and provide a broader range of relevant words based on user queries.

Rather than searching the actual word, a study from (Slaven et al., 2004) analyzes the descriptions of target words and convert them into a structured representation. This structured representation allows for efficient matching and retrieval of words that closely match the given descriptions.

Another work from (El-Kahlout and Oflazer, 2004) explores a lexical database for retrieving words based on their meanings. The method of extracting words based on their "meaning" involves comparing the user's definition with each entry in the Turkish database, without taking into account any semantic or grammatical information. The study from (Shaw et al., 2013) presents the development of a system that relies on a scalable database for efficient word retrieval. The system takes a user input phrase describing the desired concept and returns a word that satisfies the input phrase.

2.2 Graph-based Approach

A graph-based approach involves using a graph structure to represent the connections between words or concepts. This graph is built by considering semantic associations like synonyms, antonyms, hypernyms, and other related semantic links, to establish the relationships between the nodes representing the words or concepts. (Dutoit and Nugues, 2002) explores the connectivity and associations within the lexical database; by leveraging the graph structure to retrieve relevant words that align with the given definitions. Another approach focuses on utilizing the graph structure of a dictionary (Thorat and Choudhari, 2016) by investigating the sub-graph that surrounds each content word in a user query. They then prioritize and rank all the nodes encountered during the exploration, aiming to retrieve the most probable target word based on the given query. Another study from (Reyes Magaña et al., 2019) uses word association norms to establish semantic connections between words in the context of designing an electronic RD. The authors used the corpus of human-definitions and graph-based techniques, specifically a measure of betweenness centrality, to perform searches in the knowledge graph.

2.3 Mental Dictionary-based Approach

The mental dictionary-based approach depends on an individual's internal knowledge or mental lexicon to find words based on their meanings or descriptions. Instead of relying on external resources such as dictionaries or databases, this approach emphasizes using the individual's own mental representation of words and their

[1] https://github.com/Waadtss/ArReverseDictionary

associations. The study from (Zock and Bilac, 2004) proposes the concept of accessing words in an electronic dictionary by utilizing associations. This involves categorizing words based on the associations they evoke and identifying and labeling the most common or valuable associations within the dictionary. The proposal has been taken further in (Zock and Schwab, 2008) by implementing a user-guided search to the desired word that simulates human word synthesis, in order to gain a quick and intuitive access to that word.

2.4 Vector Space Model-based Semantic Analysis Approach

This approach attempts to use vector space models to transform the human-written queries into a vector by utilizing a semantic relations to improve the effectiveness of RD lookup. Another study that utilize semantic analysis with WordNet (Méndez et al., 2013) to generate vectors by identifying synsets that maximize a similarity measure. They then conduct a neighborhood search to extract the most relevant word. (Calvo et al., 2016) obtain vectors using LDA instead of WordNet.

2.5 Neural Language Model-based Approach

The neural language approach relies on encoding each input gloss into a vector representation, the output is a group of words whose embeddings are most similar to the corresponding gloss embedding. (Agrawal et al., 2021) enhance the traditional CBoW model by incorporating additional contextual information, such as word relationships and semantic associations, to better capture the nuances of word meanings. Another study that utilizes multi-sense embeddings (Hedderich et al., 2019) based on attention mechanism to enhance the representation of input queries in sentences. Focusing on eliminating the need for manually designed features, (Hill et al., 2016) propose Recurrent Neural Networks (RNN) to the RD task by encoding the definition of a word into a vector representation. The model then searches for the nearest neighbor word based on this vector. The performance was comparable to OneLook commercial RD. Another study from (Morinaga and Yamaguchi, 2018) improved the embedding accuracy by selecting better word vectors and employing category inference that eliminate irrelevant results using Convolutional Neural Network (CNN). An approach from (Morinaga and Yamaguchi, 2020) aim to better capture the nuances of a word meaning and tackle the problem of sufficient capacities by combining the bidirectional long short-term memory (BiLSTM) with Cascade Forward Neural Network (CFNN). A neural model from (Zhang et al., 2020) which is a multi-channel RD model (MRDM) that consists of BiLSTM and attention as sentence encoder. The model can help find the target words by utilizing four characteristic predictors that predict the POS, morphemes, word category and sememes. (Pilehvar, 2019) incorporating more fine-grained representations by adopting sense embeddings to disambiguate senses of polysemous target words.

The BERT models were incorporated in RD tasks as well. (Devlin et al., 2019) employs BERT that capture the bidirectional contextual information of words and sentences, allowing it to better comprehend the context and meaning of language.

(Qi et al., 2020) develop an online RD that enhanced multi-channel RD model from (Zhang et al., 2020). The model uses BERT instead of BiLSTM as a sentence encoder. Another model from (Yan et al., 2020) use BERT in both monolingual and cross-lingual RD system.

To the best of our knowledge, this shared task is the first to target Arabic RD problem and there is no available Arabic RD system that allows the user to find the best word in a specific dictionary, while most of the Arabic available digital dictionaries allow the users to search for the definition by words (Siddique and Sufyan Beg, 2019).

Based on the previous studies and approaches, the neural language model-based approach gives promising results compared to other approaches due to its ability to map word embeddings for an input definition into an embedding of the word defined by the definition using neural networks. Such a function encodes phrasal semantics and bridges the gap between them and lexical semantics. Therefore, we applied it in our baseline.

3 Task Description

This section describes the two subtasks in detail: RD and CLRD. The former converts Arabic word definitions into Arabic embeddings, while the latter converts English word definitions into Arabic embeddings.

3.1 Task 1: Reverse Dictionary

The structure of RDs (sequence-to-vector) is the opposite of traditional dictionaries lookup. This task focuses on the learning of how to convert human readable definitions into vector representation of the Arabic word.

id	ar.45
word	عين
POS	n
gloss	عضو الإبصار في الإنسان والحيوان.

(a) Example of definition in Arabic

```
{
  "id":"ar.45",
  "word":"عين",
  "gloss":"... عضو الإبصار في",
  "pos":"n",
  "electra":[0.4, 0.3, …],
  "sgns":[0.2, 0.5, …],
  "enId":"en.150",
 }
```

(b) Corresponding Arabic JSON snippet

id	en.150
word	eye
POS	n
gloss	One of the two organs in your face that are used for seeing

(c) Example of definition in English

```
{
  "id":"ar.45",
  "arword":"عين",
  "argloss":"عضو الإبصار في ...",
  "arpos":"n",
  "electra":[0.4, 0.3, …],
  "sgns":[0.2, 0.5, …],
  "enId":"en.150",
  "word":"eye",
  "gloss":"One of the two ...",
  "pos":"n",
}
```

(d) Mapped JSON dictionary between Arabic and English languages

Figure 1: The structure of a data point.

In this task, the input for the model is an Arabic word definition (gloss) and the output is the corresponding Arabic word embedding. For instance, given the Arabic gloss "المسير ليلًا," the model would generate an embedding for the Arabic word "الإسراء" which is the word corresponding to the gloss.

The task involves reconstructing the word embedding vector of the defined word, rather than simply finding the target word that is similar to the approach used by (Mickus et al., 2022; Zanzotto et al., 2010; Hill et al., 2016). This would enable the users to search for words based on the definition or meanings they anticipate.

The training data collection contains a source word vector representation “electra and sgns” and its corresponding word definition “gloss”, as illustrated in Figure 1 (a) and (b). The baseline model described in section 6.1 is designed to generate new word vector representations for the target unseen readable definitions.

3.2 Task 2: Cross-lingual Reverse Dictionary

The objective of the CLRDs task (sequence-to-vector) is to acquire the ability to transform readable definitions in the English language into a vector representation of the Arabic word. The main objective of this task is to identify the most accurate and suitable Arabic word vector that can efficiently express the identical semantic interpretation as the provided English language definition or gloss, which is commonly known as Arabicization "تَعْرِيب".

In this task, the input for the model is an English word definition (gloss) and the output is the Arabic word embeddings corresponding to the gloss. For instance, given the English gloss "Travelling at night," the model would generate an embedding for the Arabic word "الإسراء."

The task involves reconstructing the word embedding vector that represents the Arabic word to its corresponding English definition. This approach enables users to search for words in other languages based on their anticipated meanings or definitions in English. This task facilitates cross-lingual search, language understanding, and language translation.

The data collection includes the word, source word vector representation “electra and sgns”, and the definition “gloss” in both Arabic and English languages, as demonstrated in Figure 1 (d).

4 Data

This section discusses the data used in the shared task. The dataset includes two main components: the dictionary data, which is presented in section 4.1, and the word embedding vectors, which is presented in section 4.2. Section 4.3 describes further details of the dataset.

4.1 Dictionary data

To achieve the aim of the first task, known as RD, which seeks to develop a model capable of conducting reverse searches for Arabic words based on their meanings rather than their roots or lemmas, we utilized the Contemporary Arabic Language dictionary authored by Ahmed Mokhtar Omar (Omar, 2008). More specifically, we utilized the transferred version of this lexicon that adheres to the ISO standard, the Lexical Markup Framework (LMF) (Aljasim et al., 2022). It is worth mentioning that the KSAA team conducted this work. The dictionary relies on lemmas rather than roots, as discussed in the referenced study. The dataset comprises 58,000 words, commonly referred to as the lemmas that can have glosses with non-relevant information (e.g., morphological, and syntactic properties).

In the second task, our approach involved using a supplementary English dictionary, namely the English dictionary version employed in the SemEval 2022 Shared task on RD. It has a total of 63,596 lemmas that can have a different number of glosses (polysemy), and vice versa, a gloss can belong to more than one word (synonymy) (Mickus et al., 2022). This enabled us to construct a model that could effectively forecast the appropriate Arabic lemma matching to a given English meaning.

Consequently, there are two distinct datasets available: the dataset containing the Arabic dictionary and the dataset including the English dictionary. Each dataset consists of six components, including word form, part of speech, gloss, word ID, Electra embedding (Clark et al., 2020), and word2vec embedding (Mikolov et al., 2013). Within the realm of linguistic analysis, the “word” component encompasses the words. Additionally, the “part of speech” serves to denote the grammatical category to which the lemma belongs, namely noun, verb, adjective, adverb, or particle. The “gloss” serves the purpose of conveying the semantic content or meaning of a word, with the intention of excluding any phonetic, morphological, or syntactic aspects. The subsequent sections in the paper will provide explanations for the "electra” embedding and word2vec embedding components.

In order to fulfil the goals of the second task, we integrated the datasets in Arabic and English. The manual annotation procedure entailed a meticulous examination of the English gloss alongside its equivalent Arabic gloss, with the aim of attaining a thorough alignment between the two glosses across several linguistic dimensions.

To provide an instance, the English Dictionary defines the verb "cloud" as "to make obscure". This concept can be annotated with Arabic lemmas such as 'أغمض' which signifies the act of concealing, or 'أخفى', which denotes the act of covering, among other examples. The establishment of a correspondence between Arabic and English languages can be accomplished by assigning the Arabic gloss ID “id” to their corresponding English glosses “enId”. Note that a word can have other irrelevant glosses (or meanings), but they will not be assigned.

The dictionary annotation process employs a systematic approach to facilitate manual annotation. For each entry in the English dictionary, *deep-translator*[2] library is employed to provide word translation “*wt*”. Leveraging AraVec word embeddings (Soliman et al., 2017), the top ten similar word candidates are identified for *wt*. If any of these candidates align with lemmas in the Arabic dictionary, their corresponding IDs are integrated along with POS and gloss. The annotators then select the best candidate ID based on the corresponding POS and gloss that match the English dictionary.

In the manual phase, annotators meticulously select English lemmas, cross-referencing them with candidate IDs. This involves instances of confirmed matches, where corresponding Arabic lemma are included. In other cases, the process entails identifying the most suitable Arabic lemma translations within the Arabic dictionary and incorporating them. This meticulous process ensures data coherence, with lemma encompassing the word, POS, and gloss.

[2] https://pypi.org/project/deep-translator/

4.2 Embedding data

Our objective is to employ two distinct word embedding techniques, specifically contextualized word embedding and fixed word embedding. To efficiently attain this objective, we employ AraELECTRA (Antoun et al., 2021) for contextualized word embedding (referred to as “Electra”). AraELECTRA is an Arabic language representation model built upon the ELECTRA model. Unlike training a model to restore masked tokens, AraELECTRA focuses on training a discriminator model to distinguish original input tokens from replaced tokens, which have been substituted by a generator network. For single entries “***word***”, we use the token's embedding; for multi-token entries, we average the token embeddings. This approach leverages the substantial volumes of high-quality language models that have already been trained, enabling us to harness the contextualized representation of existing large pretrained models. For fixed word embedding (referred to as skip-gram with negative sampling “sgns”), we employ the AraVec skip-gram architecture from (Soliman et al., 2017).

During the word2vec embedding extraction, a two-step approach is applied.

1. For a single-token ***word***, a Unigrams skip-gram model is used to generate the embedding. When a ***word*** lacks representation (out-of-vocabulary) in the skip-gram model, the average embedding is obtained from the ***gloss*** associated with that word.
2. For a multi-token ***word***, an N-Grams skip-gram model is employed to generate the embedding. When a multi-token ***word*** lacks representation (out-of-vocabulary) in the skip-gram model, the average embedding is obtained from the ***gloss*** associated with that word.

When a gloss lacks representation in the skip-gram model, the embedding is obtained from the stemmed gloss. The stem of each word in gloss is extracted using CAMEL tool (Obeid et al., 2020). When representation remains elusive, the Farasa stemmer (Darwish and Mubarak, 2016) is employed instead of CAMEL. The two-step approach is then employed for generating embeddings. When there is no representation, the stemmer is then employed at the word level.

4.3 Dataset description

The datasets are provided in JSON format comprising nearly 58k Arabic entry data points and 63k English data entry points. The dataset is divided into three splits, including a training split that consists of almost 78% of the data points, a validation split that consists of 11% of the data points, and a test split that consists of 11% of the data points. Refer to Table 1 for data statistics.

Task	**Train**	**Dev**	**Test**
RD	45,200	6,400	6,410
CLRD	2,843	299	1,213

Table 1: Data Statistics.

5 Evaluation

The primary objective of our tasks is to find the most similar Arabic embedding for an Arabic or English definition. Thus, we consider three different approaches to measure vector similarity. The first approach is a Mean Squared Error (MSE), which calculates the average squared difference between the generated reconstructed embedding and target embeddings. The second approach is using the cosine similarity measure, where a perfect reconstructed embedding would result in a cosine similarity of 1 with the target embeddings. The challenge with the cosine measure is that language models utilizing the Transformer architecture can produce anisotropic output. Hence, it is not reasonable to use it alone to anticipate that two random contextualized embeddings will be orthogonal (Ethayarajh, 2019; Timkey and van Schijndel, 2021).

To complement the limitations of both MSE and cosine measure, a third approach known as the ranking metric has been utilized. The ranking evaluation metrics proposed in the CODWOE SemEval competition (Mickus et al., 2022). As shown in equation (1), the ranking metric is concerned with comparing and evaluating the proportion between the reconstructed embedding cosine p_i and the target embedding cosine t_i to the reconstruction embedding cosine p_i with all other targets embedding t_j in the test set. The proportion of targets with a higher correlation is determined by identifying the number of cosine values greater than $\cos(p_i, t_i)$ (Mickus et al., 2022). The ranking metric can be described as:

$$\text{Ranking}(p_i) = \frac{\sum_{t_j\,Test\,set} 1_{\cos(p_i,t_j)>\cos(p_i,t_i)}}{\#\,Test\,set} \quad (1)$$

To select the top-performing and well-rounded model, the submitted systems evaluation process follows a hierarchy of metrics. The primary metric is the ranking metric, which is used to assess how well the model ranks predictions compared to ground truth values. If models have similar rankings, the secondary metric, mean squared error (MSE), is considered. Lastly, if further differentiation is needed, the tertiary metric, cosine similarity, provides additional insights.

6 Shared Task Teams & Results

In this section, we present our baseline model and, participating teams, and results and description of submitted systems.

6.1 Our Baseline system

The baseline architecture proposed by (Mickus et al., 2022) is based on the Transformer model introduced by (Vaswani et al., 2017). The architecture involves feeding the input ***gloss***, which is represented as a sequence starting with a special token 'bos' and ending with another special token 'eos', into a straightforward Transformer encoder. The encoder generates hidden representations, which are then summed to produce the prediction. Additionally, a small non-linear feed-forward module is used to further refine the prediction. The evaluation of both tasks will be based on three different metrics including MSE, cosine similarity measure, and ranking metric.

6.2 Participating Teams

A total of 31 unique team registrations were received. A total of 39 valid submissions from 5 unique teams were received. During the testing phase, we received 5 submissions for the RD Subtask and 3 submissions for the CLRD Subtask from 4 different teams. You can find the details of these 4 teams in Table 2. Additionally, a total of 4 description papers were submitted and accepted.

Table 2: List of teams that participated

Team	Affiliation	Task
Rosetta Stone (ElBakry et al., 2023)	EPFL, Microsoft	1,2
UWB (Taylor, 2023)	University of West Bohemia	1,2
Qamosy (Sibaee et al., 2023)	Prince Sultan University	1
Abed (Qaddoumi, 2023)	NYU	1,2

6.3 Results and Description of Submitted Systems

Three teams participate in both RD and CLRD tasks, Rosetta Stone team, UWB teams, and Abed team. In the other hand, Qamosy team only submit the RD task. Results for both tasks are presented in Table 3 and Table 4.

The top team for both RD and CLRD tasks is the Rosetta Stone team (ElBakry et al., 2023). They employ an ensemble of fine-tuned Arabic BERT-based models, including camelBERT-MSA, camelBERT-Mix (Inoue et al., 2021), MARBERTv2 (Abdul-Mageed et al., 2021), and AraBERTv2 (Antoun et al., 2020). For RD, averaging the output embeddings from camelBERT-MSA and MARBERTv2 yielded a ranking of 24.20 using ELECTRA embeddings. For the CLRD task, they translate English glosses into Arabic, using the same models as in RD, achieving a rank of 12.70 with ELECTRA embeddings.

Qamosy team (Sibaee et al., 2023) methodology for RD task involves two phases: transforming the gloss into multidimensional vector representations using SBERT encoding, followed by training these vectors using the Simi-Decoder model. Their system achieved 2nd place in the RD task with a score of 28.10 in the Rank metric, utilizing ELECTRA embeddings.

Abed team (Qaddoumi, 2023) employs a modified multilingual BERT model for both RD and CLRD tasks, using data augmentation techniques like synonym replacement, random word insertion, deletion, and swapping in English, and random word deletion and swapping in Arabic. They achieved the 2nd and 3rd place in the RD and CLRD tasks, respectively, with a scores of 28.50 and 28.10 in the Rank metric with ELECTRA.

UWB teams (Taylor, 2023) utilize a rule-based approach for RD and CLRD tasks. They build a dataset-based dictionary and expand it using gloss. The dictionary-based approach with SGNS embeddings achieves 43.8 within RD task, lower than the baseline model, and 48.87 within the CLRD task.

It's evident that three teams utilizing BERT-based models outperformed our RD task Baseline, except for the dictionary-based approach by the UWB teams. However, the Rosetta Stone team's ensemble of different BERT architectures surpassed the performance of other methods.

Abed team demonstrated exceptional performance with ELECTRA embeddings, underscoring the substantial impact of data manipulation techniques on results. Surprisingly, the UWB teams, despite using a dictionary-based technique, outperformed our baseline that based on transformer model in the CLRD task.

7 Conclusion

In this paper, we present the first Arabic RD shared task, KSAA-RD, encompassing two subtasks: Arabic RD and Cross-Lingual RD, CLRD. The KSAA-RD task received 31 unique team registrations, resulting in 39 valid submissions and 4 submitted description papers. The outcomes from various teams underscore the persistent challenges posed by both RD and CLRD tasks, emphasizing the need for continued research in the field of Arabic RD tasks.

Our experience with KSAA-RD emphasize the significant impact of data manipulation techniques. Furthermore, employing an ensemble of diverse transformer architectures proved superior to other methods, highlighting the importance of model diversity in enhancing performance.

The Arabic dictionary used in this shared task is limited, compared to the newly released dictionary of the Arabic contemporary language: “Alriyadh Dictionary” (KSAA, 2023), which contains more than 120K terms compared to 58K and it is manually verified by groups of experts in the KSAA. Other aspects of future work include exploring advanced embedding techniques that might be more suitable to the semantic notion of the problem. Future work includes employing these techniques in a search engine and analyzing the user behavior of the search results. Further investigation is needed to examine whether dictionary definitions (which are usually written in a formal style) are a good representation of the users’ inquiries.

Table 3: Participants’ results for Reverse Dictionary Track (RD)

	Embedding	Dev			Test		
		Cos ↑	MSE ↓	Rank ↓	Cos ↑	MSE ↓	Rank ↓
Baseline 200 epoch	Sgns	35.61	5.03	38.52 (3)	40.58	4.49	36.28 (4)
	Electra	48.84	24.94	31.27 (3)	50.79	23.04	31.87 (4)
Rosetta Stone	Sgns	55.19	3.45	28.12 (1)	60.50	3.00	**25.40 (1)**
	Electra	63.65	16.14	21.44 (1)	64.50	15.20	**24.20 (1)**
Qamosy	Sgns	---	---	---	39.40	6.50	30.80 (3)
	Electra	18.90	54.80	50.00 (4)	51.90	23.60	28.10 (2)
Abed team	Sgns	49.45	3.48	31.45 (2)	53.80	3.10	29.10 (2)
	Electra	61.69	16.75	24.90 (2)	62.50	15.70	28.50 (3)
UWB	Sgns	---	---	---	37.50	5.17	43.80 (5)
	Electra	---	---	---	---	---	---

*The primary metric for the evaluation is the rank – the lower the rank, the better the model

Table 4: Participants’ Results for Cross-lingual Reverse Dictionary Track (CLRD)

	Embedding	Dev			Test		
		Cos ↑	MSE ↓	Rank ↓	Cos ↑	MSE ↓	Rank ↓
Baseline 300 epoch	Sgns	26.22	4.92	50.16 (3)	25.21	4.85	49.95 (4)
	Electra	54.09	22.10	36.22 (3)	51.66	23.81	40.72 (3)
Rosetta Stone	Sgns	38.74	4.84	37.15 (1)	40.00	5.30	**32.00 (1)**
	Electra	62.38	18.00	20.38 (1)	65.90	17.00	**12.70 (1)**
Abed team	Sgns	27.72	5.07	45.77 (2)	27.00	5.00	45.20 (2)
	Electra	58.06	19.55	25.88 (2)	56.50	20.60	28.10 (2)
UWB	Sgns	---	---	---	21.70	4.63	48.87 (3)
	Electra	---	---	---	---	---	---

*The primary metric for the evaluation is the rank – the lower the rank, the better the model

References

Muhammad Abdul-Mageed, AbdelRahim A. Elmadany, and El Moatez Billah Nagoudi. 2021. ARBERT & MARBERT: Deep Bidirectional Transformers for Arabic. *CoRR*, abs/2101.01785. arXiv: 2101.01785.

Aarchi Agrawal, Athulkumar R, K S Ashin Shanly, Kavita Vaishnaw, and Mayank Singh. 2021. Reverse Dictionary Using an Improved CBoW Model. In *Proceedings of the 3rd ACM India Joint International Conference on Data Science & Management of Data (8th ACM IKDD CODS & 26th COMAD)*, page 420, New York, NY, USA. Association for Computing Machinery.

Hawra Aljasim, Sultanah Alghurabi, Halah Alharbi, Abdulrahman Alosaimy, Afrah Altamimi, Asyah Almutawa, and Abdullah Alfifi. 2022. Toward an Automatic Semantic Order of Word Meanings Based on Lexically Tagged Corpus. *Manuscript submitted for publication*.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. AraBERT: Transformer-based Model for Arabic Language Understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France. European Language Resource Association.

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

Roger Brown and David McNeill. 1966. The “tip of the tongue” phenomenon. *Journal of Verbal Learning and Verbal Behavior*, 5(4):325–337.

Hiram Calvo, Oscar Méndez, and Marco A. Moreno-Armendáriz. 2016. Integrated concept blending with vector space models. *Computer Speech & Language*, 40:79–96.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. *CoRR*, abs/2003.10555. arXiv: 2003.10555.

H. Vance Crawford and Joel Crawford. 1997. Reverse electronic dictionary using synonyms to expand search capabilities.

Kareem Darwish and Hamdy Mubarak. 2016. Farasa: A New Fast and Accurate Arabic Word Segmenter. In *International Conference on Language Resources and Evaluation*.

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*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dominique Dutoit and Pierre Nugues. 2002. A lexical database and an algorithm to find words from definitions | Proceedings of the 15th European Conference on Artificial Intelligence. In *ECAI’02: Proceedings of the 15th European Conference on Artificial Intelligence*.

Ahmed ElBakry, Mohamed Gabr, Muhammad ElNokrashy, and Badr AlKhamissi. 2023. Rosetta Stone at KSAA-RD Shared Task: A Hop From Language Modeling To Word--Definition Alignment. In

˙Ilknur El-Kahlout and Kemal Oflazer. 2004. Use of Wordnet for Retrieving Words from Their Meanings. In *The Global Wordnet confrence*, pages 118–123.

Kawin Ethayarajh. 2019. How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings. 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 55–65, Hong Kong, China. Association for Computational Linguistics.

Michael A. Hedderich, Andrew Yates, Dietrich Klakow, and Gerard de Melo. 2019. Using Multi-Sense Vector Embeddings for Reverse Dictionaries. In *Proceedings of the 13th International Conference on Computational Semantics - Long Papers*, pages 247–258, Gothenburg, Sweden. Association for Computational Linguistics.

Felix Hill, Kyunghyun Cho, Anna Korhonen, and Yoshua Bengio. 2016. Learning to Understand Phrases by Embedding the Dictionary. *Transactions of the Association for Computational Linguistics*, 4:17–30.

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.

king Salman global academy for Arabic language KSAA. 2023. Alriyadh Dictionary.

Oscar Méndez, Hiram Calvo, and Marco A. Moreno-Armendáriz. 2013. A Reverse Dictionary Based on Semantic Analysis Using WordNet. In Félix Castro, Alexander Gelbukh, and Miguel González, editors, *Advances in Artificial Intelligence and Its Applications*, pages 275–285, Berlin, Heidelberg. Springer.

Timothee Mickus, Kees Van Deemter, Mathieu Constant, and Denis Paperno. 2022. Semeval-2022 Task 1: CODWOE – Comparing Dictionaries and Word Embeddings. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1–14, Seattle, United States. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. arXiv:1301.3781 [cs].

Yuya Morinaga and Kazunori Yamaguchi. 2018. Improvement of Reverse Dictionary by Tuning Word Vectors and Category Inference. In Robertas Damaševičius and Giedrė Vasiljevienė, editors, *Information and Software Technologies*, pages 533–545, Cham. Springer International Publishing.

Yuya Morinaga and Kazunori Yamaguchi. 2020. Improvement of Neural Reverse Dictionary by Using Cascade Forward Neural Network. *Journal of Information Processing*, 28:715–723.

Ossama Obeid, Nasser Zalmout, Salam Khalifa, Dima Taji, Mai Oudah, Bashar Alhafni, Go Inoue, Fadhl Eryani, Alexander Erdmann, and Nizar Habash. 2020. CAMeL Tools: An Open Source Python Toolkit for Arabic Natural Language Processing. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 7022–7032, Marseille, France. European Language Resources Association.

Ahmed Mukhtar Omar. 2008. Dictionary of Contemporary Arabic Language. *Volume One, 1st Edition, Cairo, Egypt*:2109.

Mohammad Taher Pilehvar. 2019. On the Importance of Distinguishing Word Meaning Representations: A Case Study on Reverse Dictionary Mapping. 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 2151–2156, Minneapolis, Minnesota. Association for Computational Linguistics.

Abed Qaddoumi. 2023. Abed at KSAA-RD Shared Task: Enhancing Arabic Word Embedding with Modified BERT Multilingual. In

Fanchao Qi, Lei Zhang, Yanhui Yang, Zhiyuan Liu, and Maosong Sun. 2020. WantWords: An Open-source Online Reverse Dictionary System. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–181, Online. Association for Computational Linguistics.

Jorge C. Reyes Magaña, Gemma Bel Enguix, Gerardo Sierra, and Helena Gómez-Adorno. 2019. Designing an Electronic Reverse Dictionary Based on Two Word Association Norms of English Language. In *Electronic lexicography in the 21st century. Proceedings of the eLex 2019 conference. 1-3 October 2019, Sintra, Portugal, 2019, págs. 865-880*, pages 865–880. Lexical Computing.

Ryan Shaw, Anindya Datta, Debra VanderMeer, and Kaushik Dutta. 2013. Building a Scalable Database-Driven Reverse Dictionary. *IEEE Transactions on Knowledge and Data Engineering*, 25(3):528–540.

Serry Sibaee, Samar Ahmad, Ibrahim Khurfan, Vian Sabeeh, Ahmed Bahaaulddin, Hanan M. Belhaj, and Abdullah I. Alharbi. 2023. Qamosy at KSAA-RD shared task: Semi Decoder Architecture for Reverse Dictionary with SBERT Encoder. In

Bushra Siddique and Mirza Mohd Sufyan Beg. 2019. A Review of Reverse Dictionary: Finding Words from Concept Description. In Manish Prateek, Durgansh Sharma, Rajeev Tiwari, Rashmi Sharma, Kamal Kumar, and Neeraj Kumar, editors, *Next Generation Computing Technologies on Computational Intelligence*, pages 128–139, Singapore. Springer.

Bilac Slaven, Wataru Watanabe, Taiichi Hashimoto, Takenobu Tokunaga, and Hozumi Tanaka. 2004. Dictionary search based on the target word description. In pages 556–559.

Abu Bakr Soliman, Kareem Eissa, and Samhaa R. El-Beltagy. 2017. AraVec: A set of Arabic Word Embedding Models for use in Arabic NLP. *Procedia Computer Science*, 117:256–265.

Stephen Taylor. 2023. UWB at KSAA-RD shared task: Computing the meaning of a gloss. In

Sushrut Thorat and Varad Choudhari. 2016. Implementing a Reverse Dictionary, based on word definitions, using a Node-Graph Architecture. In

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 2797–2806, Osaka, Japan. The COLING 2016 Organizing Committee.

William Timkey and Marten van Schijndel. 2021. All Bark and No Bite: Rogue Dimensions in Transformer Language Models Obscure Representational Quality. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4527–4546, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. *ArXiv*, abs/1706.03762.

Hang Yan, Xiaonan Li, Xipeng Qiu, and Bocao Deng. 2020. BERT for Monolingual and Cross-Lingual Reverse Dictionary. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4329–4338, Online. Association for Computational Linguistics.

Fabio Massimo Zanzotto, Ioannis Korkontzelos, Francesca Fallucchi, and Suresh Manandhar. 2010. Estimating Linear Models for Compositional Distributional Semantics. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1263–1271, Beijing, China. Coling 2010 Organizing Committee.

Lei Zhang, Fanchao Qi, Zhiyuan Liu, Yasheng Wang, Qun Liu, and Maosong Sun. 2020. Multi-Channel Reverse Dictionary Model. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(01):312–319.

Michael Zock and Slaven Bilac. 2004. Word Lookup on the Basis of Associations : from an Idea to a Roadmap. In *Proceedings of the Workshop on Enhancing and Using Electronic Dictionaries*, pages 29–35, Geneva, Switzerland. COLING.

Michael Zock and Didier Schwab. 2008. Lexical access based on underspecified input. In *Coling 2008: Proceedings of the Workshop on Cognitive Aspects of the Lexicon (COGALEX 2008)*, pages 9–17, Manchester, United Kingdom. Coling 2008 Organizing Committee.

UWB at KSAA-RD shared task: Computing the meaning of a gloss

Stephen Taylor
University of West Bohemia
taylor@ntis.zcu.cz

Abstract

To extract the 'meaning' of a gloss phrase, we build a list of sense-IDs for each word in the phrase which is in our vocabulary. We choose one sense-ID from each list so as to maximise similarity of all the IDs in the chosen subset. We take the meaning of the phrase in semantic space to be the weighted sum of the embedding vectors of the IDs.

1 Introduction

The KSAA Reverse Dictionary shared task is to find the embedding vector for a word, given a gloss, or short definition. The two sub-tasks use Arabic and English glosses.

The task is partly inspired by SemEval-2022 Task 1 (Mickus et al., 2022), which provided training data for English, Spanish, French, Italian, and Russion. For that task there was an additional subtask, generating a gloss from a word-embedding, which proved difficult to score; BLEU scores, which are typically used to measure the success of machine translations, seemed to perform poorly to compare the adequacy of generated glosses.

The individual records in the training data for the task provide: an ID, which corresponds to a particular sense of a word; the word; a gloss or definition; and two embedding vectors, a SGNS and an Electra vector. The SGNS vector is a word-based skipgram vector, so that each sense of the word has the same vector; the Electra vector is a transformer context-based vector, and each ID has a different vector.

Thus in all there are four subtasks: to find the SGNS or Electra semantic-vector from the Arabic or English words of the gloss. For neither language are there sufficient training data to completely populate the vectors of the gloss. There are several possible reasons for this problem. Considering just the problems for Arabic gloss phrases:

1. Some of the missing words are particles, which would probably appear in a list of stopwords; for these, the absence of a vector is a feature, not a problem.

2. Some missing words are due to the presence or absence of vowel and gemination marking. A correctly spelled Arabic word can appear with complete vocalization, with partial vocalization, or with no vocalization at all.

3. Several particles, including the prepositions ب , ف , the pronouns ه, ها , هم , كم , and the definite article ال , never stand alone, but always appear affixed to another word.

4. Arabic is an inflected language. Adjectives are inflected for gender, verbs for tense, number, person, and gender, nouns for number and case. Some of these inflections are regular, and others are not.

5. Some words just do not appear in the training data. The test words are unsurprising examples, but of course many others are also absent.

English has similar problems, but we spent more attention on Arabic.

An apparently obvious part of the solution might be to fine-tune a pre-trained transformer on the glosses, and then attempt to generate the word gloss by a transformation on the phrase embedding. This idea was used for baseline, and in the SemEval-2020 task by for example, (Li et al., 2022).

However, in the current task, we are restricted to less than fifty thousand training examples. Training a transformer on so little data seems problematic. Using an externally pre-trained transformer means bringing in external data, not in the spirit of a closed task.

We wanted to try a simple data-processing approach.

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 461–466
December 7, 2023 ©2023 Association for Computational Linguistics

Table 1: Data files and data

File name	# records	# IDs	# # words	mean(max) gloss words
`ar.train.json`	45200	45058	38498	7.03 (99)
`ar.en.train.json`	2862	2214	1697	11(65)
`en.complete.with.id.json`	63596	63596	26068	11.777 (129)

2 Data

We used the data files described in table 1, which were provided by the task organizers. The `ar.train.json` and the `dev` and `test` files are based on the LMF Contemporary Arabic Dictionary(Namly, 2015). The `en.complete.with.id.json` file is from the SemEval-2022 Task 1 data. Each entry in the `train` file is a definition for one sense of an Arabic word, with a short definition or gloss. The examples of usage found in the LMF dictionary, and sometimes part of the definition, have been dropped. Each of the IDs mentioned in Table 1 is a single sense. Although some senses are defined several times, only twenty percent of the words have more than one sense appearing in the `train` file. Although some definitions are quite long, one to three word definitions are quite common.

`ar.en.train.json` is used only to construct a cross-lingual transform(Artetxe et al., 2018; Brychcín et al., 2019) from the English embeddings of `en.complete.with.id.json` to the Arabic space of the `ar.ae.train.json` file. Since every entry in `ar.ae.train.json` has an `enId` attribute corresponding to an entry in `en.complete.with.id.json`, this does not change the amount of English data available for interpreting the gloss.

There are 35224 number of distinct tokens used for the English glosses, while the total English vocabulary of `en.complete.with.id.json` is only 26068 words. Doing a set subtraction, we see that the larger set contains many capitalized and inflected words, but also a number of words like *chat, majestic, dilemma, xanax, SiO2, inactivate*, that is, both technical terms, and relatively common words which happened not to be included in the dictionary at hand. Doing the subtraction the other way, we see that 15212 of our vocabulary words do not occur in the glosses.

We considered using a separate, larger English embedding, of which there are many available, with a cross-lingual transform, which could be easily prepared for the SGNS vectors based on common vocabulary. But it wasn't clear that 'open' as intended by the organizers included this option, and matching senses for the Electra embedding seems to be exactly the problem on which we are already working.

Similarly extending the Arabic vocabulary has the same problems, except that the organizers used the term 'closed' for subtask 1, which would clearly preclude doing it.

3 System

Our system[1] does not use a neural network. It uses `ar.train.json` for its Arabic vocabulary, and `en.complete.with.it.json` for its English vocabulary. It uses `ar.en.train.json`, which contains both Arabic and English words, in order to build a cross-lingual transform, so that the vectors built in the English space with English glosses can be converted to vectors in one of the Arabic spaces.

In addition to copying the `ar.train.json` sense dictionary, making a table of all the sense-IDs for each wordform, we also build a dictionary, `swords`, of derived values which points into that table. This includes several kinds of values:

- *Adjusted* Arabic words, with no vowels, only unmarked alifs ( ), all trailing yaas ( ) as alif maqsura ( ). This follows the convention adopted by Zahran et al. (Zahran et al., 2015a). This discards more information than probably necessary, but it works.

- Inflected verbs. The training files contain a part-of-speech field, and for one-word verbs we build *adjusted* inflected forms. Many of the verbs in the data come with indicated prepositions, given with an object pronoun. For these situations, we inflected the first of the two words in the definition. Sometimes this is *not* the verb, and as a result will never result in a meaningful match. We didn't build

[1] Code available at `github.com/StephenETaylor/KSAA-RD`

inflected nouns or adjectives during the test phase, nor did we consider one-letter prepositions, conjunctions or possessive or object pronouns.

- Single-word Arabic glosses. For this case we assume that the gloss is a near-synonym, and that the vocabulary is possibly increased by adding it as one.

Each of these substitute words points to one or more words for which we have a least one ID and embedding vector. [Although English offers similar possibilities, we did not build a substitution list for English before the end of the test period.]

3.1 Processing example

The processing loop for Arabic glosses builds a list of lists of possible IDs. For example, the first ’word’ in `ar.dev.json` is تحنَّنَ عليه . (Although there are two words here, this is an example of the data file following the dictionary practice of providing the correct preposition to use for this sense of the word.)

The gloss for this word gives three synonyms, ‘ترحّم، تعطّف عليه ورحمه’. Starting at the beginning, ‘ترحّم ’ is not in the vocabulary, but ‘ترحم ’ is in the swords list, with three possible vocabulary items, [‘ترحّم على صديقه ’ , ‘رحِم اللهُ فلانًا’ , ‘ رحِم يتيمًا’]. The first of these, ‘ترحّم على صديقه’ has two IDs, that is, two senses, [‘ar.19347’, ‘ar.19348’]. The second has the same two senses, and the third has the single sense [‘ar.19344’]. We combine these senses into a single list, and move to the second word of the gloss.

The second word is ‘تعطّف ’, which has neither a dictionary entry or an swords entry, so we append nothing to our list of gloss IDs.

Continuing to the third word of the gloss, ‘عليه ’, it is not found in the dictionary, but there is an swords entry, [‘ عليه دَم ’, ‘ عِلْيَة القوم ’, ‘ عِلِّيَّة ’], and these three entries each has a sense-ID, contributing in all [‘ar.16839’,‘ar.35831’,‘ar.35683’], so that the possible gloss senses so far are [[‘ar.19347’, ‘ar.19348’, ‘ar.19344’],[‘ar.16839’, ‘ar.35831’, ‘ar.35683’]].

The last word of the gloss, and the third synonym, is ‘ ورحمه ’, which doesn’t have a vocabulary entry, because it has both an object pronoun and a leading conjunction. It *should* have an swords entry, but we didn’t implement those features, so it contributes nothing to the possible gloss IDs.

The next step is to choose the most-compatible IDs. The routine in our system which does this is called `maxids()`, and it is the major bottleneck in processing. For this case, we would need to check only the cosines between nine pairs of possibilities, but our routine can efficiently handle more complex cases. Instead of enumerating all the possible sets of IDs, it randomly chooses a starting point from the cross product of the possibilities, and changing one ID at a time, greedily descends to a local minimum. It repeats this process up to a hundred times, and returns the least of the minima it has encountered, as well as a list of the values each ID contributes to the sum.

For this example, `maxids` returns ([‘ar.19348’, ‘ar.16839’], [0.98, 0.98]) so that the angle between the IDs is a bit less than $\pi/3$. (This angle is for the Electra embedding.) Since there are only two IDs, and one angle, both IDs contribute equally. We use the angles to build weights for the vectors, with larger angles getting smaller weights. In this case both weights are equal.

Finally, we add the weighted vectors for ar.19348 and ar.16839 and normalize the result; this is our approximation to the vector for the gloss, and thus the best guess at the vector for the original word. Since this was from the `dev` file, and not the `test` file, we can compare the guess to the correct vector.

4 Results

user	MSE	cos	rank
	Subtask 1 SGNS		
BASELINE	0.04922	0.26226	0.50167
bkhmsi	0.029 (1)	0.611 (1)	0.253 (1)
UWB	0.052 (2)	0.375 (3)	0.438 (3)
Ibraham Khurfan	0.065 (3)	0.394 (2)	0.308 (2)
SerrySibaee	- (4)	- (4)	- (4)
	Subtask 1 Electra		
BASELINE	0.22105	0.5409	0.36222
bkhmsi	0.150 (1)	0.649 (1)	0.226 (1)
Ibraham Khurfan	0.236 (2)	0.519 (2)	0.281 (2)
SerrySibaee	0.236 (2)	0.519 (2)	0.281 (2)
UWB	0.266 (3)	0.416 (3)	0.466 (3)
	Subtask 2 SGNS		
BASELINE	0.04922	0.26226	0.50167
UWB	0.046 (1)	0.217 (2)	0.489 (2)
bkhmsi	0.053 (2)	0.400 (1)	0.320 (1)
	Subtask 2 Electra		
BASELINE	0.22105	0.5409	0.36222
bkhmsi	0.170 (1)	0.659 (1)	0.127 (1)
UWB	0.266 (2)	0.479 (2)	0.452 (2)

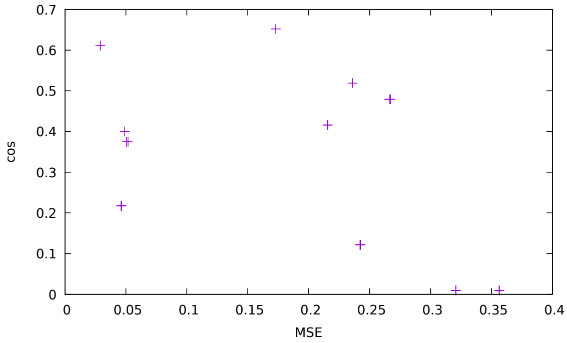


Figure 1: Scatter plot of cos vs MSE

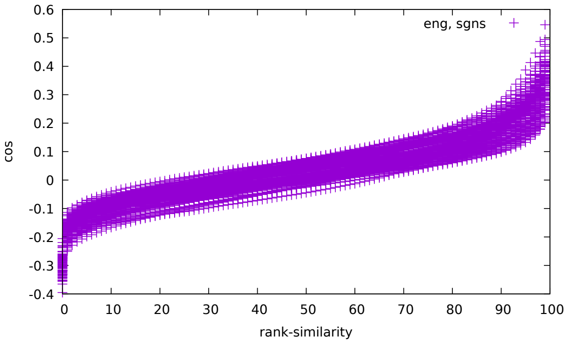


Figure 2: Scatter plot of cos versus rank-simlarity

Our system was second in rank [that is, last] for both SGNS and Electra on sub-task 2, and third and fourth on sub-task 1. It was slightly better than BASELINE on subtask 1, but not on subtask 2. Our Electra results are consistently worse than SGNS.

For individual normalized vectors, there is a straightforward relation between squared error and cosine:

$$SE = 2 - 2cos\theta \tag{1}$$

Where θ is the angle between vectors U and V and

$$SE = \sum_i (u_i - v_i)^2 \tag{2}$$

Neither SGNS nor Electra vectors were presented normalized, but the scoring code(AlShammari, 2023) shows the MSE computed on normalized vectors.

However, it is clear from our limited data that although MSE and mean cosine tend to move in opposite directions, systems with similar MSE can have very different mean cosines. See the graphed values for the leaderboard systems in Figure 1.

It's notable that MSE is generally lower for SGNS; a possible explanation is that the SGNS space has fewer vectors, so getting the sense wrong, but the word right, can happen more often.

Looking over our submissions, we deployed the vector weighting feature on August 20. Those runs were very slightly better than the runs on August 18, but typically only in the fourth or fifth digit of the rank measure.

5 Discussion

5.1 Similarity measures

The central idea in our system is to maximize the similarity of the senses in the gloss. Our measure of (dis)similarity, for each word-sense in the gloss, is the sum of the angles with all the other word-senses in the gloss.

We chose to minimize sum of angles, instead of maximize sum of cosines, because adding angles seems to make more intuitive sense than adding cosines. The best measure, and the one which would relate most closely to the rank-score on which the systems were measured, might be a rank-similarity, a measure of what fraction of the vocabulary is further from word-sense one than word-sense two is. Like cosine, this would have a best value of 1. We guessed that computing that rank would be quite a bit more expensive than computing the arc-cosine, and the maxids() routine which would call it is already the bottleneck in evaluating the gloss.

During the post-evaluation period, we tested precomputing tables of such ranks for each vector. Tables of 100 cosines, one at each percentile of rank, take up about a third of the amount of space used for the table of vectors. This seems like a reasonable amount of space; computing the values, which requires computing the cosine between all pairs of vectors, takes only about 17 minutes on a laptop, and can be done once and the (summarized, condensed) results saved to file. (The unsummarized results for 4.5E4 vectors would be 20E8 cosines or 8 GB as float32 values, an inconvenient size to cache.)

A scatter plot of cosine versus rank-similarity, showing the cached 100 points for 100 sample IDs, is shown in Figure 2. The graph illustrates that each vector has a slightly different curve, (although it seems possible that each curve could be described with a small number of parameters, probably much less than 100) and also hints at the fact that the rank-similarity is not symmetric: the rank-similarity between two vectors depends on the cosine between them, which is symmetric, and which vector you use for counting the neighbors, which is not symmetric. If we draw a horizontal line at any cosine value in the graph, it is clear that depending on which of the package of curves we stop, we can get a wide range of possible values for the rank-similarity. Averaging rank-similarity measured in both directions *would* give a symmetric measure.

Our results with the rank-similarity scheme were a little worse than with the angle scheme. One apparent problem is that the exciting part of the rank-similarity is in the last percentile, and our implementation uses linear interpolation, which is least accurate at the two edges. (Note the ends of the curves in Figure 2.) A second problem is that, like cosines, but unlike angles, this measure has less relative change as the limit of 1.0 is approached. So a value of 90, corresponding to 4500 close words, is within 10% of 99.9, corresponding to 45 close words, which is a much more interesting value. In contrast, the nearest neighbor is often at an angle of $\pi/6$, while an angle of $\pi/3$ is likely to include about 1% of the closest words. The angle difference is larger exactly where we want it to be.

5.2 English glosses versus Arabic glosses

We spent more effort on Arabic words than English ones. Possibly more effort on English might have improved the `swords` list and given better vocabulary coverage for English glosses. In any case, the Arabic results for our system are currently better than the English ones.

5.3 SGNS vs Electra

Our system performs much better for SGNS than for Electra. An important reason is that all the SGNS vectors for the senses of a word are the same, and so when we encounter a word in a gloss, we automatically get the sense vector right. Our sword scheme adds possible synonyms without regard for the context in which they are encountered. We *could* use the `maxlis()` scheme on glosses to choose sense-IDs before adding any sword entries based on them, if we did this on a second pass over the data. However, one of the benefits of the sword scheme is that those added synonyms give better gloss coverage, so a third pass, etc., might also be indicated.

5.4 Gloss coverage

A primary problem with our approach (and probably for everyone) is dictionary coverage of the glosses. Many words in the glosses are not present in the training data. Our *simplified and substitute words* (`swords`) list tries to deal with this problem by adding inflected forms and some words from the glosses as aliases for training words defined with vectors. Both of these seem like reasonable ideas, but at present we still drop about 50% of the gloss words. A more thorough and systematic approach to adding aliases might have increased our success rate.

Using the development data as extra training data for the test phase would probably also have helped.

6 Conclusion

Our earnest thanks go out to the organizers, who prepared a substantial dataset for this workshop.

Although our approach was not completely successful, it did better than the baseline for Subtask 1, Arabic definitions with SGNS vectors.

We have discussed several variations in Section 5, some of which might improve the system performance on the other three variations of the task.

We look forward to seeing designs of other workshop participants.

References

Carl Allen and Timothy Hospedales. 2019. Analogies explained: Towards understanding word embeddings. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 223–231. PMLR.

Waad AlShammari. 2023. ArReverseDictionary github site.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Tomáš Brychcín, Stephen Taylor, and Lukáš Svoboda. 2019. Cross-lingual word analogies using linear transformations between semantic spaces. *Expert Systems with Applications*, 135:287–295.

Alex Gittens, Dimitris Achlioptas, and Michael W. Mahoney. 2017. Skip-gram - Zipf + uniform = vector additivity. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 69–76, Vancouver, Canada. Association for Computational Linguistics.

Bin Li, Yixuan Weng, Fei Xia, Shizhu He, Bin Sun, and Shutao Li. 2022. LingJing at SemEval-2022 task

1: Multi-task self-supervised pre-training for multilingual reverse dictionary. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 29–35, Seattle, United States. Association for Computational Linguistics.

Timothee Mickus, Kees Van Deemter, Mathieu Constant, and Denis Paperno. 2022. Semeval-2022 task 1: CODWOE – comparing dictionaries and word embeddings. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1–14, Seattle, United States. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *Proceedings of the International Conference on Learning Representations*. arXiv1301.3781.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia. Association for Computational Linguistics.

Jeff Mitchell and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive Science*, 34:1388–1429.

Driss Namly. 2015. LMF contemporary arabic dictionary. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Stephen Taylor and Tomáš Brychcín. 2018. The representation of some phrases in arabic word semantic vector spaces. *Open Computer Science*, 8(1):182–193.

Mohamed A Zahran, Ahmed Magooda, Ashraf Y Mahgoub, Hazem Raafat, Mohsen Rashwan, and Amir Atyia. 2015a. Word representations in vector space and their applications for arabic. In *International Conference on Intelligent Text Processing and Computational Linguistics*, page 430–443. Springer.

Mohamed A. Zahran, Mohsen A. Rashwan, and Hazem Raafat. 2015b. Cross lingual lexical substitution using word representation in vector space. In *FLAIRS*.

Qamosy at KSAA-RD shared task: Semi Decoder Architecture for Reverse Dictionary with SBERT Encoder

Serry Sibaee
Serrytowork@gmail.com

Samar Ahmad
Samar.sass6@gmail.com

Ibrahim Khurfan
ibraheemkhurfan@gmail.com

Vian Sabeeh
Middle Technical University
viantalal@mtu.edu.iq

Ahmed Bahaaulddin
Middle Technical University
ahmedbahaaulddin@mtu.edu.iq

Hanan M. Belhaj
The Libyan Academy
h.belhaj@it.lam.edu.ly

Abdullah I. Alharbi
King Abdulaziz University
aamalharbe@kau.edu.sa

Abstract

A reverse dictionary takes a descriptive phrase of a particular concept and returns words with definitions that align with that phrase. While many reverse dictionaries cater to languages such as English and are readily available online or have been developed by researchers, there is a notable lack of similar resources for the Arabic language. This paper describes our participation in the Arabic Reverse Dictionary shared task. Our proposed method consists of two main steps: First, we convert word definitions into multidimensional vectors. Then, we train these encoded vectors using the SemiDecoder model for our target task. Our system secured 2nd place based on the Rank metric for both embeddings (Electra and Sgns).

1 Introduction

A reverse dictionary takes a phrase describing a specific concept as input and provides words whose definitions match that entered phrase. In contrast, a regular or common (forward) dictionary function contains word-to-meaning or definition mappings, which represents a useful solution for readers when encountering unfamiliar words in a text. For example, a forward dictionary would tell the user that (تحنن عليه - He felt compassion for him) means (ترحم، تعطف عليه ورحمه - have mercy on him) whereas a reverse dictionary allows the user to input the phrase (ترحم، تعطف عليه ورحمه - have mercy on him) and would likely produce the word(تحنن عليه - He felt compassion for him) along with other words having similar meanings as the output.

Reverse dictionaries offer significant practical value; primarily, they are highly effective in resolving the tip-of-the-tongue phenomenon which we encounter every day; people have difficulties finding the precise word to convey their thoughts, despite being on their tongue tip. As a result, they use phrases to explain the word or the concept. This challenge could stem from memory retrieval issues or a limited understanding of a particular language. Such a predicament is widespread, especially when someone is endeavoring to learn a new language and has a restricted number of vocabulary words or people who write frequently and seek a word that precisely matches their intended thought or expression.

Regarding natural language processing (NLP), reverse dictionaries serve various purposes. One of these is assessing sentence representation quality. Additionally, they prove advantageous in tasks related to text-to-entity mapping, such as question answering and information retrieval. Moreover, Reverse dictionaries consider not only the individual meanings of words but also how those meanings change when combined. Many words have synonyms, making determining the exact match for a given definition difficult. For instance, the input "to come together" could correspond to various options like "meet," "gather," "assemble," and more. Consequently, reverse dictionaries offer several potential word options rather than one possible word.

Numerous reverse dictionaries have been available online or have been created by researchers catering to different languages like English, Japanese, Turkish, French, and Persian. However, a noticeable absence of equivalent resources can be observed in the Arabic language. This shortage could stem from the lack of appropriate or substantial datasets containing words and their respective definitions which entails significant efforts in collecting and structuring language data. This paper, outlines our contribution to the Arabic Reverse Dictionary shared task. Our approach involves two

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 467–471
December 7, 2023 ©2023 Association for Computational Linguistics

phases: first, transforming word definitions into multidimensional vectors, and then training these vectors with the Simi-Decoder model for the intended task.

2 Related Work

Previously, researchers used a traditional approach for tackling the reverse dictionary problem, called semantic analysis using WordNet (Méndez et al., 2013). To determine how similar two words are, they made use of semantic similarity measurements. They used similarity between a word and an input phrase using a distance-based similarity measure. This measurement was considered necessary to determine connections between the term and the input words in the graph (Thorat and Choudhari, 2016). Recently, many researchers have been using embedding techniques in conjunction with neural networks and Deep learning(DL) to improve the generation of reverse dictionaries. Pilehvar (2019) used a combination of Bidirectional Long Short-Term Memory (BiLSTM) and cascade forward neural network (CFNN) to improve the neural reverse dictionary (NRD's) performance; outperforming a commercial reverse dictionary system (OneLook[1]) in various metrics. To find whether a proposed neural network framework is universally effective across all languages, Bendahman et al. (2022) used sequential models with a variety of neural networks, such as embedding networks, denser networks and Long Short-Term Memory LSTM networks. In (Chen and Zhao, 2022), the authors present a model that can be seen as a neural dictionary with two-way indexing and querying, embedding both words and definitions within a common semantic space. Their approach involves separate encoder and decoder networks for words and definitions. These networks are complemented by a shared layer that aligns them within the same representation space. In (Agrawal et al., 2021), they combine Continuous Bag-of-Words (CBOW) model and recurrent neural network (RNN) to employ a reverse dictionary that considers both word order and context. Another group of researchers focused on the concept of attention to better understand the context and meaning of the text. In (Hedderich et al., 2019), they used attention mechanisms to integrate multi-sense embedding using LSTM and contextual word embedding (Bidirectional Encoder Representations from Transformers (BERT) to enhance performance in the reverse dictionary task. As for (Malekzadeh et al., 2021), they utilised different models to simulate the functionality of a reverse dictionary. These included a Bag of Words (BOW) model, an RNN model with additive attention, and a BiLSTM model. Each of these was used to map a descriptive phrase to their corresponding words. Others (Qi et al., 2020; Zhang et al., 2020) used a sentence encoder based on a BiLSTM with an attention mechanism along with four characteristic predictors. These predictors assist in identifying the part-of-speech, morphemes, word category, and other relevant information.

3 Methodology

In this section, we will start describing the dataset used for our work. Then, we will explain our approach, divided into two primary steps. The first is to represent or encode the inputs (the definitions of the words) as multidimensional vectors. The second stage is to train the encoded inputs using a Simi-Decoder model for our downstream task.

3.1 Data Description

The used dataset is created and released by the shared task's organizers. They were chosen from the LMF Contemporary Arabic dictionary [2] and subsequently revised and refined by our annotation team. The total entries for all sets are 58,010 (Train: 45200, Dev: 6400 and Test: 6410). The datasets are in JSON format, comprising multiple examples. Each example within this dataset has six main elements. The "id" element indicates a language-specific unique identifier for a target "word". The "gloss" element provides a traditional dictionary definition, which is the source for the RD task. "enId" links to an identifier in the English dictionary. The remaining elements, namely "sgns" and "electra", represent different types of embeddings given as float arrays. Specifically, "sgns" relates to word2vec's skip-gram embeddings, while "electra" is tied to Transformer-based embeddings. Both can be targets in the RD task.

3.2 Encoder: encoding the input

In the first part of the work, we used the Sentence Transformer (SBERT) (Reimers and Gurevych, 2019) to represent the input (words' definitions). SBERT is a framework designed for generating

[1] https://www.onelook.com/thesaurus/

[2] https://lindat.cz/repository/xmlui/handle/11372/LRT-1944?locale-attribute=en

Model	NO.	h1	h2	h3	h4	Activation	Output	Dropout	Epochs
Electra	1	1024*6	1024*6	-	-	ReLU	256	-	100
	2	512*8	512*4	512*2	512	GELU	256	-	300
	3	512*8	512*4	512*2	512	GELU	256	-	1000
	4	512*8	512*4	512*2	512	GELU	256	-	2000
	5	512*8	512*6	512*4	512	GELU	256	0.65	4000
	6	512*8	512*4	512*2	512	LeakyReLU	256	-	2000
	7	512*8	512*6	512*4	512	GELU	256	0.60	1000
sgns	1	1024*6	1024*6	-	-	ReLU	300	-	100
	2	512*8	512*4	512*2	512	GELU	300	-	1000

Table 1: Arciticture Semi-Decoder MLP

fixed-length embeddings for sentences, optimizing for semantic similarity and efficiency over traditional BERT models. It is optimized for processing multiple sentences simultaneously, ensuring faster results. Its training structure prioritizes semantic similarity, meaning similar sentences have close vector representations. Furthermore, SBERT offers a range of pre-trained models for different tasks and languages, including Arabic.

Using SBERT in our task, the size of the encoded inputs is (d=512) for every definition of the words. This approach helped to make the training easier and more efficient by not worrying about the input size. We used The 'distiluse-base-multilingual-cased' model, which has proven effective in generating dependable embeddings across multiple languages (Reimers and Gurevych, 2020), making it an ideal choice for our focus on Arabic. The output of this step is encoded inputs that will be passed to the next stage, as can be seen in Figure 1.

3.3 Decoder: Semi-Decoder MLP

In the second part of the work, we use many Multi-Layer Perceptron(MLP) architecture (summarized in Table 1 as a decoder model to transform the inputs with the outputs (which have two dense vector dimensions for them (Electra = 265 d) and (sgns = 300 d). Due to the limited time and resources, We started our experiments with Electra embeddings, and then we selected the best-performing architecture to employ them with Sgns embeddings. The semi-decoder is a Deep Neural Network with four hidden layers where the first hidden layer has 8 times the input, the second has 4 times, the third has 2 times, and the fourth has the same as the input that will be projected to the size of the output. The dropout mentioned rate is between every hidden layer before the activation.

Figure 1 illustrates the main idea behind our proposed framework. The process can be explained mathematically as follows:

$$F : x_{(input)} \rightarrow \widehat{y}_{(output)} \tag{1}$$

where the $x_{(input)} \in R^{512}$ and the $\widehat{y}_{(output)} \in R^{o}$ where $o \in \{256, 300\}$ the dimensions of the two types of outputs (electra and sgns). F is a neural network with 4 hidden layers (this is the defult design while we also trained two networks with 2 hidden layers). x is the input vectors and $\hat{y}$ is the predicted vector (d 256 or 300)

$$E(t) = x \tag{2}$$

E is a function representing the encoder model and t is the tokens, where E will do the following:

1. tokenize the text
2. feed the encoder the tokens IDs
3. output $\sum_{i}^{i=n}(t_i)$ (max polling where n is the maximum number of tokens and if less it will be padded

$$H_i^{512\times512*m} \tag{3}$$

H is the size of hidden layer.

$$m \in \{8, 4, 2, 1\} \tag{4}$$

The x_{input} is output of a pre-trained encoder model called "SBERT" as follow:

Then the semi-Decoder:

$$D(x) = \widehat{y} \tag{5}$$

D is a function representing the semi-decoder model

then the loss function $L(\widehat{y}, y)$ which is $MSE(\widehat{y}, y)$ will update the weights of the network

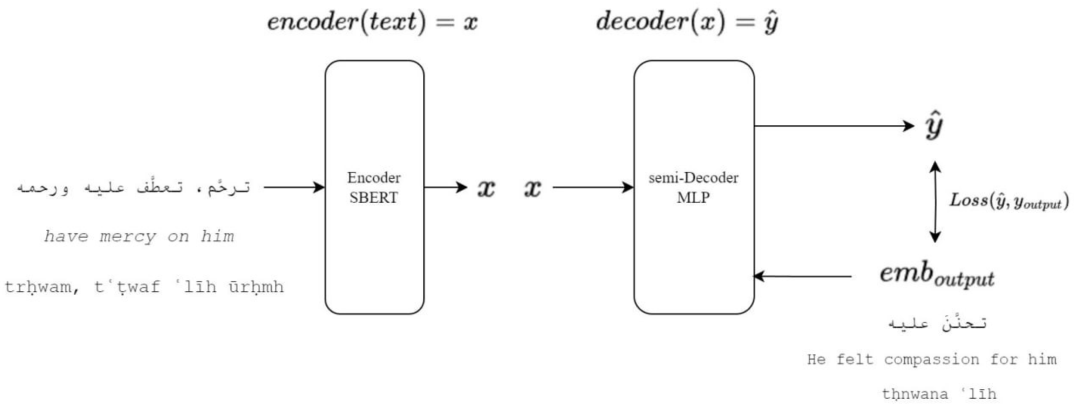


Figure 1: An example of a word's definition as input and a target word as output to show the overview of our proposed framework.

4 Result

We began our initial experiments with the development dataset. Afterward, we selected the most efficient method to produce predictions for the test dataset. Our evaluation relied on the official evaluation metric supported by the event organizers: Mean Square Error (MSE), Cosine similarity, and Rank, which evaluates the model's ability to order predictions in relation to actual values. The primary evaluation metric is Rank. If models yield similar results based on this metric, the mean squared error (MSE) is then employed as a secondary measure. In cases where further differentiation is required, cosine similarity serves as the tertiary metric.

To clarify, we experimented with seven different architecture setups to train the model, taking into account the number of embedding layers and drop-off. We chose the best model on Electra (Model NO. 3) to apply them to Sgns, in addition to the baseline model (Model NO. 1). Table 2 presents the results on the test dataset. It can be seen that Model number 3 has the highest performance (Rank = 28.05%) followed by the models 7 and 4. As for Sgns embeddings, model 2 achieved the best result with Rank of 30.78%.

The concept behind our approach is modifying the standard encoder-decoder architecture by truncating its latter section, which we have called the 'semi-decoder'. Due to the extensive scale of our model, an epoch range of 100-300 was inadequate for training. When the epochs exceeded 2000, overfitting issues emerged with our test data. This observation led us to conclude that the optimal epoch range is between 1000 and 2000. Specifically, the 2000-epoch mark resulted in a 'semi-overfitting' situation that delivered the most promising outcomes.

Model No.	MSE	COS SIM	Rank
		Electra	
1	18.89%	54.83%	50.00%
2	26.59%	21.91%	50.01%
3	23.56%	51.94%	**28.05%**
4	**17.03%**	**59.08%**	33.31%
5	17.85%	55.57%	48.15%
6	74.20%	-7.98%	37.97%
7	32.33%	46.07%	28.92%
		Sgns	
1	6.59%	21.90%	50.01%
2	**6.50%**	**39.36%**	**30.78%**

Table 2: Performance results for different models on the test set using three evaluation metrics.

4.1 Error Analysis

During the training process, a notable range of effective epochs emerged, spanning from 300 to 2000, wherein discernible patterns were successfully learned. Preceding this pivotal interval, the model's proficiency in capturing intricate patterns appeared limited. However, the subsequent epochs saw an escalated tendency towards overfitting. The employment of the GELU activation function exhibited superior performance. Conversely, the ReLU activation function demonstrated commendable potential for generalization, specifically in contexts characterized by diverse conditions ("sgns"). Nonetheless, for ranking tasks, its efficacy appeared akin to a stochastic outcome. Conversely, the Leaky ReLU activation function exhibited a subdued impact, potentially owing to the specificity of the problem domain. Notably, the application of dropout regularization yielded moderate influence on the model's performance. The chosen model architecture, designed to encapsulate definitions, demonstrated inherent promise, warranting a finer calibration to further explore the nuances of the Arabic language.

5 Conclusion

Our methodology encompasses two fundamental stages. Initially, we encode the word definitions, translating them into multidimensional vector representations. Subsequently, we subject these encoded vectors to training via the Simi-Decoder model to address our designated task. Our system secured a 2nd place based on Rank metric for both embeddings (Electra and Sgns).

Future work could involve collecting more data for training or validation, or providing the service online for public access. Improving our model's performance might be achieved by adopting the BERT or transformer model for training, known for efficient parallel processing and capturing long-term dependencies.

References

Aarchi Agrawal, KS Ashin Shanly, Kavita Vaishnaw, and Mayank Singh. 2021. Reverse dictionary using an improved cbow model. In *Proceedings of the 3rd ACM India Joint International Conference on Data Science & Management of Data (8th ACM IKDD CODS & 26th COMAD)*, pages 420–420.

Nihed Bendahman, Julien Breton, Lina Nicolaieff, Mokhtar Boumedyen Billami, Christophe Bortolaso, and Youssef Miloudi. 2022. Bl. research at semeval-2022 task 1: Deep networks for reverse dictionary using embeddings and lstm autoencoders. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 94–100.

Pinzhen Chen and Zheng Zhao. 2022. A unified model for reverse dictionary and definition modelling. *AACL-IJCNLP 2022*, page 8.

Michael A Hedderich, Andrew Yates, Dietrich Klakow, and Gerard de Melo. 2019. Using multi-sense vector embeddings for reverse dictionaries. In *Proceedings of the 13th International Conference on Computational Semantics-Long Papers*, pages 247–258.

Arman Malekzadeh, Amin Gheibi, and Ali Mohades. 2021. Predict: persian reverse dictionary. *arXiv preprint arXiv:2105.00309*.

Oscar Méndez, Hiram Calvo, and Marco A Moreno-Armendáriz. 2013. A reverse dictionary based on semantic analysis using wordnet. In *Advances in Artificial Intelligence and Its Applications: 12th Mexican International Conference on Artificial Intelligence, MICAI 2013, Mexico City, Mexico, November 24-30, 2013, Proceedings, Part I 12*, pages 275–285. Springer.

Mohammad Taher Pilehvar. 2019. On the importance of distinguishing word meaning representations: A case study on reverse dictionary mapping. 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 2151–2156.

Fanchao Qi, Lei Zhang, Yanhui Yang, Zhiyuan Liu, and Maosong Sun. 2020. Wantwords: An open-source online reverse dictionary system. *EMNLP 2020*, page 175.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Sushrut Thorat and Varad Choudhari. 2016. Implementing a reverse dictionary, based on word definitions, using a node-graph architecture. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2797–2806.

Lei Zhang, Fanchao Qi, Zhiyuan Liu, Yasheng Wang, Qun Liu, and Maosong Sun. 2020. Multi-channel reverse dictionary model. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 312–319.

Abed at KSAA-RD Shared Task: Enhancing Arabic Word Embedding with Modified BERT Multilingual

Abed Qaddoumi
Independent, NYC, NY, 11216
amq259@nyu.edu

Abstract

This paper presents a novel approach to the Arabic Reverse Dictionary Shared Task at ArabicNLP 2023 by leveraging the Bidirectional Encoder Representations from Transformers (BERT) Multilingual model and introducing modifications augmentation and using a multi attention head. The proposed method aims to enhance the performance of the model in understanding and generating word embeddings for Arabic definitions, both in monolingual and cross-lingual contexts. It achieved good results compared to benchmark and other models in the shared task 1 and 2.

1 Introduction

The Arabic Reverse Dictionary Shared Task at ArabicNLP 2023 poses unique challenges in generating word embeddings from definitions, especially in a cross-lingual setting. While traditional models have shown promise, the complexity of the Arabic language and its rich morphological structure necessitates advanced techniques. This paper introduces a modified BERT Multilingual model, incorporating changes augmentation and using a multi attention head, to address these challenges.

This paper describes the system used for the Arabic Reverse Dictionary Shared task at ArabicNLP 2023. The task was released for the the first based on the SemEval 2022 Shared Task #1: Comparing Dictionaries and Word Embeddings (CODWOE) (Mickus et al., 2022) but for the Arabic language. Competition results highlight two main trends:

1. Baseline architectures still perform competitively against new participant solutions.

2. The overall scores, especially in the definition modeling track, are unsatisfactory.

Participants identified challenges such as subpar data quality, small training corpora, and mainstream natural language generation (NLG) metrics' limited relevance. Teams have experimented with Transformer, Recurrent Neural Networks (RNN), and Convolutional neural networks (CNN) models and found success with multi-task training. There's no single architecture that stands out as the best, with some evidence suggesting that Transformers may not be ideal for this task (Mickus et al., 2022). For future research, the focus should be on enhancing dataset size and quality and re-evaluating metrics. The competition has spotlighted a variety of natural language processing (NLP) models and approaches, underscoring the field's dynamic nature.

For the Arabic Reverse Dictionary Shared task at ArabicNLP 2023, our primary objective was to assess the competitiveness of our current BERT Multilingual Cased implementation in the context of comparing dictionaries and embeddings. Additionally, we endeavored to incorporate a data augmentation strategy to enhance our results. Remarkably, our experiments with data augmentation yielded significant improvements in the development set results. The augmentation techniques employed were relatively basic, involving operations such as word addition, deletion, and swapping within sentences. Due to the limited size of the available data, even with augmentation, our training was restricted to just two epochs. Despite these constraints, our approach demonstrated competitive outcomes.

2 Literature Review

2.1 BERT:

BERT (Bidirectional Encoder Representations from Transformers) has revolutionized the field of natural language processing with its transformer architecture and pre-trained embeddings. The BERT Multilingual model, in particular, is trained on multiple languages, making it a suitable candidate for cross-lingual tasks (Devlin et al., 2018).

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 472–476
December 7, 2023 ©2023 Association for Computational Linguistics

2.2 Data Augmentation:

In a detailed survey (Feng et al., 2021) addressing data augmentation (DA) in the realm of Natural Language Processing (NLP), researchers spotlighted the increasing significance of DA, especially with the rise of low-resource domains, new NLP tasks, and expansive neural networks. Although DA has been pivotal in machine learning and computer vision, its adoption in NLP remains tentative due to the challenges arising from the discrete nature of language (Feng et al., 2021). The paper underscores the necessity of DA in NLP given the expansion of large pre-trained models and the proliferation of domains with scarce training data. The authors categorize DA techniques into rule-based, example interpolation-based, and model-based strategies, emphasizing their application across various NLP tasks, from bias mitigation to few-shot learning. They further provide a continually updated GitHub repository as a resource for researchers delving into DA in NLP .

2.3 Reverse Dictionary

Reverse dictionaries, also known as retrograde dictionaries, represent a paradigm shift from traditional dictionary structures, enabling users to locate words based on anticipated definitions. A significant challenge in this domain revolves around generating definition glosses that align with user expectations. Subsequently, a growing trend in NLP has centered on the development of dynamic reverse dictionaries capable of interpreting user-input definitions and mapping them back to corresponding words. Pioneering works in this field emphasized the augmentation of definitions using semantically linked words, including synonyms, hypernyms, or hyponyms, a strategy explored across languages like English, Turkish, and Japanese. Successive research has integrated comprehensive lexical resources, including WordNet (Fellbaum, 2010) and the Oxford dictionary, among others, to further refine this approach.

The trajectory of research also unveils a subset focused on utilizing dictionaries as benchmarks for compositional semantics, as seen in the works of (Zanzotto et al., 2010) and (Hill et al., 2016). They employed neural networks and LSTMs respectively to leverage dictionaries for training. Modern iterations of reverse dictionaries utilize neural language models, exemplified by the WantWords system, which is rooted in a Bidirectional Long Short-Term Memory (BiLSTM) architecture and embraces auxiliary tasks to enhance performance. (Yan et al., 2020) endeavored to integrate pre-trained models like BERT for cross-lingual capabilities. The most recent advancements, such as the Persian reverse dictionary by (Malekzadeh et al., 2021), maintain the momentum of NLP innovations in this realm. This evolution culminates in the CODWOE shared task's interest, which emphasizes the reconstruction of word embeddings from their definitions, a premise intimately linked to prior works.

3 Methods

This section explains the on Data Augmentation and Model Architecture part of the paper.

The Data Augmentation section talks about using this method in NLP to make the model stronger and more adaptable by exposing it to a wider variety of language. Different text changing techniques like swapping synonyms, adding or removing words, and switching word order are used to make the training data more varied, which helps the model perform better.

The Model subsection explains the design and training steps, the usage of BERT Multilingual model, andd highlights key parts like Multihead Attention, a Linear Layer, and the choice of Loss Function and Optimizer. This structured approach reflects a systematic endeavor to enhance model performance and adaptability in handling text regression tasks across varying linguistic scenarios.

3.1 Data Augmentation

In the realm of natural language processing, data augmentation is a crucial strategy to enhance the robustness and generalization capabilities of models. By introducing variations in the training data, we can simulate a broader range of linguistic structures and nuances, thereby preparing the model to handle diverse real-world scenarios more effectively.

To achieve this, we have incorporated the following text augmentation techniques:

- **Synonym Replacement:** This technique is designed to introduce variations in word choice while preserving the overall meaning of the sentence. It operates by randomly selecting words from a given sentence and substituting them with their synonyms. These synonyms are sourced from WordNet, a comprehensive lexical database. This was only used for English Task.

- **Random Insertion:** This method involves adding new words into the sentence at random positions. These additional words are synonyms of existing words in the sentence, introducing diversity and expanding the vocabulary. This was only used for English Task.

- **Random Deletion:** By probabilistically removing words from the sentence, this process mimics natural language noise and encourages the model to be more robust by learning to handle missing or incomplete input.

- **Random Swap:** The random word swap technique shuffles the positions of words within the sentence. Words are swapped randomly while ensuring that the sentence's overall structure remains intact. This operation encourages the model to understand word order more flexibly.

Through the integration of these augmentation techniques, we aim to enrich our training data, thereby enhancing the model's performance and adaptability across diverse linguistic scenarios.

3.2 Model

In the context of our text regression task, the architecture and training process of the model are of paramount importance. The BERT Multilingual model serves as the foundation of our approach. It is pre-trained on 106 languages, including Arabic and English, making it a robust choice for the task at hand. The input or the model is 256 for skip-gram with negative-sampling (SGNS) embeddings which is based on word2vec models (Mikolov et al., 2013) trained with gensim (Řehřek and Sojka, 2010). The input or the model is 300 for Electra embeddings (Clark et al., 2020). The following steps elucidate the core components of our approach:

1. **Multihead Attention Head:** Our model incorporates a Multihead Attention. This component is pivotal for text regression tasks, and a cornerstone of the Transformer architecture. It empowers the model to concentrate on various segments of the input sequence, capturing intricate patterns and relationships. The output is 256 for SGNS, and 300 for Electra.

2. **Linear Layer:** fully connected layer that transforms the attention mechanism's output to the desired dimension. It is seamlessly integrated into the model, enabling it to predict continuous values of embeddings from the input text.

3. **Loss Function and Optimizer:**

   - *Loss Function Selection:* The mean squared error (MSE) loss function is employed. This function is a standard choice for regression tasks, quantifying the squared discrepancies between the model's predictions and the actual values.
   - *Optimizer Initialization:* The AdamW optimizer is utilized for optimizing the model's parameters. This optimizer is a variant of the conventional Adam optimizer tailored for deep learning models.
   - *Learning Rate (lr):* The learning rate, a pivotal hyperparameter, is set to 2e-5. It dictates the optimization step size and plays a crucial role in model convergence.

4. **Epochs:** The training encompasses multiple iterations, referred to as epochs, over the entire dataset. For this model, only **two** epochs are executed. Limiting the training to two epochs was done because the validation loss began to increase afterward, which is likely due to the small size of the dataset.

4 Results

To validate the effectiveness of our proposed modifications, we conducted experiments on the provided dataset for the shared task.

4.1 Dataset

The dataset comprises Arabic word definitions and their corresponding word embeddings. It also includes English definitions for the cross-lingual task. The data augmentation for the Arabic => Arabic only used deletion and swapping methods from data augmentation. We generated five different variations of each sentences that was longer than two words. The punctuation was removed. For English => Arabic we used Natural Language Toolkit (NLTK) (Bird et al., 2009) word synonyms to replace words randomly.

4.2 Experimental Setup

We fine-tuned the modified BERT Multilingual model on the training dataset and evaluated its performance on the test set. The evaluation dataset was only used for inference.

4.3 Results

Table 1: Reverse Dictionary Track (RD)

Dataset	Metric	SGNS	Electra
Benchmark	Cosine	35.61%	48.85%
Benchmark	MSE	35.61%	24.94%
Benchmark	Ranking	38.52%	31.28%
Dev	Cosine	49.45%	61.69%
Dev	MSE	3.48%	16.75%
Dev	Ranking	31.45%	24.97%
Test	Cosine	53.8%	62.5%
Test	MSE	3.1%	15.7%
Test	Ranking	29.1%	28.5%

Table 2: Cross-lingual Reverse Dictionary Track (CLRD)

Dataset	Metric	SGNS	Electra
Benchmark	Cosine	26.23%	54.09%
Benchmark	MSE	4.92%	22.11%
Benchmark	Ranking	50.17%	36.22%
Dev	Cosine	27.72%	58.06%
Dev	MSE	5.07%	19.55%
Dev	Ranking	45.77%	25.88%
Test	Cosine	27.0%	56.5%
Test	MSE	5.0%	20.6%
Test	Ranking	45.2%	28.1%

The tables illustrate the model's performance on Reverse Dictionary (RD) and Cross-lingual Reverse Dictionary (CLRD) tasks, comparing the Benchmark results with the Development (Dev) results generated by the Multilingual BERT with data augmentation.

In the RD track, the development and test datasets show a notable improvement in Cosine Similarity compared to the Benchmark dataset, indicating better vector space alignment. The MSE metric in the Dev dataset is significantly lower, suggesting a reduction in error rates. The Ranking metric also shows a decrease, which might indicate an improved model performance in ranking the dictionary entries correctly. The main difference between dev and test datasets for RD is that the Electra ranking was worse in test compared to dev unlike SGNS.

Similarly, in the CLRD track, the development and test datasets show an improvement in Cosine Similarity, indicative of better alignment in the vector space. The MSE is slightly higher in the Dev dataset, suggesting a slight increase in the error rate. The Ranking metric shows a decrease in implies a better performance in ranking tasks. Similar to the previous task the ranking was worse in Electra test unlike SGNS.

The variations in performance metrics between the Benchmark and Dev datasets could be attributed to the utilization of a Multilingual BERT model coupled with data augmentation techniques, which might have contributed to enhancing the model's generalization capabilities and performance in both RD and CLRD tasks.

5 Future research:

1. Augmenting our training dataset by introducing nuanced variations to the glosses, potentially employing paraphrasing techniques or deliberately infusing noise such as typos and word order alterations.

2. Adapting of multi-task learning; alongside our primary regression task, training the model to concurrently predict attributes like part of speech (POS) might bolster its gloss representation capabilities.

3. Integrating additional features, such as the gloss length or its associated POS.

6 Discussion and Conclusion

This paper presented a novel approach to the Arabic Reverse Dictionary Shared Task using a modified BERT Multilingual model. The introduced modifications augmentation and using a multi attention head, have shown promise in enhancing the model's performance, paving the way for future research in this domain. Our experiments demonstrate the potential of the BERT Multilingual model, even simple modifications such as data augmentation and using a multi attention head still provides good results but the improvements in SGNS embeddings is less impressive.

Limitations

While our proposed model demonstrates promise in the Arabic Reverse Dictionary task, there is still room for major improvements mentioned in the discussion. The major limitation was the limited amount of training data.

Ethics Statement

We have ensured that our research adheres to the highest ethical standards. Our methodologies and data handling processes will be released as we are committed to transparency, fairness, and the responsible application of our findings in real-world scenarios.

Acknowledgements

References

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.".

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Christiane Fellbaum. 2010. Wordnet. In *Theory and applications of ontology: computer applications*, pages 231–243. Springer.

Steven Y Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for nlp. *arXiv preprint arXiv:2105.03075*.

Felix Hill, Kyunghyun Cho, Anna Korhonen, and Yoshua Bengio. 2016. Learning to understand phrases by embedding the dictionary. *Transactions of the Association for Computational Linguistics*, 4:17–30.

Arman Malekzadeh, Amin Gheibi, and Ali Mohades. 2021. Predict: persian reverse dictionary. *arXiv preprint arXiv:2105.00309*.

Timothee Mickus, Kees Van Deemter, Mathieu Constant, and Denis Paperno. 2022. Semeval-2022 task 1: CODWOE – comparing dictionaries and word embeddings. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1–14, Seattle, United States. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Radim Řehřek and Petr Sojka. 2010. Software framework for topic modelling with large corpora.

Hang Yan, Xiaonan Li, and Xipeng Qiu. 2020. Bert for monolingual and cross-lingual reverse dictionary. *arXiv preprint arXiv:2009.14790*.

FM Zanzotto, Ioannis Korkontzelos, Francesca Fallucchi, Suresh Manandhar, et al. 2010. Estimating linear models for compositional distributional semantics. In *Proceedings of the 23rd international conference on computational linguistics (COLING)(GGS Conference Rating 2 A)*.

A Example Appendix

This is a section in the appendix.

Rosetta Stone at KSAA-RD Shared Task: A Hop From Language Modeling To Word–Definition Alignment

Ahmed ElBakry[*,1] **Mohamed Gabr**[1] **Muhammad ElNokrashy**[1] **Badr AlKhamissi**[*,2]

[1] Microsoft Egypt [2] EPFL

Abstract

A Reverse Dictionary is a tool enabling users to discover a word based on its provided definition, meaning, or description. Such a technique proves valuable in various scenarios, aiding language learners who possess a description of a word without its identity, and benefiting writers seeking precise terminology. These scenarios often encapsulate what is referred to as the "Tip-of-the-Tongue" (TOT) phenomena. In this work, we present our winning solution for the Arabic Reverse Dictionary shared task. This task focuses on deriving a vector representation of an Arabic word from its accompanying description. The shared task encompasses two distinct subtasks: the first involves an Arabic definition as input, while the second employs an English definition. For the first subtask, our approach relies on an ensemble of finetuned Arabic BERT-based models, predicting the word embedding for a given definition. The final representation is obtained through averaging the output embeddings from each model within the ensemble. In contrast, the most effective solution for the second subtask involves translating the English test definitions into Arabic and applying them to the finetuned models originally trained for the first subtask. This straightforward method achieves the highest score across both subtasks.[1]

1 Introduction

The Tip-of-the-Tongue phenomena, as explained by the authors of Brown and McNeill (1966), is "a state in which one cannot quite recall a familiar word but can recall words of similar form and meaning". A straightforward way to solve this problem, is to have a reverse dictionary; a system that takes a description provided by the user as an input, and outputs the word (Bilac et al., 2004).

The initial solutions were heuristic-based. In their work, Shaw et al. (2013) suggested a method where the tokens in the user-provided description are compared to all dictionary definitions. The system then returns the word with the highest token match. Their method implements different retrieval efficiency tweaks to overcome the issue of excessive time complexity resulting from the comparison operation.

Recent approaches employ neural-based models, since that are capable of better capturing the semantics of an input description, in contrast to the earlier solutions mentioned, which relied on word overlap. In their work, Hill et al. (2015) suggest utilizing a recurrent-neural-network (RNN) to generate a vector representation based on the provided definition. This representation is then compared against a set of word embeddings to select the closest word to return to the user.

The issue of low-frequency words is one of the main challenges of building a reverse dictionary, since these words are the ones that are the less trained and thus have a worse representation compared to more frequent words. Zhang et al. (2019) tackle this problem by handcrafting predictors that extract features inspired by the thought process undergone by humans to get a word given its description.

Polysemy, which is the coexistence of many possible meanings for a word, is another obstacle when building reverse dictionaries. Most of the previously mentioned solutions rely on a a set of static word representation builders such as Word2Vec, which hinders the accuracy of such models. This motivates the use of pretrained language models to produce embeddings that vary based on context. The authors of Yan et al. (2020) probed BERT (Devlin et al., 2019) to predict the word representation, alleviating the issue of polysemy.

Reverse dictionaries can also be cross-lingual; where one aims to retrieve a word in language `X` based on a description provided in language `Y`. Employing any of the previously mentioned solutions

[*] Equal Contribution

[1] https://github.com/bkhmsi/RashidRevDict

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 477–482
December 7, 2023 ©2023 Association for Computational Linguistics

	Example Word	Example Definition
Task 1	تحنَّنَ عليه	تَرَحَم، تَعطَف عليه و رَحمَه
Task 2	زوَّر الكَلَامَ	To knowingly and willfully make a false statement of witness while in court

Table 1: Word-Definition Pairs Illustrating Subtasks 1 and 2.

for a multilingual context necessitates the alignment of word vectors across different languages, a challenging task even for two languages, not to mention when dealing with multiple languages. The authors of Chen et al. (2018) built a collection of bilingual reverse dictionaries using Wiktionary. Other solutions used existing multilingual models, such as mBERT, to reduce the issue of cross-lingual alignment Yan et al. (2020).

The shared task of the Arabic Reverse Dictionary provides a set of words, along with their `SGNS` (Mikolov et al., 2013) and `ELECTRA` Clark et al. (2020) vector representations, and their corresponding definition, in both Arabic and English. A set of Arabic-English word mappings is also supplied to help in building an alignment scheme. The goal of subtask 1 is to predict the `SGNS` and `ELECTRA` embeddings of the set of Arabic words, given the input Arabic definition. Subtask 2 has the same goal except that an English definition is provided instead of Arabic.

The shared task setup poses multiple obstacles that our solutions attempt to overcome: (1) the small size of the set of aligned words, (2) the black-box nature of the `SGNS` and `ELECTRA` word embedding generation pipeline.

Our solution simply finetunes multiple Arabic BERT-based pretrained models to predict an embedding for each word.

2 Datasets

The provided data can be categorised into three distinct datasets.

1. The Arabic Language Dictionary is a dataset with $58,010$ entries, where each of datapoint contains a word, an `ELECTRA` embedding, an `SGNS` embedding, a `gloss` (definition of the word), a `POS` tag, an `ID` and an `English ID` where applicable to link with the alignment data.

2. The English Language Dictionary dataset has $63,596$ datapoints, with the same columns as the Arabic Dictionary except that the embeddings are obtained from English words and not Arabic.

3. The English Arabic Mapped Dictionary has $4,355$ datapoints in total. Each point has the Arabic and English glosses, Arabic and English IDs, Arabic and English words, and the Arabic embeddings.

The first and third datasets are split into `train`, `dev` and `test` sets by the organizers. The English language dictionary however, isn't provided with such divisions. Therefore, we manually split the English dictionary ourselves. Table 2 shows the split sizes of each dataset. The English dictionary was divided into two sets only, `train` and `dev`, since there was no need for a test set in our case, and no submission to be made with this dictionary.

	Train	Dev	Test
Ar Dict	45,200	6,400	6,410
Ar-En Map	2,843	299	1,213
Ar Dict	50,877	12,719	N/A

Table 2: Statistics about Data Sizes

3 System

3.1 Subtask 1: Arabic Definitions to Arabic Embeddings

In this subtask, we finetune four Arabic BERT-based pretrained models. Namely: (1) `MARBERTv2` (Abdul-Mageed et al., 2021), (2) `AraBERTv2` (Antoun et al., 2020), (3) `CamelBERT-MSA` and (4) `CamelBERT-Mix` (Inoue et al., 2021). Each model is finetuned twice for this subtask, once for predicting the corresponding `SGNS` embedding for each input definition, and the other time for predicting the corresponding `ELECTRA` embedding. The final representation is computed by taking the embedding of the `CLS` and passing it through a two-layer dense network with a `Tanh` activation function in between. The model is trained by optimizing the

Mean Squared Error (MSE) between the ground-truth representation and the predicted one. For the learning rate scheduling policy, we used OneCycleLR (Smith and Topin, 2017). Throughout the finetuning process, we evaluate on the devset after every epoch, and take the checkpoint with the highest cosine similarity score. Table 3 shows the values of the hyperparameters used during finetuning.

To identify the optimal ensemble of our finetuned models, we select the model combination that exhibited the highest performance on the devset, determined by the cosine similarity metric, as our final solution. Tables 5 and 6 in the Appendix shows the performance of all model combinations on the devset. The final representation of each ensemble is taken by averaging the predicted embedding of each for a given input definition.

Hyperparameter	**Value**
Batch Size	100
lr	1.0e-4
Learning Rate Sched.	OneCycleLR
pct	0.2
$f_{initial}$	25
f_{final}	100
Weight Decay	1.0e-4
Epochs	20
Optimizer	AdamW

Table 3: Hyperparameters Used

3.2 Subtask 2: English Definitions to Arabic Embeddings

Subtask 2 differs from subtask 1 by utilizing an English definition as input instead of Arabic, with the objective of generating the embedding representation of the Arabic word as output. Several approaches were explored in pursuit of optimizing the system for superior output embedding quality.

Cross-Lingual Alignment This method involves a two-step learning process. First, we leverage the English Language Dictionary to learn to generate the English embeddings from their corresponding English definition. Then the second stage utilize the English Arabic Mapped Dictionary to learn an alignment function between both language representations. Figure 1 shows an illustration of this model. The motivation behind this is that the English pretrained models often yield superior representations compared to their Arabic counterparts due to their training on larger corpora. Here, we used RoBERTa (Liu et al., 2019) to obtain English embeddings, following the same procedure as in subtask 1, and then utilizing an autoencoder model to transform these embeddings into their Arabic representations. Both the encoder and the decoder of the Autoencoder consist of two linear layers with ReLU in between. The input and output dim is 256 and the hidden dim is 32. However, the efficacy of converting an English representation into an Arabic one is contingent upon the quantity of aligned data points available in the provided resources.

Translate-Test Our solution for subtask 2 that yielded the best results was inspired from (Artetxe et al., 2023). In their work, they show that machine translating a non-English test sets into English and then running inference on a monolingual English model can exhibit superior performance compared to using a multilingual model, such as XLM-R (Conneau et al., 2020), on the original data zeroshot. Similarly, we use the Arabic translation of the English definitions as input to our finetuned Arabic models. This approach enables the reuse of models and solutions that were initially developed for subtask 1 .

4 Results

Table 4 displays the results obtained on the test set across all metrics reported in the shared task. Interestingly, the best ensemble on both subtasks was done by taking the average of the CamelBERT-MSA and MARBERTv2 output embeddings.

4.1 Subtask 1

Table 5 shows the results on the devset that we can use for further analysis. It clearly illustrates that ensembles, regardless of the combination, enhance the scores in comparison to using individual models. Furthermore, it is evident that results involving CamelBERT-Mix tend to be less favorable than those involving CamelBERT-MSA. This observation aligns with the dataset's nature, which predominantly features MSA definitions, thus minimizing dialectal content.

Through examining the scores of ensembles and systems incorporating MARBERTv2 compared to those that do not, we can conclude that MARBERTv2 stands out as the most effective model to employ or include in an ensemble among all the tested Arabic pretrained transformers.

Subtask	Embedding	MSE	Cosine	Rank	P@1	P@10
Subtask 1	Electra	0.152 / 0.161	0.645 / 0.637	0.242 / 0.214	0.031 / 0.034	0.099 / 0.114
	SGNS	0.030 / 0.035	0.605 / 0.552	0.254 / 0.281	0.445 / 0.414	0.597 / 0.540
Subtask 2	Electra	0.170 / 0.180	0.659 / 0.624	0.127 / 0.204	0.185 / 0.120	0.407 / 0.355
	SGNS	0.053 / 0.048	0.400 / 0.387	0.320 / 0.372	0.312 / 0.316	0.375 / 0.389

Table 4: Results on TestSet / DevSet for Both Subtasks. **MSE** is Mean-Squared-Error. **P** is Precision.

4.2 Subtask 2

The findings from Subtask 1 are applicable to Subtask 2, and this consistency can be attributed to the reuse of models initially developed in Subtask 1 for Subtask 2.

5 Discussion

Exploring Cross-Lingual Alignment Further In the pursuit of optimizing our approach for the Arabic Reverse Dictionary shared task, we implemented a cross-lingual alignment method, as detailed in section 3.2. This method allowed us to bridge the gap between English and Arabic definitions, by leveraging the aligned dictionary provided as part of the shared task. Further exploration and refinement could yield promising results in that direction.

Augmenting Training Data Through Self-Synthesis In another set of experiments, we explored a very different approach that requires further investigation in future work. The idea is to finetune of an encoder-decoder model, such as AraT5 (Nagoudi et al., 2022), jointly on two interconnected tasks. The first task involves predicting the word embeddings from the encoder side, while the second task entails predicting the corresponding definition on the decoder side based on an the input word. This approach presents an intriguing opportunity to generate diverse definition-embedding pairs using a single model, which could subsequently be harnessed for more robust finetuning. This self-synthesis approach could potentially lead to better system performance by expanding the training set.

6 Conclusion

In this paper, we present our winning solution to the Arabic Reverse Dictionary shared task. The objective is to derive an Arabic word representation based on a provided definition, which can be in either Arabic or English.

Our approach simply leverages several language models pretrained on Arabic datasets. Through finetuning and ensembling the trained models, our method is capable of capturing the underlying semantics of the input definitions as well as correcting small errors done by single models.

For the first subtask, we achieve the best results by fine-tuning four Arabic pretrained language models twice, one for predicting the `Electra` embedding and once for the `SGNS` one. This involves minimizing the discrepancy between the predicted embedding and the model's final representation using an MSE loss function.

In the second subtask, our most effective solution is to repurpose the models initially developed for the first subtask by translating the English test set definitions into Arabic.

Limitations

One notable limitation is related to the second subtask, where our approach involves translating English definitions to Arabic. The results of this paper used the existing Arabic translations that comes English test set. Therefore, we have not investigated the quality of machine translation models, which can significantly influence the system's effectiveness, as inaccuracies or nuances lost in translation may affect results. Moreover, the generalization of our models to broader or different distributions may be constrained, as they are optimized on specific datasets. To achieve wider applicability, we might necessitate further finetuning on more diverse data sources. Furthermore, our choice of evaluation metrics can influence the perceived performance of the system, and different metrics may reveal varying aspects of its utility in practical applications. It is essential to consider these limitations when assessing the robustness and adaptability of our approach.

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. [Cited on page 2.]

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. In *LREC 2020 Workshop Language Resources and Evaluation Conference 11–16 May 2020*, page 9. [Cited on page 2.]

Mikel Artetxe, Vedanuj Goswami, Shruti Bhosale, Angela Fan, and Luke Zettlemoyer. 2023. Revisiting machine translation for cross-lingual classification. *ArXiv*, abs/2305.14240. [Cited on page 3.]

Slaven Bilac, Wataru Watanabe, Taiichi Hashimoto, Takenobu Tokunaga, and Hozumi Tanaka. 2004. Dictionary search based on the target word descrip tion. *In Proceedings of NLP*. [Cited on page 1.]

Roger Brown and David McNeill. 1966. The "tip of the tongue" phenomenon. *Journal of Verbal Learning and Verbal Behavior*, 5(4):325–337. [Cited on page 1.]

Muhao Chen, Yingtao Tian, Haochen Chen, Kai-Wei Chang, Steven Skiena, and Carlo Zaniolo. 2018. Learning to represent bilingual dictionaries. *CoRR*, abs/1808.03726. [Cited on page 2.]

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. *CoRR*, abs/2003.10555. [Cited on page 2.]

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. [Cited on page 3.]

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. [Cited on page 1.]

Felix Hill, Kyunghyun Cho, Anna Korhonen, and Yoshua Bengio. 2015. Learning to understand phrases by embedding the dictionary. *CoRR*, abs/1504.00548. [Cited on page 1.]

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. [Cited on page 2.]

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692. [Cited on page 3.]

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*. [Cited on page 2.]

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. [Cited on page 4.]

Ryan Shaw, Anindya Datta, Debra VanderMeer, and Kaushik Dutta. 2013. Building a scalable database-driven reverse dictionary. *IEEE Transactions on Knowledge and Data Engineering*, 25(3):528–540. [Cited on page 1.]

Leslie N. Smith and Nicholay Topin. 2017. Super-convergence: Very fast training of residual networks using large learning rates. *CoRR*, abs/1708.07120. [Cited on page 3.]

Hang Yan, Xiaonan Li, and Xipeng Qiu. 2020. BERT for monolingual and cross-lingual reverse dictionary. *CoRR*, abs/2009.14790. [Cited on pages 1 and 2.]

Lei Zhang, Fanchao Qi, Zhiyuan Liu, Yasheng Wang, Qun Liu, and Maosong Sun. 2019. Multi-channel reverse dictionary model. *CoRR*, abs/1912.08441. [Cited on page 1.]

A Cross-Lingual Alignment Model

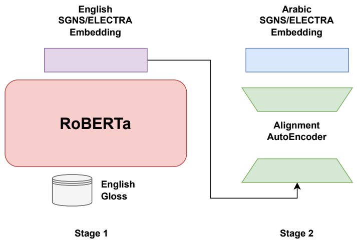


Figure 1: Method Explored for Subtask 2 (Section 3.2)

B Results on Devset

Please refer to next page.

Models	**Electra**			**SGNS**		
	MSE	Cosine	Rank	MSE	Cosine	Rank
arabert	0.1695	0.6124	0.2491	0.0368	0.4853	0.3321
camelbert-mix	0.1689	0.6134	0.2667	0.0389	0.4911	0.3221
camelbert-msa	0.1681	0.6166	0.2421	0.0379	0.4947	0.3213
marbert	0.1661	0.6265	0.2108	0.0370	0.5485	0.2782
arabert,camelbert-mix	0.1656	0.6228	0.2525	0.0360	0.5045	0.3237
arabert,camelbert-msa	0.1650	0.6247	0.2400	0.0358	0.5052	0.3235
arabert,marbert	0.1618	0.6355	0.2175	0.0337	0.5511	0.2836
camelbert-mix,camelbert-msa	0.1653	0.6239	0.2496	0.0370	0.5036	0.3208
camelbert-mix,marbert	0.1622	0.6341	0.2267	0.0348	0.5502	0.2817
camelbert-msa,marbert	0.1614	0.6365	0.2144	0.0345	0.5519	0.2812
arabert,camelbert-mix,camelbert-msa	0.1642	0.6272	0.2455	0.0357	0.5095	0.3221
arabert,camelbert-mix,marbert	0.1616	0.6356	0.2286	0.0339	0.5466	0.2862
arabert,camelbert-msa,marbert	0.1610	0.6371	0.2204	0.0338	0.5472	0.2860
camelbert-mix,camelbert-msa,marbert	0.1614	0.6361	0.2268	0.0346	0.5452	0.2849
arabert,camelbert-mix,camelbert-msa,marbert	0.1613	0.6363	0.2287	0.0341	0.5421	0.2895

Table 5: Performance Analysis on the Devset of Subtask-1 Using Various Model Ensembles.

Models	**Electra**			**SGNS**		
	MSE	Cosine	Rank	MSE	Cosine	Rank
arabert	0.1879	0.6014	0.2369	0.0491	0.3500	0.3925
camelbert-mix	0.1894	0.5974	0.2482	0.0520	0.3504	0.3956
camelbert-msa	0.1860	0.6066	0.2167	0.0498	0.3580	0.3845
marbert	0.1858	0.6108	0.2115	0.0530	0.3818	0.3739
arabert,camelbert-mix	0.1848	0.6104	0.2354	0.0486	0.3619	0.3920
arabert,camelbert-msa	0.1829	0.6149	0.2218	0.0479	0.3640	0.3855
arabert,marbert	0.1806	0.6226	0.2106	0.0478	0.3867	0.3752
camelbert-mix,camelbert-msa	0.1842	0.6117	0.2245	0.0494	0.3619	0.3894
camelbert-mix,marbert	0.1821	0.6186	0.2175	0.0493	0.3838	0.3776
camelbert-msa,marbert	0.1800	0.6238	0.2038	0.0484	0.3874	0.3715
arabert,camelbert-mix,camelbert-msa	0.1827	0.6160	0.2248	0.0481	0.3662	0.3890
arabert,camelbert-mix,marbert	0.1808	0.6222	0.2162	0.0476	0.3841	0.3778
arabert,camelbert-msa,marbert	0.1794	0.6255	0.2075	0.0472	0.3860	0.3736
camelbert-mix,camelbert-msa,marbert	0.1805	0.6228	0.2135	0.0482	0.3836	0.3761
arabert,camelbert-mix,camelbert-msa,marbert	0.1800	0.6240	0.2125	0.0474	0.3830	0.3782

Table 6: Performance Analysis on the Devset of Subtask-2 Using Various Model Ensembles.

ArAIEval Shared Task: Persuasion Techniques and Disinformation Detection in Arabic Text

Maram Hasanain[1], **Firoj Alam**[1], **Hamdy Mubarak**[1], **Samir Abdaljalil**[1],
Wajdi Zaghouani[2], **Preslav Nakov**[3],
Giovanni Da San Martino[4], **Abed Alhakim Freihat**[5]
[1]Qatar Computing Research Institute, HBKU, Qatar
[2]Hamad Bin Khalifa University, Qatar,
[3]Mohamed bin Zayed University of Artificial Intelligence, UAE,
[4]University of Padova, Italy, [5]University of Trento, Italy
{mhasanain, fialam, hmubarak, wzaghouani}@hbku.edu.qa,
preslav.nakov@mbzuai.ac.ae, dasan@math.unipd.it, abdel.fraihat@gmail.com

Abstract

We present an overview of the ArAIEval shared task, organized as part of the first ArabicNLP 2023 conference co-located with EMNLP 2023. ArAIEval offers two tasks over Arabic text: (*i*) persuasion technique detection, focusing on identifying persuasion techniques in tweets and news articles, and (*ii*) disinformation detection in binary and multiclass setups over tweets. A total of 20 teams participated in the final evaluation phase, with 14 and 16 teams participating in Tasks 1 and 2, respectively. Across both tasks, we observed that fine-tuning transformer models such as AraBERT was at the core of the majority of the participating systems. We provide a description of the task setup, including a description of the dataset construction and the evaluation setup. We further give a brief overview of the participating systems. All datasets and evaluation scripts from the shared task are released to the research community.[1] We hope this will enable further research on these important tasks in Arabic.

1 Introduction

Social media has become one of the predominant communication channels for freely sharing content online. With this freedom, misuse has emerged, turning social media platforms into potential grounds for sharing inappropriate posts, misinformation, and disinformation (Zhou et al., 2016; Alam et al., 2022a; Sharma et al., 2022). Malicious users can disseminate disinformative content, such as hate-speech, rumors, and spam, to gain social and political agendas or to harm individuals, entities and organizations. Such content can inflame tension between different groups and ignite violence among their members, making early detection and prevention essential.

Previous successful attempts to address such kinds of problems at a large scale over Arabic content include offensive and hate speech detection shared tasks (Zampieri et al., 2020; Mubarak et al., 2020b).

Social media content designed to promote hidden agendas is not limited to disinformation. In the past years, propaganda has been widely used as well, to influence and/or mislead the audience, which became a major concern for different stakeholders, social media platforms and government agencies. News reporting in the mainstream media also exhibits a similar phenomenon, where a variety of persuasion techniques (Miller, 1939) are used to promote a particular editorial agenda. To address this problem, the research area of "computational propaganda" has emerged aimed at automatically identify such techniques in textual, visual and multimodal (e.g., memes) content. Da San Martino et al. (2019) curated a set of persuasion techniques, such as *Loaded Language*, *Appeal to Fear*, *Straw Man* and *Red Herring*. The focus of the work was mainly on textual content (i.e., newspaper articles). Following this prior work, in 2021, Dimitrov et al. (2021) organized a shared task on propaganda techniques in memes. These efforts mainly focused on English. To enrich the Arabic AI research, we have organized a shared task on detection of fine-grained propaganda techniques for Arabic, which attracted many participants (Alam et al., 2022b).

Following the success of our previous shared tasks (Alam et al., 2022b; Zampieri et al., 2020; Mubarak et al., 2020b), and given the great interest from the community in further pushing research in this domain, this year we organize the **Ar**abic **AI Eval**uation (**ArAIEval**) shared task covering the following two tasks: *(i)* persuasion technique detection over tweets and news articles, and *(ii)* disinformation detection over tweets.

[1] https://araieval.gitlab.io/

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 483–493
December 7, 2023 ©2023 Association for Computational Linguistics

This edition of the shared task has attracted wide participation. The task was run in two phases: *(i)* the development phase with 38 registrations, and 14 teams submitting their systems; and *(ii)* the evaluation phase with 25 registrations, and 20 teams submitting their systems. In the remainder of this paper, we define each of the two tasks, describe the Arabic evaluation datasets that were manually constructed, and provide overview of participating systems and their official scores.

2 Related Work

2.1 Persuasion Techniques Detection

The history of studying propaganda can be traced back to the 17th century, where the focus was to understand whether manipulation techniques were used during public events at theaters, festivals, and games (Margolin, 1979; Casey, 1994). Since then, the study of propaganda has spanned across various disciplines including history, journalism, political science, sociology, and psychology (Jowett and O'donnell, 2018). Different disciplines explored propaganda for varied purposes; for instance, in political science, it is studied to analyze the ideologies of practitioners and to understand the impact of information dissemination on public opinion.

Over the last few decades, the current information ecosystem has undergone significant changes due to the emergence of social media platforms, which have become breeding grounds for the creation and dissemination of misinformation and propaganda. Consequently, there has been research aimed at understanding and automatically detecting such content by defining the rhetorical and psychological techniques employed on online platforms.

Most computational approaches for automatic detection involve identifying whether textual content contains propaganda (Barrón-Cedeno et al., 2019), identifying propagandistic techniques (Habernal et al., 2017, 2018), and detecting propagandistic text spans in news articles (Da San Martino et al., 2019, 2020). The majority of these studies have primarily focused on English. To address this issue in multilingual settings, a shared task was recently organized, focusing on nine languages (Piskorski et al., 2023). The outcomes of such initiatives highlight the importance of multilingual models. For instance, Hasanain et al. (2023) show that multilingual models significantly outperform monolingual models, even for languages unseen during training.

Other relevant shared tasks include those focusing on multimodality. Dimitrov et al. (2021) organized SemEval-2021 Task 6 on the propaganda detection in memes, which comprises a multimodal setup involving both text and images.

Along such initiatives, we have primarily focused on Arabic content. The propaganda shared task, co-located with WANLP 2022, was mainly focused on tweets in both binary and multilabel settings (Alam et al., 2022b). This year, we have expanded it on a larger scale with a larger dataset, focusing on news articles and tweets.

2.2 Disinformation Detection

Disinformation is relatively a new term and it is defined as "*fabricated or deliberately manipulated text/speech/visual context, and also intentionally created conspiracy theories or rumors*" (Ireton and Posetti, 2018). There have been several studies on the automatic detection of bad content on social media, including hate speech (Fortuna and Nunes, 2018), harmful content (Alam et al., 2021, 2022a), rumors (Meel and Vishwakarma, 2020), and offensive language (Husain and Uzuner, 2021).

In the context of Arabic social media, numerous researchers have employed different approaches to disinformation detection. For instance, Boulouard et al. (2022) investigated disinformation detection, particularly hate-speech and offensive content detection, on Arabic social media.

For this shared task on disinformation detection, our work is inspired by Mubarak et al. (2023), which primarily focused on detecting disinformative tweets that are most likely to be deleted.

3 Task 1: Propaganda Detection

The goal of this task is to identify the persuasion techniques present in a piece of text. It targets multi-genre content, including tweets and paragraphs from news articles, as persuasion techniques are commonly used within these domains. The task is organized into two subtasks.

3.1 Subtasks

Subtask 1A: Given a text snippet, identify whether it contains content with any persuasion technique. This is a *binary classification* task.

Subtask 1B: Given a text snippet, identify the propaganda techniques used in it. This is a *multilabel classification* task.

	Train	**Dev**	**Test**
true	1918 (79%)	202 (78%)	331 (66%)
false	509 (21%)	57 (22%)	172 (34%)
Total	**2427**	**259**	**503**

Table 1: Distribution of Subtask **1A** dataset. In parentheses, we show the percentage of a label in a split.

Persuasion Technique	**Train (2427)**	**Dev (259)**	**Test (503)**
Loaded Language	1574	176	253
Name Calling or Labelling	692	77	133
Questioning the Reputation	383	43	89
Exaggeration or Minimisation	292	33	40
Obfuscation, Intentional Vagueness, Confusion	240	28	25
Casting Doubt	143	16	21
Causal Oversimplification	128	15	12
Appeal to Fear, Prejudice	108	12	15
Slogans	70	8	25
Flag Waving	63	7	25
Appeal to Hypocrisy	56	7	17
Appeal to Values	37	4	29
Appeal to Authority	48	5	14
False Dilemma or No Choice	32	3	6
Consequential Oversimplification	33	3	3
Conversation Killer	28	3	7
Repetition	25	3	6
Guilt by Association	13	1	1
Appeal to Time	10	2	2
Whataboutism	9	1	2
Red Herring	8	1	3
Strawman	6	1	2
Appeal to Popularity	2	1	1
No Technique	*509*	*57*	*172*
Total	**4509**	**507**	**903**

Table 2: Distribution of the techniques for the Subtask **1B** dataset: sorted by total frequency over all splits. In parentheses, we show the total number of documents in a split.

3.2 Dataset

To construct the annotated dataset for this task, we collected different datasets consisting of tweets and news articles, as discussed below.

Tweets: We start from the same tweets dataset collected from Twitter accounts of Arabic news sources, as described in the previous edition of the shared task (Alam et al., 2022b). We randomly sampled a subset of 156 tweets for annotation to construct the *testing subset* of this task. The number of tweets selected for annotation was decided based on time and cost required for annotation.

News paragraphs: We select news articles from an existing dataset, AraFacts (Ali et al., 2021), that contains claims verified by Arabic fact-checking websites, and each claim is associated with web pages propagating or negating the claim. We keep the pages that are from news domains in the set (e.g., www.alquds.co.uk). We automatically parsed these news articles and split them into paragraphs based on blank lines.

Data annotation: For both tweets and paragraphs, we follow the same annotation process to identify the persuasion techniques in a snippet. The process includes two phases: (*i*) three annotators independently annotated the same text snippet, through an annotation interface designed for the task, and (*ii*) two consolidators reviewed the annotations and produced the gold annotations. Annotators were recruited and trained for the task in-house. We annotate text by a set of 23 persuasion techniques that is adopted from existing research (Piskorski et al., 2023). We should note here that multiple techniques can be found in the same text snippet. *For Subtask 1A (binary classification)*, the labels were generated by assigning a positive label (true) to every text snippet that had at least one persuasion technique, and a negative label was given otherwise. Below we give an example subset of the persuasion techniques, and briefly summarize them:

1. **Loaded language:** using specific emotionally-loaded words or phrases (positive or negative) to convince the audience that an argument is valid.
2. **Appeal to Fear, Prejudice:** building support or rejection for an idea by instilling fear or repulsion towards it, or to an alternative idea.
3. **Strawman:** giving the impression that an argument is being refuted, whereas the real subject of the argument was not addressed or refuted, but instead was replaced with a different one.

Data splits: The full set of annotated paragraphs is divided into three subsets: train, development, and test, using a stratified splitting approach to ensure that the distribution of persuasion techniques is consistent across the splits. For the tweets set, we split the full annotated tweet set from the previous edition of the lab (Alam et al., 2022b) into train and development subsets, while the test set is annotated for this shared task. Finally, we construct the multi-genre subsets for the task by merging the sets of paragraphs and tweets.

Statistics: In Tables 1 and 2 we show the distribution of labels across splits for Task 1.

Team		Subtask		Model								Misc.			
		1A	1B	AraBERT	MARBERT	ArabicBERT	BERT	RoBERTa	XLM-RoBERTa	AraELECTRA	GPT	Data augm.	Preprocessing	Ensamble	Loss Funct.
HTE	(Khaldi and Bouklouha, 2023)	1	5	✓	✓										✓
KnowTellConvince	(Veeramani et al., 2023)	2				✓						✓		✓	✓
rematchka	(Abdel-Salam, 2023)	3	2	✓	✓										✓
UL & UM6P	(Lamsiyah et al., 2023)	4	1	✓	✓										✓
Itri Amigos	(Ahmed et al., 2023)	5	4	✓											
Raphael	(Utsav et al., 2023)	6	6		✓			✓			✓				
Frank	(Azizov, 2023)	7			✓		✓	✓					✓		
Mavericks	(Mangalvedhekar et al., 2023)	8		✓						✓			✓	✓	
Nexus	(Xiao and Alam, 2023)	9		✓	✓								✓		
AAST-NLP	(ElSayed et al., 2023)	11	3	✓	✓							✓	✓		✓
ReDASPersuasion	(Qachfar and Verma, 2023)	13	7						✓				✓		
Legend	(Ojo et al., 2023)	14							✓						

Table 3: Overview of the systems for **Task 1**. Numbers under the subtask code indicate the position of the team in the official ranking. Data augm.: Data augmentation. Loss Funct.: Experiments with a variety of loss functions.

	Team	Micro F1	Macro F1		Team	Micro F1	Macro F1
	Subtask 1A				**Subtask 1B**		
1	HTE	0.7634	0.7321	1	UL & UM6P	0.5666	0.2156
2	KnowTellConvince	0.7575	0.7282	2	rematchka	0.5658	0.2497
3	rematchka	0.7555	0.7309	3	AAST-NLP	0.5522	0.1425
4	UL & UM6P	0.7515	0.7186	4	Itri Amigos	0.5506	0.1839
5	Itri Amigos	0.7495	0.7225	5	HTE	0.5412	0.0979
6	Raphael	0.7475	0.7221	6	Raphael	0.5347	0.1772
7	Frank	0.7455	0.7173	7	ReDASPersuasion	0.4523	0.0568
8	Mavericks	0.7416	0.7031	8	*Baseline (Majority)*	0.3599	0.0279
9	Nexus	0.7396	0.6929	9	*Baseline (Random)*	0.0868	0.0584
10	superMario	0.7316	0.7098	10	pakapro	0.0854	0.0563
11	AAST-NLP	0.7237	0.6693				
12	*Baseline (Majority)*	0.6581	0.3969				
13	ReDASPersuasion	0.6581	0.3969				
14	Legend	0.6402	0.4647				
15	pakapro	0.5030	0.4940				
16	*Baseline (Random)*	0.4771	0.4598				

Table 4: Official results for **Task 1**. Runs ranked by the official measure: Micro F1.

3.3 Evaluation Setup

The task was organized into two phases:

- **Development phase**: we released the train and development subsets, and participants submitted runs on the development set through a competition on Codalab [2].
- **Test phase**: we released the official test subset, and the participants were given a few days to submit their final predictions through a competition on Codalab.[3] Only the latest submission from each team was considered official and was used for the final team ranking.

Measures: We measure the performance of the participating systems, for all subtasks, using micro-averaged F1 as the official evaluation measure of the shared task, as these are multiclass/multilabel problems, where the labels are imbalanced. We also report macro-averaged F1, as an unofficial evaluation measure.

3.4 Overview of Participating Systems and Results

A total of 14 and 8 teams submitted runs for Subtask 1A and 1B, respectively, with 8 teams making submissions for both subtasks. Table 3, overviews 12 of the participating systems for which a description paper was submitted. Table 4 presents the results and rankings of *all* systems.

[2] https://codalab.lisn.upsaclay.fr/competitions/14563

[3] https://codalab.lisn.upsaclay.fr/competitions/15099

	Train	Dev	Test
Disinfo	2656 (19%)	397 (19%)	876 (23%)
Not-disinfo	11491 (81%)	1718 (81%)	2853 (77%)
Total	**14147**	**2115**	**3729**

Table 5: Distribution of Subtask **2A** dataset. In parentheses, we show the percentage of a label in a split.

Fine-tuning pre-trained Arabic models (specifically AraBERT (Antoun et al., 2020) and MARBERT (Abdul-Mageed et al., 2021)) was the most common system architecture. However, we observed that several systems also experimented with a variety of loss functions for model training to handle characteristics of the training dataset, like label imbalance (Lamsiyah et al., 2023; Khaldi and Bouklouha, 2023; Veeramani et al., 2023; Abdel-Salam, 2023; ElSayed et al., 2023).

When comparing the performance to the previous edition (Alam et al., 2022b) for the multilabel subtask, we observe that this year's Subtask 1B is much more challenging. In the previous edition, the best system achieved a Micro F1 of 0.649, whereas this year it is 0.566, keeping in mind that the dataset is different and may not be exactly comparable.

4 Task 2: Disinformation Detection

This task targeted tweets and was organized into two subtasks, as discussed below.

4.1 Subtasks

Subtask 2A: Given a tweet, identify whether it is disinformative. This is a *binary classification* task.

Subtask 2B: Given a tweet, detect the fine-grained disinformation class, if any. This is a *multiclass classification* task. The fine-grained labels include *hate-speech*, *offensive*, *rumor*, and *spam*.

4.2 Dataset

We have constructed an annotated dataset composed of $20K$ tweets, labeled as disinformative or not-disinformative, along with fine-grained categories for the disinformative set. These tweets are related to COVID-19 and were collected in February and March 2020. We followed the annotation guidelines described in (Mubarak et al., 2020b), (Zampieri et al., 2020), (Mubarak et al., 2022), and (Mubarak et al., 2020a), for hate speech, offensive content, rumor, and spam classes, respectively. More details about data collection and annotation can be found in (Mubarak et al., 2023). Tables 5 and 6 display the statistics of the dataset.

	Train	Dev	Test
HS	1512 (57%)	226 (57%)	442 (50%)
Off	500 (19%)	75 (19%)	160 (18%)
Rumor	191 (7%)	28 (7%)	33 (4%)
Spam	453 (17%)	68 (17%)	241 (28%)
Total	**2656**	**397**	**876**

Table 6: Distribution of Subtask **2B** dataset. In parentheses, we show the percentage of a label in a split.

4.3 Evaluation Setup and Measures

Similar to Task 1, we also conducted this task in two phases as discussed in Section 3.3. Systems were valuated using Micro F1 as the official measure, while also reporting Macro F1.

4.4 Overview of Participating Systems and Results

Table 7 and 8 overviews the submitted systems, and the official results and ranking, respectively. A total of 15 and 11 teams participated in Subtask 2A and 2B, respectively, out of which, 10 made submissions for both subtasks. Out of 17 teams, 13 outperformed the majority baseline for Subtask 2A, whereas out of 11 teams, 9 outperformed the majority baseline for Subtask 2B. These subtasks were dominated by transformer models as observed in Table 7. The most commonly used model was AraBERT (Antoun et al., 2020), followed by MARBERT (Abdul-Mageed et al., 2021), ARBERT(Abdul-Mageed et al., 2021), and QARiB (Abdelali et al., 2021). Half of the participants employed preprocessing techniques, and the top-performing teams utilized data augmentation.

5 Participating Systems

AAST-NLP (ElSayed et al., 2023) The team experimented with several transformer-based models, including MARBERT (Abdul-Mageed et al., 2021), ARBERT (Abdul-Mageed et al., 2021), and AraBERT (Antoun et al., 2020). AraBERT outperformed the others across all subtasks. Preprocessing was applied using the AraBERT preprocessor. Tweet tags, emojis, and Arabic stopwords were removed. For the final submission, binary cross entropy was selected for multilabel classification (Subtask 1B), while Dice loss was chosen for the remaining three subtasks. Although the team tried data augmentation with contextual word embeddings and a hybrid approach combining AraBERT with a CNN-BILSTM, these did not improve accuracy.

Team		Subtask		Model												Misc.	
		2A	2B	AraBERT	MARBERT	ARBERT	QARiB	CAMeLBERT	BERT	RoBERTa	XLM-RoBERTa	DistilBERT	AraELECTRA	LSTM	SVM	Data augm.	Preprocessing
DetectiveRedasers	(Tuck et al., 2023)	1	1	✓	✓		✓	✓								✓	✓
AAST-NLP	(ElSayed et al., 2023)	2	3	✓	✓	✓								✓		✓	✓
UL & UM6P	(Lamsiyah et al., 2023)	3	2	✓	✓	✓											
rematchka	(Abdel-Salam, 2023)	4	4	✓	✓	✓										✓	
PD-AR	(Deka and Revi, 2023)	5	6	✓				✓	✓	✓	✓						✓
Mavericks	(Mangalvedhekar et al., 2023)	7		✓									✓				✓
Itri Amigos	(Ahmed et al., 2023)	8	7	✓													✓
KnowTellConvince	(Veeramani et al., 2023)	9	8	✓													
Nexus	(Xiao and Alam, 2023)	10		✓	✓		✓										
PTUK-HULAT	(Jaber and Martinez, 2023)	11							✓			✓					✓
Frank	(Azizov, 2023)	12			✓				✓	✓							
USTHB	(Mohamed et al., 2023)	13	9												✓		
AraDetector	(Ahmed Bahaaulddin A. et al., 2023)	15		✓	✓		✓										✓

Table 7: Overview of the systems for **Task 2**. The numbers under the subtask code indicate the position of the team in the official ranking. Data augm.: Data augmentation.

	Team	Micro F1	Macro F1		Team	Micro F1	Macro F1
	Subtask 2A				Subtask 2B		
1	DetectiveRedasers	0.9048	0.8626	1	DetectiveRedasers	0.8356	0.7541
2	AAST-NLP	0.9043	0.8634	2	UL & UM6P	0.8333	0.7388
3	UL & UM6P	0.9040	0.8645	3	AAST-NLP	0.8253	0.7283
4	rematchka	0.9040	0.8614	4	rematchka	0.8219	0.7156
5	PD-AR	0.9021	0.8595	5	superMario	0.8208	0.7031
6	superMario	0.9019	0.8625	6	PD-AR	0.8174	0.7209
7	Mavericks	0.9010	0.8606	7	Itri Amigos	0.8139	0.7220
8	Itri Amigos	0.8984	0.8468	8	KnowTellConvince	0.8071	0.6888
9	KnowTellConvince	0.8938	0.8460	9	USTHB	0.5046	0.1677
10	Nexus	0.8935	0.8459	10	*Baseline (Majority)*	0.5046	0.1677
11	PTUK-HULAT	0.8675	0.7992	11	Ankit	0.4167	0.1993
12	Frank	0.8163	0.6378	12	*Baseline (Random)*	0.2603	0.2243
13	USTHB	0.7670	0.4418	13	pakapro	0.2317	0.1978
14	*Baseline (Majority)*	0.7651	0.4335				
15	AraDetector	0.7487	0.6498				
16	*Baseline (Random)*	0.5154	0.4764				
17	pakapro	0.4996	0.4596				

Table 8: Official results for **Task 2**. Runs ranked by the official measure: Micro F1.

AraDetector (Ahmed Bahaaulddin A. et al., 2023) The team tackled Subtask 2A using an ensemble of three classifiers: MARBERT model fine-tuned on the training data, and GPT-4 (OpenAI, 2023) in zero-shot and few-shot settings. A majority voting approach was then used to merge the binary predictions of the three classifiers. The results on the development set showed that GPT-4 in zero-shot setting outperforms the ensemble model by the Micro F1 measure.

DetectiveRedasers (Tuck et al., 2023) The team participated in subtasks 2A and 2B following a two-fold methodology. First, they conducted comprehensive preprocessing, addressing challenges like code-switching and use of emoji in tweets. Non-Arabic portions of the tweets were then automatically translated into Arabic. Instead of removing emojis and hashtags, these were converted into Arabic descriptive text to preserve the sentiment of the tweets. For Subtask 2A, the team used AraBERT-Covid19[4] with hyperparameters optimized through the optimization framework Optuna. As for Subtask 2B, a soft voting ensemble method is used with five optimized AraBERTv02-Twitter (Antoun et al., 2020) models, each with identical hyperparameters and architecture, only differing by random initialization. AraBERTv02-Twitter was selected since it is based on the effective AraBERT mode, with continued pre-training on $60M$ Arabic tweets, making it suitable for Subtask 2B focused on tweets.

[4] https://huggingface.co/moha/arabert_arabic_covid19

Frank (Azizov, 2023) After preprocessing using AraBERT preprocessor, multilingual BERT (Devlin et al., 2018) was fine-tuned for Subtask 1A, and MARBERT was fine-tuned for Subtask 2A.

HTE (Khaldi and Bouklouha, 2023) Participating in Subtask 1A, the team fine-tuned the MARBERT model in a multitask setting: a primary binary classification task to identify the presence of persuasive techniques in text generally, and an auxiliary task focused on classifying texts based on their type (tweet or news). It was expected that the auxiliary task would help the primary task in learning specific lexical and syntactic information about tweets or news related to persuasive content. Given the imbalance in the dataset, the team employed focal loss to optimize both tasks. On the test set, the system ranked highest on the leaderboard.

Itri Amigos (Ahmed et al., 2023) The team submitted runs for all four subtasks. Preprocessing was applied using AraBERT preprocessor. Further preprocessing was done for all subtasks but 1B, where links and mentions were removed. For subtasks 1A and 1B, the team fine-tuned the AraBERTv2 transformer model. To address the class imbalance in the datasets, class weights were incorporated during training. As for subtasks 2A and 2B that are mainly targeting tweets, AraBERTv02-Twitter was fine-tuned for the tasks.

KnowTellConvince (Veeramani et al., 2023) The team participated in subtasks 1A, 2A and 2B using an ensemble of the following four models. *(i)* fine-tuned BERT Arabic base model (Safaya et al., 2020) with a contrastive loss function; *(ii)* fine-tuned BERT Arabic base model with a cross entropy loss function; *(iii)* fine-tuned BERT Arabic base on XNLI dataset to capture nuances relevant to sentiment as part of the system architecture; and *(iv)* a model utilizing sentence embeddings from BERT Arabic base followed by computing cosine similarity between pairs of sentences from the data, that finally goes through Gaussian Error Linear Unit (GELU) activation.

Legend (Ojo et al., 2023) team participated in Task 1, in which XLM-RoBERTa was implemented. To address the class imbalance in the dataset, the team adjusted the learning process using class weights. A learning rate scheduler was implemented to dynamically adjust the learning rate during training. Specifically, they used a StepLR scheduler with a reduction factor of 0.85 applied every 2 epochs. This scheduling strategy contributes to the training stability and the controlled convergence.

Mavericks (Mangalvedhekar et al., 2023) Targeting subtasks 1A and 2A, several transformer-based models were fine-tuned on the provided dataset. The models include: AraBERT, MARBERT and AraELECTRA (Antoun et al., 2021). Ensembling was utilized using hard voting, where the majority vote of all the predictions is selected as the final prediction.

Nexus (Xiao and Alam, 2023) The team explored performance of fine-tuning several pre-trained language models (PLMs) including AraBERT, MARBERT, and QARiB in subtasks 1A and 2A. In addition to that, experiments with GPT-4 (OpenAI, 2023) in both zero-shot and few-shot settings were conducted for both subtasks. Performance of the GPT-4 model was notably lower than the fine-tuned models.

PD-AR (Deka and Revi, 2023) For both sub-tasks 2A and 2B, the team employed the AraBERTv0.2-Twitter-base model and utilized the provided training and development sets to train the model. Before training, some preprocessing of the text was performed. Compared to fine-tuning several other PLMs such as XLM-RoBERTa (Conneau et al., 2020), the Arabic-specific model showed significantly improved performance.

PTUK-HULAT (Jaber and Martinez, 2023) The team participated in Subtask 2A, in which they fine-tuned a multilingual DistilBERT model on the corresponding binary classification data. They then used the fine-tuned model to predict whether a tweet is dis-informative or not.

Raphael (Utsav et al., 2023) For both subtasks 1A and 1B, they used MARBERT as the encoder. In addition to that, they used GPT-3.5 (Brown et al., 2020) in order to generate English descriptions of the Arabic texts and to provide tone and emotional analysis. The resulting English text and tone descriptions were then encoded using RoBERTa (Liu et al., 2019) and were further concatenated to the MARBERT encodings. Finally, the full embeddings were passed to a binary classification head and to multilabel classification heads for Subtasks 1A and 1B, respectively.

ReDASPersuasion (Qachfar and Verma, 2023) The initial structure of the system has three main components: *(i)* A multilingual transformer model that tokenizes the input and produced a [CLS] embedding output; *(ii)* A feature engineering module designed to extract language-agnostic features for persuasion detection; *(iii)* A multi-label classification head that integrates the first and the second components, using a sigmoid activation and cross entropy loss. For subtasks 1A and 1B, the system was paired with DistilBERT (Sanh et al., 2019) for the official submission, but follow-up experiments for Subtask 1A showed that using XLM-RoBERTa, yielded the best Micro F1 score on test.

rematchka (Abdel-Salam, 2023) For all subtasks, ARBERTv2 (Abdul-Mageed et al., 2021), AraBERTv2, and MARBERT models were trained on the provided datasets. For Subtask 1A, different techniques such as fast gradient methods and contrastive learning were applied. Moreover, the team employed back-translation between Arabic and English for data augmentation. As for Subtask 1B, different loss functions, including Asymmetric loss and Distribution Balanced loss were tested. Moreover, a balanced data-sampler for multilabel datasets was used. Fro both subtasks, prefix tuning was used for model training.

UL & UM6P (Lamsiyah et al., 2023) used an Arabic pre-trained transformer combined with a classifier. The performance of three transformer models was evaluated for sentence encoding. For Subtask 1A, the MARBERTv2 encoder was used, and the model was trained with cross-entropy and regularized Mixup (RegMixup) loss functions. For Subtask 1B, the AraBERT-Twitter-v2 encoder was used, and the model was trained with the asymmetric multi-label loss. The significant impact of the training objective and text encoder on the model's performance was highlighted by the results. For Subtask 2A, the AraBERT-Twitter-v2 encoder was used, and the model was trained with cross-entropy loss. For Subtask 2B, the MARBERTv2 encoder was used, and the model was trained with the Focal Tversky loss.

USTHB (Mohamed et al., 2023) For both subtasks 2A and 2B, the system start with extensive preprocessing of the data. Then, the FastText model is used for feature extraction in addition to TF-IDF to vectorize the data. SVM was then trained as a classifier.

6 Conclusion and Future Work

We presented an overview of the ArAIEval shared task at the ArabicNLP 2023 conference, targeting two shared tasks: *(i)* persuasion technique detection, and *(ii)* disinformation detection. The task attracted the attention of many teams: a total of 25 teams registered to participate during the evaluation phase, with 14 and 16 teams eventually making an official submission on the test set for tasks 1 and 2, respectively. Finally, 17 teams submitted a task description paper. Task 1 aimed to identify the propaganda techniques used in multi-genre text snippets, including tweets and news articles, in both binary and multilabel settings. On the other hand, Task 2 aimed to detect disinformation in tweets in both binary and multiclass settings. For both tasks, the majority of the systems fine-tuned pretrained Arabic language models and used standard pre-processing. Several systems explored different loss functions, while a handful of systems utilized data augmentation and ensemble methods.

Given the success of the task this year, we plan to run a future edition with an increased data size, and with wider coverage of domains, countries, and Arabic dialects. We are also considering implementing a multi-granularity persuasion techniques detection setting.

Limitations

Task 1 was limited to binary an multilabel classification. A natural next step would have been to also run a span detection subtask, which is a more complex task. This was left for future editions of ArAIEval. This is to ensure enough participation after building a strong community working on propaganda detection over Arabic content in the less complex setups. As for Task 2, we observe the systems achieved significantly high performance, even in the more challenging multiclass setup. One potential reason might be that the dataset developed was too easy. Investigating how to make this task more challenging while reflecting real-world scenarios was not in this edition of the shared task, but is within our future plan.

Acknowledgments

M. Hasanain's contribution was funded by NPRP grant 14C-0916-210015, and W. Zaghouani contribution was partially funded by NPRP grant 13S-0206-200281 Both projects are funded by the Qatar

National Research Fund (a member of Qatar Foundation).

Part of this work was also funded by Qatar Foundation's IDKT Fund TDF 03-1209-210013: *Tanbih: Get to Know What You Are Reading*.

We thank Fatema Akter and Hussein Mohsin Al-Dobashi for helping with the persuasion techniques annotations.

The findings herein are solely the responsibility of the authors.

References

Reem Abdel-Salam. 2023. rematchka at ArAIEval Shared Task: Prefix-Tuning & Prompt-tuning for Improved Detection of Propaganda and Disinformation in Arabic Social Media Content. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore. Association for Computational Linguistics.

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

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2021. ARBERT & MARBERT: Deep bidirectional transformers for Arabic. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7088–7105, Online. Association for Computational Linguistics.

Nouman Ahmed, Natalia Flechas Manrique, and Jehad Oumer. 2023. Itri Amigos at ArAIEval Shared Task: Transformer vs. Compression-Based Models for Persuasion Techniques and Disinformation Detection. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore. Association for Computational Linguistics.

Wahhab Ahmed Bahaaulddin A., Sabeeh Vian, Belhaj Hanan Mohamed, Sibaee Serry, Samar Ahmad, Khurfan Ibrahim, and Alharbi Abdullah I. 2023. AraDetector at ArAIEval Shared Task: An Ensemble of Arabic-specific pre-trained BERT and GPT-4 for Arabic Disinformation Detection . In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore. Association for Computational Linguistics.

Firoj Alam, Stefano Cresci, Tanmoy Chakraborty, Fabrizio Silvestri, Dimiter Dimitrov, Giovanni Da San Martino, Shaden Shaar, Hamed Firooz, and Preslav Nakov. 2022a. A survey on multimodal disinformation detection. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6625–6643, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Firoj Alam, Hamdy Mubarak, Wajdi Zaghouani, Preslav Nakov, and Giovanni Da San Martino. 2022b. Overview of the WANLP 2022 shared task on propaganda detection in Arabic. In *Proceedings of the Seventh Arabic Natural Language Processing Workshop*, WANLP '22, Abu Dhabi, UAE.

Firoj Alam, Shaden Shaar, Fahim Dalvi, Hassan Sajjad, Alex Nikolov, Hamdy Mubarak, Giovanni Da San Martino, Ahmed Abdelali, Nadir Durrani, Kareem Darwish, Abdulaziz Al-Homaid, Wajdi Zaghouani, Tommaso Caselli, Gijs Danoe, Friso Stolk, Britt Bruntink, and Preslav Nakov. 2021. Fighting the COVID-19 infodemic: Modeling the perspective of journalists, fact-checkers, social media platforms, policy makers, and the society. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 611–649, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zien Sheikh Ali, Watheq Mansour, Tamer Elsayed, and Abdulaziz Al-Ali. 2021. AraFacts: the first large arabic dataset of naturally occurring claims. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 231–236.

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

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

Dilshod Azizov. 2023. Frank at ArAIEval Shared Task: Arabic Disinformation and Persuasion: Power of Pretrained Models. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore. Association for Computational Linguistics.

Alberto Barrón-Cedeno, Israa Jaradat, Giovanni Da San Martino, and Preslav Nakov. 2019. Proppy: Organizing the news based on their propagandistic content. *Information Processing & Management*, 56(5):1849–1864.

Zakaria Boulouard, Mariya Ouaissa, Mariyam Ouaissa, Moez Krichen, Mutiq Almutiq, and Gasmi Karim. 2022. Detecting hateful and offensive speech in arabic social media using transfer learning. *Applied Sciences*, 12:12823.

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

Ralph D. Casey. 1994. What is propaganda? *historians.org*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard 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.

Giovanni Da San Martino, Alberto Barrón-Cedeno, Henning Wachsmuth, Rostislav Petrov, and Preslav Nakov. 2020. Semeval-2020 task 11: Detection of propaganda techniques in news articles. In *Proceedings of the fourteenth workshop on semantic evaluation*, pages 1377–1414.

Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeño, Rostislav Petrov, and Preslav Nakov. 2019. Fine-grained analysis of propaganda in news articles. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, EMNLP-IJCNLP 2019, Hong Kong, China.

Pritam Deka and Ashwathy Revi. 2023. PD-AR at ArAIEval Shared Task: Persuasion techniques detection: an interdisciplinary approach to identifying and counteracting manipulative strategies. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dimitar Dimitrov, Bishr Bin Ali, Shaden Shaar, Firoj Alam, Fabrizio Silvestri, Hamed Firooz, Preslav Nakov, and Giovanni Da San Martino. 2021. SemEval-2021 task 6: Detection of persuasion techniques in texts and images. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 70–98, Online. Association for Computational Linguistics.

Ahmed ElSayed, Omar Nasr, and Nour Eldin Elmadany. 2023. AAST-NLP at ArAIEval Shared Task: Tackling Persuasion Technique and Disinformation Detection using Pre-Trained Language Models On Imbalanced Datasets. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore. Association for Computational Linguistics.

Paula Fortuna and Sérgio Nunes. 2018. A survey on automatic detection of hate speech in text. *ACM Computing Surveys (CSUR)*, 51(4):1–30.

Ivan Habernal, Raffael Hannemann, Christian Pollak, Christopher Klamm, Patrick Pauli, and Iryna Gurevych. 2017. Argotario: Computational argumentation meets serious games. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 7–12, Copenhagen, Denmark. Association for Computational Linguistics.

Ivan Habernal, Patrick Pauli, and Iryna Gurevych. 2018. Adapting Serious Game for Fallacious Argumentation to German: Pitfalls, Insights, and Best Practices. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 3329–3335.

Maram Hasanain, Ahmed El-Shangiti, Rabindra Nath Nandi, Preslav Nakov, and Firoj Alam. 2023. QCRI at SemEval-2023 task 3: News genre, framing and persuasion techniques detection using multilingual models. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1237–1244, Toronto, Canada. Association for Computational Linguistics.

Fatemah Husain and Ozlem Uzuner. 2021. A survey of offensive language detection for the arabic language. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 20(1):1–44.

Cherilyn Ireton and Julie Posetti. 2018. *Journalism, fake news & disinformation: handbook for journalism education and training*. Unesco Publishing.

Areej Jaber and Paloma Martinez. 2023. PTUK-HULAT at ArAIEval Shared Task: Fine-tuned Distilbert to Predict Disinformative Tweets. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore. Association for Computational Linguistics.

Garth S Jowett and Victoria O'donnell. 2018. *Propaganda & persuasion*. Sage publications.

Hadjer Khaldi and Taqiy Eddine Bouklouha. 2023. HTE at ArAIEval Shared Task: Persuasion techniques detection: an interdisciplinary approach to identifying and counteracting manipulative strategies. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore. Association for Computational Linguistics.

Salima Lamsiyah, El Mahdaouy Abdelkader, Hamza Alami, Ismail Berrada, and Christoph Schommer. 2023. UL& UM6P at ArAIEval Shared Task: Transformer-based model for Persuasion Techniques and Disinformation detection in Arabic. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore. Association for Computational Linguistics.

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

Sudeep Mangalvedhekar, Kshitij Deshpande, Yash Patwardhan, Vedant Deshpande, and Ravindra Murumkar. 2023. Mavericks at ArAIEval Shared Task: Towards a Safer Digital Space - Transformer Ensemble Models Tackling Deception and Persuasion. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore. Association for Computational Linguistics.

V. Margolin. 1979. The visual rhetoric of propaganda. *Information Design Journal*, 1:107–122.

Priyanka Meel and Dinesh Kumar Vishwakarma. 2020. Fake news, rumor, information pollution in social media and web: A contemporary survey of state-of-the-arts, challenges and opportunities. *Expert Systems with Applications*, 153:112986.

Clyde R. Miller. 1939. The Techniques of Propaganda. From "How to Detect and Analyze Propaganda," an address given at Town Hall. The Center for learning.

Lichouri Mohamed, Lounnas Khaled, Zitouni Aicha, Latrache Houda, and Djeradi Rachida. 2023. USTHB at ArAIEval Shared Task: Disinformation Detection System based on Linguistic Feature Concatenation. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore. Association for Computational Linguistics.

Hamdy Mubarak, Samir Abdaljalil, Azza Nassar, and Firoj Alam. 2023. Detecting and reasoning of deleted tweets before they are posted. *arXiv preprint arXiv:2305.04927*.

Hamdy Mubarak, Ahmed Abdelali, Sabit Hassan, and Kareem Darwish. 2020a. Spam detection on arabic twitter. In *Social Informatics: 12th International Conference, SocInfo 2020, Pisa, Italy, October 6–9, 2020, Proceedings 12*, pages 237–251. Springer.

Hamdy Mubarak, Kareem Darwish, Walid Magdy, Tamer Elsayed, and Hend Al-Khalifa. 2020b. 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, Sabit Hassan, Shammur Absar Chowdhury, and Firoj Alam. 2022. ArCovidVac: Analyzing Arabic tweets about COVID-19 vaccination. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3220–3230, Marseille, France. European Language Resources Association.

Olumide E. Ojo, Olaronke O. Adebanji, Hiram Calvo, Damian O. Dieke, Olumuyiwa E. Ojo, Seye E. Akinsanya, Tolulope O. Abiola, and Anna Feldman. 2023. Legend at ArAIEval Shared Task: Persuasion Technique Detection using a Language-Agnostic Text Representation Model. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore. Association for Computational Linguistics.

OpenAI. 2023. GPT-4 technical report. Technical report, OpenAI.

Jakub Piskorski, Nicolas Stefanovitch, Giovanni Da San Martino, and Preslav Nakov. 2023. SemEval-2023 task 3: Detecting the category, the framing, and the persuasion techniques in online news in a multilingual setup. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2343–2361, Toronto, Canada. Association for Computational Linguistics.

Fatima Zahra Qachfar and Rakesh M. Verma. 2023. ReDASPersuasion at ArAIEval Shared Task: Multilingual and Monolingual Models For Arabic Persuasion Detection. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore. Association for Computational Linguistics.

Ali Safaya, Moutasem Abdullatif, and Deniz Yuret. 2020. KUISAIL at SemEval-2020 task 12: BERT-CNN for offensive speech identification in social media. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 2054–2059, Barcelona (online). International Committee for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

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, Vienna, Austria. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

Bryan E. Tuck, Fatima Zahra Qachfar, Dainis Boumber, and Rakesh M. Verma. 2023. DetectiveRedasers at ArAIEval Shared Task: Leveraging Transformer Ensembles for Arabic Deception Detection. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore. Association for Computational Linguistics.

Shukla Utsav, Tiwari Shailendra, and Vyas Manan. 2023. Raphael at ArAIEval Shared Task: Understanding Persuasive Language and Tone, an LLM Approach. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore. Association for Computational Linguistics.

Hariram Veeramani, Surendrabikram Thapa, and Usman Naseem. 2023. KnowTellConvince at ArAIEval Shared Task: Disinformation and Persuasion Detection using Similar and Contrastive Representation Alignment. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore. Association for Computational Linguistics.

Yunze Xiao and Firoj Alam. 2023. Nexus at ArAIEval Shared Task: Fine-Tuning Arabic Language Models for Propaganda and Disinformation Detection. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore. Association for Computational Linguistics.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020. Semeval-2020 task 12: Multilingual offensive language identification in social media (offenseval 2020). *arXiv preprint arXiv:2006.07235*.

Lu Zhou, Wenbo Wang, and Keke Chen. 2016. Tweet properly: Analyzing deleted tweets to understand and identify regrettable ones. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, page 603–612, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

DetectiveRedasers at ArAIEval Shared Task: Leveraging Transformer Ensembles for Arabic Deception Detection

Bryan E. Tuck
University of Houston
betuck@uh.edu

Fatima Zahra Qachfar
University of Houston
fqachfar@uh.edu

Dainis Boumber
University of Houston
dboumber@uh.edu

Rakesh M. Verma
University of Houston
rmverma2@central.uh.edu

Abstract

This paper outlines a methodology aimed at combating disinformation in Arabic social media, a strategy that secured a first-place finish in tasks 2A and 2B at the ArAIEval shared task during the ArabicNLP 2023 conference. Our team developed a hyperparameter-optimized pipeline centered around BERT-based models for the Arabic language, enhanced by a soft-voting ensemble strategy. Subsequent evaluation on the test dataset reveals that ensembles, although generally resilient, do not always outperform individual models. The primary contributions of this paper are its multifaceted strategy, which led to winning solutions for both binary (2A) and multiclass (2B) disinformation classification tasks.

1 Introduction

The spread of disinformation across social media platforms presents an omnipresent challenge that transcends modalities, manifesting in text, audio, and images (Shu et al., 2020). Within the sphere of text, disinformation is not exclusive to one language but spans many languages and dialects. Its impact permeates several topics, including politics, entertainment, sports, and finance. In our study, we direct our efforts to detecting disinformation in Arabic on Twitter as part of the Shared Task 2: Arabic Deception Detection, at ArAIEval, ArabicNLP 2023 (Hasanain et al., 2023).

From a research standpoint, Arabic has been receiving an increasing amount of attention addressing several key problems (Farghaly and Shaalan, 2009). Investigations into disinformation in Arabic can offer valuable insights into unique linguistic and cultural aspects that influence the dissemination and impact of false information within Arabic-speaking communities. In a social aspect, the consequences of disinformation across all global communities are profound. Specifically, Arabic is spoken by hundreds of millions of people worldwide and serves as a linguistic backbone for critical geopolitical regions. While also not exclusive to Arabic, disinformation can impact democratic processes and public health (Wolfsfeld et al., 2013). Research in this critical domain has real-world implications that can influence policy decisions, governance, and public well-being.

Arabic itself presents its own set of complexities; it is a rich language featuring intricate word formations and variations (Alzanin et al., 2022). This makes the language both highly derivational, meaning words can be formed from root words in various ways, and inflectional, indicating that the form of words can change to convey different meanings. These linguistic traits add an extra layer of difficulty to the already challenging task of disinformation detection.

Shared task 2 comprises two separate sub-tasks. Task 2A is a binary classification challenge requiring us to categorize whether a given tweet is disinformative. Task 2B, on the other hand, is a more nuanced multiclass classification task, where the objective is to identify fine-grained disinformation classes such as hate speech, offensive content, rumors, or spam (Mubarak et al., 2023b). With this task, there are several open problems due to phenomena including code-switching (Bentahila and Davies, 1983), short texts, and lack of grammatical structure in tweets. These issues lead to the deterioration of the effectiveness of conventional analytical tools. Code-switching refers to the practice of switching between languages within a conversation, or text. Tweets tend to mirror the linguistic styles and variations spoken by individuals hailing from a particular region. For example, Moroccan tweets contain Moroccan Darija mixed with French, English, or Spanish. This phenomenon can occur for various reasons, including cultural exchange, or historical factors such as colonization.

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 494–501
December 7, 2023 ©2023 Association for Computational Linguistics

In addressing disinformation detection in Arabic, our multi-faceted strategy begins with specialized preprocessing, including handling code-switching and incorporating tweet elements like hashtags and URLs, which previous literature often neglects (Bennessir et al., 2022). We then utilize large language models, specifically AraBERT (Antoun et al., 2020), and experiment with a soft-voting ensemble to improve performance. While effective, these large models are computationally expensive; we seek to mitigate this through optimization pipelines, which in turn add their own computational overhead.

2 Dataset and Tasks

In the ArAIEval shared task at ArabicNLP 2023, participants are presented with two main tasks: task 1 focuses on Persuasion Technique Detection, while task 2 aims at Disinformation Detection. Each of these primary tasks are further divided into two sub-tasks. Our research specifically concentrates on task 2, which consists of sub-task 2A and sub-task 2B. In sub-task 2A, the goal is to classify tweets as either disinformative or not, a binary classification problem. For sub-task 2B, we must identify specific types of disinformation within a tweet, which involves a multiclass classification framework. The fine-grained labels that we consider include hate speech, offensive language, rumors, and spam (Hasanain et al., 2023)(Mubarak et al., 2023a). Tables 1 and 2 represent the class distributions and total size of the training, validation, and testing sets, for task 2A and 2B, respectively.

	No Disinformation	Disinformation	Total
Training	11491	2656	14147
Dev	1718	397	2115
Test	2853	876	3729

Table 1: Class Distribution for Task 2A

	Hate-Speech	Offensive	Rumor	Spam	Total
Training	1512	500	191	453	2656
Dev	226	75	28	68	397
Test	442	160	33	241	876

Table 2: Class Distribution for Task 2B

3 System

For tasks 2A and 2B, our approach adopts a specialized methodology using comprehensive preprocessing which deals with code-switching and emoji conversion. After which an intensive search for optimal large language models and hyperparameters is performed. Our decisions of which models to utilize were based on performance on the validation set. The AraBERT-Covid19 model (Antoun et al., 2020) surfaced as the best fit for task 2A. This model, an enhancement of the original AraBERTv02, has been further refined through fine-tuning on 1.5 million multi-dialect Arabic tweets. These tweets, sourced from the extensive Arabic Twitter dataset (Alqurashi et al., 2020), specifically focused on Covid-19. Conversely, for task 2B, we utilize AraBERTv02-Twitter, which was pre-trained on approximately 60 million tweets spanning various Arabic dialects. Subsequently, we employ a soft voting ensemble method, integrating five AraBERTv02-Twitter models that have been optimized. While each model maintains identical hyperparameters and architecture, they differ solely in terms of random initialization. For this process, we utilized the TorchEnsemble library[1]. We optimize both AraBERTv02-Twitter and AraBERT-Covid19 models leveraging the optimization framework Optuna (Akiba et al., 2019). By the deadline for task 2, only two optimized models were evaluated: the AraBERT-Covid19 model for task 2A and the AraBERTv02-Twitter ensemble for task 2B.

To ensure the best performance in regards to our target metric, “micro f1”, we explored a variety of models. Our initial model candidate list included the following: *a*) AraBERTv02-Twitter (Antoun et al., 2020) *b*) Arabert-Covid19 (Alqurashi et al., 2020) *c*) QCRI Arabic and Dialectal BERT (QARiB) (Abdelali et al., 2021) *d*) MARBERTV2 (Abdul-Mageed et al., 2021) *e*) and CAMeLBERT-DA SA (Inoue et al., 2021). Post-competition experimentation can be found in A.1.

3.1 Preprocessing

The Arab world has a rich and diverse history of languages, with many different dialects spoken across different regions. We have analyzed the provided data in both tasks using dialect identification, and we have found that most tweets in the dataset originated from the Kingdom of Saudi Arabia (KSA), Kuwait, and Egypt. We report these results in detail in Appendix A.

[1] https://github.com/TorchEnsemble-Community/Ensemble-Pytorch

3.1.1 Code Switching

Arabic tweeters may use code-switching to express themselves more effectively or to communicate with a diverse audience. For example, users may start a tweet in Arabic, switch to English in the middle, and then finish it off in French. We now describe the preprocessing techniques we applied to the tweets to translate code-switched text to Arabic. For each tweet, we automatically detect code-switching fragments using "*Lingua*" [2] Python package, and we translate it to Arabic using Google's translation API.

3.1.2 Emoji Conversion

In tweets, emojis are typically used to convey emotions or ideas. Mubarak et al. (2022) showed the importance of emojis in the detection of Arabic offensive language and hateful speech.

Instead of removing all emojis from tweets like (Bennessir et al., 2022), we choose to convert them to Arabic descriptive text since emojis might hold meaning in the context of a short deceptive tweet representing positive or negative sentiment. For this we add Arabic language support to the "*emoji*" [3] Python package using normalized representations from the latest release of Unicode Common Locale Data Repository (CLDR) [4] to avoid broken Unicode. We create a dictionary of Arabic emoji representation based on the *emojiterra* website.[5]

3.2 Hyperparameter Optimization

We use the Optuna framework (Akiba et al., 2019) for hyperparameter optimization, primarily due to its straightforward setup, versatility, and choices of efficient sampling and pruning algorithms. For tasks 2A and 2B, we opted for the Tree-Structured Parzen Estimator (TPE) (Bergstra et al., 2011) as our sampling method, as it offers superior efficiency compared to traditional grid search techniques. We began the optimization process with multivariate and grouping settings, integrating a Hyperband pruner (Li et al., 2018), stopping unpromising trials early. This setup allowed each trial to run for a duration ranging from two to twelve epochs. The optimization process encompassed 100 trials aimed at maximizing the "micro-f1" metric, the search space is detailed in Table 3, with the addition of the five candidate models outlined in Section 3. Post-competition, we continued to fine-tune individual models under the same conditions, and these results, along with original task hyperparameters, are located in tables 5, 6, 7, and 8 in Appendix A.1.

Parameter	Value
Learning Rate	1e-05 - 5e-05
Batch Size	8, 16, 32, 64
Dropout	0.0 - 0.5
Max Length	32 - 128

Table 3: Optuna Search Space

3.3 Voting Ensemble

Ensembling techniques, like hyperparameter optimization, come with computational expenses and tuning complexities. The success of ensemble methods hinges on several factors, including the training process of the baseline models (Mohammed and Kora, 2023). Our ensemble employs a "soft voting" scheme, guided by the performance of our top individual model identified through hyperparameter optimization. In this configuration, we employ five AraBERTv02-Twitter models for task 2A and five AraBERTv02-Covid19 models for task 2B, each optimized according to the parameters specified in Table 3. The ensemble is trained for two epochs, which was found to be the point of peak validation performance.

In the soft voting mechanism (Zhou, 2012), each individual classifier, denoted as h_i, generates a l-dimensional vector $(h_i^1(x)..., h_i^l(\mathbf{x}))^T$ for a given instance **x**. Here $h_i^j(\mathbf{x})$ represents the estimated posterior probability $P(c_j|\mathbf{x})$ and falls within the range of $[0, 1]$. The final output for class c_j is the average of all individual outputs, represented as follows:

$$H^j(\mathbf{x}) = \frac{1}{T}\sum_{i=1}^{T} h_j^i(\mathbf{x})$$

3.4 Training Procedure

Our optimization and fine-tuning pipeline uses the AdamW optimizer for effective parameter updates and Cross Entropy as the loss function, given its efficacy in classification problems. We use early stopping with five epochs as a stopping criteria, saving the model best last state. To expedite training without compromising model quality, we uti-

[2] https://github.com/pemistahl/lingua

[3] https://github.com/carpedm20/emoji/

[4] https://github.com/unicode-org/cldr/raw/release-43/common/annotations/ar.xml

[5] https://emojiterra.com/copypaste/ar/

Task	Model	Validation		Test	
		Macro F1	Micro F1	Macro F1	Micro F1
Task 2A	AraBERT-Covid19	**84.73%**	**91.06%**	**86.26%**	**90.48%**
	AraBERT-Covid19 Ensemble	84.31%	90.58%	85.84%	90.02%
Task 2B	AraBERTv02-Twitter	81.12%	84.89%	**75.51%**	**84.36%**
	AraBERTv02-Twitter-Ensemble	**82.19%**	**85.14%**	75.41%	83.56%

Table 4: Micro F1 and Macro F1 scores are presented for each task, covering both validation and testing sets. The highest values are highlighted in **bold**.

lize automatic mixed precision (AMP)[6], reducing both memory usage and training time. Notably, we choose not to employ a learning rate scheduler, deviating from some traditional approaches. As a safety measure, we also implement gradient clipping with a maximum norm of 1.0 to ensure numerical stability and avoid issues like the exploding gradient problem.

4 Results and Discussion

The top two candidate models, identified through hyperparameter optimization, were AraBERT-Covid19 for task 2A and AraBERTv02-Twitter for task 2B. These selections represented the only results submitted by the task deadline. Our results presented in Table 4 reveal some compelling patterns and anomalies. Specifically, task 2A favored the single AraBERT-Covid19 model over its ensemble counterpart. This approach led by a noticeable margin of 0.48% macro f1 and 0.43% micro f1 with the validation set.

Task 2B presents a more intricate challenge, which utilizes AraBERTv02-Twitter as the primary model. While the AraBERTv02-twitter ensemble performed better during the validation phase, it was ultimately outperformed by the single AraBERTv02-Twitter model in the test set by 0.1% macro f1 and 0.8% micro f1. The drop in macro f1 scores from the validation to the test set in task 2B suggests an issue with model generalization. This might be attributed to the inherent complexity of multiclass problems, which often require capturing more nuanced relationships in the data. This presents a challenging task compared to a binary classification task like task 2A. Another challenge for task 2B is the smaller dataset in comparison to task 2A, which can be seen in Section 2, Table 2 and Table 1 respectively. With the unbalanced nature of task 2 as a whole, the small dataset size, and a more intricate class balancing issue, our approach may have failed to learn minority classes, overfitting to the majority classes.

It's also important to highlight that we did not fine-tune the ensemble's hyperparameters, which could have contributed to its less-than-optimal performance against the single models. This supports the idea that ensemble methods, while often robust, require task-specific validation. In future work, optimization techniques specifically for ensembles and not just the individual models may prove to be beneficial, such as an varied amount of classifiers in the ensemble or different weighting techniques. The exploration of additional preprocessing techniques to better handle code-switching could also be a beneficial avenue.

Our results reiterate the importance of nuanced model selection, especially given the challenges posed by binary and multiclass classification tasks. Our findings also pave the way for future work focused on improving computational efficiency and generalization capabilities of disinformation detection models.

5 Conclusion

In this study, we tackled the nuanced problem of disinformation detection in Arabic, a language fraught with complexities like code-switching and dialectal variations. We combined meticulous preprocessing with hyperparameter-optimized AraBERT models, effectively achieving first-place performance in both binary and multiclass deception detection tasks at ArAIEval 2023. A notable insight from our empirical analysis is that individual models occasionally outperform ensembles, indicating the need for careful model selection. Our results not only validate our comprehensive approach but also invite further research into optimizing ensemble methods and addressing the challenges associated with code-switching and dialectal variations in Arabic text. Future work should look at refining these ensemble strategies and explor-

[6] https://pytorch.org/docs/stable/amp.html

ing additional preprocessing techniques, as we aim to create universally effective tools for countering disinformation.

Limitations

While our methodology is proven to work well for the Arabic language and the disinformation detection task, it may not transfer as well to other languages or other domains. Further experimentation on other languages and domains would be required to evaluate the overall efficacy of our pipelines. Lack of time with respect to the task did not allow us to delve into ensemble optimization or explore other possible ensembling techniques. The computational complexity of hyperparameter optimization with additional overhead from transformer architectures and ensemble methods may lead to scaling issues with larger datasets and other domains.

Ethics Statement

Our work complies with the ACL Ethics Policy. We report details of the hyperparameters and architectures for reproducibility. We plan to make the pipeline available in the near future for the benefit of other researchers.

Acknowledgements

We thank the ArAIEval reviewers for their constructive suggestions. This research was partly supported by NSF grants 1433817, 1950297, and 2210198, ARO grants W911NF-20-1-0254 and W911NF-23-1-0191, and ONR award N00014-19-S-F009. Verma is the founder of Everest Cyber Security and Analytics, Inc.

References

Ahmed Abdelali, Sabit Hassan, Hamdy Mubarak, Kareem Darwish, and Younes Samih. 2021. Pre-training bert on arabic tweets: Practical considerations. *ArXiv preprint*, abs/2102.10684.

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2021. ARBERT & MARBERT: Deep bidirectional transformers for Arabic. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7088–7105, Online. Association for Computational Linguistics.

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, pages 2623–2631. ACM.

Sarah Alqurashi, Ahmad Alhindi, and Eisa Alanazi. 2020. Large arabic twitter dataset on covid-19. *ArXiv preprint*, abs/2004.04315.

Samah M. Alzanin, Aqil M. Azmi, and Hatim A. Aboalsamh. 2022. Short text classification for Arabic social media tweets. *Journal of King Saud University - Computer and Information Sciences*, 34(9):6595–6604.

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.

Mohamed Aziz Bennessir, Malek Rhouma, Hatem Haddad, and Chayma Fourati. 2022. iCompass at Arabic hate speech 2022: Detect hate speech using QRNN and transformers. In *Proceedinsg of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools with Shared Tasks on Qur'an QA and Fine-Grained Hate Speech Detection*, pages 176–180, Marseille, France. European Language Resources Association.

Abdelali Bentahila and Eirlys E Davies. 1983. The syntax of arabic-french code-switching. *Lingua*, 59(4):301–330.

James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. 2011. Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain*, pages 2546–2554.

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).

Ali Farghaly and Khaled Shaalan. 2009. Arabic natural language processing: Challenges and solutions. *ACM Transactions on Asian Language Information Processing (TALIP)*, 8:1–.

Maram Hasanain, Firoj Alam, Hamdy Mubarak, Samir Abdaljalil, Wajdi Zaghouani, Preslav Nakov, Giovanni Da San Martino, and Abed Alhakim Freihat. 2023. ArAIEval Shared Task: Persuasion Techniques and Disinformation Detection in Arabic Text. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore. Association for Computational Linguistics.

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.

Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. 2018. Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization. *Journal of Machine Learning Research*, 18(185):1–52.

Ammar Mohammed and Rania Kora. 2023. A comprehensive review on ensemble deep learning: Opportunities and challenges. *Journal of King Saud University - Computer and Information Sciences*, 35(2):757–774.

Hamdy Mubarak, Samir Abdaljalil, Azza Nassar, and Firoj Alam. 2023a. Detecting and identifying the reasons for deleted tweets before they are posted. *Frontiers in Artificial Intelligence*, 6.

Hamdy Mubarak, Samir Abdaljalil, Azza Nassar, and Firoj Alam. 2023b. Detecting and reasoning of deleted tweets before they are posted. *ArXiv preprint*, abs/2305.04927.

Hamdy Mubarak, Sabit Hassan, and Shammur Absar Chowdhury. 2022. Emojis as anchors to detect arabic offensive language and hate speech.

Kai Shu, Amrita Bhattacharjee, Faisal Hammad Alatawi, Tahora H. Nazer, Kaize Ding, Mansooreh Karami, and Huan Liu. 2020. Combating disinformation in a social media age. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10.

Gadi Wolfsfeld, Elad Segev, and Tamir Sheafer. 2013. Social Media and the Arab Spring: Politics Comes First. *The International Journal of Press/Politics*, 18(2):115–137. Publisher: SAGE Publications Inc.

Zhi-Hua Zhou. 2012. *Ensemble Methods: Foundations and Algorithms*, 1st edition. Chapman & Hall/CRC.

A Appendix

A.1 Experimental Results

In this section, we present our continued experimentation. Instead of including all the models in the search space, individual hyperparameter optimizations were conducted on each model. This resulted in hyperparameters that differed from those in our original experiments. These results are displayed below in tables 5 and 6. Macro precision, recall, F1-score, and accuracy are reported.

In task 2A, QARIB secures the second-highest precision on the validation set and the highest on the test set, suggesting its proficiency in accurately identifying positive classes and minimizing false positives. AraBERTv02-Twitter leads recall for both validation and test sets, indicating its strength in identifying actual positive instances. Both QARIB and AraBERTv02-Twitter demonstrate robust performance, leading in various metrics.

For task 2B, AraBERTv02-Twitter continues its strong performance, showing the highest precision on the test set. Meanwhile, AraBERT-Covid19 achieves the highest recall and F1-score across both sets, indicating a balanced strength in precision and recall, closely followed by MARBERTv2.

The results underscore that no single model consistently outperforms across all metrics, suggesting that model selection should consider the specific performance metrics of interest. The varied leadership in different metrics across both tasks implies a lack of a universally superior model.

Ultimately, our findings revealed a distinct set of optimal parameters divergent from those in our original search space, which encompassed all candidate models. The specifics of these parameters are detailed in tables 7 and 8. Interestingly, for task 2A, the AraBERT-Covid19 model exhibited superior performance with parameters derived from our initial, more generalized search space, as opposed to those obtained from a model-specific search. In contrast, for task 2B, the AraBERTv02-Twitter model demonstrated enhanced performance when employing parameters from a search space tailored for that specific model.

A.2 Dialect Language Identification

For Arabic dialect language detection, we used the “bert-base-arabic” model (Inoue et al., 2021) provided by CAMel (Computational Approaches to Modeling Language) Laboratory on the Hugging-Face Hub [7] trained on MADAR (Bouamor et al., 2018) Twitter dataset which contains Arabic dialect tweets originating from 25 regions. We show in Figure 1 and Figure 2 the distribution of dialects in the Training and Development Sets for tasks 2A and 2B.

The top three dialects used in the provided data are from Saudi Arabia, Kuwait and Egypt. While

[7] https://huggingface.co/CAMeL-Lab/bert-base-arabic-camelbert-msa-did-madar-twitter5

Model	Validation				Test			
	Prec.	Recall	F1	Acc.	Prec.	Recall	F1	Acc.
AraBERT-Covid19	85.73%	83.31%	84.44%	**90.83%**	86.94%	84.10%	85.39%	89.89%
AraBERTv02-Twitter	83.72%	**85.32%**	**84.48%**	90.31%	86.99%	**86.96%**	**86.98%**	**90.64%**
QARIB	85.01%	82.79%	83.83%	90.45%	**87.81%**	84.77%	86.14%	90.43%
MARBERTv2	**86.14%**	81.52%	83.54%	90.59%	86.68%	82.66%	84.40%	89.38%
CAMeLBERT-DA SA	84.63%	81.37%	82.85%	90.02%	86.48%	83.51%	84.85%	89.54%

Table 5: Task 2A hyperparameter optimized models post-hoc comparison of macro validation and test metrics. Highest values are in **bold**.

Model	Validation				Test			
	Prec.	Recall	F1	Acc.	Prec.	Recall	F1	Acc.
AraBERT-Covid19	83.15%	**78.91%**	**80.79%**	84.13%	73.36%	**74.29%**	**73.55%**	81.85%
AraBERTv02-Twitter	**85.99%**	75.30%	79.15%	83.88%	**75.84%**	71.39%	73.21%	**82.65%**
QARIB	85.40%	75.45%	78.49%	**84.38%**	73.97%	72.65%	71.92%	81.85%
MARBERTv2	83.51%	75.53%	78.27%	83.38%	74.79%	73.57%	73.54%	82.08%
CAMeLBERT-DA SA	78.78%	76.31%	76.90%	83.12%	70.14%	73.14%	70.28%	80.02%

Table 6: Task 2B hyperparameter optimized models post-hoc comparison of macro validation and test metrics. Highest values are in **bold**.

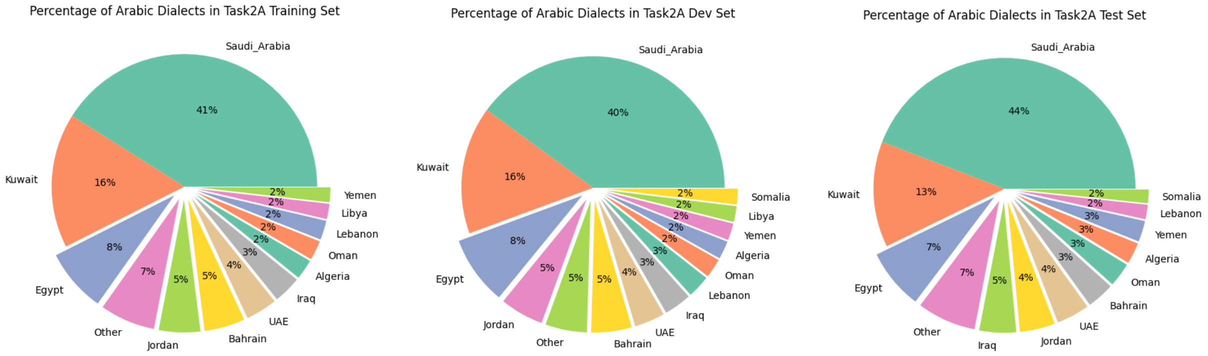
Model	Learning Rate	Batch Size	Dropout	Max Length
AraBERT-Covid19	1.38e-05	32	0.325	115
AraBERT-Covid19 *	1.0e-05	8	0.375	78
AraBERTv02-Twitter	1.74e-05	64	0.0	79
QARIB	1.73e-05	32	0.15	94
MARBERTv2	1.03e-05	64	0.5	99
CAMeLBERT-DA SA	1.62e-05	16	0.0	67

Table 7: Task 2A best hyperparameters for each model, determined post-hoc. Models marked with an asterisk (*) indicate the hyperparameters of the task submitted model.

Model	Learning Rate	Batch Size	Dropout	Max Length
AraBERT-Covid19	1.14e-05	8	0.25	88
AraBERTv02-Twitter	2.82e-05	32	0.2	100
AraBERTv02-Twitter*	5.0e-05	64	0.4	57
QARIB	2.00e-05	32	0.4	93
MARBERTv2	1.17e-05	8	0.1	60
CAMeLBERT-DA SA	1.32e-05	32	0.125	91

Table 8: Task 2B best hyperparameters for each model, determined post-hoc. Models marked with an asterisk (*) indicate the hyperparameters of the task submitted model.

the top three represent about 65% and 64% of the datasets for Task 2A, their percentage drops off particularly in task 2B Development set to 60% whereas the task 2B Training set is still at 65%. Thus, dialect-wise task 2B showed much more variation. The high concentrations of specific dialects imply that our models are significantly influenced by the linguistic features of Saudi Arabia, Kuwait, and Egypt. Upon reviewing the generalization error in Table 6, which compares the validation to testing set metrics, we hypothesize that this dialect variance may adversely affect model generalization. Such variance can introduce additional complexity and nuance to the classification task. When training a language model on a dataset largely influenced by three dialects and then tests it on a broader dialectal range, the model may find it challenging to generalize effectively.

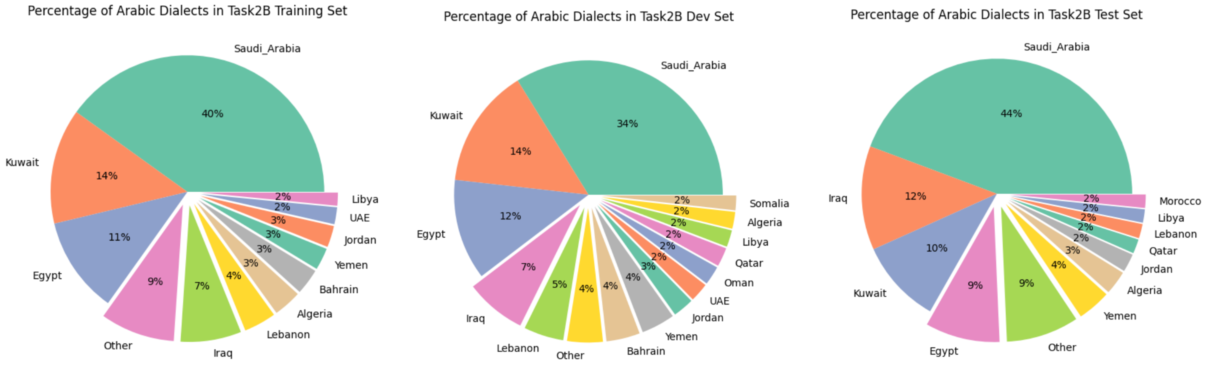


(a) Training Set

(b) Development Set

(c) Test Set

Figure 1: Task 2A - Arabic Dialect Language Identification



(a) Training Set

(b) Development Set

(c) Test Set

Figure 2: Task 2B - Arabic Dialect Language Identification

HTE at ArAIEval Shared Task: Integrating Content Type Information in Binary Persuasive Technique Detection

Hadjer Khaldi
Geotrend
Toulouse, France
hadjer@geotrend.fr

Taqiy Eddine Bouklouha
SolutionData Group
Toulouse, France
tbouklouha@solutiondatagroup.fr

Abstract

Propaganda frequently employs sophisticated persuasive strategies in order to influence public opinion and manipulate perceptions. As a result, automating the detection of persuasive techniques is critical in identifying and mitigating propaganda on social media and in mainstream media. This paper proposes a set of transformer-based models for detecting persuasive techniques in tweets and news that incorporate content type information as extra features or as an extra learning objective in a multitask learning setting. In addition to learning to detect the presence of persuasive techniques in text, our best model learns specific syntactic and lexical cues used to express them based on text genre (type) as an auxiliary task. To optimize the model and deal with data imbalance, a focal loss is used. As part of ArabicNLP2023-ArAIEval shared task, this model achieves the highest score in the shared task 1A out of 13 participants, according to the official results, with a micro-F1 of 76.34% and a macro-F1 of 73.21% on the test dataset. [1]

1 Introduction

In an era marked by the proliferation of information via digital platforms, separating fact from fiction has become an increasingly difficult task, nearly impossible to achieve manually. News and social media platforms are effective tools for disseminating information, but they also serve as breeding grounds for propaganda, misinformation, and manipulation. Propaganda messages can be used to influence people's opinions, beliefs, and behaviours by appealing to their emotions or by using persuasive techniques and arguments that may sound convincing but are based on faulty logic and thus invalid. To combat this, persuasive technique detection has emerged as an important component in the fight against deceptive content.

Most studies in this field focus on one genre of textual content for detecting persuasive techniques (Da San Martino et al., 2020; Barrón-Cedeño et al., 2019; Dimitrov et al., 2021; Carik and Yeniterzi, 2021; Alam et al., 2022). Handling multi-genre text has received little attention.

In this paper, we concentrate on the automatic detection of persuasive techniques in tweets and news. We propose various transformer-based systems for detecting persuasive techniques that implicitly and explicitly utilize content type to enhance detection in multi-genre text. As part of the ArabicNLP2023-ArAIEval shared task (Hasanain et al., 2023a), the task in which we participate (task 1A) involves a collection of Arabic tweets and news paragraphs annotated to indicate the presence or absence of persuasive content.

The rest of the paper is organized as follows: Section 2 gives an overview of related work. In Section 3, we present the data used. The proposed system is described in Section 4. In Section 5, we provide the details of our experiments, and then the results for our official runs are presented in Section 6. In Section 7, a discussion of the results is presented. We conclude the paper in Section 8.

2 Related Work

Over the past few years, there has been an increase in concern about the spread of opinion-shaping news and misinformation, particularly in the context of critical events such as COVID-19, elections, and conflicts. As a result, identifying propaganda content and persuasive techniques has gained more importance.

Research on propaganda content detection has targeted various media platform contents, including news (Da San Martino et al., 2020; Barrón-Cedeño et al., 2019), memes (Dimitrov et al., 2021), and tweets (Carik and Yeniterzi, 2021; Alam et al., 2022; Vijayaraghavan and Vosoughi, 2022; Mubarak et al., 2023).

[1] Code available at https://github.com/TaqiyEddine-B/Transformers-for-Propaganda-Detection

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 502–507
December 7, 2023 ©2023 Association for Computational Linguistics

With the introduction of Transformer models (Vaswani et al., 2017), the detection of propaganda and fake news has seen significant improvement in performance. Some works relied solely on real-world data to fine-tune a pre-trained language model for propaganda and persuasive technique detection (Costa et al., 2023), whereas others combined it with synthetically augmented data (Hasanain et al., 2023b). The ensemble approach was also investigated, in which various combined pre-trained language models are fine-tuned in a vanilla setting (Purificato and Navigli, 2023), or by using adapters (Wu et al., 2023).

One major limitation of all the preceding works is that they focus on one type of media platform content at a time. In this paper, we investigate the detection of persuasive techniques from multi-genre text extracted from tweets and news articles. The task is made more difficult by the differences in writing styles and contexts in both texts.

In line with the assumption proposed by (Barrón-Cedeño et al., 2019), which affirms that sentence representations incorporating information about writing style tend to exhibit better generalization than word-level representations in news propaganda detection, we delve into the integration of content type (genre) information within transformer-based models for the detection of persuasive techniques.

3 Data

Our data comes from the proposed dataset for persuasion technique detection as part of the ArAIEval 2023 shared task 1A. [2] No additional data was used. Each entry in the data file is composed of three fields: `text` referring to the textual content, `type` referring to the genre of text: tweet or news, and `label` referring to the presence or absence of persuasive technique in the text: True or False.

Table 1 describes the data distribution per `type` and per `label`. Overall, we can notice that dataset is imbalanced in terms of label and type distributions. Texts of type news paragraph are over-represented in the dataset, representing 65% compared to tweets (35%). Then, texts using persuasive techniques are more prevalent (79%) than non-persuasive content (21%%).

	TRUE	FALSE	**#Total**
Paragraph	1201 (76%)	374 (24%)	1575 (65%)
Tweet	717 (84%)	135 (16%)	852 (35%)
#Total	1918 (79%)	509 (21%)	2427

Table 1: Data distribution in train dataset per label and content type.

When comparing persuasive tweets and news paragraphs, the context used to express news paragraphs is twice as long as the context used to express tweets, with the average length of news being 211 characters compared to 100 characters for tweets. We expect that the length of the context will influence the syntactic and lexical cues used to express persuasive techniques in news paragraphs and tweets. We should point out that no text pre-processing was done on the dataset for training.

4 System Overview

The system's goal is to determine whether a multi-genre (a tweet or news paragraph) snippet contains persuasive content. Our proposed system (cf. Figure 1) is made up of : **(1)** a transformer-based encoder Enc_i (Vaswani et al., 2017) that encodes the input texts into a fixed-size contextualized vector, **(2)** followed by a feature injection layer ($Feat$) that concatenates the content type vector with the input vector, and **(3)** two parallel classifiers C, each of which is made up of a fully-connected layer, a dropout layer, and an activation layer, and perform two different tasks:

– C_{main}: a classifier that performs the main task that learns to recognize the presence of persuasive techniques in texts (binary classification).

– C_{aux}: a classifier that performs an auxiliary (support) task that learns to identify the type of text: tweet or news (binary classification).

Each task calculates one loss, and optimizing the model optimizes the sum of the two losses. Because the two tasks share the same encoder, the auxiliary task can help the main task learn additional specific syntactic and lexical cues for tweets or news content used to express persuasive arguments.

According to recent research, jointly learning common characteristics shared across multiple tasks can have a significant impact on NLP classification performances as it enhances the perfor-

[2] ArAIEval dataset for persuasion technique detection

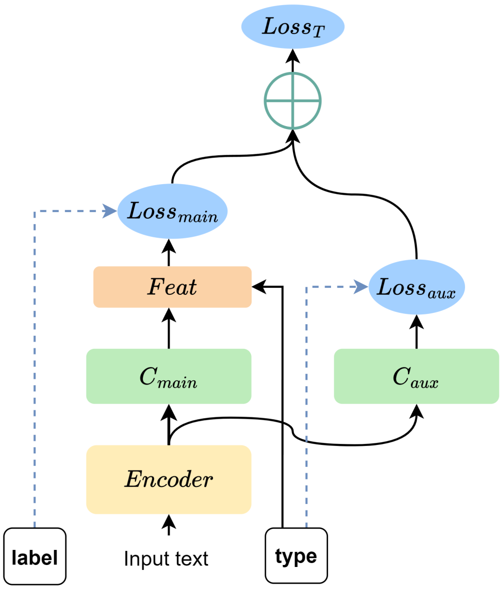


Figure 1: Proposed System architecture.

mance of the main task by incorporating other related tasks, making it easier to combine information from multiple resources (Ye et al., 2019; Liu et al., 2017; He et al., 2019; Khaldi et al., 2022; Tafreshi and Diab, 2018).

5 Experimental Setup

We experiment with different configurations for the proposed system components. We evaluated two transformer-based encoders: MARBERT (noted M) (Abdul-Mageed et al., 2021) and AraBERT (noted A) (Abdul-Mageed et al., 2021), noting Enc_i with $i \in \{M, A\}$. Because $Feat$ directly injects the text type into the system as a feature and C_{aux} learns to predict it, it is not possible to enable both $Feat$ and C_{aux} in a model, thus enabling one disables the other. The resulted systems from various configurations are shown below:

– Enc_M+C_{main} and $Enc_A + C_{main}$, that we consider as baseline models, that perform a classical binary classification based on the input vector, in a monotask setting without any additional features.

– Enc_M+$Feat$+C_{main} and Enc_A+$Feat$+C_{main} that additionally inject the content type (genre) as a feature alongside the input representation.

– Enc_M+C_{main}+C_{aux} and Enc_A+C_{main}+C_{aux} that perform two binary classification tasks, namely: persuasive technique detection as a main task and type detection as an auxiliary task.

A cross-entropy loss (noted CE) is used to optimize the systems. We also experiment with a focal loss (Lin et al., 2017) to deal with data imbalance in the train data, as it has been shown to be effective in many imbalanced NLP classification problems (Liu et al., 2021; Ma et al., 2020; Huang et al., 2021). Train data is used to fine-tune all systems. The development data is used to fine-tune the model's hyperparameters, where the best ones are reported in Table 2. We evaluate the fine-tuned models on the development dataset, the official micro-F1 score is shown in Table 3. The best performing one was selected to be submitted for the official ranking on the test set.

Hyperparameter	Value
learning_rate	2e^{-5}
epochs	5
batch_size	16

Table 2: Best Hyperparameters after fine-tuning on development dataset.

6 Results

In general, we found that explicitly or implicitly incorporating content type information into the system could improve overall results for both MARBERT and AraBERT encoders when either cross-entropy or focal loss was used. For example, both Enc_M+$Feat$+C_{main} and Enc_M+C_{aux}+C_{main} beat Enc_M+C_{main}, with almost + 2% and + 1% on micro-F1. Among the twelve evaluated system configurations, Enc_A+C_{main}+C_{aux} optimized using a focal loss represents our best performing one during the development phase, achieving an increase of nearly 3% above baselines. This model was submitted for official ranking on the test dataset for *task 1A*, and the obtained micro-F1 is 0.7634, which is the highest score on the leaderboard for this task.

7 Discussion

The test dataset contains 503 inputs, of which 119 are classified incorrectly. Out of these, 103 are paragraphs (87%) and 16 are tweets (13%).

A closer examination of the confusion matrices per content type for the best-performing system configuration (cf. figure 3 for tweet text and figure 2 for news paragraphs) reveals that the majority of tweet misclassifications (75%) involve non-persuasive content being mistaken for per-

Models	CE	Focal
Enc_M+C_{main}	0.8301	0.8263
Enc_A+C_{main}	0.8108	0.8301
Enc_M+$Feat$+C_{main}	0.8533	0.8417
Enc_A+$Feat$+C_{main}	0.8108	0.8571
Enc_M+C_{main}+C_{aux}	0.8378	**0.8610** †
Enc_A+C_{main}+C_{aux}	0.8147	0.8340

Table 3: Evaluation of proposed system configurations on the development dataset of *task 1A*. Official metric micro-F1 is reported. The best result is in bold and † marks the system submitted for the official ranking.

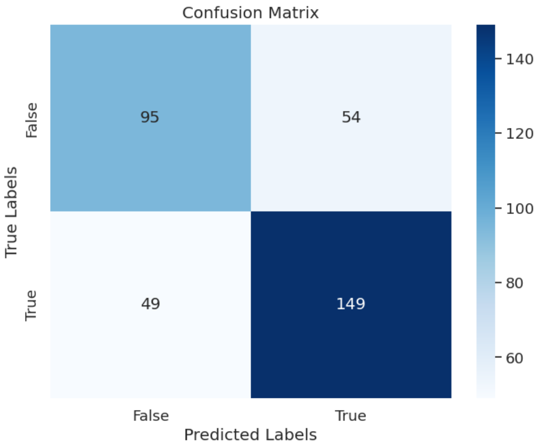


Figure 2: Confusion matrix for paragraph news.

suasive content. This could be explained by the over-representation of persuasive content in tweets (84%). In contrast, the misclassification rate of persuasive or non-persuasive texts in news paragraphs is nearly identical.

8 Conclusion

In this paper, we present our experiments and findings on the detection of persuasive techniques in multi-genre texts, which encompass tweets and news paragraphs. This research was part of the ArabicNLP2023-ArAIEval Task1A shared task, focusing on identifying persuasive techniques through binary classification. Our team proposed a system based on fine-tuning a transformer-based model to assess the impact of integrating content type information on persuasive technique detection. We experimented with two different approaches to information integration: implicitly, by adding an additional learning objective to the model, or explicitly, as an additional feature.

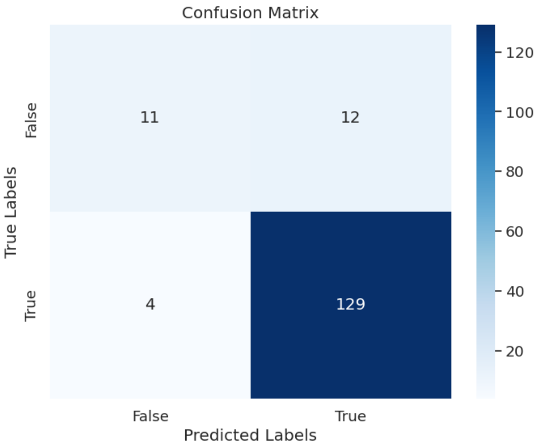


Figure 3: Confusion matrix for tweets.

We evaluated two pre-trained language models for Arabic and optimized the system using two different loss functions: cross-entropy and focal loss to address data imbalance. Our highest scores on the development dataset were achieved by the MARBERT model, trained using focal loss, for both persuasive technique detection and content type detection. As a result, our model secured the first position in the ArabicNLP2023-ArAIEval Task1A shared task during the test phase.

Our future work will involve exploring data augmentation techniques to address data imbalance and integrating multiple pre-trained language models. Finally, our results indicate that our system faces challenges in identifying persuasive content in news paragraphs. To pinpoint the causes of misclassifications, a deeper investigation into incorrectly classified sentences is warranted.

Limitations

Firstly, when applied to Arabic text, incorporating content type information as a new learning objective in a persuasive technique detection task yielded satisfactory results. However, it's important to note that this outcome may not necessarily hold true for other languages; extensive testing is required to confirm the results across different linguistic contexts.

Additionally, the distribution of data in terms of content type may exert an influence on the task of content type identification. It's worth highlighting that a significant imbalance between the two types, or considering more than two content types in the dataset, can potentially impact the overall results.

References

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2021. ARBERT & MARBERT: Deep bidirectional transformers for Arabic. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7088–7105, Online. Association for Computational Linguistics.

Firoj Alam, Hamdy Mubarak, Wajdi Zaghouani, Giovanni Da San Martino, and Preslav Nakov. 2022. Overview of the WANLP 2022 shared task on propaganda detection in Arabic. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 108–118, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Alberto Barrón-Cedeño, Israa Jaradat, Giovanni Da San Martino, and Preslav Nakov. 2019. Proppy: Organizing the news based on their propagandistic content. *Information Processing Management*, 56(5):1849–1864.

Buse Carik and Reyyan Yeniterzi. 2021. SU-NLP at CheckThat! 2021: Check-worthiness of Turkish tweets.

Nelson Filipe Costa, Bryce Hamilton, and Leila Kosseim. 2023. CLaC at SemEval-2023 task 3: Language potluck RoBERTa detects online persuasion techniques in a multilingual setup. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1613–1618, Toronto, Canada. Association for Computational Linguistics.

Giovanni Da San Martino, Alberto Barrón-Cedeno, Henning Wachsmuth, Rostislav Petrov, and Preslav Nakov. 2020. SemEval-2020 task 11: Detection of propaganda techniques in news articles. In *Proceedings of the 14th Workshop on Semantic Evaluation*, SemEval '20, pages 1377–1414.

Dimitar Dimitrov, Bishr Bin Ali, Shaden Shaar, Firoj Alam, Fabrizio Silvestri, Hamed Firooz, Preslav Nakov, and Giovanni Da San Martino. 2021. SemEval-2021 task 6: Detection of persuasion techniques in texts and images. In *Proceedings of the International Workshop on Semantic Evaluation*, SemEval '21, pages 70–98.

Maram Hasanain, Firoj Alam, Hamdy Mubarak, Samir Abdaljalil, Wajdi Zaghouani, Preslav Nakov, Giovanni Da San Martino, and Abdelhakim Freihat. 2023a. Araieval shared task: Persuasion techniques and disinformation detection in arabic text. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore. Association for Computational Linguistics.

Maram Hasanain, Ahmed El-Shangiti, Rabindra Nath Nandi, Preslav Nakov, and Firoj Alam. 2023b. QCRI at SemEval-2023 task 3: News genre, framing and persuasion techniques detection using multilingual models. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1237–1244, Toronto, Canada. Association for Computational Linguistics.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2019. An interactive multi-task learning network for end-to-end aspect-based sentiment analysis. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 504–515, Florence, Italy. Association for Computational Linguistics.

Yi Huang, Buse Giledereli, Abdullatif Köksal, Arzucan Özgür, and Elif Ozkirimli. 2021. Balancing methods for multi-label text classification with long-tailed class distribution. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8153–8161.

Hadjer Khaldi, Farah Benamara, Camille Pradel, and Nathalie Aussenac Gilles. 2022. A closer look to your business network: Multitask relation extraction from economic and financial french content. In *The AAAI-22 Workshop on Knowledge Discovery from Unstructured Data in Financial Services*.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

Jianyi Liu, Xi Duan, Ru Zhang, Youqiang Sun, Lei Guan, and Bingjie Lin. 2021. Relation classification via bert with piecewise convolution and focal loss. *Plos one*, 16(9):e0257092.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial multi-task learning for text classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1–10, Vancouver, Canada. Association for Computational Linguistics.

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

Hamdy Mubarak, Samir Abdaljalil, Azza Nassar, and Firoj Alam. 2023. Detecting and reasoning of deleted tweets before they are posted. *arXiv preprint arXiv:2305.04927*.

Antonio Purificato and Roberto Navigli. 2023. APatt at SemEval-2023 task 3: The sapienza NLP system for ensemble-based multilingual propaganda detection. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 382–388, Toronto, Canada. Association for Computational Linguistics.

Shabnam Tafreshi and Mona Diab. 2018. Emotion detection and classification in a multigenre corpus with joint multi-task deep learning. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2905–2913, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Prashanth Vijayaraghavan and Soroush Vosoughi. 2022. TWEETSPIN: Fine-grained propaganda detection in social media using multi-view representations. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3433–3448, Seattle, United States. Association for Computational Linguistics.

Ben Wu, Olesya Razuvayevskaya, Freddy Heppell, João A. Leite, Carolina Scarton, Kalina Bontcheva, and Xingyi Song. 2023. SheffieldVeraAI at SemEval-2023 task 3: Mono and multilingual approaches for news genre, topic and persuasion technique classification. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1995–2008, Toronto, Canada. Association for Computational Linguistics.

Wei Ye, Bo Li, Rui Xie, Zhonghao Sheng, Long Chen, and Shikun Zhang. 2019. Exploiting entity BIO tag embeddings and multi-task learning for relation extraction with imbalanced data. In *Proceedings of ACL*, pages 1351–1360.

USTHB at ArAIEval'23 Shared Task: Disinformation Detection System based on Linguistic Feature Concatenation

Mohamed Lichouri
LCPTS-USTHB, Algiers, Algeria
mlichouri@usthb.dz

Khaled Lounnas, Aicha Zitouni
LCPTS-USTHB, Algiers, Algeria
CRSTDLA, Algiers, Algeria
{k.lounnas, a.zitouni}@crstdla.dz

Houda Latrache
CRSTDLA, Algiers, Algeria
h.latrache@crstdla.dz

Rachida Djeradi
LCPTS-USTHB, Algiers, Algeria
rdjeradi@usthb.dz

Abstract

In this research paper, we undertake a comprehensive examination of several pivotal factors that impact the performance of Arabic Disinformation Detection in the ArAIEval'2023 shared task. Our exploration encompasses the influence of surface preprocessing, morphological preprocessing, the FastText vector model, and the weighted fusion of TF-IDF features. To carry out classification tasks, we employ the Linear Support Vector Classification (LSVC) model. In the evaluation phase, our system showcases significant results, achieving an F_1 micro score of 76.70% and 50.46% for binary and multiclass classification scenarios, respectively. These accomplishments closely correspond to the average F_1 micro scores achieved by other systems submitted for the second subtask, standing at 77.96% and 64.85% for binary and multiclass classification scenarios, respectively.

1 Introduction

In recent years, the detection of disinformation in digital content has become a critical challenge at the intersection of natural language processing and information security, spurred by the growing influence of online platforms (Shu et al., 2020). The Arabic-speaking digital landscape, in particular, has witnessed an alarming increase in susceptibility to the dissemination of false or misleading information, a phenomenon well-documented in recent research (Harrag and Djahli, 2022). The ramifications of disinformation extend beyond individual deception; they cover broader societal consequences, affecting public opinion, social cohesion, and even national security.

Recognizing the gravity of this issue, we actively participate in the inaugural shared task organized by ArAIEval'2023, which focuses on disinformation detection in Arabic text (Hasanain et al., 2023). Our engagement in this task reflects our commitment to addressing this pressing challenge. By harnessing advanced natural language processing techniques and machine learning models, we endeavor to contribute to the development of effective disinformation detection systems tailored to the nuances of the Arabic language. Through rigorous experimentation and evaluation, we aim to enhance our understanding of the complexities involved and offer practical solutions to safeguard the integrity of digital discourse and information dissemination in the Arabic-speaking world.

To combat the proliferation of disinformation in Arabic text, a growing number of research has been dedicated to developing robust and effective detection systems (Alam et al., 2022; Mubarak et al., 2023). Much like the endeavors undertaken in the field of Arabic dialect identification (Lichouri et al., 2021b), disinformation detection in Arabic requires a nuanced understanding of the language's intricacies (Nagoudi et al., 2020), as well as the ability to sift through vast amounts of textual data (Himdi et al., 2022) to identify instances of deceptive or misleading content.

In this paper, we embark on an extensive exploration of disinformation detection in Arabic, drawing inspiration from the methodologies and techniques employed in previous shared tasks (Lichouri et al., 2020). Leveraging these insights, we aim to build upon existing research and contribute to the ongoing efforts to enhance the accuracy and effectiveness of disinformation detection systems in Arabic text.

Our study encompasses a comprehensive analysis of various factors influencing the performance of Arabic disinformation detection, including surface and morphological preprocessing techniques (Lichouri et al., 2021a), feature engineering strategies (Fouad et al., 2022), and the implementation of

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 508–512
December 7, 2023 ©2023 Association for Computational Linguistics

state-of-the-art machine learning models. Through rigorous experimentation and evaluation, we seek to provide valuable insights and practical solutions that can aid in the identification and mitigation of disinformation.

This paper is organized as follows: Section 2 offers insights into the dataset we have employed. Moving on to Section 3, we introduce our proposed system, which includes details about the cleaning and preprocessing steps discussed in Section 3.1. The process of feature engineering is elucidated in Section 3.2. Section 3.3 is dedicated to a comprehensive discussion of our findings. Finally, we wrap up the paper in Section 4 with a conclusive summary of our contributions and key findings.

2 Description of the Dataset

A disinformation dataset constitutes a crucial resource for studying and comprehending the multifaceted landscape of misinformation, misleading content, and fabricated information within various digital platforms. Such datasets encompass a diverse array of textual, visual, and multimedia content intentionally designed to deceive, mislead, or manipulate audiences. These datasets serve as invaluable assets for researchers, data scientists, and machine learning practitioners engaged in the development of advanced algorithms and models aimed at detecting, analyzing, and combating disinformation. By analyzing patterns, linguistic cues, and contextual elements within disinformation datasets, researchers gain insights into the tactics, strategies, and evolving nature of disinformation campaigns, thereby contributing to the enhancement of society's ability to discern and mitigate the harmful impacts of deceptive content in an increasingly interconnected information landscape.

Additional information regarding this dataset can be found in Table 1, where we took part for the first time this year in both editions of the Disinformation Detection Definition shared task. This task involves classifying binary and fine-grained disinformation categories based solely on the text of a tweet. Please note that these statistics pertain to the dataset after we removed punctuation and emojis. Imbalanced datasets can have a pronounced effect on system performance, causing the development of biased models that prioritize the dominant class (e.g., "no-disinformation" in binary classification and "HS" in multi-class classification). This can result in decreased predictive accuracy for the underrepresented classes, such as "disinformation" in binary classification, "Rumor", and "Spam" in the multi-class scenario, and compromised decision-making in applications like fraud detection or medical diagnosis. Addressing class imbalance through techniques like oversampling, undersampling, or using appropriate evaluation metrics is crucial for more equitable and accurate model outcomes.

3 Proposed system

3.1 Data Cleaning and Preprocessing

In the challenging domain of disinformation detection within Arabic text, it becomes imperative to adeptly capture essential information while efficiently removing undesirable elements. This task is known for its complexity and nuance, demanding a detailed approach. To address this challenge, we have implemented a two-phase preprocessing strategy:

Phase 1: Surface Preprocessing - In this initial phase, we execute a range of foundational procedures:

- *Arabic Letter Normalization*: Ensuring consistency in Arabic script characters (Sallam et al., 2016).
- *Punctuation and Emoji Removal*: Eliminating punctuation marks and emoticons (Shiha and Ayvaz, 2017).
- *Stop Words Removal*: Handling common words that do not contribute substantially to meaning.
- *Diacritics Removal*: Removing diacritical marks for text clarity (Jbara et al., 2009).
- *Exclusion of Non-Arabic Content*: Ensuring that only Arabic text remains (Omar et al., 2021).

These collective measures ensure text clarity, uniformity, and the removal of any distractions.

Phase 2: Morphological Preprocessing - In this phase, our focus shifts to the intricacies of language. Here, we employ the following techniques:

- *Lemmatization*: Simplifying word forms to their base or dictionary form (El Kah and Zeroual, 2021).
- *Stemming*: Reducing words to their root forms, aiding in the identification of core word meanings and structures (Atwan et al., 2021).

Table 1: ArAIEval (Task2A/2B) dataset statistics where : Task2A for Binary classification whereas Task2B for Multiclass classification problem.

	Train	Dev	Test
# sentences	14147/2656	2115/397	3729/876
# words	324727/68073	48917/10062	100646/27312
Max # word per sentence	65/67	65/59	62 /62
Min # word per sentence	0 / 1	0 / 1	1 / 1
Max # char per sentence	280/290	280 /285	311 /311
Min # char per sentence	0 / 3	0 / 3	2 / 2

Table 2: The various combinations and parameter used in our work

Settings	**Range**
ngram_range	(m,n) with m=1 to 3 and n=1 to 10
tfidf_weights	0.5 - 1
tfidf max_features	1000 -25000
SVM	C=100, gamma=1-10
fasttext_supervised	epoch=100, loss=’ova’

fasttext_unsupervised	epoch=100, ws=6
	model=’skipgram’
	dim=1000

Throughout both phases, we intricately harmonize and fine-tune various techniques to arrive at the optimal configuration for our preprocessing pipeline.

3.2 Feature engineering

Our system operates through a well-defined structure consisting of four distinct phases, offering the flexibility to be applied individually or collectively. The initial two phases, Surface Preprocessing and Morphological Preprocessing, have been expounded upon in the previous section. The subsequent phases are detailed as follows:

Phase 3: Feature Extraction - In this stage, we employ a dual-model approach. Firstly, the FastText model undergoes comprehensive training in two modes: supervised and unsupervised, drawing from the training dataset. Then, we use this model to extract features from both the development and test datasets. Secondly, we leverage the TF-IDFVectorizer, an adept tool offering three distinct analyzers (Word, Char, and Char_wb), each encompassing variable n-gram ranges. As a default configuration, we combine these three TF-IDF features, affording them equal weights, all set to 1.

Phase 4: Weighted Fusion - In this phase, we combined the three TF-IDF features, supported by a weight vector featuring three distinctive values (w1, w2, w3) that correspond to the Word, Char, and Char_wb TF-IDF features, respectively.

Having presented these four distinctive phases, we executed four designed experiments that were inspired by our prior works (Lichouri et al., 2018; Abbas et al., 2019; Lichouri and Abbas, 2020a), where each embody distinct configurations:

Experiment 1 (Lichouri et al., 2021a; Lichouri and Abbas, 2020b): In this first experiment, we initiated with the first phase, by considering all the possible permutations of surface processing techniques. Following this, we considered the third phase, marked by the employment of a union of TF-IDF features. During the feature extraction process, we explored a range of n-gram values, spanning from $n = 1$ to 10. Finally, we finished by the training of the SVC classifier.

Experiment 2 (Lichouri et al., 2020): In this specific scenario, we worked with the second phase, by exploring various combinations of morphological processing techniques. Similar to Experiment 1, we progressed to the third phase, where we concat the TF-IDF features, all while varying the n-gram parameters. We then finished this experiment by training of the SVC classifier.

Experiment 3: For this unique experiment, we focused on the third phase, where we used FastText model for feature extraction, followed by the rigorous training of the SVC classifier.

Experiment 4 (Lichouri et al., 2021b): In this distinctive scenario, we executed the fourth phase, by applying a weighted union of TF-IDF features for feature extraction. Then, we concluded with

Task	**Binary**				**Multiple**			
Desc	MP	SP	F-Vec	WF	MP	SP	F-Vec	WF
Run 1	81,08	**81,23**	48,45	81,13	56,92	**57,43**	27.57	56,93
Run 2	81,08	81,18	48.27	78,91	56,92	56,68	27.68	56,92
Run 3	81,08	81,09	46.54	75,74	56,92	56,93	22.44	56,68

Table 3: The F1-micro percentages obtained using the proposed system Where: SP (Surface Preprocessing), MP (Morphological Preprocessing), F-Vec (Vectorisation), and WF (Weighted Fusion)

the training of the SVC classifier.

Following many iterations of these four experiments on both the training and development datasets, we recorded the best results attained for each experiment, along with the precise configurations that yielded these outcomes, as presented in Table 2.

3.3 Results and Discussion

In this study, we conducted a series of experiments aimed at detecting Arabic disinformation. These experiments were centered around the utilization of various descriptors, encompassing Surface Preprocessing (SP), Morphological Preprocessing (MP), the vectorisation model (F-Vec), and Weighted Fusion of TF-IDF (WF).

To explore the effectiveness of these descriptors, we employed a range of combinations and settings. This involved modifying n-gram values and TF-IDF weights to investigate the impact of word sequence length on results and term weighting in the text, respectively. Table 2 provides a comprehensive summary of the different combinations and parameters used in our study, while Table 3 presents the results obtained using these combinations.

Our experiments yielded valuable insights into the efficacy of various techniques for disinformation detection, specifically in binary and multiclass classification tasks. Notably, for the binary subtask, Surface Preprocessing demonstrated the highest performance, achieving an impressive F1-score of 81.23%. It was closely followed by the Weighted Union of TF-IDF features, with an F1-score of 81.13%, while Morphological Preprocessing exhibited slightly lower performance, resulting in an F1-score of 81.08%. Intriguingly, the FastText model underperformed in this context, attaining the lowest F1-score at 48.45%.

However, a fascinating observation emerged when we transitioned to the multiclass classification subtask. Surprisingly, the same observation held true, but the obtained results dropped significantly, by approximately 20%, compared to the binary case. We hypothesize that this decline in performance could be attributed to the imbalanced nature of the dataset, which has a more pronounced impact in the multiclass scenario.

4 Conclusion

In conclusion, our comprehensive analysis of key factors in Arabic Disinformation Detection has shed light on critical aspects that significantly influence performance. Through a meticulous exploration of surface preprocessing, morphological preprocessing, the FastText vector model, and the weighted fusion of TF-IDF features, we have gained valuable insights into their impact on classification tasks.

Our system's noteworthy achievement of an F_1 micro score of 76.70% and 50.46% for binary and multiclass classification setups, respectively, closely aligns with the performance of other systems submitted for the second subtask. This not only reaffirms the significance of surface preprocessing and weighted TF-IDF feature fusion but also positions them as robust techniques in the domain of Arabic Disinformation Detection.

References

Mourad Abbas, Mohamed Lichouri, and Abed Alhakim Freihat. 2019. St madar 2019 shared task: Arabic fine-grained dialect identification. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 269–273.

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, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Jaffar Atwan, Mohammad Wedyan, Qusay Bsoul, Ahmad Hamadeen, Ryan Alturki, and Mohammed

Ikram. 2021. The effect of using light stemming for arabic text classification. *International Journal of Advanced Computer Science and Applications*, 12(5).

Anoual El Kah and Imad Zeroual. 2021. The effects of pre-processing techniques on arabic text classification. *Int. J*, 10(1):1–12.

Khaled M Fouad, Sahar F Sabbeh, and Walaa Medhat. 2022. Arabic fake news detection using deep learning. *Computers, Materials & Continua*, 71(2).

Fouzi Harrag and Mohamed Khalil Djahli. 2022. Arabic fake news detection: A fact checking based deep learning approach. *Transactions on Asian and Low-Resource Language Information Processing*, 21(4):1–34.

Maram Hasanain, Firoj Alam, Hamdy Mubarak, Samir Abdaljalil, Wajdi Zaghouani, Preslav Nakov, Giovanni Da San Martino, and Abdelhakim Freihat. 2023. Araieval shared task: Persuasion techniques and disinformation detection in arabic text. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore. Association for Computational Linguistics.

Hanen Himdi, George Weir, Fatmah Assiri, and Hassanin Al-Barhamtoshy. 2022. Arabic fake news detection based on textual analysis. *Arabian Journal for Science and Engineering*, 47(8):10453–10469.

Khitam Mahmoud Abdalla Jbara, Azzam T Sleit, and Bassam H Hammo. 2009. *Knowledge discovery in Al-Hadith using text classification algorithm.* University of Jordan.

Mohamed Lichouri and Mourad Abbas. 2020a. Simple vs oversampling-based classification methods for fine grained arabic dialect identification in twitter. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 250–256.

Mohamed Lichouri and Mourad Abbas. 2020b. Speechtrans@ smm4h'20: Impact of preprocessing and n-grams on automatic classification of tweets that mention medications. In *Proceedings of the Fifth Social Media Mining for Health Applications Workshop & Shared Task*, pages 118–120.

Mohamed Lichouri, Mourad Abbas, and Besma Benaziz. 2020. Profiling fake news spreaders on twitter based on tfidf features and morphological process.

Mohamed Lichouri, Mourad Abbas, Besma Benaziz, Aicha Zitouni, and Khaled Lounnas. 2021a. Preprocessing solutions for detection of sarcasm and sentiment for Arabic. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 376–380, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Mohamed Lichouri, Mourad Abbas, Abed Alhakim Freihat, and Dhiya El Hak Megtouf. 2018. Word-level vs sentence-level language identification: Application to algerian and arabic dialects. *Procedia Computer Science*, 142:246–253.

Mohamed Lichouri, Mourad Abbas, Khaled Lounnas, Besma Benaziz, and Aicha Zitouni. 2021b. Arabic dialect identification based on a weighted concatenation of TF-IDF features. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 282–286, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Hamdy Mubarak, Samir Abdaljalil, Azza Nassar, and Firoj Alam. 2023. Detecting and identifying the reasons for deleted tweets before they are posted. *Frontiers in Artificial Intelligence*, 6.

El Moatez Billah Nagoudi, AbdelRahim Elmadany, Muhammad Abdul-Mageed, Tariq Alhindi, and Hasan Cavusoglu. 2020. Machine generation and detection of arabic manipulated and fake news. *arXiv preprint arXiv:2011.03092*.

Ahmed Omar, Tarek M Mahmoud, Tarek Abd-El-Hafeez, and Ahmed Mahfouz. 2021. Multi-label arabic text classification in online social networks. *Information Systems*, 100:101785.

Rouhia M Sallam, Hamdy M Mousa, and Mahmoud Hussein. 2016. Improving arabic text categorization using normalization and stemming techniques. *Int. J. Comput. Appl*, 135(2):38–43.

Mohammed Shiha and Serkan Ayvaz. 2017. The effects of emoji in sentiment analysis. *Int. J. Comput. Electr. Eng.(IJCEE.)*, 9(1):360–369.

Kai Shu, Amrita Bhattacharjee, Faisal Alatawi, Tahora H Nazer, Kaize Ding, Mansooreh Karami, and Huan Liu. 2020. Combating disinformation in a social media age. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10(6):e1385.

Mavericks at ArAIEval Shared Task: Towards a Safer Digital Space - Transformer Ensemble Models Tackling Deception and Persuasion

Sudeep Mangalvedhekar[*], **Kshitij Deshpande**[*], **Yash Patwardhan**[*],
Vedant Deshpande[*] and **Ravindra Murumkar**[*]
Pune Institute of Computer Technology, Pune
{sudeepm117,kshitij.deshpande7,yash23pat,vedantd41}@gmail.com,
rbmurumkar@pict.edu

Abstract

In this paper, we highlight our approach for the "Arabic AI Tasks Evaluation (ArAiEval) Shared Task 2023". We present our approaches for task 1-A and task 2-A of the shared task which focus on persuasion technique detection and disinformation detection respectively. Detection of persuasion techniques and disinformation has become imperative to avoid distortion of authentic information. The tasks use multigenre snippets of tweets and news articles for the given binary classification problem. We experiment with several transformer-based models that were pre-trained on the Arabic language. We fine-tune these state-of-the-art models on the provided dataset. Ensembling is employed to enhance the performance of the systems. We achieved a micro F1-score of 0.742 on task 1-A (8th rank on the leaderboard) and 0.901 on task 2-A (7th rank on the leaderboard) respectively.

1 Introduction

In today's digital age, numerous platforms aid people in reaching out to the world. However, some individuals resort to disinformation and persuasion techniques to influence people, keeping in mind a certain biased agenda, which can have negative societal effects. Disinformation (Wardle and Derakhshan, 2017) is an intentional effort to disseminate malicious, manipulative, and misleading information for espionage. The propagation of incorrect information can be deleterious to an individual, an organization, or a nation. Hence, disinformation detection has become imperative to catch false reports and avoid social upheaval. Persuasion is the act of changing someone's convictions, views, or conduct through interaction or exchange. Persuasion techniques can be employed to propagate propaganda (Alam et al., 2022) and influence the behavioral patterns of the targeted audience. Persuasion can be done via textual mediums such as news articles and tweets. Social media can act as a key instrument to proliferate persuasive content as well as disinformation among the masses.

With advancements in science and technology, specifically in the domain of machine learning, machines are now capable of detecting persuasion techniques as well as disinformation from the given data. However, the detection techniques have certain limitations. The tactics used to spread disinformation constantly evolve, and the sheer volume is immense. Understanding context and intent is another challenge when it comes to detecting persuasion and disinformation. Detecting and countering these instruments of influence across multiple languages and cultural contexts can be daunting.

This paper demonstrates our work on Task 1 - Persuasion Technique Detection and Task 2 - Disinformation Detection (Hasanain et al., 2023). We intend to examine whether the given multigenre textual snippets contain persuasive content in Task 1 and classify whether the given tweet (Mubarak et al., 2023) is disinformation or not in Task 2. Our approach highlights the use of various transformer-based models for binary classification on the given Arabic data. Ensemble-based techniques have also been employed to yield better results.

2 Related Work

In the pre-internet era, traditional media analysis, fact-checking, and investigative journalism were employed to detect disinformation and persuasion techniques. With the emergence of the internet, keyword-based approaches and sentimental analysis techniques found their groove in detecting fake news and persuading content. An analysis of linguistic features (Conroy et al., 2015), lexical patterns (Feng et al., 2012), and rhetorical structures (Rubin et al., 2015) was used for this purpose. Further advancements in text analysis (Pérez-Rosas

[*] Equal contribution

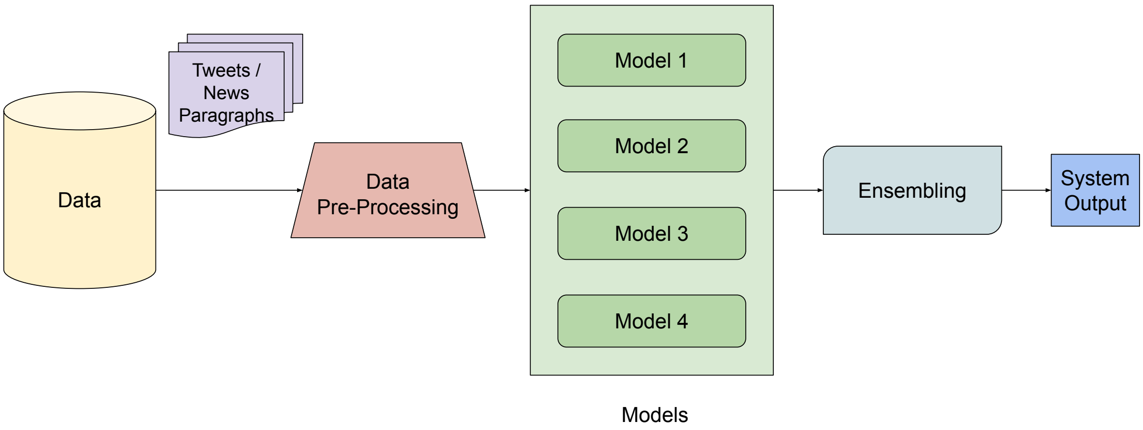


Figure 1: System architecture used for both Task 1 and Task 2

et al., 2017) proved fruitful for this task.

Although mitigating disinformation and persuasion techniques had been a difficult undertaking, machine learning techniques (Manzoor et al., 2019; Khanam et al., 2021) showed promise to address this issue. Supervised machine learning methods (Reis et al., 2019) such as Support Vector Machine(SVM), XGBoost, and Naive Bayes have been used for this purpose. (Iyer and Sycara, 2019) stated that the detection of persuasive tactics in a text can be automated using unsupervised learning. Network analysis methods (Shu et al., 2019) such as centrality measures can be used to identify coordinated behavior. This technique can help to highlight the propagation of disinformation or persuasive content over social networks.

Subsequently, deep learning techniques (Kumar et al., 2020) also contributed to fine-tuning the results. The utilization of word embeddings and convolutional neural networks (CNNs) for recognizing persuasion at an early stage also fueled the prevention of social engineering attacks (Tsinganos et al., 2022). Multiple state-of-the-art systems such as LSTMs (Kumar et al., 2020) are also used to detect fake news. Further research revealed that transfer learning approaches like BERT produced more promising results than other cutting-edge NLP techniques (Qasim et al., 2022). Ensembling techniques (Ahmad et al., 2020) were utilized to further enhance the results by integrating various approaches into a single one. Hybrid architectures like combining BERT with a recurrent neural network (RNN) (Kula et al., 2021) or a combination of parallel CNNs with BERT (Kaliyar et al., 2021) achieved a significant score. Recent developments suggest that AI approaches such as explainable AI (XAI) (Chien et al., 2022) are being experimented with for the task of disinformation and persuasion technique detection.

In this paper, we present our approach, which encompasses the utilization of transformer-based models for classification. Variations of BERT are used to develop an ensemble-based system for the given classification tasks.

3 Data

The dataset provided for Task 1 - Persuasion Technique Detection comprises multigenre text snippets, which are either tweets or news paragraphs. The training data has 2427 samples of such snippets, the development data has 259 samples, and the testing data has 503 samples. The training data contains features such as the id, text, label, and type. Each snippet in the training dataset is labeled as either ’true’ or ’false’ based on the presence of persuasion techniques in the given sample. This task falls under the category of binary classification.

The dataset provided for Task 2 - Disinformation Detection comprises tweets. The training data has 14147 (14126 non-null) samples of such tweets, the development data has 2111 samples, and the testing data has 3729 samples. The training data contains features such as the id, text, and label. Each tweet in the training dataset is labeled as either ’disinfo’ or 'no-disinfo' based on the content in every sample. This task falls under the category of binary classification.

The provided dataset is preprocessed using regular expressions to remove irrelevant strings such as "@USER", "LINK" and "RT" to reduce the noise.

4 System

This shared task discusses the problems of Disinformation and Persuasion detection. These problems come under the umbrella of classification problems for which Transformer-based Models have been widely used and have achieved impressive performance. Thus, we have utilized several transformer-based models and ensembling methods in our research as shown in figure 1. The models are trained for 10 epochs with a learning rate of 1e-5, a batch size of 32, and the AdamW optimizer. The methodologies have been briefly discussed in the section below.

4.1 BERT

Antoun et al. (2020) discusses how BERT models which are pre-trained on a large corpus of a specific language like Arabic, perform well on language understanding tasks. They propose several such models that help provide state-of-the-art results for the Arabic language and thus have been utilized for our research.

The pre-training dataset used for the models comprises 70 million sentences which is about 24GB in size. The data consists of news that spans multiple topics and thus represents a variety that is useful for numerous downstream tasks. The Masked Language Modeling and Next Sentence Prediction Tasks have been used as the pre-training objectives which help the models develop a good contextual understanding of the input sequence. AraBERT was evaluated on three NLP tasks namely, Question Answering, Sentiment Analysis, and Named-Entity Recognition to prove its effectiveness across various tasks and domains.

Various variants of the AraBERT model have been provided with slight tweaks in their pre-training phases and parameters used. AraBERT v1 or v0.1 are the original models, while v2 or v0.2 are the newer versions with better vocabulary and pre-processing. AraBERTv0.2-Twitter-base consists of 136M parameters, it is pre-trained with 60M multi-dialect tweets besides the dataset used for the other v0.2 models. AraBERTv2-base is pre-trained on 420M examples that have a sequence length of 128 and on 207M examples that have a sequence length of 512.

To pre-train MARBERT (Abdul-Mageed et al., 2021), 1B Arabic tweets were selected at random from a sizable internal dataset of roughly 6B tweets. Unlike AraBERT, the MARBERT model is trained on Twitter data, which involves both MSA and diverse dialects. It is trained using 163M parameters. This model is trained with a batch size of 256 and a maximum sequence length of 128. It is fine-tuned on several downstream tasks such as social meaning and sentiment analysis.

4.2 ELECTRA

Although, Masked Language Modeling pre-training for BERT-based models has given impressive results, the "Efficiently Learning an Encoder that Classifies Token Replacements Accurately" (ELECTRA) approach has yielded better results whilst being more efficient in terms of model size and compute needed for pre-training. AraELECTRA is the discriminator model (araelectra-base-discriminator) and the generator is a BERT model (araelectra-base-generator).

The data used for pre-training consists of mostly news articles and the size of the dataset is 77GB which consists of 8.8 billion words. The model is pre-trained for 2 million steps with a batch size of 256.

AraELECTRA is a BERT-based model with 12 encoding layers consisting of 12 attention heads. Its hidden size is 768 and has a maximum input sequence length of 512. The total parameters in AraELECTRA are 136 million. The generator Model (araelectra-base-generator) used in the ELECTRA approach for pre-training is a BERT model of a considerably smaller size with 60 million total parameters. AraELECTRA is evaluated on three NLP tasks namely, Question Answering, Sentiment Analysis, and Named-Entity Recognition.

5 Ensembling

Ensembling is a technique that combines the results of various models to generate the eventual intended result of the system. Statistical as well as non-statistical methods are used for this purpose. Ensembling is useful as it helps generate results that are better than the results given by the individual models.

Amongst several methods leveraged for ensembling, we observed that the "hard voting" ensemble technique proved to be the most efficient and accurate. In hard voting, the majority vote or the "mode" of all the predictions is selected as the final prediction. It helps improve the robustness of the system and minimizes the variance in the results.

6 Results

We discuss the results of our experiments for tasks 1-A and 2-A in this section. Table 2 and Table 4 contain our results for the models and the ensembled score for the respective tasks. The micro F1 score serves as the official score metric for both tasks 1-A and 2-A.

Model	Micro F1 Score
Araelectra-base-discriminator	**0.872**
AraBERTv0.2-Twitter-base	0.842
MARBERTv2 (Post-evaluation)	**0.876**
AraBERTv1-base	0.823
AraBERTv2-base	0.849
Ensemble - Hard Voting	**0.865**
Ensemble - Hard Voting (Post-evaluation)	**0.869**

Table 1: Results for Task 1-A on Development dataset

Model	Micro F1 Score
Araelectra-base-discriminator	**0.750**
AraBERTv0.2-Twitter-base	0.746
MARBERTv2 (Post-evaluation)	0.732
AraBERTv1-base	0.702
AraBERTv2-base	0.728
Ensemble - Hard Voting	**0.742**
Ensemble - Hard Voting (Post-evaluation)	**0.751**

Table 2: Results for Task 1-A on Test dataset

6.1 Task 1-A

Araelectra-base-discriminator performs best with a micro F1 score of 0.872 on the development dataset and 0.750 on the test dataset as seen in Table 1 and Table 2 respectively. This performance is indicative of the advantages of utilizing the ELECTRA pre-training approach, where the Replaced Token Detection (RTD) is the objective for pre-training. It achieves a marginally better micro F1 score than the hard voting-based ensembled result of the four models. Despite this, we use the ensemble-based system as our final approach because it generates low-variance results and provides stable predictions. Our system achieved a micro F1 score of 0.742 on the test dataset.

In the post-evaluation phase (after submission of the official scores), out of the various models we experiment with for the given task, MARBERTv2 outperforms Araelectra-base-discriminator and emerges as the best model with a micro F1 score of 0.876 on development dataset. This can be attributed to the large size of the tweet-based training corpus. It boosts the ensemble scores to the 0.869 on development dataset and the 0.751 on test dataset.

Model	Micro F1 Score
Araelectra-base-generator	0.893
AraBERTv0.2-Twitter-base	**0.907**
MARBERTv2 (Post-evaluation)	**0.909**
AraBERTv1-base	0.882
AraBERTv2-base	0.897
Ensemble - Hard Voting	**0.909**
Ensemble - Hard Voting (Post-evaluation)	**0.914**

Table 3: Results for Task 2-A on Development dataset

6.2 Task 2-A

AraBERTv0.2-Twitter-base achieves the best results with a micro F1 score of 0.907 on the development dataset and 0.900 on the test dataset as seen in Table 3 and Table 4 respectively among the four models. This is suggestive of the benefits of the model being pre-trained on a dataset consisting of tweets. The hard voting-based ensemble provides the best results as mentioned in Table 4. In addition to achieving the best performance, ensembling also generates results with greater generalizability and stable predictions and is therefore chosen as the final approach for the system. Our system achieved a micro F1 score of 0.901 and a macro of F1 score

Model	Micro F1 Score
Araelectra-base-generator	0.882
AraBERTv0.2-Twitter-base	**0.900**
MARBERTv2 (Post-evaluation)	**0.903**
AraBERTv1-base	0.882
AraBERTv2-base	0.894
Ensemble - Hard Voting	**0.901**
Ensemble - Hard Voting (Post-evaluation)	**0.905**

Table 4: Results for Task 2-A on Test dataset

0.861 on the test dataset.

In the post-evaluation phase (after submission of the official scores), out of the various models we experiment with for the given task, MARBERTv2 outperforms AraBERTv0.2-Twitter-base and emerges as the best model with a micro F1 score of 0.909 on development dataset and 0.903 on test dataset. This can be attributed to the large size of the tweet-based training corpus. It boosts the ensemble scores to 0.914 on the development dataset and 0.905 on the test dataset.

7 Conclusion

In this paper, we compared the performance of several transformer-based models on the tasks of Persuasion technique detection and Disinformation detection. For the final submission, amongst the individual models, it is observed that the Araelectra-base-discriminator achieved the best performance for Task 1-A. This model was able to achieve a micro F1 score of 0.742. Likewise, AraBERTv0.2-Twitter-base achieved the best results for Task 2-A and the final system yielded a micro F1 score of 0.901. Hard voting-based ensembling is used for our final systems to improve performance whilst also generating stable predictions. In the future, with the availability of better computational resources, we can enhance the system's performance by training it for longer and by using larger models. Moreover, we can experiment with other suitable ensembling techniques to gauge their effectiveness.

8 Limitations

Language Models used here are compute-intensive and thus may not always be suitable for application in real-world and real-time systems that have constraints on computational resources. The pre-training datasets may have certain biases in them, even though they might be rich in information. They may thus not represent the real-world picture accurately.

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.

Iftikhar Ahmad, Muhammad Yousaf, Suhail Yousaf, and Muhammad Ovais Ahmad. 2020. Fake news detection using machine learning ensemble methods. *Complexity*, 2020:1–11.

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, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. AraBERT: Transformer-based model for Arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France. European Language Resource Association.

Shih-Yi Chien, Cheng-Jun Yang, and Fang Yu. 2022. Xflag: Explainable fake news detection model on social media. *International Journal of Human–Computer Interaction*, 38(18-20):1808–1827.

Nadia K Conroy, Victoria L Rubin, and Yimin Chen. 2015. Automatic deception detection: Methods for finding fake news. *Proceedings of the association for information science and technology*, 52(1):1–4.

Song Feng, Ritwik Banerjee, and Yejin Choi. 2012. Syntactic stylometry for deception detection. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 171–175.

Maram Hasanain, Firoj Alam, Hamdy Mubarak, Samir Abdaljalil, Wajdi Zaghouani, Preslav Nakov, Giovanni Da San Martino, and Abed Alhakim Freihat. 2023. ArAIEval Shared Task: Persuasion Techniques and Disinformation Detection in Arabic Text. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore. Association for Computational Linguistics.

Rahul Radhakrishnan Iyer and Katia Sycara. 2019. An unsupervised domain-independent framework for automated detection of persuasion tactics in text. *arXiv preprint arXiv:1912.06745*.

Rohit Kumar Kaliyar, Anurag Goswami, and Pratik Narang. 2021. Fakebert: Fake news detection in social media with a bert-based deep learning approach. *Multimedia tools and applications*, 80(8):11765–11788.

Z Khanam, BN Alwasel, H Sirafi, and Mamoon Rashid. 2021. Fake news detection using machine learning approaches. In *IOP conference series: materials science and engineering*, volume 1099, page 012040. IOP Publishing.

Sebastian Kula, Michał Choraś, and Rafał Kozik. 2021. Application of the bert-based architecture in fake news detection. In *13th International Conference on Computational Intelligence in Security for Information Systems (CISIS 2020) 12*, pages 239–249. Springer.

Sachin Kumar, Rohan Asthana, Shashwat Upadhyay, Nidhi Upreti, and Mohammad Akbar. 2020. Fake news detection using deep learning models: A novel approach. *Transactions on Emerging Telecommunications Technologies*, 31(2):e3767.

Syed Ishfaq Manzoor, Jimmy Singla, et al. 2019. Fake news detection using machine learning approaches: A systematic review. In *2019 3rd international conference on trends in electronics and informatics (ICOEI)*, pages 230–234. IEEE.

Hamdy Mubarak, Samir Abdaljalil, Azza Nassar, and Firoj Alam. 2023. Detecting and reasoning of deleted tweets before they are posted. *arXiv preprint arXiv:2305.04927*.

Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. 2017. Automatic detection of fake news. *arXiv preprint arXiv:1708.07104*.

Rukhma Qasim, Waqas Haider Bangyal, Mohammed A Alqarni, Abdulwahab Ali Almazroi, et al. 2022. A fine-tuned bert-based transfer learning approach for text classification. *Journal of healthcare engineering*, 2022.

Julio CS Reis, André Correia, Fabrício Murai, Adriano Veloso, and Fabrício Benevenuto. 2019. Supervised learning for fake news detection. *IEEE Intelligent Systems*, 34(2):76–81.

Victoria L Rubin, Niall J Conroy, and Yimin Chen. 2015. Towards news verification: Deception detection methods for news discourse. In *Hawaii international conference on system sciences*, pages 5–8.

Kai Shu, H Russell Bernard, and Huan Liu. 2019. Studying fake news via network analysis: detection and mitigation. *Emerging research challenges and opportunities in computational social network analysis and mining*, pages 43–65.

Nikolaos Tsinganos, Ioannis Mavridis, and Dimitris Gritzalis. 2022. Utilizing convolutional neural networks and word embeddings for early-stage recognition of persuasion in chat-based social engineering attacks. *IEEE Access*, 10:108517–108529.

Claire Wardle and Hossein Derakhshan. 2017. *Information disorder: Toward an interdisciplinary framework for research and policymaking*, volume 27. Council of Europe Strasbourg.

KnowTellConvince at ArAIEval Shared Task: Disinformation and Persuasion Detection in Arabic using Similar and Contrastive Representation Alignment

Hariram Veeramani
Department of Electrical and Computer Engineering, UCLA, USA
hariram@ucla.edu

Surendrabikram Thapa
Department of Computer Science, Virginia Tech, Blacksburg, USA
sbt@vt.edu

Usman Naseem
College of Science and Engineering, James Cook University, Australia
usman.naseem@jcu.edu.au

Abstract

In an era of widespread digital communication, the challenge of identifying and countering disinformation has become increasingly critical. However, compared to the solutions available in the English language, the resources and strategies for tackling this multifaceted problem in Arabic are relatively scarce. To address this issue, this paper presents our solutions to tasks in ArAIEval 2023. Task 1 focuses on detecting persuasion techniques, while Task 2 centers on disinformation detection within Arabic text. Leveraging a multi-head model architecture, fine-tuning techniques, sequential learning, and innovative activation functions, our contributions significantly enhance persuasion techniques and disinformation detection accuracy. Beyond improving performance, our work fills a critical research gap in content analysis for Arabic, empowering individuals, communities, and digital platforms to combat deceptive content effectively and preserve the credibility of information sources within the Arabic-speaking world.

1 Introduction

In today's information age, the rapid dissemination of digital content across various platforms has revolutionized the way information is produced, shared, and consumed. This unprecedented accessibility to information has brought numerous benefits, but it has also given rise to new challenges, particularly in the realms of misinformation, propaganda, and disinformation (Alam et al., 2022a). Identifying and addressing these issues is paramount for ensuring the integrity and credibility of information sources.

While the English language has garnered substantial attention in the realm of misinformation, persuasion, and disinformation detection, it is imperative that we recognize the equal, if not greater, significance of these endeavors in the Arabic language. Less research on these areas will leave the Arabic-speaking world vulnerable to the harmful effects of deceptive content. Arabic's linguistic and cultural nuances demand tailored approaches to combat these issues effectively (Sheikh Ali et al., 2023; Alyoubi et al., 2023; Fouad et al., 2022).

Disinformation, encompassing hate speech, offensive content, rumors, spam, and propaganda, presents formidable challenges in the Arabic-speaking world, shaped by linguistic diversity and cultural nuances (Nakov et al., 2022; Alam et al., 2022b). Hate speech and offensive content, intensified by cultural sensitivities, demand effective detection and mitigation to avert real-world repercussions (Albadi et al., 2018; Al-Hassan and Al-Dossari, 2022; Chowdhury et al., 2019). Rumors, highly contagious within tight-knit Arabic communities, necessitate vigilant monitoring to counteract panic and misinformation, exploiting cultural contexts for added complexity (Nakov et al., 2021; Harrag and Djahli, 2022). Spam, spanning fraudulent ads and misleading claims, pervades digital spaces in all languages, underlining the need to distinguish it from credible content for online source credibility (Kaddoura et al., 2023; Alkadri et al., 2022). Propaganda, a pivotal element of disinformation campaigns, influences public opinion and necessitates understanding and countering within the Arabic-speaking context to protect individuals and communities from manipulation by misleading narratives (Sharara et al., 2022; Feldman et al., 2021). Addressing these multifaceted challenges requires comprehensive research efforts and robust detection models that account for linguistic and cultural intricacies, preserving the credibility of information sources, online discourse, and public opinion in the diverse and dynamic Arabic-speaking linguistic landscape.

In order to address the problems mentioned above and to extend the previous related works (Habernal et al., 2017, 2018; Da San Martino et al., 2019; Barrón-Cedeno et al., 2019), in this paper, we

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 519–524
December 7, 2023 ©2023 Association for Computational Linguistics

present a multi-faceted approach that combines innovative model architectures, fine-tuning strategies, and sequential learning techniques to effectively address subtask 1A of Task 1 (persuasion or propaganda detection) and both subtasks of Task 2 (disinformation detection) in ArAIEval 2023 (Hasanain et al., 2023). Our incorporation of contrastive learning, renormalization, sentence embedding, cosine similarity checks, and GELU activation functions within the Arabic BERT framework demonstrates a comprehensive strategy for detecting disinformation subtleties, including hate speech, offensive content, rumors, spam, and propaganda. Our contributions not only enhance disinformation detection accuracy but also bridge the research gap in content analysis for the Arabic language.

2 Task Description

Task 1: This task mainly deals with persuasion techniques (propagandistic content) and has two subtasks. Our participation is focused on subtask 1A, which involves analyzing individual paragraphs of text from various genres to determine whether they contain persuasive content, with a binary classification of "Yes" or "No" as the output.

Task 2: This task centers on disinformation detection with two subtasks: subtask 2A for binary classification to identify disinformation in tweets and subtask 2B for multi-class classification, categorizing tweets into hate speech, offensive content, rumors, or spam categories.

3 Dataset

Task 1: The dataset comprises tweets and news paragraphs that have been annotated to identify the use of persuasion techniques. These annotations are provided in binary and multilabel settings, allowing for the classification of the presence or absence of persuasion techniques and, in the multilabel setting, the identification of multiple propaganda techniques within the same text. Since we only participate in subtask 1A, we only use binary annotation data. The development set contains approximately 78% of the data without propaganda and 22% of the data with propaganda. Similarly, the test set comprises roughly 34.2% of the data without propaganda and 65.8% of the data with propaganda.

Task 2: Similar to Task 1, this task also contains tweets annotated for binary and multiclass labels for subtask 2A and subtask 2B, respectively.

4 System Descriptions

Our system is an ensemble of four models, as shown in Figure 1. Below, we explain every component in detail.

Model A - Supervised Contrastive Learning with Arabic BERT: In Model A, we employ contrastive learning to enhance Arabic text representations. The motivation is to empower the model for binary classification tasks by improving its ability to distinguish between positive and negative examples in Arabic text (Alam et al., 2022b; Veeramani et al., 2023b,d,c). Contrastive learning encourages the model to capture semantic relationships effectively, benefiting applications like sentiment analysis. We fine-tune BERT Arabic Base (Safaya et al., 2020) with a contrastive loss function, pushing the model to generate embeddings emphasizing semantic similarity and dissimilarity. During training, it promotes similar representations for similar sentences and different representations for dissimilar sentences, enhancing the model's semantic understanding.

Model B - Sequential Learning with ArabicBERT: Model B adopts a sequential learning approach, fine-tuning ArabicBERT on task-specific data to adapt to various Arabic NLP tasks. The motivation is to enable the model to comprehend sequential relationships and context in textual data, which is crucial for tasks like text generation and named entity recognition. Rationally, sequential learning involves taking the pretrained BERT model and fine-tuning it on specific tasks, transferring knowledge from its general language understanding capabilities to task-specific nuances. We adjust learning rates and batch sizes for different tasks and employ task-specific loss functions during fine-tuning.

Model C - Fine-Tuned Arabic BERT with Renormalized XNLI Data and Sequential Learning: This process entails taking the pretrained Arabic BERT model and fine-tuning it on the renormalized XNLI dataset. During this phase, the labels in the dataset are adjusted to reflect the sentiment perspective, where neutral and entailment labels are unified into one class, and contradiction remains as a separate class. Subsequently, applying sequential learning further enhances the model's adaptation to the task-specific nuances (Gururangan et al.,

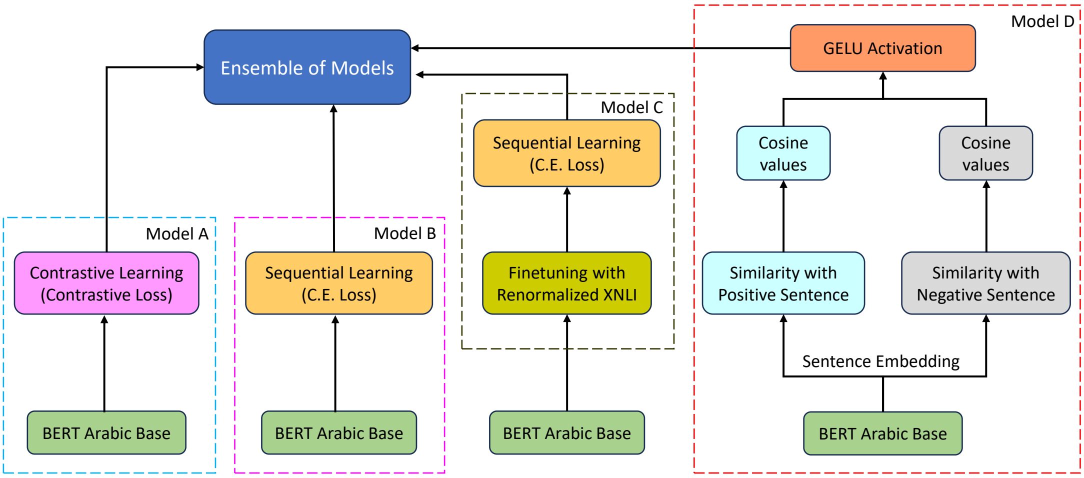

Model D
GELU Activation
Ensemble of Models
Model C
Sequential Learning (C.E. Loss)
Cosine values
Cosine values
Model A
Model B
Contrastive Learning (Contrastive Loss)
Sequential Learning (C.E. Loss)
Finetuning with Renormalized XNLI
Similarity with Positive Sentence
Similarity with Negative Sentence
Sentence Embedding
BERT Arabic Base
BERT Arabic Base
BERT Arabic Base
BERT Arabic Base

Figure 1: Overall framework of our proposed methodology.

2020; Da San Martino et al., 2019; Veeramani et al., 2023e,a,f). During sequential learning, the model adjusts its internal representations based on the fine-tuned XNLI data, further refining its understanding of sentiment-related features and patterns unique to Arabic text. This sequential fine-tuning ensures that the model aligns precisely with the propaganda/disinformation analysis and classification requirements.

Model D - Sentence Embeddings with GELU Activation: The process begins with calculating semantic similarity between sentences in Arabic text, serving as a foundational step for tasks demanding an understanding how closely related or similar two sentences are. The primary motivation behind this approach is to excel in applications that rely on measuring the semantic similarity between sentences in Arabic, such as persuasion detection, disinformation detection, and multiclass classification (Kanagasabai et al., 2023). The model extracts sentence embeddings from Arabic BERT representations to capture the essential features and semantics of each sentence. Subsequently, it calculates the cosine similarity between pairs of sentences, providing a quantitative measure of their semantic relatedness. To capture complex and non-linear relationships within the sentence embeddings, the model then applies the GELU (Gaussian Error Linear Unit) activation function. This step enhances the model's ability to discern intricate semantic nuances. Ultimately, the GELU-activated cosine similarity scores enable the model to assess the degree of semantic similarity between sentences, making Model D a valuable asset for tasks like persuasion and disinformation detection, which requires semantic understanding and similarity assessment in Arabic text processing.

5 Results and Discussion

This section discusses the results of our runs. Apart from the above mentioned, we also tested to ArSAS BERT[1]. We perform a detailed ablation of what factors contribute to the better performance of the system.

5.1 Task 1A (Persuasion Detection)

In the context of persuasion detection, the presented Table 1 reveals a comprehensive evaluation of various models designed to excel in this task. Arabic-BERT demonstrated commendable effectiveness with a micro-averaged F1-score of 72.23, emphasizing its proficiency in classifying instances. ArSAS BERT slightly outperformed Arabic-BERT with a micro-averaged F1-score of 73.4, highlighting its capabilities in persuasion detection. However, the combination of components B and D notably improved model performance, resulting in Model (B + D) achieving a micro-averaged F1-score of 74.44. Model (A + B) further enhanced performance to a micro-averaged F1-score of 74.75 by combining components A and B, showcasing the value of ensemble models. Nevertheless, the Model (A + D) also boasted a micro-averaged F1-score of 75.77, emphasizing the effectiveness of combining components A and D. The most comprehensive approach, Model (A + B + C + D),

[1] https://huggingface.co/Osaleh/sagemaker-bert-base-arabic-ArSAS

Models	F1$_{mic}$	Pre$_{mac}$	Rec$_{mac}$	F1$_{mac}$
Arabic-BERT	72.23	73.51	72.06	71.0
ArSAS BERT	73.4	73.17	74.8	73.2
Model (B + D)	74.44	75.09	74.58	74.67
Model (A + B)	74.75	75.48	74.75	75.05
Model (A + D)	75.77	76.9	75.26	76.05
Model (A+B+C+D)	76.14	78.11	76.14	76.82

Table 1: Results for task 1A (persuasion detection). The F1$_{mic}$ stands for micro-averaged F1-score. Similarly, Pre$_{mac}$, Rec$_{mac}$, and F1$_{mac}$ represents macro-averaged precision, recall and F1-score.

achieved the highest micro-averaged F1-score at 76.14, reaffirming the synergy of all four components in tackling persuasion detection effectively. These results underscore the significance of model combinations and component choices in optimizing performance for this task.

5.2 Task 2A (Disinformation Detection)

In disinformation detection, the provided Table 2 showcases a comprehensive evaluation of diverse models. Arabic-BERT demonstrated strong performance with a micro-averaged F1-score of 86.4, underscoring its effectiveness in identifying disinformation. ArSAS BERT improved upon this, achieving a micro-averaged F1-score of 87.26, signifying its proficiency in detecting false information. However, the strategic combination of components B and D notably enhanced model performance, resulting in Model (B + D) achieving a micro-averaged F1-score of 88.5. Model (A + B) excelled further with an impressive micro-averaged F1-score of 89.05, indicating its strength in disinformation detection. However, Model (A + D) emerged as a better performer, boasting a micro-averaged F1-score of 89.38 and demonstrating its exceptional capability in detecting disinformation. The most encompassing approach, Model (A + B + C + D), outshone the rest with the highest micro-averaged F1-score at 89.67, reaffirming the synergy of all four components in effectively combatting disinformation.

Models	F1$_{mic}$	Pre$_{mac}$	Rec$_{mac}$	F1$_{mac}$
Arabic-BERT	86.4	87	86.22	86.32
ArSAS BERT	87.26	88.5	87.15	87.2
Model (B + D)	88.5	89.02	88.46	88.9
Model (A + B)	89.05	89.88	89.06	89.35
Model (A + D)	89.38	90	89.38	89.61
Model (A+B+C+D)	89.67	90.39	89.68	89.93

Table 2: Results for task 2A (disinformation detection).

5.3 Task 2B (Disinformation Class Detection)

In disinformation class detection, as shown in Table 3, Model B achieved a micro-averaged F1-score of 80.36, while Model (B + D) improved performance slightly with a micro-averaged F1-score of 80.71. This shows that combining components B and D enhanced disinformation class detection, emphasizing the value of collaboration between these elements for improved accuracy in identifying specific disinformation classes.

Models	F1$_{mic}$	Pre$_{mac}$	Rec$_{mac}$	F1$_{mac}$
Model B	80.36	83.42	80.51	80.36
Model (B + D)	80.71	83.85	80.71	81.81

Table 3: Results for task 2B (disinformation class detection).

In summary, these performance tables demonstrate the power of ensemble models and collaborative approaches in improving the accuracy of persuasion and disinformation detection tasks. Combining different components and models enhanced overall performance, with micro and macro F1-scores consistently rising.

6 Conclusion

In summary, our study emphasizes the power of ensemble models and collaborative approaches in improving the accuracy of persuasion and disinformation detection tasks in Arabic text. We consistently observed enhanced performance through rigorous experimentation, as evidenced by rising micro and macro F1-scores across various model combinations. These results underscore the importance of adaptability and synergy in addressing the nuanced challenges of natural language understanding tasks. Whether it is fine-tuning sentiment semantics, leveraging sentence embeddings, or combining all components, ensemble models consistently outperform individual approaches. These findings offer valuable insights for Arabic text processing and as a model for tackling similar challenges across languages and domains. In an ever-evolving landscape of language processing, our study highlights the significance of diverse techniques and collaborative strategies to effectively meet the complexity of natural language understanding tasks. Ultimately, our research contributes to more accurate solutions and a deeper understanding of persuasive and deceptive language in the digital age.

Limitations

This study, while providing valuable insights into ensemble models for persuasion and disinformation detection in Arabic text, is subject to certain limitations. Using data from limited sources may have restricted the comprehensiveness and generalizability of our findings. Additionally, the complexity and computational demands associated with ensemble models could pose practical constraints in real-world applications, warranting further investigation into model efficiency. Furthermore, the domain-specific focus of our work on persuasion and disinformation detection might limit its direct applicability to other natural language processing tasks or domains. Finally, the interpretability of ensemble models and the potential influence of temporal dynamics in text data represent additional aspects for future research to explore.

Ethics Statement

This research adheres to ethical guidelines and principles in all aspects of data analysis and reporting. The datasets used in this study were sourced from authorized sources, and no personally identifiable information or sensitive data was utilized.

References

Areej Al-Hassan and Hmood Al-Dossari. 2022. Detection of hate speech in arabic tweets using deep learning. *Multimedia systems*, 28(6):1963–1974.

Firoj Alam, Stefano Cresci, Tanmoy Chakraborty, Fabrizio Silvestri, Dimiter Dimitrov, Giovanni Da San Martino, Shaden Shaar, Hamed Firooz, and Preslav Nakov. 2022a. A survey on multimodal disinformation detection. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6625–6643.

Firoj Alam, Hamdy Mubarak, Wajdi Zaghouani, Giovanni Da San Martino, and Preslav Nakov. 2022b. Overview of the wanlp 2022 shared task on propaganda detection in arabic. *WANLP 2022*, page 108.

Nuha Albadi, Maram Kurdi, and Shivakant Mishra. 2018. Are they our brothers? analysis and detection of religious hate speech in the arabic twittersphere. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 69–76. IEEE.

Abdullah M Alkadri, Abeer Elkorany, and Cherry Ahmed. 2022. Enhancing detection of arabic social spam using data augmentation and machine learning. *Applied Sciences*, 12(22):11388.

Shatha Alyoubi, Manal Kalkatawi, and Felwa Abukhodair. 2023. The detection of fake news in arabic tweets using deep learning. *Applied Sciences*, 13(14):8209.

Alberto Barrón-Cedeno, Israa Jaradat, Giovanni Da San Martino, and Preslav Nakov. 2019. Proppy: Organizing the news based on their propagandistic content. *Information Processing & Management*, 56(5):1849–1864.

Arijit Ghosh Chowdhury, Aniket Didolkar, Ramit Sawhney, and Rajiv Shah. 2019. Arhnet-leveraging community interaction for detection of religious hate speech in arabic. In *Proceedings of the 57th annual meeting of the association for computational linguistics: student research workshop*, pages 273–280.

Giovanni Da San Martino, Yu Seunghak, Alberto Barrón-Cedeno, Rostislav Petrov, Preslav Nakov, et al. 2019. Fine-grained analysis of propaganda in news article. 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 5636–5646. Association for Computational Linguistics.

Anna Feldman, Giovanni Da San Martino, Chris Leberknight, and Preslav Nakov. 2021. Proceedings of the fourth workshop on nlp for internet freedom: Censorship, disinformation, and propaganda. In *Proceedings of the Fourth Workshop on NLP for Internet Freedom: Censorship, Disinformation, and Propaganda*.

Khaled M Fouad, Sahar F Sabbeh, and Walaa Medhat. 2022. Arabic fake news detection using deep learning. *Computers, Materials & Continua*, 71(2).

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*.

Ivan Habernal, Raffael Hannemann, Christian Pollak, Christopher Klamm, Patrick Pauli, and Iryna Gurevych. 2017. Argotario: Computational argumentation meets serious games. *EMNLP 2017*, page 7.

Ivan Habernal, Patrick Pauli, and Iryna Gurevych. 2018. Adapting serious game for fallacious argumentation to german: Pitfalls, insights, and best practices. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Fouzi Harrag and Mohamed Khalil Djahli. 2022. Arabic fake news detection: A fact checking based deep learning approach. *Transactions on Asian and Low-Resource Language Information Processing*, 21(4):1–34.

Maram Hasanain, Firoj Alam, Hamdy Mubarak, Samir Abdaljalil, Wajdi Zaghouani, Preslav Nakov, Giovanni Da San Martino, and Abed Alhakim Freihat.

2023. ArAIEval Shared Task: Persuasion Techniques and Disinformation Detection in Arabic Text. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore. Association for Computational Linguistics.

Sanaa Kaddoura, Suja A Alex, Maher Itani, Safaa Henno, Asma AlNashash, and D Jude Hemanth. 2023. Arabic spam tweets classification using deep learning. *Neural Computing and Applications*, pages 1–14.

Rajaraman Kanagasabai, Saravanan Rajamanickam, Hariram Veeramani, Adam Westerski, and Kim Jung Jae. 2023. Semantists at ImageArg-2023: Exploring Cross-modal Contrastive and Ensemble Models for Multimodal Stance and Persuasiveness Classification. In *Proceedings of the 10th Workshop on Argument Mining*, Online and in Singapore. Association for Computational Linguistics.

Preslav Nakov, Firoj Alam, Shaden Shaar, Giovanni Da San Martino, and Yifan Zhang. 2021. A second pandemic? analysis of fake news about covid-19 vaccines in qatar. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 1010–1021.

Preslav Nakov, Alberto Barrón-Cedeño, Giovanni Da San Martino, Firoj Alam, Julia Maria Struß, Thomas Mandl, Rubén Míguez, Tommaso Caselli, Mucahid Kutlu, Wajdi Zaghouani, et al. 2022. The clef-2022 checkthat! lab on fighting the covid-19 infodemic and fake news detection. In *European Conference on Information Retrieval*, pages 416–428. Springer.

Ali Safaya, Moutasem Abdullatif, and Deniz Yuret. 2020. KUISAIL at SemEval-2020 task 12: BERT-CNN for offensive speech identification in social media. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 2054–2059, Barcelona (online). International Committee for Computational Linguistics.

Mohamad Sharara, Wissam Mohamad, Ralph Tawil, Ralph Chobok, Wolf Assi, and Antonio Tannoury. 2022. Arabert model for propaganda detection. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 520–523.

Zien Sheikh Ali, Watheq Mansour, Fatima Haouari, Maram Hasanain, Tamer Elsayed, and Abdulaziz Al-Ali. 2023. Tahaqqaq: A real-time system for assisting twitter users in arabic claim verification. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3170–3174.

Hariram Veeramani, Surendrabikram Thapa, and Usman Naseem. 2023a. Automated Citation Function Classification and Context Extraction in Astrophysics: Leveraging Paraphrasing and Question Answering. In *Proceedings of the second Workshop on Information Extraction from Scientific Publications*, Online. Association for Computational Linguistics.

Hariram Veeramani, Surendrabikram Thapa, and Usman Naseem. 2023b. DialectNLU at NADI 2023 Shared Task: Transformer Based Multitask Approach Jointly Integrating Dialect and Machine Translation Tasks in Arabic. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore. Association for Computational Linguistics.

Hariram Veeramani, Surendrabikram Thapa, and Usman Naseem. 2023c. Enhancing ESG Impact Type Identification through Early Fusion and Multilingual Models. In *Proceedings of the Sixth Workshop on Financial Technology and Natural Language Processing (FinNLP)*. Association for Computational Linguistics.

Hariram Veeramani, Surendrabikram Thapa, and Usman Naseem. 2023d. LowResContextQA at Qur'an QA 2023 Shared Task: Temporal and Sequential Representation Augmented Question Answering Span Detection in Arabic. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore. Association for Computational Linguistics.

Hariram Veeramani, Surendrabikram Thapa, and Usman Naseem. 2023e. LowResourceNLU at BLP-2023 Task 1 2: Enhancing Sentiment Classification and Violence Incitement Detection in Bangla Through Aggregated Language Models. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Hariram Veeramani, Surendrabikram Thapa, and Usman Naseem. 2023f. Temporally Dynamic Session-Keyword Aware Sequential Recommendation system. In *2023 International Conference on Data Mining Workshops (ICDMW)*.

PTUK-HULAT at ArAIEval Shared Task: Fine-tuned Distilbert to Predict Disinformative Tweets

Areej Jaber
Palestine Technical University - Khadoorie
a.jabir@ptuk.edu.ps

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

Abstract

Disinformation involves the dissemination of incomplete, inaccurate, or misleading information; it has the objective, goal, or purpose of deliberately or intentionally lying to others about the truth. The spread of disinformative information on social media has serious implications, and it causes concern among internet users in different aspects. Automatic classification models are required to detect disinformative posts on social media, especially on Twitter. In this article, DistilBERT multilingual model was fine-tuned to classify tweets either as dis-informative or not dis-informative in Subtask 2A of the ArAIEval shared task. The system outperformed the baseline and achieved F1 micro 87% and F1 macro 80%. Our system ranked 11 compared with all participants.

1 Introduction

Nowadays, social media has advanced to the point that it can compete with traditional media. The freedom of user participation could have negative consequences Dhiman (2023). Disinformation is one of the side effects of this intentionally aiming to mislead the truth that could affect negatively people in many fields like politics, and health, among others.

Spreading fake news can lead to misunderstanding, harm individuals or groups, damage reputation, or even influence public opinion and decision-making, Nasery et al. (2023). Thus, automatic detection of these kinds of data is a very important issue. For a while, it seemed so easy to detect disinformation data by domain experts or fact-checkers, but with daily huge propagation data in social media, more resources are needed to automate and speed up the process of detection of this kind of information.

Arabic language is one of the languages spoken in the world, with 422 million people including native and non-native speakers. It is the official language in 22 countries with at least 30 distinct dialects. However, there are three categories: Classical Arabic (CA), Modern Standard Arabic (MSA), and Dialectal Arabic (DA) Kadaoui et al. (2023). CA is the original Arabic language that has been used for over 1,500 years, and it is usually used in most Arabic religious texts. MSA is one of the official languages of the United Nations and is widely used in todays Arabic newspapers, letters, and formal meetings, which are also focused on by researchers. DA is spoken Arabic used in informal daily communication.

With the recent advanced improvements in the natural language processing (NLP) field and the evolution of large language modeling which is based on transformers architecture Wolf et al. (2020) the development of Arabic language solutions in the NLP field has evolved. To mention some previous efforts devoted to creating Arabic datasets to train and test systems, the Arabic fact-checking and stance detection corpus Baly et al. (2018) contains 422 claims: 219 false claims from Verify [1], and 203 true claims from Reuters. All these claims were made about the war in Syria and related Middle East political issues. Alkhair et al. (2019) describes an Arabic corpus of 342 rumors and 3,000 no rumors about death personalities.

Automatic disinformation classification is the most straightforward way of disinformation analysis. Some previous work has been developed in the Arabic language such as Harrag and Djahli (2022) that explored convolutional neural networks (CNNs) for fact-checking using Baly et al. (2018) to evaluate the proposal obtaining an accuracy averaged from 0.886 to 0.898. A more recent work Nassif et al. (2022) evaluated a transformer-based classifier to recognize fake news using Arabic word embeddings. Authors reported a performance accuracy of 98% using models such as

[1] https://verify-sy.com/

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 525–529
December 7, 2023 ©2023 Association for Computational Linguistics

QuariBert-Bse and Arabic-BERT among others.

The remainder of this article is organized as follows: The task definition is described in Section 2. Section 3 describes the data set that was used, then an explanation of the baselines and the proposed system are given in Section 4. System evaluation is introduced in Section 5. Finally, conclusions are given in Section 7.

2 Task Definition

Text classification is a machine-learning process that assigns a document to one or more predetermined categories based on its content Abdulghani and Abdullah (2022). It is a key problem in NLP, with applications ranging from sentiment analysis to email routing to offensive language detection, spam filtering, and language identification.

Disinformation classification is a form of text classification that is normally identified as binary classification Mu et al. (2022). Given a set of labeled contexts that will be modeled as a feature f, the task aims to predict whether f is disinformative or not.

$$g(f) = \begin{cases} 1, & \text{if f is dis-informative text} \\ 0, & \text{if f is not dis-informative text} \end{cases}$$

where **g** is the function we want to learn from the available data. The combination of the features to obtain **g** can be done manually or automatically.

3 Data

The organizers of the ArAIEval shared task Hasanain et al. (2023) released a data set that aims to categorize a tweet whether it is a disinformative or not. These shared tasks represent continuous works on the Arabic language after Alam et al. (2022).

The data set is extracted from the Twitter website by Twarc package Mubarak et al. (2023). These tweets were extracted by using the word corona in Arabic in February and March 2020. Each sample in the data sets is composed of three fields, the ID which represents the sample identifier, the text field which includes the tweet text, and the label which represents the annotated label for the text either "**disinfo**" or "**no-disinfo**". Table 1 illustrates a set of examples of the provided data.

Three separate data sets were released in two phases. The training and the developing data set were released in the first phase, containing 14,147 and 2,115 samples respectively. Then the test data set was released in the second phase with 3729 samples. Table 2 shows the stats of the data provided by the organizers and it is clear that the data sets are imbalanced.

4 System

Baseline: In order to familiarize the participants with the task, the organizers provided two baselines in the code repository, random and majority baselines.

Proposed System: Pre-trained transformer-based architectures have recently proven to be particularly efficient at language modeling and understanding when trained on a large enough corpus. Bidirectional Encoder Representations from Transformers (BERT) Vaswani et al. (2017) is one of these models that gains the attention of the researchers due to its ability to predict words considering left and right context sides.

Two model sizes are released for BERT as modeling language goals, both of them depend on encoder architecture, BERT$_{large}$ and BERT$_{base}$. The main difference between them is the number of encoders. BERT$_{base}$ consists of a stack of 12 encoders, on the other hand, BERT$_{large}$ consists of a stack of 24 encoders. In addition, they differ in the number of hidden units (768, 1,024) and attention heads (12,16).

Despite the notable results of pre-trained BERT models, it has a drawback which makes it very slow due to its parameter numbers Han et al. (2021). So, the distillation process, which is known as a compression technique in which a small model (the student) is trained to mimic the behavior of a bigger model (the teacher) or an ensemble of models Gou et al. (2021) is produced to deal with this issue.

Based on the current resources for NLP, 90% of the worlds population speaks languages that do not benefit from recent language technologies due to the lack of resources Joshi et al. (2020). Arabic NLP is among these languages that still need more interest to make it mature Bourahouat et al. (2023).

Cross-language transfer is considered the main technique used for addressing the lack of resources in the target language, in which higher resource language models are adapted to the low resource language. The cross-lingual transfer could be achieved in two ways, the first one is by using a trained single high-resource language model, or

ID	Text	Label
0	"الله يلعن ابو الساعة اللي عرفنا فيها كورونا اقسم بالله ماناقص الا الطعوس والبدو يعرفونها	no-disinfo
1	حفل زفاف في القاهرة ...الشعب المصري هو اللي بجيب الجلطة لفيروس كورونا	disinfo
2	البقاء في المنزل يقينا من كورونا حفظ الله بلادنا من كل شر كلنامسؤول	no-disinfo

Table 1: Examples of the ArAIEval dataset.

Data set	Disinfo	No-disinfo	Total
Training	2,656	11,491	14,147
Development	397	1,718	2,115
Test	876	2,853	3,729

Table 2: Description of training, development, and test data sets.

the second way is by using multiple languages with varying amounts of resources. The idea behind these strategies is that the lower-resource language benefits from the model's learning of language invariant features from a huge amount of data in the high-resource language.

Thus, to overcome the low resources problems for the Arabic language and the slowness of the BERT model, in our proposed model we used a distilled multilingual version of BERT which was released by Sanh et al. (2019).

DistilBERT is a multilingual model that is trained in 104 different languages including the Arabic language from the Wikipedia website. Thus, the Distilbert model has 6 layers, 768 dimensions, and 12 heads, totalizing 134M parameters. Table 3 illustrates the main differences between BERT_$base$ and DistilBERT. As shown, the model was able to reduce the size of a BERT model by 40% while retraining 97% of its language understanding capabilities and being 60% faster.

In the following sub-sections, the description of two phases, Development, and evaluation, will be described in detail.

4.1 System Development Phase

In this phase, the organizers of the ArAIEval shared task Hasanain et al. (2023) released training and development datasets. First, fundamental cleaning and preprocessing were performed on both data sets to improve their quality. Hence, white spaces, punctuation marks, hashtags, URLs, special characters, and hyperlinks were removed from the texts, and the null values were dropped. Therefore, the final samples for the experiments were 14126, and 2110 for the training and developing data sets respectively.

For each sample, the labeling is converted to either 1 to represent "no-disinfo" or 0 to represent "disinfo".

As known, input IDs' (which encode the words of the text to sequences of numbers) and attention mask (to tell the model which numbers of input_ids to pay attention to or to ignore) vectors should be generated from the DistilBERT tokenizer for each sample. During the fine-tuning, the training data set was used for optimization and model parameters. On the other hand, the developing data set was used as an evaluation data set to validate the results of the model updates independent of the data it is trained on.

The training arguments were adjusted before running the experiment, the learning rate was 2e-5, and the number of epochs was 2.

4.2 Final Evaluation Phase

When the testing data set was released by the organizers, the same preprocessing and cleaning processes were done on the test samples. Then, the data set was fed to the generated model after converted it into real numbers from the previous phase to get the predictions and submitted to the task portal. After the submission was closed, the organizers published the golden standard for the test data set for analysis of the errors. Figure 1 shows the whole pipeline during the two phases.

5 Results

The system performance was evaluated by using F1 macro, and F1 micro; micro and macro averages are aggregation methods for the F1 score, a metric that is used to measure the performance of classification machine learning models.

F1 score is calculated per class, which means that if you want to calculate the overall F1 score for a dataset with more than one class you will need to aggregate in some way. Micro F1 score is the normal F1 formula but calculated using the total number of True Positives (TP), False Positives

	BERT	DistilBERT
Parameters (millions)	base: 110	base:66
Training Time (days)	8 X V100 X 12	4 times less than BERT
Performance	Outperforms state-of-the-art	3% degradation from BERT

Table 3: Comparison between BERT_$base$ and distilBERT

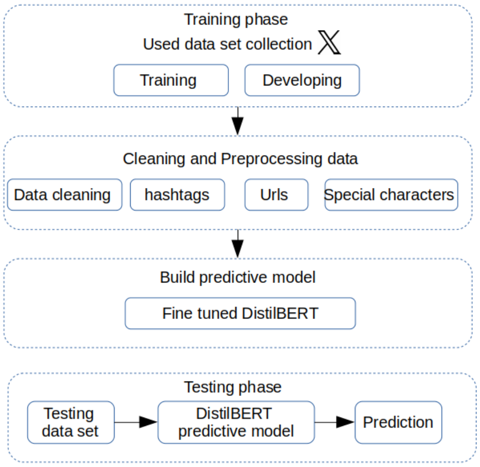


Figure 1: Overview of the proposed approach to predict the dis-informative tweets.

(FP), and False Negatives (FN), instead of individually for each class.

The formula for the micro F1 score is therefore:

$$Micro\, F_1 = \frac{TP}{TP + \frac{1}{2} \times (FP + FN)} \quad (1)$$

The Macro F1 score is the unweighted mean of the F1 scores calculated per class. It is the simplest aggregation for the F1 score. The formula for the macro F1 score is therefore:

$$Macro\, F_1 = \frac{sum(F1\, scores)}{number\, of\, classes} \quad (2)$$

In the training phase, the system achieved 81% F1 micro and 72% F1 macro. The system achieved 87% F1 micro and 80% F1 macro on the testing data set. To get further, the F1 score was computed per class; for the "disinfo" class the system achieved 68% , and "no-disinfo" class, the proposed model achieved 92% f1 score.

6 Discussion

In this work, a distilled multilingual version of BERT was fine-tuned to predict disinformative tweets that are extracted from the social media website Twitter. As shown in Table 4, the system outperformed the baselines in the two phases, training and evaluating phases.

Data set	F1 micro	F1 macro
our result (testing)	0.8675	0.7992
majority-baseline	0.7651	0.4335
random-baseline	0.5154	0.4764

Table 4: The performance of the proposed system compared with the baselines.

The data set which are provided by the organizers is imbalanced and this affects the results as shown in the result section. The proposed system failed to predict 494 samples in total, 346 samples related to "disinfo" class which is the minority class in the data set. On the other hand, 148 samples that were labeled with "no-disinfo" were predicted false from the data set.

Based on our in-depth failure analysis, we found that the system failed to predict correctly the examples containing English words in Arabic letters such as " فولو" which means "follow" in English. Another reason of failure is that some users repeat some characters in some words to express their emotions such as "طفلللل".

7 Conclusion

In this work, we described our proposed system to classify Arabic tweets as either disinformative or not. Distilbert's multilingual model was fine-tuned on the task dataset. The system overcomes the baselines and achieves F1 micro 87% and F1 macro 80% on the testing data set.

The Arabic language is the official language of 22 countries and it is spoken by over 422 million people, but more efforts are needed to get benefits of the recent NLP technologies. Writing a foreign language in Arabic letter should be taken into account to improve the proposed model, in addition to using repeated characters to express emotions.

8 Acknowledgments

This work has been supported by Palestine Technical University - Khadoori (Palestine) and Madrid Regional Government (Comunidad de Madrid-Spain) under the Multiannual Agreement with UC3M in the line of Excellence of University Professors (EPUC3M17) and in the context of the V PRICIT (Regional Programme of Research and Technological Innovation.

References

Farah A Abdulghani and Nada AZ Abdullah. 2022. A survey on arabic text classification using deep and machine learning algorithms. *Iraqi Journal of Science*, pages 409--419.

Firoj Alam, Hamdy Mubarak, Wajdi Zaghouani, Giovanni Da San Martino, and Preslav Nakov. 2022. Overview of the WANLP 2022 shared task on propaganda detection in arabic. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop, WANLP@EMNLP 2022, Abu Dhabi, United Arab Emirates (Hybrid), December 8, 2022*, pages 108--118. Association for Computational Linguistics.

Maysoon Alkhair, Karima Meftouh, Kamel Smaïli, and Nouha Othman. 2019. An arabic corpus of fake news: Collection, analysis and classification. In *Arabic Language Processing: From Theory to Practice: 7th International Conference, ICALP 2019, Nancy, France, October 16--17, 2019, Proceedings 7*, pages 292--302. Springer.

Ramy Baly, Mitra Mohtarami, James Glass, Lluís Màrquez, Alessandro Moschitti, and Preslav Nakov. 2018. Integrating stance detection and fact checking in a unified corpus. *arXiv preprint arXiv:1804.08012*.

GHIZLANE Bourahouat, MANAR ABOUREZQ, and NAJIMA DAOUDI. 2023. Systematic review of the arabic natural language processing: Challenges, techniques and new trends. *Journal of Theoretical and Applied Information Technology*, 101(3).

Dr Bharat Dhiman. 2023. Ethical issues and challenges in social media: A current scenario. *Available at SSRN 4406610*.

Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *Int. J. Comput. Vis.*, 129(6):1789--1819.

Xu Han, Zhengyan Zhang, Ning Ding, Yuxian Gu, Xiao Liu, Yuqi Huo, Jiezhong Qiu, Yuan Yao, Ao Zhang, Liang Zhang, et al. 2021. Pre-trained models: Past, present and future. *AI Open*, 2:225--250.

Fouzi Harrag and Mohamed Khalil Djahli. 2022. Arabic fake news detection: A fact checking based deep learning approach. 21(4).

Maram Hasanain, Firoj Alam, Hamdy Mubarak, Samir Abdaljalil, Wajdi Zaghouani, Preslav Nakov, Giovanni Da San Martino, and Abdelhakim Freihat. 2023. Araieval shared task: Persuasion techniques and disinformation detection in arabic text. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore. Association for Computational Linguistics.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the nlp world. *arXiv preprint arXiv:2004.09095*.

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. *arXiv preprint arXiv:2308.03051*.

Yida Mu, Pu Niu, and Nikolaos Aletras. 2022. Identifying and characterizing active citizens who refute misinformation in social media. In *Proceedings of the 14th ACM Web Science Conference 2022*, pages 401--410.

Hamdy Mubarak, Samir Abdaljalil, Azza Nassar, and Firoj Alam. 2023. Detecting and reasoning of deleted tweets before they are posted. *arXiv preprint arXiv:2305.04927*.

Mona Nasery, Ofir Turel, and Yufei Yuan. 2023. Combating fake news on social media: A framework, review, and future opportunities. *Communications of the Association for Information Systems*, 53(1):9.

Ali Bou Nassif, Ashraf Elnagar, Omar Elgendy, and Yaman Afadar. 2022. Arabic fake news detection based on deep contextualized embedding models. *Neural Computing and Applications*, 34(18):16019--16032.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

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

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38--45.

AraDetector at ArAIEval Shared Task: An Ensemble of Arabic-specific pre-trained BERT and GPT-4 for Arabic Disinformation Detection

Ahmed Bahaaulddin
Middle Technical University
ahmedbahaaulddin@mtu.edu.iq

Vian Sabeeh
Middle Technical University
viantalal@mtu.edu.iq

Hanan M. Belhaj
The Libyan Academy
h.belhaj@it.lam.edu.ly

Serry Sibaee
Serrytowork@gmail.com

Samar Ahmad
Samar.sass6@gmail.com

Ibrahim Khurfan
ibraheemkhurfan@gmail.com

Abdullah I. Alharbi
King Abdulaziz University
aamalharbe@kau.edu.sa

Abstract

The rapid proliferation of disinformation through social media has become one of the most dangerous means to deceive and influence people's thoughts, viewpoints, or behaviors due to social media's facilities, such as rapid access, lower cost, and ease of use. Disinformation can spread through social media in different ways, such as fake news stories, doctored images or videos, deceptive data, and even conspiracy theories, thus making detecting disinformation challenging. This paper is a part of participation in the ArAIEval competition that relate to disinformation detection. This work evaluated four models: MARBERT, the proposed ensemble model, and two tests over GPT-4 (zero-shot and Few-shot). GPT-4 achieved micro-F1 79.01% while the ensemble method obtained 76.83%. Despite no improvement in the micro-F1 score on the dev dataset using the ensemble approach, we still used it for the test dataset predictions. We believed that merging different classifiers might enhance the system's prediction accuracy.

1 Introduction

Approximately 66% [1]of individuals in the Middle East utilize social media to seek out daily news. The rise of rapid development in social media and online communication, such as chat platforms (WhatsApp, Facebook Messenger, Snapchat, and LINE), have emerged as prevalent means to facilitate the widespread dissemination of disinformation at an unprecedented pace. Disinformation is the phenomenon that refers to how individuals or groups can be deceived or manipulated by false or misleading information. As disinformation spreads gradually, it can boost existing biases, polarize viewpoints, and hinder constructive dialogue, compromising the collaborative spirit essential for a healthy democracy. This far-reaching phenomenon can affect opinion decision-making and can threaten different foundations of democratic societies by eroding public trust in different institutions and planting seeds of divisions among communities (Himdi et al., 2022; Shu et al., 2020; Freelon and Wells, 2020). To detect disinformation and prevent it from spreading, modern methods use transformer-based architectures that are trained specifically on Arabic text and are available in public, such as AraBERT (Antoun et al., 2020) and MARBERT (Abdul-Mageed et al., 2021).

This paper outlines our participation in the disinformation detection (Task 2) of the ArAIEval Shared Task (Hasanain et al., 2023). We introduce a method incorporating three distinct classifiers: the MARBERT Pre-trained Language Model (PLM) and both zero-shot and few-shot models. Our objective is to improve the accuracy of disinformation identification in tweets by adopting a majority voting ensemble strategy. The subsequent sections are structured as follows: Section 2 reviews prior studies; Section 3 describes our proposed method; Section 4 details our experimental result; and finally, we conclude with a summarisation of our main findings.

2 Related Work

Nowadays, various types of disinformation have swiftly disseminated across social media platforms and digital news outlets, Each possessing distinct attributes and objectives to deceive and influence people. Due to the simplicity of sharing data online, it has become challenging to differentiate between trustworthy information and fake ones (Aïmeur et al., 2023). Many research studies have been con-

[1] https://www.mideastmedia.org/survey/2017/chapter/social-media/#s225

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 530–535
December 7, 2023 ©2023 Association for Computational Linguistics

ducted to detect disinformation. In this section, we provide a concise overview of the recent research on disinformation. In Their Study ABOUT DISINFORMATION DETECTION, Bahurmuz N. et al. 2017 used two transformers, AraBERT and MARABERT. The proposed paradigm removes all the non-textual, non-linguistic URL features to get a real dataset. Two sampling techniques have been used to solve the unbalanced dataset. Transformers models were trained by fine-tuning the hyper-parameters using freezing model techniques. MARABERT shows better performance results (Bahurmuz et al., 2022). In 2021, Al-Yahya M et al. examined various neural networks and transformer models for Arabic fake news detection. The experiments were conducted by using document and word embedding to test multiple neural network models like CNN, RNN, and GRU, then compared to the transformers like (ARABERT V1, ARABERT V2, ArElectra, QARiB, Arbert, and Marbert. QARiB obtained high accuracy scores compared to the limitation of small data size, repeated tweets, and noisy tweets that do not belong to any class (Al-Yahya et al., 2021).

ALbalawi. R. et al. 2022 proposed a model that relies on textual visual features to detect disinformation. They used MRABERT for text feature extraction, while RESNET-50 was used to extract image features. The text and visual features were combined and input into one multi-modal classifier to detect rumors from non. Early fusion of features achieved an accuracy of 0.85. The efficiency of the proposed model could not outperform the text-based models in accuracy. This is due to the size of the dataset (R. M. Albalawi et al., 2023).

Obeidat R. et al. (2022) worked on collecting a dataset related to COVID-19 disinformation news from Twitter; this dataset was the first Arabic COVID-19 dataset comprising about 6.7K tweets. Word cloud has been used to obtain crucial words to analyze both real and fake news. They also prepared a version of ARaBert trained based on COVID-19 tweets known as AraBERT-COV19. To reach a more accurate result than previous models, authors have been dependent on preparing and labeling the collected dataset manually. (Obeidat et al., 2022).

Hate speech and fake news can work together as a powerful weapon against society; for example, an article claiming that a particular group of people is planning to commit violence can justify hate speech against that group. This can lead to real-world violence, as seen in cases such as the Rohingya genocide in Myanmar (Doncel-Martín et al., 2023). A study by researchers at the University of Southern California's Information Sciences Institute found that 20 percent of tweets containing hate speech were also fake (Zheng et al., 2020).; Therefore, Ameur M. et al. (2021) used fine-tuned two pre-trained models, AraBERT COV19" and "mBERT COV19. The work aimed to build a model that can detect fake news about COVID-19 and hate speech simultaneously (Ameur and Aliane, 2021).

3 Methodology

The proposed system is composed of three distinct models. Raw tweets were prepared and preprocessed as inputs to the models, as outlined in Section 3.1. Sections 3.2, 3.3, and 3.4 explain the three models incorporated using an ensemble technique, as clarified in Section 3.4.

3.1 Preprocessing

Pre-processing was conducted using a methodology previously employed by various researchers (Duwairi and El-Orfali, 2014; Abu Farha and Magdy, 2019). The initial step involved eliminating unfamiliar symbols and characters, such as letters from different languages, punctuation, and diacritics. Emojis were retained because they may be used to express hate, obscenity, and abusive content (Mubarak et al., 2023). Additionally, certain letters that exhibited diverse forms within the original tweets were standardized to a singular form. For instance, characters like 'hamza' {إ,أ} were substituted with {ا}, and the 't marabout {ة} was changed to ه}.

3.2 Fine-tuning pre-trained Language Models

Due to the contextual nature of disinformation textual content, contextualized language models would be beneficial in addressing this task. Transformer architectures like BERT (Devlin et al., 2019) have demonstrated exceptional success across diverse NLP tasks. Our study employed three Arabic language models that have attained cutting-edge performance in various Arabic NLP applications. These models were fine-tuned for disinformation detection, enabling us to conduct a comparative analysis of their capabilities. The specific models employed are as follows:

AraBERT: Antoun et al. (2020) introduced a BERT-based model explicitly trained for the Arabic language. It emerged as the first Arabic-specific BERT model to achieve competitive results across most Arabic NLP tasks. This model was pre-trained on an extensive dataset encompassing 24 GB of text sourced from Wikipedia and various news outlets across the Arab region.

MARBERT: As presented by Abdul-Mageed et al. (2021), this model was designed for transfer learning in Arabic dialects. MARBERT's pre-training involved a massive dataset comprising 6 billion tweets, leading to state-of-the-art performance across multiple Arabic-language NLP tasks.

QARiB: Developed by Abdelali et al. (2021), this model underwent training using a mix of Modern Standard Arabic (MSA) and dialectal sources. The training dataset encompassed approximately 420 million tweets and 180 million sentences from news articles. Notably, the utilization of this combination of sources, comprising MSA and dialectal content for language model pre-training, is observed to enhance performance in classification tasks, according to the author's observations.

3.3 Large Language Models (LLMs)

Large Language Models (LLMs) have recently become essential in Natural Language Processing (NLP). They effectively utilize vast knowledge sources and deeply comprehend complex language details (Alyafeai et al., 2023; Zhang et al., 2023). One significant model in this area is OpenAI's GPT-4, an advanced language model supported by a transformer architecture and a massive 1.76 trillion parameters (OpenAI, 2023). While its effectiveness can vary by task, its strengths in sentiment analysis and emotion detection highlight its utility (Wang et al., 2023). Thus, the exploration of such models is essential for other NLP studies.

Zero-shot: We use GPT 4 as a zero-shot classifier; the model was never trained explicitly on our task. The key to our approach lies in the prompting strategy. Constructing effective prompts is vital; it is an implicit instruction to guide the model to understand and perform the desired classification, ensuring accurate and reliable outputs. Figure 1 illustrates an example of a zero-shot prompt, highlighting instructions for data and categorization. Considering GPT-4's potential, we designate its role as an "annotator expert." We introduce labels to steer the LLMs alongside the primary directive. The guidance specifies the format of the LLMs' responses, seeking to make any other adjustments.

Few-shot: The foundational research from Brown et al. (2020) highlighted the enhanced outcomes of few-shot learning relative to zero-shot configurations. In our work, we used a few-shot setting leveraging GPT-4. We selected nine examples from the available training data rather than selecting samples at random. To achieve this, we used the "sentence-transformers" library to obtain embeddings for Arabic tweets. It starts by choosing a random tweet from the training set and then iteratively picks the most dissimilar tweet based on cosine distance from the already selected ones. Specifically, each addition computes the sum of lengths to all previously selected tweets, ensuring a diverse selection. The "distiluse-base-multilingual-cased-v2" model is used for multilingual support, including Arabic. Since the proportion of the disinformation class in the training data set is small, we chose to increase the number of disinformation tweets (6 examples) compared to (3 models) for the class that does not contain misleading information. For each category, we applied the dissimilarity above selected samples approach. Figure 3 illustrates the details of the utilized prompt.

3.4 Ensemble

At this step, we have three individual classifiers: the best-performing Pre-trained Language Model (PLM) MARBERT, zero-shot, and few-shot models. Each model's output is a determination of whether a tweet is disinformation or not. By using different classifiers together, we can reduce their individual weaknesses and benefit from their strengths. Using an ensemble method, we employed a majority voting approach to merge the classifiers. We assume that combining multiple classifiers might generalize predictions on unseen tweets. This is based on the idea that multiple models may capture different aspects or features of the tweet, leading to a more comprehensive and reliable decision when combined.

4 Result

Given the nature of the shared tasks, we conducted our initial experiments on the development dataset and accordingly selected the best-performing method for delivering predictions on the test dataset. The organizers of this shared task have shared an annotated dataset sourced from

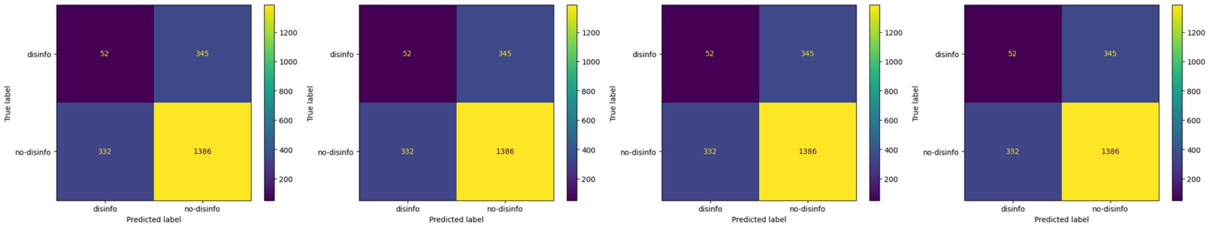


Figure 1: The confusion matrices for the three voter models: a) MARBERT, b) few-shot, c) zero-shot and d) our proposed method.

Pre-processing	AraBERT	QARiB	MARBERT
No	68.16%	69.04%	68.27%
Yes	68.64%	69.22%	**69.79%**

Table 1: Performance results (micro-F1) for fine-tuning three PLMs on the Disinformation Detection task.

Twitter (Hasanain et al., 2023). This dataset is noteworthy for being one of the most extensive publicly accessible Arabic datasets focused on disinformation content [2]. It contains 14126 tweets as a training set and 2115 tweets as a development set. Specifically, we compared three pre-trained models as well as the GPT models in Zero-Shot and View-Shot settings. We used the official evaluation measure adopted by the organizers (micro-F1).

We performed experiments to assess three PLMs trained specifically for Arabic: AraBERT, QARiB, and MARBERT. Each model was fine-tuned on the provided training set, and their performances were measured on the dev dataset using the official metric (micro-F1). Table 1 presents the performance results for fine-tuning three PLMs on the dev dataset. MARBERT showed the highest performance, securing a micro-F1 of 0.698, while QARiB was a close second at 0.693. When it comes to examining the pre-processing impact, the performance of models with preprocessing is better than without, with varying effects. MARABERT Results improved relatively with the use of preprocessing (1.52%), followed by an improvement of (0.68%), compared to a slight improvement of (0.18%) for QARiB model.

Additionally, we used two experimental settings: zero-shot and few-shot prompting strategies. Due to the cost of using such models, we used the pre-processed text based on previous experiments' findings. In future work, we will study the impact of pre-processed text for GPT models on Arabic user-generated text extracted from social media. Table 2 presents the performance of zero-shot and few-shot classifiers and our proposed ensemble approach. We observe that the performance of the zero-shot setup is generally higher than the few-shot setting, with a significant improvement of 15% (micro-F1). However, we found that the few-shot setting excelled when it came specifically to the disinformation class, as it predicted 337 tweets out of 397, while the zero-shot setting only recognized 168 tweets. This shows the importance of providing generalized examples, as we explained in Section 3.3. In future work, we will study the effect of the number of examples in general and their proportion for each class.

Finally, after studying and analyzing the performance of the models, we decided to take advantage of each one of them using a majority voting approach to merge three classifiers: MARBERT, few-shot and zero-shot prompting strategies. Although the micro-F1 score on the dev dataset was not improved using the ensemble approach, we used it to deliver predictions of the test dataset. We hypothesized that the system's ability to generalize by combining different classifiers may balance out better classification prediction. Table 2 presents the performance of our proposed ensemble approach on the test set, which is the official result for our participant. Figure 1 presents the confusion matrices for the three voter models and our proposed method.

5 Conclusion

Disinformation on social media may be biased in the society's collective opinion. Consequently, this may lead to social abuse action. Accordingly, social media needs an apparatus to help people reveal false claims. This study used three Arabic transformers for comparison (AraBERT, MARBERT, QARIB). From the experiments, we conclude that

[2] https://gitlab.com/araieval/wanlp2023_araieval/-/tree/main/task2

Model	macro-F1 disinfo	macro-F1	micro-F1
Dev Dataset			
MARBET	15.13%	48.84%	69.79%
Few-shot	41.86%	53.07%	55.74%
Zero-shot	43.08%	65.10%	79.01%
Ensemble	43.42%	64.43%	76.83%
Test Dataset - Formal submersion			
Ensemble	-	64.98%	74.87%

Table 2: Performance results for the three voter models: (MARBERT,few-shot and zero-shot) and our proposed method on the dev and test dataset.

there is an influence of pre-processing on model performance. To reach a generalized approach, two settings for the test were conducted depending on GPT-4: few-shot and zero-shot and one proposed ensemble learning. Zero-shot by GPT-4 achieves the best performance. Even though the ensemble approach did not boost the micro-F1 score on the dev dataset, we employed it in the test dataset, assuming that integrating various classifiers might improve prediction accuracy.

References

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

Muhammad Abdul-Mageed, AbdelRahim Elmadany, et al. 2021. Arbert & marbert: Deep bidirectional transformers for arabic. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7088–7105.

Ibrahim Abu Farha and Walid Magdy. 2019. Mazajak: An online Arabic sentiment analyser. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 192–198, Florence, Italy. Association for Computational Linguistics.

Maha Al-Yahya, Hend Al-Khalifa, Heyam Al-Baity, Duaa AlSaeed, and Amr Essam. 2021. Arabic Fake News Detection: Comparative Study of Neural Networks and Transformer-Based Approaches. *Complexity*, 2021:5516945. Publisher: Hindawi.

Zaid Alyafeai, Maged S Alshaibani, Badr AlKhamissi, Hamzah Luqman, Ebrahim Alareqi, and Ali Fadel. 2023. Taqyim: Evaluating arabic nlp tasks using chatgpt models. *arXiv preprint arXiv:2306.16322*.

Mohamed Seghir Hadj Ameur and Hassina Aliane. 2021. AraCOVID19-MFH: Arabic COVID-19 Multi-label Fake News and Hate Speech Detection Dataset. ArXiv:2105.03143 [cs].

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.

Esma Aïmeur, Sabrine Amri, and Gilles Brassard. 2023. Fake news, disinformation and misinformation in social media: a review. *Social Network Analysis and Mining*, 13(1):30.

Naelah O Bahurmuz, Ghada A Amoudi, Fatmah A Baothman, Amani T Jamal, Hanan S Alghamdi, and Areej M Alhothali. 2022. Arabic Rumor Detection Using Contextual Deep Bidirectional Language Modeling. *IEEE Access*, 10:114907–114918. Publisher: IEEE.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

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. Association for Computational Linguistics.

Israel Doncel-Martín, Daniel Catalan-Matamoros, and Carlos Elías. 2023. Corporate social responsibility and public diplomacy as formulas to reduce hate speech on social media in the fake news era. *Corporate Communications: An International Journal*, 28(2):340–352. Publisher: Emerald Publishing Limited.

Rehab Duwairi and Mahmoud El-Orfali. 2014. A study of the effects of preprocessing strategies on sentiment analysis for Arabic text. *Journal of Information Science*, 40(4):501–513.

Deen Freelon and Chris Wells. 2020. Disinformation as political communication.

Maram Hasanain, Firoj Alam, Hamdy Mubarak, Samir Abdaljalil, Wajdi Zaghouani, Preslav Nakov, Giovanni Da San Martino, and Abdelhakim Freihat. 2023. ArAIEval Shared Task: Persuasion Techniques and Disinformation Detection in Arabic Text. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore. Association for Computational Linguistics.

Hanen Himdi, George Weir, Fatmah Assiri, and Hassanin Al-Barhamtoshy. 2022. Arabic fake news detection based on textual analysis. *Arabian Journal for Science and Engineering*, 47(8):10453–10469.

Hamdy Mubarak, Sabit Hassan, and Shammur Absar Chowdhury. 2023. Emojis as anchors to detect arabic offensive language and hate speech. *Natural Language Engineering*, page 1–22.

Rasha Obeidat, Maram Gharaibeh, Malak Abdullah, and Yara Alharahsheh. 2022. Multi-label multi-class COVID-19 Arabic Twitter dataset with fine-grained misinformation and situational information annotations. *PeerJ Computer Science*, 8:e1151.

OpenAI. 2023. Gpt-4 technical report.

R. M. Albalawi, A. T. Jamal, A. O. Khadidos, and A. M. Alhothali. 2023. Multimodal Arabic Rumors Detection. *IEEE Access*, 11:9716–9730.

Kai Shu, Amrita Bhattacharjee, Faisal Alatawi, Tahora H Nazer, Kaize Ding, Mansooreh Karami, and Huan Liu. 2020. Combating disinformation in a social media age. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10(6):e1385.

Zengzhi Wang, Qiming Xie, Zixiang Ding, Yi Feng, and Rui Xia. 2023. Is chatgpt a good sentiment analyzer? a preliminary study.

Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B Hashimoto. 2023. Benchmarking large language models for news summarization. *arXiv preprint arXiv:2301.13848*.

Han Zheng, Jinhui Li, Charles T. Salmon, and Yin-Leng Theng. 2020. The effects of exergames on emotional well-being of older adults. *Computers in Human Behavior*, 110:106383.

rematchka at ArAIEval Shared Task: Prefix-Tuning & Prompt-tuning for Improved Detection of Propaganda and Disinformation in Arabic Social Media Content

Reem Abdel-Salam
Cairo University, Faculty of Engineering, Computer Engineering / Giza, Egypt
reem.abdelsalam13@gmail.com

Abstract

The rise of propaganda and disinformation in the digital age has necessitated the development of effective detection methods to combat the spread of deceptive information. In this paper, we present our approach proposed for the ArAIEval shared task: propaganda and disinformation detection in Arabic text. Our system utilized different pre-trained BERT based models, that make use of prompt-learning based on knowledgeable expansion and prefix-tuning. The proposed approach secured third place in subtask-1A with a 0.7555 F1-micro score, and second place in subtask-1B with a 0.5658 F1-micro score. However, for subtask-2A & 2B, the proposed system achieved fourth place with an F1-micro score of 0.9040, and 0.8219 respectively. Our findings suggest that prompt-tuning-based & prefix-tuning based models performed better than conventional fine-tuning. Furthermore, using loss-aware class imbalance, improved performance.

1 Introduction

With the growing popularity of social media in our current society, platforms such as Twitter, and Reddit have become critical tools for influencing people. People on social media prefer to express their opinions, points of view more freely, and share information. However, these platforms can be used to deceive and manipulate individuals. In addition to spreading rumors, and fake news. This can be done through propaganda techniques. Propaganda refers to the systemic dissemination of information, ideas, or opinions, often through biased or misleading means, with the intention of influencing or manipulating public perception, attitudes, behaviors, or beliefs. It is a persuasive communication technique employed by individuals, organizations, or governments to shape public opinion and advance specific agendas. The rise of propaganda and disinformation has necessitated the development of effective detection methods to combat the spread of deceptive information. With the advent of pre-trained language models, there has been a significant advancement in the field of natural language processing (NLP), offering promising opportunities for combating the dissemination of false information. Several works have been proposed to improve the identification of persuasion techniques in text as the recent shared-task propaganda detection in Arabic (Alam et al., 2022). (Samir et al., 2022) and (Laskar et al., 2022) utilized AraBERT for this task. (Attieh and Hassan, 2022) utilized A multi-Task learning model, which includes a shared AraBERT encoder and task-specific binary classification layers. This model has been trained to learn one binary classification task per propaganda approach jointly. In this paper, we present our solution to the ArAIEval shared task (Hasanain et al., 2023). The ArAIEval shared task is held with the 1st Arabic Natural Language Processing Conference co-located with the EMNLP 2023. The goal of the task is to build models for identifying propaganda and disinformation in Arabic content. The shared task consists of two tasks. The first task is persuasion technique detection in Arabic text. The second task is disinformation detection in the text.

This paper describes the system developed for addressing propaganda and disinformation detection in text, for both subtasks. Given that a key challenge in this task is the unbalanced distribution of the dataset. Additionally, the contextual nature of language and the cultural nuances involved in the text. We follow best practices from recent work on enhancing model generalization and robustness, by using different parameter-efficient techniques (PEFT), contrastive loss, adversarial training, and loss-aware class imbalance methods. The Parameter-Efficient Fine Tuning (PEFT) is a technique used to improve the performance of pre-trained language models on specific downstream tasks. PEFT methods freeze the pretrained model

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 536–542
December 7, 2023 ©2023 Association for Computational Linguistics

parameters during fine-tuning and put a few trainable parameters (the adapters) on top of it. The adapters are taught how to pick up knowledge appropriate to a given task. PEFT of pre-trained language models has recently demonstrated remarkable results, effectively matching the performance of full fine-tuning while utilizing significantly fewer trainable parameters (Fu et al., 2023; Liu et al., 2022; Wang et al., 2022), thereby addressing storage and communication constraints. Such approaches include prefix-tuning (Li and Liang, 2021), prompt-tuning (Hu et al., 2021b), soft-prompting (Lester et al., 2021) and LoRa (Hu et al., 2021a). Adversarial training (AT) (Goodfellow et al., 2014) is a method to improve the model's resistance to adversarial examples and acts as a regularizer. The key is to disturb the input example using a gradient-based perturbation, and then train the model on both clean and perturbed examples. Contrastive loss is one of the first training objectives that was used for contrastive learning. It takes as input a pair of samples that are either similar or dissimilar, and it brings similar samples closer and dissimilar samples far apart in embedding space (Khosla et al., 2020). Such loss has shown model performance improvement compared to cross-entropy on multiple problems (Chi et al., 2022; Chen et al., 2022; Pan et al., 2022).

The rest of the paper goes as follows: section 2 gives an overview of the dataset, section 3 discusses the proposed methods, section 4 shows experimental results, and section 5 concludes the paper.

2 Data

The dataset used has been provided by the organizers for the ArAIEval shared task. Table 1 summarizes the distribution of the provided dataset. For subtask-1A the dataset consists of the text of Arabic tweets, the type of the text whether it is a tweet or text, and the label. The train, validation, and consist of 2427, and 259 examples. The provided data is unbalanced as for the non-persuasion class 509 is presented whilst, the other class 1918 example is presented. For subtask-2A&2B the provided dataset consists of the text and the label. In subtask-2A, the distribution of labels in the train-set goes as follows: 2656 examples for the disinformation class, and 11491 examples for the non-disinformation text class. The distribution of labels in subtask-2B in the train-set is as follows: hate speech 1512 examples, 453 examples for the spam class, 500 examples for the offensive class, and 191 examples for the rumor class. Accordingly, a major issue in this dataset is the nature of the unbalance of the class distribution, which poses a challenge.

Task	Train-size	Dev-size	Test-size
Subtask-1A	2427	259	503
Subtask-1B	2427	259	503
Subtask-2A	14147	2111	3729
Subtask-2B	2656	397	876

Table 1: Distribution of the provided dataset

3 Methodology

This section presents the various approaches used while developing the final models: a weighted ensemble of BERT-based models.

3.1 Task-1

Task-1 was composed of two subtasks, subtask-1A and subtask-1B. The goal of subtask-1A is to detect whether a given text contains content with a persuasion technique. The goal of subtask-1B is to identify which of the 24 propaganda techniques is used in a given text. In order to address these subtasks, we tried a variety of ways. The majority of the models employed were BERT-based, such as MARBERT (Abdul-Mageed et al., 2020) and AraBERT (Antoun et al., 2020).

subtask-1A In subtask-1A two methods were used: conventional fine-tuning and prefix-tuning. In order to make the model more robust so that similar inputs derive semantically similar outcomes two approaches were explored fast gradient methods (FGM) (Wang et al., 2021) and supervised contrastive learning (Chen et al., 2022). In addition, back-translation between Arabic and English languages was used as an augmentation, to upsample the dataset for the lower class. Prefix tuning is an additive technique that only attaches a continuous set of task-specific vectors to the input's beginning. In each layer of the model, the hidden states are only added and the prefix parameters are optimised. The input sequence's tokens can still serve as virtual tokens to the prefix. Fast Gradient Method (FGM), is a popular technique for generating adversarial examples. It works by adding small, carefully crafted perturbations to the input data, in our case, the perturbations are added to the model's embedding, such that the model's prediction changes to a wrong answer. The Fast Gradient Method is based on the concept of a "fast gradient" - a gradient that is calculated with respect to the input data, instead of the model's parameters.

subtask-1B The challenge of this subtask was to correctly identify labels for each text, in a given unbalanced dataset. To address these issues two approaches have been investigated: 1) loss aware class imbalance such as Asymmetric loss for multi-label classification (Ridnik et al., 2021), and Distribution Balanced Loss (Wu et al., 2020) 2) balanced data-Sampler for multi-label problems. In this task all models were trained using prefix-tuning.

3.2 Task-2

Task-2 was composed of two subtasks, subtask-2A and subtask-2B. The goal in subtask-2A is to classify whether a given text is disinformation or not. However, in subtask-2B the goal was to predict the disinformation class of a given text.

subtask-2A In this subtask, the same experiments conducted in subtask-1A were used in this subtask.

subtask-2B In this subtask, prompt-tuning was utilized using openprompt library (Ding et al., 2021). Prompt tuning is the process of feeding front-end prompts into the model in the context of a specific task. These prompts could be either text related to the task or virtual tokens. Prompt tuning is used to guide a model toward a particular prediction. Prompts are only introduced into the input embedding sequence and this embedding is fed to the language model head and output to the linear classification head, as shown in the figure 1. One of the difficulties in promoting is the design of the prompt and the model's output. For the prompt, we used [MASK] فئة المعلومات المضللة ("The disinformation class is [MASK]"), and For the output, we have used label names translated into Arabic. Two models were used: AraBERT and AraGPT.

Experimental Set-up for the fine-tuned models the learning rate was set to 4e-5 or 4e-6, a cosine-annealing learning rate scheduler was used, the model's weight decay was set to 1e-8 and the length of the sentence for tokenization was set to 128 or 256. During training, batch size was set to 32, and at the end of each epoch, the model was evaluated on dev-set. The best-performing model in terms of F1-micro is saved.

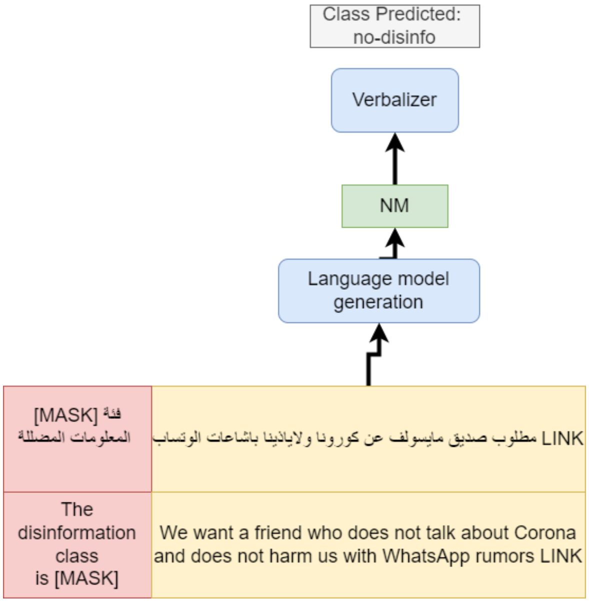

Class Predicted:
no-disinfo
Verbalizer
NM
Language model generation
[MASK] فئة المعلومات المضللة
مطلوب صديق مايسولف عن كورونا ولايإذينا باشاعات الوتساب LINK
The disinformation class is [MASK]
We want a friend who does not talk about Corona and does not harm us with WhatsApp rumors LINK

Figure 1: Prompt-tuning architecture.

4 Results and Discussion

In this section, The performance of the model is reported based on the official metric during dev-phase and test-phase. The official metric used for all tasks is the micro average F1-score. Table 3 shows results for subtask-1A on dev-set and test-set. In the dev-set, the outperforming model was Arabert v2 with prefix-tuning, which comes in second place with MARBERT with prefix-tuning and contrastive learning. Surprisingly, the performance of the model is switched in the test-set. It is noticed that high performance in dev-set does not necessarily mirror the test-set. The reason behind it is the nature of the training. For instance, contrastive loss and FGM make models robust so that similar inputs derive in semantically similar outcomes. Table 4 shows results in subtask-1B. It could be concluded that class-aware loss function with a balanced sampler improves model performance over simple binary cross-entropy loss with random samplers. Table 9 and 2 show results in subtask-2A and subtask-2B, similar to subtask-1A outperforming models in dev-set are interleaved in test-set. Tables 5,6,7, and 6 shows different teams run in the shared task. For single models, in table 3 both Arabert with Prefix-tuning and MARBERT with Prefix-tuning contrastive loss with Cross entropy loss show high competence with submitted models on the leaderboard 5, as they could have secured first and second places. For subtask-1B based on tables 6,4, the ensemble model seems to be on par with single models. Since, Arabert with Asymmetric Loss for a single model run, shows similar results to the ensemble and would have secured the same place in leaderboard. For subtask-2A&2B based on tables 9,7, 2 and 8, the ensemble model seems to be the best solution over single models. Non of the single models could have secured a higher place than the ensemble model.

4.1 Error Analysis

Further investigations have been carried out to analyze the potential limitations of the system. For subtask-1A, the model could not correctly identify the following text into the correct class: persuasion class.

شهدت مجموعة من مدن المملكة، اليوم الجمعة
(١٦ أكتوبر)، أول صلاة جمعة في زمن كورونا،
بعد أزيد من ٧ شهور من تعليقها، من طرف
السلطات للحد من تفشي فيروس كورونا المستجد.

Today, Friday (October 16), a group of cities in the Kingdom witnessed the first Friday prayer in the time of Corona, more than 7 months after it was suspended by the authorities to limit the spread of the new Coronavirus. The reason behind this is that the model has no knowledge of previous information about coronavirus lockdown, and its consequences. Therefore, it is hard to assess the facts in the text. Another miss-classification error, where true class is non-persuasion is

عترف بأن حزناً عميقاً راح يعبر القلب لحظة
وقوع ذلك الطائر المهاجر الذي كان يقصد تلك
الفيافي ليرتاح على حصاها قليلاً ثم لا يلبث أن
يغادر المكان الى حيث الدفء الذي يبحث عنه..
فلم يجد الا الغدر وخيانتي للضيف.

He admitted that a deep sadness began to cross his heart the moment that migratory bird fell, which was heading to that desert to rest on its pebbles for a little while, and then quickly left the place for the warmth he was looking for.. He found nothing but treachery and my betrayal of the guest. The model failed to understand that the provided text is a poem rather than a piece of news. So it could be concluded that some of the errors are related are due to the model not able to handle different domains and gain knowledge about them and their differences.

Model	F1-micro Dev-set	F1-micro test set
Arabert v2	78	81
Aragpt	79	77
Final Model	-	82.19

Table 2: Results on our dev-set and test-set for the developed models in subtask-2b

Figure 2, shows model MARBERT performance in subtask-2B. The model confuses between hate speech class and the offensive class. As well as, between the offensive class and the rumor class.

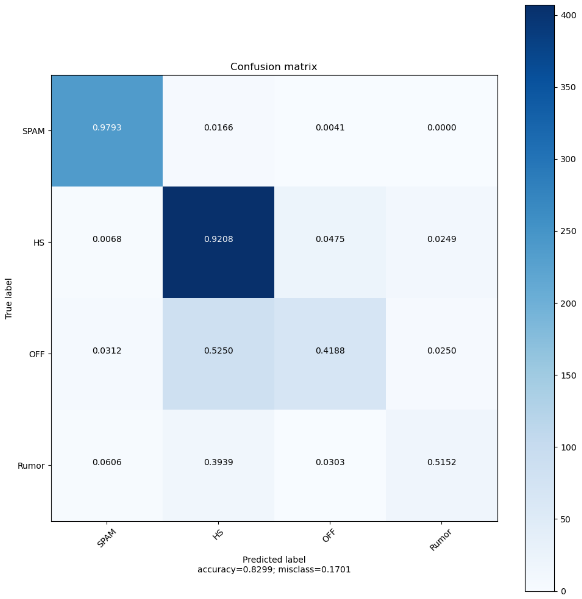


Figure 2: Confusion matrix of the predictions of the submission-3 model in subtask 1 on the dev-set.

5 Conclusion

In this paper, the results and the main findings of ArAIEval shared task were presented, in which different experiments were carried out with MARBERT, Arabert v2, and Aragpt models. Our models secured third place in subtask-1A, second place in subtask-1B, and Fourth place in subtask-2A&2B. Our proposed solution is an ensemble of different BERT-based models. These Models are developed differently, some are trained using prefix-tuning, and others are trained using fine-tuning and prompt-tuning. leverages fine-tuned, per-trained models. In addition, training tricks were utilized as FGM, contrastive learning, and balanced sampler. In future efforts, we plan to further improve our model to better handle data-imbalance constraints and world knowledge needed to improve model performance.

Model	Technique	F1-micro Dev-set	F1-micro test set
Arabert v2	Prefix-tuning	84.8	75.8
	Prefix-tuning Back Translation	77	72.7
	Prefix-tuning FGM	85	74.2
	Prefix-tuning Type of text specificed	85.9	73.9
	Prefix-tuning Focal loss	83.8	75
	Fine-tuning	83.6	72.2
MARBERT	Prefix-tuning contrastive loss with Cross entropy loss	84.5	76.5
Final Model	Ensemble	-	75.55

Table 3: Results on our dev-set and test-set for the developed models in subtask-1A

Model	Technique	F1-micro Dev-set	F1-micro Test-set
Arabert V2	Resample Loss Sampler	66	54
	Binary Cross Entropy Loss	62	51
	Asymmetric Loss Sampler	64.89	56
Final Model	Ensemble	-	56.58

Table 4: Results on our dev-set and test-set for the developed models in subtask-1B

Team	Micro F1
HTE	76.34
KnowTellConvince	75.75
rematchka	75.55
UL & UM6P	75.15

Table 5: Leaderboard results on test-set for subtask-1A

Team	Micro F1
UL&UM6P	56.66
rematchka	56.58
AAST-NLP	55.22
Itri Amigos	55.06

Table 6: Leaderboard results on test-set for subtask-1B

Team	Micro F1
DetectiveRedasers	90.48
AAST-NLP	90.43
UL&UM6P	90.40
rematchka	90.40
PD-AR	90.21

Table 7: Leaderboard results on test-set for subtask-2A

Team	Micro F1
DetectiveRedasers	8356
UL&UM6P	83.33
AAST-NLP	82.53
rematchka	82.19
superMario	8.208

Table 8: Leaderboard results on test-set for subtask-2B

Model	Technique	F1-micro Dev-set	F1-micro test set
Arabert v2	Prefix-tuning	88.3	89.2
Arabert v2	Prefix-tuning Back Translation	90.01	89.5
Arabert v2	Prefix-tuning FGM	89.8	88
Final Model	Ensemble	-	90.40

Table 9: Results on our dev-set and test-set for the developed models in subtask-2A

References

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

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, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

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

Joseph Attieh and Fadi Hassan. 2022. Pythoneers at wanlp 2022 shared task: Monolingual arabert for arabic propaganda detection and span extraction. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 534–540.

Qianben Chen, Richong Zhang, Yaowei Zheng, and Yongyi Mao. 2022. Dual contrastive learning: Text classification via label-aware data augmentation. *arXiv preprint arXiv:2201.08702*.

Jianfeng Chi, William Shand, Yaodong Yu, Kai-Wei Chang, Han Zhao, and Yuan Tian. 2022. Conditional supervised contrastive learning for fair text classification. *arXiv preprint arXiv:2205.11485*.

Ning Ding, Shengding Hu, Weilin Zhao, Yulin Chen, Zhiyuan Liu, Hai-Tao Zheng, and Maosong Sun. 2021. Openprompt: An open-source framework for prompt-learning. *arXiv preprint arXiv:2111.01998*.

Zihao Fu, Haoran Yang, Anthony Man-Cho So, Wai Lam, Lidong Bing, and Nigel Collier. 2023. On the effectiveness of parameter-efficient fine-tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 12799–12807.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Maram Hasanain, Firoj Alam, Hamdy Mubarak, Samir Abdaljalil, Wajdi Zaghouani, Preslav Nakov, Giovanni Da San Martino, and Abed Alhakim Freihat. 2023. ArAIEval Shared Task: Persuasion Techniques and Disinformation Detection in Arabic Text. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore. Association for Computational Linguistics.

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

Shengding Hu, Ning Ding, Huadong Wang, Zhiyuan Liu, Jingang Wang, Juanzi Li, Wei Wu, and Maosong Sun. 2021b. Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. *arXiv preprint arXiv:2108.02035*.

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

Sahinur Rahman Laskar, Rahul Singh, Abdullah Faiz Ur Rahman Khilji, Riyanka Manna, Partha Pakray, and Sivaji Bandyopadhyay. 2022. Cnlp-nits-pp at wanlp 2022 shared task: Propaganda detection in arabic using data augmentation and arabert pre-trained model. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 541–544.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35:1950–1965.

Lin Pan, Chung-Wei Hang, Avirup Sil, and Saloni Potdar. 2022. Improved text classification via contrastive adversarial training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11130–11138.

Tal Ridnik, Emanuel Ben-Baruch, Nadav Zamir, Asaf Noy, Itamar Friedman, Matan Protter, and Lihi Zelnik-Manor. 2021. Asymmetric loss for multi-label classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 82–91.

Ahmed Samir, Abu Bakr Soliman, Mohamed Ibrahim, Laila Hesham, and Samhaa R El-Beltagy. 2022. Ngu_cnlp at wanlp 2022 shared task: Propaganda detection in arabic. *WANLP 2022*, page 545.

Xiaosen Wang, Yichen Yang, Yihe Deng, and Kun He. 2021. Adversarial training with fast gradient projection method against synonym substitution based text attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13997–14005.

Yaqing Wang, Subhabrata Mukherjee, Xiaodong Liu, Jing Gao, Ahmed Hassan Awadallah, and Jianfeng Gao. 2022. Adamix: Mixture-of-adapter for parameter-efficient tuning of large language models. *arXiv preprint arXiv:2205.12410*, 1(2):4.

Tong Wu, Qingqiu Huang, Ziwei Liu, Yu Wang, and Dahua Lin. 2020. Distribution-balanced loss for multi-label classification in long-tailed datasets. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 162–178. Springer.

Itri Amigos at ArAIEval Shared Task: Transformer vs. Compression-Based Models for Persuasion Techniques and Disinformation Detection

Nouman Ahmed, Natalia Flechas Manrique, and **Jehad Oumer**
University of the Basque Country (UPV/EHU)
{anouman001, nflechas001, joumer001}@ikasle.ehu.eus

Abstract

Social media has significantly amplified the dissemination of misinformation. Researchers have employed natural language processing and machine learning techniques to identify and categorize false information on these platforms. While there is a well-established body of research on detecting fake news in English and Latin languages, the study of Arabic fake news detection remains limited. This paper describes the methods used to tackle the challenges of the ArAIEval shared Task 2023. We conducted experiments with both monolingual Arabic and multi-lingual pre-trained Language Models (LM). We found that the monolingual Arabic models outperformed in all four subtasks. Additionally, we explored a novel lossless compression method, which, while not surpassing pretrained LM performance, presents an intriguing avenue for future experimentation to achieve comparable results in a more efficient and rapid manner.

1 Introduction

The growing presence of social media as a way to quickly disseminate information to broad audiences, has had an undeniable shaping the sphere of public opinion. By their very nature, social media platforms have the associated peril of carrying messages that are erroneous at best, or carefully crafted to misinform and manipulate, at worst (e.g. Ishmuradova, 2019, Iida et al., 2022).

The development of NLP tools to fact check and explore persuasion techniques is a potential approach to counteract the effect of misinformation in social media. While this is an active area of research that is well established for English and other Latin languages, for Arabic news, there is still much room to explore. This paper describes the methodology used to tackle the classification tasks presented by the ArAIREval shared 2023 task (Hasanain et al., 2023), which builds upon WANLP 2022 (Alam et al., 2022). The tasks are described in Section 3.

We have mainly focused our efforts on two distinct approaches: on the one hand, the use of pre-trained Language Models (LMs), which has been an established way to achieve state-of-the-art results in a range of NLP tasks (e.g. Devlin et al., 2018, Radford et al., 2019). Pre-trained LMs are advanced models, often based on Transformer architectures, that are pre-trained on massive datasets. On the other hand, we explore the approach presented by Jiang et al., 2023, which advocates for the use of simpler models that are less resource intensive and more interpretable. This method uses lossless compression and a distance metric with a k-nearest-neighbor classifier for text classification. The inconsistencies detected in this implementation will be detailed later on. The paper is organized as follows: Section 2 briefly talks about related work, Section 3 summarizes the datasets on each sub-task and Section 4 the methodology. Finally, Sections 5 and 6 present the conclusions and limitations, respectively.

All of the source code to reproduce the results is available in a Github repository [1].

2 Related Work

Prior research in the field of automated Arabic fake news detection predominantly relied on traditional machine learning classifiers, focusing mainly on binary classification scenarios. Mahlous and Al-laith, 2021 applied NB, LR, SVM, RF, and XGB methods to classify Arabic news tweets as either fake or not. Among these, the Logistic Regression (LR) classifier achieved 87.8% accuracy using TF-IDF features at the n-grams level. Recent research has focused on assessing the performance of Transformer-based models. For example, Antoun et al., 2020 showed that AraBERT v02 achieved

[1] https://github.com/nouman-10/ArAIEval-Shared-Task/

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 543–548
December 7, 2023 ©2023 Association for Computational Linguistics

high accuracy across various experimental scenarios. Similarly, Nassif et al., 2022 achieved favorable results on a Covid-19 fake news dataset using pre-trained models like RoBERTa-Base (Liu et al., 2019), ARBERT (Abdul-Mageed et al., 2021) and Arabic-BERT (Safaya et al., 2020). Alyoubi et al., 2023 show the good performance of MARABERT with CNNs for tweet classification.

While binary classification of news content has been a traditional approach, there is an emerging interest in multi-label classification scenarios. Several studies have ventured into this realm. Argotario, as introduced by Habernal et al., 2017, is a game-based platform designed to accumulate a dataset portraying a spectrum of fallacious arguments, with labels like: *ad hominem, appeal to emotion, red herring, hasty generalization, irrelevant authority*. In parallel Da San Martino et al., 2019 extracted and analyzed 451 articles sourced from 48 news outlets. These articles were annotated to highlight 18 unique propaganda techniques. These efforts emphasize the pivotal role of multi-label classification in revealing the nuanced tactics inherent in news narratives.

3 Sub-Tasks

Task 1, `Persuasion Technique Detection`, involves identifying persuasive elements within text snippets. Subtask A focuses on determining if a given multigenre snippet (composed of tweets and news paragraphs) contains content utilizing persuasion techniques, making it a binary classification task. Subtask B expands this by requiring further identification of specific propaganda techniques employed in the same multigenre snippet, turning the task into a multilabel classification.

Task 2, `Disinformation Detection`, centers around identifying and categorizing disinformation within tweets. Subtask 2A involves a binary classification task where the goal is to determine whether a tweet contains disinformation. Subtask 2B further refines this by requiring the detection of fine-grained disinformation classes, including *hate-speech, offensive content, rumors*, and *spam*.

4 Methodology

In this section, we describe our approach to processing the data, the models, and the experiments we conducted for all tasks.

4.1 Data Preparation and Preprocessing

During data preparation, we identified that in Subtasks 2A and 2B that a significant number of data points in these subtasks had the “text” feature set to the “NaN” (Not a Number) data type. Table 1 provides a breakdown of data points of all the sub-tasks including that lack of data in the “text” feature across the train and dev sets of Subtasks 2A and 2B. To address these anomalies, we converted “NaN” entries to strings. While we contemplated removing these anomalies from the dataset, the scoring system for the SharedTask mandated that all data points in the Dev set remain present and in their original sequence. Moreover, a clear class imbalance was identified at this stage, which we tried to tackle later by adding class weights to the model training.

Upon loading the data, we structured our experiments around three preprocessing settings: 1) Raw Data Processing: in this approach, no alterations were made. The text "feature" was used directly in its original form. 2) AraBERT Preprocessing: this method made use of the AraBERT preprocessing function. Key steps involved removing Arabic diacritic marks, stripping elongation characters and adding white spaces. Additionally, Hindi numerals were converted into their Arabic equivalents. 3) Link and Hashtag Removal: Building on the AraBERT prepossessing setting, this configuration further involved cleansing the text of "LINK" and "#" references. In Subsection 4.2, we detail the specific preprocessing configurations employed for our models across the various sub-tasks.

4.2 Our Approach

Our primary objective was to evaluate the efficacy of BERT-based models for persuasion and disinformation detection tasks. In this endeavor, we mainly examined AraBERT (Antoun et al.) [2]. AraBERT is an Arabic pretrained language model based on Google’s BERT architecture (Devlin et al., 2018) with the BERT-Base configuration. The training dataset for AraBERT was curated from a myriad of sources, including OSCAR (Abadji et al., 2022), Arabic Wikipedia dump [3], and the 1.5B words Arabic Corpus (El-Khair, 2016) among others.

Beyond AraBERT, we experimented with models such as mBERT and XLM-RoBERTa (Conneau

[2] https://github.com/aub-mind/arabert/tree/master#AraBERT

[3] https://archive.org/details/arwiki-20190201

Sub-Task	Split	# Data Points	# NaN Data Points	Per Class Data Points
1A	Train	2427	0	'true': 1918, 'false': 509
1A	Dev	259	0	'true': 202, 'false': 57
1A	Test	503	0	'true': 331, 'false': 172
2A	Train	14147	21	'no-disinfo': 11491, 'disinfo': 2656
2A	Dev	2115	4	'no-disinfo': 1718, 'disinfo': 397
2A	Test	3729	0	'no-disinfo': 2853, 'disinfo': 876
2B	Train	2656	8	'HS': 1512, 'OFF': 500, 'SPAM': 453, 'Rumor': 191
2B	Dev	397	1	'HS': 226, 'OFF': 75, 'SPAM': 68, 'Rumor': 28
2B	Test	876	0	'HS': 442, 'SPAM': 241, 'OFF': 160, 'Rumor': 33

Table 1: Statistics of the data regarding all subtasks. Note that the number of data-points for sub-task 1B were the same as 1A but the dataset has too many classes to include here.

et al., 2019), with the aim of discerning the impact of multilingual data on our tasks. However, during the development phase, AraBERT consistently surpassed the performance of these multilingual models, likely due to its training on a substantial Arabic corpus. This observation aligns with studies like that of Alammary, 2022, emphasizing the efficacy of monolingual models in specific contexts. As a result, we opted for AraBERT.

Our secondary objective was to assess the performance of the model introduced by Jiang et al., 2023, who leveraged lossless compressors and the k-nearest-neighbor (kNN) algorithm for classification tasks. Their method is founded on the principle that lossless compressors (e.g., gzip, z2, lzma, and zstandard) are adept at representing regularities in data and that textual data within the same category share more similarities and regularities than those from distinct categories. By measuring the Normalized Compression Distance (NCD) between texts, this method capitalizes on the compression lengths to approximate the Kolmogorov complexity of data. This subsequently serves as the foundation for a distance metric used in kNN classification.

Substantial controversy has surfaced within the online research community concerning the work of Jiang et al., 2023. Prominent among these critiques are those from Sebastian Raschka[4] and Ken Schutte[5]. Both researchers highlighted potential discrepancies in the original paper's code and implementation. Specifically, they pinpointed an error in the kNN accuracy computation resulting from a flawed tie-breaking strategy, which may have inflated the reported results. Despite the critiques, Jiang et al.'s methodology offers a compelling approach to text classification. Seizing the opportunity presented by this shared task, we undertake an evaluation of the method through an independent implementation of the compressor-based classifier, adopting a different tie-breaking strategy for the kNN classifier. Our aim is to assess its performance on Arabic persuasion technique detection and disinformation detection tasks, and subsequently, to share these insights transparently with the research community.

4.3 Evaluation

In this section, we describe the results we achieved including the experimental setup.

4.3.1 Experimental Setup

In our experiments, we employed an updated version of AraBERT, which was trained on a substantially larger dataset, thus incorporating an expanded lexicon. The authors of the original model pinpointed a flaw in AraBERTv1's wordpiece vocabulary. The issue came from punctuation and numbers that were still attached to words when they trained the wordpiece vocab. They have since rectified this by introducing spaces around numerical digits and punctuation marks. To make sure this is compatible with any new downstream task, they have released a preprocessing function as well, that we apply in all our tasks before fine-tuning.

For the models, we chose to experiment with the three different versions of the base model. The first two models are trained on the same dataset but one (v2) uses pre-segmentation and the other (v02) does not. The last model (v02-Twitter) is trained on the combination of the same dataset plus 60M multi-dialect tweets from twitter as well. For all tasks, only the text data was used as a feature for training. To address the issue of class imbalance, class weights were computed and used during the training process. In addition to this, we experimented with removing hashtags and links, to see

[4] https://magazine.sebastianraschka.com/p/large-language-models-and-nearest

[5] https://kenschutte.com/gzip-knn-paper/

	Preprocessing		Task 1A		Task 1B		Task 2A		Task 2B	
Model	**+**	**++**	**Dev**	**Test**	**Dev**	**Test**	**Dev**	**Test**	**Dev**	**Test**
AraBERT-v0.2	Yes	No	0.849	**0.755**	0.548	0.471	0.900	**0.904**	0.836	**0.828**
AraBERT-v2	Yes	No	0.861	0.748	0.598	0.550*	0.901	0.902	0.823	0.816
AraBERT-Twitter	Yes	No	0.868	0.747	0.538	0.481	0.909	0.898	**0.843**	0.817
AraBERT-v0.2	Yes	Yes	0.780	0.658	0.605	0.577	0.820	0.786	0.631	0.663
AraBERT-v2	Yes	Yes	**0.868**	0.749*	**0.606**	0.537	0.812	0.765	0.646	0.683
AraBERT-Twitter	Yes	Yes	0.779	0.658	0.601	**0.570**	**0.912**	0.898*	0.841	0.814*
gzip+knn (lowest-label-index)	No	No	0.803	0.658	0.499	0.393	0.800	0.772	0.687	0.713
gzip+knn (closer-neighbor)	No	No	0.745	0.636	0.489	0.345	0.830	0.801	0.664	0.681
gzip+knn (random-selection)	No	No	0.752	0.616	0.455	0.326	0.818	0.798	0.636	0.688
gzip+knn (k=3)	No	No	0.764	0.654	0.471	0.334	0.848	0.825	0.634	0.687
Majority baseline				0.658		0.360		0.765		0.505

Table 2: Results of AraBERT experiments on all Sub-Tasks. $+$ and $++$ denotes preprocessing using AraBERT preprocessor and removal of hashtags and LINKs respectively. Note that the results in bold are the models that performed the best but the models with $*$ are the ones that were submitted which may not align with the best score as some of the experiments were carried out after the deadline.

if they have a positive effect on the performance as well. All the models are trained for 10 epochs with the model performing best on validation set chosen for test evaluation. The learning rate and batch size was set to $2e-5$ and 16 respectively, with the model evaluated on the dev set after every epoch.

Alongside our submissions for the shared task using AraBERT pretrained models, we applied Jiang et al., 2023 approach to this specific shared task context. We utilized the gzip compressor for encoding the text data and calculated inter-textual distances using the Normalized Compression Distance (NCD). The k-Nearest Neighbors (kNN) classifier was employed with $k = 2$, mirroring the setup in Jiang et al. 2023's study.

For the kNN's tie-breaking mechanism, we evaluated three strategies: 1) Lowest-label-index: this method, which follows the convention employed in the original study, selects the label with the lowest index during a tie. 2) Random-selection: in instances of ties, this strategy randomly selects among the tied labels. 3) Closer-neighbor: this method gives preference to the label of the nearest tied neighbor. Furthermore, we conducted experiments using $k = 3$ for the kNN classifier, where tie-breaking mechanisms are inherently unnecessary due to the odd number of neighbors. In all subtasks, we opted for no preprocessing of the data, as our preliminary experiments revealed that preprocessing adversely affected the performance of the compression-based approach.

4.3.2 Results

Tasks 1A and 1B: Persuasion Technique Detection: Our submission with the AraBERT-v2 model recorded a Test score of 0.749 and 0.550, achieving 5th and 4th position in the leaderboard for the Task 1A and 1B respectively. Parallel to our primary experiments, our exploration into the methodology of Jiang et al., 2023 bore intriguing results. The compressor-based approach with the "lowest-label-index" tie-breaking strategy for the kNN classifier achieved a Test score of 0.658 in Task 1A, closely mirroring the majority baseline of 0.658. For Task 1B, the strategy performed above the baseline, achieving a score of 0.393 compared to the baseline of 0.360. It's noteworthy to mention that while the AraBERT models capitalized on their training over an expansive Arabic corpus, the gzip+knn approach showcased potential, particularly when considering its resource-efficient nature.

Tasks 2A and 2B: Disinformation Detection: The AraBERT-v02-Twitter displayed good performance, with Test scores of 0.898 and 0.814 for Tasks 2A and 2B respectively, achieving 8th and 7th position on the leaderboard. Meanwhile, the compressor-based classifier showed its merits once again. Using the "closer-neighbor" tie-breaking strategy, the gzip+knn approach produced a Test score of 0.801 for Task 2A, not far from the baseline of 0.765. In Task 2B, the "lowest-label-index" strategy yielded a score of 0.713, surpassing the baseline of 0.505.

5 Conclusions

In our participation in the ArAIEval Shared Task, we predominantly employed Transformer-based models for persuasion techniques and disinformation detection tasks, given their demonstrated proficiency with Arabic textual data. Although our results highlighted the strengths of these models, we simultaneously recognized the emerging potential of the compression-based approach to text classification. While these compression-based methods are in their infancy, they offer exciting opportunities for continued research. Future studies should delve deeper into the applicability of lossless compressors for text classification and seek to identify non-parametric machine learning algorithms that best align with these compressors. Importantly, compressor-driven systems might be more advantageous in situations where resource efficiency and rapid processing take precedence over accuracy.

6 Limitations

While these experiments give us a promising avenue to explore in terms of detecting persuasion techniques and disinformation in Arabic text, even in a low-resource setting using compressors, there are a lot of limitations to these approaches. One thing to note is that although Pretrained LMs seem to recognize disinformation in these texts, there is no reliability to this score, as in order to fact-check any news, you need consolidating evidence to see if it is fake or not, rather than only looking at how it is worded. It can be argued that those instances in which the wording of a fake piece of news is indistinguishable from a truthful one are even more dangerous. To tackle this, ways to include other sources of data would help improve results.

References

Julien Abadji, Pedro Ortiz Suarez, Laurent Romary, and Benoît Sagot. 2022. Towards a Cleaner Document-Oriented Multilingual Crawled Corpus. *arXiv e-prints*, page arXiv:2201.06642.

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, 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, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Ali Saleh Alammary. 2022. Bert models for arabic text classification: a systematic review. *Applied Sciences*, 12(11):5720.

Shatha Alyoubi, Manal Kalkatawi, and Felwa A. Abukhodair. 2023. The detection of fake news in arabic tweets using deep learning. *Applied Sciences*.

Wissam Antoun, Fady Baly, Rim Achour, A. Hussein, and Hazem M. Hajj. 2020. State of the art models for fake news detection tasks. *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*, pages 519–524.

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

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

Giovanni Da San Martino, Yu Seunghak, Alberto Barrón-Cedeno, Rostislav Petrov, Preslav Nakov, et al. 2019. Fine-grained analysis of propaganda in news article. 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 5636–5646. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ibrahim Abu El-Khair. 2016. 1.5 billion words arabic corpus. *ArXiv*, abs/1611.04033.

Ivan Habernal, Raffael Hannemann, Christian Pollak, Christopher Klamm, Patrick Pauli, and Iryna Gurevych. 2017. Argotario: Computational argumentation meets serious games. *arXiv preprint arXiv:1707.06002*.

Maram Hasanain, Firoj Alam, Hamdy Mubarak, Samir Abdaljalil, Wajdi Zaghouani, Preslav Nakov, Giovanni Da San Martino, and Abed Alhakim Freihat. 2023. ArAIEval Shared Task: Persuasion Techniques and Disinformation Detection in Arabic Text. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore. Association for Computational Linguistics.

Takeshi Iida, Jaehyun Song, José Luis Estrada, and Yuriko Takahashi. 2022. Fake news and its electoral consequences: a survey experiment on mexico. *AI & SOCIETY*.

Madinabonu Ishmuradova. 2019. Strategies for using fake news as a tool to manipulate public opinion.

Zhiying Jiang, Matthew Yang, Mikhail Tsirlin, Raphael Tang, Yiqin Dai, and Jimmy Lin. 2023. "low-resource" text classification: A parameter-free classification method with compressors. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6810–6828, Toronto, Canada. Association for Computational Linguistics.

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

Ahmed Redha Mahlous and Ali Al-laith. 2021. Fake news detection in arabic tweets during the covid-19 pandemic. *International Journal of Advanced Computer Science and Applications*, 12.

Ali Bou Nassif, Ashraf Elnagar, Omar A. Elgendy, and Yaman Afadar. 2022. Arabic fake news detection based on deep contextualized embedding models. *Neural Computing & Applications*, 34:16019 – 16032.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Ali Safaya, Moutasem Abdullatif, and Deniz Yuret. 2020. KUISAIL at SemEval-2020 task 12: BERT-CNN for offensive speech identification in social media. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 2054–2059, Barcelona (online). International Committee for Computational Linguistics.

ReDASPersuasion at ArAIEval Shared Task: Multilingual and Monolingual Models For Arabic Persuasion Detection

Fatima Zahra Qachfar
fqachfar@uh.edu
University of Houston
Houston, TX, USA

Rakesh M. Verma
rmverma2@central.uh.edu
University of Houston
Houston, TX, USA

Abstract

To enhance persuasion detection, we investigate the use of multilingual systems on Arabic data by conducting a total of 22 experiments using baselines, multilingual, and monolingual language transformers. Our aim is to provide a comprehensive evaluation of the various systems employed throughout this task, with the ultimate goal of comparing their performance and identifying the most effective approach. Our empirical analysis shows that *ReDASPersuasion* system performs best when combined with multilingual "XLM-RoBERTa" and monolingual pre-trained transformers on Arabic dialects like "CAMeLBERT-DA SA" depending on the NLP classification task.

1 Introduction

In recent years, the detection of persuasion techniques in text has gained a significant attention in research. Persuasion techniques can be used for either positive or negative ends. On one hand, persuasion can be used to convince people to support noble causes, promote social justice, and bring about positive change. However, these same techniques can also be exploited by individuals with ill intentions to manipulate and deceive others for personal gain or to perpetuate harmful beliefs and behaviors. Moreover, persuasion techniques can be employed with malicious intent including : 1. phishing scams, 2. propagandistic content (Barrón-Cedeño et al., 2019), 3. fallacy argumentation (Habernal et al., 2017, 2018), and 4. coercive extortion tactics.

With the increasing use of Arabic language in various forms of media, including social media, and news articles, it has become crucial to develop effective methods for identifying persuasive strategies in Arabic text. This task is challenging due to the complexities of the Arabic language, which includes various dialects, nuances, and cultural references (Glenn et al., 1977) that can affect the interpretation of persuasive elements. Researchers have employed various approaches, such as rhetoric methods (Koch, 1983), and deep learning models (Brahem et al., 2022), to automatically detect propaganda and persuasion in Arabic text. These techniques aim to identify specific linguistic features, such as sentiment analysis, and lexical semantics commonly used in persuasion.

Through this task, we have realized the importance of taking into consideration the different Arabic nuances and dialects in Multi-label and binary classification tasks. We also observe that Arabic writing styles vary immensely depending on the type of the data (paragraphs vs tweets) where *paragraphs* mainly use Modern Standard Arabic (MSA) or Classic Arabic (CA) while *tweets* contain a diversity of Arabic dialects using code-switching with other foreign languages and emojis.

We begin by providing an overview of the ArAiEval Shared-Task in Section 2. Next, we present a detailed description of the various systems utilized in our empirical study. We then delve into the preprocessing methods employed in Section 4, before presenting the results in Section 5. An error analysis is provided in Section 6, followed by a discussion section offering insights and perspectives on the task at hand. Finally, we summarize our findings and outline potential avenues for future research in Section 8.

2 Dataset and Tasks

Hasanain et al. (2023) organized the ArAIEval 2023 Shared-Task which includes two tasks in the Arabic language. The first task introduces *persuasion technique detection* while the second task introduces *disinformation detection* (Mubarak et al., 2023). Previously, Alam et al. (2022) described 20 propaganda techniques in the WANLP 2022 Shared Task adopting the same techniques as (Da San Martino et al., 2019) to Arabic news articles.

In this paper, we solely focus on the first task to investigate existing systems and enhance its per-

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 549–557
December 7, 2023 ©2023 Association for Computational Linguistics

formance. For the persuasion technique detection task, the organizers offered two subtasks : i) *Task 1A*, and ii) *Task 1B*. We will describe these two subtasks in the following sections.

2.1 Task 1A : Binary Classification

This task involves classifying instances as either "true" or "false", where "true" indicates the presence of persuasion techniques in a given text, and "false" implies their absence. We report the class distribution in each subset (training, dev, and test) in Table 4 in the supplementary material.

2.2 Task 1B: Multi-Label Classification

The task involves assigning one or more labels from a predefined set, representing 23 types of persuasion techniques used in propaganda. Similarly, Piskorski et al. (2023) provided shared-task on a multilingual setting for multi-label classification. They have mapped these 23 techniques to six major categories (1. *Justification*, and 2. *Simplification*, and 3. *Distraction*, and 4. *Call*, and 5. *Manipulative wording*, and 6. *Attack on reputation*.). This is a multi-label classification problem where multiple propaganda techniques might be present in the same example. Samples with no persuasion technique are labeled with "no technique". We also report all the persuasion techniques in Table 5 in Appendix A.1.1.

3 Systems

We will describe thoroughly the different systems we used during this task for an end of comparing their performance and finding the best system.

3.1 Baseline Algorithms

For the baseline models, we implement a pipeline object that extracts the TF-IDF features and vectorizes textual content using unigram and bigram count vectorizer. We choose four traditional baseline algorithms (LR (Wright, 1995), RF (Breiman, 2001), XGB (Chen et al., 2015), and SVM(Cortes and Vapnik, 1995)).

We have defined a search space of hyperparameters using distributed hyperparameter optimization package "HyperOpt" [1] (Bergstra et al., 2013) with 5 trials 2 cross-validation splits. The baseline hyperparameter tuning include parameters like regularization strength (C), number of maximum iteration (*max_iter*), and the number of estimators (*n_estimators*). The best estimator is used to predict the testing set. We include the best hyperparameters in Appendix A.3

3.2 ReDASPersuasion System

Qachfar and Verma (2023) present a multilingual system for persuasion detection on a total of five languages (En, Fr, Ru, It, Po, Ge). This system leverages the power of multilingual transformers "XLM-RoBERTa" and language agnostic features to perform persuasion detection across multiple languages.

The initial structure of the *ReDASPersuasion* system is composed of three main components:

- A multilingual transformer model that can process input in various languages.
- A feature engineering module that extracts language-agnostic features suitable for cross-lingual classification of persuasion.
- A multi-label classification module that combines the transformer output with persuasion features using a dropout layer, a dense linear layer, and a sigmoid activation function to produce multiple classification labels.

For task 1A, We modify this system to perform a binary classification task by using the sigmoid activation function with one output node while in task 1B, we used the sigmoid activation function with one node per persuasion class (23 techniques). We also change the criterion loss function from "*BCEWithLogitsLoss*" for Multilabel classification in Task 1B to "*CrossEntropy*" loss for binary classification in Task 1A.

To prevent the model from predicting a combination of "no technique" and other techniques, we treat samples with the "no technique" label as having no label at all.

4 Preprocessing

As illustrated in Figure 1, ArAiEval's first task persuasion dataset contains two data types:

1) ***Paragraph*:** a passage from news articles written in Modern Standard Arabic (MSA) which does not include code-switching or any specific keywords unlike tweets.
2) ***Tweet*:** a social media message written in diverse Arabic dialects mixed from different regions containing code-switching, specific Twitter keywords and emojis.

[1] https://github.com/hyperopt/hyperopt

Arabic Tweet		RT @USER My message to Y'all!! 😂#كورونا #صباح_الخير LINK
Preprocessing Steps	*KT*	[إعادة_التغريد] [مستخدم] #صباح_الخير #كورونا😂 My message to Y'all!! [موقع_الكتروني]
	CS	[إعادة_التغريد] [مستخدم] رسالتي لكم جميعا!!😂 #صباح_الخير #كورونا [موقع_الكتروني]
	EC	[إعادة_التغريد] [مستخدم] رسالتي لكم جميعا!! :وجه_بدموع_فرح: #صباح_الخير #كورونا [موقع_الكتروني]

***KT** :Keyword Translation.* ***CS** : Code Switching.* ***EC** : Emoji Conversion.*

Table 1: Preprocessing Techniques Applied to Arabic Text

We describe three preprocessing techniques we applied to the tweets to translate code-switched text to Arabic. An example of these techniques are shown in Table 1.

4.1 Keyword Translation (KT)

In the "tweet" data type, we have certain keywords like retweet (RT), username (@USER), and website (LINK). We replace these terms with Arabic words using regular expressions, maintaining the proper right-to-left alignment of Arabic words.

4.2 Code Switching (CS)

Arabic tweeters may use code-switching to express themselves more effectively, or to communicate with a diverse audience. For example, a user may start a tweet in Arabic, switch to English in the middle, and then finish it off in French. For each tweet, we automatically detect code-switching fragments using "*Lingua*" [2] Python package, and we translate it to Arabic using Google's translation API.

4.3 Emoji Conversion (EC)

In tweets, emojis are typically used to convey emotions or ideas. Mubarak et al. (2022) showed the importance of emojis in the detection of Arabic offensive language and hateful speech.

Instead of removing all emojis from tweets like (Bennessir et al., 2022), we choose to convert them to Arabic descriptive text since emojis might hold meaning in the context of a short deceptive tweet representing positive or negative sentiment. For this we add Arabic language support to the "*emoji*" [3] Python package using normalized representations from the latest release of Unicode Common Locale Data Repository (CLDR) [4] to avoid broken Unicode. We create a dictionary of Arabic emoji representations based on the *emojiterra* website.[5]

5 Experimental Results

We ran all classification experiments on a high performing cluster machine with an Intel® Xeon® Gold 6252 (3.70GHz) processor with 24 cores and 48 threads running Linux Red Hat Enterprise Server 8.6 with Nvidia® Volta V100 GPUs.

For task 1A, the initial structure of the *ReDASPersuasion* system with "XLM-RoBERTa" (Conneau et al., 2020) achieves the best F1-Micro score of 0.7336 on the test set.

For task 1B, the *ReDASPersuasion* system with "CAMeLBERT-DA SA" (Inoue et al., 2021) fine-tuned on sentiment analysis for Dialect Arabic (DA) achieves the best performance on the testing set with a F1-Micro score of 0.5584.

According to Table 3, the combination of *ReDASPersuasion* and "XLM-RoBERTa" yields the highest F1-score macro weighted strategy, with a value of 0.1449, for task 1B.

Our investigation reveals that during the development process, the *ReDASPersuasion* system powered by "XLM-RoBERTa" shows the most promise, with the *ReDASPersuasion* system using monolingual "CAMeLBERT-DA SA" coming in a close second. However, when it comes to the testing phase, one method excels in the first task, while the other method excels in the second task, as evidenced by their respective F1-Micro scores.

In Table 3 Task 1A, logistic regression, majority class baseline, and *ReDASPersuasion* system with "DistilBERT" all achieve an F1-score of 0.6581 which means these models fail to accurately predict the test set, as they simply assign the majority class "true" to all samples.

Due to the lack of visibility during the testing phase evaluation, we accidentally submitted wrong prediction results from the *ReDASPersuasion* system using "DistilBERT" instead of the intended top-performing *ReDASPersuasion* system using "XLM-RoBERTa". We have also encountered technical difficulties on the ArAIEval competition's hosting platform, *CodaLab*, mostly stemming from their HTTP backend server.

[2] https://github.com/pemistahl/lingua

[3] https://github.com/carpedm20/emoji/

[4] https://github.com/unicode-org/cldr/raw/release-43/common/annotations/ar.xml

[5] https://emojiterra.com/copypaste/ar/

	Models	Task 1A Evaluation		Task 1B Evaluation	
		F1-score (Micro)	F1-score (Macro)	F1-score (Micro)	F1-score (Macro)
Baselines	LR (Wright, 1995)	0.7799	0.4382	0.4701	0.0393
	RF (Breiman, 2001)	0.7761	0.4687	0.4647	0.0582
	XGB (Chen et al., 2015)	0.7452	0.5971	0.4417	0.0951
	Linear SVM (Cortes and Vapnik, 1995)	0.7954	0.5076	0.5178	0.0699
	Random-Guess Baseline	0.5405	0.4774	0.0938	0.0573
	Majority-Class Baseline	0.7799	0.4382	0.4595	0.0337
ReDASPersuasion with Multilingual Transformers	mBERT (Devlin et al., 2019)	0.8263	0.6639	0.5922	0.1453
	DistilBERT (Sanh et al., 2020)	0.7992	0.6306	0.5658	0.1295
	XLM RoBERTa (Conneau et al., 2020)	**0.8764**	**0.8017**	**0.6454**	**0.1884**
ReDASPersuasion with Monolingual Transformers	AraBERT (Antoun et al.)	0.8340	0.7597	0.6064	0.1792
	MarBERT (Abdul-Mageed et al., 2021)	0.8224	0.7059	0.6249	0.1194
	CAMeLBERT-DA SA (Inoue et al., 2021)	0.7954	0.7001	0.6048	0.1594
	CAMeLBERT-MIX SA (Inoue et al., 2021)	0.8340	0.7030	0.6215	0.1813

Table 2: Evaluation Results on the Development Set using ArAiEval Scorer

	Models	Task 1A Evaluation		Task 1B Evaluation	
		F1-score (Micro)	F1-score (Macro)	F1-score (Micro)	F1-score (Macro)
Baselines	LR (Wright, 1995)	0.6581	0.3969	0.3629	0.0302
	RF (Breiman, 2001)	0.6600	0.4190	0.3585	0.0378
	XGB (Chen et al., 2015)	0.6640	0.5732	0.3275	0.0688
	Linear SVM (Cortes and Vapnik, 1995)	0.6600	0.4031	0.3760	0.0412
	Random-Guess Baseline	0.4771	0.4598	0.0868	0.0584
	Majority-Class Baseline	0.6581	0.3969	0.3599	0.0279
ReDASPersuasion with Multilingual Transformers	mBERT (Devlin et al., 2019)	0.6899	0.5656	0.4923	0.1083
	DistilBERT (Sanh et al., 2020)	0.6581	0.3969	0.4523	0.0568
	XLM RoBERTa (Conneau et al., 2020)	**0.7336**	0.6684	0.5555	**0.1449**
ReDASPersuasion with Monolingual Transformers	AraBERT (Antoun et al.)	0.7117	0.6967	0.5154	0.1344
	MarBERT (Abdul-Mageed et al., 2021)	0.7197	0.6826	0.5549	0.0988
	CAMeLBERT-DA SA (Inoue et al., 2021)	0.7217	**0.7007**	**0.5584**	0.1313
	CAMeLBERT-MIX SA (Inoue et al., 2021)	0.7217	0.6712	0.5565	0.1372

Table 3: Evaluation Results on the Testing Set using ArAiEval Scorer

6 Error Analysis

Another limitation we faced in this task is the imbalanced nature of the data and the small number of examples in certain persuasion techniques. For example, "*Appeal to Popularity*" persuasive technique occurs only twice in the training set and once in both the dev and test set as described in Table 5. Thus, the system was unable to accurately predict any of the labels for that particular class resulted in a zero F1 score, which had a negative impact on the overall performance in multi-label classification.

To provide a more detailed examination of the prediction errors, we present the confusion matrices for the top-performing models on both tasks in Figure 3, and Figure 4 in the supplementary material. As shown in Figure 4, "Name_Calling-Labeling" and "Loaded_Language" had the highest accuracy rates, whereas all other persuasive technique were inaccurately predicted. This can be attributed to the limited quantity of training data available for these categories.

7 Discussion

As shown in Figure 2, our Arabic dialect identification process reveals that the ArAiEval dataset encompasses a diverse array of dialects, with the most prominent languages being Saudi Arabian, Egyptian, and Palestinian dialects.

Different dialects have different vocabularies, and certain words or phrases might be interpreted differently in another dialect or deemed offensive. A concrete example would be the word "العافية" in Egyptian dialect means "health" while the same word means "fire" in Moroccan dialect. These differences in Arabic dialects can significantly impact persuasion strategies. An interesting take would be to consider the unique features of each dialect.

8 Conclusion

We described our systems for the two subtasks of the ArAiEval 2023 shared task on persuasion detection in Arabic to detect a total of 23 persuasion techniques for multi-label classification. We experiment with different combinations of multilingual and monolingual transformers. We have proven that the *ReDASPersuasion* model can benefit from both the multilingual "XLM-RoBERTa" transformer and the monolingual Dialect Arabic "CAMelBERT-DA SA" model depending on the NLP task. This task was an opportunity to evaluate the *ReDASPersuasion* model in depth and to conduct an error analysis to enhance our persuasion detection model for future works.

Limitations

Each dialect has its own unique features, such as vocabulary, grammar, and pronunciation, which can impact the way messages are conveyed and received by audiences. Therefore, one of the shortcomings of the *ReDASPersuasion* system is to detect persuasive words in the various Arabic dialects in classification.

Because of time constraints, we were unable to apply training data augmentation; however, we have translated the SemEval 2023 shared task (Piskorski et al., 2023) into Arabic text, which we will add to the imbalanced training dataset in future work to further analyze the behavior of the *ReDASPersuasion* system.

Ethics Statement

The dataset used in this paper, provided by the organizers (Hasanain et al., 2023), already adheres to data privacy regulations by eliminating all usernames and links from the tweets.

To ensure reproducible and ethical research, we provide models' hyperparameters used to achieve our experiments. Our work complies with the ACL Ethics Policy.

Acknowledgements

We would like to thank ArAiEval 2023 shared task organizers for their efforts in preparing Arabic datasets for propaganda detection including persuasive and disinformative cues.

We also thank the reviewers for their constructive feedback. This research was partly supported by NSF grants 1433817, 1950297, and 2210198, ARO grants W911NF-20-1-0254 and W911NF-23-1-0191, and ONR award N00014-19-S-F009. Verma is the founder of Everest Cyber Security and Analytics, Inc.

References

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2021. ARBERT & MARBERT: Deep bidirectional transformers for Arabic. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7088–7105, Online. Association for Computational Linguistics.

Firoj Alam, 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, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

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

Alberto Barrón-Cedeño, Israa Jaradat, Giovanni Da San Martino, and Preslav Nakov. 2019. Proppy: Organizing the news based on their propagandistic content. *Information Processing & Management*, 56(5):1849–1864.

Mohamed Aziz Bennessir, Malek Rhouma, Hatem Haddad, and Chayma Fourati. 2022. icompass at arabic hate speech 2022: Detect hate speech using qrnn and transformers. In *Proceedinsg of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools with Shared Tasks on Qur'an QA and Fine-Grained Hate Speech Detection*, pages 176–180.

James Bergstra, Daniel Yamins, and David Cox. 2013. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *International conference on machine learning*, pages 115–123. PMLR.

Houda Bouamor, Nizar Habash, Mohammad Salameh, Wajdi Zaghouani, Owen Rambow, Dana Abdulrahim, Ossama Obeid, Salam Khalifa, Fadhl Eryani, Alexander Erdmann, et al. 2018. The madar arabic dialect corpus and lexicon. In *Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018).*

Bechir Brahem, Hatem Haddad, et al. 2022. icompass at wanlp 2022 shared task: Arbert and marbert for multilabel propaganda classification of arabic tweets. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 511–514.

Leo Breiman. 2001. Random forests. *Machine learning*, 45:5–32.

Tianqi Chen, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang, Hyunsu Cho, Kailong Chen, Rory Mitchell, Ignacio Cano, Tianyi Zhou, et al. 2015. Xgboost: extreme gradient boosting. *R package version 0.4-2*, 1(4):1–4.

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.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning*, 20:273–297.

Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeño, Rostislav Petrov, and Preslav Nakov. 2019. Fine-grained analysis of propaganda in news article. 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 5636–5646, Hong Kong, China. 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.

E.S. Glenn, D. Witmeyer, and K.A. Stevenson. 1977. Cultural styles of persuasion. *International Journal of Intercultural Relations*, 1(3):52–66.

Ivan Habernal, Raffael Hannemann, Christian Pollak, Christopher Klamm, Patrick Pauli, and Iryna Gurevych. 2017. Argotario: Computational argumentation meets serious games. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 7–12, Copenhagen, Denmark. Association for Computational Linguistics.

Ivan Habernal, Patrick Pauli, and Iryna Gurevych. 2018. Adapting serious game for fallacious argumentation to german: Pitfalls, insights, and best practices. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, page in press, Miyazaki, Japan. European Language Resources Association (ELRA).

Maram Hasanain, Firoj Alam, Hamdy Mubarak, Samir Abdaljalil, Wajdi Zaghouani, Preslav Nakov, Giovanni Da San Martino, and Abed Alhakim Freihat. 2023. ArAIEval Shared Task: Persuasion Techniques and Disinformation Detection in Arabic Text. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore. Association for Computational Linguistics.

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.

Barbara Johnstone Koch. 1983. Presentation as proof: The language of arabic rhetoric. *Anthropological linguistics*, pages 47–60.

Hamdy Mubarak, Samir Abdaljalil, Azza Nassar, and Firoj Alam. 2023. Detecting and reasoning of deleted tweets before they are posted.

Hamdy Mubarak, Sabit Hassan, and Shammur Absar Chowdhury. 2022. Emojis as anchors to detect arabic offensive language and hate speech.

Jakub Piskorski, Nicolas Stefanovitch, Giovanni Da San Martino, and Preslav Nakov. 2023. Semeval-2023 task 3: Detecting the category, the framing, and the persuasion techniques in online news in a multi-lingual setup. In *Proceedings of the the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2343–2361.

Fatima Zahra Qachfar and Rakesh Verma. 2023. ReDASPersuasion at SemEval-2023 task 3: Persuasion detection using multilingual transformers and language agnostic features. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2124–2132, Toronto, Canada. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Raymond E Wright. 1995. Logistic regression.

A Appendix

A.1 Data Distribution

We will describe the class and type distribution of the different subsets provided in Task 1 for persuasion detection.

A.1.1 Class Distribution

We observe that the binary class distribution is imbalanced of an approximately 1 to 4 True to False ratio.

Binary Class Distribution			
	Train	Dev	Test
True	1919	202	331
False	509	57	172

Table 4: Binary Labels in Arabic Task 1A

A.1.2 Data Type Distribution

As illustrated in Figure 1, most samples are categorized as paragraph data type, accounting for over 65% of the total samples in each subset. This introduces new challenges to the classification task where the structure of tweets and article paragraph news differ substantially.

Total Number of Persuasion Techniques			
	Train	Dev	Test
Appeal to Authority	48	5	14
Appeal to Fear Prejudice	108	12	15
Appeal to Hypocrisy	56	7	17
Appeal to Popularity	2	1	1
Appeal to Time	10	2	2
Appeal to Values	37	4	29
Causal Oversimplification	128	15	12
Consequential Oversimplification	33	3	3
Conversation Killer	28	3	7
Doubt	143	16	21
Exaggeration Minimisation	292	33	40
False Dilemma No choice	32	3	6
Flag Waving	63	7	25
Guilt by Association	13	1	1
Loaded Language	1574	176	253
Name Calling Labeling	692	77	133
Obfuscation Vagueness Confusion	240	28	25
Questioning Reputation	383	43	89
Red Herring	8	1	3
Repetition	25	3	6
Slogans	70	8	25
Straw Man	6	1	2
Whataboutism	9	1	2

Table 5: Persuasion Techniques in Arabic Task 1B

A.2 Dialect Language Identification

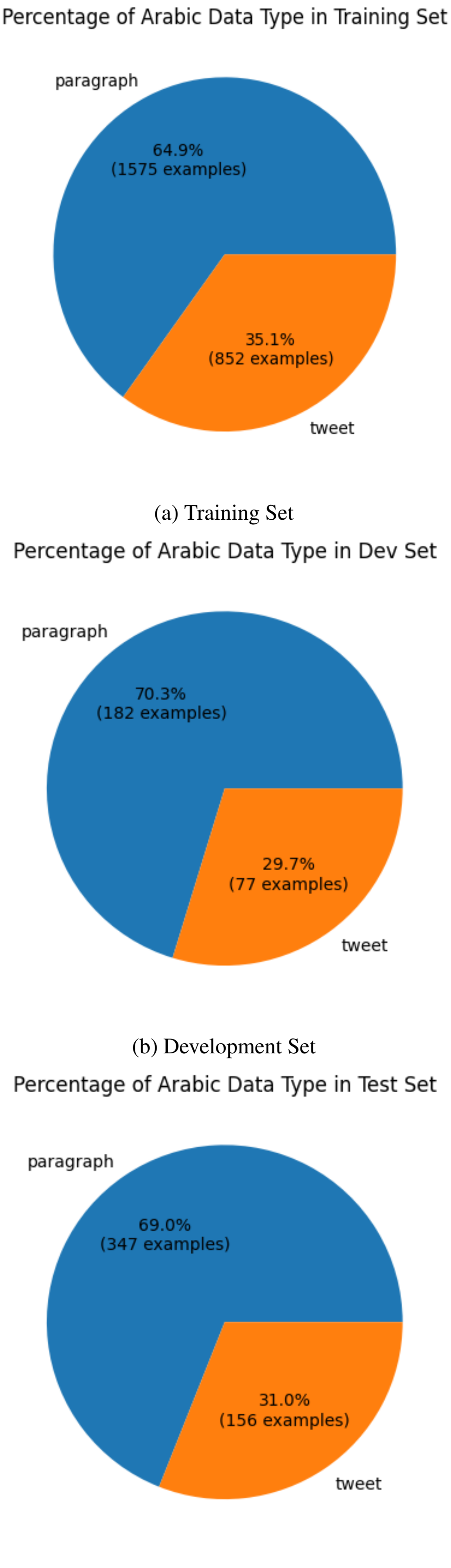
For Arabic dialect language detection, we used the bert-base-arabic model provided by CAMel (Computational Approaches to Modeling Language) Laboratory on the HuggingFace Hub [6] trained on MADAR (Bouamor et al., 2018) Twitter dataset which contains Arabic dialect tweets originating from 25 regions. In Figure 2, we observe that the top five Arabic dialects originate from Saudi Arabia, Egypt, Kuwait, Palestine, and Jordan throughout the training, dev, and test sets.

These different Arabic dialects from different regions have distinct grammatical structures, vocabularies, and idiomatic expressions that can be challenging to reconcile within one classification model. In this manner, we fine-tune the CAMELBERT-MIX SA model (Inoue et al., 2021) on our tasks which shows significant performance in predicting persuasive writing in Arabic text.

A.3 Model Hyperparameters

In this section, we report in Tables 6 , and 7 all the hyperparameters used in optimization for transformer and baseline models respectively.

[6] https://huggingface.co/CAMeL-Lab/bert-base-arabic-camelbert-msa-did-madar-twitter5


Percentage of Arabic Data Type in Training Set

(a) Training Set

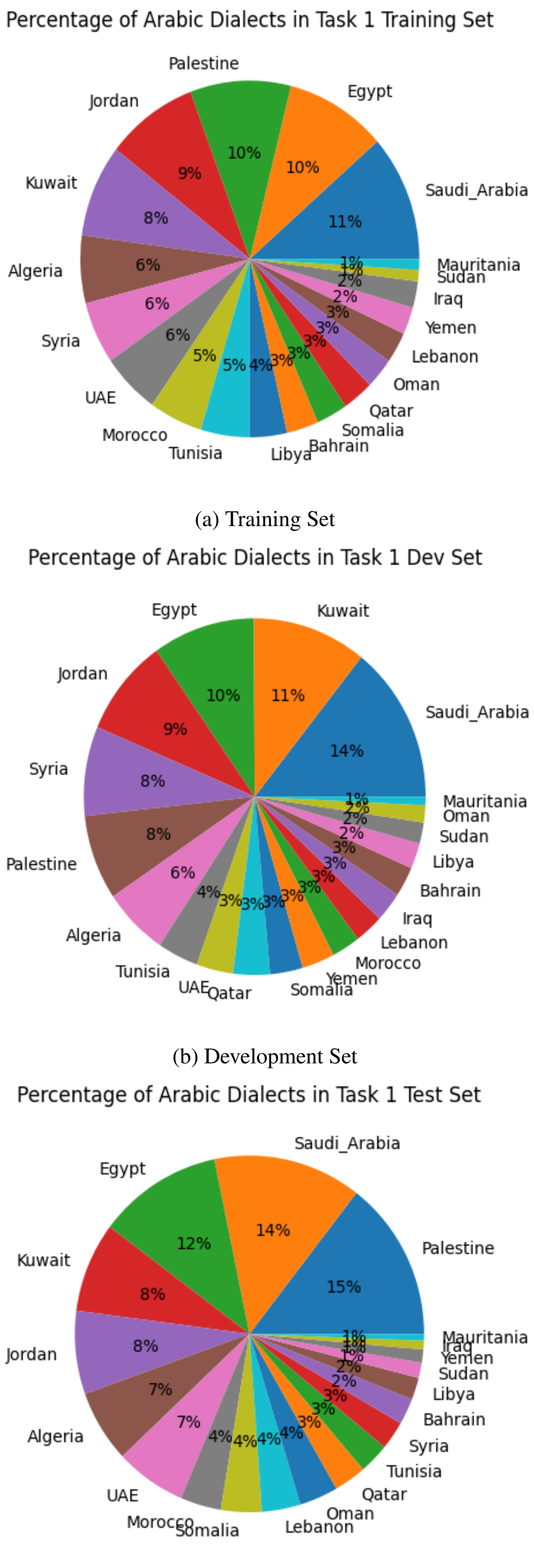
Percentage of Arabic Data Type in Dev Set

(b) Development Set

Percentage of Arabic Data Type in Test Set

(c) Testing Set

Figure 1: Text Type Distribution in Task 1



(a) Training Set

(b) Development Set

(c) Testing Set

Figure 2: Arabic Dialect Identification in Task 1

Hyperparameters	Range Or Value
Batch Size	8
Random Seed	42
Learning Rate	2e-05
Number of Epochs	10
Max Length	512
Total Steps	600
Optimizer	AdamW

Table 6: Hyperparameters for System Implementation

Fine-tuned SVM

Hyperparameter	Value
C	1
max_iter	1000

Fine-tuned RF

Hyperparameter	Value
criterion	gini
n_estimators	200

Fine-tuned LR

Hyperparameter	Value
C	100
penalty	L2

Fine-tuned XGB

Hyperparameter	Value
max_depth	6
n_estimators	200

Table 7: Hyperparameters for Baseline Models

A.4 Best Model Performance

After conducting a total of 22 experiments across two subtasks using 11 models. We will focus on the two best performing models in each subtask on the testing subsets:

i. Best performing *ReDASPersuasion* with Multilingual Transformers: “XLM RoBERTa” on Task 1A, and

ii. Best performing *ReDASPersuasion* with Monolingual Transformers: “CAMeLBERT-DA SA” on Task 1B.

We carefully examine the confusion matrix plots to gain insights into the performance of our models. By doing so, we can determine which classes posed more difficulty for the classifiers.

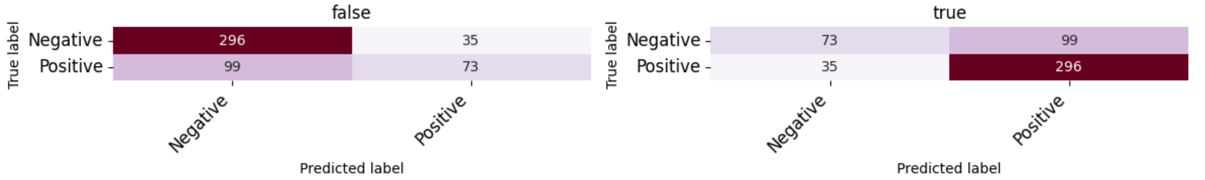


Figure 3: Confusion Matrix of *ReDASPersuasion* employing **XLM RoBERTa** for Binary Persuasion Classification on *Task 1A Testing Set.*

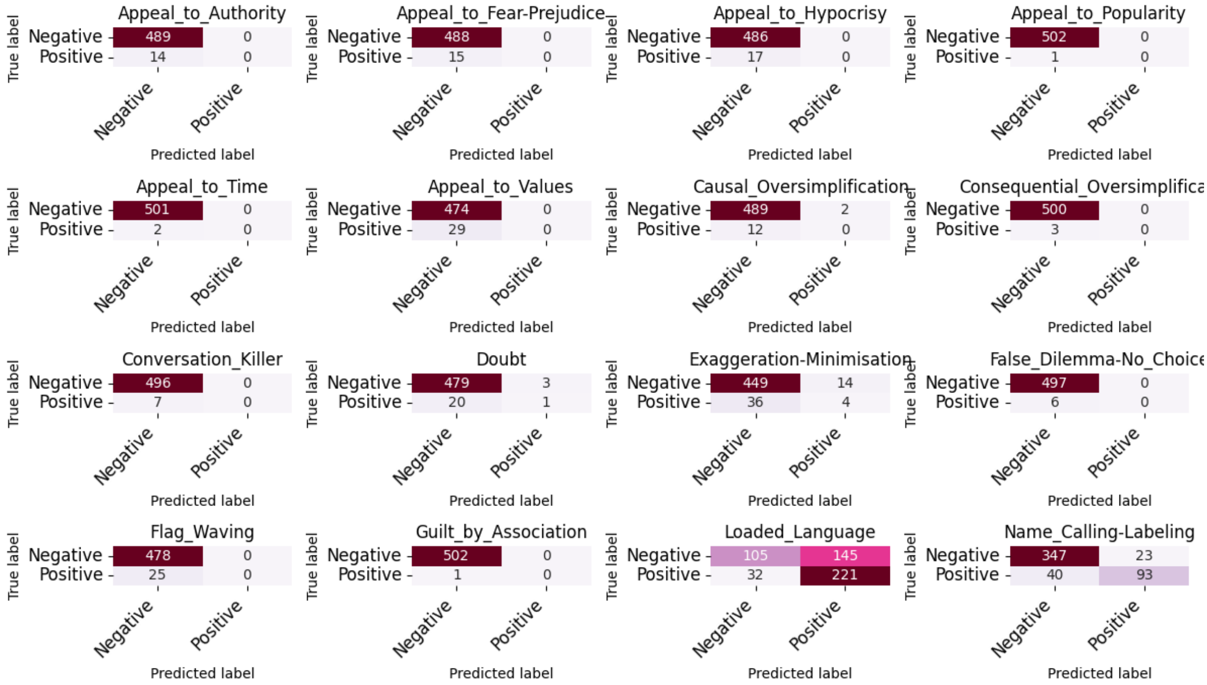


Figure 4: Confusion Matrix of *ReDASPersuasion* employing **CAMeLBERT-DA SA** for Multi-Label Persuasion Classification on *Task 1B Testing Set.*

UL & UM6P at ArAIEval Shared Task: Transformer-based Model for Persuasion Techniques and Disinformation Detection in Arabic

Salima Lamsiyah[1], **Abdelkader El Mahdaouy**[2], **Hamza Alami**[2]
Ismail Berrada[2] and **Christoph Schommer**[1]
[1]Dept. of Computer Science, Faculty of Science, Technology and Medicine,
University of Luxembourg, Luxembourg
[2]College of Computing, Mohammed VI Polytechnic University, Morocco
`{firstname.lastname}@{uni.lu`[1]`, um6p.ma`[2]`}`

Abstract

In this paper, we introduce our participating system to the ArAIEval Shared Task, addressing both the detection of persuasion techniques and disinformation tasks. Our proposed system employs a pre-trained transformer-based language model for Arabic, alongside a classifier. We have assessed the performance of three Arabic Pre-trained Language Models (PLMs) for sentence encoding. Additionally, to enhance our model's performance, we have explored various training objectives, including Cross-Entropy loss, regularized Mixup loss, asymmetric multi-label loss, and Focal Tversky loss. On the official test set, our system has achieved micro-F1 scores of 0.7515, 0.5666, 0.904, and 0.8333 for Sub-Task 1A, Sub-Task 1B, Sub-Task 2A, and Sub-Task 2B, respectively. Furthermore, our system has secured the 4th, 1st, 3rd, and 2nd positions, respectively, among all participating systems in sub-tasks 1A, 1B, 2A, and 2B of the ArAIEval shared task.

1 Introduction

Social media platforms have transformed into significant spaces where people communicate and collect information from various sources. However, along with this positive shift, a significant amount of false, misleading, and harmful content has also emerged. This includes various forms of problematic content like misinformation, disinformation, and malinformation in the form of spreading propaganda, conspiracy theories, rumors, hoaxes, fake news, false statements, hate speech, cyberbullying, and among others (Oshikawa et al., 2020; Alam et al., 2021; Sharara et al., 2022; Essefar et al., 2021; Nakov et al., 2021a; Alam et al., 2022; Lamsiyah et al., 2023; Mubarak et al., 2023).

Furthermore, the surge in online communication platforms has also made it more important to understand how people try to persuade each other. The persuasion detection task involves the identification and analysis of communication strategies aimed at influencing individuals' beliefs or actions. It encompasses recognizing techniques such as emotional appeals, logical reasoning, and rhetorical devices in various forms of content (Dimitrov et al., 2021). Propaganda, a subset of persuasive communication, refers to the deliberate dissemination of information, often with a biased or misleading intent, to manipulate opinions or behaviors. It involves employing well-defined psychological and rhetorical methods to sway audiences (Alam et al., 2022). Several shared tasks have been organized for the detection of propaganda techniques in text and memes. This includes the NLP4IF-2019 shared task on Fine-Grained Propaganda Detection (Da San Martino et al., 2019), SemEval-2020 task 11 on Detection of Persuasion Techniques in News Articles (Da San Martino et al., 2020), and SemEval-2021 task 6 on Detection of Persuasion Techniques in Texts and Images (Dimitrov et al., 2021). In addition to detecting propaganda techniques, another intriguing task is to identify misleading content within social media. This aims to uncover various forms of disinformation, such as hate speech, offensive language, rumors, and spam (Barrón-Cedeno et al., 2020; Nakov et al., 2021b; Shahi et al., 2021).

Most of the previously mentioned research works have primarily focused on the English language. Therefore, there is a noteworthy need to develop such methods for the Arabic language, which is spoken by a considerable number of people globally, with an estimated 372 to 446 million speakers worldwide. With the aim of bridging this language gap, Hasanain et al. (2020) have presented a description of three Arabic tasks that were offered as part of the third edition of the CheckThat! lab at CLEF 2020. It focused on false information propagated on Arabic social media, particularly on Twitter. Furthermore, Alam et al. (2022) have run a shared task on detecting propaganda techniques in Arabic tweets as part of the WANLP 2022 work-

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 558–564
December 7, 2023 ©2023 Association for Computational Linguistics

shop. More recently, Hasanain et al. (2023) have introduced the ArAIEval shared task that includes two tasks: (i) persuasion techniques detection (Sub-Task 1A and Sub-Task 1B), and (ii) disinformation detection (Sub-Task 2A and Sub-Task 2B) in the Arabic Language.

In this paper, we present our submitted system for the ArAIEval shared task (Hasanain et al., 2023), where we tackle both the tasks of detecting persuasion techniques and identifying disinformation. Our system utilizes a deep learning model that comprises a transformer-based Pretrained Language Model (PLM) encoder designed for the Arabic language, coupled with a classifier. The classifier consists of a dropout layer followed by a linear layer. To encode text inputs, we have evaluated the performance of three Arabic PLMs: ARBERTv2, MARBERTv2, and AraBERT-large (Abdul-Mageed et al., 2021; Elmadany et al., 2022; Antoun et al., 2020). During the model training process, we have explored the following training objectives:

- **Sub-Task 1A and Sub-Task 2A**: We have used the cross-entropy loss and the regularized Mixup (RegMixup) loss (Pinto et al., 2023).

- **Sub-Task 1B**: We have evaluated the binary cross-entropy loss, the asymmetric loss for multi-label classification (Ben-Baruch et al., 2020), and the RegMixup loss (Pinto et al., 2023).

- **Sub-Task 2B**: We have employed the cross-entropy loss and the Focal Tversky loss (Abraham and Khan, 2018).

Our system is evaluated using the weighted-average Precision and Recall as well as the micro and macro F1 score. It has achieved micro-F1 scores of 0.7515, 0.5666, 0.904, and 0.8333 on the test sets of Sub-Task 1A, Sub-Task 1B, Sub-Task 2A, and Sub-Task 2B, respectively. Furthermore, our system has secured the 4th, 1st, 3rd, and 2nd positions, respectively, among all participating systems in the corresponding Sub-Tasks of the ArAIEval shared task. It is worth mentioning that the best results have been obtained using the ARBERT sentence encoder for both Sub-Task 1B and Sub-Task 2A. While, for Sub-Task 1A and Sub-Task 2B, the best performance has been achieved using the MARBERTv2 encoder.

2 Data

The ArAIEval shared task (Hasanain et al., 2023) comprises two tasks: persuasion techniques detecting (Sub-Task 1A and Sub-Task 1B), as well as disinformation detection (Sub-Task 2A and Sub-Task 2B) in Arabic. Table 1 describes the provided data for each sub-task. For persuasion techniques detection, the ArAIEval organizers propose the following two sub-tasks:

- **Sub-Task 1A**: is a binary classification task that detects whether a given input tweet or news paragraph contains a persuasion technique.

- **Sub-Task 1B**: is a multi-label classification task that aims to identify the persuasion techniques in a given tweet or news paragraph. The label set of this sub-task contains 24 labels.

For disinformation detection, the ArAIEval organizers provide data for the following two sub-tasks:

- **Sub-Task 2A**: is a binary classification task that aims to detect whether a given input tweet is disinformative.

- **Sub-Task 2B**: is a multi-class classification task that aims to identify the disinformation class of a given input tweet. The class labels include hate-speech, offensive, rumor, and spam.

Task	Train Set	Dev Set	Test Set	Num of classes	Domain
Sub-Task 1A	2427	259	503	2	Twitter and News
Sub-Task 1B	2427	259	503	24	Twitter and News
Sub-Task 2A	14147	2115	3729	2	Twitter
Sub-Task 2B	2656	397	876	4	Twitter

Table 1: ArAIEval subtasks data description

3 System Overview

3.1 Model Architecture

The proposed system comprises a BERT-based Arabic PLM encoder and a single classifier. The classifier consists of a dropout layer followed by a linear layer (feed-forward layer) with an activation function. The number of output units in the linear layer matches the number of classes. For Sub-Task 1A, 2A, and 2B, we have employed the Softmax activation, while for Sub-Task 1B, we have used the

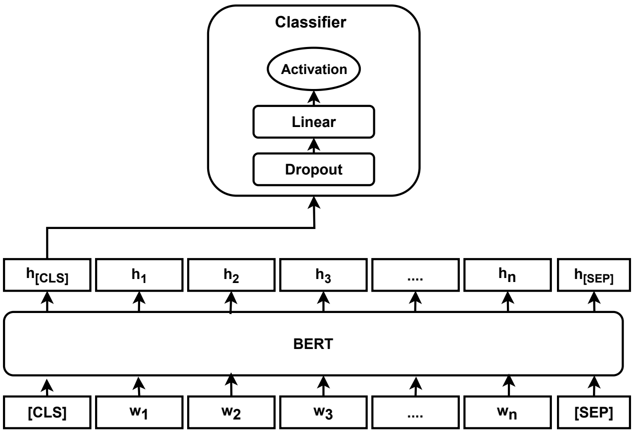


Figure 1: Overall Model Architecture

Sigmoid activation. The overall model architecture is depicted in Figure 1.

For the input texts encoding, we have explored the performance of three existing BERT-based Arabic PLMs, including ARBERTv2, MARBERTv2, and AraBERT-large(Abdul-Mageed et al., 2021; Elmadany et al., 2022; Antoun et al., 2020). These PLMs have been trained on large Arabic textual corpora, covering both Modern Standard Arabic and Dialectal Arabic, using the masked language modeling objective function.

As shown in Figure 1, given an input text of length m, the PLM's tokenizer split it into n sub-words and append the $[CLS]$ and $[SEP]$ special tokens, representing the start and end of the input sequence, to the tokenized text $([CLS], w_1, w_2, w_3, ..., w_n, [SEP])$. Then, the BERT-based encoder is fed with the tokenized text and outputs the contextualized word embedding $h_{[CLS]}, h_1, h_2, h_3, ..., h_n, h_{[SEP]}$. Finally, the pooled output of the $[CLS]$ token is passed to the classifier to predict the class label of the input text.

3.2 Training objectives

For model training, we have explored the following training objectives:

- $\mathcal{L}_{CE}$ denotes the Cross-Entropy (CE) loss;
- $\mathcal{L}_{BCE}$ denotes the Binary Cross-Entropy (BCE) loss;
- $\mathcal{L}_{ASL}$ denotes the Asymmetric Loss (ASL) for multi-label classification (Ben-Baruch et al., 2020). This loss function deals with the negative-positive imbalance in multi-label classification;
- $\mathcal{L}_{FT}$ denotes the Focal Tversky (FT) loss (Abraham and Khan, 2018). This loss function is a generalization of the focal loss and employs the Tversky index. It deals with the class imbalance problem.
- $\mathcal{L}_{RegMix}$ denotes the Regularized Mixup (RegMix) loss (Pinto et al., 2023). This loss is employed as a regularizer to the cross-entropy loss to improve the model's generalization. Formally, give two pair of examples and their corresponding labels from the training dataset (x_i, y_i) and (x_j, y_j), the Mixup is calculated as $\tilde{x_i} = \lambda \cdot x_i + (1 - \lambda) \cdot x_j$ and $\tilde{y_i} = \lambda \cdot y_i + (1 - \lambda) \cdot y_j$. Where $\lambda \sim Beta(\alpha, \alpha) \in [0, 1]$ for $\alpha \in [0, \infty[$. Then, the RegMix loss is calculated as follows:

$$\mathcal{L}^{*}_{RegMix} = \mathcal{L}_{*}(x, y) + p \cdot \mathcal{L}_{*}(\tilde{x}, \tilde{y}) \quad (1)$$

where $*$ and p denote a loss function like cross-entropy loss and Mixup weighting hyperparameter. Since text mixup is not feasible, we employ mixup of the pooled output (h_i and h_j) of x_i and x_j.

For our models training on each sub-task, we have investigated the following training objectives:

- **Sub-Task 1A and Sub-Task 2A**: $\mathcal{L}_{CE}$ and $\mathcal{L}^{CE}_{RegMix}$
- **Sub-Task 1B**: $\mathcal{L}_{BCE}$, $\mathcal{L}_{ASL}$, and $\mathcal{L}^{ASL}_{RegMix}$
- **Sub-Task 2B**: $\mathcal{L}_{CE}$ and $\mathcal{L}_{FT}$

4 Experiments and Results

In this section, we present the experiment settings and the obtained results for each sub-task.

4.1 Experiment Settings

All our models have been implemented using the Pytorch deep learning framework, Pytorch Lightning, and Hugging Face Transformers library. We have performed our experiments on a Dell PowerEdge C4140 server, having 4 Nvidia V100 SXM2 32GB. For all sub-tasks, we have trained our models for a maximum of 10 epochs with a batch size of 16 examples and a learning rate of 1×10^{-5}. Early stopping is configured to 3 epochs. Besides, a weight decay of 1×10^{-3} is applied to all the layers of the model weights except biases and Layer Normalization (LayerNorm). In all our experiments,

		Dev				**Test**			
Encoder	**Loss**	**Precision**	**Recall**	**F1-macro**	**F1-micro**	**Precision**	**Recall**	**F1-macro**	**F1-micro**
AraBERT	$\mathcal{L}_{CE}$	0.826	0.834	0.7432	0.834	0.7438	0.7416	0.7152	0.7416
ARBERTv2		0.8102	0.8147	0.723	0.8147	0.7494	0.7455	0.721	0.7455
MARBERTv2		0.8437	0.8494	0.7703	0.8494	0.7569	0.7495	0.7281	0.7495
AraBERT	$\mathcal{L}_{RegMix}^{CE}$	0.8452	0.8533	0.7489	0.8533	0.7409	0.7475	0.7085	0.7475
ARBERTv2		0.8122	0.8263	0.6833	0.8263	0.7259	0.7356	0.6847	0.7356
MARBERTv2		**0.8622**	**0.8687**	**0.7893**	**0.8687**	**0.7476**	**0.7515**	**0.7186**	**0.7515**†

Table 2: The obtained results of our system on Sub-Task 1A. Our official submission results are highlighted in bold font. † is attached to the best obtained micro-F1 score.

		Dev				**Test**			
Encoder	**Loss**	**Precision**	**Recall**	**F1-macro**	**F1-micro**	**Precision**	**Recall**	**F1-macro**	**F1-micro**
AraBERT	$\mathcal{L}_{BCE}$	0.5757	0.5286	0.1166	0.6175	0.5181	0.4574	0.1035	0.5427
ARBERTv2		0.5879	0.5207	0.1176	0.619	0.527	0.4695	0.1044	0.5546
MARBERTv2		0.5397	0.5247	0.1098	0.6011	0.4808	0.4585	0.0976	0.5401
AraBERT	$\mathcal{L}_{ASL}$	**0.6286**	**0.6864**	**0.3296**	**0.6622**	**0.5833**	**0.5415**	**0.2156**	**0.5666**
ARBERTv2		0.592	0.6568	0.3315	0.6201	0.56	0.5526	0.2242	0.5538
MARBERTv2		0.6206	0.6844	0.2971	0.6438	0.5578	0.5604	0.1908	0.5766†
AraBERT	$\mathcal{L}_{RegMix}^{ASL}$	0.6059	0.6726	0.3285	0.644	0.5747	0.5482	0.2286	0.5651
ARBERTv2		0.5819	0.6785	0.3168	0.6243	0.5555	0.5637	0.2064	0.5678
MARBERTv2		0.6124	0.6903	0.2966	0.6512	0.5809	0.5648	0.2082	0.5756

Table 3: The obtained results of our system on Sub-Task 1B. Our official submission results are highlighted in bold font. † is attached to the best obtained micro-F1 score.

		Dev				**Test**			
Encoder	**Loss**	**Precision**	**Recall**	**F1-macro**	**F1-micro**	**Precision**	**Recall**	**F1-macro**	**F1-micro**
AraBERT	$\mathcal{L}_{CE}$	**0.9118**	**0.9144**	**0.8535**	**0.9144**	**0.9028**	**0.904**	**0.8645**	**0.904**
ARBERTv2		0.8972	0.9012	0.8283	0.9012	0.895	0.8976	0.8521	0.8976
MARBERTv2		0.9064	0.9078	0.8463	0.9078	0.905	0.9067	0.8672	0.9067†
AraBERT	$\mathcal{L}_{RegMix}^{CE}$	0.9101	0.9125	0.8515	0.9125	0.9034	0.9037	0.8656	0.9037
ARBERTv2		0.9002	0.9045	0.8294	0.9045	0.8935	0.8965	0.8479	0.8965
MARBERTv2		0.9096	0.913	0.845	0.913	0.9016	0.904	0.8583	0.904

Table 4: The obtained results of our system on Sub-Task 2A. Our official submission results are highlighted in bold font. † is attached to the best obtained micro-F1 score.

		Dev				**Test**			
Encoder	**Loss**	**Precision**	**Recall**	**F1-macro**	**F1-micro**	**Precision**	**Recall**	**F1-macro**	**F1-micro**
AraBERT	$\mathcal{L}_{CE}$	0.8234	0.8262	0.7973	0.8262	0.8205	0.8242	0.724	0.8242
ARBERTv2		0.8261	0.8287	0.8109	0.8287	0.8174	0.8151	0.7336	0.8151
MARBERTv2		0.8327	0.8363	0.795	0.8363	0.8345	0.8379	0.7443	0.8379†
AraBERT	$\mathcal{L}_{FT}$	0.8182	0.8212	0.7898	0.8212	0.818	0.8208	0.7231	0.8208
ARBERTv2		0.835	0.8388	0.8	0.8388	0.8055	0.8071	0.7024	0.8071
MARBERTv2		**0.8471**	**0.8514**	**0.8121**	**0.8514**	**0.8367**	**0.8333**	**0.7388**	**0.8333**

Table 5: The obtained results of our system on Sub-Task 2B. Our official submission results are highlighted in bold font. † is attached to the best obtained micro-F1 score.

we have fixed the maximum sequence length to 128. The hyper-parameters α (Beta distribution parameter) and p of the $\mathcal{L}_{RegMix}$ are set to 20 and 0.2, respectively. For $\mathcal{L}_{ASL}$ loss function, the hyper-parameters γ_{-} and γ_{+} are fixed to 4 and 1, respectively. The hyper-parameter α of the Focal Tversky loss ($\mathcal{L}_{FT}$) is set to 0.5. It is worth mentioning that we have trained, validated, and evaluated our models on the officially provided splits for training, validation, and development, respectively. For the evaluation purpose, we have employed the weighted Recall and Precision as well as the micro and macro F1 scores.

4.2 Results

4.2.1 Sub-Task 1A

Table 2 summarizes our obtained results for Sub-Task 1A. The overall results show that employing the MARBERTv2 encoder leads to better performance using both the cross-entropy loss and the RegMix loss. Although the RegMix training objective largely enhances the results on the dev set, it achieves small performance improvements on the test set when AraBERT and MARBERTv2 encoders are utilized. The best results are obtained using the RegMix training objective in conjunction with the MARBERTv2 encoder. The latter corresponds to our official submission.

4.2.2 Sub-Task 1B

Table 3 shows our system's obtained results for Sub-Task 1B. The overall results demonstrate that the AraBERT and MARBERTv2 lead to better results for most training objectives. The asymmetric loss ($\mathcal{L}_{ASL}$) improves the classification results of all the used encoders and shows important performance increments for the macro-F1 and micro-F1 scores. Besides, the best micro-F1 score on the test set is obtained using the asymmetric loss in conjunction with the MARBERTv2 encoder. The RegMix training objective with the ASL loss ($\mathcal{L}^{ASL}_{RegMix}$) enhances the results when the ARBERTv2 encoder is employed. However, it negatively impacts the performance when the other two encoders are utilized. For the official evaluation, we have submitted our model that uses an AraBERT encoder, trained using the ASL loss.

4.2.3 Sub-Task 2A

Table 4 presents our obtained results for Sub-Task 2A. The overall results show that AraBERT and MARBERTv2 encoders yield better results than ARBERTv2. The RegMix training objective slightly degrades the F1 scores performance of our systems. Our best micro-F1 score is obtained using MARBERTv2 in conjunction with the CE training objective. Whereas, our official submitted model is trained using the CE loss and AraBERT encoder.

4.2.4 Sub-Task 2B

Table 5 summarizes our obtained results for Sub-Task 2B. The overall results show that the MARBERTv2 outperforms the other pre-trained models. Although the Focal Tversky loss has been shown to improve the results of ARBERTv2 and MARBERTv2 on the dev set, it negatively impacts our model performance on the test set. The best micro-F1 score is achieved by using the MARBERTv2 encoder in conjunction with CE loss. Whereas, our official submitted model is trained using the FT loss and MARBERTv2 encoder.

5 Discussion

The obtained results have shown that the training objective and the text encoder have a significant impact on our models' performance. The overall results demonstrate the effectiveness of the PLMs encoders that are pre-trained on large text corpora from the same domain as the target downstream tasks (MARBERTv2). A straightforward path of future research work is to investigate the performances of other state-of-the-art Arabic PLMs and other training objectives that deal with the class imbalance problem.

6 Conclusion

In this paper, we have introduced our submitted system to the ArAIEval Shared Task for persuasion techniques and disinformation detection in Arabic. Our System uses a deep learning model that consists of a transformer-based Pre-trained Language Model (PLM) encoder for the Arabic language and a classifier. For the model training, we have explored several training objectives and assessed the performance of three Arabic PLMs. On the official test set, our system has obtained micro-F1 scores of 0.7515, 0.5666, 0.904, and 0.8333 for Sub-Task 1A, Sub-Task 1B, Sub-Task 2A, and Sub-Task 2B, respectively. Besides, it has been ranked in the 4th, 1st, 3rd, and 2nd positions among all participating systems in Sub-Task 1A, Sub-Task 1B, Sub-Task 2A, and Sub-Task 2B, respectively.

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.

Nabila Abraham and Naimul Mefraz Khan. 2018. A novel focal tversky loss function with improved attention u-net for lesion segmentation.

Firoj Alam, Hamdy Mubarak, Wajdi Zaghouani, Giovanni Da San Martino, and Preslav Nakov. 2022. Overview of the WANLP 2022 shared task on propaganda detection in Arabic. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 108–118, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Firoj Alam, Shaden Shaar, Fahim Dalvi, Hassan Sajjad, Alex Nikolov, Hamdy Mubarak, Giovanni Da San Martino, Ahmed Abdelali, Nadir Durrani, Kareem Darwish, Abdulaziz Al-Homaid, Wajdi Zaghouani, Tommaso Caselli, Gijs Danoe, Friso Stolk, Britt Bruntink, and Preslav Nakov. 2021. Fighting the COVID-19 infodemic: Modeling the perspective of journalists, fact-checkers, social media platforms, policy makers, and the society. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 611–649, Punta Cana, Dominican Republic. Association for Computational Linguistics.

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.

Alberto Barrón-Cedeno, Tamer Elsayed, Preslav Nakov, Giovanni Da San Martino, Maram Hasanain, Reem Suwaileh, Fatima Haouari, Nikolay Babulkov, Bayan Hamdan, Alex Nikolov, et al. 2020. Overview of checkthat! 2020: Automatic identification and verification of claims in social media. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction: 11th International Conference of the CLEF Association, CLEF 2020, Thessaloniki, Greece, September 22–25, 2020, Proceedings 11*, pages 215–236. Springer.

Emanuel Ben-Baruch, Tal Ridnik, Nadav Zamir, Asaf Noy, Itamar Friedman, Matan Protter, and Lihi Zelnik-Manor. 2020. Asymmetric loss for multi-label classification.

Giovanni Da San Martino, Alberto Barrón-Cedeño, and Preslav Nakov. 2019. Findings of the NLP4IF-2019 shared task on fine-grained propaganda detection. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, pages 162–170, Hong Kong, China. Association for Computational Linguistics.

Giovanni Da San Martino, Alberto Barrón-Cedeño, Henning Wachsmuth, Rostislav Petrov, and Preslav Nakov. 2020. SemEval-2020 task 11: Detection of propaganda techniques in news articles. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1377–1414, Barcelona (online). International Committee for Computational Linguistics.

Dimitar Dimitrov, Bishr Bin Ali, Shaden Shaar, Firoj Alam, Fabrizio Silvestri, Hamed Firooz, Preslav Nakov, and Giovanni Da San Martino. 2021. SemEval-2021 task 6: Detection of persuasion techniques in texts and images. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 70–98, Online. Association for Computational Linguistics.

AbdelRahim Elmadany, El Moatez Billah Nagoudi, and Muhammad Abdul-Mageed. 2022. Orca: A challenging benchmark for arabic language understanding. *arXiv preprint arXiv:2212.10758*.

Kabil Essefar, Abdellah El Mekki, Abdelkader El Mahdaouy, Nabil El Mamoun, and Ismail Berrada. 2021. CS-UM6P at SemEval-2021 task 7: Deep multi-task learning model for detecting and rating humor and offense. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 1135–1140, Online. Association for Computational Linguistics.

Maram Hasanain, Firoj Alam, Hamdy Mubarak, Samir Abdaljalil, Wajdi Zaghouani, Preslav Nakov, Giovanni Da San Martino, and Abed Alhakim Freihat. 2023. ArAIEval Shared Task: Persuasion Techniques and Disinformation Detection in Arabic Text. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore. Association for Computational Linguistics.

Maram Hasanain, Fatima Haouari, Reem Suwaileh, Zien Sheikh Ali, Bayan Hamdan, Tamer Elsayed, Alberto Barrón-Cedeño, Giovanni Da San Martino, and Preslav Nakov. 2020. Overview of checkthat! 2020i arabic: Automatic identification and verification of claims in social media. In *Conference and Labs of the Evaluation Forum*.

Salima Lamsiyah, Abdelkader El Mahdaouy, Hamza Alami, Ismail Berrada, and Christoph Schommer. 2023. UL & UM6P at SemEval-2023 task 10: Semi-supervised multi-task learning for explainable detection of online sexism. In *Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 644–650, Toronto, Canada. Association for Computational Linguistics.

Hamdy Mubarak, Samir Abdaljalil, Azza Nassar, and Firoj Alam. 2023. Detecting and identifying the reasons for deleted tweets before they are posted. *Frontiers in Artificial Intelligence*, 6.

Preslav Nakov, Firoj Alam, Shaden Shaar, Giovanni Da San Martino, and Yifan Zhang. 2021a. Covid-19 in bulgarian social media: Factuality, harmfulness, propaganda, and framing. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 997–1009.

Preslav Nakov, Giovanni Da San Martino, Tamer Elsayed, Alberto Barrón-Cedeno, Rubén Míguez, Shaden Shaar, Firoj Alam, Fatima Haouari, Maram Hasanain, Nikolay Babulkov, et al. 2021b. The clef-2021 checkthat! lab on detecting check-worthy claims, previously fact-checked claims, and fake news. In *Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28–April 1, 2021, Proceedings, Part II 43*, pages 639–649. Springer.

Ray Oshikawa, Jing Qian, and William Yang Wang. 2020. A survey on natural language processing for fake news detection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6086–6093, Marseille, France. European Language Resources Association.

Francesco Pinto, Harry Yang, Ser-Nam Lim, Philip H. S. Torr, and Puneet K. Dokania. 2023. Regmixup: Mixup as a regularizer can surprisingly improve accuracy and out distribution robustness.

Gautam Kishore Shahi, Julia Maria Struß, and Thomas Mandl. 2021. Overview of the clef-2021 checkthat! lab: Task 3 on fake news detection. In *CLEF (Working Notes)*, pages 406–423.

Mohamad Sharara, Wissam Mohamad, Ralph Tawil, Ralph Chobok, Wolf Assi, and Antonio Tannoury. 2022. Arabert model for propaganda detection. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 520–523.

AAST-NLP at ArAIEval Shared Task: Tackling Persuasion Technique and Disinformation Detection using Pre-Trained Language Models On Imbalanced Datasets

Ahmed El-Sayed[1*], **Omar Nasr**[1*], **Nour El-din El-madany**[2]
Intelligent System Lab
[1]Department of Computer Engineering, Arab Academy for Science and Technology
[2]Department of Electronics And Communication, Arab Academy for Science and Technology
{ahmedelsayedhabashy,omarnasr5206}@gmail.com,nourelmadany@aast.edu

Abstract

This paper presents the pipeline developed by the AAST-NLP team to address both the persuasion technique detection and disinformation detection shared tasks. The proposed system for all the tasks' sub-tasks consisted of preprocessing the data and finetuning AraBERT on the given datasets, in addition to several procedures performed for each subtask to adapt to the problems faced in it. The previously described system was used in addition to Dice loss as the loss function for sub-task 1A, which consisted of a binary classification problem. In that sub-task, the system came in eleventh place. We trained AraBERT for task 1B, which was a multi-label problem with 24 distinct labels, using binary cross-entropy to train a classifier for each label. On that sub-task, the system came in third place. We utilised AraBERT with Dice loss on both subtasks 2A and 2B, ranking second and third among the proposed models for the respective subtasks.

1 Introduction

Social media has become part and parcel of our everyday lives and a main source of information for every individual. Unfortunately, due to the nature of social media, the spread of disinformation (Alam et al., 2022a) is very relevant and causes major troubles. For example back in the COVID-19 pandemic, some researchers coined the term "infodemic" to describe the spread of false information among people during that period (Geldsetzer, 2020). Many researchers have proposed their systems to fight the spread of disinformation on social media platforms, powered by recent advances in NLP and the introduction of Large Language models including BERT (Devlin et al., 2019) which revolutionized NLP and was adapted to many tasks.

Persuasion is a type of social interaction that attempts to influence and change attitudes in an atmosphere of free choice (Perloff, 2017). Persuasion techniques are incredibly important linguistic techniques that can have massive effects on different fields and industries. An example of this is the usage of these techniques in advertising campaigns, which can lead to impressive results when it comes to changing customers attitudes and receiving their responses without imposing on them (Romanova and Smirnova, 2019). This paper tackles the various systems our team attempted for the ArAIEval 2023 shared tasks (ove). The first step was to look at some of the earlier publications from WANLP 2022 (Alam et al., 2022b), which provided a number of crucial insights that served as a foundation for our work. Related work includes the system presented in (Mubarak et al., 2023) for the identification of disinformation through samples, combined with many additional significant results as well as fine-grained disinformation labels from those samples. The following sections of the paper comprise a data section which describes the data sources and preprocessing methods applied to the data. A system section describing the pipeline, a results section, a discussion and a summary.

2 Data

In this section, we will describe the data sources and the preprocessing methods that we applied to prepare the data. We will also provide some descriptive statistics and visualizations of the data to give an overview of its characteristics and distribution.

2.1 Data Description

2.1.1 Persuasion Technique Detection

Task 1 consists of two subtasks, namely subtask 1A and 1B. The first is to determine whether the

[*]. equally contributed

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 565–569
December 7, 2023 ©2023 Association for Computational Linguistics

tweets and paragraphs contain any persuasive techniques. The second sub-task expanded on the first by identifying the various persuasive strategies that were found in those samples.

Dataset	Train	Dev	Test
Texts	2427	259	503

Table 1: Data distribution for task 1.

The training dataset includes 2427 samples labelled as True or False, with a distribution of 1918 to 509, respectively. This indicates that the ratio of true to false cases is roughly 65.8% to 34.2%, as illustrated in Figure 1, demonstrating that the dataset had a class imbalance. This percentage was matched in the development data, which had a distribution of 202 to 57 respectively.

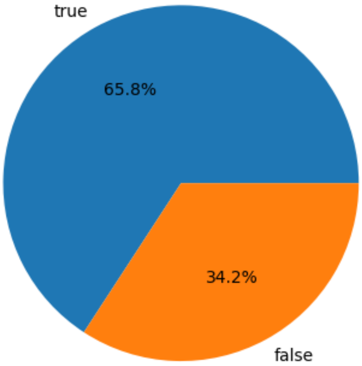


Figure 1: Data distribution for the training data for subtask 1A.

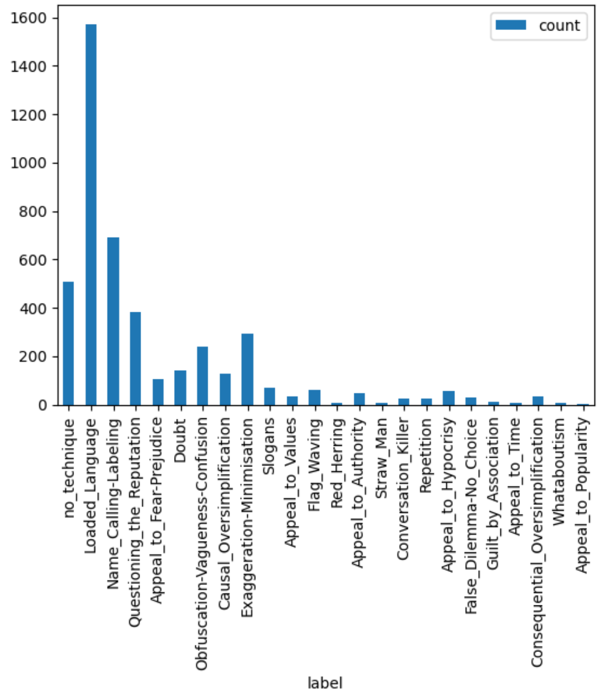


Figure 2: Subtask 1B Training Class Distribution.

The data for subtask 1B consists of samples labelled from 24 different class labels which represents the different persuasive techniques. Some of the common techniques found are "Loaded Language", "Name calling/labelling" and "Questioning the Reputation". We hypothesize that there are underlying dependencies between the techniques and correlations between different combinations which makes it a very interesting task and worthy of further exploration. As shown in Figure 2 the class distribution is severely unbalanced with underrepresented classes including "Appeal to Popularity", "Whataboutism" and several others.

2.1.2 Disinformation Detection

The objective of Task 2 comprises two subtasks. The first is classifying the samples into information and disinformation. The second involves classifying the given samples into one of four sub-classes: HS, OFF, Spam, and Rumour.

Dataset	Train	Dev	Test
Tweets	14147	2115	3729

Table 2: Data distribution for task 2A.

Dataset	Train	Dev	Test
Tweets	2648	396	876

Table 3: Data distribution for task 2B.

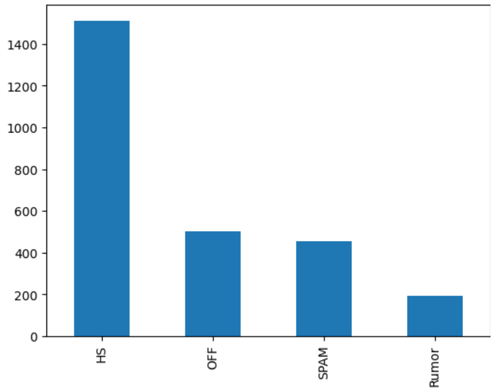


Figure 3: Data distribution for subtask 2B.

Task 2 suffers from a class imbalance problem. Figure 3 demonstrates that the rumour class is significantly underrepresented, but the HS class is significantly overrepresented. The validation set is distributed in a similar manner. Subtask 1A has a similar problem, with a distribution of 11419 to 2656 for no-disinformation to disinformation.

The validation set has a similar problem, with a distribution of 1718 to 397 for no-disinformation to disinformation.

2.2 Preprocessing

The preprocessing procedure of our data took the following steps.

- Removing Arabic stop words.
- Removing tweet related tags like LINK , RT, [مستخدم] and [فيديو].
- Applying AraBERT preprocessor, removing tashkeel, tatweel and emojis.
- Removing '_' as some tweets were ambiguously written and formatted with '_' between each letter.

3 System Description

3.1 Model Description

Our initial experiments on the conducted on the development data consisted of comparing several BERT-based models to choose from to build upon, we experimented on AraBERT (Antoun et al., 2021), MarBERT (Muhammad Abdul-Mageed and Nagoudi), ArBERT and bert-base-arabic (Safaya et al., 2020), AraBERT outperformed its peers on the 4 subtasks and was the one chosen to further experiment upon.

3.2 Addressing Class Imbalance

After inspecting the data, it was clear that one of the problems that would hinder our experiments would be the severe case of class imbalance that the provided datasets were suffering from. We experiment with three methods to mitigate the effects of imbalances in datasets. The following sections give details of each method and its corresponding effects on the results.

3.2.1 Re-Sampling

Re-sampling is the process of increasing the importance of minority classes by altering the distribution of the training datasets (Kraiem et al.). Random under sampling (RUS) consists of randomly removing datapoints from the majority class. Random oversampling (ROS) consists of randomly duplicating minority class instances. Both ROS and RUS were used to offset the data imbalance in the dataset.

3.2.2 Data Augmentation

Synthetic data was created using the NLPAUG package[1]. Contextual Word Embeddings Augmentation was used based on AraBERT, and the dataset distributions were altered to increase the importance of classes underepresented in our datasets, but one important remark was that the data created was noisy and required filtering. For example the sample shown below had it's meaning completely changed from the original sentence.
Original Data:
احساسي بقول لي ماف كورونا حتنتشر البلد.
Synthetic Data:
احساسي بقول انه فيروسات كورونا حتنتشر علي هاي البلد.
The augmented data was filtered and revised manually to check if the meaning of the new synthetic sentence matches the original sentence. Synthetic data created using this method resulted in a huge decrease in our micro-F1 score.

3.2.3 Custom Loss Functions

Several loss functions were experimented upon, initially we used Weighted Cross-Entropy loss (Ozdemir and Sonmez, 2020) for our subtasks with weights calculated via scikit (Pedregosa et al., 2011) class weight function it resulted in a slight improvement on the binary classification tasks. Although the adaptation of focal loss appeared interesting at first, it was not robust in handling the imbalance difficulties and led to overfitting. Ultimately, we conducted an experiment using Dice Loss (Li et al., 2019), a customized loss function tailored to NLP tasks based on the Sørensen–Dice coefficient (Li et al., 2019).

$$Diceloss(p, y) = 1 - \frac{2 * \sum_1^t p_i * y_i + smooth}{\sum_1^t p_i + \sum_1^t y_i + smooth} \tag{1}$$

This particular loss function led to an improvement in the F1 score for each of the corresponding tasks.

3.3 Experiment Settings

The training procedure was conducted using the Google Colab platform for training our pipeline, which has 12.68 GB of RAM, a 14.75

[1] https://github.com/makcedward/nlpaug

GB NVIDIA Tesla T4 GPU, and Python language. We used ktrain's (Maiya, 2020) autofit, which applies a triangular learning rate policy (Smith, 2015). The learning rate was determined via the lr_plot function, which experiments with a range of learning rates and suggests multiple possible learning rates. The parameters set for our experiment are mentioned in the table below.

Parameter	Value
Epochs	30
Learning Rate	1e-5
Batch Size	16
Max Length	128
Optimizer	AdamW
Early Stopping Patience	5
Reduce on pleateau	2
Dice loss smoothing	1e-6

Table 4: Training parameters.

Modifications were made to adapt to the task requiremets including changing the loss function to Dice loss for binary and multiclass classification task with smoothing set to 1e-6. For the multilabel task 1B, we used a binary cross entropy loss to train 24 different classifiers each to one of the labels found in the provided dataset.

4 Results

Task	Validation	Test
1A	0.5405	0.4771
1B	0.0938	0.0868
2A	0.5173	0.5154
2B	0.2191	0.2603

Table 5: Baseline micro-f1 scores for all subtasks.

Table 5 presents the random baseline micro-f1 scores on all the respective subtasks. These micro-f1 scores were obtained through the official website of the shared task. These baselines provide a point of reference for the obtained results. The system consistently outperformed these baselines by a significant margin throughout the development process and the outline of the results of the given system is presented in the rest of this section.

Task	Training	Validation	Test
1A	0.9782	0.8301	0.7237
1B	0.8101	0.6295	0.5522
2A	0.9414	0.9031	0.9043
2B	0.9782	0.8301	0.8253

Table 6: Achieved micro-f1 scores for all subtasks.

The micro-f1 scores of the previously mentioned system, which uses AraBERT paired with task specific loss function; Dice loss for the first part of the persuasion technique detection problem and both tasks of the disinformation detection problem, and Binary Cross Entropy for the second task of persuasion technique detection labeling, are shown in Table 6. Micro-f1 was chosen as the competition's evaluation metric, and testing results were obtained once the evaluation process was completed. The results of the persuasion technique detection ranked 11th and 3rd, respectively, while the results of the misinformation detection tasks ranked 2nd and 3rd, respectively.

5 Discussion

A diverse set of limitations were encountered during the development of the aforementioned systems. Another drawback stemmed from the underlying dependencies among task 1B labels, as attempting a direct approach did not lead to optimal outcomes. The subjective labelling of tasks 1B and 2B made it difficult to leverage external data sources to further train our model. One strategy worth highlighting is the use of a CNN-BILSTM and ARABERT hybrid model (Hengle et al., 2021). However, this did not produce satisfactory results since the model appeared to overfit the training instances.With few modifications, this strategy may be viable. Furthermore, the unexpected decline in task 1A's performance necessitates further investigation and experimentation to determine the cause.

6 Summary

The proposed system based on AraBERT was detailed, and the experiments conducted were all addressed. The adaptation of dice loss boosted our performance on all of the tasks and partially addressed the issue of class imbalance yet there is a huge room for improvement. There are other intriguing future directions, like the development of a data augmentation package that supports different data augmentation techniques. Furthering the solution to the issue of class imbalance is another intriguing path. Last but not least, the problem of the underlying dependencies and ways of tackling multilabel tasks should be inspected, and new methods should be investigated and developed in the near future. We intend to invesigate these various approaches in detail in the future since we believe there is still room for improvement in finetuning as well as experimenting with other approaches such as different hybrid model architectures and different data augmentation methods. In the future, we plan to thoroughly explore these diverse approaches because we are convinced that there is further potential for enhancing fine-tuning. This includes experimenting with alternative hybrid model architectures and various data augmentation techniques.

References

Firoj Alam, Stefano Cresci, Tanmoy Chakraborty, Fabrizio Silvestri, Dimiter Dimitrov, Giovanni Da San Martino, Shaden Shaar, Hamed Firooz, and Preslav Nakov. 2022a. A survey on multimodal disinformation detection. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6625–6643, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Firoj Alam, Hamdy Mubarak, Wajdi Zaghouani, Giovanni Da San Martino, and Preslav Nakov. 2022b. 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, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2021. Arabert: Transformer-based model for arabic language understanding.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Pascal Geldsetzer. 2020. Knowledge and Perceptions of COVID-19 among the general public in the United States and the United Kingdom: a cross-sectional online survey. *Annals of Internal Medicine*, 173(2):157–160.

Amey Hengle, Atharva Kshirsagar, Shaily Desai, and Manisha Marathe. 2021. Combining context-free and contextualized representations for Arabic sarcasm detection and sentiment identification. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 357–363, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Mohamed S. Kraiem, F. Sánchez, and María N. Moreno García. Selecting the suitable resampling strategy for imbalanced data classification regarding dataset properties. an approach based on association models. (18):8546.

Xiaoya Li, Xiaofei Sun, Yuxian Meng, Junjun Liang, Fei Wu, and Jiwei Li. 2019. Dice Loss for Data-imbalanced NLP Tasks. *arXiv (Cornell University)*.

Arun S. Maiya. 2020. ktrain: A low-code library for augmented machine learning. *arXiv preprint arXiv:2004.10703*.

Hamdy Mubarak, Samir Abdaljalil, Azza Nassar, and Firoj Alam. 2023. Detecting and reasoning of deleted tweets before they are posted. *arXiv preprint arXiv:2305.04927*.

AbdelRahim A. Elmadany Muhammad Abdul-Mageed and El Moatez Billah Nagoudi. Arbert & marbert: Deep bidirectional transformers for arabic.

Ozgur Ozdemir and Elena Battini Sonmez. 2020. Weighted Cross-Entropy for Unbalanced Data with Application on COVID X-ray images. *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Richard Perloff. 2017. *The dynamics of persuasion: Communication and attitudes in the 21st century*.

Irina Romanova and Irina Smirnova. 2019. Persuasive techniques in advertising. *Training Language and Culture*, 3:55–70.

Ali Safaya, Moutasem Abdullatif, and Deniz Yuret. 2020. KUISAIL at SemEval-2020 task 12: BERT-CNN for offensive speech identification in social media. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 2054–2059, Barcelona (online). International Committee for Computational Linguistics.

Lauren Smith. 2015. Cyclical learning rates for training neural networks. *arXiv (Cornell University)*.

PD-AR at ArAIEval Shared Task: A BERT-Centric Approach to Tackle Arabic Disinformation

Pritam Deka
Queen's University Belfast, UK
pdeka01@qub.ac.uk

Ashwathy T. Revi
University of Southampton, UK
atr1n17@soton.ac.uk

Abstract

This work explores Arabic disinformation identification, a crucial task in natural language processing, using a state-of-the-art NLP model. We highlight the performance of our system model against baseline models, including multilingual and Arabic-specific ones, and showcase the effectiveness of domain-specific pre-trained models. This work advocates for the adoption of tailored pre-trained models in NLP, emphasizing their significance in understanding diverse languages. By merging advanced NLP techniques with domain-specific pre-training, it advances Arabic disinformation identification.

1 Introduction

Disinformation is the deliberate creation and spreading of false or misleading information that can cause public harm or generate profit for organizations that participate in such practices (Tandoc Jr et al., 2018; de Cock Buning, 2018). The consequences of disinformation can be significant, affecting political decisions (Allcott and Gentzkow, 2017), manipulating public opinion, or even inciting violence. Detecting disinformation can be challenging because it can appear similar to real information and spread very quickly. Additionally, creators are constantly evolving their methods, making it more difficult to detect their content.

While disinformation detection in English has received much attention, the nuances, dialectical variations, and morphological richness of Arabic present unique challenges that have not been comprehensively addressed. The ArAlEval[1] shared Task 2: Disinformation Detection (Hasanain et al., 2023) aims to encourage further exploration of disinformation detection in Arabic content. It includes two sub-tasks: (A) to categorize whether a given tweet is disinformative, modelled as a binary classification task, and (B) detecting the fine-grained disinformation class for a tweet, modelled as a multiclass classification task with labels indicating the subtype of disinformation contained - hate speech, offensive, rumour or spam.

BERT-based (Devlin et al., 2018) models have been shown to be successful in understanding the context behind language and benefit from being able to transfer learned knowledge to various tasks. Due to these advantages, such models are very good in text classification tasks. We have, therefore, utilised BERT-based models for the shared task which has been pre-trained over Arabic text. We hypothesized that such pre-trained models will better understand Arabic text than a BERT model that has been pre-trained over English text. However, there are certain challenges when we are dealing with text that is code-mixed. Tweets usually contain texts that contain code-mixed text which may prove to be difficult to work with.

2 Related Work

Techniques used for disinformation detection include manually or automatically analyzing the content of a piece of information to identify features that are associated with disinformation, analyzing the social media activity around a piece of information to identify patterns that suggest it is being spread as disinformation and verifying the claims made in a piece of information using external knowledge (Hu et al., 2022a). In the domain of fake news detection, significant work has already been done which are covered by many seminal survey works on fake news detection such as (Shu et al., 2017; Oshikawa et al., 2018; Bondielli and Marcelloni, 2019; Elhadad et al., 2019; Zhou and Zafarani, 2020; Zhang and Ghorbani, 2020; Mridha et al., 2021; Hu et al., 2022b). Given that the surveys encompass research endeavors concerning fake news, encompassing both misinformation and disinformation detection, we shall employ these terms interchangeably within this section.

[1] https://gitlab.com/araieval/wanlp2023_araieval

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 570–575
December 7, 2023 ©2023 Association for Computational Linguistics

However, with the advent of transformer models (Vaswani et al., 2017), the prospect of training neural network models on languages beyond English has become increasingly prominent. The application of transformer models to Arabic text is highly promising. These models, trained on extensive text data, can excel in Arabic NLP tasks. They can grasp sentiment nuances, crucial for sentiment analysis, and enhance translation accuracy in challenging Arabic-English translation tasks. For information retrieval, understanding Arabic query and document semantics is vital, where transformers show exceptional performance. This advancement has significantly improved Arabic NLP across domains. Consequently, a plethora of research studies focusing on Arabic fake news detection has emerged, many of which have been reviewed in prominent surveys like those by (Fouad et al., 2022; Nassif et al., 2022; Harrag and Djahli, 2022; Al-Yahya et al., 2021). Other related works also include shared task results on propaganda (Alam et al., 2022) and detection and reasoning of tweets (Mubarak et al., 2023).

The Covid-19 pandemic also led to a range of research works focusing on Arabic misinformation regarding the pandemic. The work by (Haouari et al., 2020) introduces a dataset for misinformation detection, covering various topical categories influenced by COVID-19, and presents benchmarking results for tweet-level verification. (Al-Rawi et al., 2022) examines the scale of Arabic COVID-19 disinformation, identifying prominent topics related to violations of civil liberties, vaccine-related conspiracies, and calls for action. (Ashraf et al., 2022) presents a machine learning-based system for detecting misinformation in Arabic tweets related to COVID-19 vaccination, achieving promising performance. The work by (Obeidat et al., 2022) introduces a comprehensive dataset annotated with fine-grained misinformation classes and situational information, and presents baseline results using various classifiers.

In contrast to the aforementioned work, this system paper investigates the effectiveness of a fine-tuned BERT model in binary and multi-class classification of disinformation work, thereby capturing a broader aspect of disinformation regarding Arabic Twitter data.

3 Data

For both the sub-tasks, we used the training and development sets for the competition since the test set labels were part of the competition. However, after the competition the test set labels were also released which is why in this paper we are including the details of the whole dataset for both the sub-tasks. For subtask 2A, the details of the dataset are shown in Table 1.

Dataset Details	disinfo	no-disinfo
train	2656	11491
dev	397	1718
test	876	2853
Overall	3929	16062

Table 1: Dataset details for subtask 2A

The columns disinfo and no-disinfo are the labels for the subtask where disinfo means having disinformation and no-disinfo means having no disinformation.

For the subtask 2B, the details of the dataset is shown in Table 2.

Dataset Details	HS	SPAM	OFF	Rumor
train	1512	453	500	191
dev	226	68	75	28
test	442	241	160	33
Overall	2180	762	735	252

Table 2: Dataset details for subtask 2B

Before training, we performed some preprocessing of the text using the Python RegEx[2] library as well as removal of NaN entries. The preprocessing steps include the removal of punctuation including symbols that includes both English and Arabic punctuation. We also normalized certain Arabic symbols, removal of repeating characters and hashtags, URLs and mentions.

4 System

In this section,we will first describe the system architecture and then discuss the implementation of the system.

4.1 System architecture

Our system is built upon the foundation of the AraBERTv0.2-Twitter model[3], which is an Arabic language model pre-trained on Arabic twitter

[2] https://docs.python.org/3/library/re.html

[3] https://huggingface.co/aubmindlab/bert-base-arabertv02-twitter

specific text corpus, as described in (Antoun et al., 2021). The base model is a BERT model which has been pre-trained with Arabic text. To adapt this model to our specific tasks, we employ the model after extensive data preprocessing. All our experiments are conducted using the Huggingface framework (Wolf et al., 2020).

4.2 System implementation

Our workflow begins with text tokenization using the model's tokenizer. This crucial step breaks down the input text into its constituent tokens, ensuring compatibility with the model's architecture. Subsequently, we generate text embeddings using the model, creating numerical representations of the input text. For the training phase, we carefully select hyperparameters to optimize model performance. These hyperparameters include a maximum input length of 64 tokens, a batch size of 16, and a training duration of 20 epochs. The choice of a suitable learning rate is pivotal in fine-tuning, and we set it at 2e-5. To optimize model weights, we employ the Adam optimizer with an epsilon value of 1e-08 and a weight decay of 0.01. Throughout the training process, we periodically save model checkpoints, specifically after every 200 steps. This strategy allows us to monitor the model's progress and select the best-performing version based on accuracy for our subsequent tasks. Importantly, these hyperparameter settings draw inspiration from established works in the field, notably the work by (Devlin et al., 2018) and (Antoun et al., 2021). To effectively handle both subtask tasks, we employ different loss functions. For subtask 2A, a binary cross-entropy loss is used, as it aligns with the binary nature of this classification problem. In contrast, subtask 2B involves multi-class classification, thus necessitating the use of categorical cross-entropy loss. These choices of loss functions are made to suit the specific requirements and nature of each subtask.

5 Results

Since the test file with gold standard annotations was available for both sub-tasks, we have evaluated our model on this file. The results for both the sub-tasks are shown below in Table 3 and Table 4. For comparison, we have used the baseline approaches provided for the task. We also compared various other transformer based models with our system model for a fair comaprison. The baseline transformer models used are as follows: BERT-base (Devlin et al., 2018), XLM-RoBERTa-base (Conneau et al., 2019), RoBERTa-base (Liu et al., 2019), multlingual-BERT-base (Devlin et al., 2018) and CamelBERT (Inoue et al., 2021). To provide context, it's important to note that BERT-base and RoBERTa-base are transformer models pre-trained on English text. In contrast, XLM-RoBERTa-base and mBERT have undergone training on multilingual text, making them suitable for a broader range of languages. Lastly, the CamelBERT model has been pre-trained on Arabic text, rendering it particularly well-suited for the specific tasks this paper addresses. The performance comparison, as illustrated in the table, unequivocally underscores the superior capabilities of our system. Across both sub-tasks, our system consistently outperformed the baseline approaches as well as the other transformer-based models. We also report the top performing team results for a fair comparison with our results.

These results substantiate the efficacy of our approach, highlighting its robustness and suitability for the given tasks. The superior performance of our system showcases the importance of specialized pre-trained models, in enhancing the accuracy and effectiveness of domain specific natural language processing tasks.

Model	**Macro F-1**	**Micro F-1**
BERT-base-uncased	0.7921	0.8278
RoBERTa-base	0.4939	0.7758
XLM-RoBERTa-base	0.7618	0.8404
BERT-base-multilingual-uncased	0.8013	0.8696
BERT-base-arabic-camelbert-mix	0.8428	0.8924
Task Baseline (Random)	0.4763	0.5154
Task Baseline (Majority)	0.4335	0.7651
Top Team (DetectiveRedasers)	**0.8626**	**0.9048**
Our system	0.8595	0.9021

Table 3: Macro and micro f-1 comparison for subtask 2A

Model	**Macro F-1**	**Micro F-1**
BERT-base-uncased	0.4856	0.7271
RoBERTa-base	0.3905	0.6872
XLM-RoBERTa-base	0.4287	0.7431
BERT-base-multilingual-uncased	0.6303	0.7659
BERT-base-arabic-camelbert-mix	0.6809	0.8002
Task Baseline (Random)	0.2243	0.2603
Task Baseline (Majority)	0.1677	0.5046
Top Team (DetectiveRedasers)	**0.7541**	**0.8356**
Our system	0.7209	0.8174

Table 4: Macro and micro f-1 comparison for subtask 2B

6 Discussion

In this section, we delve into the discussion of the results obtained from our experiments with various BERT-based models. Our results, as illustrated in Table 3 and Table 4, offer valuable insights into the effectiveness of different pre-trained BERT-based models. Notably, we observed that BERT-based models specifically pre-trained on Arabic text consistently outperformed their generic and multilingual counterparts. This observation underscores the importance of leveraging language-specific pre-trained models when working with Arabic language data. Furthermore, our experiments revealed an intriguing finding regarding the role of training data sources. Specifically, we noted that a BERT model pre-trained on Arabic Twitter data exhibited superior performance compared to models trained on more general Arabic text. This outcome suggests that the unique characteristics of Twitter data, such as the distinctive writing style shaped by the platform's character limitations, can be harnessed to enhance the performance of NLP models for tasks involving Twitter content. It is worth highlighting that while the CamelBERT model has been trained on Arabic text, the Twitter-specific Arabic BERT model that we opted for our work showed better performance. This preference demonstrates that, even within the domain of Arabic language, domain-specific pre-trained models can offer advantages over more generalized alternatives. In essence, our findings emphasize the significance of tailoring pre-training data to the specific characteristics and requirements of the target task.

6.1 Ablation Study

As part of our discussion, we also did an ablation study wherein we experimented with our model for the multi-class task by dropping some of the classes. Based on the class instances, we first drop the Rumor class since it has the least number of instances across the train, dev and test sets. We then proceeded with the same experiments whose details are presented in the Table 5.

Comparing Table 5 with Table 4, we can see that there is an increase in the macro as well as micro f-1 scores across all the models. One reason for this could be the class imbalance in the dataset. Across the whole dataset, the rumor class has the lowest number of instances. Therefore, removing those instances may lead to a more balanced dataset thereby increasing the model performance. However, in order to verify this, we experimented by keeping the Rumor class and dropping a different class, SPAM which has a higher number of instances than Rumor and has a similar number of instances with the OFF class. The results of this experiment is shown in Table 6.

Model	Macro f-1	Micro f-1
BERT-base-uncased	0.6459	0.7663
RoBERTa-base	0.4963	0.6856
XLM-roBERTa-base	0.5811	0.7746
BERT-base-multilingual-uncased	0.7150	0.7781
BERT-base-arabic-camelbert-mix	0.7331	0.8173
Our system	0.7926	0.8505

Table 5: Results without the Rumor class

Model	Macro f-1	Micro f-1
BERT-base-uncased	0.3237	0.6881
RoBERTa-base	0.2736	0.6960
XLM-roBERTa-base	0.2823	0.6992
BERT-base-multilingual-uncased	0.5424	0.7102
BERT-base-arabic-camelbert-mix	0.5604	0.7370
Our system	0.6373	0.7574

Table 6: Results without the SPAM class

We can see from Table 6 that dropping the SPAM class and keeping the Rumor class leads to a decrease in model performance across all models for both macro as well as micro f-1. This shows that there is an imbalance in the dataset with the low instances of the Rumor class. In order to mitigate this issue, one way would be to increase the number of instances while data collection and the other would be to make use of data augmentation synthetically and append the new synthetic data to the dataset. However, although the data augmentation seems like a viable option without having to collect new data, further research is required in order to find suitable augmentation methods that can improve the performance of the model without generating noise and bias.

7 Conclusion

In this study, we have presented a comprehensive analysis of our system's performance in addressing the task of Arabic disinformation. The results of our evaluation unequivocally illustrate the superiority of our system over various baseline approaches, including those based on generic and multilingual transformer models. Notably, our system's outstanding performance in both sub-tasks underscores the significance of such language-specific pre-trained models in enhancing the precision and utility of natural language processing applications.

Furthermore, the superiority of our system, even when compared to CamelBERT, a model pre-trained on Arabic text, highlights the importance of considering the specific nuances of data sources. In our case, a pre-trained model on Twitter-specific Arabic text data proved to be an advantageous choice, particularly for tasks involving Twitter data, where the writing style is distinct due to character limitations.

References

Ahmed Al-Rawi, Abdelrahman Fakida, Kelly Grounds, et al. 2022. Investigation of covid-19 misinformation in arabic on twitter: Content analysis. *Jmir Infodemiology*, 2(2):e37007.

Maha Al-Yahya, Hend Al-Khalifa, Heyam Al-Baity, Duaa AlSaeed, and Amr Essam. 2021. Arabic fake news detection: comparative study of neural networks and transformer-based approaches. *Complexity*, 2021:1–10.

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, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Hunt Allcott and Matthew Gentzkow. 2017. Social media and fake news in the 2016 election. *Journal of economic perspectives*, 31(2):211–236.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2021. Arabert: Transformer-based model for arabic language understanding.

Nsrin Ashraf, Hamada Nayel, and Mohamed Taha. 2022. Misinformation detection in arabic tweets: A case study about covid-19 vaccination. *Benha Journal of Applied Sciences*, 7(5):265–268.

Alessandro Bondielli and Francesco Marcelloni. 2019. A survey on fake news and rumour detection techniques. *Information Sciences*, 497:38–55.

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

Madeleine de Cock Buning. 2018. *A multi-dimensional approach to disinformation: Report of the independent high level group on fake news and online disinformation*. Publications Office of the European Union.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Mohamed K Elhadad, Kin Fun Li, and Fayez Gebali. 2019. Fake news detection on social media: a systematic survey. In *2019 IEEE Pacific Rim conference on communications, computers and signal processing (PACRIM)*, pages 1–8. IEEE.

Khaled M Fouad, Sahar F Sabbeh, and Walaa Medhat. 2022. Arabic fake news detection using deep learning. *Computers, Materials & Continua*, 71(2).

Fatima Haouari, Maram Hasanain, Reem Suwaileh, and Tamer Elsayed. 2020. Arcov19-rumors: Arabic covid-19 twitter dataset for misinformation detection. *arXiv preprint arXiv:2010.08768*.

Fouzi Harrag and Mohamed Khalil Djahli. 2022. Arabic fake news detection: A fact checking based deep learning approach. *Transactions on Asian and Low-Resource Language Information Processing*, 21(4):1–34.

Maram Hasanain, Firoj Alam, Hamdy Mubarak, Samir Abdaljalil, Wajdi Zaghouani, Preslav Nakov, Giovanni Da San Martino, and Abed Alhakim Freihat. 2023. ArAIEval Shared Task: Persuasion Techniques and Disinformation Detection in Arabic Text. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore. Association for Computational Linguistics.

Linmei Hu, Siqi Wei, Ziwang Zhao, and Bin Wu. 2022a. Deep learning for fake news detection: A comprehensive survey. *AI Open*, 3:133–155.

LinMei Hu, SiQi Wei, Ziwang Zhao, and Bin Wu. 2022b. Deep learning for fake news detection: A comprehensive survey. *AI Open*.

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.

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

Muhammad F Mridha, Ashfia Jannat Keya, Md Abdul Hamid, Muhammad Mostafa Monowar, and Md Saifur Rahman. 2021. A comprehensive review on fake news detection with deep learning. *IEEE Access*, 9:156151–156170.

Hamdy Mubarak, Samir Abdaljalil, Azza Nassar, and Firoj Alam. 2023. Detecting and reasoning of deleted tweets before they are posted. *arXiv preprint arXiv:2305.04927*.

Ali Bou Nassif, Ashraf Elnagar, Omar Elgendy, and Yaman Afadar. 2022. Arabic fake news detection based on deep contextualized embedding models. *Neural Computing and Applications*, 34(18):16019–16032.

Rasha Obeidat, Maram Gharaibeh, Malak Abdullah, and Yara Alharahsheh. 2022. Multi-label multi-class covid-19 arabic twitter dataset with fine-grained misinformation and situational information annotations. *PeerJ Computer Science*, 8:e1151.

Ray Oshikawa, Jing Qian, and William Yang Wang. 2018. A survey on natural language processing for fake news detection. *arXiv preprint arXiv:1811.00770*.

Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake news detection on social media: A data mining perspective. *ACM SIGKDD explorations newsletter*, 19(1):22–36.

Edson C Tandoc Jr, Zheng Wei Lim, and Richard Ling. 2018. Defining "fake news" a typology of scholarly definitions. *Digital journalism*, 6(2):137–153.

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

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Xichen Zhang and Ali A Ghorbani. 2020. An overview of online fake news: Characterization, detection, and discussion. *Information Processing & Management*, 57(2):102025.

Xinyi Zhou and Reza Zafarani. 2020. A survey of fake news: Fundamental theories, detection methods, and opportunities. *ACM Computing Surveys (CSUR)*, 53(5):1–40.

Nexus at ArAIEval Shared Task: Fine-Tuning Arabic Language Models for Propaganda and Disinformation Detection

Yunze Xiao[1], **Firoj Alam**[2]
[1]Carnegie Mellon University in Qatar, Doha, Qatar
[2]Qatar Computing Research Institute, HBKU, Doha, Qatar
yunzex@andrew.cmu.edu,falam@hbku.edu.qa,

Abstract

The spread of disinformation and propagandistic content poses a threat to societal harmony, undermining informed decision-making and trust in reliable sources. Online platforms often serve as breeding grounds for such content, and malicious actors exploit the vulnerabilities of audiences to shape public opinion. Although there have been research efforts aimed at the automatic identification of disinformation and propaganda in social media content, there remain challenges in terms of performance. The ArAIEval shared task aims to further research on these particular issues within the context of the Arabic language. In this paper, we discuss our participation in these shared tasks. We competed in subtasks 1A and 2A, where our submitted system secured positions 9th and 10th, respectively. Our experiments consist of fine-tuning transformer models and using zero- and few-shot learning with GPT-4.

1 Introduction

In various communication channels, propaganda, also known as persuasive techniques, is disseminated through a wide set of methods. These techniques can range from appealing to the audience's emotions—known as the *"emotional technique"* — to employing logical fallacies. Examples of such fallacies include *"straw man"* arguments, which misrepresent someone's opinion; covert *"ad hominem"* attacks; and *"red herrings"*, which introduce irrelevant data to divert attention from the issue at hand (Miller, 1939).

Previous research in this area has taken various approaches to identify propagandistic content. These include assessing content based on writing style and readability levels in articles (Rashkin et al., 2017; Barrón-Cedeno et al., 2019), examining sentences and specific fragments within news articles using fine-grained techniques (Da San Martino et al., 2019), as well as evaluating memes for propagandistic elements (Dimitrov et al., 2021a).

Propagandistic text:
مقطع فيديو جديد يظهر محاولة الزميلة #شيرين_أبو_عاقلة الاحتماء من رصاص الاحتلال الإسرائيلي بالجدار والشجرة في موقعها قبيل اغتيالها\n#الأخبار LINK
Translation: A new video clip shows the attempt of colleague #Sherine_Abu_Aqla to take refuge from the bullets of the Israeli occupation using a wall and a tree at her location before her assassination\n#News LINK

Disinformative (hate speech) text:
@NourOusama@ therachellekayr البابا تبعكم فيه كورونا وعم ينشر الكورونا في لبنان عن طريق المسيحيين اللبنانيين الراجعين من روما والي ماتوا كلهم يعنى كلهم منكم مش من مناطق الشيعه
Translation: @therachellekayr @NourOusama Your Pope has Corona and is spreading Corona in Lebanon through the Lebanese Christians returning from Rome and all the people who died, I mean all of them, are from your group, they are not from the Shia area

Figure 1: Examples of propagandistic and disinformative text.

Moreover, malicious actors manipulate media platforms to shape public opinion, disseminate hate speech, target individuals' subconscious minds, spread offensive content, and fabricate falsehoods, among other. These efforts are part of broader strategies to influence people's thoughts and actions (Zhou et al., 2016; Alam et al., 2022a; Sharma et al., 2022).

In a broader context, the proliferation of such disinformation can pose significant threats to societal harmony and undermine the trust individuals have in reliable sources (Mubarak et al., 2023). Currently, these manipulative strategies are widespread across various online platforms, where they are employed to influence public opinion and distort perceptions, taking advantage of the vulnerabilities of unsuspecting audiences (Oshikawa et al., 2018, 2020).

The far-reaching consequences of misinformation and propaganda include the incitement of prejudices and discriminatory behaviors, as well as the exacerbation of social divisions and polarization (Fortuna and Nunes, 2018; Zampieri et al., 2019, 2020; Da San Martino et al., 2019). In extreme cases, such false narratives can even fuel radicalization, threatening societal stability. Ultimately, the spread of misinformation undermines democracy

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 576–582
December 7, 2023 ©2023 Association for Computational Linguistics

by depriving citizens of the accurate information needed for informed decision-making (Li et al., 2016). The digital age has expanded the reach of propaganda, subtly influencing individuals' perspectives even in their most private spheres.

Since propaganda can manifest in a variety of forms, detecting it and other types of misinformation has always been a challenging task. This task necessitates a deeper analysis of the context in which the content is presented. Therefore, the goal of the shared task is to advance research by developing methods and algorithms for identifying disinformation and propagandistic content. In Figure 1, we provide examples that depict such content.

In the ArAIEval shared task at ArabicNLP 2023 (Hasanain et al., 2023a), there are two tasks with two subtasks each: *(i)* **Task 1 Persuasion Technique Detection** and *(ii)* **Task 2: Disinformation Detection**. Each has two subtasks. We used pre-trained transformer-based models to fine-tune them on the task specific datasets.

We participated in subtasks 1A and 2A, where we fine-tuned pretrained models to predict whether the texts contain persuasion techniques (1A) or are disinformative (2A). We also explored zero-shot and few-shot learning using GPT-4 to understand its performance for these tasks. Both subtasks in which we participated fall under binary classification settings.

2 Related Work

In this section, we discuss the research related to the automatic detection of persuasion techniques and disinformation.

Over the past few decades, the use of persuasion techniques, often in the form of propaganda, has proliferated on social media platforms, aiming to influence or mislead audiences. This has become a major concern for a wide range of stakeholders, including social media companies and government agencies. In response to this growing issue, the emerging field of "computational propaganda" aims to automatically identify such manipulative techniques across various forms of content—textual, visual, and multimodal (e.g., memes).

Recently, the study by (Da San Martino et al., 2019) curated a variety of persuasive techniques. These range from emotional manipulations, such as using *Loaded Language* and *Appeal to Fear*, to logical fallacies like *Straw Man* (misrepresenting someone's opinion) and *Red Herring* (introducing irrelevant data). The study primarily focused on textual content, such as newspaper articles. In a similar vein, (Da San Martino et al., 2020) organized a shared task on the "Detection of Propaganda Techniques in News Articles." Building on these previous efforts, (Dimitrov et al., 2021b)[1] orchestrated the *SemEval-2021 Shared Task 6 on Detection of Propaganda Techniques in Memes* in 2021. This task had a multimodal setup, integrating both text and images, and challenged participants to construct systems capable of identifying the propaganda techniques employed in specific memes. Efforts have also been made towards multilingual propaganda detection. (Hasanain et al., 2023b) demonstrates that multilingual models significantly outperform monolingual ones, even in languages that are unseen.

While most of these efforts have focused primarily on English, Alam et al. (2022b) organized a shared task on fine-grained propaganda techniques in Arabic to enrich the field of Arabic AI research. This event attracted numerous participants.

In addition to the use of propaganda, malicious social media users frequently disseminate disinformative content—including hate speech, offensive material, rumors, and spam—to advance social and political agendas or to harm individuals, entities, and organizations. To address this issue, the current literature has explored automated techniques for detecting disinformation on social media platforms. For example, the study by Demilie and Salau (2022) investigated the detection of fake news and hate speech in Ethiopian social media. The researchers found that a hybrid approach, combining both deep learning and traditional machine learning techniques, proved to be the most effective in identifying disinformation in that context.

In the field of Arabic social media, numerous researchers have used various approaches for disinformation detection. For example, the study by Boulouard et al. (2022) focused on identifying hate speech and offensive content in Arabic social media platforms. By employing transfer learning techniques, they found that BERT (Devlin et al., 2018) and AraBERT (Antoun et al., 2020) yielded the highest accuracy rates, at 98% and 96%, respectively. Other significant contributions to the area

[1] http://propaganda.math.unipd.it/semeval2021task6/

of Arabic hate speech and offensive content detection include works by Zampieri et al. (2020) and Mubarak et al. (2020).

3 Task and Dataset

As discussed earlier we used the datasets released as a part of the ArAIEval shared task (Hasanain et al., 2023a). We participated in subtask 1A and 2A. They are defined as follows.

Subtask 1A: Given a multigenre (tweet and news paragraphs of the news articles) snippet, identify whether it contains content with persuasion technique. This is a binary classification task.

The data for Subtask 1A is composed of IDs, text, and labels. These labels are either ‘true’ or ‘false’, indicating whether the content contains a propagandistic technique. As observed in our analysis, there is a significant skew in the label distribution. As shown in Table 1, only 21% of the data is labeled as ‘false,’ while the remaining 79% carries a ‘true’ label. This imbalance in classes could introduce challenges during the training phase. Furthermore, we found that 64.9% of the data originates from paragraphs, while the remaining 35.1% is sourced from tweets.

Subtask 2A: Given a tweet, categorize whether it is disinformative. This is a binary classification task.

The data format for Subtask 2A is identical to that of Subtask 1A. Similar to Subtask 1A, this subtask also shows a skewed label distribution. Specifically, only 18.8% of the data is tagged as **disinfo**, while the remaining 79% carries the **no-disinfo** tag, as can be seen in Table 1. This imbalance in class distribution could present challenges during the model training process.

For our experiments, we used the same training, development, and test datasets as provided by the organizers. Details on the data distribution can be found in Table 1.

Evaluation Measures: The official evaluation metric for Subtask A is Micro-F1, while for Subtask B, it is Macro-F1.

4 Methodology

4.1 Pre-trained Models

Given that large-scale pre-trained Transformer models have achieved state-of-the-art performance for several NLP tasks. Therefore, as deep learning algorithms, we used deep contextualized text representations based on such pre-trained transformer models. We used AraBERT (Antoun et al., 2020), MarBERT (Abdul-Mageed et al., 2021) and Qarib (Abdelali et al., 2021) due to their promising performance in other Arabic NLP tasks.

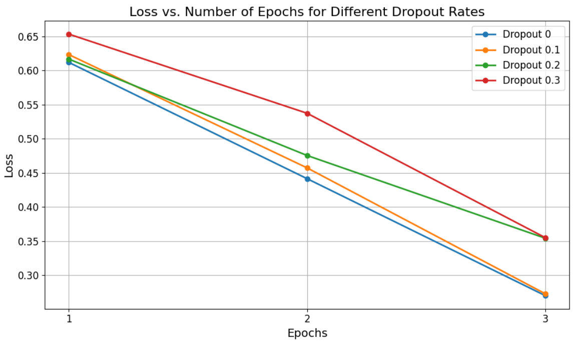


Figure 2: Loss per epoch with different dropout rate.

Consequently, text preprocessing was done using the AraBERT preprocessor with the default configuration. Hyperparameters were tuned and optimized through the use of randomized grid search. The chosen configuration for the task involved a maximum tokenization length of 128, a batch size of 16, running for a total of 3 epochs during training, with a learning rate set at 4e-5, and utilizing the AdamW optimizer. As a loss function, we used cross-entropy loss:

$$\text{CrossEntropyLoss} = -\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij} \cdot \log(p_{ij})$$

where, N is the number of samples, C is the number of classes, y_{ij} is the ground truth label (1 if the sample i belongs to class j, 0 otherwise), and p_{ij} is the predicted probability of sample i belonging to class j.

After closely examining the weights in the cross-entropy loss function, we chose to assign four times the weight to the ‘false’ tag compared to the ‘true’ tag, resulting in a weight array of [1.0, 4.0] for the cross-entropy loss.

Additionally, we observed that the dataset is highly imbalanced. Incorporating a dropout layer improved the model’s performance. To optimize this, we experimented with varying dropout rates and monitored the corresponding loss across different epochs, as illustrated in Figure 2.

Surprisingly, the models with lower dropout rates, which exhibited lower loss in the final epoch,

	Task 1A		**Task 2A**	
	Prop	**Non-Prop**	**Disinfo**	**No-Disinfo**
Train	1,918 (79%)	509 (21%)	2,656 (19.8%)	11,491 (81.2%)
Dev	202 (78%)	57 (22%)	397 (18.8%)	1,718 (81.2%)
Test	331 (65.8%)	172 (34.2%)	876 (23.8%)	2,853 (76.2%)
Total	2451	733	3929	15062

Table 1: Class label distribution for task 1A and 2A. Prop. – Contains propagandistic technique; Non-Prop – does not contain any propagandistic technique.

performed worse than those with slightly higher dropout rates. We suspect that the models may have overfitted when using lower dropout rates, resulting in subpar performance on the test set.

4.2 Large Language Models (LLMs)

For the LLMs, we investigate their performance in both in-context zero-shot and few-shot learning settings. This involves prompting and post-processing the output to extract the expected content. We utilized GPT-4 (OpenAI, 2023) in both zero-shot and few-shot settings for both subtasks. To ensure reproducibility, we set the temperature to zero for all settings. Note that for GPT-4, we used version 0314, which was released in June 2023. Our choice of this model was based on its accessibility. For the experiments, we employed the LLMeBench framework (Dalvi et al., 2023), following the prompts and instructions previously studied for Arabic in (Abdelali et al., 2023).

Model	**Dropout**	**Micro F1**		**Macro F1**	
		Dev	**Test**	**Dev**	**Test**
Submission			0.740		0.693
AraBERT	0	0.656	0.625	0.723	0.712
	0.1	0.772	0.704	0.725	0.714
	0.2	0.772	0.692	0.739	0.740
	0.3	n/a	n/a	0.743	0.713
MarBERT	0	0.810	**0.756**	0.707	0.696
	0.1	0.841	0.731	0.745	0.718
	0.2	0.818	0.746	0.769	0.731
	0.3	n/a	n/a	0.737	0.708

Table 2: Results with different dropout rates and submitted system for subtask 1A. n/a refers to the number was not ready at time of preparing the paper.

Model	**Dropout**	**Test**	
		Micro F1	**Macro F1**
Submission	0.2	0.893	0.845
Qarib	0	0.889	0.822
	0.1	0.898	0.844
	0.2	0.903	**0.869**
	0.3	0.897	0.849
MarBERT	0.1	0.898	0.843
	0.2	0.898	0.846
	0.3	0.899	0.849
AraBERT	0	0.802	0.794
	0.1	0.846	0.813
	0.2	0.893	0.846

Table 3: Model performance with different dropout rates and submitted system for subtask 2A (disinformative vs. not-disinformative).

5 Results and Discussion

5.1 Subtask 1A

For this shared task, we were given a dataset containing 504 text entries. We employed the model described in the previous section to predict various labels for each tweet. The final results released by the task organizers indicated that our model achieved a Micro F1 of 0.740 and a Macro F1 of 0.693. In Table 2, we present the performance metrics for our submitted system, comparing them with other models and various dropout rates.

Through our discovery, we realize that MarBERT performed extremely well compared to Arabert. This is expected as MarBERT is trained on tweets, which is very similar to the data provided. Nevertheless, we found it even more surprising that MarBERT's performance dropped after applying the dropout layer. This potentially indicates that the model might be undertrained and we might need to run a few more epochs.

	Shot	Micro F1	Macro F1
Task 1A	0-shot	0.600	0.600
	5-shot	0.614	0.614
Task 2A	0-shot	0.759	0.707
	5-shot	0.852	0.804

Table 4: Results on the test set with zero- and few-shot learning using GPT-4.

5.2 Subtask 2A

For this shared task, we are provided with 3729 entries of text. The model described in the previous section was used to predict various labels for each tweet. The final results released by the task organizers have shown that the model that we have scored 0.7396 in Micro F1 and 0.74 in Macro F1. In Table 3 we have displayed some of our attempts, and after more experiments we are able to achieve higher result.

We noticed that in task2A that qarib outperformed MarBERT, despite both trained using a variety of tweets. This could be the result of better/bigger training set or the result of longer training duration. To discover why, further investigation and experimentation have to be made.

In Table 4, we report the results on the test sets for both tasks with zero and 5-shots learning using GPT-4. It appears that the performances are significantly lower than fine-tuned models. We see an improvement with 5-shots, which was also observed in prior studies (Abdelali et al., 2023). However, such performances are still lower than fine-tuned models. Further studies are required to understand their capabilities as prompt engineering is the key factor to achive a desired results with LLMs.

6 Conclusion and Future work

In this paper, we report on our participation in the ArAIEval 2023 shared task, which focuses on propaganda and disinformation detection. We experimented with various transformer-based models and fine-tuned them for our specific tasks. Despite challenges such as imbalanced data, we optimized our models and achieved commendable results. Our submitted system ranked 9th and 10th in subtasks 1A and 2A, respectively, on the leaderboard. In the future, our research will take advantage of the latest Large Language Models (LLMs) such as Llama, Alpaca, Bloom and more. We plan to do more experiment with data augmentation.

Limitations

Our study primarily focused on fine-tuned transformer-based models and zero-shot and few-shot learning with GPT-4. Given that the dataset is heavily skewed towards certain classes, our study did not address these aspects. However, this will be the focus of a future study.

References

Ahmed Abdelali, Sabit Hassan, Hamdy Mubarak, Kareem Darwish, and Younes Samih. 2021. Pre-training bert on arabic tweets: Practical considerations. *arXiv preprint arXiv:2102.10684*.

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. 2023. Benchmarking arabic ai with large language models.

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, Stefano Cresci, Tanmoy Chakraborty, Fabrizio Silvestri, Dimiter Dimitrov, Giovanni Da San Martino, Shaden Shaar, Hamed Firooz, and Preslav Nakov. 2022a. A survey on multimodal disinformation detection. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6625–6643, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Firoj Alam, Hamdy Mubarak, Wajdi Zaghouani, Preslav Nakov, and Giovanni Da San Martino. 2022b. Overview of the WANLP 2022 shared task on propaganda detection in Arabic. In *Proceedings of the Seventh Arabic Natural Language Processing Workshop*, 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 Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15.

Alberto Barrón-Cedeno, Israa Jaradat, Giovanni Da San Martino, and Preslav Nakov. 2019. Proppy: Organizing the news based on their propagandistic content. *In IPM*, 56(5):1849–1864.

Zakaria Boulouard, Mariya Ouaissa, Mariyam Ouaissa, Moez Krichen, Mutiq Almutiq, and Gasmi Karim. 2022. Detecting hateful and offensive speech in arabic social media using transfer learning. *Applied Sciences*, 12:12823.

Giovanni Da San Martino, Alberto Barrón-Cedeno, Henning Wachsmuth, Rostislav Petrov, and Preslav Nakov. 2020. SemEval-2020 task 11: Detection of propaganda techniques in news articles. In *Proceedings of the 14th Workshop on Semantic Evaluation*, SemEval '20, pages 1377–1414.

Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeño, Rostislav Petrov, and Preslav Nakov. 2019. Fine-grained analysis of propaganda in news article. In *EMNLP-IJCNLP*, pages 5636–5646.

Fahim Dalvi, Maram Hasanain, Sabri Boughorbel, Basel Mousi, Samir Abdaljalil, Nizi Nazar, Ahmed Abdelali, Shammur Absar Chowdhury, Hamdy Mubarak, Ahmed Ali, Majd Hawasly, Nadir Durrani, and Firoj Alam. 2023. LLMeBench: A flexible framework for accelerating llms benchmarking. *arXiv:2308.04945*.

W.B. Demilie and A.O. Salau. 2022. Detection of fake news and hate speech for ethiopian languages: a systematic review of the approaches. *J Big Data*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dimitar Dimitrov, Bishr Bin Ali, Shaden Shaar, Firoj Alam, Fabrizio Silvestri, Hamed Firooz, Preslav Nakov, and Giovanni Da San Martino. 2021a. Detecting propaganda techniques in memes. In *ACL-IJCNLP*, ACL-IJCNLP '21, pages 6603–6617, Online. Association for Computational Linguistics.

Dimitar Dimitrov, Bishr Bin Ali, Shaden Shaar, Firoj Alam, Fabrizio Silvestri, Hamed Firooz, Preslav Nakov, and Giovanni Da San Martino. 2021b. Task 6 at SemEval-2021: Detection of persuasion techniques in texts and images. In *SemEval*.

Paula Fortuna and Sérgio Nunes. 2018. A survey on automatic detection of hate speech in text. *CSUR*, 51(4):1–30.

Maram Hasanain, Firoj Alam, Hamdy Mubarak, Samir Abdaljalil, Wajdi Zaghouani, Preslav Nakov, Giovanni Da San Martino, and Abed Alhakim Freihat. 2023a. ArAIEval Shared Task: Persuasion Techniques and Disinformation Detection in Arabic Text. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore. Association for Computational Linguistics.

Maram Hasanain, Ahmed El-Shangiti, Rabindra Nath Nandi, Preslav Nakov, and Firoj Alam. 2023b. QCRI at SemEval-2023 task 3: News genre, framing and persuasion techniques detection using multilingual models. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1237–1244, Toronto, Canada. Association for Computational Linguistics.

Yaliang Li, Jing Gao, Chuishi Meng, Qi Li, Lu Su, Bo Zhao, Wei Fan, and Jiawei Han. 2016. A survey on truth discovery. *ACM Sigkdd Explorations Newsletter*, 17(2):1–16.

Clyde R. Miller. 1939. The Techniques of Propaganda. pages 27–29.

Hamdy Mubarak, Samir Abdaljalil, Azza Nassar, and Firoj Alam. 2023. Detecting and reasoning of deleted tweets before they are posted. *arXiv preprint arXiv:2305.04927*.

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.

R OpenAI. 2023. Gpt-4 technical report. *arXiv*, pages 2303–08774.

Ray Oshikawa, Jing Qian, and William Yang Wang. 2018. A survey on natural language processing for fake news detection. *arXiv preprint arXiv:1811.00770*.

Ray Oshikawa, Jing Qian, and William Yang Wang. 2020. A survey on natural language processing for fake news detection. In *Proceedings of the 12th Language Resources and Evaluation Conference*, LREC '20, pages 6086–6093.

Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. 2017. Truth of varying shades: Analyzing language in fake news and political fact-checking. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2931–2937. Association for Computational Linguistics.

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.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Predicting the type and target of offensive posts in social media. In *NAACL-HLT*, pages 1415–1420.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin.

2020. SemEval-2020 task 12: Multilingual offensive language identification in social media (OffensEval 2020). In *SemEval*, pages 1425–1447.

Lu Zhou, Wenbo Wang, and Keke Chen. 2016. Tweet properly: Analyzing deleted tweets to understand and identify regrettable ones. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, page 603–612, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Frank at ArAIEval Shared Task: Arabic Persuasion and Disinformation: The Power of Pretrained Models

Dilshod Azizov[1], **Jiyong Li**[2], **Shangsong Liang**[1,*]
[1]Mohamed bin Zayed University of Artificial Intelligence
[2]Sun Yat-sen University
dilshod.azizov@mbzuai.ac.ae, lijy373@mail2.sysu.edu.cn
[*]Corresponding author, liangshangsong@gmail.com

Abstract

In this work, we present our systems developed for "ARAIEVAL" shared task of ArabicNLP 2023 (Hasanain et al., 2023a). We used an mBERT transformer for Subtask 1A, which targets persuasion in Arabic tweets, and we used the MARBERT transformer for Subtask 2A to identify disinformation in Arabic tweets. Our persuasion detection system achieved micro-F1 of **0.745** by surpassing the baseline by 13.2%, and registered a macro-F1 of 0.717 based on leaderboard scores. Similarly, our disinformation system recorded a micro-F1 of **0.816**, besting the naïve majority by 6.7%, with a macro-F1 of 0.637. Furthermore, we present our preliminary results on a variety of pre-trained models. In terms of overall ranking, our systems placed 7th out of 16 and 12th out of 17 teams for Subtasks 1A and 2A, respectively.

1 Introduction

The digital communication landscape, vast in its dynamism, constantly evolves, presenting unique challenges in diverse cultural and linguistic contexts. Arabic, with its rich historical and poetic traditions, spoken by more than 420 million people, is no exception (Qu et al., 2023). In the age of digital connectivity, platforms like Twitter have become a boon and a bane. They enable rapid dissemination of information, but also disinformation spread, which can manipulate public perceptions and cause socio-political instability (Raj and Goswami, 2020). Because tweets are so brief, accuracy is essential. This requires the use of strong rhetorical elements, making them an ideal environment for disinformation and persuasive strategies (Hasanain et al., 2023b; Hardalov et al., 2021; Nakov and Da San Martino, 2021).

Taking into account the vast diversity of Arabic dialects and cultural nuances, identifying these strategies is challenging, especially with the rise in misinformation campaigns (Dimitrov et al., 2021). The importance of addressing this issue is amplified by the geopolitical significance of Arabic-speaking regions, where digital narratives can influence diplomacy, policy decisions, and public sentiment (Al-Rawi et al., 2022; Guellil et al., 2021; Cheng et al., 2021).

In response, the ARAIEVAL shared task of ArabicNLP 2023 focuses on two critical areas:

Subtask 1A: *Given a multigenre (tweet and news paragraphs of the news articles) snippet, identify whether it contains content with persuasion technique.*

Subtask 2A: *Given a tweet, categorize whether it is disinformative (Hasanain et al., 2023a).*

Our paper offers the following contributions:

- We propose systems that use mBERT (Devlin et al., 2018) for persusaion detection and MARBERT (Abdul-Mageed et al., 2020) for disinfromation identification.
- We compare the performance of mBERT *vs.* XLM-RoBERTa (Liu et al., 2019) and MARBERT for subtask 1A. In Subtask 2A, we compare MARBERT *vs.* AraBERT(Antoun et al., 2020), and ALBERT (Lan et al., 2019).

In Section 2, we outline previous and more recent studies on the identification of persuasion and disinformation. In Section 3, we illustrate a thorough examination of the dataset. In Section 4 we describe the systems and the results. Finally, Section 5 presents our conclusion and suggests directions for future exploration.

2 Related Work

In recent years, Natural Language Processing (NLP) has experienced significant advances, particularly in detecting persuasive techniques and misinformation across various languages. Historically, English-centric models have been at the forefront, showcasing breakthroughs in understanding and

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 583–588
December 7, 2023 ©2023 Association for Computational Linguistics

	Train	Dev	Test	Total
Subtask 1A	2,120	566	503	3,189
Subtask 2A	14,147	2,115	3,739	20,002

Table 1: Statistics about distributon of tweets in train/dev/test of Subtasks 1A and 2A

auto-detecting persuasion (Haouari et al., 2020; Piskorski et al., 2023). On the contrary, Arabic, characterized by its linguistic diversity and cultural richness, has seen relatively limited focus in the domain of persuasion detection. While the foundational research of Arabic NLP revolved around sentiment analysis (Abdulla et al., 2013) and stance detection (Almiman et al., 2020; Hardalov et al., 2021), the nuanced domain of detecting persuasion techniques in Arabic remained underexplored due to the complex morphology of the language and the diverse dialects (Alam et al., 2022).

Furthermore, recent studies, such as Al-Sallab et al. (2017), have emphasized the need for advanced embeddings and specialized datasets tailored to Arabic peculiarities. An emerging trend marries traditional Arabic linguistic studies with contemporary machine learning, targeting the precise detection of persuasive techniques in Arabic content (Alam et al., 2022).

The rapid proliferation of disinformation in the digital age, especially via social media platforms, requires the inter-related studies on persuasion and misinformation studies (Peng et al., 2023). In particular, CheckThat! lab at CLEF has embarked on multifaceted research on misinformation in different languages, encompassing fact-checking, check-worthiness, bias identification, and source credibility assessment (Da San Martino et al., 2023; Azizov et al., 2023; Nakov et al., 2023; Barrón-Cedeño et al., 2023; Barrón-Cedeño et al., 2023; Elsayed et al., 2019a,b; Hasanain et al.; Barrón-Cedeño et al.; Nakov et al., 2021a,b). This is consistent with the contemporary research trends that have changed from analyzing only news articles to scrutinizing social media for propaganda detection (Woolley and Howard, 2018; Martino et al., 2020b). Interestingly, another study by (Zhang et al., 2019) proposes a Bayesian deep learning model for misinformation detection, incorporating claim responses and quantifying prediction uncertainty, achieving superior performance in public datasets.

Moreover, Da San Martino et al. (2019) delved deeply into persuasive techniques, highlighting the emotional signals that resonate with readers. This foundational work paved the way for subsequent endeavors, notably the "Detection of Propaganda Techniques in News Articles" challenge posited by (Martino et al., 2020a). Building on this momentum, a recent investigation by (Mubarak et al., 2023) sought to discern and categorize the underlying reasons for the deletion of Arabic tweets, and later designed predictive models for potential deletions. In the multimedia realm, Dimitrov et al. (2021) emphasized the importance of detecting propaganda within memes, thus underscoring the convergence of text and imagery in disinformation campaigns.

3 Data

In this section, a detailed description of the dataset released by the ARAIEVAL shared task organizers is provided. Our primary focus is on the binary classification challenge subtasks 1A and 2A persuasion and disinformation detection dataset.

Data Attributes: *Both subtasks consist of a dataset with the same structure, comprising an ID, text, and label. However, the labels differentiate between the subtasks*

- **ID**: Numerical index of the data point.
- **Tweet for Subtask 1A**: Arabic tweet potentially containing persuasion.
- **Tweet for Subtask 2A**: Arabic tweet potentially containing disinformation .
- **Label for Subtask 1A**: “True” (indicating the presence of persuasion) and “False” (indicating the absence of persuaion).
- **Label for Subtask 2A**: “Disinfo” (denoting the text as a rumor) and “No-Disinfo” (indicating the absence of disinformation).

Dataset Size:

The dataset from ARAIEVAL is detailed in Table 1. Subtask 1A consists of less than 3.2k tweets,

	Subtask 1A		Subtask 2A	
	True	False	Disinfo	No-Disinfo
Train	1,918	509	2,656	11,491
Dev	202	57	397	1,718

Table 2: Labels distribution over the train and development set in Subtasks 1A and Subtask 2A.

while Subtask 2A contains slightly more than 20k tweets. The distribution of labels within the training and development sets can be seen in Table 2. In particular, both subtasks have an imbalance distribution of the datasets.

4 System Descriptions and Results

4.1 System Descriptions

For the assessment, we used the official evaluation tools designated for the shared task. The official measure for both subtasks is micro-F1, although the macro-F1 measure is also generated by the evaluation tools. Our models training was carried out using two NVIDIA Tesla T4 GPUs, each with 16GB memory.

Subtask 1A

mBERT. We used the mBERT base architecture. Our configuration involved a batch size of 16 and a training duration of 5 epochs with a learning rate of 5e-5. Measures were logged every 500 steps. Gradient norms were reduced to a maximum value of 1.0. ADAMW optimizer was used with a weight decay of 0.01 to mitigate overfitting. Model checkpoints were saved every 500 steps and after the end of each epoch. Both the warm-up ratio and the warm-up steps were set to zero.

Subtask 2A

MARBERT. For our binary classification task, we utilized the MARBERT base architecture, which is equipped with 12 self-attention heads, has 163M parameters and an embedding dimensionality of 768. We use the Adam optimizer with a learning rate set at 5e-5. To balance computational efficiency with model convergence, we settled on a batch size of 32. Labels “no-disinfo” and “disinfo” were encoded in 0 and 1, respectively, using a *label2id* dictionary, and decoded with a *id2label* dictionary for predictions. The training was conducted over five epochs, after which the model achieved convergence without evident signs of overfitting.

Note: For both subtasks, the data set was preprocessed using the AraBERT pre-processor and tokenizer. Text inputs were standardized to a sequence length of 512 tokens through truncation and padding.

4.2 Results

In the initial stages, we experimented with the development set, as we used it as a test set, and from the train set we cut 10% out of the total tweets for the development set. All models have been trained on 3 epochs, the rest of the hyperparameters have been used as default. Below, we dive deeper into each model’s performance and postulate the reasons behind their relative successes and shortcomings.

Subtask 1A. mBERT exhibits exemplary performance in this subtask, registering the highest micro-F1 score of 0.889. Its efficiency in maintaining a balance between precision and recall is evident in its scores of 0.855 and 0.887, respectively. MARBERT closely follows with a commendable micro-F1 score of 0.881, and its precision and recall stand at 0.847 and 0.877, respectively. This suggests that while mBERT slightly edges out in terms of overall performance, MARBERT remains a strong contender. XLM-RoBERTa, although competitive, falls slightly behind with a micro-F1 score of 0.876. It has a precision score of 0.780 and a recall of 0.870, indicating that it can be more conservative in its predictions compared to the other models.

Subtask 2A. MARBERT secures the top position for this subtask with a micro-F1 score of 0.866. Its precision and recall scores are 0.856 and 0.878, respectively, indicating a balanced performance. ALBERT, with a micro-F1 score of 0.846, also shows commendable results. Its precision is slightly lower than that of MARBERT at 0.842, but it manages a recall of 0.871. mBERT has a micro-F1 score of 0.840 and exhibits similar precision and recall values of 0.840 and 0.862, respectively. This demonstrates that while MARBERT is leading in this subtask, ALBERT and mBERT remain closely competitive.

	Subtask 1A					Subtask 2A			
	Micro F1	**Accuracy**	**Precision**	**Recall**		**Micro F1**	**Accuracy**	**Precision**	**Recall**
mBERT	**0.889**	**0.853**	**0.855**	**0.887**	MARBERT	**0.866**	**0.854**	**0.856**	**0.878**
XLM-RoBERTa	0.876	0.781	0.780	0.870	mBERT	0.840	0.841	0.840	0.862
MARBERT	0.881	0.838	0.847	0.877	ALBERT	0.846	0.843	0.842	0.871

Table 3: The experimental results of various frameworks on the development sets of Subtasks 1A and 2A.

For the tasks at hand, our described configuration yielded notable results. In Subtask 1A, our model recorded a micro-F1 of 0.745 and a macro-F1 of 0.717. Meanwhile, for Subtask 2A, the corresponding scores were 0.816 and 0.637. The significant performance of the system can be attributed to judicious choice of models and meticulous fine-tuning. Such efforts positioned us competitively in the leaderboard rankings.

4.3 Analysis

Delving into the observed differences:

MARBERT: Continues it's impressive streak across both subtasks. It's architecture demonstrates finely tuned for classification tasks, but with the close competition in Subtask 2A, it shows that it can have specific strengths for different types of data.

mBERT: While shining brightly in Subtask 1A, it faces closer competition in Subtask 2A. Its strong recall figures hint at its efficiency in capturing most positive instances.

XLM-RoBERTa: Although trailing behind mBERT and MARBERT in Subtask 1A, its competitive scores show its capabilities. The drop in recall could suggest specific challenges in capturing all positive instances.

ALBERT: In Subtask 2A, its scores are quite competitive, especially given the close figures in the recall. This suggests that, while it may have precision challenges, it is quite adept at capturing positive instances.

Finally, after a comprehensive comparison analysis, we opt to integrate mBERT for the persuasion detection system and MARBERT for the disinformation system.

5 Conclusion and Future Work

In this paper, we discussed our approaches for the subtasks 1A and 2A of the shared task ARAIEVAL 2023 on persuasion and disinformation detection in Arabic. We employed the mBERT model for Subtask 1A and the MARBERT framework for Subtask 2A, and according to the official leaderboard results, our system achieved a micro-F1 of 0.745 and a macro-F1 of 0.717 for Subtask 1A, and a micro-F1 of 0.816 and macro-F1 of 0.637. We also detailed a series of experiments and made initial comparisons of our systems with various state-of-the-art frameworks.

In future work, we plan to delve into feature engineering, potentially integrating meta-features associated with the text, such as text length, unique word count, and sentiment analysis.

References

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

Nawaf A Abdulla, Nizar A Ahmed, Mohammed A Shehab, and Mahmoud Al-Ayyoub. 2013. Arabic sentiment analysis: Lexicon-based and corpus-based. In *2013 IEEE Jordan conference on applied electrical engineering and computing technologies (AEECT)*, pages 1–6. IEEE.

Ahmed Al-Rawi, Abdelrahman Fakida, Kelly Grounds, et al. 2022. Investigation of covid-19 misinformation in arabic on twitter: Content analysis. *Jmir Infodemiology*, 2(2):e37007.

Ahmad Al-Sallab, Ramy Baly, Hazem Hajj, Khaled Bashir Shaban, Wassim El-Hajj, and Gilbert Badaro. 2017. Aroma: A recursive deep learning model for opinion mining in arabic as a low resource language. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 16(4):1–20.

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, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Ali Almiman, Nada Osman, and Marwan Torki. 2020. Deep neural network approach for arabic community question answering. *Alexandria Engineering Journal*, 59(6):4427–4434.

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

Dilshod Azizov, S Liang, and P Nakov. 2023. Frank at checkthat! 2023: Detecting the political bias of news articles and news media. *Working Notes of CLEF*.

Alberto Barrón-Cedeño, Firoj Alam, Tommaso Caselli, Giovanni Da San Martino, Tamer Elsayed, Andrea Galassi, Fatima Haouari, Federico Ruggeri, Julia Maria Struß, Rabindra Nath Nandi, et al. 2023. The clef-2023 checkthat! lab: Checkworthiness, subjectivity, political bias, factuality, and authority. In *European Conference on Information Retrieval*, pages 506–517. Springer.

Alberto Barrón-Cedeño, Firoj Alam, Andrea Galassi, Giovanni Da San Martino, Preslav Nakov, , Tamer Elsayed, Dilshod Azizov, Tommaso Caselli, Gullal S. Cheema, Fatima Haouari, Maram Hasanain, Mucahid Kutlu, Chengkai Li, Federico Ruggeri, Julia Maria Struß, and Wajdi Zaghouani. 2023. Overview of the CLEF–2023 CheckThat! Lab checkworthiness, subjectivity, political bias, factuality, and authority of news articles and their source. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Fourteenth International Conference of the CLEF Association (CLEF 2023)*.

Alberto Barrón-Cedeño, Tamer Elsayed, Preslav Nakov, Giovanni Da San Martino, Maram Hasanain, Reem Suwaileh, Fatima Haouari, Nikolay Babulkov, Bayan Hamdan, Alex Nikolov, Shaden Shaar, and Zien Sheikh Ali. Overview of CheckThat! 2020: Automatic identification and verification of claims in social media. pages 215–236.

Qiao Cheng, Juntao Liu, Xiaoye Qu, Jin Zhao, Jiaqing Liang, Zhefeng Wang, Baoxing Huai, Nicholas Jing Yuan, and Yanghua Xiao. 2021. Hacred: A large-scale relation extraction dataset toward hard cases in practical applications. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2819–2831.

Giovanni Da San Martino, Firoj Alam, Maram Hasanain, Rabindra Nath Nandi, Dilshod Azizov, and Preslav Nakov. 2023. Overview of the clef-2023 checkthat! lab task 3 on political bias of news articles and news media. *Working Notes of CLEF*.

Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeno, Rostislav Petrov, and Preslav Nakov. 2019. Fine-grained analysis of propaganda in news article. 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 5636–5646.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dimitar Dimitrov, Bishr Bin Ali, Shaden Shaar, Firoj Alam, Fabrizio Silvestri, Hamed Firooz, Preslav Nakov, and Giovanni Da San Martino. 2021. Semeval-2021 task 6: detection of persuasion techniques in texts and images. *arXiv preprint arXiv:2105.09284*.

Tamer Elsayed, Preslav Nakov, Alberto Barrón-Cedeño, Maram Hasanain, Reem Suwaileh, Giovanni Da San Martino, and Pepa Atanasova. 2019a. CheckThat! at CLEF 2019: Automatic identification and verification of claims. In *Advances in Information Retrieval – European Conference on IR Research*, ECIR '19, pages 309–315.

Tamer Elsayed, Preslav Nakov, Alberto Barrón-Cedeño, Maram Hasanain, Reem Suwaileh, Giovanni Da San Martino, and Pepa Atanasova. 2019b. Overview of the CLEF-2019 CheckThat!: Automatic identification and verification of claims. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, LNCS, pages 301–321.

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

Fatima Haouari, Maram Hasanain, Reem Suwaileh, and Tamer Elsayed. 2020. Arcov19-rumors: Arabic covid-19 twitter dataset for misinformation detection. *arXiv preprint arXiv:2010.08768*.

Momchil Hardalov, Arnav Arora, Preslav Nakov, and Isabelle Augenstein. 2021. A survey on stance detection for mis-and disinformation identification. *arXiv preprint arXiv:2103.00242*.

Maram Hasanain, Firoj Alam, Hamdy Mubarak, Samir Abdaljalil, Wajdi Zaghouani, Preslav Nakov, Giovanni Da San Martino, and Abdelhakim Freihat. 2023a. Araieval shared task: Persuasion techniques and disinformation detection in arabic text. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore. Association for Computational Linguistics.

Maram Hasanain, Ahmed Oumar El-Shangiti, Rabindra Nath Nandi, Preslav Nakov, and Firoj Alam. 2023b. Qcri at semeval-2023 task 3: News genre, framing and persuasion techniques detection using multilingual models. *arXiv preprint arXiv:2305.03336*.

Maram Hasanain, Fatima Haouari, Reem Suwaileh, Zien Sheikh Ali, Bayan Hamdan, Tamer Elsayed, Alberto Barrón-Cedeño, Giovanni Da San Martino, and Preslav Nakov. Overview of CheckThat! 2020 Arabic: Automatic identification and verification of claims in social media.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut.

2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

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

G Martino, Alberto Barrón-Cedeno, Henning Wachsmuth, Rostislav Petrov, and Preslav Nakov. 2020a. Semeval-2020 task 11: Detection of propaganda techniques in news articles. *arXiv preprint arXiv:2009.02696*.

Giovanni Da San Martino, Stefano Cresci, Alberto Barrón-Cedeño, Seunghak Yu, Roberto Di Pietro, and Preslav Nakov. 2020b. A survey on computational propaganda detection. *arXiv preprint arXiv:2007.08024*.

Hamdy Mubarak, Samir Abdaljalil, Azza Nassar, and Firoj Alam. 2023. Detecting and reasoning of deleted tweets before they are posted. *arXiv preprint arXiv:2305.04927*.

Preslav Nakov, Firoj Alam, Giovanni Da San Martino, Maram Hasanain, RN Nandi, D Azizov, and P Panayotov. 2023. Overview of the clef-2023 checkthat! lab task 4 on factuality of reporting of news media. *Working Notes of CLEF*.

Preslav Nakov and Giovanni Da San Martino. 2021. Fake news, disinformation, propaganda, and media bias. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 4862–4865.

Preslav Nakov, Giovanni Da San Martino, Tamer Elsayed, Alberto Barrón-Cedeño, Rubén Míguez, Shaden Shaar, Firoj Alam, Fatima Haouari, Maram Hasanain, Nikolay Babulkov, Alex Nikolov, Gautam Kishore Shahi, Julia Maria Struß, and Thomas Mandl. 2021a. The CLEF-2021 CheckThat! lab on detecting check-worthy claims, previously fact-checked claims, and fake news. In *Advances in Information Retrieval – European Conference on IR Research*, pages 639–649.

Preslav Nakov, Giovanni Da San Martino, Tamer Elsayed, Alberto Barrón-Cedeño, Rubén Míguez, Shaden Shaar, Firoj Alam, Fatima Haouari, Maram Hasanain, Nikolay Babulkov, Alex Nikolov, Gautam Kishore Shahi, Julia Maria Struß, and Thomas Mandl. 2021b. The CLEF-2021 CheckThat! lab on detecting check-worthy claims, previously fact-checked claims, and fake news. In *Advances in Information Retrieval – 43rd European Conference on IR Research*, volume 12657 of *ECIR '21*, pages 639–649.

Wei Peng, Sue Lim, and Jingbo Meng. 2023. Persuasive strategies in online health misinformation: a systematic review. *Information, Communication & Society*, 26(11):2131–2148.

Jakub Piskorski, Nicolas Stefanovitch, Giovanni Da San Martino, and Preslav Nakov. 2023. Semeval-2023 task 3: Detecting the category, the framing, and the persuasion techniques in online news in a multilingual setup. In *Proceedings of the the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2343–2361.

Xiaoye Qu, Yingjie Gu, Qingrong Xia, Zechang Li, Zhefeng Wang, and Baoxing Huai. 2023. A survey on arabic named entity recognition: Past, recent advances, and future trends. *arXiv preprint arXiv:2302.03512*.

Adharsh Raj and Manash Pratim Goswami. 2020. Is fake news spreading more rapidly than covid-19 in india. *Journal of Content, Community and Communication*, 11(10):208–220.

Samuel C Woolley and Philip N Howard. 2018. *Computational propaganda: Political parties, politicians, and political manipulation on social media*. Oxford University Press.

Qiang Zhang, Aldo Lipani, Shangsong Liang, and Emine Yilmaz. 2019. Reply-aided detection of misinformation via bayesian deep learning. In *The world wide web conference*, pages 2333–2343.

Raphael at ArAIEval Shared Task: Understanding Persuasive Language and Tone, an LLM Approach

Utsav Shukla, Manan Vyas, Shailendra Tiwari
Thapar Institute of Engineering and Technology
{ushukla_be17, mvyas_bemba, shailendra}@thapar.edu

Abstract

The widespread dissemination of propaganda and disinformation on both social media and mainstream media platforms has become an urgent concern, attracting the interest of various stakeholders such as government bodies and social media companies. The challenge intensifies when dealing with understudied languages like Arabic. In this paper, we outline our approach for detecting persuasion techniques in Arabic tweets and news article paragraphs. We submitted our system to ArAIEval 2023 Shared Task 1, covering both subtasks. Our main contributions include utilizing GPT-3 to discern tone and potential persuasion techniques in text, exploring various base language models, and employing a multi-task learning approach for the specified subtasks.

1 Introduction

In today's world, an average person encounters a plethora of information via social media platforms and online news resources. While this accessibility to up-to-date news and opinions is convenient and keeps individuals informed, it also raises concerns about disinformation, propaganda, hate speech, and political bias. Over the past decade, various efforts have been made to detect disinformation and fake news through the analysis of textual content in articles (Wang, 2017), social media content (Lu and Li, 2020), website metrics (Panayotov et al., 2022), and images (Zlatkova et al., 2019).

While significant progress has been made in the detection of disinformation, bias, and hate speech, the automated detection of propaganda and persuasion is a relatively newer domain. Propaganda and persuasion serve as instruments to influence public opinion or evoke emotional responses and has proved to be very harmful for society in last decade. Although there is substantial work in propaganda detection in English (Da San Martino et al., 2021), Arabic remain understudied for this problem. Collecting datasets and training models for Arabic becomes more challenging because of its numerous dialects spoken all around the world.

The ArAIEval 2023 Task 1 (Hasanain et al., 2023) is a collaborative effort that aims to address this gap by detecting persuasion techniques in Arabic. Task 1 is divided into two subtasks: Subtask 1A is a binary classification task that detects the presence of a persuasion technique in a given Arabic text, while Subtask 1B is a multi-label classification problem that identifies which of the 24 possible persuasion techniques are present. In our experiments, we designed and implemented multiple systems for both subtasks. We found that leveraging outputs from large language models (LLMs) for supervised training led to significant performance gains. Our systems ranked 4th and 3rd in Subtasks 1A and 1B respectively on development sets and 5th and 6th in Subtasks 1A and 1B, respectively on test sets. In the following sections, we discuss related work and elaborate on our experiments and submissions. We have made our code, prompts, and GPT-3.5 outputs publicly available for future reproducibility [1].

2 Related Work

Disinformation and fake news detection is a vibrant area of research within the NLP community, with methodologies ranging from text-based approaches to multimodal analyses that incorporate images and graphs. Propaganda detection has also garnered attention; (Da San Martino et al., 2020) introduced a shared task that identifies both the span and type of propaganda technique present. This was extended by (Dimitrov et al., 2021), who added a sub-task focused on recognizing persuasion techniques in memes, thus incorporating image modality.

To the best of our knowledge, apart from

[1] github.com/us241098/araieval_submission

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 589–593
December 7, 2023 ©2023 Association for Computational Linguistics

Label	Training	Development
true	1918	202
false	519	517

Table 1: Label distribution for subtask 1A

Label	Training	Development
Loaded Language	1574	176
Name Calling Labeling	692	77
No Technique	509	57
Questioning the Reputation	383	43
Exaggeration Minimisation	292	33
Obfuscation Vagueness Confusion	240	28
Doubt	143	16
Causal Oversimplification	128	15
Appeal to Fear Prejudice	108	12
Slogans	70	8
Flag Waving	63	7
Appeal to Hypocrisy	56	7
Appeal to Authority	48	5
Appeal to Values	37	4
Consequential Oversimplification	33	3
False Dilemma No Choice	32	3
Conversation Killer	28	3
Repetition	25	3
Guilt by Association	13	1
Appeal to Time	10	2
Whataboutism	9	1
Red Herring	8	1
Straw Man	6	1
Appeal to Popularity	2	1

Table 2: Label distribution for subtask 1B

ArAIEval 2023 (Hasanain et al., 2023), (Alam et al., 2022) is the only other work that specifically focuses on the detection of propaganda/persuasion techniques in the Arabic language.

3 Data

Our submitted system relies solely on the dataset provided by the organizers, without any additional data or augmentations. Subtask 1A is a binary classification task featuring two labels: ’true’ and ’false,’ which signify the presence or absence of persuasion techniques in the text. Subtask 1B, a multi-label classification problem, involves 24 labels representing various potential persuasion techniques. The data for these tasks come from two sources: tweets and paragraphs from news articles. During data preprocessing, we removed emojis and the text string "LINK" from all entries. Table 1 and Table 2 describe the label distribution of both subtasks.

4 System

For both subtasks, we conducted multiple experiments that included using various base models, employing large language models (LLMs) for reasoning, and adjusting both the architecture and loss functions. We discuss these major components and their applications in the subsequent sub-sections.

4.1 MARBERT

We leverage MARBERT, a state-of-the-art BERT-based model specifically pretrained on a large corpus of Arabic text, encompassing both Modern Standard Arabic and various dialects (Abdul-Mageed et al., 2021). The utilization of MARBERT allows us to capture intricate language features that are particularly pertinent to Arabic text.

Upon passing an arabic input text through the MARBERT encoder, the resulting contextual embeddings are generated. We specifically extract the embedding corresponding to the [CLS] token. This [CLS] token’s embedding is then forwarded to binary classification head and multi label classification heads for subtask 1A and 1B respectively.

4.2 GPT 3.5

We utilize the Generative Pre-trained Transformer 3.5 (Brown et al., 2020) (GPT-3.5) for the task of generating description of Arabic texts in English and conducting tone and emotional analysis. The resultant English text and tone descriptions are subsequently encoded using either BERT (Devlin et al.,

Methodology	Micro F1	Macro F1
MARBERT	0.8145	0.7192
MARBERT+GPT 3.5(BERT)	0.8412	0.7490
MARBERT+GPT 3.5(RoBERTa)	0.8427	0.7571
MARBERT+GPT 3.5(RoBERTa)+Source as Feature Gate	0.8610	0.7922
MultiTask	0.8509	0.7698

Table 3: Results of our different systems on subtask 1A dev set

Methodology	Micro F1	Macro F1
MARBERT	0.6088	0.1996
MARBERT+GPT 3.5 (BERT)	0.6227	0.2056
MARBERT+GPT 3.5 (RoBERTa)	0.6399	0.2365
MARBERT+GPT 3.5 (RoBERTa)+Source as Feature Gate	0.6304	0.2287
MultiTask	0.5694	0.1602

Table 4: Results of our different systems on subtask 1B dev set

Task	Micro F1	Macro F1
Subtask 1A (Ours)	0.7475	0.7221
Subtask 1A (Best)	0.7634	0.7321
Subtask 1B (Ours)	0.5347	0.1772
Subtask 1B (Best)	0.5666	0.2156

Table 5: Our Submission to Subtask 1A and Subtask 1B compared to best performing systems

2019) or RoBERTa (Liu et al., 2019). When these encodings are concatenated with MARBERT encodings of original arabic texts before feeding them to respective classification heads, we observe a significant improvement over our MARBERT baseline performance.

4.3 MultiTask Training

Subtask 1A and 1B being on the same features allow us to formulate a multi task learning objective. During the forward pass [CLS] token encodings are passed through two separate fully-connected layers to produce logits for binary and multi-label classification. During backpropagation, the loss is calculated for both sub-tasks and weighted according to a learned parameter. The gradient of this total loss is then computed with respect to the model parameters. This dual-task learning enables the model to simultaneously optimize for binary and multi-label classification.

4.4 Source as Feature Gate

We use the Source provided in the datasets (Tweet or Paragraph) as feature gate for our concatenated encodings (MARBERT+BERT/RoBERTa). We have found the using the source as feature gate performs better in comparison to just concatenating the source vector to the embeddings.

5 Experiment Setting

All our experiments are done on single 12 GB GPU and our models take 5-15 minutes to be trained. We use "bert-base-cased", "roberta-base" and "MARBERTv2" variants from HuggingFace (Wolf et al., 2020) as our base models. We train our models upto 7 epochs and use AdamW optimizer (Andrew and Gao, 2007) with learning rate being set to 2e-5, and epsilon set to 1e-8.

6 Results

Table 3 and Table 4 shows our performance on subtask 1A and 1B development sets respectively. We observe that when GPT 3.5 outputs are used in training and inference we get significant gains over our MARBERT baseline in both sub-task 1A and 1B. For encoding the English outputs from GPT 3.5, RoBERTa is found to be better than BERT. We also observe that using source as feature gate give us gains in subtask 1A but not in subtask 1B. MARBERT+GPT 3.5(RoBERTa)+Source as Feature Gate is our submitted system for subtask 1A and MARBERT+GPT 3.5 (RoBERTa) is our submitted system for subtask 1B for both development and test sets. On development sets our systems ranked 4th and 3rd on subtask 1A and 1B respectively.

Table 5 shows our performance on test set. Here our submitted system in subtask 1A ranked 5th in terms of both macro and micro F1. While in subtask 1B we ranked 6th in terms of micro F1 and 4th in terms of macro F1.

7 Discussions and Limitations

Our experiments indicate that using prompts with large language models (LLMs) and leveraging their outputs as features in supervised training environments show promise, especially for understudied languages like Arabic. In the future, we plan to explore additional LLMs, with a preference for open-source options. Another exciting avenue we aim to investigate is fine-tuning these LLMs on Arabic-specific data to enhance performance. We also aim to Benchmark the only LLM performance without using their outputs for supervised models.

One limitation we've identified is the high computational/financial cost associated with closed LLM inference. However, this challenge may be mitigated as more open-source LLMs become available and as optimization techniques such as PEFT (Liu et al., 2022), QLoRA (Dettmers et al., 2023), and quantization continue to evolve.

8 Conclusion

The widespread dissemination of propaganda and misinformation through various media channels, including social media and mainstream outlets, has garnered considerable attention from key players like government agencies and social media companies. In this study, we outline our methodology for identifying persuasive tactics employed in Arabic-language tweets and text segments. For the 2023 ArAIEval shared task 1, we have used GPT-3.5 as the cornerstone of our system to analyze the tone and potential persuasion strategies in the text. We have also discussed the limitations of the system proposed and suggested to incorporate Open Source LLMs and multiple optimization techniques in our 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, Online. 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, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Galen Andrew and Jianfeng Gao. 2007. Scalable training of L_1-regularized log-linear models. In *Proceedings of the 24th International Conference on Machine Learning*, pages 33–40.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Giovanni Da San Martino, Alberto Barrón-Cedeño, Henning Wachsmuth, Rostislav Petrov, and Preslav Nakov. 2020. SemEval-2020 task 11: Detection of propaganda techniques in news articles. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1377–1414, Barcelona (online). International Committee for Computational Linguistics.

Giovanni Da San Martino, Stefano Cresci, Alberto Barrón-Cedeño, Seunghak Yu, Roberto Di Pietro, and Preslav Nakov. 2021. A survey on computational propaganda detection. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, IJCAI'20.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Dimitar Dimitrov, Bishr Bin Ali, Shaden Shaar, Firoj Alam, Fabrizio Silvestri, Hamed Firooz, Preslav Nakov, and Giovanni Da San Martino. 2021. SemEval-2021 task 6: Detection of persuasion techniques in texts and images. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 70–98, Online. Association for Computational Linguistics.

Maram Hasanain, Firoj Alam, Hamdy Mubarak, Samir Abdaljalil, Wajdi Zaghouani, Preslav Nakov, Giovanni Da San Martino, and Abed Alhakim Freihat. 2023. ArAIEval Shared Task: Persuasion Techniques and Disinformation Detection in Arabic Text. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore. Association for Computational Linguistics.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning.

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

Yi-Ju Lu and Cheng-Te Li. 2020. Gcan: Graph-aware co-attention networks for explainable fake news detection on social media.

Panayot Panayotov, Utsav Shukla, Husrev Taha Sencar, Mohamed Nabeel, and Preslav Nakov. 2022. GREENER: Graph neural networks for news media profiling. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7470–7480, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

William Yang Wang. 2017. “liar, liar pants on fire”: A new benchmark dataset for fake news detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 422–426, Vancouver, Canada. Association for Computational Linguistics.

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, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface’s transformers: State-of-the-art natural language processing.

Dimitrina Zlatkova, Preslav Nakov, and Ivan Koychev. 2019. Fact-checking meets fauxtography: Verifying claims about images. 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 2099–2108, Hong Kong, China. Association for Computational Linguistics.

Legend at ArAIEval Shared Task: Persuasion Technique Detection using a Language-Agnostic Text Representation Model

Olumide E. Ojo[1,5,a], **Olaronke O. Adebanji**[1,b], **Hiram Calvo**[1,c], **Damian O. Dieke**[3,d],
Olumuyiwa E. Ojo[4,e], Seye E. Akinsanya[2,f], Tolulope O. Abiola[2,g], Anna Feldman[5,h]
[1]Instituto Politécnico Nacional, Centro de Investigación en Computación, CDMX, Mexico
[2]Federal University Oye-Ekiti, Nigeria; [3]Caritas University, Nigeria; [4]Lead City University, Nigeria;
[5]Montclair State University, USA
{[a]olumideoea, [b]olaronke.oluwayemisi, [c]hiramcalvo, [d]dieketobesky, [e]muyeskin, [f]akinsanyaseye, [g]abiolato92}@gmail.com;
[h]feldmana@montclair.edu

Abstract

In this paper, we share our best performing submission to the Arabic AI Tasks Evaluation Challenge (ArAIEval) at ArabicNLP 2023. Our focus was on Task 1, which involves identifying persuasion techniques in excerpts from tweets and news articles. The persuasion technique in Arabic texts was detected using a training loop with XLM-RoBERTa, a language-agnostic text representation model. This approach proved to be potent, leveraging fine-tuning of a multilingual language model. In our evaluation of the test set, we achieved a micro F1 score of 0.64 for subtask A of the competition.

1 Introduction

In an era defined by the rapid dissemination of information through digital channels, the task of recognizing persuasion techniques in text is now more crucial than ever (Hasanain et al., 2023; Hossain et al., 2021; Gupta et al., 2021; Sadeghi et al., 2023; Alam et al., 2022; Abujaber et al., 2021). The advent of the Internet and social media has created new avenues for influence and manipulation (Dholakia et al., 2023; Ruffo et al., 2023; Botes, 2023). Although these technological advances have undoubtedly given people unparalleled access to information and a platform to express their thoughts, they have also introduced new avenues for persuasion, influence, and even propaganda. At the heart of this shift in our way of life is the fundamental challenge of distinguishing between genuinely informative and impartial content and content subtly crafted to promote a specific agenda or ideology. A critical component of the fight against spreading misinformation is the development of tools and resources in NLP that can detect persuasion technique in news articles and posts on social media.

Arabic language is among the most spoken languages in the world (Ghazzawi, 1992). The Arabic speaking world stands out in importance due to its intricate mix of language, culture, and geography. The diversity of linguistic expressions in Arabic extends beyond the spoken word and permeates every aspect of life, including digital text. Arabic connects a vast and diverse population of native speakers and foreigners with its rich heritage and multiple dialects. Its influence extends over a wide territory, from Arab nations in the southern part of the Arabian Peninsula, to Asia, and to the Maghreb in North Africa and the heart of the Arabian Peninsula (Huafeng et al., 2019). In this digital age, the importance and extensive use of Arabic in various forms reflects the profound impact of technology on this linguistic community. It has provided a platform for Arabic speakers around the world to engage in dialogue, share ideas, and express their thoughts in a global context. The diverse Arab-speaking populations foster a rich and dynamic environment where individuals can connect, collaborate, and debate issues of global significance. However, the proliferation of digital media in Arabic also presents challenges. Digital media have become a fertile ground for the dissemination of persuasive content, including propaganda, misinformation, and various forms of manipulation (Aleroud et al., 2023; Abd Elaziz et al., 2023). The use of technology in preserving the integrity of the Arabic language and ensuring responsible use of digital media is of utmost importance.

In today's world, we are surrounded by information, especially on the Internet and social media. Text classification can serve as a foundational step for the detection of the persuasion technique in text on social media and the Internet. These texts can be classified according to their emotional tone using sentiment analysis techniques (Nikolaidis et al., 2023; Ojo et al., 2022b, 2021, 2020, 2023, 2022a; Piskorski et al., 2023; Hromadka et al., 2023). Persuasion techniques can be tricky to spot because they come in many forms, such as stories, logical arguments, or even subtle language tricks to

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 594–599
December 7, 2023 ©2023 Association for Computational Linguistics

sway our thinking. These techniques are powerful and are not always used for good reasons. To deal with this, researchers and experts are working on ways to detect when someone is trying to persuade people through text. In this way, we will be able to recognize when we are influenced and when people are spreading false or misleading information, which can be harmful. In this paper, we develop a methodology to automatically identify and analyze persuasion techniques in text using the XLM-RoBERTa model. Using the power of NLP, we can harness the capabilities of state-of-the-art models and unravel the persuasion techniques embedded within Arabic text, contributing to both media ethics and the combating of misinformation. The intricacies of persuasion technique detection in Arabic are discussed in detail, along with the possibility of applying this knowledge across different languages.

2 Related Work

A significant amount of research has been conducted on the detection of persuasion techniques in text (Hasanain et al., 2023; Modzelewski et al., 2023; Hossain et al., 2021; Sadeghi et al., 2023; Abujaber et al., 2021). Researchers have explored binary and multilabel approaches to detecting persuasion techniques.

In their article, the authors in (Modzelewski et al., 2023) focused on detecting genres and persuasion techniques in multiple languages using various data augmentation techniques to enhance their models. For genre detection, they created synthetic texts using the GPT-3 Davinci language model, while for persuasion technique detection, they augmented the dataset using text translation with the DeepL translator. Their fine-tuned models achieved top ten rankings in all languages, demonstrating the effectiveness of their approach. They also excelled in genre detection, securing top positions in Spanish, German, and Italian. They presented a single multilingual system using the RoBERTa model to classify online news genres and improved this system by adding texts generated by the GPT3-Davinci model to the training dataset.

(Maram et al., 2023) addressed misinformation in mainstream and social media and the challenges faced by manual detection and verification efforts by journalists and fact checkers in the SemEval-2023 task (Piskorski et al., 2023). The task included three subtasks, six languages, and three surprise test languages, totaling 27 different test scenarios. The authors successfully submitted entries for all 27 test setups and the official results placed their system among the top three for 10 of these setups. They fine-tuned transformer models in the multiclass and multilabel classification settings, experimenting with both monolingual and multilingual pre-trained models, as well as data augmentation. Their multilingual model based on XLM-RoBERTa demonstrated superior performance in all tasks, even for languages not seen during training.

The most effective solution in the detection of persuasion techniques for Subtask 3 of SemEval 2023 Task 3 by (Hromadka et al., 2023) delivered promising performance, with micro-F1 scores ranging from 36 to 55% for languages seen during training and 26 to 45% for languages unseen. Given the multilingual nature of the data and the presence of 23 labels (resulting in limited labeled data for some language-label combinations), the authors chose to fine-tune pre-trained transformer-based language models. Through extensive experimentation, they identified the optimal configuration, featuring a large multilingual model (XLM-RoBERTa) trained on all input data, with carefully calibrated confidence thresholds for known and surprise languages separately. Their final system demonstrated superior performance, ranking first in six of nine languages, including two surprise languages, and achieving highly competitive results in the remaining three languages.

The experiments of (Nikolaidis et al., 2023) focused on the detection of persuasion techniques in online news articles in Polish and Russian. These experiments used a taxonomy comprising 23 distinct persuasion techniques. Persuasion techniques were evaluated in several ways, including the granularity of the classification (coarse with six labels or fine with 23 labels) and the level of location of the labels (subword, sentence, paragraph). The study compared the performance of state-of-the-art transformer-based models trained both monolingually and multilingually. The findings indicate that multilingual models generally outperform monolingual models in various evaluation scenarios. However, due to the complexity of the task, there remains substantial room for improvement in the field of persuasion technique detection within online news articles.

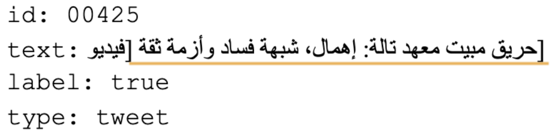
Inspired by (Maram et al., 2023; Hromadka et al., 2023), our research focus is to determine whether a multi-genre snippet (tweets and news paragraphs of news articles) contains a persuasion technique or not. This is a binary classification task, and we are categorizing Arabic text either as containing a persuasion technique or as lacking persuasion technique. Our approach involves fine-tuning a large pre-trained language model based on transformers. We conducted experiments with different language models and concluded with XLM-RoBERTa due to its superior performance. Furthermore, we fine-tuned our system by tweaking hyperparameters and conducting multiple iterations. This process involved calculating cross-entropy loss, performing backpropagation, and updating the model's weights.

3 Persuation Technique Detection

3.1 Dataset Analysis

Datasets provided by the organizers consist of a diverse collection of Arabic text samples, each associated with a label. These texts have been carefully selected to represent where persuasion techniques are present or not. The dataset were labeled as 'true' or 'false', where the true class represents texts where persuasion techniques are used to influence the reader's opinion, and the false class consists of texts that do not use persuasion techniques. The method involves binary classification to determine the presence of a persuasion technique within the document.

The dataset comprises 2,427 samples in the training set and 503 in the test set. Within the training data, there is an imbalance in the label distribution, with 1,918 samples labeled as "true" and 509 labeled as "false". An example of text that represents the persuasion technique in the dataset is shown in Figure 1.



```
id: 00425
text: حريق مبيت معهد تالة: إهمال، شبهة فساد وأزمة ثقة [فيديو]
label: true
type: tweet
```

Figure 1: Persuation technique in the Arabic text

3.2 Application of XLM-RoBERTa

Leveraging the Hugging Face Transformers library, we prepared and formatted the data so that it could be seamlessly incorporated into our model. In line with the methodology outlined in Section 4, we designed a custom model architecture and then proceeded to encode and load the data for further processing. During training, we optimized the model's classification head to minimize cross-entropy loss, and predictions are made on unseen text, followed by post-processing. This approach demonstrates the adaptability and effectiveness of XLM-RoBERTa in diverse linguistic contexts, offering practical solutions to automate the detection of persuasion techniques across languages.

4 System Setup and Experiments

4.1 Training Strategy

Our training strategy involves fine-tuning the XLM-RoBERTa model using the provided dataset. To optimize the performance of the model, we incorporate several key components and hyperparameters, which are summarized in Table 1.

4.2 Model Fine-Tuning

During model fine-tuning, we added a classification layer at the end of the pre-trained XLM-RoBERTa model. This additional layer allows the model to perform the specific classification task required. To prevent overfitting, a dropout layer was incorporated.

4.3 Class Weights

To address the class imbalance in the dataset, we adjusted the learning process using class weights. This ensures that the model effectively learns from all classes and is not biased by data imbalance.

4.4 Evaluation Metric

The model's performance was evaluated using the micro-score F1, the default metric for this subtask. This metric provides a comprehensive measure of the model's classification performance.

4.5 Training Process

The training process involved conducting a total of 6 epochs, where each epoch represents a complete pass through the entire training dataset. To avoid overfitting, early stopping was employed on the basis of cross-entropy loss.

4.6 Learning Rate Scheduler

A learning rate scheduler was implemented to dynamically adjust the learning rate during training.

Hyperparameter	Value
Learning rate	5×10^{-5}
Batch size	16
Epochs	6
Optimizer	ADAM
Early stopping	Cross-entropy loss
Learning rate scheduler	StepLR (factor=0.85, step size=2)

Table 1: Model Hyperparameters

Specifically, we used a StepLR scheduler with a reduction factor of 0.85 applied every 2 epochs. This scheduling strategy contributes to training stability and controlled convergence.

4.7 Optimization Process

To initiate the optimization process, we reset the gradients (setting them to zero). Subsequently, we conducted backpropagation to calculate gradients for all model parameters and, lastly, update the model's parameters using the computed gradients.

Our approach leverages these components and hyperparameters to fine-tune the model effectively, ensuring robust and controlled training.

5 Results

The evolution of pre-trained language models has ushered in significant advancements across different NLP tasks. In this section, we present the results of our experiments on persuasion technique detection in Arabic text. Our findings revealed the versatility of XLM-RoBERTa in effectively handling multilingual data. We evaluated the model using the micro-F1 score and an overview of the results achieved by our model on the dataset is shown in Table 2.

Model	F1-Score
Baseline model	0.4771
XLM-RoBERTa	0.6402

Table 2: Performance Comparison of Models for Persuasion Technique Detection in Arabic Text

From Table 1, our findings reveal the effectiveness of XLM-RoBERTa in identifying persuasion techniques within the Arabic text. The model's pre-trained knowledge, combined with its cross-lingual capabilities, makes it a promising tool for similar tasks in other languages as well. As Arabic is a morphologically rich language with various dialects, the success of XLM-RoBERTa in this task is a testament to its robustness and versatility.

6 Conclusion

The application of XLM-RoBERTa, a language-agnostic text representation model, for the detection of the persuasion technique in Arabic text illustrates the growing potential of cross-lingual models in specialized NLP tasks. Arabic, with its rich morphology and diverse dialects, presents unique challenges for text analysis. Our proposed model has the ability to capture the underlying structure and semantics of persuasion technique in text, regardless of language. The results obtained in our analysis demonstrate that XLM-RoBERTa can adapt effectively and perform well on such intricate tasks, even in languages that are structurally different from the ones they were originally trained on. This not only underscores the versatility of XLM-RoBERTa but also sets a promising direction for further research in detecting persuasion techniques across various languages. In future work, we plan to accommodate more languages in the dataset, and fine-tune other multilingual models for this task.

Ethics Statement

We acknowledge the influence that scientific research can have on society and recognize our responsibility to ensure its positive impact. Our research is carried out with the aim of addressing real-world challenges, promoting well-being, and improving quality of life. We actively seek feedback and input from those affected by our research. In the event that our research may have unintended negative impacts, we are committed to addressing and rectifying those issues promptly.

Acknowledgments

This work was done with partial support from the Mexican Government through the grant A1-S-47854 of CONACYT, Mexico, grants 20232138, 20230140, 20232080 and 20231567 of the Secretaría de Investigación y Posgrado of the Instituto Politécnico Nacional, Mexico. The authors thank CONACYT for the computing resources brought to them through the Plataforma de Aprendizaje Profundo para Tecnologías del Lenguaje of the Laboratorio de Supercómputo of the INAOE, Mexico and acknowledge the support of Microsoft through the Microsoft Latin America PhD Award.

References

Mohamed Abd Elaziz, Abdelghani Dahou, Dina Ahmed Orabi, Samah Alshathri, Eman M Soliman, and Ahmed A Ewees. 2023. A hybrid multitask learning framework with a fire hawk optimizer for arabic fake news detection. *Mathematics*, 11(2):258.

Dia Abujaber, Ahmed Qarqaz, and Malak A Abdullah. 2021. Lecun at semeval-2021 task 6: detecting persuasion techniques in text using ensembled pretrained transformers and data augmentation. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 1068–1074.

Firoj Alam, Hamdy Mubarak, Wajdi Zaghouani, Giovanni Da San Martino, and Preslav Nakov. 2022. Overview of the WANLP 2022 shared task on propaganda detection in Arabic. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 108–118, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Ahmed Aleroud, Abdullah Bani Melhem, Nour Alhussien, and Craig Douglas Albert. 2023. span-prop: Combatting contextualized social media state-linked propaganda in the middle east.

Marietjie Botes. 2023. Autonomy and the social dilemma of online manipulative behavior. *AI and Ethics*, 3(1):315–323.

Nikhilesh Dholakia, Aras Ozgun, and Deniz Atik. 2023. The miasma of misinformation: a social analysis of media, markets, and manipulation. *Consumption Markets & Culture*, pages 1–16.

Sabah Ghazzawi. 1992. The arabic language. *Washington DC: Center for Contemporary Arab Studies*.

Kshitij Gupta, Devansh Gautam, and Radhika Mamidi. 2021. Volta at semeval-2021 task 6: Towards detecting persuasive texts and images using textual and multimodal ensemble. *arXiv preprint arXiv:2106.00240*.

Maram Hasanain, Firoj Alam, Hamdy Mubarak, Samir Abdaljalil, Wajdi Zaghouani, Preslav Nakov, Giovanni Da San Martino, and Abed Alhakim Freihat. 2023. ArAIEval Shared Task: Persuasion Techniques and Disinformation Detection in Arabic Text. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore. Association for Computational Linguistics.

Tashin Hossain, Jannatun Naim, Fareen Tasneem, Radiathun Tasnia, and Abu Nowshed Chy. 2021. Csecudsg at semeval-2021 task 6: Orchestrating multimodal neural architectures for identifying persuasion techniques in texts and images. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 1088–1095.

Timo Hromadka, Timotej Smolen, Tomas Remis, Branislav Pecher, and Ivan Srba. 2023. Kinitveraai at semeval-2023 task 3: Simple yet powerful multilingual fine-tuning for persuasion techniques detection. *arXiv preprint arXiv:2304.11924*.

Hu Huafeng, Fan Zhihao, and Gao Xuezhen. 2019. The history of the arab nation and the arabic language. *Arabic Language, Literature & Culture*, 4(3):48.

Hasanain Maram, El-Shangiti Ahmed Oumar, Nandi Rabindra Nath, Nakov Preslav, and Alam Firoj. 2023. Qcri at semeval-2023 task 3: News genre, framing and persuasion techniques detection using multilingual models. *arXiv preprint arXiv:2305.03336*.

Arkadiusz Modzelewski, Witold Sosnowski, Magdalena Wilczynska, and Adam Wierzbicki. 2023. Dshacker at semeval-2023 task 3: Genres and persuasion techniques detection with multilingual data augmentation through machine translation and text generation. In *Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1582–1591.

Nikolaos Nikolaidis, Nicolas Stefanovitch, and Jakub Piskorski. 2023. On experiments of detecting persuasion techniques in polish and russian online news: Preliminary study. In *Proceedings of the 9th Workshop on Slavic Natural Language Processing 2023 (SlavicNLP 2023)*, pages 155–164.

OE Ojo, A Gelbukh, H Calvo, and OO Adebanji. 2021. Performance study of n-grams in the analysis of sentiments. *Journal of the Nigerian Society of Physical Sciences*, pages 477–483.

OE Ojo, A Gelbukh, H Calvo, A Feldman, OO Adebanji, and J Armenta-Segura. 2022a. Language identification at the word level in code-mixed texts using character sequence and word embedding. In *Proceedings of the 19th International Conference on Natural Language Processing (ICON): Shared Task on Word Level Language Identification in Code-mixed Kannada-English Texts*, pages 1–6.

Olumide E Ojo, Alexander Gelbukh, Hiram Calvo, Olaronke O Adebanji, and Grigori Sidorov. 2020.

Sentiment detection in economics texts. In *Mexican International Conference on Artificial Intelligence*, pages 271–281. Springer.

Olumide Ebenezer Ojo, Thang Ta Hoang, Alexander Gelbukh, Hiram Calvo, Grigori Sidorov, and Olaronke Oluwayemisi Adebanji. 2022b. Automatic hate speech detection using cnn model and word embedding. *Computación y Sistemas*, 26(2).

Olumide Ebenezer Ojo, Hoang Thang Ta, Alexander Gelbukh, Hiram Calvo, Olaronke Oluwayemisi Adebanji, and Grigori Sidorov. 2023. Transformer-based approaches to sentiment detection. In *Recent Developments and the New Directions of Research, Foundations, and Applications: Selected Papers of the 8th World Conference on Soft Computing, February 03–05, 2022, Baku, Azerbaijan, Vol. II*, pages 101–110. Springer.

Jakub Piskorski, Nicolas Stefanovitch, Giovanni Da San Martino, and Preslav Nakov. 2023. Semeval-2023 task 3: Detecting the category, the framing, and the persuasion techniques in online news in a multilingual setup. In *Proceedings of the the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2343–2361.

Giancarlo Ruffo, Alfonso Semeraro, Anastasia Giachanou, and Paolo Rosso. 2023. Studying fake news spreading, polarisation dynamics, and manipulation by bots: A tale of networks and language. *Computer science review*, 47:100531.

Aryan Sadeghi, Reza Alipour, Kamyar Taeb, Parimehr Morassafar, Nima Salemahim, and Ehsaneddin Asgari. 2023. Sinaai at semeval-2023 task 3: A multilingual transformer language model-based approach for the detection of news genre, framing and persuasion techniques. In *Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2168–2173.

NADI 2023:
The Fourth Nuanced Arabic Dialect Identification Shared Task

Muhammad Abdul-Mageed,[λ,ξ] **AbdelRahim Elmadany,**[λ] **Chiyu Zhang,**[λ]
El Moatez Billah Nagoudi,[λ] **Houda Bouamor,**[δ] **Nizar Habash**[μ]
[λ]The University of British Columbia, Vancouver, Canada; [ξ]MBZUAI, Abu Dhabi, UAE
[δ]Carnegie Mellon University in Qatar, Qatar
[μ]New York University Abu Dhabi, UAE
{muhammad.mageed@,a.elmadany@,chiyuzh@mail,moatez.nagoudi@}.ubc.ca
hbouamor@cmu.edu   nizar.habash@nyu.edu

Abstract

We describe the findings of the fourth Nuanced Arabic Dialect Identification Shared Task (NADI 2023). The objective of NADI is to help advance state-of-the-art Arabic NLP by creating opportunities for teams of researchers to collaboratively compete under standardized conditions. It does so with a focus on Arabic dialects, offering novel datasets and defining subtasks that allow for meaningful comparisons between different approaches. NADI 2023 targeted both dialect identification (Subtask 1) and dialect-to-MSA machine translation (Subtask 2 and Subtask 3). A total of 58 unique teams registered for the shared task, of whom 18 teams have participated (with 76 valid submissions during test phase). Among these, 16 teams participated in Subtask 1, 5 participated in Subtask 2, and 3 participated in Subtask 3. The winning teams achieved 87.27 F_1 on Subtask 1, 14.76 Bleu in Subtask 2, and 21.10 Bleu in Subtask 3, respectively. Results show that all three subtasks remain challenging, thereby motivating future work in this area. We describe the methods employed by the participating teams and briefly offer an outlook for NADI.

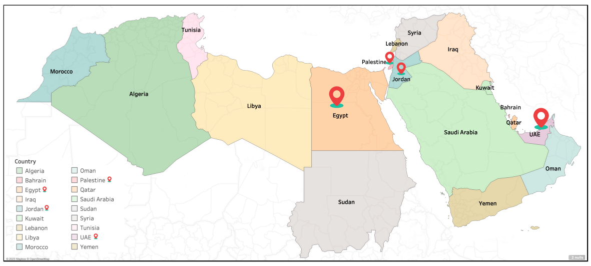


Figure 1: A map of the Arab World showing the 18 countries in the *Subtask 1* dataset and the 4 countries in the *Subtask 2* and *Subtask 3* datasets. Each country is coded in a color different from neighboring countries. Subtasks 2 and 3 countries are coded as red pins.

1 Introduction

Arabic is a term usually used to collectively refer to a host of languages and language varieties, rather than a single language. While most of these languages and varieties are similar to one another, there can be significant differences between some of them. For example, Egyptian Arabic and Moroccan Arabic are not mutually intelligible. Arabic can also be classified into three broad categories, classical, modern standard, and dialectal. Of these, *Classical Arabic (CA)* represents the variety used in old forms of literature such as poetry and the Qur'an, the Holy Book of Islam. Association with religion and literary expression endows CA with prestige, and it continues to be used to date side by side with other varieties. *Modern Standard Arabic (MSA)* (Badawi, 1973; Abdul-Mageed et al., 2020b) is a modern-day variety that is more familiar to native speakers and is usually employed by pan-Arab media organizations, government, and in education. The third category, *Dialectal Arabic (DA)*, is itself a superclass that is collectively assigned to a host of varieties that are sometimes defined regionally (e.g., Gulf, Levantine, Nile Basin, and North African (Habash, 2010; Abdul-Mageed, 2015)), but are increasingly recognized at the more nuanced levels of country or even sub-country (Bouamor et al., 2018; Abdul-Mageed et al., 2020b)). NLP treatment of Arabic dialects has thus far focused more on dialect identification (Abdul-Mageed et al., 2020b; Bouamor et al., 2019; Darwish et al., 2018), machine translation (MT) (Zbib et al., 2012), morphosyntax (Obeid et al., 2020).

Dialect identification is the task of automatically detecting the source variety of a given text or speech segment, and is the main focus of the current work where we introduce the findings and results of the fourth Nuanced Arabic Dialect Identification Shared Task (NADI 2024). The main objective of NADI is to encourage research on Ara-

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 600–613
December 7, 2023 ©2023 Association for Computational Linguistics

bic dialect processing by offering datasets and facilitating diverse modeling opportunities under a common evaluation setup. The first instance of the shared task, NADI 2020 (Abdul-Mageed et al., 2020a), focused on province-level dialects. NADI 2021 (Abdul-Mageed et al., 2021b), the second iteration of NADI, focused on distinguishing both MSA and DA according to their geographical origin at the country level. The third instance, NADI 2022 (Abdul-Mageed et al., 2022), investigated both Arabic dialect identification and dialectal sentiment analysis. NADI 2023, the current edition, continues this tradition of extending to tasks beyond dialect identification. Namely, we propose new subtasks focused at machine translation from Arabic dialects into MSA.

More concretely, NADI 2023 shared task is comprised of three subtasks: **Subtask 1** on dialect identification, while **Subtask 2** and **Subtask 3** are on dialect MT. The difference between Subtask 2 and Subtask 3 is that the former is a *closed track* where participants are allowed to use only our provided training data, whereas the latter is *open track* and so allows participants to train their systems on any additional datasets so long as these additional training datasets are public at the time of submission. While we invited participation in any of the three subtasks, we encouraged teams to submit systems to *all* subtasks. By offering three subtasks, our hope was to receive systems that exploit different methods and architectures. Many of the submitted systems investigated diverse approaches, thus fulfilling our objective. A total of 58 unique teams registered for NADI 2023. Of these, 18 unique teams actually made submissions to our leaderboard (n=76 valid submissions during test phase). We received 14 papers from 14 teams, of which we accepted 13 for publication. Results from participating teams show that both dialect identification at the country level and dialectal MT remain challenging even to complex neural methods. These findings clearly motivate future work on all tasks.

The rest of the paper is organized as follows: Section 2 provides a brief overview of Arabic dialect identification and sentiment analysis. We describe the two subtasks and NADI 2023 restrictions in Section 3. Section 4 introduces shared task datasets and evaluation setup. We present participating teams and shared task results and provide a high-level description of submitted systems in Section 5. We conclude in Section 6.

2 Literature Review

2.1 Arabic Dialects

As stated earlier, Arabic can be broadly categorized into CA, DA, and MSA. While CA and MSA have been examined extensively (Harrell, 1962; Cowell, 1964; Badawi, 1973; Brustad, 2000; Holes, 2004), DA became the center of attention only relatively recently. A significant challenge in studying DA has been the scarcity of resources. This prompted researchers to create new DA datasets, usually targeting a limited number of specific regions or countries (Gadalla et al., 1997; Diab et al., 2010; Al-Sabbagh and Girju, 2012; Sadat et al., 2014; Harrat et al., 2014; Jarrar et al., 2016; Khalifa et al., 2016; Al-Twairesh et al., 2018; Alsarsour et al., 2018; Kwaik et al., 2018; El-Haj, 2020). This was followed by several works that introduced multi-dialectal datasets and models for region-level dialect identification (Zaidan and Callison-Burch, 2011; Elfardy et al., 2014; Bouamor et al., 2014; Meftouh et al., 2015). The initial Arabic dialect identification shared tasks were part of the VarDial workshop series, primarily utilizing transcriptions of speech broadcasts (Malmasi et al., 2016). This was followed by creation of the Multi-Arabic Dialects Application and Resources project (MADAR), which provided finer-grained data and a lexicon (Bouamor et al., 2018). Although MADAR's dataset was used for identifying dialects at both the country and city levels (Salameh et al., 2018; Obeid et al., 2019), the fact that it is commissioned, rather than naturally occurring, makes it not be optimal for dialect identification especially in contexts such as social media.

Subsequently, larger datasets that cover between 10 to 21 countries were introduced (Mubarak and Darwish, 2014; Abdul-Mageed et al., 2018; Zaghouani and Charfi, 2018; Abdelali et al., 2021; Issa et al., 2021; Baimukan et al., 2022; Althobaiti, 2022). The majority of these datasets are compiled from social media posts, especially Twitter. Other works collect data at a more granular level. For instance, Abdul-Mageed et al. (2020b) introduces a Twitter dataset along with several models to identify variations in Arabic dialects at the country, province, and city levels. Althobaiti (2020) provides an overview of computational work on Arabic dialects. More recently, benchmarks such as ORCA (Elmadany et al., 2023) and DOLPHIN (Nagoudi et al., 2023) boast dialectal coverage. The NADI shared task continues to lead efforts on providing datasets and common evaluation settings for identifying Arabic dialects (Abdul-Mageed et al., 2020a, 2021b, 2022).

2.2 Machine Translation of Arabic Dialects

Several studies focus on machine translation of Arabic dialects. For example, Zbib et al. (2012) demonstrate effects of using both MSA and DA data on performance of Dialect/MSA to English MT. Sajjad et al. (2013) employs MSA as an intermediary language for translating Arabic dialects into English. Salloum et al. (2014) examine the impact of sentence-level dialect identification and various linguistic features on Dialect/MSA to English translation. Guellil et al. (2017) propose a neural system for translating Algerian Arabic written in Arabizi and Arabic script into MSA, while Baniata et al. (2018) introduce a system that translates Levantine (Jordanian, Syrian, Palestinian) and Maghrebi (Algerian, Moroccan, Tunisian) into MSA. Sajjad et al. (2020) propose an evaluation benchmark for Dialectal Arabic to English MT, along with several NMT systems using different training setups such as fine-tuning, data augmentation, and back-translation. Farhan et al. (2020) offer an unsupervised dialectal system where the source dialect (zero-shot) is not represented in training data. Nagoudi et al. (2021) propose a transformer-based MT system for translating from code-mixed MSA and Egyptian Arabic into English. More recently, Kadaoui et al. (2023) present a comprehensive evaluation of large language models (LLMs), including Bard and ChatGPT, on the machine translation of ten Arabic varieties. To the best of our knowledge, our work is the first shared task to enable investigating MT in $four$ Arabic dialects, namely *Egyptian*, *Emirati*, *Jordanian*, and *Palestinian*. For our MT subtasks, we also annotate and release a novel dataset and facilitate comparisons in a standardized experimental setting.

2.3 Previous NADI Shared Tasks

NADI 2020 The first NADI shared task, (Abdul-Mageed et al., 2020a) was co-located with the fifth Arabic Natural Language Processing Workshop (WANLP 2020) (Zitouni et al., 2020). NADI 2020 targeted both country- and province-level dialects. It covered a total of 100 provinces from 21 Arab countries, with data collected from Twitter. It was the first shared task to target naturally occurring fine-grained dialectal text at the sub-country level.

NADI 2021 The second edition of the shared task (Abdul-Mageed et al., 2021b) was co-located with WANLP 2021 (Habash et al., 2021). It targeted the same 21 Arab countries and 100 corresponding provinces as NADI 2020, also exploiting Twitter data. NADI 2021 improved over NADI 2020 in that non-Arabic data were removed. In addition, NADI-2021 teased apart the data into MSA and DA and focused on classifying MSA and DA tweets into the countries and provinces from which they are collected. As such, NADI 2021 had four subtasks: MSA-country, DA-country, MSA-province, and DA-province.

NADI 2022 The third edition of the shared task (Abdul-Mageed et al., 2022) was co-located with WANLP 2021.[1] It focused on studying Arabic dialects at the country level as well as dialectal sentiment (i.e., sentiment analysis of data tagged with dialect labels). We discuss NADI 2023 in more detail in the next section.

3 Task Description

In NADI-2023, we place our emphasis on two NLP tasks, both crucial to processing of dialectal Arabic. Dialect identification remains an important step in any pipeline for processing dialects, for which reason NADI-2023 **Subtask 1** maintains the focus on identification of Arabic dialects. In particular, Subtask 1 targets dialect at the country level. Another important NLP task that has not particularly witnessed accelerated progress over the past few years is machine translation of Arabic dialects. For this reason, we take as our second focal point MT of dialects through **Subtask 2** and **Subtask 3**. We now describe each subtask in detail.

3.1 Subtask 1: Dialect Identification

Dialect identification has consistently been central to the NADI shared task over the years (2020a; 2021b; 2022). In NADI-2023, we continue to focus on dialect identification through Subtask 1.

Data For this purpose, we provide a new Twitter dataset (i.e., TWT-2023), encompassing 18 distinct dialects, totaling 23.4K tweets. We also provide access to additional datasets for training. These are NADI-2020 (Abdul-Mageed et al., 2020a), NADI-2021 (Abdul-Mageed et al., 2021b), and MADAR (Bouamor et al., 2018) *training* splits. We

[1] https://sites.google.com/view/wanlp2021

Country	Content
Algeria	مهم جميع واحد انا ثقيل عليه يبلوكيني و ميضليش ينفخلي فيهم بهدره تع تقي يكونو معاك هاك يشوفوك وليت هاك يولو هاك و يكتلوك بهاك سورتو لا صابوك هاك يولو عليك غي هاك هوما هاك
Iraq	بس ما اعتقد لان هيج وقت كل الاكلات حلوه بو خوشي بهس تهت فيت زيانه ته سبي بيت اوب ههمي دلي هه
Jordan	من لما صحيت حاسه في اشي غريب اشي ناقص لحتي امي سالتني شربتي قهوه اليوم ها ها ها هه ممكن والله كل اشي بهالبلد ممكن يصير
Saudi Arabia	ذيك الايام خشم بخاري بغيبوبه وش اسمه يمدي حرمته تدعي علي هه
Sudan	اكتر من ١٠ سنين شغاله يا استنكرت موضوع يا ذرفت دموع اسه جابت ليها قطع رؤوس #محن _الكيزان في #السودان اي زول في العلاقه عاوز اعمل تقيل بقتل العلاقه والله يعني الطرفين ساكتين لو في طرف م عاوز التاني اصرفها ليهو

Table 1: Random examples from NADI-2023 Subtask-1 training dataset spanning five different countries.

Dialect	Source (Dialect)	Target (MSA)
Egyptain	ايوا حضرتك ده حق ابويا، حقي أنا بقى؟	إي حضرتك هذا حق أبي ، أين حقي أنا إذن ؟
	حتى ابوك نفسه مش حيقدر يشفعلك عنده.	وأبوك نفسه لن يقدر يشفع لك عنده .
	يا عم احنا مش ناقصينك الله يسترك عيب كده اختشي فوق فوق.	يا عم ! نحن فاض بنا الكيل ، الله يسترك ، هذا عيب ! استحِ ! أفِق ! أفِق!
Emirati	زين خبروني شو السالفة ؟	إذا أخبروني، ما القصة؟
	مابا حد يدري في الفريج، إن بو محمد انسرق	لا أريد أن يعلم أحدًا في الحي، أن أبو محمد انسرق
	انزين، بغينا اثنين زنجبيل حار	حسنا، نريد اثنين من الزنجبيل الحار
Jordanian	بس ما بعرف شو، مش عارف شو صارلي	لكن لا أعلم ماذا، لا أعلم ما حدث لي
	كله منك انت السبب ليش ماخليتني ماسك بخناقه	كله بسببك أنت السبب لماذا لم تدعني ممسكاً بعنقه
	بكير من عمرك	بارك الله في عمرك
Palestinian	طيب و هاي الملاعب وين؟	حسناً، وهذه الملاعب أين؟
	شو يا أبو ناصر سمعت صاير تهدد في القتل	ما هذا يا أبو ناصر، سمعت أنك أصبحت تهدد بالقتل
	شاطر، إلك عندي باكيت حلقوم	حصيف، سأكافئك بعلبة كاملة من حلوى الحلقوم

Table 2: Random examples from MT-2023-DEV dataset spanning the four covered dialects.

refer to these datasets as NADI-2020-TWT, NADI-2021-TWT, and MADAR-2018, respectively. We provide further details about these datasets in Section 4.1. Table 1 shows examples from tweets in our NADI-2023 dataset for five countries.

Restrictions It is essential to note that Subtask 1 operates under a ***closed-track*** policy where participants are allowed to use for system training *only* datasets we provide. That is, no external data sources can be used for training purposes in this subtask.

3.2 Subtasks 2 and 3: Machine Translation

In this competition version, we introduce a new theme to NADI centered around machine translation from *four* Arabic dialects to Modern Standard Arabic (MSA) at the sentence level. We present two versions of this competition, one is a closed track (Subtask 2), and the other is an open track (Subtask 3).

Dev and Test Data For both Subtask 2 and Subtask 3, we manually curate new development and test datasets that each cover *four* Arabic dialects: *Egyptian*, *Emirati*, *Jordanian*, and *Palestinian*. We refer to these new datasets as MT-2023-DEV and MT-2023-TEST, respectively. MT-2023-DEV comprises 400 sentences, with 100 sentences representing each of the four dialects; whereas MT-2023-TEST has a total of $2,000$ sentences, 500 from each dialect. Table 2 shows example sentences from MT-2023-DEV for each of the four countries. During the competition, we intentionally kept the source domain of these datasets undisclosed. Since we typically keep a live leaderboard for post-competition evaluation, we will not disclose the MT-2023* data domain.

Restrictions For the MT theme, restrictions on use of training datasets depend on the type of track. We offer two tracks, one closed and another open each with its own subtask. We introduce these subtasks now, detailing respective track information.

Subtask 2 – Closed-Track Dialect to MSA MT For Subtask 2 training, we restrict to the MADAR parallel dataset (Bouamor et al., 2019). More precisely, participants were allowed to use only the training split of MADAR parallel corpus for this subtask, and report on the development and test sets we provide. This meant that use of MADAR development and test datasets was not allowed for Subtask 2.

Subtask 3 – Open-Track Dialect to MSA MT For Subtask 3 training, participants were allowed to train their systems on any additional datasets of their choice so long as these additional training datasets are public at the time of submission. For example, participants were allowed to manually create new parallel datasets. For transparency and wider community benefits, we required researchers participating in the open track subtask to submit the datasets they create along with their Test set submissions.

4 Shared Task Datasets and Evaluation

In this section, we describe the datasets we make available to participants, introduce the chosen evaluation metrics, and outline the clear instructions we provided for the submission process.

4.1 Datasets

- **TWT-2023**: Abdul-Mageed et al. (2020b) introduce a vast dataset comprising ∼6B tweets from 2.7M users. They systematically extract tweets that contain geographic information and subsequently embark on a manual annotation process for each user, classifying their location at the city, state, and country levels. This effort results in the identification of ∼ 500M tweets coming from 233K users spread across 319 cities within 21 Arab countries. For Subtask 1, we randomly select from this data 1,000 training, 100 development, and 200 testing tweets for each of the 18 covered countries. In total, this amounts to 23,400 tweets that we refer to as TWT-2023. We split TWT-2023 into Train (18K), Dev (1.8K), and Test (3.6K).

Country	**NADI-2020**	**NADI-2021**	**MADAR-18**
Algeria	1,491	1,809	1,600
Bahrain	210	215	–
Egypt	4,473	4,283	4,800
Iraq	2,556	2,729	4,800
Jordan	426	429	3,200
Kuwait	420	429	–
Lebanon	639	644	1,600
Libya	1,070	1,286	3,200
Morocco	1,070	858	3,200
Oman	1,098	1,501	1,600
Palestine	420	428	1,600
Qatar	234	215	1,600
Saudi Arabia	2,312	2,140	3,200
Sudan	210	215	1,600
Syria	1,070	1,287	3,200
Tunisia	750	859	3,200
UAE	1,070	642	–
Yemen	851	429	1,600
Total	**20,370**	**20,398**	**40,000**

Table 3: Distribution of Subtask-1 additional training data. For NADI-2023, we also distribute a total of 18,000 tweets for Train, 1,800 for Dev, and 3,600 for Test (with 1,000, 100, and 200 from each country for 18 countries listed in the table for Train, Dev, and Test, respectively). For Subtask 2 and Subtask-3, we extract MADAR-4-MT from Egyptian, Emirati, Jordanian, and Palestinian data in MADAR-18 (see Section 4).

- **NADI-202X-TWT**. We also distribute **NADI-2020-TWT** and **NADI-2021-TWT** datasets. These datasets are similarly collected from Twitter. For both of them, we use the Twitter API to crawl data from 21 Arab countries for a period of 10 months (Jan. to Oct., 2019). For each case, we label tweets from each user with the country from which they posted for the whole of the 10 months period, thus exploiting *consistent posting location* as a proxy for *dialect labels*. We use the same training splits as both NADI-2020 and NADI-2021, but only include data that cover the 18 Arab countries we target in the current 2023 edition. It is also noteworthy that we do not provide the NADI-2022 training dataset since it is identical to the training set used in NADI 2021.

- **MADAR-18**: The MADAR corpus is a collection of parallel sentences encompassing the dialects of 25 cities from across the Arab world, along with English, French, and MSA. Since this dataset does not originally have country-level labels, we map the 25 cities to their respective countries. As a result, we acquire a customized version of MADAR that we refer to as MADAR-18. We offer the dialectal side of MADAR-18 for optional use for training systems for Subtask-1.

- **MADAR-4-MT**: We extract parallel dialectal-to-MSA data of four dialects from MADAR-18 for training MT systems for Subtask-2 and Subtask-3. The four pairs involve Egyptian, Emirati, Jordanian, and Palestinian at the dialectal side.

Table 3 present the statistics and characteristics of NADI-2023's Subtask-1 training, development, and test datasets, along with the distribution of our additional resources, i.e, NADI-2020-TWT, NADI-2021-TWT, and MADAR-18.[2]

4.2 Evaluation Metrics

The official evaluation metric for Subtask-1 is the macro-averaged F_1 score. In addition to this metric, we also report system performance in terms of `Precision`, `Recall`, and `Accuracy` for submissions to this Subtask 1. For both Subtask 2 and Subtask 3, we use the `Bleu` score as the official metric. The `Bleu` score is computed separately for each of the four dialects (i.e., Egyptian, Emirati, Jordanian, and Palestinian). We then use the average of these individual Bleu scores to rank the submitted systems for Subtask 2 and Subtask 3.

4.3 Submission Roles

We allowed participant teams to submit up to *five* runs for each test set, for each of the three subtasks. In each case, we only retain the submission with the highest score for each team. While the official results were exclusively based on a blind test set, we also requested participants to include their results on the development datasets (Dev) in their papers.

To facilitate the evaluation of participant systems, we established a CodaLab competition for scoring each subtask (i.e., a total of three Codalabs).[3] Similar to previous NADI editions, we are keeping the CodaLab for each subtask active even after official competition has concluded. This is to encourage researchers interested in training models and assessing systems using the shared task's blind test sets. Consequently, we will not disclose the labels for the test sets of any of the subtasks.

5 Shared Task Teams & Results

5.1 Participating Teams

We received a total of 58 unique team registrations. At the testing phase, a total of 76 valid entries were submitted by 18 unique teams. The breakdown across the subtasks is as follows: 49 submission for Subtask 1 from 16 teams, 16 submissions for Subtask 2 from 5 teams, and 11 submissions for Subtask 3 from 3 teams. Table 4 lists the 18 teams. A total of 14 teams submitted 14 description papers from which we accepted 13 papers for publication. Accepted papers are cited in Table 4.

5.2 Baselines

For comparison, we provide three baselines for each of the three subtasks. For **Subtask 1**, we finetune MARBERT$_{\text{v2}}$ (Abdul-Mageed et al., 2021a), AraBERT$_{\text{twitter}}$ (Antoun et al., 2021), and CAMeLBERT$_{\text{da}}$ (Obeid et al., 2020), on `TWT-2023` training data (see Section 3.1). For **Subtask 2** and **3**,[4] we finetune AraT5$_{\text{v2}}$ (Nagoudi et al., 2022), mT5 (Xue et al., 2020), and AraBART (Eddine et al., 2022) on MADAR-4-MT (see Section 3.1). In each subtask, we label these baselines as **Baseline I**, **II**, and **III**, respectively.

For all the baselines in both tasks, we finetune each model using the training data specific to each subtask (i.e., TWT-2023 for Subtask 1 and MADAR-4-MT for Subtask 2 and Subtask 3) for 10 epochs with a learning rate of $2e-5$ and batch size of 32. The maximum length is set to 256 tokens and we set an early stopping patience to 5. Following each epoch, we evaluate each model and select the best the best-performing model on the respective Dev set. Subsequently, we present the performance metrics of this best-performing model on the test datasets.

5.3 Shared Task Results

Table 5 presents the leaderboard of Subtask 1 and is sorted by macro-F_1. As Table 5 shows, for each team, we take their best macro-F_1 score to represent them. `Team NLPeople` (Elkaref et al., 2023)

[2] Recall that MADAR-4-MT is extracted from MADAR-2018

[3] The different CodaLab competitions are available at the following links: `subtask-1`, `subtask-2`, and `subtask-3`.

[4] We use the same baseline models to both Subtask 2 and Subtask 3.

Team	Affiliation	Tasks
AIC	Applied Innovation Center, Egypt	1
ANLP-RG (Derouich et al., 2023)	University of Sfax, Tunisia	2
Arabitools	STEAM Solutions, Palestine	1
Cordyceps	University of Toronto, Canada	1
DialectNLU (Veeramani et al., 2023)	UCLA, USA	1, 2
Exa	Exa, Iran	1
Frank (Azizov et al., 2023)	MBZUAI, UAE	1
Fraunhofer IAIS	Fraunhofer IAIS, Germany	1, 2
Helsinki-NLP (Kuparinen et al., 2023)	University of Helsinki, Finland	2, 3
ISL-AAST (El-sayed and Elmadany, 2023)	Arab Academy for Science and Technology, Egypt	1
IUNADI (Hatekar and Abdo, 2023)	Indiana University Bloomington, USA	1
Mavericks (Deshpande et al., 2023)	Pune Institute of Computer Technology, India	1
NAYEL	Benha University, Egypt	1
NLPeople (Elkaref et al., 2023)	IBM Research Europe, UK	1
rematchka (Abdel-Salam, 2023)	Cairo University, Egypt	1, 2, 3
SANA (Almarwani and Aloufi, 2023)	Taibah University, KSA	1
UniManc (Khered et al., 2023)	University of Manchester, UK	2, 3
UoT (Nwesri et al., 2023)	University of Tripoli, Libya	1
usthb (Lichouri et al., 2023)	USTHB, Alegria	1

Table 4: List of teams that participated in NADI-2023 shared task. Teams with accepted papers are cited.

Rank	Team	F1	Acc.	Pre.	Rec.
1	NLPeople	87.27	87.22	87.37	87.22
2	rematchka	86.18	86.17	86.29	86.17
3	Arabitools	85.86	85.81	86.10	85.81
4	SANA	85.43	85.39	85.60	85.39
5	Frank	84.76	84.75	84.95	84.75
6	ISL-AAST	83.73	83.67	83.87	83.67
7	UoT	82.87	82.86	83.17	82.86
8	AIC	82.37	82.42	82.57	82.42
9	Cordyceps	82.17	82.14	82.57	82.14
Baseline I	MARBERT$_{v2}$	81.44	81.36	81.68	81.36
10	DialectNLU	80.56	80.50	80.92	80.50
Baseline II	AraBERT$_{twitter}$	77.02	76.97	77.54	76.97
11	Mavericks	76.65	76.47	77.43	76.47
Baseline III	CAMeLBERT$_{da}$	74.56	74.47	74.90	74.47
12	exa	70.72	71.03	72.26	71.03
13	IUNADI	70.22	70.78	71.32	70.78
14	NAYEL	63.09	63.39	63.30	63.39
15	usthb	62.51	62.17	63.07	62.17
16	Fraunhofer IAIS	29.91	33.14	38.47	31.39

Table 5: Results of Subtask 1 (Country-Level DA).

Rk	Team	Overall	Egy.	Emi.	Jor.	Pal.
1	UniManc	14.76	16.04	14.30	12.55	13.55
2	Helsinki	14.28	12.22	23.13	11.15	13.42
3	DialectNLU	13.43	11.45	21.59	10.64	12.66
4	rematchka	11.37	11.18	11.99	10.47	10.86
5	ANLP-RG	10.02	10.25	8.50	10.26	9.33
Baseline I	AraT5$_{v2}$	7.70	5.50	10.45	9.51	6.48
6	Fraunhofer IAIS	5.85	8.08	3.90	4.96	6.01
Baseline II	mT5	2.98	4.17	3.66	3.89	3.95
Baseline III	AraBART	2.63	2.44	3.16	1.89	2.60

Table 6: Results of Subtask 2 (Closed AD to MSA MT)

Rank	Team	Overall	Egy.	Emi.	Jor.	Pal.
1	UniManc	21.10	17.65	28.46	22.03	17.29
2	Helsinki-NLP	17.69	16.11	25.81	15.60	15.91
3	rematchka	11.37	11.18	11.99	10.47	10.86
Baseline I	AraT5$_{v2}$	5.41	5.50	5.84	6.06	4.47
Baseline II	mT5	2.98	4.17	3.66	3.89	3.95
Baseline III	AraBART	1.12	0.00	0.00	1.17	1.10

Table 7: Results of Subtask 3 (Open DA to MSA MT).

obtained the best performance on Subtask 1 with 87.27 macro-F_1. We can observe that 9 teams outperform our strongest baseline, MARBAET (i.e, Baseline I). Table 6 and Table 7 show the leaderboard of Subtask 2 and 3, respectively. Both are sorted by their main metrics, the overall BLEU score. Team UniManc (Khered et al., 2023) won both subtasks, achieving the best BLEU scores of 14.76 and 21.10 on Subtask 2 and 3, respectively. We observe that *five* teams outperform our Baseline I on Subtask 2.

5.4 General Description of Submitted Systems

In Tables 8 and 9, we provide a high-level summary of the submitted systems. For each team, we list

Team	# submit	F_1	Features				Techniques								
			N-gram	*TF-IDF*	*Linguistic*	*Word embeds*	*Classical ML*	*Neu. nets*	*PLM*	*Ensemble*	*Adapter*	*Hie. Cls*	*Prompting*	*Contrast. L*	*Data Aug.*
NLPeople	5	87.27					✓	✓	✓	✓					✓
rematchka	3	86.18							✓	✓	✓		✓	✓	
Arabitools	4	85.86			✓				✓	✓					
SANA	2	85.43							✓	✓					
Frank	2	84.76							✓	✓					
ISL-AAST	5	83.73						✓	✓	✓					
UoT	2	82.87					✓		✓						✓
AIC	5	82.37	✓		✓				✓	✓		✓			✓
Cordyceps	4	82.17													
DialectNLU	5	80.56		✓	✓			✓	✓	✓					
IUNADI	1	70.22			✓				✓	✓					
Mavericks	1	76.65							✓	✓					
NAYEL	5	63.09	✓	✓			✓								
usthb	3	62.51	✓	✓	✓	✓	✓								✓

Table 8: Summary of approaches used by participating teams in Subtask 1. Teams are sorted by their performance on the official metric, Macro-F_1 score. Classical machine learning (ML) indicates any non-neural machine learning methods such as naive Bayes and support vector machines. The term "neural nets" refers to any model based on neural networks (e.g., FFNN, RNN, CNN, and Transformer) trained from scratch. PLM refers to neural networks pretrained with unlabeled data such as MARBERT. (Hie. Cls, hierarchical classification approach); (Contrast. L, contrastive learning); (Data Aug., data Augmentation).

Team	# submit	BLUE	Techniques				
			Classic ML	NN	PLM	Ensemble	Aug.
			Subtask 2				
UniManc	5	14.76			✓	✓	
Helsinki	3	14.28	✓	✓	✓		
DialectNLU	5	13.43			✓	✓	
rematchka	1	11.37		✓	✓		
ANLP-RG	1	10.02		✓	✓		✓
			Subtask 3				
UniManc	5	21.10			✓	✓	✓
Helsinki-NLP	5	17.69	✓	✓	✓		✓
rematchka	1	11.37			✓		

Table 9: Summary of approaches used by participating teams in Subtask 2 and 3. Teams are sorted by their performance on BLEU score for both Subtasks. Classical machine learning (ML) indicates any non-neural machine learning methods such as naive Bayes and support vector machines. "NN" refers to any model based on neural networks (e.g., FFNN, RNN, CNN, and Transformer) trained from scratch. PLM refers to neural networks pretrained with unlabeled data such as AraT5. (Aug., data augmentation).

the best score with the main metric of each subtask and the number of submissions made by the team. As shown in these tables, most teams use pretrained language models (PLM), including Transformer encoder-based PLMs (e.g., AraBERT (Antoun et al., 2020) and MARBERT (Abdul-Mageed et al., 2021a)) for Subtask 1 and Transformer encoder-decoder PLMs (e.g., ArabT5 (Nagoudi et al., 2022)) for Subtask 2 and Subtask 3. Ensemble voting is also an effective approach most teams employ in Subtask 1.

The top team of Subtask 1, i.e., NLPeople (Elkaref et al., 2023), exploits MARBERT, AraBERT, and AraT5 with different finetuning strategies (e.g., staged finetuning). To enrich the learning context, they use a retrieval method to find similar texts from the training set for a given text and then append the retrieved texts along with corresponding labels as additional input. Their best submission is an ensemble with ten models. Team rematchka (Abdel-Salam, 2023), exploits MARBERT, AraBERT, AraELECTRA (Antoun et al., 2021), and CAMeLBERT (Obeid et al., 2020) with different prompting techniques and add linguistic features to their models. They also use supervised contrastive loss (Gunel et al., 2021) to enhance model finetuning. Teams SANA (Almarwani and Aloufi, 2023) and Frank (Azizov et al., 2023) both finetune PLMs and apply ensemble voting to achieve their best performance.

On Subtask 2 (closed track), the winning team, Team UniManc (Khered et al., 2023), finetune three variants of T5 models (i.e., mT5 (Xue et al., 2021), mT0 (Muennighoff et al., 2023), and AraT5) with the officially released dataset. For Subtask 3 (open track), Team UniManc collects four additional supervised datasets and uses GPT-3.5-turbo to translate 2,712 samples from Subtask 1. Team Helsinki-NLP (Kuparinen et al., 2023) finetune ByT5 (Xue et al., 2022) and AraT5 with the officially released dataset of Subtask 2. For Subtask 3, they collect six monolingual MSA datasets and synthesize a parallel dataset by exploiting character-level statistical machine translation models to translate the MSA to different dialects. They then finetune PLMs with the supervised dataset from Subtask 2 and their synthetic dataset. Similarly, both teams DialectNLU and rematchka finetune AraT5 with the training data of Subtask 2.

6 Conclusion

We presented findings and results of NADI-2023, the fourth edition of the NADI shared task focused on fine-grained Arabic dialect identification. This edition also introduced two subtasks centered on machine translation from four Arabic dialects into MSA. Results acquired by participant teams show that dialect identification remains a challenging task but that various types of approaches, many of which involve exploiting language models, can be used to handle the task. Similarly, translating Arabic dialects is unsurprisingly very challenging due to lack of training data. In the future, we plan to continue supporting both dialect identification and machine translation through NADI.

7 Limitations

Our work has a number of limitations, as follows:

- Although we strive for widest coverage, this edition of NADI focused on only 18 country-level dialects. This is due to our inability to develop high quality datasets for a few countries such as *Comoros*, *Djibouti*, *Mauritania*, and *Somalia*.

- NADI continues to use short texts for the Arabic dialects. Due to lack of dialectal data from other sources, we depend on short posts from Twitter. Although these data have thus far empowered development of effective dialect identification models, it is desirable to afford data from other domains that have longer texts. This will allow development of more widely applicable models.

- Our MADAR-18 dataset is commissioned and, although useful, should not be used to analyze Arabic dialects as a replacement for naturally occurring data.

- Our machine translation subtasks focus only on four dialects and do not offer sizeable datasets. Modern MT systems need much larger data to perform well. Again, in spite of our best efforts, parallel datasets involving dialects remain limited.

8 Ethical Considerations

The NADI-2023 Subtask 1 dataset is sourced from the public domain (i.e., X former Twitter), with user personal information and identity carefully concealed. Similarly, the NADI-2023 Subtask 2 and Subtask 3 datasets are manually created. Again, we take meticulous measures to remove user identities and personal information from this dataset. As a result, we have minimal concerns about the retrieval of personal information from our data. However, it is crucial to acknowledge that the datasets we collect to construct NADI-2023 Subtask 1 may contain potentially harmful content. Additionally, during model evaluation, there is a possibility of exposure to biases that could unintentionally generate problematic content.

Acknowledgments

We acknowledge support from Canada Research Chairs (CRC), the Natural Sciences and Engineering Research Council of Canada (NSERC; RGPIN-2018-04267), the Social Sciences and Humanities Research Council of Canada (SSHRC; 435-2018-0576; 895-2020-1004; 895-2021-1008), Canadian Foundation for Innovation (CFI; 37771), Digital Research Alliance of Canada,[5] and UBC Advanced Research Computing-Sockeye.[6]

References

Reem Abdel-Salam. 2023. rematchka at NADI 2023 shared task: Parameter Efficient tuning for Dialect Identification and Dialect Machine Translation. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*.

[5] https://alliancecan.ca

[6] https://arc.ubc.ca/ubc-arc-sockeye

Ahmed Abdelali, Hamdy Mubarak, Younes Samih, Sabit Hassan, and Kareem Darwish. 2021. QADI: Arabic dialect identification in the wild. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 1–10, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Muhammad Abdul-Mageed. 2015. *Subjectivity and sentiment analysis of Arabic as a morophologically-rich language*. Ph.D. thesis, Indiana University.

Muhammad Abdul-Mageed, Hassan Alhuzali, and Mohamed Elaraby. 2018. You tweet what you speak: A city-level dataset of Arabic dialects. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2021a. 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.

Muhammad Abdul-Mageed, Chiyu Zhang, Houda Bouamor, and Nizar Habash. 2020a. NADI 2020: The first nuanced Arabic dialect identification shared task. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 97–110, Barcelona, Spain (Online). Association for Computational Linguistics.

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, Houda Bouamor, and Nizar Habash. 2021b. NADI 2021: The second nuanced Arabic dialect identification shared task. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 244–259, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, Houda Bouamor, and Nizar Habash. 2022. Nadi 2022: The third nuanced arabic dialect identification shared task. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 85–97.

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, and Lyle Ungar. 2020b. Toward micro-dialect identification in diaglossic and code-switched environments. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5855–5876, Online. Association for Computational Linguistics.

Rania Al-Sabbagh and Roxana Girju. 2012. YADAC: Yet another dialectal Arabic corpus. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2882–2889, Istanbul, Turkey. European Language Resources Association (ELRA).

Nora Al-Twairesh, Rawan N. Al-Matham, Nora Madi, Nada Almugren, Al-Hanouf Al-Aljmi, Shahad Alshalan, Raghad Alshalan, Nafla Alrumayyan, Shams Al-Manea, Sumayah Bawazeer, Nourah Al-Mutlaq, Nada Almanea, Waad Bin Huwaymil, Dalal Alqusair, Reem Alotaibi, Suha Al-Senaydi, and Abeer Alfutamani. 2018. SUAR: towards building a corpus for the saudi dialect. In *Fourth International Conference On Arabic Computational Linguistics, ACLING 2018, November 17-19, 2018, Dubai, United Arab Emirates*, volume 142 of *Procedia Computer Science*, pages 72–82. Elsevier.

Nada Almarwani and Samah Aloufi. 2023. SANA at NADI 2023 shared task: Ensemble of Layer-Wise BERT-based models for Dialectal Arabic Identification. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*.

Israa Alsarsour, Esraa Mohamed, Reem Suwaileh, and Tamer Elsayed. 2018. Dart: A large dataset of dialectal arabic tweets. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Maha J Althobaiti. 2020. Automatic Arabic dialect identification systems for written texts: A survey. *arXiv preprint arXiv:2009.12622*.

Maha J. Althobaiti. 2022. Creation of annotated country-level dialectal Arabic resources: An unsupervised approach. *Nat. Lang. Eng.*, 28(5):607–648.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. AraBERT: Transformer-based model for Arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France. European Language Resource Association.

Wissam Antoun, Fady Baly, and Hazem M. Hajj. 2021. Araelectra: Pre-training text discriminators for arabic language understanding. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop, WANLP 2021, Kyiv, Ukraine (Virtual), April 9, 2021*, pages 191–195. Association for Computational Linguistics.

Dilshod Azizov, Jiyong Lee, Siwei Liu, and Shangsong Liang. 2023. Frank at NADI 2023 Shared Task: Trio-Based Ensemble Approach for Arabic Dialect Identification. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*.

MS Badawi. 1973. Levels of contemporary Arabic in Egypt. *Cairo: Dâr al Ma'ârif*.

Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2022. Hierarchical aggregation of dialectal data for Arabic dialect identification. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference, LREC 2022, Marseille, France, 20-25 June 2022*, pages 4586–4596. European Language Resources Association.

Laith H Baniata, Seyoung Park, and Seong-Bae Park. 2018. A neural machine translation model for arabic dialects that utilizes multitask learning (mtl). *Computational intelligence and neuroscience*, 2018.

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).

Houda Bouamor, Sabit Hassan, and Nizar Habash. 2019. The MADAR shared task on Arabic fine-grained dialect identification. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 199–207, Florence, Italy. Association for Computational Linguistics.

Kristen Brustad. 2000. *The Syntax of Spoken Arabic: A Comparative Study of Moroccan, Egyptian, Syrian, and Kuwaiti Dialects*. Georgetown University Press.

Mark W. Cowell. 1964. *A Reference Grammar of Syrian Arabic*. Georgetown University Press, Washington, D.C.

Kareem Darwish, Hamdy Mubarak, Ahmed Abdelali, Mohamed Eldesouki, Younes Samih, Randah Alharbi, Mohammed Attia, Walid Magdy, and Laura Kallmeyer. 2018. Multi-dialect Arabic POS tagging: A CRF approach. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Wiem Derouich, Sameh Kchaou, and Rahma Boujelbane. 2023. ANLP-RG at NADI 2023 shared task: Machine Translation of Arabic Dialects: A Comparative Study of Transformer Models. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*.

Kshitij Deshpande, Yash Patwardhan, Vedant Deshpande, and Sudeep Mangalvedhekar. 2023. Mavericks at NADI 2023 Shared Task: Unravelling Regional Nuances through Dialect Identification using Transformer-based Approach. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*.

Mona Diab, Nizar Habash, Owen Rambow, Mohamed Altantawy, and Yassine Benajiba. 2010. COLABA: Arabic dialect annotation and processing. In *LREC workshop on Semitic language processing*, pages 66–74.

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

Mahmoud El-Haj. 2020. Habibi - a multi dialect multi national Arabic song lyrics corpus. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1318–1326, Marseille, France. European Language Resources Association.

Shorouk Adel El-sayed and Noureldin Elmadany. 2023. ISL-AAST at NADI 2023 shared task: Enhancing Arabic Dialect Identification in the Era of Globalization and Technological Progress. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*.

Heba Elfardy, Mohamed Al-Badrashiny, and Mona Diab. 2014. AIDA: Identifying code switching in informal Arabic text. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 94–101, Doha, Qatar. Association for Computational Linguistics.

Mohab Elkaref, Movina Moses, Shinnosuke Tanaka, James Barry, and Geeth De Mel. 2023. NLPeople at NADI 2023 Shared Task: Arabic Dialect Identification with Augmented Context and Multi-Stage Tuning. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*.

AbdelRahim Elmadany, ElMoatez Billah Nagoudi, and Muhammad Abdul-Mageed. 2023. ORCA: A challenging benchmark for Arabic language understanding. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9559–9586, Toronto, Canada. Association for Computational Linguistics.

Wael Farhan, Bashar Talafha, Analle Abuammar, Ruba Jaikat, Mahmoud Al-Ayyoub, Ahmad Bisher Tarakji, and Anas Toma. 2020. Unsupervised dialectal neural machine translation. *Information Processing & Management*, 57(3):102181.

Hassan Gadalla, Hanaa Kilany, Howaida Arram, Ashraf Yacoub, Alaa El-Habashi, Amr Shalaby, Krisjanis Karins, Everett Rowson, Robert MacIntyre, Paul Kingsbury, David Graff, and Cynthia McLemore. 1997. CALLHOME Egyptian Arabic transcripts LDC97T19. Web Download. Philadelphia: Linguistic Data Consortium.

Imane Guellil, Faical Azouaou, and Mourad Abbas. 2017. Neural vs statistical translation of algerian arabic dialect written with arabizi and arabic letter. In *The 31st Pacific Asia Conference on Language, Information and Computation PACLIC*, volume 31, page 2017.

Beliz Gunel, Jingfei Du, Alexis Conneau, and Veselin Stoyanov. 2021. Supervised contrastive learning for pre-trained language model fine-tuning. In *9th International Conference on Learning Representations,* *ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Nizar Habash, Houda Bouamor, Hazem Hajj, Walid Magdy, Wajdi Zaghouani, Fethi Bougares, Nadi Tomeh, Ibrahim Abu Farha, and Samia Touileb, editors. 2021. *Proceedings of the Sixth Arabic Natural Language Processing Workshop*. Association for Computational Linguistics, Kyiv, Ukraine (Virtual).

Nizar Y Habash. 2010. *Introduction to Arabic natural language processing*, volume 3. Morgan & Claypool Publishers.

Salima Harrat, Karima Meftouh, Mourad Abbas, and Kamel Smaïli. 2014. Building resources for algerian arabic dialects. In *INTERSPEECH 2014, 15th Annual Conference of the International Speech Communication Association, Singapore, September 14-18, 2014*, pages 2123–2127. ISCA.

R.S. Harrell. 1962. *A Short Reference Grammar of Moroccan Arabic: With Audio CD*. Georgetown classics in Arabic language and linguistics. Georgetown University Press.

Yash A. Hatekar and Muhammad S. Abdo. 2023. IUNADI: Country-level Arabic Dialect Classification in Tweets for the Shared Task NADI 2023. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*.

Clive Holes. 2004. *Modern Arabic: Structures, Functions, and Varieties*. Georgetown Classics in Arabic Language and Linguistics. Georgetown University Press.

Elsayed Issa, Mohammed AlShakhori1, Reda Al-Bahrani, and Gus Hahn-Powell. 2021. Country-level Arabic dialect identification using RNNs with and without linguistic features. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 276–281, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Mustafa Jarrar, Nizar Habash, Faeq Alrimawi, Diyam Akra, and Nasser Zalmout. 2016. Curras: an annotated corpus for the Palestinian Arabic dialect. *Language Resources and Evaluation*, pages 1–31.

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. *arXiv preprint arXiv:2308.03051*.

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).

Abdullah Khered, Ingy Yasser Abdelhalim, Nadine Abdelhalim, Ahmed Soliman, and Riza Batista-Navarro. 2023. UniManc at NADI 2023 Shared Task: A Comparison of Various T5-based Models for Translating Arabic Dialectical Text to Modern Standard Arabic. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*.

Olli Kuparinen, Aleksandra Miletić, and Yves Scherrer. 2023. The Helsinki-NLP Submissions at NADI 2023 Shared Task: Walking the Baseline. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*.

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)*.

Mohamed Lichouri, Khaled Lounnas, Aicha Zitouni, Houda Latrache, and Rachida Djeradi. 2023. USTHB at NADI 2023 shared task: Exploring Preprocessing and Feature Engineering Strategies for Arabic Dialect Identification. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*.

Shervin Malmasi, Marcos Zampieri, Nikola Ljubešić, Preslav Nakov, Ahmed Ali, and Jörg Tiedemann. 2016. Discriminating between similar languages and Arabic dialect identification: A report on the third DSL shared task. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pages 1–14, Osaka, Japan. The COLING 2016 Organizing Committee.

Karima Meftouh, Salima Harrat, Salma Jamoussi, Mourad Abbas, and Kamel Smaili. 2015. Machine translation experiments on PADIC: A parallel Arabic DIalect corpus. In *Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation*, pages 26–34, Shanghai, China.

Hamdy Mubarak and Kareem Darwish. 2014. Using Twitter to collect a multi-dialectal corpus of Arabic. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 1–7, Doha, Qatar. Association for Computational Linguistics.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M. Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual generalization through multitask finetuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 15991–16111. Association for Computational Linguistics.

El Moatez Billah Nagoudi, Ahmed El-Shangiti, AbdelRahim Elmadany, and Muhammad Abdul-Mageed. 2023. Dolphin: A challenging and diverse benchmark for arabic nlg. *arXiv preprint arXiv:2305.14989*.

El Moatez Billah Nagoudi, AbdelRahim Elmadany, and Muhammad Abdul-Mageed. 2021. Investigating code-mixed Modern Standard Arabic-Egyptian to English machine translation. In *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*, pages 56–64, Online. 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.

Abdusalam Nwesri, Nabila Shinbir, and Hasan Hasan. 2023. UoT at NADI 2023 shared task: Automatic Arabic Dialect Identification is Made Possible. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*.

Ossama Obeid, Mohammad Salameh, Houda Bouamor, and Nizar Habash. 2019. ADIDA: Automatic dialect identification for Arabic. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 6–11, Minneapolis, Minnesota. Association for Computational Linguistics.

Ossama Obeid, Nasser Zalmout, Salam Khalifa, Dima Taji, Mai Oudah, Bashar Alhafni, Go Inoue, Fadhl Eryani, Alexander Erdmann, and Nizar Habash. 2020. CAMeL tools: An open source python toolkit for Arabic natural language processing. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 7022–7032, Marseille, France. European Language Resources Association.

Fatiha Sadat, Farzindar Kazemi, and Atefeh Farzindar. 2014. Automatic identification of Arabic language varieties and dialects in social media. In *Proceedings of the Second Workshop on Natural Language Processing for Social Media (SocialNLP)*, pages 22–27, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

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.

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.

Mohammad Salameh, Houda Bouamor, and Nizar Habash. 2018. Fine-grained Arabic dialect identification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1332–1344, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Wael Salloum, Heba Elfardy, Linda Alamir-Salloum, Nizar Habash, and Mona Diab. 2014. Sentence level dialect identification for machine translation system selection. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 772–778.

Hariram Veeramani, Surendrabikram Thapa, and Usman Naseem. 2023. DialectNLU at NADI 2023 Shared Task: Transformer Based MultiTask Approach Jointly Integrating Dialect and Machine Translation Tasks. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*.

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.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*.

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, Online. Association for Computational Linguistics.

Wajdi Zaghouani and Anis Charfi. 2018. Arap-tweet: A large multi-dialect Twitter corpus for gender, age and language variety identification. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Omar F. Zaidan and Chris Callison-Burch. 2011. The Arabic online commentary dataset: an annotated dataset of informal Arabic with high dialectal content. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 37–41, Portland, Oregon, USA. Association for Computational Linguistics.

Rabih Zbib, Erika Malchiodi, Jacob Devlin, David Stallard, Spyros Matsoukas, Richard Schwartz, John Makhoul, Omar 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.

Imed Zitouni, Muhammad Abdul-Mageed, Houda Bouamor, Fethi Bougares, Mahmoud El-Haj, Nadi Tomeh, and Wajdi Zaghouani, editors. 2020. *Proceedings of the Fifth Arabic Natural Language Processing Workshop*. Association for Computational Linguistics, Barcelona, Spain (Online).

DialectNLU at NADI 2023 Shared Task: Transformer Based Multitask Approach Jointly Integrating Dialect and Machine Translation Tasks in Arabic

Hariram Veeramani
Department of Electrical
and Computer Engineering,
UCLA, USA
hariram@ucla.edu

Surendrabikram Thapa
Department of Computer
Science, Virginia Tech,
Blacksburg, USA
sbt@vt.edu

Usman Naseem
College of Science and
Engineering, James Cook
University, Australia
usman.naseem@jcu.edu.au

Abstract

With approximately 400 million speakers worldwide, Arabic ranks as the fifth most-spoken language globally, necessitating advancements in natural language processing. This paper describes the approaches employed for the subtasks outlined in the Nuanced Arabic Dialect Identification (NADI) task at EMNLP 2023. We employ an ensemble of two Arabic language models for the first subtask involving closed country-level dialect identification classification. Similarly, for the second subtask, focused on closed dialect to Modern Standard Arabic (MSA) machine translation, our approach combines sequence-to-sequence models trained on an Arabic-specific dataset. Our team ranks 10th and 3rd on subtask 1 and subtask 2, respectively.

1 Introduction

The Arabic language, with approximately 400 million speakers across the globe, stands as the fifth most widely spoken language worldwide (Mohammed Ameen et al., 2023). Its vast linguistic diversity, rooted in rich historical and regional variations, necessitates continuous advancements in the field of natural language processing (NLP) (Abdul-Mageed et al., 2020). Within the scope of this linguistic diversity, Modern Standard Arabic (MSA) serves as the standardized form of the language, fostering communication across Arabic-speaking regions. However, coexisting with MSA are numerous dialects, each bearing its unique linguistic features and nuances (Abdul-Mageed et al., 2021b).

Arabic encompasses a wide array of languages and language variations, with some of them lacking mutual intelligibility (Abdul-Mageed et al., 2022a) (Veeramani et al., 2023d,c,b). Despite this diversity, Arabic is frequently misconceived as a single, uniform language. Thus, identifying these different dialects plays a pivotal role in the realm of Arabic language understanding, primarily due to the contextual intricacies they introduce (Salameh et al., 2018). Dialect identification serves as the bedrock for a multitude of NLP applications, enabling accurate language understanding, effective communication, sentiment analysis, and sociolinguistic insights (Malmasi et al., 2015; Veeramani et al., 2023e,a,f). Furthermore, dialect classification preserves and celebrates the rich linguistic diversity encapsulated within the Arabic language landscape (Zaidan and Callison-Burch, 2014; Salameh et al., 2018).

Similarly, machine translation, in particular, holds profound significance within this rich Arabic linguistic landscape (Kchaou et al., 2023). Bridging the gap between dialects and the standardized form of the language, MSA, and machine translation facilitates seamless communication across Arabic-speaking communities (Al-Ibrahim and Duwairi, 2020). In an interconnected world where communication knows no borders, machine translation becomes the vital bridge that transcends linguistic differences (Ameur et al., 2020).

In this paper, we address the pressing need for advancements in Arabic dialect identification and machine translation. Specifically, we present our contributions to the Nuanced Arabic Dialect Identification (NADI) task (Abdul-Mageed et al., 2023) at 1st ArabicNLP colacted with EMNLP 2023. Our work centers on two crucial subtasks:

- **Closed Country-Level Dialect Identification**: To tackle this subtask, we leverage an ensemble of two Arabic language models, harnessing the power of natural language processing to classify dialects at the country level.

- **Closed Dialect to MSA Machine Translation**: For this subtask, we employ a combination of sequence-to-sequence models, all meticulously trained on an Arabic-specific dataset. Our goal is to enhance the translation accuracy of Arabic dialects into the standardized MSA, thereby promoting effective

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 614–619
December 7, 2023 ©2023 Association for Computational Linguistics

cross-dialect communication.

This paper explains our systems in detail, offering comprehensive insights into our approach, the rationale behind our methodology, a thorough analysis of our results, and valuable insights derived from our findings.

2 Task Descriptions

We submitted results for the first two out of three subtasks.

Subtask 1: This subtask involves identifying the dialect of a given text, with a particular emphasis on Arabic dialects that lack well-defined linguistic conventions and structures. The evaluation metric for this subtask is the macro-averaged F1-score.

Subtask 2: This subtask entails translating non-MSA dialects into Modern Standard Arabic (MSA) across four specified dialects. Evaluation is based on the BLEU score.

3 Dataset

Subtask 1 focuses on informal Twitter discourse featuring languages from Arabic-speaking nations, including Qatar, Syria, Libya, Yemen, Kuwait, Morocco, UAE, Jordan, Palestine, Tunisia, Saudi Arabia, Egypt, Iraq, Algeria, Bahrain, Sudan, Lebanon, and Oman. The dataset, NADI-2023-TWT, consists of 18,000 training tweets (1,000 per country), 1,800 tweets in the dev dataset (100 per country), and 3,600 tweets in the test dataset. I

Subtask 2, on the other hand, provides a manually curated dataset focusing on four urban dialects: Egyptian, Emirati, Jordanian, and Palestinian. The training data primarily originates from the MADAR-parallel corpus (Bouamor et al., 2018), comprising 40,000 sentences. The test set consists of 2,000 sentences (500 from each dialect), while the dev set comprises 400 sentences (100 from each of the four dialects). It is important to note that we did not use any external data or augmentation.

4 System Description

4.1 Subtask 1

In addressing the classification problem of subtask 1, we employ a strategic combination of ARBERT and MARBERT (Abdul-Mageed et al., 2021a). ARBERT is trained on Modern Standard Arabic (MSA) data, whereas MARBERT specializes in learning from the informal dialects commonly found in Twitter data. This selection is grounded in the recognition that Arabic encompasses diverse language styles. ARBERT excels in comprehending formal Arabic, particularly MSA rules, making it proficient in handling general aspects of the language. On the other hand, MARBERT, fine-tuned on Twitter's informal dialects, adeptly captures the nuances of day-to-day expressions. We perform the combination with various strategies.

4.1.1 Max-voting Ensemble

As a first strategy, at the logit level, we implemented a weighted ensemble approach. ARBERT was assigned a weight of 0.4, while MARBERT received a weight of 0.6. This weighting strategy was adopted to optimize the ensemble's performance by capitalizing on the unique strengths of each model. The higher weight assigned to MARBERT reflects its proficiency in capturing informal nuances from Twitter data, ensuring robust and accurate dialect classification across diverse Arabic language variations encountered in online contexts.

4.1.2 Fusion Representation Technique

In our second strategy, we fuse the hidden representations of both ARBERT and MARBERT models to leverage their complementary strengths (Abdul-Mageed et al., 2022b), enhancing the model's ability to capture nuanced dialect features. Throughout this paper, this representation of dimensions 2×768 will be referred to as fusion representation. Following this, we incorporate a dropout layer (DO) to enhance the model's performance. This approach has proven to be the most effective model for subtask 1. Additionally, we also experiment by incorporating a label-aware technique (LAT) by appending the respective label to the beginning of the text input.

4.2 Subtask 2

For the machine translation challenge of subtask 2, we applied various models, including AraBART, AraT5 (base), and AraT5 (base-1024). We also choose an ensemble approach that combines dialect classifier and AraBART in two settings. This ensemble strategy was selected because each model has unique strengths and weaknesses. By merging them, we effectively mitigate these weaknesses, resulting in more precise and robust translations. The standalone models and ensemble approach are

Model	Dev Dataset				Test Dataset (F1 score)
	F1-score	Precision	Recall	Accuracy	
ARBERT	75.5	76.1	75.34	75.38	73.16
MARBERT	76.3	76.9	76.0	76.0	73.25
Max-voting Ensemble	77.5	79.6	77.2	77.2	77.19
Fusion Representation	79.75	80.97	79.66	79.66	79.06
Fusion Representation + DO + LAT	79.83	79.81	79.75	79.81	79.06
Fusion Representation + DO	**80.55**	**80.98**	**80.44**	**80.44**	**80.56**

Table 1: Performance of various model combinations for task 1 on dev and test dataset. All scores reported are macro-averaged scores. The abbreviations are introduced in section 4.1.2.

defined below.

4.2.1 Standalone Models

We use three different standalone models for our machine translation task. The models and the motivation for using them are explained below:

AraT5 (base): This sequence-to-sequence model is pre-trained on a substantial Arabic text corpus, encompassing Modern Standard Arabic (MSA) and Arabic dialects. This extensive training gives AraT5 (base) a profound understanding of Arabic grammar and vocabulary, prerequisites for accurate translation. AraT5 (base)[1] is also trained using a denoising-based pre-training methodology, which enhances its capacity to handle noisy data—an invaluable trait for machine translation, where source texts may contain errors or typos.

AraBART: This is another powerful machine translation model, albeit trained on a comparatively smaller corpus of Arabic text (Kamal Eddine et al., 2022). Because of its modeling architecture, AraT5 (base)[2] may have a superior grasp of Arabic grammar and vocabulary. Additionally, AraBART undergoes training with a distinct denoising-based pre-training method, potentially better suited for processing noisy data compared to AraT5.

AraT5 (base-1024): This variant of AraT5 benefits from training on an even larger corpus of text and boasts a more extensive vocabulary compared to AraT5 (base). Its broader lexicon and nuanced understanding of the Arabic language make AraT5 (base-1024)[3] particularly good at capturing subtleties in translation. Moreover, AraT5 (base-1024) features an extended sequence length and faster convergence during fine-tuning, expediting the training process.

4.2.2 Ensemble Approach

In our ensemble approach for Subtask 2, we leverage AraBART, a sequence-to-sequence classifier, in two distinct settings. In both settings, we initially fine-tuned AraBART for the dialect classification task. Subsequently, we remove the classifier head and perform an additional fine-tuning phase, focusing on sequence-to-sequence translation. The key difference between the two settings lies in the learning scheduler employed for AraBART. One setting utilizes a 'linear' learning scheduler, while the other adopts a 'cosine' learning scheduler. This variation in learning scheduler choice allows us to explore different training dynamics. When determining which translation to use in the ensemble, we opt for the model that excels in the specific task (Kanagasabai et al., 2023), dialect classification. This approach helps optimize the overall performance of the ensemble in accurately translating non-MSA dialects into Modern Standard Arabic.

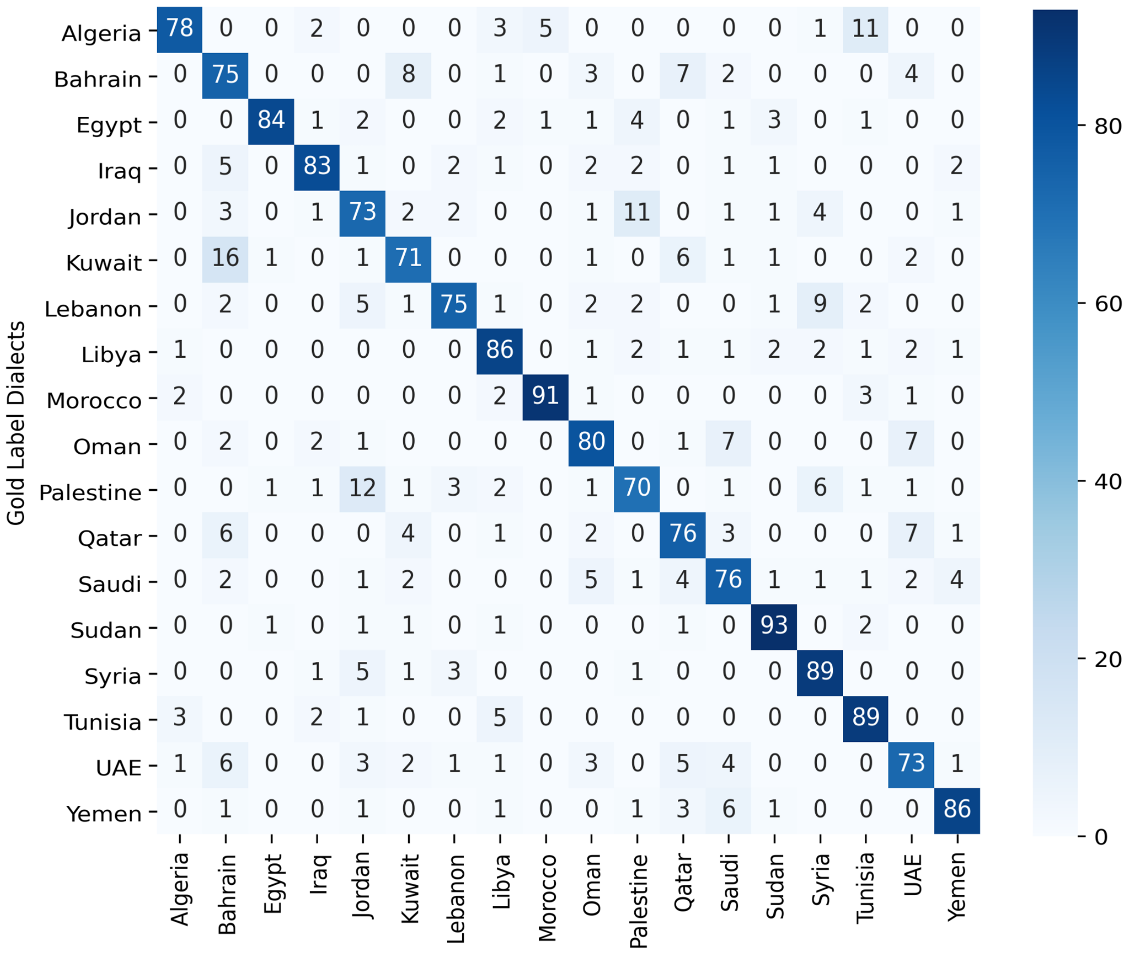
5 Results

Table 1 presents a comprehensive evaluation of various models and model combinations used for Subtask 1, focusing on dialect classification. It includes key metrics such as F1-score, precision, recall, and accuracy for both the development (Dev) and Test datasets. The Dev dataset serves as a validation set for fine-tuning, while the Test dataset represents the models' expected performance in real-world scenarios. Notably, ARBERT and MARBERT exhibit strong dialect classification capabilities on the Dev dataset, achieving scores of 75.5 and 76.3, respectively. The Max-voting Ensemble strategy enhances performance, yielding an F1-score of 77.5. Fusion Representation further elevates dialect identification with a score of 79.75. The Fusion Representation model with Dropout and Label-aware Training (LAT) attains an even higher performance on the Dev dataset, registering an F1-score of 79.83.

[1] https://huggingface.co/UBC-NLP/AraT5-base

[2] https://huggingface.co/moussaKam/AraBART

[3] https://huggingface.co/UBC-NLP/AraT5v2-base-1024



	Algeria	Bahrain	Egypt	Iraq	Jordan	Kuwait	Lebanon	Libya	Morocco	Oman	Palestine	Qatar	Saudi	Sudan	Syria	Tunisia	UAE	Yemen
Algeria	78	0	0	2	0	0	0	3	5	0	0	0	0	0	1	11	0	0
Bahrain	0	75	0	0	0	8	0	1	0	3	0	7	2	0	0	0	4	0
Egypt	0	0	84	1	2	0	0	2	1	1	4	0	1	3	0	1	0	0
Iraq	0	5	0	83	1	0	2	1	0	2	2	0	1	1	0	0	0	2
Jordan	0	3	0	1	73	2	2	0	0	1	11	0	1	1	4	0	0	1
Kuwait	0	16	1	0	1	71	0	0	0	1	0	6	1	1	0	0	2	0
Lebanon	0	2	0	0	5	1	75	1	0	2	2	0	0	1	9	2	0	0
Libya	1	0	0	0	0	0	0	86	0	1	2	1	1	2	2	1	2	1
Morocco	2	0	0	0	0	0	0	2	91	1	0	0	0	0	0	3	1	0
Oman	0	2	0	2	1	0	0	0	0	80	0	1	7	0	0	0	7	0
Palestine	0	0	1	1	12	1	3	2	0	1	70	0	1	0	6	1	1	0
Qatar	0	6	0	0	0	4	0	1	0	2	0	76	3	0	0	0	7	1
Saudi	0	2	0	0	1	2	0	0	0	5	1	4	76	1	1	1	2	4
Sudan	0	0	1	0	1	1	0	1	0	0	0	1	0	93	0	2	0	0
Syria	0	0	0	1	5	1	3	0	0	0	1	0	0	0	89	0	0	0
Tunisia	3	0	0	2	1	0	0	5	0	0	0	0	0	0	0	89	0	0
UAE	1	6	0	0	3	2	1	1	0	3	0	5	4	0	0	0	73	1
Yemen	0	1	0	0	1	0	0	1	0	0	1	3	6	1	0	0	0	86

Figure 1: Confusion matrix for our best-performing model for country-wise dialect classification (task 1). We use the dev set data for this analysis.

The Fusion Representation model with Dropout stands out as the top-performing model, achieving an impressive F1-score of 80.55. A similar trend can be seen with the test dataset as well. Our fusion representation model performs the best in the test dataset with an F1-score of 80.56.

To provide deeper insights into our models' performance, Figure 1 presents confusion matrices for all 18 dialects in Subtask 1. These matrices offer a detailed breakdown of classification results, shedding light on how well our models identify each dialect. Notably, Sudanese and Moroccan dialects exhibit a strong classification, while some challenges persist in accurately classifying Palestinian and Kuwaiti dialects. The confusion matrix serves as a valuable tool for understanding model performance in specific dialects and identifying areas for improvement, further enriching our analysis of dialect classification results.

Similarly, Table 2 comprehensively compare model performance in Subtask 2, focusing on the machine translation of non-Modern Standard Arabic (MSA) dialects into MSA. The models assessed include AraT5 (base), which achieves a BLEU score of 0.54 on the evaluation dataset and 0.014 on the test dataset, indicating translation challenges. AraT5 (base-1024) exhibits improvement with BLEU scores of 1.03 and 0.07 on the evaluation and test datasets, respectively. Among standalone models, AraBART excels with high BLEU

Model	Eval BLEU	Test BLEU
AraT5 (base)	0.54	0.014
AraT5 (base-1024)	1.03	0.07
AraBART	12.01	13.42
Ensemble Approach	**12.9**	**13.43**

Table 2: Performance of various standalone models along with our ensemble approach for machine translation subtask. Overall BLEU scores are presented for both eval and test datasets.

scores, achieving 12.01 on the evaluation dataset and 13.42 on the test dataset, showcasing its proficiency in accurate dialect translation. Most notably, our novel ensemble approach outperforms individual standalone models, achieving the highest BLEU scores of 12.9 on the evaluation dataset and 13.43 on the test dataset, highlighting the efficacy of ensemble techniques in enhancing machine translation quality for Arabic dialects.

In Subtask 1, focused on dialect identification and Subtask 2, addressing machine translation, ensemble techniques (fusion-level or decision-level) have consistently demonstrated outstanding performance. By strategically combining multiple models, we have harnessed the collective strengths of various standalone models to achieve remarkable results. This underscores the pivotal role of ensemble methodologies in enhancing the accuracy and robustness of Arabic dialect identification and machine translation, reaffirming their effectiveness across diverse linguistic challenges.

6 Conclusion

In conclusion, our participation in the Nuanced Arabic Dialect Identification (NADI) task at EMNLP 2023 has demonstrated the effectiveness of innovative approaches in addressing the intricate challenges posed by Arabic dialect identification and machine translation. With its diverse linguistic landscape, Arabic presents a unique and formidable set of hurdles for natural language processing tasks. The high performance of ensemble strategies that involve carefully combining various models has showcased remarkable achievements in dialect classification and machine translation, underlining the power of ensemble techniques. Furthermore, our contributions extend beyond performance metrics, encompassing comprehensive system descriptions, model rationale, and insights from experimentation. These approaches pave the way for tackling the multi-aspects challenges of Arabic NLP forward.

Ethics Statement

It is important to acknowledge that this research does not include a comprehensive assessment of potential bias in the models deployed. Before real-world applications, models should undergo thorough bias assessments to ensure fairness and equity in their predictions. We encourage future research and practitioners to consider bias assessments as an integral part of deploying these models in practical settings, emphasizing ethical AI practices and responsible AI development.

Limitations

While our approaches have shown promising results, several limitations are worth noting. First, our ensemble strategies, while effective, are computationally intensive and require substantial resources. Implementing these approaches at scale may pose challenges in resource-constrained environments. Second, our models' performance may be influenced by the availability and quality of training data. The scarcity of annotated data for some Arabic dialects could impact the generalization of our models. Additionally, our current strategies primarily focus on closed-track evaluations; extending them to open-domain scenarios remains an avenue for future exploration. Finally, as the field of natural language processing evolves rapidly, newer models and techniques may offer even more robust solutions in Arabic dialect identification and machine translation, necessitating ongoing research and adaptation.

References

Muhammad Abdul-Mageed, AbdelRahim Elmadany, Chiyu Zhang, ElMoatez Billah Nagoudi, Houda Bouamor, and Nizar Habash. 2023. NADI 2023: The Fourth Nuanced Arabic Dialect Identification Shared Task. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*.

Muhammad Abdul-Mageed, AbdelRahim Elmadany, et al. 2021a. 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, Barcelona, Spain (Online). Association for Computational Linguistics.

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, Houda Bouamor, and Nizar Habash. 2021b. NADI 2021: The second nuanced Arabic dialect identification shared task. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 244–259, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, Houda Bouamor, and Nizar Habash. 2022a. Nadi 2022: The third nuanced arabic dialect identification shared task. *WANLP 2022*, page 85.

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, Houda Bouamor, and Nizar Habash. 2022b. NADI 2022: The third nuanced Arabic dialect identification shared task. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Roqayah Al-Ibrahim and Rehab M Duwairi. 2020. Neural machine translation from jordanian dialect to modern standard arabic. In *2020 11th International Conference on Information and Communication Systems (ICICS)*, pages 173–178. IEEE.

Mohamed Seghir Hadj Ameur, Farid Meziane, and Ahmed Guessoum. 2020. Arabic machine translation: A survey of the latest trends and challenges. *Computer Science Review*, 38:100305.

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).

Moussa Kamal Eddine, Nadi Tomeh, Nizar Habash, Joseph Le Roux, and Michalis Vazirgiannis. 2022. AraBART: a pretrained Arabic sequence-to-sequence model for abstractive summarization. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 31–42, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Rajaraman Kanagasabai, Saravanan Rajamanickam, Hariram Veeramani, Adam Westerski, and Kim Jung Jae. 2023. Semantists at ImageArg-2023: Exploring Cross-modal Contrastive and Ensemble Models for Multimodal Stance and Persuasiveness Classification. In *Proceedings of the 10th Workshop on Argument Mining*, Online and in Singapore. Association for Computational Linguistics.

Saméh Kchaou, Rahma Boujelbane, and Lamia Hadrich. 2023. Hybrid pipeline for building arabic tunisian dialect-standard arabic neural machine translation model from scratch. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22(3):1–21.

Shervin Malmasi, Eshrag Refaee, and Mark Dras. 2015. Arabic dialect identification using a parallel multidialectal corpus. In *International conference of the pacific association for computational linguistics*, pages 35–53. Springer.

Zinah J Mohammed Ameen, Abdulkareem Abdulrahman Kadhim, et al. 2023. Deep learning methods for arabic autoencoder speech recognition system for electro-larynx device. *Advances in Human-Computer Interaction*, 2023.

Mohammad Salameh, Houda Bouamor, and Nizar Habash. 2018. Fine-grained arabic dialect identification. In *Proceedings of the 27th international conference on computational linguistics*, pages 1332–1344.

Hariram Veeramani, Surendrabikram Thapa, and Usman Naseem. 2023a. Automated Citation Function Classification and Context Extraction in Astrophysics: Leveraging Paraphrasing and Question Answering. In *Proceedings of the second Workshop on Information Extraction from Scientific Publications*, Online. Association for Computational Linguistics.

Hariram Veeramani, Surendrabikram Thapa, and Usman Naseem. 2023b. Enhancing ESG Impact Type Identification through Early Fusion and Multilingual Models. In *Proceedings of the Sixth Workshop on Financial Technology and Natural Language Processing (FinNLP)*. Association for Computational Linguistics.

Hariram Veeramani, Surendrabikram Thapa, and Usman Naseem. 2023c. KnowTellConvince at ArAIEval Shared Task: Disinformation and Persuasion Detection in Arabic using Similar and Contrastive Representation Alignment. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore. Association for Computational Linguistics.

Hariram Veeramani, Surendrabikram Thapa, and Usman Naseem. 2023d. LowResContextQA at Qur'an QA 2023 Shared Task: Temporal and Sequential Representation Augmented Question Answering Span Detection in Arabic. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore. Association for Computational Linguistics.

Hariram Veeramani, Surendrabikram Thapa, and Usman Naseem. 2023e. LowResourceNLU at BLP-2023 Task 1 2: Enhancing Sentiment Classification and Violence Incitement Detection in Bangla Through Aggregated Language Models. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Hariram Veeramani, Surendrabikram Thapa, and Usman Naseem. 2023f. Temporally Dynamic Session-Keyword Aware Sequential Recommendation system. In *2023 International Conference on Data Mining Workshops (ICDMW)*.

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

The University of Tripoli at NADI 2023 shared task: Automatic Arabic Dialect Identification is Made Possible

Abdusalam F A Nwesri
University of Tripoli
Tripoli - Libya
a.nwesri@uot.edu.ly

Nabila A S Shinbir
Tripoli College of Sci. & Tech.
Tripoli - Libya
shinbir@tcst.edu.ly

Hassan A H Ebrahem
University of Tripoli
Tripoli - Libya
h.ebrahem@uot.edu.ly

Abstract

In this paper we present our approach towards Arabic Dialect identification which was part of the The Fourth Nuanced Arabic Dialect Identification Shared Task (NADI 2023). We tested several techniques to identify Arabic dialects. We obtained the best result by fine-tuning the pre-trained MARBERTv2 model with a modified training dataset. The training set was expanded by sorting tweets based on dialects, concatenating every two adjacent tweets, and adding them to the original dataset as new tweets. We achieved 82.87 on F1 score and we were at the seventh position among 16 participants.

1 Introduction

Arabic dialects are different spoken versions of Modern Standard Arabic (MSA) which become to increasingly emerge in a written format recently. Although Arabic dialects have common linguistic features with MSA, they have different features where NLP tools used for MSA fail to work properly. There are more than 27 Arabic dialects (Elgabou and Kazakov, 2017) which need different NLP techniques than those used for MSA. It was proven that NLP tools for MSA is less efficient with Arabic dialects (Khalifa et al., 2016). As such, it is crucial to identify a dialect version in order to properly apply proper NLP techniques on it.

Arabic dialect identification is very challenging for several reasons. First, Arabic dialects are all originating from MSA and share common features and words. As MSA is the formal language of writing across Arabic countries, writing dialectal phrases are usually mixed with MSA complete phrases. Furthermore, dialectal Arabic has no official spelling standards and usually written differently by different people (Darwish et al., 2021).

Second, With the absence of short vowels (diacritics) in Arabic text, it is hard to know the phrase dialect, for example, the word إنتِ /enti/ (you) in Tunisian dialect is used to address both a Masculine or a feminine third person, while إنتَ /enta/ is used to address a masculine and إنتِ /enti/ to address a feminine third person in several other dialects, while in MSA أنتَ /anta/ and أنتِ /anti/ are used respectively for the same purpose.

Third, tweets are usually short and in many cases it is hard not only for a learning model, but for an Arabic reader to guess the dialect of the tweet based on its words.

Previous work on Arabic dialect identification were mostly carried out through the Nuanced Arabic Dialect Identification (NADI) shared tasks series (Abdul-Mageed et al., 2020, 2021b, 2022). The goal of these shared tasks is to improve dialect identification and other dialect processing tasks such as sentiment analysis and machine translation from dialects to MSA. The organizers provide required resources such as datasets to participants who carry research on those tasks. The forth Nuanced Arabic Dialect Identification (Abdul-Mageed et al., 2023) has three subtasks:

- Subtask 1 (Closed Country-level Dialect ID): dialect identification using provided datasets only. No External datasets should be used.
- Subtask 2 (Closed Dialect to MSA MT): Sentence-level machine translation from Egyptian, Emirati, Jordanian, and Palestinian dialects to MSA using only provided training data.
- Subtask 3 (Open Dialect to MSA MT): Sentence-level machine translation from Egyptian, Emirati, Jordanian, and Palestinian dialects to MSA using provided training data and any publicly available datasets.

We participated in Subtask 1 only. We tested several machine learning and deep learning models which we report in this paper.

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 620–624
December 7, 2023 ©2023 Association for Computational Linguistics

Dataset	Type	Dialects	Tweets
MADAR-2018	Imbalanced	15	40K
NADI-2020	Imbalanced	17	19.3K
NADI-2021	Imbalanced	17	19.7K
NADI-2023	balanced	18	18K

Table 1: Subtask 1 training datasets provided by NADI 2023.

The remaining part of this paper is structured as follows: Section 2 describes the data used in our experiments, Section 3 describes our experiments and proposed systems, and in Section 4 we present our results proceeded by our discussions and conclusion.

2 Data

For Subtask 1, the organizers provided a 23.4k tweets dataset that covers 18 dialects. the dataset is split into 18k training set, 1.8k development set and 3.6k test set. Extra datasets was also provided and can be used by participants. Particularly, data used in previous NADI competitions plus the MADAR dataset (Bouamor et al., 2018). As a closed-country subtask, participants were not allowed to use other external data to train their systems. Datasets and their size are presented in Table 1.

3 Experiments

We run several experiments using both machine learning and deep learning models. We determined our baseline and officially submit the best three outputs of our systems to be scored on the leaderboard.

3.1 Machine Learning Models

We tested several Machine Learning classifiers, namely: Multi-layer perceptron classifier (MLPClassifier), Support Vector Machines (SVC), Naive Bayes classifier for multivariate Bernoulli models (BernoulliNB), and Naive Bayes classifier for multinomial models (MultinomialNB) (Pedregosa et al., 2011). For each model, we calculate the Accuracy (A), Precision (P), Recall (R), and the normal F1-measure. We obtained best results on the original training dataset after normalizing text and removing non-Arabic characters. Results are shown in Table 2.

Classifier	F1	A	P	R
SVC	0.60	0.61	0.61	0.60
MLPClassifier	0.62	0.63	0.62	0.62
MultinomialNB	**0.63**	**0.64**	0.64	**0.64**
BernoulliNB	0.58	0.56	**0.67**	0.56

Table 2: Results obtained using Machine Learning classifiers on the training datasets.

Classifier	F1	A	P	R
SVC	0.59	0.58	0.60	0.58
MLPClassifier	0.61	0.61	0.61	0.61
MultinomialNB	**0.62**	**0.62**	0.63	lbf 0.62
BernoulliNB	0.55	0.53	**0.66**	0.53

Table 3: Results obtained using Machine Learning classifiers on the training datasets when removing stopwords.

We also removed a list of known stopwords in Arabic and used the Snowball stemmer [1] on the original datasets, but results dropped down in both cases. Table 3 shows the results when using stopwords and Table 4 shows results using both stopwords and stemming.

3.2 Transformer Based Models

It was reported that deep learning techniques are superior to machine learning models. The introduction of transformers based approaches have significantly improved results of NLP tasks such as text classification (Chang et al., 2020). Transformers allow building proficient language models that can be fine-tuned for a specific task. The introduction of Bidirectional Encoder Representations from Transformers (BERT) model (Devlin et al., 2019) by google AI Language resulted in the stat-of-the-art results in a wide variety of NLP tasks. Several versions based on this model have been developed for Arabic Language including AraBERT (Antoun et al., 2021) and MARBERT (Abdul-Mageed et al., 2021a). Results in (Abdul-Mageed et al., 2021a) show that MARBERTv2 was superior to ARBERT, and AraBERT in an Arabic dialect iden-

Classifier	F1	A	P	R
SVC	0.57	0.56	0.58	0.56
MLPClassifier	0.55	0.55	0.56	0.55
MultinomialNB	**0.59**	**0.59**	0.60	**0.59**
BernoulliNB	0.55	0.53	**0.62**	0.53

Table 4: Results obtained using Machine Learning classifiers on the training datasets when removing stopwords and using the Snowball stemmer.

[1] https://pypi.org/project/snowballstemmer/

Model	F1	A	P	R
MARBERTv2	**84.47**	**84.39**	**84.87**	**80.39**
arabertv02	79.31	79.22	79.62	79.22

Table 5: Baseline results by fine-tuning both MARBERTv2 and bert-base-arabertv02-twitter models using the training and the development datasets.

Model	F1	A	P	R
MARBERTv2	**80.12**	**80.00**	**80.60**	**80.00**
arabertv02	76.44	76.44	76.75	76.44

Table 6: The effects of pre-processing tweets on the Baseline results.

Model	F1	A	P	R
MARBERTv2	**83.19**	**83.11**	**83.36**	**83.11**
arabertv02	78.12	78.06	78.28	78.06

Table 7: The effects of stopwords removal on the Baseline results.

tification task. We decided to use MARBERTv2 and AraBERT as our baseline models since they were trained on different datasets and were reported to achieve better result than other models.

3.3 Baseline

We run the script provided by the organizers and fine-tuned the "UBC-NLP/MARBERTv2" and the "aubmindlab/bert-base-arabertv02-twitter" models with the initial following parameters: maximum sequence length is set to 256, training batch is set to 32, learning rate is set to 1e-5 ,and number of epochs is set to 3. We used the training dataset for training and the development set for testing. Scores are then calculated using Accuracy (A), macro average precision (P), macro average recall (R), and macro average F1 (F1) using the provided script. The identification scores is shown in Table 5.

We run several experiments using the baseline models in order to obtain better results than the baseline.

3.4 Pre-processing

We pre-processed the training and the development datasets by removing any non-Arabic characters including emojis, and URLs from tweets; reducing repeated characters to two occurrences; and normalizing the different shapes of Arabic letters such as "اإأآ", "ي", and "ة" to "ا", "ى", and "ه" respectively.

The pre-processed datasets are used to fine-tune our baseline models. This step negatively affected our baseline. Results are shown in Table 6.

3.5 Stop-words Removal

Based on the idea that dialects share the same words originated from MSA, we calculated the frequency of the top 50 words in the training dataset and considered them as our stopwords list. Applying stopwords removal on our baseline decreased our scores as shown in Table 7.

3.6 Tweets Expansion

Our officially reported results came by increasing the tweets length. The idea comes from the fact that a human who reads one sentence, might not be able to recognize a writer's dialect until reading another one. As tweets are usually short with a minimum of three words in the case of our dataset, we made new longer tweets by sorting tweets based on their dialect, and then combining every two adjacent tweets belong to the same dialect together adding the combination to the dataset. The new dataset contains 35898 tweets with a maximum tweet length of 540.

We fine-tuned the pre-trained "UBC-NLP/MARBERTv2" model using the new generated dataset. We set the maximum sequence length to 512, the training batch to 32, and number of epochs to 3. We used the default values for the learning rate. The model was first fine-tuned on a 16GB RAM with core i5 processor. It took around 6 hours to complete. However, using the google Colab T4 GPU (Bisong, 2019), it only took 30 minutes to finish. This technique achieved the best score that was above our baseline. The results are shown in Table 8 as UoT-1 (UoT stands for the University of Tripoli, the name of our team).

We have also run the same experiment (labeled UoT-2) on the same dataset, however, we applied the above mentioned pre-processing technique on the new dataset. This action caused scores to drop below the baseline.

The third run we submitted (Uot-3) is similar to UoT-1, however, the fine-tuning was done using the "aubmindlab/bert-base-arabertv02-twitter" pre-trained model.

We finally run the unlabeled testset against our models and submitted our predictions to leaderboard. Table 9 shows the results of our system using the testset as officially reported by the organisers.

Run	F1	A	P	R
UoT-1	**84.70**	**84.67**	**85.01**	**84.67**
UoT-2	80.64	80.61	80.93	80.61
UoT-3	80.38	80.39	80.54	80.39

Table 8: Results obtained using tweet expansion using the training and Development datasets.

Run	F1	A	P	R
UoT-1	82.87	82.86	83.17	82.68
UoT-2	80.70	80.69	81.18	80.69
UoT-3	74.45	74.44	75.01	74.44

Table 9: Official results in the leaderboard using the output of our systems with the unlabeled testset.

Table 10 shows our best result among the participating teams.

Team	F1	A	P	R
NLPeople	87.27	87.22	87.37	87.22
rematchka	86.18	86.17	86.29	86.17
Arabitools	85.86	85.81	86.10	85.81
SANA	85.43	85.39	85.60	85.39
Frank	84.76	84.75	84.95	84.75
ISL-AAST	83.73	83.67	83.87	83.67
UoT	82.87	82.86	83.17	82.86
AIC	82.37	82.42	82.57	82.42
Cordyceps	82.17	82.14	82.57	82.14
DialectNLU	80.56	80.50	80.92	80.50
Mavericks	76.65	76.47	77.43	76.47
exa	70.72	71.03	72.26	71.03
IUNADI	70.22	70.78	71.32	70.78
NAYEL	63.09	63.39	63.30	63.39
ustdb	62.51	62.17	63.07	62.17
Frau. IAIS	29.91	33.14	38.47	31.39

Table 10: The leaderboard showing our scores in the seventh position (UoT) among participating teams.

4 Discussion

Dialect Identification of a written text is uneasy task. By going through tweets in the development dataset, We found a considerable overlap between regional dialects which is natural, for example Gulf dialects usually overlap and are miss judged by language models. for example, Saudi-Arabian dialect overlaps with Qatar, UAE, and Omani dialects. And Maghrebi dialects such as Tunisian are falsely judged as Algeria and Libyan tweets only; and Levantine dialects such as Syrian are falsely judged as Lebanese, Jordanian, and Palestinian tweets. The best judgement was achieved on Moroccan dialect with only 3 tweets judged as Tunisian and one as Palestinian. False predicted tweets are usually short and are hard for a human to judge. For instance, "سكر الباب وراك" meaning "close the door behind you", is a Kuwaiti tweet which is falsely judged as Egyptian. This tweet can also be Libyan and it is hard to detect its origin dialect. That is why our approach was beneficial in clarifying such tweets. Expanding tweets should be explored further. for instance expanding the dataset with a combination of only shorter tweets within the same dialect.

We expected that pre-processing would improve identification as it cleans text, however, for dialects it did not. After deep analysis of the training dataset, we realized that removing none Arabic characters and normalization should be handled carefully as there are several Arabic tweets written in Farsi characters which fall out of the range of Arabic characters. For example removing characters such as "گ" which is used to represent "ك" in the word "ملگت" would leave the word "مل ت" in the tweet. Such mistake should be corrected by normalizing the letter "گ" to "ك" in the tweets.

5 Conclusions

We used several machine learning classifiers and pre-trained language models to identify Arabic dialects. We also showed the affects of pre-processing, stemming and sotpwords removal on the identification results. our best results are obtained using two pre-trained Models namely: the MARBERTv2 Model and the AraBERT model. We fine-tuned those models with an expanded version of the training dataset. This approach resulted in improving our baseline and put us in the seventh position among 16 participating teams in the Fourth Nuanced Arabic Dialect Identification Shared Task.

6 Limitations

Identifying Arabic dialects is a hard task as dialects follow no standards in their structure. They also share MSA phrases due to the fact that MSA is the formal written language in the Arabic world. Our approach of extending tweets improves dialect detection, however, long tweets on a large dataset requires large memory and computing power. For example, when changing the setting of the maximum sequence length to 512 and using the combination of all datasets provided by the organizers for training, our models crashed due to memory shortage. This was overcome by limiting the tweets length to 256 to allow the model to run without crashing.

References

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2021a. 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.

Muhammad Abdul-Mageed, AbdelRahim Elmadany, Chiyu Zhang, ElMoatez Billah Nagoudi, Houda Bouamor, and Nizar Habash. 2023. NADI 2023: The Fourth Nuanced Arabic Dialect Identification Shared Task. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*.

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, Barcelona, Spain (Online). Association for Computational Linguistics.

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, Houda Bouamor, and Nizar Habash. 2021b. NADI 2021: The second nuanced Arabic dialect identification shared task. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 244–259, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, Houda Bouamor, and Nizar Habash. 2022. NADI 2022: The third nuanced Arabic dialect identification shared task. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 85–97, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2021. Arabert: Transformer-based model for arabic language understanding.

Ekaba Bisong. 2019. *Google Colaboratory*, pages 59–64. Apress, Berkeley, CA.

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).

Wei-Cheng Chang, Hsiang-Fu Yu, Kai Zhong, Yiming Yang, and Inderjit Dhillon. 2020. Taming pretrained transformers for extreme multi-label text classification.

Kareem Darwish, Nizar Habash, Mourad Abbas, Hend Al-Khalifa, Huseein T. Al-Natsheh, Samhaa R. El-Beltagy, Houda Bouamor, Karim Bouzoubaa, Violetta Cavalli-Sforza, Wassim El-Hajj, Mustafa Jarrar, and Hamdy Mubarak. 2021. A panoramic survey of natural language processing in the arab world.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Hani Elgabou and Dimitar Kazakov. 2017. Building dialectal Arabic corpora. In *Proceedings of the Workshop Human-Informed Translation and Interpreting Technology*, pages 52–57, Varna, Bulgaria. Association for Computational Linguistics, Shoumen, Bulgaria.

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).

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

SANA at NADI 2023 shared task: Ensemble of Layer-Wise BERT-based models for Dialectal Arabic Identification

Nada Almarwani, and Samah Aloufi
Dep. of Computer Science, College of Computer Science and Engineering, Taibah University
nmarwani@taibahu.edu.sa, slhebi@taibahu.edu.sa

Abstract

Our system, submitted to the Nuanced Arabic Dialect Identification (NADI-23), tackles the first sub-task: Closed Country-level dialect identification. In this work, we propose a model that is based on an ensemble of layer-wise fine-tuned BERT-based models. The proposed model ranked fourth out of sixteen submissions, with an F1-macro score of 85.43.

1 Introduction

Arabic is the national language of 25 countries spoken by more than 372 million speakers[1]. While Modern Standard Arabic (MSA) is the formal written language and is used in speech in a formal contexts such as in academia, official communications, and education (Althobaiti, 2020), each country has its own culturally-based dialect that is used in daily communication and informal situations (Elnagar et al., 2021). Nowadays, within the evolution of virtual communication technologies and the intense popularity of social media platforms, dialectal Arabic has replaced MSA as the primary written form of Arabic to generate online informal content. For example, users on social media share news, comment on political and social events, and express opinions concerning various aspects of life using their own dialect. Accordingly, social media is an invaluable resource for harvesting huge amounts of dialectal Arabic data which can be utilized in numerous computational linguistics and Natural Language Processing (NLP) applications. Due to variations between dialects in term of vocabulary usage, meaning, and sense of given words or phrase, automatic identification between unique dialects is a crucial component for improving several downstream applications such as sentiment analysis, speech recognition, and machine translation.

In order to increase the efficiency of Arabic NLP, the Nuanced Arabic Dialect Identification (NADI) shared task series are dedicated to developing solutions for Arabic dialects identification and other related dialectal processes (Abdul-Mageed et al., 2020, 2021b, 2022, 2023). The majority of the works submitted to the NADI-22 employed pre-trained BERT-based models that are specifically trained on Arabic corpus, such as MARBERT (Abdul-Mageed et al., 2021a), ArabBERT (Antoun et al., 2020), and AraGPT2 (Antoun et al., 2021) using various tuning and data augmentation techniques (Abdel-Salam, 2022; Shammary et al., 2022). Other researchers, such as (AlShenaifi and Azmi, 2022) and (Sobhy et al., 2022), used classical machine learning algorithms with TF-IDF and word embeddings. In this paper, following the first line of work, we present our system submitted to the NADI-2023 shared task (Abdul-Mageed et al., 2023). Specifically, to address the first shared sub-task, our approach is based on an ensemble of layer-wise BERT-based models. Each model is trained independently by accessing hidden states from a designated BERT layer and averaging them to generate the final text embeddings.

This paper is organized as follows: Section 2 presents the dataset utilized in our work, Section 3 introduces the proposed system for Arabic dialect identification, Section 4 provides details experimental results and evaluation, Section 5 discusses the model's results and analyze its errors, and finally, Section 6 summarizes findings and possible future work.

[1] https://lingua.edu/the-most-spoken-languages-in-the-world/

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 625–630
December 7, 2023 ©2023 Association for Computational Linguistics

Model	Freeze Embeddings	Fine Tuned layers
layer 1	0.760	0.736
layer 2	0.799	0.778
layer 3	0.807	0.795
layer 4	0.819	0.799
layer 5	0.824	0.799
layer 6	0.824	0.803
layer 7	0.835	0.827
layer 8	0.841	0.826
layer 9	0.844	0.830
layer 10	**0.855**	**0.840**
layer 11	0.844	0.839

Table 1: The F1-score macro metrics that were computed independently for each layer-wise model on the development set.

Ensemble Model	Freeze Embeddings	Fine Tuned layers
layers(1-11)	0.865	0.851
layers(2-11)	0.865	0.853
layers(3-11)	0.867	0.856
layers(4-11)	0.866	0.856
layers(5-11)	0.870	0.854
layers(6-11)	**0.874**	0.857
layers(7-11)	0.870	0.857
layers(8-11)	0.872	0.850
layers(9-11)	0.870	0.850
layers(10-11)	0.865	0.853
layer(11)	0.844	0.839

Table 2: The results of F1-score macro metrics on the development set for our ablation study, which is based on an ensemble of the layer-based models.

Rank	Team	F1-Score	Accuracy
1	NLPeople	87.27	87.22
2	rematchka	86.18	86.17
3	Arabitools	85.86	85.81
4	Our team	85.43	85.39

Table 3: Performance of the submitted systems on the leaderboard of sub-task1

2 Dataset

The NADI-2023 Shared Task provided the TWT-23 dataset for the Arabic dialects identification task. The dataset contained a total of 23,400 tweets that included 18 Arabic dialects. The dataset was categorized into 18K tweets for training, 1800 tweets for development, and 3600 samples for testing. The training set contained 1000 samples for each dialect class, and the development set included 100 samples for each target class.

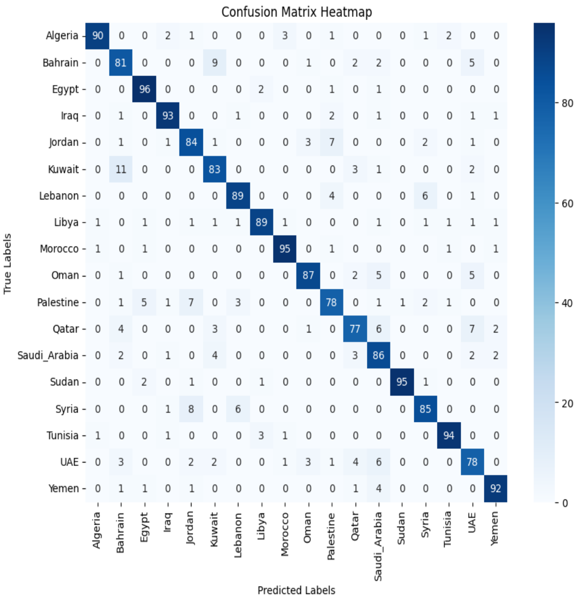
3 System Description

Interpretability of pre-trained language models is an outstanding and active research area in NLP. Various studies have been proposed including studies that investigate and analyze the model's implicit representations across intermediate layers (Kakouros and O'Mahony, 2023; Song et al., 2022). Motivated by this line of work, in this paper, we explore the potential of the MARBERTv2 model (Abdul-Mageed et al., 2021a)[2], on Country-level dialects identification task. It should be noted that we also tested other Arabic pre-trained models, such as AraBERT; however, we achieved the best results using MARBERTv2.

Specifically, during the training phase, we fine-tuned 12 independent models based on MARBERTv2. For each model, we chose a specific layer and averaged its hidden states to generate the text embeddings, which then fed through task-specific linear classifier to make the final prediction. Furthermore, we experimented with the model parameters to identify which one to freeze during the fine-tuning, in which the optimal results were obtained by freezing the embeddings layer. During the validation phase, we used a soft voting ensemble method and an ablation study, which we will detail in Section 4.1, to determine the best model. Hence, our final submission was an ensemble of models from layers 6 to 11.

Experimental setup We mainly followed the same experimental setups used in (Abdel-Salam, 2022) to fine-tune the model with the exception of the learning rate, weight decay and sentence length, which was set to 2e-5, 1e-2, and 512, respectively. We trained the model with a batch size of 8, for 10 epochs. After each epoch, the model was evaluated on the development set, and the best performant parameters were saved.

[2] Arabic-based pre-trained BERT model that is publicly available in the HuggingFace library (Wolf et al., 2020)



	Algeria	Bahrain	Egypt	Iraq	Jordan	Kuwait	Lebanon	Libya	Morocco	Oman	Palestine	Qatar	Saudi_Arabia	Sudan	Syria	Tunisia	UAE	Yemen
Algeria	90	0	0	2	1	0	0	0	3	0	1	0	0	0	1	2	0	0
Bahrain	0	81	0	0	0	9	0	0	0	1	0	2	2	0	0	0	5	0
Egypt	0	0	96	0	0	0	0	2	0	0	1	0	1	0	0	0	0	0
Iraq	0	1	0	93	0	0	1	0	0	0	2	0	1	0	0	0	1	1
Jordan	0	1	0	1	84	1	0	0	0	3	7	0	0	0	2	0	1	0
Kuwait	0	11	0	0	0	83	0	0	0	0	0	3	1	0	0	0	2	0
Lebanon	0	0	0	0	0	0	89	0	0	0	4	0	0	0	6	0	1	0
Libya	1	0	1	0	1	1	1	89	1	0	0	0	1	0	1	1	1	1
Morocco	1	0	1	0	0	0	0	0	95	0	1	0	0	0	0	1	0	1
Oman	0	1	0	0	0	0	0	0	0	87	0	2	5	0	0	0	5	0
Palestine	0	1	5	1	7	0	3	0	0	0	78	0	1	1	2	1	0	0
Qatar	0	4	0	0	0	3	0	0	0	1	0	77	6	0	0	0	7	2
Saudi_Arabia	0	2	0	1	0	4	0	0	0	0	0	3	86	0	0	0	2	2
Sudan	0	0	2	0	1	0	0	1	0	0	0	0	0	95	1	0	0	0
Syria	0	0	0	1	8	0	6	0	0	0	0	0	0	0	85	0	0	0
Tunisia	1	0	0	1	0	0	0	3	1	0	0	0	0	0	0	94	0	0
UAE	0	3	0	0	2	2	0	0	1	3	1	4	6	0	0	0	78	0
Yemen	0	1	1	0	1	0	0	0	0	0	0	1	4	0	0	0	0	92

Figure 1: Confusion Matrix Heat-map for development set classification.

4 Results

We evaluated the performance of our proposed method on the dialects identification task through a set of experiments to investigate the impact of each layer on classifying the 18 dialects. Table 1 demonstrates the performance of the layer-wise-based models on the development set. The performance improved with the higher layers. Notably, freezing the embeddings during the fine-tuning yielded a better overall performance. Also, averaging the corresponding hidden states of layer 10 while freezing the embeddings achieved the best result with an F1-score of 85.5%. In addition to exploring the layer-wise models' performance independently, we used a soft voting ensemble technique along with an ablation study to select the combination of independent models that yield the best performance on the development set.

4.1 Ablation Study Result

The goal of an ablation study is to examine the impact of removing components of an Artificial intelligence-based system on the system's performance (Zschech, 2022). We examine the impact of different layer-based models on the final model's performance using a soft voting ensemble, as shown in Table 2. Combining models trained on layers 1-11 results in the worst performance; however, removing lower-layer-trained models improved the results. Also, the performance slightly decreased when using models trained only on higher layers (8, 9, 10, and 11). The best results were obtained with an ensemble of models that trained on layers 6 through 11, with an F1-score of 0.874 when embeddings were frozen and 0.857 when embeddings were included in the fine-tuning.

Class	Precision	Recall	F1
Algeria	0.97	0.90	0.93
Bahrain	0.76	0.81	0.79
Egypt	0.91	0.96	0.93
Iraq	0.93	0.93	0.93
Jordan	0.80	0.84	0.82
Kuwait	0.81	0.83	0.82
Lebanon	0.89	0.89	0.89
Libya	0.94	0.89	0.91
Morocco	0.94	0.95	0.95
Oman	0.92	0.87	0.89
Palestine	0.82	0.78	0.80
Qatar	0.84	0.77	0.80
KSA	0.75	0.86	0.80
Sudan	0.99	0.95	0.97
Syria	0.87	0.85	0.86
Tunisia	0.95	0.94	0.94
UAE	0.76	0.78	0.77
Yemen	0.93	0.92	0.92

Table 4: F1-score, recall, and precision breakdown of how well the model performs for each individual class.

Testing Phase: Table 3 shows the performance of our system submitted to the NADI-2023 shared task: closed country-level dialect identification compared to the top 3 systems.

5 Error Analysis and Discussion

Table 4 shows a detailed evaluation of the model's performance across the 18 distinct classes. Precision values are relatively high at 0.80 for most of the classes. This indicates a strong overall performance, except for the KSA and UAE dialects, where the precision falls under 0.80. Conversely, recall values have less variation. The Algeria, Egypt, Iraq, Morocco, Sudan, and Tunisia classes have high recall rate, which reflects the models' abilities to capture instances from these classes. The F1-score results show the model's strong per-

True Label	Predicted Label	Text	English
Egypt	KSA	لا ياصاحبي مع نفسك هه	No, my friend, with yourself
Egypt	Libya	فيه راجل ينور وشك وفيه راجل يطفيه	There is a man who lights up your face, and there is a man who extinguishes it
Oman	KSA	عطوها قهوه مره	Give her bitter coffee
Yemen	KSA	يمكن النت عندك بطي والا قد نشرتها	Maybe your internet is slow or else you have already posted it
Palestine	KSA	حطي بودره ومي وكم نقطه من ماء الزهر وحطيه ع وجهك لينشف وبعد مابنشف كتيه وغسليه بمي فاتره وشكرا باي عفوا	Apply some powder and water, and a few drops of blossom water on your face. Let it dry, and once it dries, peel it off and wash with water. Thank you, bye, you're welcome
Qatar	Kuwait	يارب بكرا انصدم من سهوله الامتحان واطلع مستانسه وحاله عدل	Dear God, I hope that tomorrow I'll be surprised by how easy the exam is, and I'll come out happy and in a good mood.
Oman	UAE	شو دخل هذا في هذاا	What does this have to do with that
Libya	Morocco	افهمها وهي طايره تويتر راه مش فيس باش نكتب ع راحتي	I understand that this is Twitter not Facebook to write my mind.
Syria	Lebanon	الصغير بدو و الكبير بدو وما حدا عاجبو حالو	The young one wants, and the old one wants, and no one is pleased.
Jordan	Palestine	طيب دعمل حالي مكيفه ع الدوام	Alright, I'll pretend to be cool at work.

Table 5: Examples of Incorrect Predictions from the Development Set.

formance, with most of the classes achieving score of 0.80 or higher.

Figure 1 shows a heat-map of confusion matrix for the development set to further analyze the margin of error in the model's predictions. In general, with minor exceptions, the model seems to perform well for most of the classes. For example, the model preforms well at predicting instances for Egypt, Morocco, and Sudan classes, with true positive exceeding 95 instances. Conversely, the number of true positives are as low as 79 instances or less when predicting instances for the Palestine, UAE, and Qatar classes.

To further analyse, Table 5 shows examples from the development set that our model failed to predict correctly. We observed that the errors of the models of False Positive (FP) and False Negative (FN) fall in one of the following categories:

Missing of diacritics: In Arabic, while different Arabic dialects share common linguistic features, differences remain in the usage of the vocabulary and its meaning. Diacritics plays a crucial role in disambiguate the senses, meanings, and semantics of Arabic language (Matrane et al., 2023; Almuqren and Cristea, 2016; Azmi and Almajed, 2015). We hypothesize that adding diacritics may improve the model's performance in predicting the dialect of a given text. To illustrate more, the first two examples in Table 5 presents this case of ambiguity which might be resolved by diacritics. As can be seen from the confusion matrix, the Egypt class has the least number of FN. We noted that correctly classifying these examples is challenging, even for humans, using the written text only without any context. However, for example, adding diacritics to the word ”صاحبي SAHby”, which translate in English to ”My friend”, might help the model to identify the correct class. In particular, in the Egyptian dialect this word would be pronounced with the following diacritics ”صاحَبَيِ SAHabayi”, where in the KSA dialect it would be pronounced with the following diacritics ”صاحِبَيِ SAHibayi”. Including diacritics may also resolve the ambiguity in the second example, where the words ”راجل rAjl and ينور ynwr” in the example, which translate respectively to ”man” and ”lights up”, pronounced differently in both Egyptian and Libyan dialects.

Regional Varieties: Among the 18 dialects classes, the KSA class has the largest variety of dialects due to the geographical diversity and historical migration of people from different linguistic backgrounds. Thus, the East region of KSA tends to share a lot of linguistic similarities with Egypt, while the Southern region share similarities with Yemen, the Northern region is similar to the Levantine dialect (this includes: Syria, Jordan, Palestine, and Lebanon), and the Middle and Western regions congruent with rest of Gulf countries (Bayazed et al., 2020). Also, according to (Alruily, 2020), the majority of most active twitter users are from KSA. Hence, we believe that these factors affected the performance of our model, as the majority of the FP predictions were a result of flawed prediction where other classes were categorized as KSA, examples $3-5$ in Table 5.

Dialects Family: We noted that most of the FP and FN between classes occur among dialects that belong to the same family, or regional varieties of a given dialect. For example, many of the FP and FN occurred in the Gulf dialects family, which includes UAE, Qatar, Bahrain, Kuwait, Oman, Iraq, and certain parts of KSA. This also evident in examples from the Levantine and North African dialects family, example $6-10$ in Table 5.

6 Conclusion

This work describes our proposed system to automatically identifying dialectal Arabic, which has been submitted to the NADI-2023 shared task. The proposed system leveraged the intermediate layers of the pre-trained MARBERTv2 in identifying the Arabic dialects instead of relying on the final layer for text representation. The proposed layer-wise BERT-based models demonstrate a strong overall performance in distinguishing 18 Arabic dialects, achieving an F1 score of 87% on the development set and 85% on the test set. Furthermore, we analyzed the performance of our model and discuss the factors that caused FP and FN predictions. Hence, further elaboration could be followed to study the impact of using diacritics on model performance.

References

Reem Abdel-Salam. 2022. Dialect & sentiment identification in nuanced Arabic tweets using an ensemble of prompt-based, fine-tuned, and multitask BERT-based models. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 452–457, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2021a. 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.

Muhammad Abdul-Mageed, AbdelRahim Elmadany, Chiyu Zhang, ElMoatez Billah Nagoudi, Houda Bouamor, and Nizar Habash. 2023. NADI 2023: The Fourth Nuanced Arabic Dialect Identification Shared Task. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*.

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, Barcelona, Spain (Online). Association for Computational Linguistics.

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, Houda Bouamor, and Nizar Habash. 2021b. NADI 2021: The second nuanced Arabic dialect identification shared task. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 244–259, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, Houda Bouamor, and Nizar Habash. 2022. NADI 2022: The third nuanced Arabic dialect identification shared task. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 85–97, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Latifah Almuqren and Alexandra I Cristea. 2016. Framework for sentiment analysis of arabic text. In *Proceedings of the 27th ACM conference on hypertext and social media*, pages 315–317.

Meshrif Alruily. 2020. Issues of dialectal saudi twitter corpus. *Int. Arab J. Inf. Technol.*, 17(3):367–374.

Nouf AlShenaifi and Aqil Azmi. 2022. Arabic dialect identification using machine learning and transformer-based models: Submission to the NADI 2022 shared task. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 464–467, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Maha J Althobaiti. 2020. Automatic arabic dialect identification systems for written texts: A survey. *arXiv preprint arXiv:2009.12622*.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. AraBERT: Transformer-based model for Arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France. European Language Resource Association.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2021. AraGPT2: Pre-trained transformer for Arabic language generation. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 196–207, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Aqil M Azmi and Reham S Almajed. 2015. A survey of automatic arabic diacritization techniques. *Natural Language Engineering*, 21(3):477–495.

Afnan Bayazed, Ola Torabah, Redha AlSulami, Dimah Alahmadi, Amal Babour, and Kawther Saeedi. 2020. Sdct: Multi-dialects corpus classification for saudi tweets. *International Journal of Advanced Computer Science and Applications*, 11(11).

Ashraf Elnagar, Sane M. Yagi, Ali Bou Nassif, Ismail Shahin, and Said A. Salloum. 2021. Systematic literature review of dialectal arabic: Identification and detection. *IEEE Access*, 9:31010–31042.

Sofoklis Kakouros and Johannah O'Mahony. 2023. What does bert learn about prosody? In *20th International Congress of Phonetic Sciences ICPhS*. International Phonetics Association.

Yassir Matrane, Faouzia Benabbou, and Nawal Sael. 2023. A systematic literature review of arabic dialect sentiment analysis. *Journal of King Saud University-Computer and Information Sciences*, page 101570.

Fouad Shammary, Yiyi Chen, Zsolt T Kardkovacs, Mehwish Alam, and Haithem Afli. 2022. TF-IDF or transformers for Arabic dialect identification? ITFLOWS participation in the NADI 2022 shared task. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 420–424, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Mahmoud Sobhy, Ahmed H. Abu El-Atta, Ahmed A. El-Sawy, and Hamada Nayel. 2022. Word representation models for Arabic dialect identification. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 474–478, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Mingyang Song, Yi Feng, and Liping Jing. 2022. Utilizing bert intermediate layers for unsupervised keyphrase extraction. In *Proceedings of the 5th International Conference on Natural Language and Speech Processing (ICNLSP 2022)*, pages 277–281.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Patrick Zschech. 2022. Beyond descriptive taxonomies in data analytics: A systematic evaluation approach for data-driven method pipelines. *Inf. Syst. E-Bus. Manag.*, 21(1):193–227.

ISL-AAST at NADI 2023 shared task: Enhancing Arabic Dialect Identification in the Era of Globalization and Technological Progress

Shorouk Adel
Arab Academy for Science,
Technology, and Maritime Transport,
Alexandria, Egypt
shoroukadel@student.aast.edu

Noureldin Elmadany
Arab Academy for Science,
Technology, and Maritime Transport,
Alexandria, Egypt
nourelmadany@aast.edu

Abstract

Arabic dialects have extensive global usage owing to their significance and the vast number of Arabic speakers. However, technological progress and globalization are leading to significant transformations within Arabic dialects. They are acquiring new characteristics involving novel vocabulary and integrating linguistic elements from diverse dialects. Consequently, sentiment analysis of these dialects is becoming more challenging. This study categorizes dialects among 18 countries, as introduced by the Nuanced Arabic Dialect Identification (NADI) shared task competition. The study approach incorporates the utilization of the MARBERT and MARBERT v2 models with a fine tunning processes. The findings reveal that the most effective model is achieved by applying averaging and concatenation to the hidden layers of MARBERT v2, followed by feeding the resulting output into convolutional layers.Furthermore, employing the ensemble method on various methods enhances the model's performance. Our system secures the 6th position among the top performers in the First subtask, achieving an F1 score of 83.73%.

1 Introduction

The Arabic region encompasses numerous cultures, each characterized by dialectal variations influenced by historical, geographical, and sociopolitical factors (Bouamor et al., 2014). While this variety showcases the region's cultural wealth, it creates difficulties when analyzing Arabic information, especially on social media networks. Moreover, the rapid evolution of the language in the digital age and the widespread use of social media are presenting a new era for the Arabic language. Modern communication tools are enabling speakers of various Arabic dialects to interact globally. This interaction is leading to a dynamic evolution of the language, characterized by the emergence of new vocabulary, slang, and expressions (Darvin, 2016). The continuous generation of new words and language adaptations is presenting a unique challenge for linguistic analysis. Therefore, Modern Standard Arabic has a disequilibrium between preserving tradition and adjusting to the demands of modern communication. Moreover, Arabic dialect identification plays a pivotal role in understanding regional language variations on social media. Improving this task has implications for cultural preservation, social analysis, and natural language processing technology. However, the presence of diverse Arabic dialects with distinct linguistic traits can pose challenges in analyzing and interpreting social media content (Salameh et al., 2018). People from different regions might use completely different words to express the same concepts.

Recent advancements in Arabic Dialect Identification research have been notable, with various studies addressing the intricate nuances of Arabic dialects. The MADAR shared task on fine-grained dialect identification (Bouamor et al., 2019) delved into sub-dialect distinctions, highlighting the complexity of Arabic language variations. Machine Translation of Arabic Dialects (Salloum, 2018) focused on adapting translation models to handle dialect-specific expressions, facilitating communication across dialect differences. Moreover, efforts in the Automatic Identification of Arabic Dialects in Social Media (Sadat et al., 2014) utilized natural language processing and machine learning to automate dialect recognition, revealing regional language trends online. Various methods, including feature extraction and machine learning algorithms (Zaidan and Callison-Burch, 2014), have contributed to improving automated dialect identification accuracy and uncovering the rich diversity of Arabic dialects. In the recent NADI shared task series (Abdul-Mageed et al.,

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 631–636
December 7, 2023 ©2023 Association for Computational Linguistics

2020b, 2021, 2022), teams employed a range of approaches, including traditional methods like SVM with TF-IDF (Nayel et al., 2021), customized Bert-based models (AlKhamissi et al., 2021), and deep learning techniques with models like MARBERT and AraBERT (Messaoudi et al., 2022; Abdel-Salam, 2022; Attieh and Hassan, 2022). These efforts collectively contribute to the advancement of Arabic dialect identification, showcasing diverse methodologies and approaches in the field.

In this research, we aim to enhance the F1 score of Arabic dialect identification, provided by NADI shared task 2023 (Abdul-Mageed et al., 2023), by investigating the impact of various model enhancements. our study conducts a series of experiments using MARBERT and MARBERT v2 models (Abdul-Mageed et al., 2020a), involving various techniques. This approach includes concatenating hidden layers (Devlin et al., 2018) and processing the resulting outputs using CNN layers (Jacovi et al., 2018), BILSTM models (Graves et al., 2005), or a combination of BILSTM and CNN. Additionally, Experiments involve adapters with the MARBERT model (Pfeiffer et al., 2020). Finally, to maximize our results, our work utilizes ensemble methods that combine the outcomes of the majority of these experiments (Re and Valentini, 2012).

The rest of the paper is organized as follows: providing the dataset and its preparation are presented in Section 2. In Section 3, we explain the methodologies employed for Arabic Dialect Identification. Subsequently, Section 4 presents the results of our model's performance, including an analysis of our findings. In Section 6, we summarize and conclude our findings.

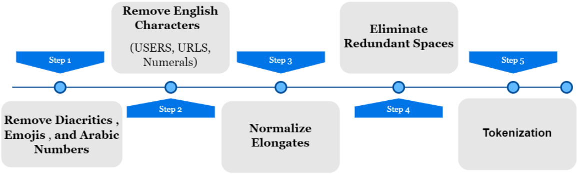

Step 1
Remove Diacritics , Emojis , and Arabic Numbers
Remove English Characters
(USERS, URLS, Numerals)
Step 2
Step 3
Normalize Elongates
Eliminate Redundant Spaces
Step 4
Step 5
Tokenization

Figure 1: Pre-processing steps on the dataset

2 Data

2.1 Dataset Description

The presented approach utilized the training and validation data provided by the task organizers (Abdul-Mageed et al., 2023). The training set for Subtask 1 consists of around 18,000 tweets with 18 different labels representing 18 country dialects. While the development set consists of 1,800 labeled tweets. The submitted results were evaluated on a test set consisting of 3,600 tweets covering 18 country-level dialects.

2.2 Dataset Pre-processing

The dataset pre-processing is shown in Figure 1. The initial steps involved removing diacritics, which are modifications to Arabic characters. Subsequently, specific words were removed, such as mentions of users, URLs, and numerical values. Additionally, elongated characters were normalized to a single occurrence of the represented character. Emojis were also removed from the text. To further enhance the text, a series of processes were applied. Non-Arabic characters and redundant spaces were eliminated. Stemming or Lemmatization was not performed due to the intricacies of the Arabic language. These linguistic intricacies include the rich morphology and variability in Arabic dialects, where words may undergo significant changes in form and structure. Applying stemming or lemmatization involves reducing words to their root or base form. By observation, it could potentially result in the loss of valuable dialect-specific information and hinder the accuracy of the classification process. Finally, the text was tokenized by MARBERT and MARBERTv2 tokenizer utilizing the Transformers library.

3 System Description

This study conducted comprehensive experiments to explore various modifications to our baseline models, MARBERT and MARBERTv2 (Abdul-Mageed et al., 2020a), as detailed in Table 1. We maintained a constant batch size of 64 throughout our experiments and conducted 15 epochs, saving the epoch with the best F1 score by using early stopping. The Adam optimizer (Jais et al., 2019) and Cross Entropy Loss (Smith and Johnson, 2022)

Experiment	Description	Test		Dev	
		Accuracy(%)	F1(%)	Accuracy(%)	F1(%)
Exp.1	MARBERT+ adapter (LR=2e-5)	74.78	74.63	76.06	75.76
Exp.2	MARBERTv2+ adapter (LR=2e-5)	73.58	73.35	75.20	75.01
Exp.3	MARBERT(LR=2e-5)	79.86	80.03	81.61	81.86
Exp.4	MARBERTV2(LR=2e-5)	79.06	79.14	81.11	81.18
Exp.5	MARBERT (last 4 Layers Conc.)(LR=2e-5)	78.39	78.36	79.94	79.87
Exp.6	MARBERTv2(last 4 Layers Conc.) (LR=2e-5)	80.28	80.33	82.44	82.56
Exp.7	MARBERT (average layers 4-7 and conc. output with last 4 layers)(LR=2e-5)	79.86	80.03	80.61	80.72
Exp.8	MARBERTv2 ((average layers 4-7 and conc. output with last 4 layers) (LR=2e-5)	80.83	80.94	81.61	81.86
Exp.9	Repeat Exp.7 + utilizing 1 Conv. Filter(kernel size=5) + MP (LR=2e-5)	81.50	80.83	81.83	81.91
Exp.10	Repeat Exp.8+ utilizing 1 Conv. Filter(kernel size=5)+ MP (LR=2e-5)	81.47	81.43	82.56	82.51
Exp.11	Repeat Exp.7 + BILSTM as classifier (LR=2e-5)	77.72	77.84	78.44	78.33
Exp.12	Repeat Exp.7 +BILSTM + 1 Conv. Filter(kernel size=5) + MP (LR=2e-5)	78.36	78.49	79.11	79.30
Exp.13	Repeat Exp.7 +3 Conv. Filters: kernel sizes(5,4,3) + MP (LR=1e-5)	79.86	80.00	81.83	81.91
Exp.14	Repeat Exp.7 +3 Conv. Filters:kernel sizes(10,8,6) + MP (LR=1e-5)	81.56	81.67	83.06	83.14
Exp.15	Repeat Exp.7 +3 Conv. Filters: kernel sizes(7,7,7) + MP (LR=1e-5)	81.64	81.64	82.72	82.84
Exp.16	Repeat Exp.7 +3 Conv. Filters:kernel sizes(12,10,8) + MP (LR=1e-5)	80.72	80.83	81.61	81.86
Exp.17	Voting Ensemble(Exp 3-16)	83.67	83.73	85.20	85.27
Exp.18	Average Ensemble(Exp 3-16)	83.31	83.36	84.11	84.16

Table 1: Experimental Results for Different Models on Test and Dev Datasets(Abbreviations: F1 - F1-score, MP - Maxpooling, Conc. - Concatenate, Conv. - Convolution)

were utilized in all cases. Learning rates (LR) varied by experimental setup between 1e-5 and 2e-5. Let us delve into more details about each of the 18 different experiments (Exp.) and their significance within this study:

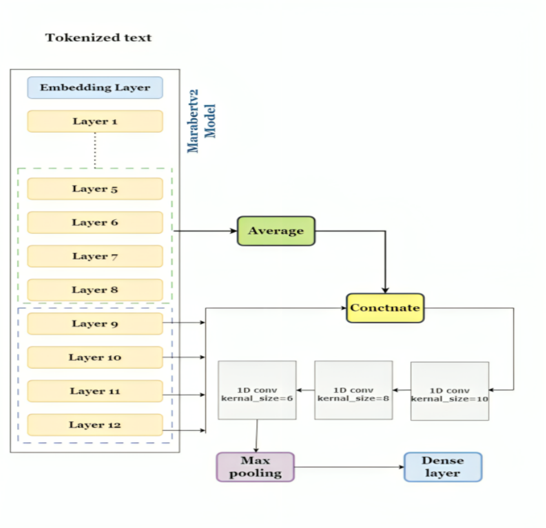


Figure 2: Best Model architecture

3.1 Adapters for MARBERT and MARBERTv2 (Exp.1 and Exp.2)

These experiments explored the impact of adding adapters to the baseline models, MARBERT and MARBERTv2. Adapters are specialized neural network modules added to the models to fine-tune their performance for specific tasks (Pfeiffer et al., 2020) .

3.2 Layer Concatenation (Exp.5 and Exp.6)

In these experiments, the study investigated the concatenation of the last four layers of BERT-Base (Devlin et al., 2018), MARBERT and MARBERT v2 (Abdul-Mageed et al., 2020a). This approach aimed to capture and combine features from different model layers, potentially improving its representation learning capabilities.

3.3 Average Layer 4-7 and Concatenation (Exp.7 and Exp.8)

Experiments 7 and 8 focused on taking an average of layers 4-7 and concatenating it with the last four layers of the models, MARBERT and MARBERTv2, respectively. This approach aimed to leverage layer stacking for enhanced model performance. The results provide insights into the combined impact of these modifications for each model.

3.4 Convolutional Layers with Varying Kernel Sizes (Exp.9 to Exp.16)

These experiments introduced leveraging a series of convolutional layers with varying filter sizes. The ReLU activation function was used within these convolutional layers to introduce non-linearity and enhance the model's capacity to learn complex representations. Following the convolutional layers, max-pooling layers (MaxPool1D) were utilized to reduce the spatial dimensions of the feature maps. The size of the pooling window was determined dynamically based on the length of the convolutional filter. Specifically, the filter size of the last convolutional layer was subtracted from the sequence length, and the result was then added to the stride value. The outputs of these convolutional and max-pooling layers were then flattened. Subsequently, a fully connected dense layer was employed to process the sentence embedding further. (Jacovi et al., 2018).

3.5 Bidirectional LSTM (Exp.11)

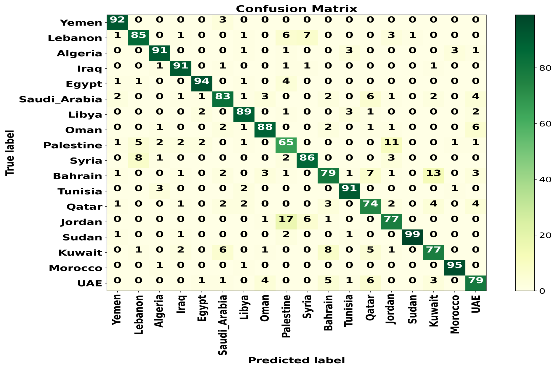
Experiment 11 involved adding Bidirectional Long Short-Term Memory (BILSTM) layers as a classifier layer for the MARBERTv2 model. BILSTM layers process input sequences in both forward and backward directions, potentially capturing dependencies in the data more effectively (Graves et al., 2005).

3.6 Ensemble Methods (Exp.17 and Exp.18)

These experiments leveraged ensemble methods to enhance model performance further (Re and Valentini, 2012). The Voting Ensemble (Exp.17) and Average Ensemble (Exp.18) combine the outputs of multiple experiments (Exp.3 to Exp.16) to make predictions. The Voting Ensemble considers the majority or weighted votes, while the Average Ensemble computes the mean of probabilities for predictions.

4 Results and discussion

We present a summary of our experimentation and evaluation of various model enhancements, as



	Yemen	Lebanon	Algeria	Iraq	Egypt	Saudi_Arabia	Libya	Oman	Palestine	Syria	Bahrain	Tunisia	Qatar	Jordan	Sudan	Kuwait	Morocco	UAE
Yemen	92	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0
Lebanon	1	85	0	1	0	0	1	0	6	7	0	0	0	3	1	0	0	0
Algeria	0	0	91	0	0	0	1	0	1	0	0	3	0	0	0	0	3	1
Iraq	0	0	1	91	0	1	0	0	1	1	0	0	0	0	0	1	0	0
Egypt	1	1	0	0	94	0	1	0	4	0	0	0	0	0	0	0	0	0
Saudi_Arabia	2	0	0	1	1	83	1	3	0	0	2	0	6	1	0	2	0	4
Libya	0	0	0	0	2	0	89	0	1	0	0	3	1	0	0	0	0	2
Oman	0	0	1	0	0	2	1	88	0	0	2	0	1	1	0	0	0	6
Palestine	1	5	2	2	2	0	1	0	65	0	0	0	0	11	0	0	1	1
Syria	0	8	1	0	0	0	0	0	2	86	0	0	0	3	0	0	0	0
Bahrain	1	0	0	1	0	2	0	3	1	0	79	1	7	1	0	13	0	3
Tunisia	0	0	3	0	0	0	2	0	0	0	0	91	0	0	0	0	1	0
Qatar	1	0	0	1	0	2	2	0	0	0	3	0	74	2	0	4	0	4
Jordan	0	0	0	0	0	0	0	1	17	6	1	0	0	77	0	0	0	0
Sudan	1	0	0	1	0	0	0	0	2	0	0	1	0	0	99	0	0	0
Kuwait	0	1	0	2	0	6	0	1	0	0	8	0	5	1	0	77	0	0
Morocco	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	95	0
UAE	0	0	0	0	1	1	0	4	0	0	5	1	6	0	0	3	0	79

Figure 3: Confusion Matrix for DEV set: voting Ensemble method

reported in Table 1. With a particular emphasis on the F1-score, each experiment assesses the performance achieved by adapting and modifying the baseline models, MARBERT and MARBERTv2.

In our extensive series of experiments, we conduct an exploration of various model enhancements with a primary focus on optimizing F1 scores. Among these experiments, Exp.14 showcases the best results as a standalone model. As shown in Figure 2, this model is built upon the foundation of Exp.7 with the addition of three convolution filters (kernel sizes: 10, 8, 6), followed by max-pooling, and also demonstrates robustness with an impressive F1-score of 81.67% on the test dataset. These results emphasize the significance of spatial feature extraction in text classification tasks. Regarding our methodological approach, Exp.17 represents the most effective method. It serves as our submission and leverages ensemble techniques to combine the predictions of multiple models. This ensemble method significantly outperforms individual models, achieving outstanding F1-scores of 85.27% for the DEV dataset and 83.73% for the test dataset.

Notably, we observe instances of misclassification between the two classes, notably between Jordan and Palestine, as well as between Kuwait and Bahrain, as illustrated in Figure 3. These misclassifications can be attributed to several factors, including historical, cultural, and linguistic nuances that may pose challenges for natural language processing models. The misclassification of content related to Kuwait and Bahrain is a result of shared geographical proximity and cultural ties, leading to overlapping themes and terminology in text data. These overlapping characteristics can cause our models to occasionally struggle in correctly differentiating between the two, resulting in fluctuations in classification performance. These observed misclassifications underscore the need for continued research and model refinement, especially when dealing with regions or topics characterized by subtle distinctions. Addressing such complexities will contribute to improving the accuracy and robustness of models in handling cases with inherent challenges like those presented by Jordan vs. Palestine and Kuwait vs. Bahrain.

With more time available, we will delve into training on larger datasets. Additionally, our study will explore the use of different loss functions for various hyperparameters and incorporate additional ensemble methods such as stacking, bagging, boosting, random forests, AdaBoost, and gradient boosting.

5 Conclusion

Overall, This paper outlines our methods for solving Nuanced Arabic Dialect Identification (NADI) shared task 2023 subtask-1. The extensive experimentation and analysis highlighted the nuanced nature of model enhancements and adaptations. Some modifications, like layer concatenation and the addition of convolution layers, exhibited clear benefits. On the other hand, adapters had more limited impacts. Additionally, ensemble methods emerged as a powerful tool for boosting the score. These findings emphasize the need for a thoughtful and data-driven approach when fine-tuning models for specific tasks and domains in natural language processing. Our system ranks in the 6th best spots of the leaderboards of the first subtask with an F1-score of 83.73%. Future research directions include investigating the impact of larger training datasets on model performance.

6 Limitations

We focused on MARBERT and MARBERTv2 models without comparing them to alternative models. Furthermore, we should have leveraged the advantages of more extensive datasets and various hyperparameters. However, a significant strength of our study lies in exploring the integration of transformers with deep-learning models and adapters.

References

Reem Abdel-Salam. 2022. Dialect & sentiment identification in nuanced arabic tweets using an ensemble of prompt-based, fine-tuned, and multitask bert-based models. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 452–457.

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

Muhammad Abdul-Mageed, AbdelRahim Elmadany, Chiyu Zhang, ElMoatez Billah Nagoudi, Houda Bouamor, and Nizar Habash. 2023. NADI 2023: The Fourth Nuanced Arabic Dialect Identification Shared Task. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*.

Muhammad Abdul-Mageed, Chiyu Zhang, Houda Bouamor, and Nizar Habash. 2020b. Nadi 2020: The first nuanced arabic dialect identification shared task. *arXiv preprint arXiv:2010.11334*.

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, Houda Bouamor, and Nizar Habash. 2021. Nadi 2021: The second nuanced arabic dialect identification shared task. *arXiv preprint arXiv:2103.08466*.

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, Houda Bouamor, and Nizar Habash. 2022. Nadi 2022: The third nuanced arabic dialect identification shared task. *arXiv preprint arXiv:2210.09582*.

Badr AlKhamissi, Mohamed Gabr, Muhammad El-Nokrashy, and Khaled Essam. 2021. Adapting marbert for improved arabic dialect identification: Submission to the nadi 2021 shared task. *arXiv preprint arXiv:2103.01065*.

Joseph Attieh and Fadi Hassan. 2022. Arabic dialect identification and sentiment classification using transformer-based models. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 485–490.

Houda Bouamor, Nizar Habash, and Kemal Oflazer. 2014. A multidialectal parallel corpus of arabic. In *LREC*, pages 1240–1245.

Houda Bouamor, Sabit Hassan, and Nizar Habash. 2019. The madar shared task on arabic fine-grained dialect identification. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 199–207.

Ron Darvin. 2016. Language and identity in the digital age. *The Routledge handbook of language and identity*, pages 523–540.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. 2005. Bidirectional lstm networks for improved phoneme classification and recognition. In *International conference on artificial neural networks*, pages 799–804. Springer.

Alon Jacovi, Oren Sar Shalom, and Yoav Goldberg. 2018. Understanding convolutional neural networks for text classification. *arXiv preprint arXiv:1809.08037*.

Imran Khan Mohd Jais, Amelia Ritahani Ismail, and Syed Qamrun Nisa. 2019. Adam optimization algorithm for wide and deep neural network. *Knowledge Engineering and Data Science*, 2(1):41–46.

Abir Messaoudi, Chayma Fourati, Hatem Haddad, and Moez BenHajhmida. 2022. icompass working notes for the nuanced arabic dialect identification shared task. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 415–419.

Hamada Nayel, Ahmed Hassan, Mahmoud Sobhi, and Ahmed El-Sawy. 2021. Machine learning-based approach for arabic dialect identification. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 287–290.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020. Adapterhub: A framework for adapting transformers. *arXiv preprint arXiv:2007.07779*.

Matteo Re and Giorgio Valentini. 2012. Ensemble methods. *Advances in machine learning and data mining for astronomy*, pages 563–593.

Fatiha Sadat, Farnazeh Kazemi, and Atefeh Farzindar. 2014. Automatic identification of arabic dialects in social media. In *Proceedings of the first international workshop on Social media retrieval and analysis*, pages 35–40.

Mohammad Salameh, Houda Bouamor, and Nizar Habash. 2018. Fine-grained arabic dialect identification. In *Proceedings of the 27th international conference on computational linguistics*, pages 1332–1344.

Wael Salloum. 2018. *Machine Translation of Arabic Dialects*. Columbia University.

John Smith and Lisa Johnson. 2022. Categorical cross entropy loss for multi-class classification. *Journal of Machine Learning Research*, 30(1):100–120.

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

Frank at NADI 2023 Shared Task: Trio-Based Ensemble Approach for Arabic Dialect Identification

Dilshod Azizov[1], **Jiyong Li**[2], **Shangsong Liang**[1,*]
[1]Mohamed bin Zayed University of Artificial Intelligence
[2]Sun Yat-sen University
dilshod.azizov@mbzuai.ac.ae, lijy373@mail2.sysu.edu.cn
[*]Corresponding author, liangshangsong@gmail.com

Abstract

We present our system designed for Subtask 1 in the shared task NADI on Arabic Dialect Identification, which is part of ARABICNLP 2023. In our approach, we utilized models such as: MARBERT, MARBERTv2 (A) and MARBERTv2 (B). Subsequently, we created a majority-voting ensemble of these models. We used MARBERTv2 with different hyperparameters, which significantly improved the overall performance of the ensemble model. In terms of performance, our system achieved a competitive an F1 score of **84.76**. Overall, our system secured the 5th position out of 16 participating teams.

1 Introduction

The Arabic language, with its vast and varied tapestry of dialects, offers a mesmerizing blend of history, culture and linguistic evolution. Each dialect, from the mellifluous notes of Levantine to the rhythmic cadences of Maghrebi, narrates a unique story of its people, their journeys, and their experiences. However, such linguistic richness often goes unnoticed, overshadowed by mainstream dialects and a lack of comprehensive research tools. The persistent gaps in our understanding, exacerbated by limited resources, such as datasets, have made the exploration of these dialects both a challenge and a treasure hunt for researchers (Althobaiti, 2020).

In response to this, the series of nuanced Arabic dialect identification (NADI) shared tasks, initiated by (Abdul-Mageed et al., 2020b), emerged as a beacon of hope, spotlighting lesser studied dialects. Over the years 2020 (Abdul-Mageed et al., 2020b), 2021 (Abdul-Mageed et al., 2021), and 2022 (Abdul-Mageed et al., 2022), NADI provided invaluable datasets and created a vibrant platform where scholars and enthusiasts could exchange insights, challenge conventional methodologies, and ignite renewed interest in dialect identification. This discipline, which is based on determining the variety of sources of textual or spoken content, has now become central to understanding the rich fabric of the Arabic linguistic diversity.

The subtask can be formulated as follows:

Identify the specific country-level dialect of a given Arabic tweet.

This task is armed with the novel TWT-2023 dataset, which covers 18 mesmerizing dialects, and is supplemented by external datasets such as NADI-2020-TWT, NADI-2021-TWT and MADAR-2018 (Bouamor et al., 2018).

Our contributions are as follows:

- We propose an automated system based on the majority-voting ensemble that uses MARBERT (Abdul-Mageed et al., 2020a), MARBERTv2 (A) and MARBERTv2 (B) for the Dialect Identification.
- We compare the performance of MARBERT, MARBERTv2 (A) and MARBERTv2 (B).

In Section 2, we outline previous and more recent studies on dialect identification. In Section 3, we illustrate a thorough examination of the dataset. In Section 4 we describe the system and the results. Lastly, Section 5 presents our conclusion and proposes potential avenues for future research.

2 Related Work

Arabic exists in three main forms: Classical Arabic (CA), Modern Standard Arabic (MSA), and Dialectal Arabic (DA). Although CA and MSA have been thoroughly explored in previous research, interest in DA has recently risen due to limited resources (Holes, 2004; Brustad, 2000).

The initial research on DA was regional (Gadalla and ElMaraghy, 1997; Diab et al., 2010), later expanding to multi-dialectal studies (Zaidan and Callison-Burch, 2011; Elfardy et al., 2014;

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 637–641
December 7, 2023 ©2023 Association for Computational Linguistics

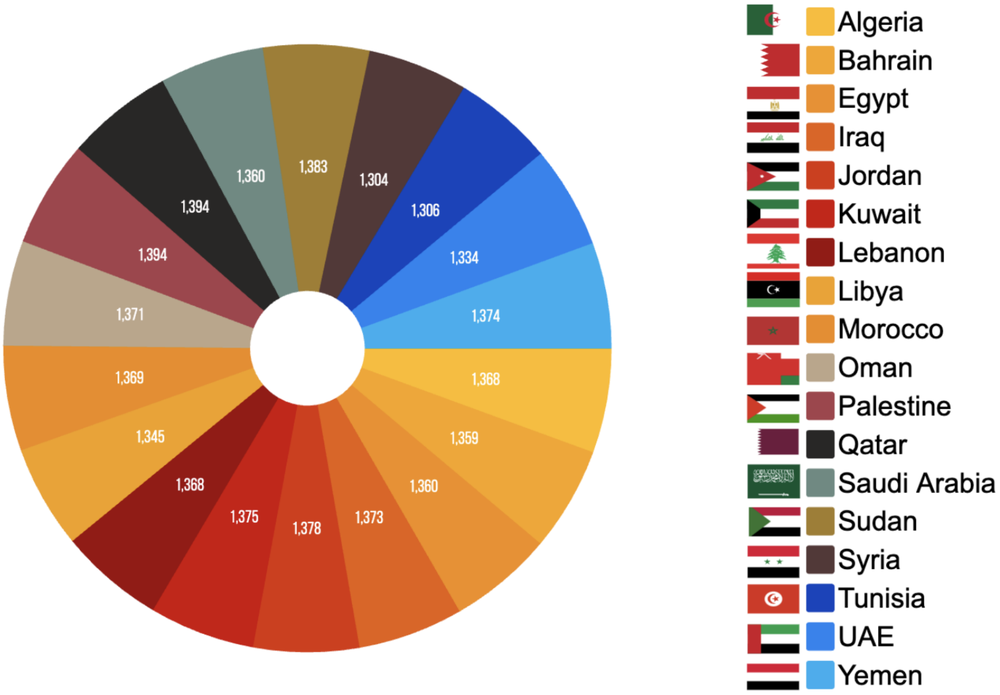


Figure 1: Statistics about tweets distribution in train and development sets.

Bouamor et al., 2014). The VarDial workshop highlighted the identification of dialects using acoustic and phonetic traits (Zampieri et al., 2017).

MADAR (Bouamor et al., 2018) provided enriched dialect data, but faced authenticity questions in online contexts. Recent work has taken advantage of the vast Twitter datasets (Mubarak and Darwish, 2014; Abdelali et al., 2021), with Althobaiti (2022) introducing an unsupervised dialect-tagging approach. Further, Abdul-Mageed et al. (2020b) investigated city-specific dialect variations.

NADI's initiatives produced notable datasets on Arabic dialect identification, including a detailed review by Althobaiti (2020). NADI 2020 collaborated with WANLP 2020, leading to the categorization of dialects from 21 Arab countries via Twitter. NADI 2021, in association with WANLP 2021, improved its dataset, distinguishing between MSA and DA. This led to the development of four specific subtasks (Abdul-Mageed et al., 2021). In NADI 2022, the focus had shifted to sentiment analysis of data tagged with dialects. In particular, Alsudais et al. (2022) integrated the MADAR and NADI datasets into their research. Lastly, NADI 2023 introduced three subtasks: country-level dialect identification and closed- and open-speech machine translation from four dialects to MSA.

3 Data

This section provides a detailed explanation of the dataset made available by the NADI shared task organizers.

Data Attributes:

- **ID:** A numerical index assigned to each data point.
- **Tweet:** An Arabic tweet written in various dialects.
- **Label:** Indicates the specific dialect corresponding to one of the 18 countries (e.g., UAE, Morocco, etc.).

Dataset Size:

The statistics of the dataset for this task are detailed in Figure 2. In total we have slightly more than 28K. We used an external dataset from the set, which is provided by organizers (NADI-2021-TWT). The distribution of labels within the training and development sets can be seen in Figure 1. In particular, the dataset has a balanced distribution.

4 System Description and Results

4.1 System Description

For evaluation, we use the official evaluation scorers provided for the shared task. The primary measure for our subtask is an F1 score. Our model was executed on 2 NVIDIA Tesla T4 (16GB) GPU.

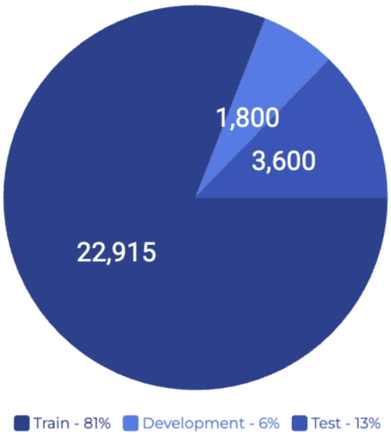


Figure 2: Statistics about tweets distribution in train/development/test sets.

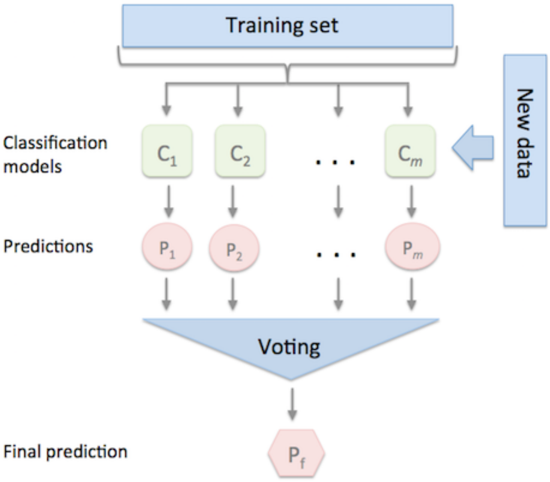


Figure 3: Majority voting architecture. Source: www.researchgate.net

We take advantage of the majority vote technique in ensemble learning as an alternative method (Dietterich et al., 2002; Sagi and Rokach, 2018; Zhu et al., 2021). We opted for the majority-voting ensemble due to our balanced dataset. This technique aggregates predictions from multiple models for a given input. The architecture is shown in Figure 3, where the final prediction is derived from the class or result that receives the majority vote from the ensemble (Da San Martino et al., 2023; Azizov et al., 2023; Barrón-Cedeño et al., 2023a,b).

Consider m classifiers, $C_1, C_2, \ldots, C_m$, predicting the class label for an input x as $P_1, P_2, \ldots, P_m$. The majority-voting classifier gives the final class label, P_f, based on the most frequent prediction:

$$P_f = \text{mode}(P_1, P_2, \ldots, P_m) \qquad (1)$$

For our task, we opt for hard voting, addressing concerns of classifier calibration and avoiding potential overconfidence in predictions. This ensures that the majority consensus dictates the final prediction. Although our method relies on the most reliable framework in the case of varying model predictions.

The following is the experimental setup for our models:

MARBERT: This model was trained for 1 epoch using a learning rate of 5e-5 and a weight decay of 0.001.

MARBERTv2 (A): MARBERTv2 was trained for 2 epochs with a weight decay of 0.0.

MARBERTv2 (B): This version of MARBERTv2 was trained for 2 epochs with a weight decay of 0.001.

Unless specified otherwise, all other hyperparameters were kept at their default values.

All these mentioned models were combined using the architecture shown in Figure 3. In case of differing predictions across all three models, we prioritize the prediction MARBERTv2 (B) due to its superior performance.

To maximize performance, we used a customized training approach in our study with three model versions (MARBERT, MARBERT A, and MARBERT B). The models showed inherent similarities, but different optimal training epochs were identified: MARBERT peaked at the first epoch, whereas both MARBERT A and MARBERT B performed optimally in the second epoch. To avoid overfitting, training was stopped in these instances.

4.2 Results

In this section, we discuss the results of our models.

We experimented with the development set, since we used it as a test set, and from the train set we cut 10% out of the total tweets for the development set.

MARBERT vs. MARBERTv2 (A): A comparison between the original MARBERT model and its first variant MARBERTv2 (A) shows noticeable improvements in all measures in the latter. An F1 score sees an increase of 1.99 percentage points, moving from 82.40 in MARBERT to 84.39 in MARBERTv2 (A). Similarly, the precision in MARBERTv2 (A) is higher by 2.14 percentage points than the original MARBERT, which is 84.73.

MARBERTv2 (A) vs. MARBERTv2 (B):

	F1 Score	Accuracy	Precision	Recall
MARBERT	82.40	82.38	82.59	82.38
MARBERTv2 (A)	84.39	84.33	84.73	84.33
MARBERTv2 (B)	84.44	84.38	84.72	84.38
Majority voting	**85.90**	**85.83**	**86.12**	**85.83**

Table 1: Experimental results of our frameworks on development set.

When comparing the two versions of MARBERTv2, the improvements in the (B) version, although modest, are discernible. An F1 score is marginally better by 0.05 percentage points in the (B) version. The precision in MARBERTv2 (B) is nearly the same as its counterpart (A), but sees a tiny decrease of 0.01 percentage points. This suggests that the adjustments made between the two versions of MARBERTv2 led to slight improvements in certain areas, but had a negligible impact on precision.

MARBERTv2 (B) vs. Majority Voting: The ensemble model, using a majority voting approach, clearly outshines the best performing MARBERTv2 version. An F1 score in the majority voting approach is higher by a significant 1.46 percentage points compared to MARBERTv2 (B). The precision is also improved in the majority voting method by 1.4 percentage points, making it the most precise model among the ones evaluated.

Overall Observations: Across the board, each subsequent version of the model or approach appears to bring about performance improvements, with the majority-voting method standing out as the most effective.

Based on the leaderboard results, we secured the fifth rank. Our achieved an F1 score is 84.76. For other evaluation measures, we recorded an accuracy of 84.75, a precision of 84.95, and a recall of 84.75.

5 Conclusion and Future Work

In this paper, we discussed our approach for subtask 1 of the shared task NADI in Arabic Dialect Identification. We used the majority-voting ensemble with the MARBERT and MARBERTv2 (A) and MARBERTv2 (B) models and according to the official leaderboard results, our system achieved an F1 score of **84.76** outperforming two-thirds of the participating teams. We also detailed a series of experiments and made comparisons of our models with a majority-voting ensemble.

In future work, we plan to enhance our ensemble approach with advanced transformer architectures (e.g., mBERT and XLM-RoBERTa) and data augmentation specific to Arabic dialects (e.g., back-translation or dialectical synonym replacement). Moreover, we would like to investigate classifier calibration and soft voting.

References

Ahmed Abdelali, Sabit Hassan, Hamdy Mubarak, Kareem Darwish, and Younes Samih. 2021. Pre-training bert on arabic tweets: Practical considerations. *arXiv preprint arXiv:2102.10684*.

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

Muhammad Abdul-Mageed, Chiyu Zhang, Houda Bouamor, and Nizar Habash. 2020b. NADI 2020: The first nuanced Arabic dialect identification shared task. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 97–110, Barcelona, Spain (Online). Association for Computational Linguistics.

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, Houda Bouamor, and Nizar Habash. 2021. NADI 2021: The second nuanced Arabic dialect identification shared task. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 244–259, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, Houda Bouamor, and Nizar Habash. 2022. NADI 2022: The third nuanced Arabic dialect identification shared task. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 85–97, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Abdulkareem Alsudais, Wafa Alotaibi, and Faye Alomary. 2022. Similarities between arabic dialects: Investigating geographical proximity. *Information Processing & Management*, 59(1):102770.

Maha J Althobaiti. 2020. Automatic arabic dialect identification systems for written texts: A survey. *arXiv preprint arXiv:2009.12622*.

Maha J Althobaiti. 2022. Creation of annotated country-level dialectal arabic resources: An unsupervised approach. *Natural Language Engineering*, 28(5):607–648.

Dilshod Azizov, S Liang, and P Nakov. 2023. Frank at checkthat! 2023: Detecting the political bias of news articles and news media. *Working Notes of CLEF*.

Alberto Barrón-Cedeño, Firoj Alam, Tommaso Caselli, Giovanni Da San Martino, Tamer Elsayed, Andrea Galassi, Fatima Haouari, Federico Ruggeri, Julia Maria Struß, Rabindra Nath Nandi, et al. 2023a. The clef-2023 checkthat! lab: Checkworthiness, subjectivity, political bias, factuality, and authority. In *European Conference on Information Retrieval*, pages 506–517. Springer.

Alberto Barrón-Cedeño, Firoj Alam, Andrea Galassi, Giovanni Da San Martino, Preslav Nakov, Tamer Elsayed, Dilshod Azizov, Tommaso Caselli, Gullal S Cheema, Fatima Haouari, et al. 2023b. Overview of the clef–2023 checkthat! lab on checkworthiness, subjectivity, political bias, factuality, and authority of news articles and their source. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 251–275. Springer.

Houda Bouamor, Nizar Habash, and Kemal Oflazer. 2014. A multidialectal parallel corpus of arabic. In *LREC*, pages 1240–1245.

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).

Kristen Brustad. 2000. *The syntax of spoken Arabic: A comparative study of Moroccan, Egyptian, Syrian, and Kuwaiti dialects*. Georgetown University Press.

Giovanni Da San Martino, Firoj Alam, Maram Hasanain, Rabindra Nath Nandi, Dilshod Azizov, and Preslav Nakov. 2023. Overview of the clef-2023 checkthat! lab task 3 on political bias of news articles and news media. *Working Notes of CLEF*.

Mona Diab, Nizar Habash, Owen Rambow, Mohamed Altantawy, and Yassine Benajiba. 2010. Colaba: Arabic dialect annotation and processing. In *Lrec workshop on semitic language processing*, pages 66–74. Citeseer.

Thomas G Dietterich et al. 2002. Ensemble learning. *The handbook of brain theory and neural networks*, 2(1):110–125.

Heba Elfardy, Mohamed Al-Badrashiny, and Mona Diab. 2014. Aida: Identifying code switching in informal arabic text. In *Proceedings of The First Workshop on Computational Approaches to Code Switching*, pages 94–101.

Mohamed AE Gadalla and Waguih H ElMaraghy. 1997. Improving the accuracy of machined parametric surfaces using cutting force synthesis and surface offset techniques. In *ASME International Mechanical Engineering Congress and Exposition*, volume 26782, pages 181–187. American Society of Mechanical Engineers.

Clive Holes. 2004. *Modern Arabic: Structures, functions, and varieties*. Georgetown University Press.

Hamdy Mubarak and Kareem Darwish. 2014. Using twitter to collect a multi-dialectal corpus of arabic. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 1–7.

Omer Sagi and Lior Rokach. 2018. Ensemble learning: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1249.

Omar Zaidan and Chris Callison-Burch. 2011. The arabic online commentary dataset: an annotated dataset of informal arabic with high dialectal content. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 37–41.

Matteo Zampieri, Andrej Ceglar, Frank Dentener, and Andrea Toreti. 2017. Wheat yield loss attributable to heat waves, drought and water excess at the global, national and subnational scales. *Environmental Research Letters*, 12(6):064008.

Yadong Zhu, Xiliang Wang, Qing Li, Tianjun Yao, and Shangsong Liang. 2021. Botspot++: A hierarchical deep ensemble model for bots install fraud detection in mobile advertising. *ACM Transactions on Information Systems (TOIS)*, 40(3):1–28.

NLPeople at NADI 2023 Shared Task: Arabic Dialect Identification with Augmented Context and Multi-Stage Tuning

Mohab Elkaref[1] and **Movina Moses**[2] and **Shinnosuke Tanaka**[1] and
James Barry[1] and **Geeth De Mel**[1]
IBM Research Europe[1] and IBM Research[2]
{mohab.elkaref, movina.moses, shinnosuke.tanaka, james.barry}@ibm.com
geeth.demel@uk.ibm.com

Abstract

This paper presents the approach of the **NLPeople** team to the Nuanced Arabic Dialect Identification (NADI) 2023 shared task. Subtask 1 involves identifying the dialect of a source text at the country level. Our approach to Subtask 1 makes use of language-specific language models, a clustering and retrieval method to provide additional context to a target sentence, a fine-tuning strategy which makes use of the provided data from the 2020 and 2021 shared tasks, and finally, ensembling over the predictions of multiple models. Our submission achieves a macro-averaged F1 score of 87.27, ranking 1st among the other participants in the task.

1 Introduction

The task of dialect identification involves predicting the source variety of a given text or speech segment. Recently, there have been a number of shared tasks that have focused on predicting the nuanced dialects of Arabic (Abdul-Mageed et al., 2020, 2021, 2022). Arabic can be broadly categorised into the following three languages: Classical Arabic (CA), Modern Standard Arabic (MSA), and Dialectal Arabic (DA), where DA can be further sub-categorised based on the geographic region where it is spoken.

Arabic dialect identification represents a challenging task for a number of reasons. Firstly, Arabic languages exhibit rich morphology, where words are highly-inflected, which can lead to issues related to data sparsity. Another challenge present in the NADI shared tasks, is that the text to be classified consists of tweets, a form of user-generated content (UGC). As pointed out by Cassidy et al. (2022), UGC contains features not typically found in other forms of text data such as spoken language and standardised written language. For instance, UGC in the form of tweets tend to be short, exhibit non-standard use of grammar, and contain increased usage of emojis and abbreviated text.

This paper describes the **NLPeople** submission to the 2023 NADI shared task (Abdul-Mageed et al., 2023). In order to deal with the challenge of Arabic dialect identification, we develop a system which makes use of the following components:

- **Language-specific language models:** We utilise language models trained on Arabic and Arabic UGC.
- **Additional context retrieval:** We retrieve similar texts from a reference set for a given target text and append the retrieved text and corresponding labels as additional input.
- **Staged fine-tuning on additional data:** We first perform generic fine-tuning on the 2020 and 2021 data that was made available to participants, followed by a final round of fine-tuning on the 2023 data.
- **Model ensembling**: We combine the predictions of numerous models.

We empirically show that each of these components improves upon the metric of macro-averaged F1 score over the included dialects. Overall, our results rank 1st among 16 participants with a macro-averaged F1 score of 87.27.

2 Dataset

The label distribution of the used datasets are given in Figure 1. For the NADI-2023 data, a total of 18 country-level labels are present, and the training and development data have an equal distribution of 1000 and 100 labels, respectively. Additionally, we include the NADI-2020 and NADI-2021 datasets that were released by the shared task organisers as additional data for training our models. These datasets exhibit an imbalanced label distribution compared to the NADI-2023 data, with the UAE label being absent, and certain dialects such as Bahranian and Qatari being less represented than

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 642–646
December 7, 2023 ©2023 Association for Computational Linguistics

dialects such as Egyptian and Saudi Arabian. The total number of unlabelled instances in the 2023 test set is 3600.

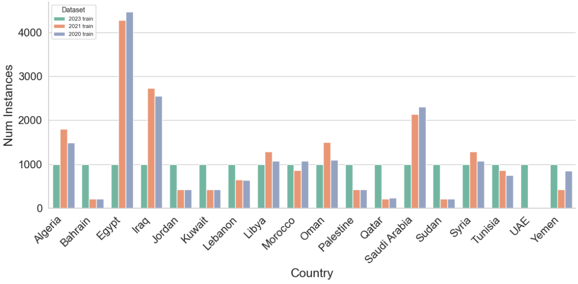


Figure 1: Number of instances per dialect across the 2023, 2021 and 2020 training data.

3 System Description

3.1 Initial System

In this section we discuss the NLPeople system. At its core, our model relies on a Transformer encoder model (Vaswani et al., 2017) to encode a sequence of words into a sequence of hidden states, which are passed to a feedforward network to predict the label. More formally, given a sentence $X = x_1, \ldots, x_n$ containing n words, a pre-trained language model LM is used to extract features $[x^l_{CLS}, x^l_1, \ldots, x^l_n] = LM^l([CLS], x_1, \ldots, x_n)$, where l is the last layer of the encoder, and x^l_i is the layer-l vector corresponding to the first word-piece in the word x_i. We take the output vector corresponding to the special [CLS] token x^l_{CLS} and pass this vector into a two-layer feedforward network to produce scores for all possible tags.

Model hyperparameters are given in Table 1. The models were trained on an NVIDIA A100 GPU with 80GB of VRAM. Training took around 1.5 hours for the 2023 training data.

Hyperparameter	**Value**
Learning Rate	1e-5
Batch Size	8
Transformer embedding size (base)	768
Transformer embedding size (large)	1024
Feedforward Size	768
Num. Feedforward Layers	2
Feedforward activation (first-layer)	ReLU
Dropout Rate	0.3
Epochs	10

Table 1: Model Hyperparameters

3.2 System Enhancements

Language-specific Language Models In order to deal with the morphological complexity of the Arabic dialects, we utilise pre-trained language models trained on Arabic. In particular, we experiment with the `MARBERTv2`[1] and `bert-large-arabertv02-twitter`[2] models. In the case of the `bert-large-arabertv02-twitter` model, it is trained on Twitter data which should be similar to the domain of the shared task data.

Additional Context Retrieval Given that the shared task data consists of short texts in the form of tweets, we experiment with adding context to the input data. For a given target item, which in this case can be a text instance from the training, development or test set, we retrieve the top-k most similar texts from the training data. Specifically, the fine-tuned `MARBERTv2` model is employed to obtain dense vectors for all instances in the training, development and test data, and for a given target item, instances from the train set with the k-nearest Euclidean distances are appended after the target text. In the additional context, the corresponding labels of the retrieved items are also included as special tokens. The augmented instances are shown below where we refer to x_i as a target text, y_i as the target label, and x_{top_j} and y_{top_j} represent the top-jth retrieved item's text and label, respectively:

$$x_i, [y_{top_1}]x_{top_1}, \ldots, [y_{top_k}]x_{top_k} = y_i$$

Training and evaluation then proceeds as normal using the augmented train, development and test sets.

Staged Fine-tuning on Additional Data Along with the 2023 training and development data, the shared task organisers provided participants with training data from the 2020 and 2021 shared tasks. We conduct a number of experiments involving the mixture of data to use for model training, and also consider a staged fine-tuning approach where the model is first fine-tuned on the data from the previous years, and is then fine-tuned on the current 2023 data.

Model Ensembling We consider model ensembling via two approaches: 1) *score ensembling*

[1] `https://huggingface.co/UBC-NLP/MARBERTv2`

[2] `https://huggingface.co/aubmindlab/bert-large-arabertv02-twitter`

Model	Type	Macro F1
arabertv02	MLM	76.62
arabertv02-twitter	MLM	80.61
AraT5-base	Gen	75.67
AraT5-tweet-base	Gen	78.53
JABER	MLM	78.95
MARBERT	MLM	84.65
MARBERTv2	MLM	**86.05**
XLM-R	MLM	68.44

Table 2: Development scores using different pre-trained language models. MLM: masked language model, Gen: generative model.

where we stack the raw score predictions from multiple models and select the highest-scoring label, and 2) *label ensembling* where we perform majority voting on the predicted label for each test instance.

4 Results and Discussion

4.1 Development Experiments

Choice of Language Model The first set of experiments involve the choice of language model. The results are reported in Table 2. We considered two types of language models: masked language models (MLMs) and generative language models (Gen). In the former, the model is used to encode an input sentence which is then fed to a classifier component (Section 3.1). In the latter, the model is tasked with *generating* the output label in an auto-regressive manner given an input sentence.[3]

For the MLM models, when considering the arabert models, we note that the version trained on Twitter data performs better on the shared task data (80.61 vs. 76.62 F1). The MARBERT models perform the best among the Arabic language models, where the MARBERTv2 model has an F1 score of 86.05, the highest-scoring model overall. For the generative modelling approach, we tried various T5 variants, where the tweet content is fed as input and the model is tasked with generating the label. We also note that the variant of this model trained on Twitter data performs better (78.53 vs 75.67 F1). Finally, we consider a multilingual MLM baseline in XLM-R which performs worse than the Arabic language models with an F1 score of 68.44.

Context size	Macro F1
none	86.05
1	86.58
5	86.71
10	**86.79**

Table 3: Development scores using different counts for the number of retrieved texts.

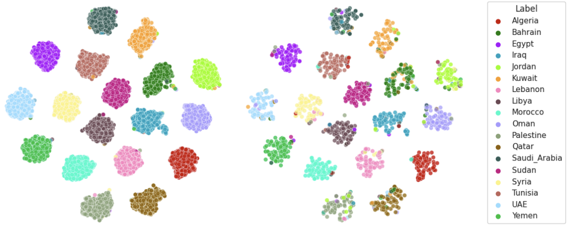


Figure 2: t-SNE visualisation of embeddings produced by fine-tuned MARBERTv2. The left plot corresponds to the training set, while the right plot corresponds to the development set.

Additional Context Retrieval The results concerning additional context retrieval are given in Table 3. We use the best-performing language model from the previous set of experiments, i.e. the MARBERTv2 model. Firstly, using the standard 2023 training data provides an F1 score of 86.05. By retrieving the top-1 most similar context, the score increases to 86.58. When retrieving the top-5 and top-10 most similar contexts to a target item, the score increases to 86.71 and 86.79 F1, respectively. To demonstrate the effectiveness of the retrieval, we present t-SNE plots depicting the embeddings of the training and development sets in Figure 2. Notably, distinct clusters form for each label, revealing that data points in proximity to target sentences often belong to the same cluster.

Staged Fine-tuning on Additional Data We experiment with using different variations of the provided data. The results are given in Table 4. We find that adding the 2020 data to the 2023 data harms performance when compared to training on the 2023 data alone, where the F1 score decreases from 86.05 to 83.51. The same is the case when adding the 2021 data to the 2023 data and adding both the 2020 and 2021 data to the 2023 data. In a final experiment, we first trained a model on the 2020 data, which was further fine-tuned on the 2021 data, and finally fine-tuned on the 2023 data. Interestingly, performing generic fine-tuning on the

[3] To fine-tune the T5 models, we use the resources released by Nagoudi et al. (2022).

Additional Data	Macro F1
2023	86.05
2023, 2020	83.51
2023, 2021	83.19
2023, 2021, 2020	83.01
Three-staged finetune	**87.02**

Table 4: Development scores using different sources of data.

Ensemble type	Count	Macro F1
none	1	86.05
score	5	86.78
score	10	**86.88**
label	5	86.07
label	10	86.74

Table 5: Development scores using different ensemble techniques.

noisier additional data followed by fine-tuning on the task-specific data results in the best-performing model with an F1 score of 87.02.

Model Ensembling To examine the effect of model ensembling, we utilised a selection of models that were trained as part of a hyperparameter sweep for the `MARBERTv2` model. The models were trained between 20-50 epochs, had a batch size of either 8 or 16, and used the CLS representation for classification. We consider two types of model ensembling: 1) score-ensembling where the scores of multiple models are stacked, and 2) label-ensembling where we perform majority voting on the predicted labels. Results are given in Table 5.

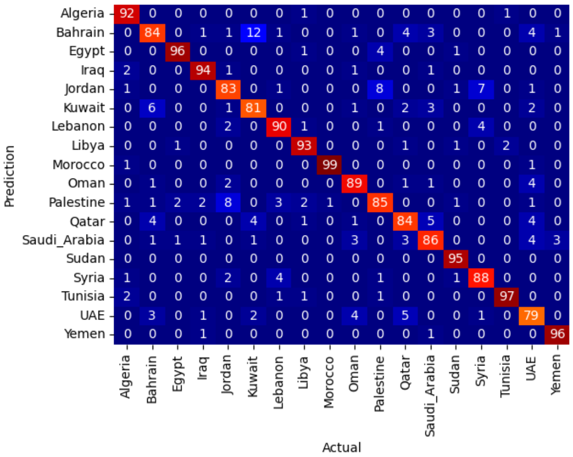
We find that combining model predictions is helpful in all cases. When considering score-based ensembling, the ensemble with 10 predictions performs best with a score of 86.88, which is the best score overall for this experiment. When considering label-based ensembling, the ensemble with 10 predictions performs best with a score of 86.74.

4.2 Official Results

Submitted System We trained up to 10 models for each setting using different random seeds through language model selection, additional context retrieval, staged fine-tuning, and combinations thereof. For the ensemble, from the pool of all trained models, we randomly selected between 2 and 20 models and recorded the development set score from the particular ensemble. We repeated this process until the highest-scoring ensemble was found. Details of the models used for the highest-performing system are presented in Table 6. This system employed `MARBERTv2` and `arabertv02-twitter` as language models, utilising both regular and staged fine-tuning techniques, resulting in remarkable performance through score ensembling. Unexpectedly, despite achieving high individual scores, additional context models were absent from this top ensemble. Individual model F1 scores ranged from 81.30 to 86.05 and extended to 89.56 through ensembling.

Language Model	Additional Data	Count	Macro F1 (range)
`arabertv02-twitter`	2023	5	81.30 - 81.90
`arabertv02-twitter`	Three-staged	3	81.47 - 81.49
`MARBERTv2`	2023	5	85.25 - 86.05
`MARBERTv2`	Three-staged	2	85.57 - 86.04

Table 6: 15 models used for the score ensemble which achieved the highest performance.



Prediction	Algeria	Bahrain	Egypt	Iraq	Jordan	Kuwait	Lebanon	Libya	Morocco	Oman	Palestine	Qatar	Saudi_Arabia	Sudan	Syria	Tunisia	UAE	Yemen
Algeria	92	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0
Bahrain	0	84	0	1	1	12	1	0	0	1	0	4	3	0	0	0	4	1
Egypt	0	0	96	0	0	0	0	1	0	0	4	0	0	1	0	0	0	0
Iraq	2	0	0	94	1	0	0	0	0	1	0	0	1	0	0	0	0	0
Jordan	1	0	0	0	83	0	1	0	0	0	8	0	0	1	7	0	1	0
Kuwait	0	6	0	0	1	81	0	0	0	1	0	2	3	0	0	0	2	0
Lebanon	0	0	0	0	2	0	90	1	0	0	1	0	0	0	4	0	0	0
Libya	0	0	1	0	0	0	0	93	0	0	0	1	0	1	0	2	0	0
Morocco	1	0	0	0	0	0	0	0	99	0	0	0	0	0	0	0	1	0
Oman	0	1	0	0	2	0	0	0	0	89	0	1	1	0	0	0	4	0
Palestine	1	1	2	2	8	0	3	2	1	0	85	0	0	1	0	0	1	0
Qatar	0	4	0	0	0	4	0	1	0	1	0	84	5	0	0	0	4	0
Saudi_Arabia	0	1	1	1	0	1	0	0	0	3	0	3	86	0	0	0	4	3
Sudan	0	0	0	0	0	0	0	0	0	0	0	0	0	95	0	0	0	0
Syria	1	0	0	0	2	0	4	0	0	0	1	0	0	1	88	0	0	0
Tunisia	2	0	0	0	0	0	1	1	0	0	1	0	0	0	0	97	0	0
UAE	0	3	0	1	0	2	0	0	0	4	0	5	0	0	1	0	79	0
Yemen	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	96

Figure 3: Confusion matrix for the submitted system on the development set.

The confusion matrix for the submitted system is shown in Fig 3. Among the 18 labels, it indicates that predictions are accurate for 90% or more for 9 of these labels. Particularly, Morocco achieves a remarkable accuracy by correctly predicting 99 out of 100 instances. On the other hand, UAE exhibits the highest error rate, with results falling below 80%. In the pair analysis, the most significant misprediction was observed, where 12% of Kuwait data was incorrectly labelled as Bahrain.

Results on the Test Set The official results on the final test set for the top five teams are presented in Table 7. Our system outperformed in not only

Team	Macro F1	Accuracy	Precision	Recall	Rank
rematchka	86.18	86.17	86.29	86.17	2
Arabitools	85.86	85.81	86.10	85.81	3
SANA	85.43	85.39	85.60	85.39	4
Frank	84.76	84.75	84.95	84.75	5
NLPeople (ours)	**87.27**	**87.22**	**87.37**	**87.22**	1

Table 7: Top five results on the test set from the official leaderboard.

F1 score but also across all other metrics.

5 Conclusion

In this work, we described the NLPeople submission to the 2023 NADI shared task (Abdul-Mageed et al., 2023). Our submission combines four different techniques: (1) language-specific language models (2), similar context retrieval (3), a staged fine-tuning approach over all available data, and (4) model ensembling. We demonstrated that each of the above components impacts our evaluation scores positively, and our final submission which uses the above techniques achieves a score of 87.27, which ranks 1st among 16 participants. Furthermore, our system is less impacted by the short input length due to our step of augmenting the input sentence with retrieved similar contexts.

Limitations

In the context of this study, it is essential to consider several limitations. Firstly, our retrieval methodology entails embedding the train, development, and test sets separately for the additional context retrieval method. This process imposes additional computational demands. Secondly, our adoption of staged fine-tuning introduces a similar computational overhead by training on more data. Furthermore, our findings have demonstrated that incorporating supplementary data adversely affects performance. Therefore, future works in this domain should carefully consider their data augmentation strategy, as indiscriminate inclusion of additional data may not yield improved results. Lastly, our ensemble approach, while effective, is computationally intensive. This technique may pose challenges in resource-constrained or time-sensitive scenarios where loading and maintaining multiple models concurrently may be impractical.

References

Muhammad Abdul-Mageed, AbdelRahim Elmadany, Chiyu Zhang, ElMoatez Billah Nagoudi, Houda Bouamor, and Nizar Habash. 2023. NADI 2023: The Fourth Nuanced Arabic Dialect Identification Shared Task. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*.

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, Barcelona, Spain (Online). Association for Computational Linguistics.

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, Houda Bouamor, and Nizar Habash. 2021. NADI 2021: The second nuanced Arabic dialect identification shared task. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 244–259, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, Houda Bouamor, and Nizar Habash. 2022. NADI 2022: The third nuanced Arabic dialect identification shared task. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 85–97, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Lauren Cassidy, Teresa Lynn, James Barry, and Jennifer Foster. 2022. TwittIrish: A Universal Dependencies treebank of tweets in Modern Irish. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6869–6884, Dublin, Ireland. 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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

USTHB at ArAIEval23 Shared Task: Disinformation Detection System based on Linguistic Feature Concatenation

Mohamed Lichouri
LCPTS-USTHB, Algiers, Algeria
mlichouri@usthb.dz

Khaled Lounnas, Aicha Zitouni
LCPTS-USTHB, Algiers, Algeria
CRSTDLA, Algiers, Algeria
{k.lounnas, a.zitouni}@crstdla.dz

Houda Latrache
CRSTDLA, Algiers, Algeria
h.latrache@crstdla.dz

Rachida Djeradi
LCPTS-USTHB, Algiers, Algeria
rdjeradi@usthb.dz

Abstract

In this research paper, we undertake a comprehensive examination of several pivotal factors that impact the performance of Arabic Disinformation Detection in the ArAIEval'2023 shared task. Our exploration encompasses the influence of surface preprocessing, morphological preprocessing, the FastText vector model, and the weighted fusion of TF-IDF features. To carry out classification tasks, we employ the Linear Support Vector Classification (LSVC) model. In the evaluation phase, our system showcases significant results, achieving an F_1 micro score of 76.70% and 50.46% for binary and multiclass classification scenarios, respectively. These accomplishments closely correspond to the average F_1 micro scores achieved by other systems submitted for the second subtask, standing at 77.96% and 64.85% for binary and multiclass classification scenarios, respectively.

1 Introduction

In recent years, the detection of disinformation in digital content has become a critical challenge at the intersection of natural language processing and information security, spurred by the growing influence of online platforms (Shu et al., 2020). The Arabic-speaking digital landscape, in particular, has witnessed an alarming increase in susceptibility to the dissemination of false or misleading information, a phenomenon well-documented in recent research (Harrag and Djahli, 2022). The ramifications of disinformation extend beyond individual deception; they cover broader societal consequences, affecting public opinion, social cohesion, and even national security.

Recognizing the gravity of this issue, we actively participate in the inaugural shared task organized by ArAIEval'2023, which focuses on disinformation detection in Arabic text (Hasanain et al., 2023). Our engagement in this task reflects our commitment to addressing this pressing challenge. By harnessing advanced natural language processing techniques and machine learning models, we endeavor to contribute to the development of effective disinformation detection systems tailored to the nuances of the Arabic language. Through rigorous experimentation and evaluation, we aim to enhance our understanding of the complexities involved and offer practical solutions to safeguard the integrity of digital discourse and information dissemination in the Arabic-speaking world.

To combat the proliferation of disinformation in Arabic text, a growing number of research has been dedicated to developing robust and effective detection systems (Alam et al., 2022; Mubarak et al., 2023). Much like the endeavors undertaken in the field of Arabic dialect identification (Lichouri et al., 2021b), disinformation detection in Arabic requires a nuanced understanding of the language's intricacies (Nagoudi et al., 2020), as well as the ability to sift through vast amounts of textual data (Himdi et al., 2022) to identify instances of deceptive or misleading content.

In this paper, we embark on an extensive exploration of disinformation detection in Arabic, drawing inspiration from the methodologies and techniques employed in previous shared tasks (Lichouri et al., 2020). Leveraging these insights, we aim to build upon existing research and contribute to the ongoing efforts to enhance the accuracy and effectiveness of disinformation detection systems in Arabic text.

Our study encompasses a comprehensive analysis of various factors influencing the performance of Arabic disinformation detection, including surface and morphological preprocessing techniques (Lichouri et al., 2021a), feature engineering strategies (Fouad et al., 2022), and the implementation of

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 647–651
December 7, 2023 ©2023 Association for Computational Linguistics

state-of-the-art machine learning models. Through rigorous experimentation and evaluation, we seek to provide valuable insights and practical solutions that can aid in the identification and mitigation of disinformation.

This paper is organized as follows: Section 2 offers insights into the dataset we have employed. Moving on to Section 3, we introduce our proposed system, which includes details about the cleaning and preprocessing steps discussed in Section 3.1. The process of feature engineering is elucidated in Section 3.2. Section 3.3 is dedicated to a comprehensive discussion of our findings. Finally, we wrap up the paper in Section 4 with a conclusive summary of our contributions and key findings.

2 Description of the Dataset

A disinformation dataset constitutes a crucial resource for studying and comprehending the multifaceted landscape of misinformation, misleading content, and fabricated information within various digital platforms. Such datasets encompass a diverse array of textual, visual, and multimedia content intentionally designed to deceive, mislead, or manipulate audiences. These datasets serve as invaluable assets for researchers, data scientists, and machine learning practitioners engaged in the development of advanced algorithms and models aimed at detecting, analyzing, and combating disinformation. By analyzing patterns, linguistic cues, and contextual elements within disinformation datasets, researchers gain insights into the tactics, strategies, and evolving nature of disinformation campaigns, thereby contributing to the enhancement of society's ability to discern and mitigate the harmful impacts of deceptive content in an increasingly interconnected information landscape.

Additional information regarding this dataset can be found in Table 1, where we took part for the first time this year in both editions of the Disinformation Detection Definition shared task. This task involves classifying binary and fine-grained disinformation categories based solely on the text of a tweet. Please note that these statistics pertain to the dataset after we removed punctuation and emojis. Imbalanced datasets can have a pronounced effect on system performance, causing the development of biased models that prioritize the dominant class (e.g., "no-disinformation" in binary classification and "HS" in multi-class classification). This can result in decreased predictive accuracy for the underrepresented classes, such as "disinformation" in binary classification, "Rumor", and "Spam" in the multi-class scenario, and compromised decision-making in applications like fraud detection or medical diagnosis. Addressing class imbalance through techniques like oversampling, undersampling, or using appropriate evaluation metrics is crucial for more equitable and accurate model outcomes.

3 Proposed system

3.1 Data Cleaning and Preprocessing

In the challenging domain of disinformation detection within Arabic text, it becomes imperative to adeptly capture essential information while efficiently removing undesirable elements. This task is known for its complexity and nuance, demanding a detailed approach. To address this challenge, we have implemented a two-phase preprocessing strategy:

Phase 1: Surface Preprocessing - In this initial phase, we execute a range of foundational procedures:

- *Arabic Letter Normalization*: Ensuring consistency in Arabic script characters (Sallam et al., 2016).
- *Punctuation and Emoji Removal*: Eliminating punctuation marks and emoticons (Shiha and Ayvaz, 2017).
- *Stop Words Removal*: Handling common words that do not contribute substantially to meaning.
- *Diacritics Removal*: Removing diacritical marks for text clarity (Jbara et al., 2009).
- *Exclusion of Non-Arabic Content*: Ensuring that only Arabic text remains (Omar et al., 2021).

These collective measures ensure text clarity, uniformity, and the removal of any distractions.

Phase 2: Morphological Preprocessing - In this phase, our focus shifts to the intricacies of language. Here, we employ the following techniques:

- *Lemmatization*: Simplifying word forms to their base or dictionary form (El Kah and Zeroual, 2021).
- *Stemming*: Reducing words to their root forms, aiding in the identification of core word meanings and structures (Atwan et al., 2021).

Table 1: ArAIEval (Task2A/2B) dataset statistics where : Task2A for Binary classification whereas Task2B for Multiclass classification problem.

	Train	Dev	Test
# sentences	14147/2656	2115/397	3729/876
# words	324727/68073	48917/10062	100646/27312
Max # word per sentence Min # word per sentence	65/67 0 / 1	65/59 0 / 1	62 /62 1 / 1
Max # char per sentence	280/290	280 /285	311 /311
Min # char per sentence	0 / 3	0 / 3	2 / 2

Table 2: The various combinations and parameter used in our work

Settings	Range
ngram_range	(m,n) with m=1 to 3 and n=1 to 10
tfidf_weights	0.5 - 1
tfidf max_features	1000 -25000
SVM	C=100, gamma=1-10
fasttext_supervised	epoch=100, loss='ova'

fasttext_unsupervised	epoch=100, ws=6
	model='skipgram'
	dim=1000

Throughout both phases, we intricately harmonize and fine-tune various techniques to arrive at the optimal configuration for our preprocessing pipeline.

3.2 Feature engineering

Our system operates through a well-defined structure consisting of four distinct phases, offering the flexibility to be applied individually or collectively. The initial two phases, Surface Preprocessing and Morphological Preprocessing, have been expounded upon in the previous section. The subsequent phases are detailed as follows:

Phase 3: Feature Extraction - In this stage, we employ a dual-model approach. Firstly, the FastText model undergoes comprehensive training in two modes: supervised and unsupervised, drawing from the training dataset. Then, we use this model to extract features from both the development and test datasets. Secondly, we leverage the TF-IDFVectorizer, an adept tool offering three distinct analyzers (Word, Char, and Char_wb), each encompassing variable n-gram ranges. As a default configuration, we combine these three TF-IDF features, affording them equal weights, all set to 1.

Phase 4: Weighted Fusion - In this phase, we combined the three TF-IDF features, supported by a weight vector featuring three distinctive values (w1, w2, w3) that correspond to the Word, Char, and Char_wb TF-IDF features, respectively.

Having presented these four distinctive phases, we executed four designed experiments that were inspired by our prior works (Lichouri et al., 2018; Abbas et al., 2019; Lichouri and Abbas, 2020a), where each embody distinct configurations:

Experiment 1 (Lichouri et al., 2021a; Lichouri and Abbas, 2020b): In this first experiment, we initiated with the first phase, by considering all the possible permutations of surface processing techniques. Following this, we considered the third phase, marked by the employment of a union of TF-IDF features. During the feature extraction process, we explored a range of n-gram values, spanning from $n = 1$ to 10. Finally, we finished by the training of the SVC classifier.

Experiment 2 (Lichouri et al., 2020): In this specific scenario, we worked with the second phase, by exploring various combinations of morphological processing techniques. Similar to Experiment 1, we progressed to the third phase, where we concat the TF-IDF features, all while varying the n-gram parameters. We then finished this experiment by training of the SVC classifier.

Experiment 3: For this unique experiment, we focused on the third phase, where we used FastText model for feature extraction, followed by the rigorous training of the SVC classifier.

Experiment 4 (Lichouri et al., 2021b): In this distinctive scenario, we executed the fourth phase, by applying a weighted union of TF-IDF features for feature extraction. Then, we concluded with

Task	**Binary**				**Multiple**			
Desc	MP	SP	F-Vec	WF	MP	SP	F-Vec	WF
Run 1	81,08	**81,23**	48,45	81,13	56,92	**57,43**	27.57	56,93
Run 2	81,08	81,18	48.27	78,91	56,92	56,68	27.68	56,92
Run 3	81,08	81,09	46.54	75,74	56,92	56,93	22.44	56,68

Table 3: The F1-micro percentages obtained using the proposed system Where: SP (Surface Preprocessing), MP (Morphological Preprocessing), F-Vec (Vectorisation), and WF (Weighted Fusion)

the training of the SVC classifier.

Following many iterations of these four experiments on both the training and development datasets, we recorded the best results attained for each experiment, along with the precise configurations that yielded these outcomes, as presented in Table 2.

3.3 Results and Discussion

In this study, we conducted a series of experiments aimed at detecting Arabic disinformation. These experiments were centered around the utilization of various descriptors, encompassing Surface Preprocessing (SP), Morphological Preprocessing (MP), the vectorisation model (F-Vec), and Weighted Fusion of TF-IDF (WF).

To explore the effectiveness of these descriptors, we employed a range of combinations and settings. This involved modifying n-gram values and TF-IDF weights to investigate the impact of word sequence length on results and term weighting in the text, respectively. Table 2 provides a comprehensive summary of the different combinations and parameters used in our study, while Table 3 presents the results obtained using these combinations.

Our experiments yielded valuable insights into the efficacy of various techniques for disinformation detection, specifically in binary and multiclass classification tasks. Notably, for the binary subtask, Surface Preprocessing demonstrated the highest performance, achieving an impressive F1-score of 81.23%. It was closely followed by the Weighted Union of TF-IDF features, with an F1-score of 81.13%, while Morphological Preprocessing exhibited slightly lower performance, resulting in an F1-score of 81.08%. Intriguingly, the FastText model underperformed in this context, attaining the lowest F1-score at 48.45%.

However, a fascinating observation emerged when we transitioned to the multiclass classification subtask. Surprisingly, the same observation held true, but the obtained results dropped significantly, by approximately 20%, compared to the binary case. We hypothesize that this decline in performance could be attributed to the imbalanced nature of the dataset, which has a more pronounced impact in the multiclass scenario.

4 Conclusion

In conclusion, our comprehensive analysis of key factors in Arabic Disinformation Detection has shed light on critical aspects that significantly influence performance. Through a meticulous exploration of surface preprocessing, morphological preprocessing, the FastText vector model, and the weighted fusion of TF-IDF features, we have gained valuable insights into their impact on classification tasks.

Our system's noteworthy achievement of an F_1 micro score of 76.70% and 50.46% for binary and multiclass classification setups, respectively, closely aligns with the performance of other systems submitted for the second subtask. This not only reaffirms the significance of surface preprocessing and weighted TF-IDF feature fusion but also positions them as robust techniques in the domain of Arabic Disinformation Detection.

References

Mourad Abbas, Mohamed Lichouri, and Abed Alhakim Freihat. 2019. St madar 2019 shared task: Arabic fine-grained dialect identification. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 269–273.

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, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Jaffar Atwan, Mohammad Wedyan, Qusay Bsoul, Ahmad Hamadeen, Ryan Alturki, and Mohammed

Ikram. 2021. The effect of using light stemming for arabic text classification. *International Journal of Advanced Computer Science and Applications*, 12(5).

Anoual El Kah and Imad Zeroual. 2021. The effects of pre-processing techniques on arabic text classification. *Int. J*, 10(1):1–12.

Khaled M Fouad, Sahar F Sabbeh, and Walaa Medhat. 2022. Arabic fake news detection using deep learning. *Computers, Materials & Continua*, 71(2).

Fouzi Harrag and Mohamed Khalil Djahli. 2022. Arabic fake news detection: A fact checking based deep learning approach. *Transactions on Asian and Low-Resource Language Information Processing*, 21(4):1–34.

Maram Hasanain, Firoj Alam, Hamdy Mubarak, Samir Abdaljalil, Wajdi Zaghouani, Preslav Nakov, Giovanni Da San Martino, and Abed Alhakim Freihat. 2023. ArAIEval Shared Task: Persuasion Techniques and Disinformation Detection in Arabic Text. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore. Association for Computational Linguistics.

Hanen Himdi, George Weir, Fatmah Assiri, and Hassanin Al-Barhamtoshy. 2022. Arabic fake news detection based on textual analysis. *Arabian Journal for Science and Engineering*, 47(8):10453–10469.

Khitam Mahmoud Abdalla Jbara, Azzam T Sleit, and Bassam H Hammo. 2009. *Knowledge discovery in Al-Hadith using text classification algorithm*. University of Jordan.

Mohamed Lichouri and Mourad Abbas. 2020a. Simple vs oversampling-based classification methods for fine grained arabic dialect identification in twitter. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 250–256.

Mohamed Lichouri and Mourad Abbas. 2020b. Speechtrans@ smm4h'20: Impact of preprocessing and n-grams on automatic classification of tweets that mention medications. In *Proceedings of the Fifth Social Media Mining for Health Applications Workshop & Shared Task*, pages 118–120.

Mohamed Lichouri, Mourad Abbas, and Besma Benaziz. 2020. Profiling fake news spreaders on twitter based on tfidf features and morphological process.

Mohamed Lichouri, Mourad Abbas, Besma Benaziz, Aicha Zitouni, and Khaled Lounnas. 2021a. Preprocessing solutions for detection of sarcasm and sentiment for Arabic. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 376–380, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Mohamed Lichouri, Mourad Abbas, Abed Alhakim Freihat, and Dhiya El Hak Megtouf. 2018. Word-level vs sentence-level language identification: Application to algerian and arabic dialects. *Procedia Computer Science*, 142:246–253.

Mohamed Lichouri, Mourad Abbas, Khaled Lounnas, Besma Benaziz, and Aicha Zitouni. 2021b. Arabic dialect identification based on a weighted concatenation of TF-IDF features. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 282–286, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Hamdy Mubarak, Samir Abdaljalil, Azza Nassar, and Firoj Alam. 2023. Detecting and identifying the reasons for deleted tweets before they are posted. *Frontiers in Artificial Intelligence*, 6.

El Moatez Billah Nagoudi, AbdelRahim Elmadany, Muhammad Abdul-Mageed, Tariq Alhindi, and Hasan Cavusoglu. 2020. Machine generation and detection of arabic manipulated and fake news. *arXiv preprint arXiv:2011.03092*.

Ahmed Omar, Tarek M Mahmoud, Tarek Abd-El-Hafeez, and Ahmed Mahfouz. 2021. Multi-label arabic text classification in online social networks. *Information Systems*, 100:101785.

Rouhia M Sallam, Hamdy M Mousa, and Mahmoud Hussein. 2016. Improving arabic text categorization using normalization and stemming techniques. *Int. J. Comput. Appl*, 135(2):38–43.

Mohammed Shiha and Serkan Ayvaz. 2017. The effects of emoji in sentiment analysis. *Int. J. Comput. Electr. Eng.(IJCEE.)*, 9(1):360–369.

Kai Shu, Amrita Bhattacharjee, Faisal Alatawi, Tahora H Nazer, Kaize Ding, Mansooreh Karami, and Huan Liu. 2020. Combating disinformation in a social media age. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10(6):e1385.

rematchka at NADI 2023 shared task: Parameter Efficient tuning for Dialect Identification and Dialect Machine Translation

Reem Abdel-Salam
Cairo University, Faculty of Engineering, Computer Engineering / Giza, Egypt
reem.abdelsalam13@gmail.com

Abstract

Dialect identification systems play a significant role in various fields and applications as in speech and language technologies, facilitating language education, supporting sociolinguistic research, preserving linguistic diversity, and enhancing text-to-speech systems. In this paper, we provide our findings and results in the NADI 2023 shared task for country-level dialect identification and machine translation (MT) from dialect to MSA. The proposed models achieved an F1-score of 86.18 at the dialect identification task, securing second place in the first subtask. Whereas for the machine translation task, the submitted model achieved a BLEU score of 11.37 securing fourth and third place in the second and third subtasks. The proposed model utilizes parameter-efficient training methods which achieves better performance when compared to conventional fine-tuning during the experimentation phase.

1 Introduction

Dialect identification plays a crucial role in understanding and analyzing linguistic variation within a language. This importance extends to the Arabic language, which encompasses a wide range of dialects spoken across various regions. With the advancements in natural language processing and language models, dialect identification systems have become increasingly valuable in accurately identifying and distinguishing Arabic dialects. By accurately identifying Arabic dialects, language models contribute to fields such as speech recognition, language learning, and even cultural preservation. However, Dialect identification in the Arabic language presents unique challenges due to the extensive linguistic diversity and complexity of Arabic dialects. Language models, while powerful tools for natural language processing, face inherent difficulties when applied to Arabic dialect identification. These challenges arise from dialectal variations, limited training data, and data scarcity for certain dialects. The NADI shared task series (Abdul-Mageed et al., 2020, 2021b, 2022) is a well-known competition that offers datasets and modeling opportunities in order to improve research work developed for dialect identification. In previous versions of the competitions, various teams have participated. (Messaoudi et al., 2022) fine-tuned MARBERT using two different approaches. The first approach uses model embedding along with a CNN classifier. The other approach is to use model embedding with quasi-recurrent neural networks. (Abdel-Salam, 2022) used is an ensemble between fine-tuned BERT-based models and various approaches of parameter efficient tuning including p-tuning and prompt-tuning. (Bayrak and Issifu, 2022) used general pre-training as a first step followed by fine-tuning. AlKhamissi et al. (2021) added an adapter layer on top of the MARBERT model.

This paper presents our work and findings in the NADI 2023 shared task (Abdul-Mageed et al., 2023). The NADI 2023 shared task consists of three subtasks. The first subtask is a country-level dialect identification, while the second and third subtasks are a sentence-level machine translation from four dialects to MSA, given that a key challenge is the hard nature of the problem. We use best practices from recent research on improving model generalization and robustness by using different parameter-efficient techniques (PEFT). Parameter-efficient fine-tuning (PEFT) is an alternative to full model fine-tuning, where a small number of task-specific parameters are updated and the majority of language model parameters are frozen. In this way, only one general language model alongside the modified parameters for each task is saved or transferred. PEFT techniques include Prefix-tuning (Li and Liang, 2021), LoRa (Hu et al., 2021), Prompt-tuning (Lester et al., 2021) and Soft-prompting (Liu et al., 2023). The rest of the paper is structured as follows: section 3 discusses the proposed methods,

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 652–657
December 7, 2023 ©2023 Association for Computational Linguistics

section 4 shows experimental results, and section 5 concludes the paper.

2 Dataset

Subtask 1 of NADI 2023 (Abdul-Mageed et al., 2023) provides training and development sets with 18 country dialects. The training set constitutes 18K instances and the development set 1.8K instances. In the evaluation phase, the test set provided contains 3.6K instances. For subtask 2, the provided dataset was MADAR-parallel-corpus (Bouamor et al., 2018). The training set consisted of 12000 examples, while validation and test sets consisted of 400 and 2,000 examples.

3 Methodology

This section presents the various approaches used while developing the final models. For subtask 1, the final model is a weighted ensemble of PEFT BERT-based models and fine-tuned models. For subtasks 2&3, a single model is used.

3.1 Subtask 1 models

In subtask 1, the goal was to identify 18 different Arabic dialects. In order to tackle this problem, we have experimented with several approaches. Most of the models used were BERT-based models such as MARBERT (Abdul-Mageed et al., 2021a), AraBERT (Antoun et al.), QARiB (Abdelali et al., 2021), AraELECTRA discriminator (Antoun et al., 2021), and CAMeLBERT (Inoue et al., 2021). Multiple methods were used: 1) fine-tuning, 2) prompt-tuning, 3) prefix-tuning, 4) soft-prompting, 5) few-shot with contrastive learning, 6) adapter based fine-tuning, and 7) pre-training followed by fine-tuning. **In prompt-tuning** only prompts are introduced into the input embedding sequence, which is fed to the language model head and output to the linear classification head. One of the difficulties in prompting is the design of the prompt and the model's output. For the prompt we have used [MASK]اللغة هي (“**language is [MASK]**"), and [MASK]تصنيف اللهجات في التغريده (“ **the dialect in the tweet is [MASK]**") .’and for the output, we have used country names translated into Arabic, as shown in figure 1. **In Soft-prompting** virtual learnable tokens are inserted into the input embedding sequence along with input text, and then this representation is fed to a classifier head, as shown in figure 2. **In prefix-tuning** virtual learnable tokens are inserted into every layer in the model. **In the few shot** settings we have used 100 samples from each class then we have applied supervised contrastive loss along with cross-entropy loss. **For the pre-training followed by fine-tuning**, we first pre-train BERT-based models on the previous year's dataset, and then we fine-tune the model on the newly provided dataset.

Experimental Set-up For the fine-tuned models the learning rate was set to 3e-5 or 4e-6, a cosine-annealing learning rate scheduler was used, the model's weight decay was set to 1e-2 and the length of the sentence for tokenization was set to 256. During training, batch size was set to 8, and at the end of each epoch, the model was evaluated on dev-set. The best-performing model in terms of F1-micro is saved. In all experiments, the first two layers and the embedding were kept frozen.

Submitted systems For this subtask, three different systems were submitted. The first system is a weighted ensemble of all models listed in table 2. For determining the weights of each, we used an optimization method, where the goal is to find the best set of weights that minimize log-loss between the weighted prediction of all models and the true labels of the dev-set. For the second and the third system, we have chosen the best combination of models that yields a high F1-score in the dev-set, through an exhaustive search, as well as optimization to determine the best set of ensemble weights. The experiment goes as follows: we first generate each possible combination of the developed models. Then for each combination, we apply an optimization scheme to determine the best set of weights for each model based on the F1-score calculated between the weighted prediction and actual labels of the dev-set. Finally, we choose the best combination that yields the best F1 score. The models for the second system were: MARBERT with adapter layer, MARBERT with prefix tuning, CAMeLBERT, and QARiB. The models for the third system were: MARBERT with prompt-tuning, MARBERT with soft prompting, MARBERT with prefix-tuning, and MARBERT with pre-training and then fine-tuning.

3.2 Subtask 2&3 models

In this subtask, the goal is to translate a dialectal sentence into MSA. To tackle this problem we have experimented with several approaches in the development phase (dev-phase). The model used

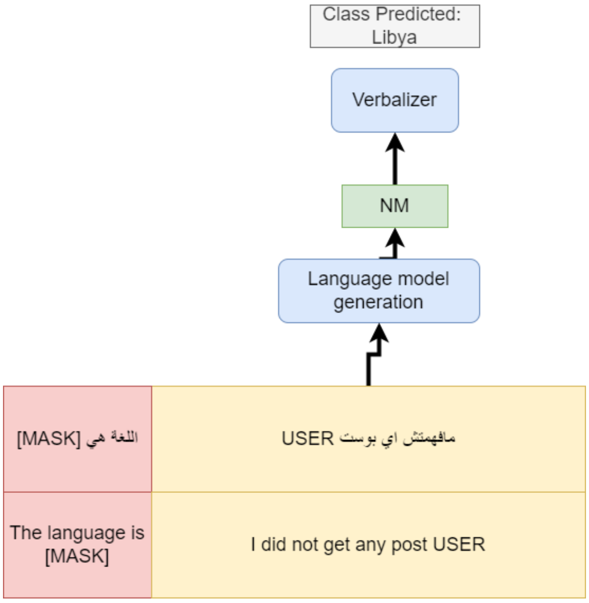


Figure 1: Prompt-tuning architecture.

was AraT5v2 (Nagoudi et al., 2022). Several methods have been investigated as 1) conventional-fine-tuning, 2) LoRa, and 3) prompt-tuning. **In LoRa** instead of fine-tuning all the weights that constitute the weight matrix of the pre-trained large language model, two smaller matrices that approximate this larger matrix are fine-tuned. These matrices constitute the LoRA adapter. This fine-tuned adapter is then loaded to the pre-trained model and used for inference. **In prompt-tuning** the following prompt was added before each text to be translated أعد صياغه الجمله للعربيه الفصحى . **("Rephrase the following to modern standard Arabic")** another prompt investigated was text followed by **source dialect => target dialect, example: CAI => MSA.**.

Experimental Set-up In all of the configurations the encoder and decoder embedding were frozen. The learning rate was set to 6e-6, with a model weight decay of 1e-2. Linear learning rate scheduler was used and the length of the sentence for tokenization was set to 256. Models were fine-tuned for 10 epochs with a batch size of 2. The best-performing model in terms of BLEU score is saved. For LoRa, the following parameters were used: the scaling factor was set to 4, while the rank: was set to 1.

Submitted systems In these subtasks, only one submission was made based on the conventional-fine-tuning method.

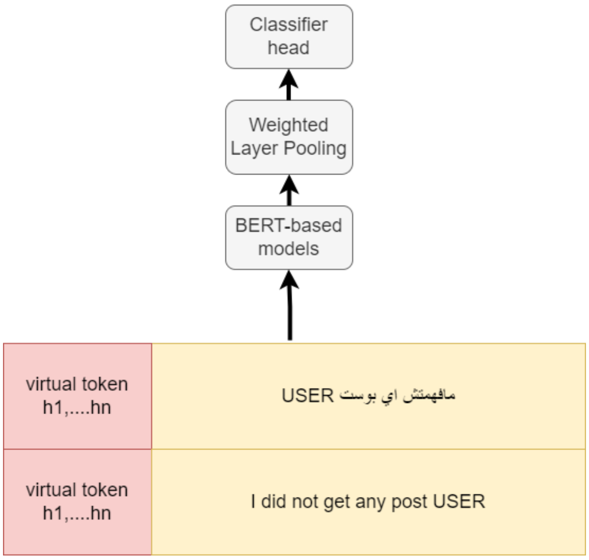


Figure 2: Soft-prompting architecture.

4 Results and Discussion

In this section, the performance of the models is reported based on the official metric during dev-phase and test-phase. Moreover, error analysis is conducted to identify weaknesses of the proposed models. For subtask 1 the official metric is the micro average F1-score, while for subtask 2&3 the official metric is the BLEU score.

4.1 Dev-phase results

Table 2 shows results on dev-set for subtask 1. It can be concluded that prompt-based model performed better than fine-tuning methods, prefix-tuning, and soft prompting. The margin difference is around 1%. Table 3 shows submission scores based on the F1-score on the dev-set. All model ensemble underperforms when compared to selective model ensemble. On the other hand, it takes a lot of time to search all possible combinations to select the best one. During experimentation, the model performance decreased while using a combined dataset of the previous year's dataset and the current year's dataset, compared to using only this year's dataset. Our key findings were: PEFT techniques outperform conventional fine-tuning by a magnitude of a maximum of 8% and a minimum of 3%. Prompt-based models were the best-performing models in PEFT, however, they are sensitive to the prompt used. For instance, the results when using the prompt [MASK]اللغة هي ("**language is [MASK]**"), outperform the results from [MASK]تصنيف اللهجات في التغريده (" **the**

dialect in the tweet is [MASK]") by a magnitude of 1%. Table 1, shows the BLEU score achieved using different techniques for subtask 2&3. LoRa shows significant performance compared to other techniques such as prompting and conventional fine-tuning, with a margin of 3%. This might be due to the fact that the prompt needs more engineering and the hyperparameters re-adjustment. For instance, to our surprise, the second prompt achieved better performance than the first prompt, described in section 3.2. During experimentation, the model showed high sensitivity to the learning rate and weight decay. For instance, we have conducted 3 runs for each experimentation. In the setup, all configurations were kept the same except for the learning rate. The learning rate was set to 1e-6, 3e-6, 6e-6. There were high variation in the results by a magnitude of 2%. For the experiment with a learning rate of 1e-6 the BLEU score was around 8, for a learning rate of 3e-6 the score was around 9, and for a learning rate 6e-6, the score increased to 11.

Model	Technique	BLEU score
AraT5	Conventional	11.136
	LoRA	11.04
	Prompting with prompt Rephrase the following to modern standard Arabic	8.54
	Prompting with prompt source dialect =>target dialect	13.503

Table 1: Models and techniques developed during the experimental phase for subtask 2&3.

4.2 Test-phase results

Table 4 and 5 show the performance of the submitted model in the test-phase for all subtasks. For subtask 1, most models had a near performance with a 0.1 present error, unlike in the dev-set. However, top-performing systems in dev-phase are not the same during the test-phase. For instance, submission-2 and submission-1 interleaved places. Although there is a margin difference of 0.02 in the dev-phase, this changes to 0.001 in the test-phase.

4.3 Error Analysis

Further investigations have been carried out to analyze the potential limitations of the system. As seen in Figure 3, our model performs well when predicting most dialects. However, the model confuses between Kuwait and Bahrain, as well as Syrian and Lebanese dialects. . We believe this is due to the geographic natures between those dialects, as these countries are geographically near each other. Thus it is hard to distinguish between them. For subtask 2&3 one of the major problems was slow convergence of the model in the translation task and fast overfitting.

Model	Technique	F1-Score
MARBERT	Prefix-tuning	0.859
	Adapter	0.755
	Soft-Prompt	0.857
	Prompt-tuning	0.83
	pre-training then fine-tuning	0.828
AraBERT v2	Prompt-tuning	0.857
CAMeLBERT	Prefix-tuning	0.76
QARiB	Fine-tuning	0.77
AraELECTRA	Fine-tuning	0.77

Table 2: Models and techniques developed during the experimental phase for subtask 1.

5 Conclusion

We presented our attempts for the NADI shared task in this article. Our solution is an ensemble of many BERT-based models. These models are created in a variety of ways, including prefix-based models, fine-tuned models, and prompt-based models. The findings reveal that our suggested models perform well in the three subtasks, taking second place in subtask 1 and fourth and third places in subtask 2&3. Future work will concentrate on developing a robust model to improve dialect recognition. Furthermore, to research and identify traits that better distinguish dialects.

References

Reem Abdel-Salam. 2022. Dialect & sentiment identification in nuanced Arabic tweets using an ensemble of prompt-based, fine-tuned, and multitask BERT-based models. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*,

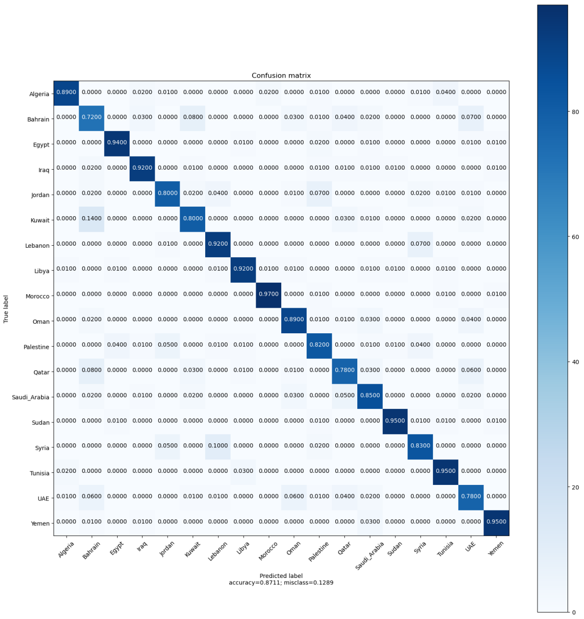


Figure 3: Confusion matrix of the predictions of the submission-3 model in subtask 1 on the dev-set.

Submission	F1-Score
Submission-1	0.859
Submission-2	0.876
Submission-3	0.880

Table 3: Performance of the submitted models on the dev-set in subtask 1. Submission-1 is a weighted ensemble of all models developed, submission-2 is a weighted ensemble of MARBERT with adapter layer, MARBERT with prefix tuning, CAMeLBERT, and QARiB, while submission-3 is a weighted ensemble of MARBERT with prompt-tuning, MARBERT with soft prompting, MARBERT with prefix-tuning, and MARBERT with pre-training then fine-tuning.

Submission Number	F1-Score
Submission-1	0.855
Submission-2	0.853
Submission-3	0.861

Table 4: Performance of the submitted models on the leaderboard in subtask 1. Submission-1 is a weighted ensemble of all models developed, submission-2 is a weighted ensemble of MARBERT with adapter layer, MARBERT with prefix tuning, CAMeLBERT, and QARiB, while submission-3 is a weighted ensemble of MARBERT with prompt-tuning, MARBERT with soft prompting, MARBERT with prefix-tuning, and MARBERT with pre-training then fine-tuning.

Task	BLEU score
Subtask 2	11.37
Subtask 3	11.37

Table 5: Performance of the submitted models on the leaderboard in subtask 2&3

pages 452–457, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

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

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2021a. 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.

Muhammad Abdul-Mageed, AbdelRahim Elmadany, Chiyu Zhang, ElMoatez Billah Nagoudi, Houda Bouamor, and Nizar Habash. 2023. NADI 2023: The Fourth Nuanced Arabic Dialect Identification Shared Task. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*.

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, Barcelona, Spain (Online). Association for Computational Linguistics.

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, Houda Bouamor, and Nizar Habash. 2021b. NADI 2021: The second nuanced Arabic dialect identification shared task. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 244–259, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, Houda Bouamor, and Nizar Habash. 2022. NADI 2022: The third nuanced Arabic dialect identification shared task. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 85–97, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Badr AlKhamissi, Mohamed Gabr, Muhammad El-Nokrashy, and Khaled Essam. 2021. Adapting marbert for improved arabic dialect identification: Submission to the nadi 2021 shared task. *arXiv preprint arXiv:2103.01065*.

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

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

Giyaseddin Bayrak and Abdul Majeed Issifu. 2022. Domain-adapted BERT-based models for nuanced Arabic dialect identification and tweet sentiment analysis. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 425–430, Abu Dhabi, United Arab Emirates (Hybrid). 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 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).

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*.

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.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2023. Gpt understands, too. *AI Open*.

Abir Messaoudi, Chayma Fourati, Hatem Haddad, and Moez BenHajhmida. 2022. iCompass working notes for the nuanced Arabic dialect identification shared task. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 415–419, Abu Dhabi, United Arab Emirates (Hybrid). 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.

UniManc at NADI 2023 Shared Task: A Comparison of Various T5-based Models for Translating Arabic Dialectal Text to Modern Standard Arabic

Abdullah Khered[1,2] , **Ingy Yasser Abdelhalim** [1], **Nadine Abdelhalim**[1],
Ahmed Soliman [1] and **Riza Batista-Navarro**[1]
[1]The University of Manchester, UK [2]King Abdulaziz University, Saudi Arabia
abdullah.khered@manchester.ac.uk, ingyyh@live.com, nadineyh@live.com,
ahmedsoliman360@gmail.com, riza.batista@manchester.ac.uk

Abstract

This paper presents the methods we developed for the Nuanced Arabic Dialect Identification (NADI) 2023 shared task, specifically targeting the two subtasks focussed on sentence-level machine translation (MT) of text written in any of four Arabic dialects (Egyptian, Emirati, Jordanian and Palestinian) to Modern Standard Arabic (MSA). Our team, UniManc, employed models based on T5: multilingual T5 (mT5), multi-task fine-tuned mT5 (mT0) and AraT5. These models were trained based on two configurations: joint model training for all regional dialects (J-R) and independent model training for every regional dialect (I-R). Based on the results of the official NADI 2023 evaluation, our I-R AraT5 model obtained an overall BLEU score of 14.76, ranking first in the Closed Dialect-to-MSA MT subtask. Moreover, in the Open Dialect-to-MSA MT subtask, our J-R AraT5 model also ranked first, obtaining an overall BLEU score of 21.10.

1 Introduction

The Arabic language serves as a linguistic umbrella for approximately 420 million speakers, predominantly dispersed across 22 countries in the Middle East and North Africa (MENA) region. A defining characteristic of the language is its diglossic nature, where Modern Standard Arabic (MSA) coexists with a myriad of dialects, commonly referred to as Dialectal Arabic (DA) (Al-Sobh et al., 2015; Abdul-Mageed et al., 2022).

MSA is the formal version of the Arabic language, employed in educational settings, official documents and written literature. It serves as a standardised communication medium across Arabic-speaking countries. In contrast, DA represents the colloquial forms of Arabic, which are more region-specific and employed in day-to-day verbal interactions (Shoufan and Alameri, 2015). Notably, dialects can vary significantly based on geographic location and socio-economic factors, ranging from subtle differences to being nearly mutually unintelligible. This linguistic variation presents considerable challenges for machine translation (MT) models trained on MSA. These models often fail to capture the nuanced differences in dialects, resulting in poor translation performance when applied to DA. Compounding this issue is the scarcity of parallel corpora containing MSA translations of text written in DA, limiting resources for model training and evaluation (Harrat et al., 2019).

In the context of these challenges, this paper aims to explore the extent to which various sequence-to-sequence models based on the Text-to-Text Transfer Transformer, popularly known as T5 (Raffel et al., 2020), can translate a source text written in an Arabic dialect to a target text that is written in MSA. We participated in the Nuanced Arabic Dialect Identification (NADI) 2023 Shared Task (Abdul-Mageed et al., 2023), specifically in Subtasks 2 and 3, described below.

Subtask 2: Dialect-to-MSA MT - Closed Task. The objective of this subtask is sentence-level machine translation from four dialects (Egyptian, Emirati, Jordanian and Palestinian) to MSA. Participants were restricted to using the MADAR parallel corpus (Bouamor et al., 2019) for training and were asked to evaluate their models on newly released development and test sets.

Subtask 3: Dialect-to-MSA MT - Open Task. This subtask is similar to Subtask 2, except for the fact that participants were allowed to utilise additional datasets for model training. One of the goals of this subtask is to encourage the creation of new parallel corpora to facilitate future research.

Apart from investigating the performance of various T5-based models on the above-mentioned tasks, our work makes an additional contribution by developing a new dataset, Emi-NADI, which contains MSA translations of sentences written in Emirati, one of the most under-resourced dialects.

The Emi-NADI dataset and the code for developing and evaluating our models for Subtasks 2 and 3 have been made publicly available.[1]

2 Datasets

This section describes the datasets that were utilised in training our models.

2.1 The MADAR Corpus

As mentioned in the previous section, participants in the closed version of the dialect-to-MSA translation task, Subtask 2, were allowed to use only the MADAR parallel corpus (Bouamor et al., 2019), which covers the dialects used in 25 Arabic-speaking cities, as well as English and MSA.

2.2 Additional Corpora

In the open version of the dialect-to-MSA machine translation task, Subtask 3, participants were allowed to leverage any dataset. As we searched for potentially useful publicly available datasets, we considered those that cover various Arabic dialects, including regional ones that are relevant to the four countries of interest in NADI. For example, the Gulf dialect is relevant to Emirati (since the United Arab Emirates is one of the Gulf countries), and the Levantine dialect is relevant to Jordanian and Palestinian (since Jordan and Palestine belong to the Levant). Apart from the MADAR corpus, we identified and made use of four datasets: (1) PADIC, (2) Dial2MSA, (3) a semantic textual similarity (STS) dataset for Arabic dialects, and (4) our own Emi-NADI dataset containing Emirati-to-MSA translations. Table 1 provides information on the size of each dataset in terms of number of dialectal sentences with translations to MSA.

Dataset	Egy.	Gulf	Lev.
MADAR	13,800	15,400	18,600
PADIC	0	0	12,824
Dial2MSA	16,355	0	0
Arabic STS	2,758	2,758	0
Emi-NADI	0	2,712	0
Total	32,913	20,870	31,424

Table 1: The number of dialect-to-MSA translation pairs in each of the datasets used in Subtask 3.

PADIC (Meftouh et al., 2018) is a parallel corpus containing dialectal Arabic texts covering six Arab cities including Gaza and Damascus, which are both in the Levant region. Meanwhile, the Dial2MSA dataset (Mubarak, 2018) consists of tweets written in four Arabic dialects (Egyptian, Gulf, Levantine, Maghrebi) and their corresponding MSA translations. As only the translations for Egyptian and Maghrebi were manually validated, we made use of the Egyptian-to-MSA translations only. In the work of Al Sulaiman et al. (2022) that focussed on Arabic STS (i.e., determining the semantic similarity between two given sentences), they manually produced MSA, Egyptian and Saudi dialect translations for 2758 English sentences, which we also utilised in our work.

Our own dataset, Emi-NADI, was created to address the scarcity of parallel corpora covering the Emirati dialect, and contains MSA translations of the Emirati tweets in the training datasets provided as part of NADI Subtask 1 (country-level dialect identification) (Abdul-Mageed et al., 2020, 2021, 2023). The translations were generated by a large language model (LLM), specifically the GPT 3.5 Turbo model,[2] resulting in a total of 2712 translations. A subset of 1000 automatically generated translations were manually validated (by native Arabic speakers who understand Emirati) to ensure quality. Both the validated and the non-validated samples were used in model training.

3 Methodology

In this section, we introduce the T5-based models that we built upon, explain how they were fine-tuned and discuss hyperparameter optimisation.

3.1 Models

T5 casts different natural language processing (NLP) tasks into a standard text-to-text format. One of the NLP tasks that T5 was already trained on is machine translation (Raffel et al., 2020). In this work, we fine-tuned three types of T5 models, namely, AraT5, mT5 and mT0.

AraT5. AraT5 (Nagoudi et al., 2022) is based on the same architectural foundation as the original T5 models, but trained specifically on Arabic data encompassing both MSA and dialectal Arabic (tweets). The most recent version of AraT5, AraT5$_{\text{v2}}$,[3] was used in all our experiments.

Multilingual T5 (mT5). mT5 (Xue et al., 2021) is a multilingual variant of T5 that underwent pre-

[1] https://github.com/khered20/UniManc_NADI2023_ArabicDialectToMSA_MT

[2] https://platform.openai.com/docs/models/gpt-3-5

[3] https://huggingface.co/UBC-NLP/AraT5v2-base-1024

Source	Target
Original Pair	
الجرح بتاعي بيألم	جرحي يؤلمني
Additional Pair	
جرحي يؤلمني	جرحي يؤلمني

Table 2: An example of the additional training pair where each of the source and target is the text written in MSA (English translation: *"My wound hurts"*). The tokens shown in grey in the Egyptian source text of the original pair share the same root as the tokens in grey in the target MSA text.

training using a novel dataset sourced from Common Crawl, encompassing 101 languages. Although its pre-training process is underpinned by the original T5 architecture, it incorporated some improvements, such as the adoption of a different activation function in the feed-forward layer (i.e., GeGLU instead of the conventional RELU).

Multi-task fine-tuned mT5 (mT0). Multitask-prompted fine-tuning (MTF) has demonstrated its efficacy in assisting LLMs in adapting to novel tasks within a zero-shot setting. In this vein, mT0 is a multitask-prompted fine-tuned version of mT5. mT0 has showcased remarkable zero-shot generalisation capabilities, even when presented with languages it has never encountered before (Muennighoff et al., 2023).

3.2 Training Configurations

In the early stages of our experimentation, we noticed that many dialectal texts contain words that are shared between a dialect and MSA. Thus, for every translation pair in our training data, we generated an additional pair where each of the source and target is the text written in MSA. An example is provided in Table 2. Our models were then trained — based on the two different configurations outlined below — using these additional pairs, enabling them to learn how to handle sentences that include words that are also used in MSA.

Training a joint model for all regional dialects (J-R). In this configuration, all dialect-to-MSA translation pairs (in the training sets for Subtasks 2 and 3) that correspond to the regions relevant to the four dialects of interest were utilised in training one model. Therefore, translation pairs from datasets that cover the Egyptian, Gulf and Levantine dialects were utilised in model training. The result is one joint model trained to translate dialectal text to MSA, regardless of which dialect it was written in.

Training an independent model for each regional dialect (I-R). In this configuration, one model was trained for every relevant regional dialect. This resulted in four separate models, where each model was independently trained to translate texts written in one specific dialect only, to MSA.

3.3 Hyperparameter Optimisation

For each of the two subtasks, we trained our models using two Nvidia A100 GPUs based on the configurations described above. All models accept input sequences with a maximum length of 128 tokens and generate output text also with a maximum length of 128 tokens. Learning rate and batch size were fixed at 5e-5 and 16, respectively. The maximum number of epochs was set to 40, although we always selected the model produced in the epoch that yielded the best performance on the development (dev) set provided by the NADI organisers. Importantly, we investigated whether incorporating beam search (Freitag and Al-Onaizan, 2017) during translation leads to improved performance, experimenting with different beam sizes ranging from 1 to 5.

4 Evaluation and Results

All models for Subtasks 2 and 3 were evaluated using the BiLingual Evaluation Understudy (BLEU) metric (Papineni et al., 2002), which estimates the similarity between a machine-translated text and a reference translation based on overlapping tokens.

The results of our joint regional (J-R) and independent regional (I-R) models for Subtasks 2 and 3, without using beam search (i.e., beam size = 1), are shown in Tables 3 and 4, respectively. One can observe in Table 3 that for Subtask 2, in all cases (except for Jordanian), the I-R version of a model consistently outperforms its J-R counterpart. This finding led us to further experiment with the I-R models by investigating different values for beam size. The results, shown in Table 7 in the Appendix, helped us in identifying the best-performing I-R models. Based on this, we selected two I-R AraT5 models, one I-R mT5 model and one I-R mT0 model to comprise our set of models for the official evaluation (on the NADI test set), together with the best J-R model.

Model	Egy.	Emi.	Jor.	Pal.	Overall
Joint Regional Models (J-R)					
mT0	12.28	**10.98**	10.06	9.65	11.12
mT5	12.16	10.93	9.14	9.49	11.13
AraT5$_{v2}$	**14.65**	10.65	**11.20**	**10.53**	**13.30**
Independent Regional Models (I-R)					
mT0	13.88	12.91	9.55	10.91	12.53
mT5	15.02	**15.25**	10.32	10.69	13.57
AraT5$_{v2}$	**17.21**	14.13	**12.14**	**13.33**	**15.14**

Table 3: Comparison of joint regional (J-R) and independent regional (I-R) models for Subtask 2, based on the development set. Beam size = 1.

Model	Egy.	Emi.	Jor.	Pal.	Overall
Joint Regional Models (J-R)					
mT0	15.11	26.85	18.44	15.16	18.25
mT5	18.80	29.04	18.63	15.50	19.81
AraT5$_{v2}$	**20.23**	**32.84**	**24.85**	**18.27**	**23.37**
Independent Regional Models (I-R)					
mT0	18.28	27.35	19.82	16.46	19.96
mT5	18.26	26.83	21.45	16.48	20.25
AraT5$_{v2}$	**21.90**	**31.28**	**24.45**	**18.08**	**23.45**

Table 4: Comparison of joint regional (J-R) and independent regional (I-R) models for Subtask 3, based on the development set. Beam size = 1.

In the comparison of the J-R and I-R models (without beam search) for Subtask 3 shown in Table 4, it is evident that the AraT5 models outperform both mT0 and mT5 by a noticeable margin, and that the I-R models outperform their J-R counterparts overall. We thus further experimented with the I-R versions of the AraT5 model by investigating different beam sizes. The results, shown in Table 8 in the Appendix, informed our selection of models for the official evaluation (on the NADI test set), which consists of the three best I-R AraT5 models, one I-R mT5 model and the best J-R model.

Tables 5 and 6 present the results of our chosen models on the NADI test sets for Subtasks 2 and 3, respectively. As shown in Table 5, the I-R AraT5 model with beam size = 3 outperformed our other models (obtaining a score of 14.76). Meanwhile, our Subtask 3 results, shown in Table 6, demonstrate that the J-R AraT5 model (with beam size = 1) performs best overall (21.10). To investigate whether adjusting the beam size of the J-R AraT5 model will lead to even better performance, we submitted the same model to the post-evaluation phase of Subtask 3, but this time with beam size = 5. The overall score did increase to 21.87, implying once again that incorporating beam search leads to better performance.

5 Discussion

In Tables 3 and 4, it can be observed that for both subtasks the independent regional (I-R) models performed better compared to the joint regional (J-R) models, with AraT5 performing the best overall. This can be explained by the fact that AraT5 was trained with a specific focus on Arabic whereas the others (mT0 and mT5) were trained on many other languages apart from Arabic. This implies that for the dialect-to-MSA translation task, a model that was trained solely on the Arabic language is superior over multilingual models.

Given that the I-R models performed better, multiple beam sizes were explored. Our results show that increasing the beam size leads to an improvement in overall performance. However, it is worth noting that the optimal beam size could vary between the development and test sets (e.g., beam size = 4 on the development set and beam size = 3 on the test set for Subtask 2), although the difference in performance is very marginal.

Error analysis was conducted to qualitatively evaluate our best-performing model for Subtask 3. Specifically, we analysed cases where the model obtained low BLEU scores and manually assessed the quality of the translations produced by the model. An example for each dialect is shown in Table 10 in the Appendix. Interestingly, the model's translations of the Egyptian, Emirati and Jordanian source texts are arguably correct, as they convey the same meaning as the reference translations. They, however, obtained low BLEU scores due to the fact that the BLEU metric takes into account lexical but not semantic similarity, in comparing a generated translation with a reference one. As for the Palestinian example, the model's failed translation can be attributed to code-mixing, i.e., the presence of the non-Arabic word "bravo" (written in Arabic script) in the source text.

6 Conclusion and Future Work

In this paper, we describe the approaches we developed for NADI 2023 Subtask 2 (Closed Dialect-to-MSA MT) and Subtask 3 (Open Dialect-to-MSA MT). Our results reveal that fine-tuning AraT5 and incorporating beam search during translation lead to top-ranking performance. Possible future directions include the development of a multilingual model focussed on Arabic dialects and MSA, and the creation of further parallel corpora covering low-resourced Arabic dialects.

Model	Configuration	Beam	Egy.	Emi.	Jor.	Pal.	Overall
AraT5$_{v2}$	J-R	1	12.50	10.15	11.39	10.28	12.12
mT0	I-R	3	13.64	12.43	7.67	9.32	11.37
mT5	I-R	2	14.04	10.42	10.65	11.66	12.38
AraT5$_{v2}$	I-R	3	16.04	**14.30**	12.55	**13.55**	**14.76**
AraT5$_{v2}$	I-R	4	**16.54**	14.20	**12.73**	13.04	14.73

Table 5: Results of evaluating our submitted models on the NADI Subtask 2 test set.

Model	Configuration	Beam	Egy.	Emi.	Jor.	Pal.	Overall
AraT5$_{v2}$	J-R	1	17.65	**28.46**	**22.03**	17.29	**21.10**
mT5	I-R	1	15.75	25.15	16.44	16.15	17.95
AraT5$_{v2}$	I-R	1	17.95	24.94	20.84	17.67	20.22
AraT5$_{v2}$	I-R	3	19.61	25.79	20.95	**18.31**	21.02
AraT5$_{v2}$	I-R	4	**19.70**	26.02	21.00	18.27	21.08

Table 6: Results of evaluating our submitted models on the NADI Subtask 3 test set.

Limitations

Due to time and computational resource constraints, we were unable to conduct a more systematic investigation of the effect of different beam size values for the joint regional AraT5, mT5 and mT0 models that we employed.

Furthermore, most of the models that we submitted to the official NADI 2023 Subtasks 2 and 3 evaluation were trained following a configuration whereby a separate model was independently trained on every dialect. This means that prior to translation, the dialect in which an input text was written in needs to be predetermined, so that the relevant model can be applied.

References

Muhammad Abdul-Mageed, AbdelRahim Elmadany, Chiyu Zhang, ElMoatez Billah Nagoudi, Houda Bouamor, and Nizar Habash. 2023. NADI 2023: The Fourth Nuanced Arabic Dialect Identification Shared Task. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*.

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, Barcelona, Spain (Online). Association for Computational Linguistics.

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, Houda Bouamor, and Nizar Habash. 2021. NADI 2021: The second nuanced Arabic dialect identification shared task. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 244–259, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, Houda Bouamor, and Nizar Habash. 2022. NADI 2022: The third nuanced Arabic dialect identification shared task. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 85–97, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Mahmoud Al-Sobh, Abdel-Rahman Abu-Melhim, and Nedal Bani Hani. 2015. Diglossia as a result of language variation in arabic: Possible solutions in light of language planning. *Journal of Language Teaching and Research*, 6:274.

Mansour Al Sulaiman, Abdullah M. Moussa, Sherif Abdou, Hebah Elgibreen, Mohammed Faisal, and Mohsen Rashwan. 2022. Semantic textual similarity for modern standard and dialectal arabic using transfer learning. *PLOS ONE*, 17(8):1–14.

Houda Bouamor, Sabit Hassan, and Nizar Habash. 2019. The MADAR shared task on Arabic fine-grained dialect identification. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 199–207, Florence, Italy. Association for Computational Linguistics.

Markus Freitag and Yaser Al-Onaizan. 2017. Beam search strategies for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 56–60, Vancouver. Association for Computational Linguistics.

Salima Harrat, Karima Meftouh, and Kamel Smaili. 2019. Machine translation for arabic dialects (survey). *Information Processing Management*, 56(2):262–273. Advance Arabic Natural Language Processing (ANLP) and its Applications.

Karima Meftouh, Salima Harrat, and Kamel Smaïli. 2018. PADIC: extension and new experiments. In *7th International Conference on Advanced Technologies ICAT*.

Hamdy Mubarak. 2018. Dial2MSA: A Tweets Corpus for Converting Dialectal Arabic to Modern Standard Arabic. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), OSACT2018 Workshop*, pages 49–53.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, and Teven et al. Le Scao. 2023. Crosslingual generalization through multitask finetuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15991–16111, Toronto, Canada. 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.

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 ACL*, pages 311–318, Philadelphia, Pennsylvania, USA. 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.

Abdulhadi Shoufan and Sumaya Alameri. 2015. Natural language processing for dialectical Arabic: A survey. In *Proceedings of the Second Workshop on Arabic Natural Language Processing*, pages 36–48, Beijing, China. Association for Computational Linguistics.

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, Online. Association for Computational Linguistics.

Appendix

Model	Beam size	Egy.	Emi.	Jor.	Pal.	Overall
AraT5$_{v2}$	1	17.209	14.127	12.143	13.329	15.139
	2	17.152	15.197	12.906	14.458	15.828
	3	18.702	14.798	12.892	14.507	**16.166**
	4	19.092	15.281	12.478	14.552	**16.173**
	5	19.274	15.052	12.213	14.267	16.037
mT5	1	15.023	15.253	10.324	10.689	13.57
	2	15.755	14.888	11.924	10.345	**13.919**
	3	16.121	14.903	11.395	9.962	13.757
	4	16.076	14.857	11.754	10.015	13.873
	5	16.071	14.744	11.519	10.205	13.909
mT0	1	13.882	12.914	9.551	10.907	12.525
	2	13.199	12.371	10.398	10.884	12.389
	3	14.498	12.336	11.198	11.692	**13.222**
	4	14.432	12.69	10.643	11.633	13.085
	5	14.439	12.554	10.283	11.429	12.989

Table 7: Results of using different values for beam size on the Subtask 2 development set, obtained by independent regional (I-R) models. The best performing models (whose overall scores are shown in bold) formed the basis of our submission to the official NADI 2023 evaluation.

Model	Beam size	Egy.	Emi.	Jor.	Pal.	Overall
AraT5$_{v2}$	1	21.897	31.28	24.451	18.077	**23.452**
	2	22.153	30.65	23.922	17.628	23.151
	3	22.788	31.123	24.035	17.525	**23.447**
	4	22.59	31.174	23.763	17.559	**23.334**
	5	22.401	31.291	23.915	17.56	23.319

Table 8: Results of using different values for beam size on the Subtask 3 development set, obtained by independent regional (I-R) versions of the AraT5 model. The best performing models (whose overall scores are shown in bold) formed the basis of our submission to the official NADI 2023 evaluation.

Dialect	Dialectal Text	English Translation
Egy	لكن التأمين أنا مبفهمش فيها خالص.	But the insurance I don't understand it at all
Emi	إنزين و اللولو جم يسوى؟	Then how much is this pearl worth?
Jor	يسلموا ايديكي عالفطور الطيب	Thank you for the delicious breakfast
Pal	برافو يا شفيق ترابيتي	Bravo Shafiq, my upbringing

Table 9: English translations of the dialectal Arabic examples shown in Table 10, provided for the reader's convenience.

Dialect	Dialectal Text	Reference MSA Translation	Model Translation
Egy	لكن التأمين أنا مبفهمش فيها خالص.	لكن التأمين ، أنا لا أفهم فيه أبداً.	لكن التأمين لا أفهم فيه على الإطلاق.
Emi	إنزين و اللولو جم يسوى؟	حسنًا و اللؤلؤ كم يساوى؟	ما قيمة هذا اللولو ؟
Jor	يسلموا ايديكي عالفطور الطيب	سلمت يداكِ على هذا الفطور اللذيذ	شكراً جزيلاً على الإفطار الطيب.
Pal	برافو يا شفيق ترابيتي	أحسنت يا شفيق، تعليمي	برافو يا شفيق تربيتي

Table 10: Examples showing cases where the translation generated by our best-performing Subtask 3 model was given a low BLEU score despite being semantically correct. For English translations of the dialectal examples, we refer the reader to Table 9.

IUNADI at NADI 2023 shared task: Country-level Arabic Dialect Classification in Tweets for the Shared Task NADI 2023

Yash A. Hatekar
Indiana University Bloomington
yhatekar@iu.edu

Muhammad S. Abdo
Indiana University Bloomington
mabdo@iu.edu

Abstract

In this paper, we describe our participation in the NADI2023 shared task for the classification of Arabic dialects in tweets. For training, evaluation, and testing purposes, a primary dataset comprising tweets from 18 Arab countries is provided, along with three older datasets. The main objective is to develop a model capable of classifying tweets from these 18 countries. We outline our approach, which leverages various machine learning models. Our experiments demonstrate that large language models, particularly Arabertv2-Large, Arabertv2-Base, and CAMeLBERT-Mix DID MADAR, consistently outperform traditional methods such as SVM, XGBOOST, Multinomial Naive Bayes, AdaBoost, and Random Forests.

1 Introduction

Officially Spoken in more than 20 countries, and in a myriad of regional variations, the Arabic language has consistently piqued the curiosity of researchers across various disciplines. This is because of Arabic's historical significance and pivotal role in shaping the cultural, religious, social, and political fabric of the Arab world. Historically, Arabic has often been typologically classified into three distinct categories: Classical Arabic, Modern Standard Arabic (MSA), and Dialectal Arabic (DA). Classical Arabic refers to the language used in the Holy Qur'an and pre-Islamic poetry, while MSA pertains to the language of newspapers, literature, education, official documents, and formal media and news broadcasts. DA, which is the primary focus of this paper, is more concerned with the language used in daily communication by speakers of Arabic. These dialects are often classified into: Egyptian, Levantine, Gulf, and Maghrebi Arabic. Within each of these distinct communities, an array of subdialects can be found in different geographical regions (Diab et al., 2010; Zaidan and Callison-Burch, 2014; Jarrar et al., 2017).

For the most part, MSA was the predominant variation used in written Arabic. But the advent of online forums and social media platforms, such as Twitter, gave the variations of DA a space to grow their written content presence. These dialects differ phonologically, morphologically, syntactically, and semantically. Yet, it is noteworthy to mention that there can still be some degree of overlap between DA and MSA. This is due to the fact that Arabic is a root-based language, which means that many words share common roots consisting of three or four letters. This unprecedented massive increase in digital content in DA has propelled the development of NLP tools that can read, manipulate, and potentially generate this content. While developing such tools to handle text in MSA has posed many challenges, this task has been even more arduous to do for text written in DA, e.g., tweets. Arab users of Twitter mainly use no standardized orthographic variation (e.g.,الأهلي, الاهلي, الأهلى), emphasize their thoughts or sentiments through elongation by excessively repeating certain letters (e.g., مستحيييل, ليبييييش), miss or add extra spaces between words (e.g.,ما يصير, الحمدلله), vary their word choice to the same referent (e.g., عايز, عاوز, أبغي, أريد), to name but a few observations. All these issues present many challenges for developing a single system that can accurately classify all Arabic dialects (Darwish et al., 2014; Jarrar et al., 2014; Lulu and Elnagar, 2018).

In this article, we outline our system, which we entered in Task 1 of the NADI2023 shared task focusing on Arabic dialects classification (Abdul-Mageed et al., 2023). As with the three preceding NADI shared tasks (Abdul-Mageed et al., 2020), (Abdul-Mageed et al., 2021), and (Abdul-Mageed et al., 2022), the primary objective of this task is to develop models capable of categorizing tweets originating from 18 distinct Arab countries.

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 665–669
December 7, 2023 ©2023 Association for Computational Linguistics

2 Methodology

2.1 Data

For the purposes of this task, a Twitter dataset of 23.4K tweets, covering 18 different dialects from 18 countries, is provided. This dataset is divided into 3 smaller sets: 18K tweets for training, 1.8K tweets for development, and 3.6K tweets for testing. Additionally, datasets from the previous two NADI tasks (Abdul-Mageed et al., 2020, 2021), and MADAR (Bouamor et al., 2018) were provided. Participants in this task were not allowed to use any other datasets.

2.2 Data Pre-Processing

In the pre-processing phase of our research, we implemented a series of essential steps to prepare the datasets for model training and evaluation. These steps aimed to enhance the quality and consistency of the data, ensuring optimal model performance. To accomplish this, we followed the data pre-processing methods outlined in previous studies (Badaro et al., 2018; Muaad et al., 2022). These pre-processing procedures collectively served to optimize the datasets for subsequent training and evaluation of our models. The pre-processing techniques are as follows:

Diacritics Removal: The small marks used to indicate pronunciation in Arabic were systematically eliminated from the datasets (e.g., مُجتَمع > مجتمع).

Hamza Normalization: A glottal stop represented in multiple ways in Arabic, underwent a normalization process (أ، إ، آ > ا). This in turn included normalizing Lam Alif.

Kashida Removal: Excessive elongation of Arabic letters was adjusted (e.g., فلسطــــــــين > فلسطين).

Punctuation Removal: All punctuation marks were removed from the datasets.

Spelling Error Correction: Common spelling errors in the text were systematically corrected.

In addition, as part of our pre-processing pipeline, we implemented another step involving the mapping of numerical labels to their corresponding country names. Linking numerical labels to countries helped us associate data with geographic regions during the stages of analysis and training. It was an important initial step in preparing the data for further processing. The labels 0 to 17 were respectively associated with the following countries: Iraq, Oman, Syria, Yemen, Morocco, Lebanon, Tunisia, Kuwait, Algeria, UAE, Sudan, Libya, Jordan, Egypt, Bahrain, Palestine, Saudi Arabia, and Qatar.

2.3 Classifiers

We deemed this as a classification task. We used Transformer-based models such as Arabertv2 base, and large (Antoun et al., 2020) and CAMeLBERT-Mix DID MADAR (Inoue et al., 2021). The choice of these BERT-based models was because they were trained on data we were allowed to use. We also used traditional models such as Naive Bayes, SVC, XGBoost, AdaBoost, and Random Forests. All the models were trained on a combined dataset of all the provided datasets. Our BERT-based model Arabertv2-large performed the best on the development dataset. To fine-tune AraBERT for sequence classification, we employ the same approach that (Antoun et al., 2020) used. This involves taking the final hidden state of the initial token, specifically associated with the word embedding of the special "[CLS]" token positioned at the beginning of each sentence. Subsequently, we integrate a basic feed-forward layer coupled with the standard Softmax function to yield a probability distribution across the predicted output classes. During the fine-tuning process, both the classifier and the pre-trained model's weights are collaboratively trained to maximize the log probability of correctly predicting the class.

In terms of the training setup, we utilize a set of configuration parameters encapsulated in the 'TrainingArguments' variable. The parameters we used are similar to that of (Antoun et al., 2020) provided in their examples notebook. We set 'adam_epsilon' to a value of 1e-8 for optimization, 'learning_rate' at 2e-5 for the learning rate, and 'fp16' can be enabled when using high-performance GPUs like V100 or T4. The 'per_device_train_batch_size' is set at 16, although it can go up to 64 when working with 16GB of GPU memory and sequences of a maximum length of 128. To manage memory effectively, 'gradient_accumulation_steps' is configured at 2, allowing for an increase in batch size.

The training process spans 'num_train_epochs' for 3 cycles. 'warmup_ratio' is set to 0, indicating no warm-up steps. Evaluation is incorporated ('do_eval = True'), and this evaluation strat-

Model Name	F1-Score	Accuracy	Precision	Recall
Arabertv2-Large	0.71	0.71	0.71	0.71
Arabertv2-base	0.71	0.71	0.70	0.71
CAMeLBERT-Mix DID MADAR	0.71	0.71	0.70	0.71
XGBoost	0.52	0.51	0.60	0.51
Random Forest	0.43	0.42	0.51	0.42
Naïve Bayes	0.41	0.45	0.73	0.45
SVC	0.39	0.40	0.58	0.40
AdaBoost	0.18	0.18	0.50	0.18

Table 1: Model Performance Comparison on Development Data

Model Name	F1	Accuracy	Precision	Recall
Arabertv2-large	70.22	70.78	71.32	70.78

Table 2: Official results of the IUNADI submission

egy is executed 'epoch' by 'epoch'. Further, the 'save_strategy' also operates 'epoch' by 'epoch', and it's designed to 'load_best_model_at_end' for automatic selection of the best model based on a specified metric, 'macro_f1', where 'greater_is_better' is set to true. 'Macro_f1' is used because F1 was the official metric. Lastly, a 'seed' value of 47 is employed for reproducibility.

Finally, during the training of Arabert-Large, we employed a training ensemble methodology within the framework of a 5-fold cross-validation setup. Our final predictions were derived by aggregating the scores of the individual models. This ensemble approach facilitated improved model performance and robustness in our research.

3 Evaluation

For subtask 1, the evaluation metrics will include precision, recall, f-score, and accuracy. Macro-averaged F-score will be the official metric; hence we report our results using this metric along with all the evaluation metrics. We decided which models to submit based on the model's performance on the development dataset provided by the organizers.

4 Results

As shown in table 2, we only submitted a single system for evaluation, namely, Arabertv2-large. We selected this model because it has over 2.5 times more parameters than Arabertv2-base and CAMeLBERT-Mix DID MADAR. Our system achieved an F-1 score of 70.22 on the test set.

In addition to the officially submitted systems, we performed a more extensive evaluation of the development set. We trained and evaluated 8 different classifiers. The results of these experiments are shown in table 1. The best model performance was achieved by the three models Arabertv2-large, Arabertv2-base, and CAMeLBERT-Mix DID MADAR. The non-neural classifiers generally showed lower performance than transformers.

Our pre-processing pipeline had a positive effect on the Random Forests model, improving the F1 score to 0.43, compared to 0.39 without pre-processing. In contrast, it had a detrimental impact on the Naive Bayes model, reducing F1 to 0.41 from 0.43 without pre-processing. The pipeline had no impact on the results of XGBOOST, SVC, and AdaBoost. It is important to note that pre-trained models already incorporate their own internal pre-processing pipelines. Even though the pipeline did not achieve significant results, we still believe it was necessary to eliminate redundancy and reduce data size.

5 Discussion

The Arabic dialect identification task, as explored in this research, addresses a crucial challenge in natural language processing, particularly for applications involving Arabic text. We observed promising results during the evaluation phase, demonstrating the system's ability to correctly identify Arabic dialects with a high degree of accuracy. However, it is essential to recognize that the task itself presents inherent challenges due to the nuances and variations present within Arabic dialects. Arabic speakers often code-switch between dialects and Standard Arabic, which affects the performance of models. Given an additional three months to work on this task, several avenues for improvement and further development can be pursued:

Fine-Tuning Strategies: Experimenting with advanced fine-tuning techniques, such as domain adaptation or multi-task learning, may help the model handle ambiguous phrases and code-switching more effectively.

Post-Processing Techniques: Implementing post-processing techniques, such as dialect consistency checks, to ensure that the identified dialect remains consistent within a given text could mitigate errors caused by code-switching.

6 Conclusion

In this paper, we have detailed our contributions to the NADI 2023 shared task on Arabic tweet classification across 18 Arab countries. Our experiments have revealed that employing Arabertv2-large yields the most promising results. Our system achieved a ranking of 13th out of 16 participating teams. Looking ahead, our future research will explore the potential benefits of employing ensemble-based approaches with transformer-based models. Additionally, we are keen to investigate the potential advantages of incorporating tokenization, stop word removal or splitting, and stemming into our pre-processing pipeline.

Ethics Statement

This work is primarily for the benefit of the Arabic language community, which despite having hundreds of millions of speakers, still lacks computational resources. While we believe that our project does not pose any potential harm, we urge users to take all ethical considerations into account when using it.

Acknowledgments

This research was supported in part by Lilly Endowment, Inc., through its support for the Indiana University Pervasive Technology Institute. We would also like to thank Indiana University for providing access to their computing resources and servers, which were instrumental in conducting the experiments for this research.

References

Muhammad Abdul-Mageed, AbdelRahim Elmadany, Chiyu Zhang, ElMoatez Billah Nagoudi, Houda Bouamor, and Nizar Habash. 2023. NADI 2023: The Fourth Nuanced Arabic Dialect Identification Shared Task. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*.

Muhammad Abdul-Mageed, Chiyu Zhang, Houda Bouamor, and Nizar Habash. 2020. Nadi 2020: The first nuanced arabic dialect identification shared task. *arXiv preprint arXiv:2010.11334*.

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, Houda Bouamor, and Nizar Habash. 2021. Nadi 2021: The second nuanced arabic dialect identification shared task. *arXiv preprint arXiv:2103.08466*.

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, Houda Bouamor, and Nizar Habash. 2022. NADI 2022: The third nuanced Arabic dialect identification shared task. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 85–97, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

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

Gilbert Badaro, Ouijdane El Jundi, Ali Khaddaj, Ahmad Maarouf, Reine Kain, Hazem Hajj, and Wassim El-Hajj. 2018. Ema at semeval-2018 task 1: Emotion mining for arabic. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 236–244.

Houda Bouamor, Nizar Habash, Mohammad Salameh, Wajdi Zaghouani, Owen Rambow, Dana Abdulrahim, Dima Obeid, Salam Khalifa, Fatima Eryani, Andreas 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)*.

Kareem Darwish, Hassan Sajjad, and Hamdy Mubarak. 2014. Verifiably effective arabic dialect identification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1465–1468.

Mona Diab, Nizar Habash, Owen Rambow, Mohamed Altantawy, and Youssef Benajiba. 2010. Colaba: Arabic dialect annotation and processing. In *Lrec workshop on semitic language processing*, pages 66–74.

Genichiro Inoue, Basel Alhafni, Nazym Baimukan, Houda Bouamor, and Nizar Habash. 2021. The interplay of variant, size, and task type in arabic pre-trained language models. *arXiv preprint arXiv:2103.06678*.

Maha Jarrar, Nizar Habash, Dana Akra, and Nasser Zalmout. 2014. Building a corpus for palestinian arabic: a preliminary study. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 18–27.

Maha Jarrar, Nizar Habash, Fatima Alrimawi, Dana Akra, and Nasser Zalmout. 2017. Curras: an annotated corpus for the palestinian arabic dialect. *Language Resources and Evaluation*, 51:745–775.

Luma Lulu and Ahmed Elnagar. 2018. Automatic arabic dialect classification using deep learning models. *Procedia computer science*, 142:262–269.

Ali Y Muaad, Harisha J Davanagere, D S Guru, Jelili B Benifa, Chanda Chola, Hani AlSalman, Abdullah Gumaei, and Moulay A Al-antari. 2022. Arabic document classification: performance investigation of preprocessing and representation techniques. *Mathematical Problems in Engineering*, 2022:1–16.

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

The Helsinki-NLP Submissions at NADI 2023 Shared Task: Walking the Baseline

Yves Scherrer[1,2] first.last@ifi.uio.no
Aleksandra Miletić[1] first.last@helsinki.fi
Olli Kuparinen[1,3] first.last@tuni.fi

[1]Department of Digital Humanities, University of Helsinki
[2]Department of Informatics, University of Oslo
[3]Faculty of Information Technology and Commmunication Sciences, Tampere University

Abstract

The Helsinki-NLP team participated in the NADI 2023 shared tasks on Arabic dialect translation with seven submissions. We used statistical (SMT) and neural machine translation (NMT) methods and explored character- and subword-based data preprocessing. Our submissions placed second in both tracks. In the open track, our winning submission is a character-level SMT system with additional Modern Standard Arabic language models. In the closed track, our best BLEU scores were obtained with the leave-as-is baseline, a simple copy of the input, and narrowly followed by SMT systems. In both tracks, fine-tuning existing multilingual models such as AraT5 or ByT5 did not yield superior performance compared to SMT.

1 Introduction

This paper presents the Helsinki-NLP submissions to the NADI 2023 shared tasks. We participated in Subtasks 2 and 3, which consisted in translating dialectal data into Modern Standard Arabic (MSA) (Abdul-Mageed et al., 2023). This was the first time the NADI shared task involved translation, following past tasks on dialect identification and sentiment analysis (Abdul-Mageed et al., 2020, 2021, 2022).

The Arabic dialectal continuum stretches from Morocco in the west to Oman in the east. Various classifications of the dialects have been proposed, ranging from large regions to country-level or even city-level divisions (Bouamor et al., 2018; Habash, 2022). The Arabic language area is also well known for its diglossic situation. While Modern Standard Arabic is used in education, media and culture across the continuum, it is not native to any of the dialectal regions.

The translation subtasks focused on four Arabic dialects: Egyptian, Emirati, Jordanian, and Palestinian. The shared task organizers provided the MADAR corpus (Bouamor et al., 2018) as the training material for the closed track (Subtask 2), which did not allow for the use of additional training data. The Subtask 3 was described as open track where any additional training material was allowed.

Since our initial experiments showed that neural models were particularly affected by the small size of the MADAR training data, a large part of our efforts went into creating additional parallel data for the in Subtask 3 models. In particular, we focused on freely available monolingual MSA corpora, which we then back-translated to three target dialects, grouping Jordanian and Palestinian together. Adding the back-translated data to the original training corpus allowed our neural models to perform on par with less data-hungry statistical models.

We participated in Subtask 2 with three submissions and in Subtask 3 with four submissions. Our submissions can be divided into four different approaches:

- **LAI** – the leave-as-is baseline consisting of a copy of the input text,
- **SMT** – character-level statistical machine translation models;
- **NMT** – Transformer-based neural machine translation models;
- **ByT5** and **AraT5** – pretrained sequence-to-sequence models fine-tuned with task-specific data.

Our best performing translation system was SMT for both subtasks, but it was not able to outperform the LAI baseline in Subtask 2, at least in terms of the BLEU score (Papineni et al. 2002; see Section 5.1 for a critical discussion of evaluation measures). Our submissions placed second on both subtasks.

Section 2 describes the data collection and preparation whereas Section 3 outlines the proposed models in more detail. Our results are presented in Section 4 and further discussed in Section 5.

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 670–677
December 7, 2023 ©2023 Association for Computational Linguistics

Section 6 offers conclusions of our work.

2 Data Collection and Augmentation

2.1 MADAR3

The training resource provided by the organizers was the MADAR corpus (Bouamor et al., 2018). The dataset contains the same sentences in different Arabic dialects from 25 cities, as well as in English, French and MSA. The corpus was created by translating sentences from the Basic Traveling Expression Corpus (BTEC) (Takezawa et al., 2007) into Arabic dialects.

We found in early experiments that our models achieved better results when excluding the data from Maghrebi and Yemeni dialects, since the development and test sets do not cover these dialect groups. Therefore, for all our submissions, we use the subset of MADAR that covers the Nile Basin, Levant and Gulf regions. We refer to this subcorpus as **MADAR3** throughout the paper.

2.2 MSA data

Considering that parallel resources for the target dialects are hard to come by, we focused our collection efforts on monolingual MSA data, taking inspiration from the AraT5 pretraining setup (Nagoudi et al., 2022). In particular, we used the following resources:

AraNews is a collection of Arabic newspaper texts from 15 Arab nations, the United States of America and the United Kingdom (Nagoudi et al., 2020).

Leipzig News is a dataset of Arabic news curated by the Leipzig Corpora Collection. The data comes from mostly Saudi Arabian news outlets (Goldhahn et al., 2012).

OSIAN is an Arabic news corpus crawled from the web (Zeroual et al., 2019). It contains articles from 31 international Arabic news broadcasting platforms.

Tatoeba is a project collecting translations of sentences in the web. The data is available in OPUS (Tiedemann, 2012).

TED is a corpus of translated subtitles from over 4000 TED talks (Reimers and Gurevych, 2020). The data is available in OPUS (Tiedemann, 2012).

Wikipedia is a Wikipedia-based corpus we extracted from the Arabic Wikipedia using WikiExtractor (Attardi, 2015)[1]. The extracted data was subsequently sentence-segmented and deduplicated at sentence level (only the exact matches were removed).

Corpus	Sentences	Words
AraNews	59,270	2,643,313
Leipzig News	1,000,000	23,972,851
OSIAN	1,000,000	21,532,389
Tatoeba	47,471	231,507
TED	403,845	5,652,867
Wikipedia	11,368,818	193,912,867

Table 1: Size of additional datasets

An overview of corpus sizes is given in Table 1.[2] Of these resources, AraNews, OSIAN and Wikipedia were used to pretrain AraT5 (V1).

2.3 Backtranslation of MSA data

While monolingual target-side data can easily be included into SMT systems in the form of additional language models, this is more difficult for neural models. The most common approach in this situation is to produce synthetic parallel data using backtranslation (Sennrich et al., 2016).

To this end, we reversed our dialect-specific SMT-mono models from Subtask 2 (see Section 3.2 for details) to produce three dialectal versions of all monolingual MSA data presented in Section 2.2. The backtranslated data was used to train or fine-tune the neural models for Subtask 3 (see Sections 3.3, 3.4 and 3.5).

The quality of the backtranslations is most likely poor, but we nevertheless expect backtranslation to work better than simpler data augmentation methods such as noise injection. Since the authors are not speakers of Arabic, the quality of the backtranslations could not be evaluated.

3 Models

3.1 LAI

As the shared task organizers did not provide an official baseline, we propose the leave-as-is (LAI)

[1] The extraction was done from the Wikimedia data dump `arwiki-20230801-pages-articles-multistream.xml.bz2`

[2] Note that we did not perform full tokenization of the corpora: the word counts in the table are based on whitespace-delimited tokens.

	Training data			Development set BLEU						Test set BLEU				
Model	MADAR3	MSA	BT MSA	Overall	EGY	EMI	JOR	PAL	Subm.	Overall	EGY	EMI	JOR	PAL
LAI	—	—	—	**15.78**	14.87	**26.31**	12.75	**12.90**	2.2	14.28	12.22	**23.13**	11.15	13.41
SMT-multi	✓	—	—	15.62	15.01	25.74	12.52	12.64	2.1	13.60	12.02	21.82	10.46	12.66
SMT-mono	✓	—	—	15.39	**15.91**	17.94	**14.84**	11.78	2.3	12.53	11.91	16.50	9.83	11.42
NMT	✓	—	—	2.61	3.24	2.52	0.00	2.32	—	—	—	—	—	—
ByT5	✓	✓	—	6.63	6.89	4.86	4.94	7.60	—	—	—	—	—	—
AraT5 V2	✓	✓	—	7.41	7.61	5.55	5.98	8.01	—	—	—	—	—	—
Best competitor										**14.76**	**16.04**	14.30	**12.55**	**13.55**
SMT-multi	✓	✓	—	**19.19**	18.88	25.66	17.24	**17.16**	3.1	17.69	16.11	25.81	15.60	15.91
SMT-mono	✓	✓	—	18.61	**19.03**	26.89	13.12	17.14	—	—	—	—	—	—
NMT	✓	—	✓	18.40	16.78	25.34	**19.59**	14.36	3.2	16.88	15.17	24.77	15.41	14.45
ByT5	✓	✓	✓	17.69	16.68	24.90	16.01	14.03	3.3	16.10	15.55	21.79	13.73	13.34
AraT5 V2	✓	✓	✓	19.14	18.49	**28.25**	17.19	14.80	3.4	17.46	15.50	25.06	15.97	15.06
Best competitor										**21.10**	**17.65**	**28.46**	**22.03**	**17.29**

Table 2: Overview of the tested models and their BLEU scores (↑) on the development and test sets. *MSA*: monolingual MSA data, *BT MSA* monolingual MSA data back-translated to three dialects. ✓: used for pre-training, ✓: used for training or fine-tuning. EGY: Egyptian, EMI: Emirati, JOR: Jordanian, PAL: Palestinian. The horizontal line separates closed (Subtask 2) from open (Subtask 3) submissions according to the organizer-defined criteria.

baseline: an unchanged copy of the input file. We do not suggest that LAI is a potential solution to the task; rather, we introduce it as a way of estimating the task difficulty.

We were unable to beat this baseline with the systems that only use the MADAR corpus for training or fine-tuning in terms of BLEU score. Therefore, we decided to submit LAI as one of our contributions. We think it is interesting to also compare the other participants' systems with this baseline. For example, even the best submitted subtask 2 system scores behind LAI on the Emirati dialect (see Table 2).

3.2 SMT

We use a character-level statistical machine translation model based on the Moses toolkit. We split all sentences into character sequences and treat each character as a separate translation unit.[3]

We provide two variants of the SMT approach. **SMT-multi** is a single model trained on all dialects from MADAR3. **SMT-mono** is a collection of 3 models, each of which is trained on the MADAR texts of one major dialect area (Nile Basin, Levant, Gulf). At prediction time, the relevant model is chosen according to the provided dialect labels.

Furthermore, each of the two models is made available in a closed and an open variant. The closed variant contains a single language model trained on the MSA side of MADAR. The open variant contains a total of 7 language models, corresponding to the different MSA corpora listed in Section 2.2 in addition to MADAR.

3.3 NMT

Our neural machine translation method is based on the Transformer architecture. The model was trained with OpenNMT-py (Klein et al., 2017).[4]

We tokenized the data using the unigram model implemented in the SentencePiece library (Kudo and Richardson, 2018), as it has outperformed BPE-based segmentation when the studied texts include inconsistent writing or non-standard language (Kanjirangat et al., 2023). We experimented with three different vocabulary sizes (300, 500, 1000) and found the smallest (300) to offer the best performance.

Furthermore, we found that the NMT model's performance was enhanced by adding a dialect tag at the beginning of the source sentence. We used the three dialect labels of MADAR3.

The NMT model trained on MADAR3 alone did not produce competitive scores. We only submitted an NMT model trained both on MADAR3 and on the backtranslations.

3.4 ByT5

ByT5 (Xue et al., 2022) is a multilingual pre-trained model of the T5 family (Raffel et al., 2020)

[3] Character-level models outperformed SMT models with words and subwords in preliminary experiments. Model parameters are presented in Table 5 in the Appendix.

[4] Experimental details for each model are provided in Table 5 in Appendix A.

that encodes all text as UTF-8 encoded byte sequences. It is pre-trained on the multilingual m4C corpus (Xue et al., 2021), with 1.66% of the data in Arabic. ByT5 was used by the winning team (Samuel and Straka, 2021) in the MultiLexNorm shared task (van der Goot et al., 2021), in which the participants had to normalize social media texts of various languages. We expect that Arabic dialect-to-standard translation consists to a large extent of local changes of individual characters. We therefore find that a byte-based model is a good fit for this task.

We fine-tuned the `byt5-base` model with MADAR3, but found the performance subpar. For our submission, due to computational limitations, we fine-tuned the `byt5-small` model with a random sample of 1M sentences from our backtranslated data and MADAR3.

3.5 AraT5

AraT5 (Nagoudi et al., 2022) is a pre-trained model of the T5 family specifically focused on Arabic, enabling tasks like machine translation into and out of Arabic, summarization, transliteration and other sequence-to-sequence transformation tasks. During the competition, the second version AraT5-V2 was made available. We use the `AraT5v2-base-1024` foundation model for our experiments.

Fine-tuning AraT5-V2 on MADAR3 only did not yield competitive results. Instead, we submitted a model fine-tuned on MADAR3 and a random sample of the backtranslations, with a total of 1.4M sentence pairs (15% of the full dataset).[5]

4 Results

4.1 Results on the development set

Our results on the development set are shown in the middle panel of Table 2 with the official evaluation metric BLEU. Our best submission in Subtask 2 is the leave-as-is baseline (LAI; Section 3.1). The fact that unmodified input achieves better results than machine translation approaches can be taken as an indicator of the difficulty of the closed track task. Note, however, that our best-performing machine translation approach (SMT-multi) is in general less than one BLEU point below LAI.

The inclusion of additional training material in Subtask 3 led to a significant improvement for neural methods, as illustrated by the results of NMT, ByT5 and AraT5 in the lower part of Table 2. Nevertheless, our best performing approach remains SMT-multi, which scores first overall, and for all individual dialects except for Jordanian. AraT5 is the second best model overall, but note that SMT-mono outperforms it on Egyptian and Palestinian. The scores across different models are the most stable for Egyptian, and they vary the most on Emirati, where the difference between the best (SMT-multi) and worst model (ByT5) is around 4 BLEU points.

4.2 Official results

The right-hand panel of Table 2 shows the official results on the test set. For comparison, we added the results of the top-performing system of each subtask.

For Subtask 2, the LAI baseline outperformed both of our SMT systems, and got close to the best submission. It can be noted that our LAI model outperformed the best competitor on Emirati by a large margin, suggesting that models tend to over-normalize this dialect.

For Subtask 3, SMT and AraT5 were our best submissions, as could be expected from the development set scores. However, there is a significant gap to the best competitor, especially for Emirati and Jordanian. We would like to note however that our Subtask 3 submissions rely on similar training data as was used for AraT5 pretraining, but in a smaller volume. In that sense, it may be more relevant to compare our systems with Subtask 2 submissions that are based on AraT5.

Note that in all our experiments we systematically use sentence-level contexts. However, our previous work has shown that contexts of sliding windows of three words can bring significant improvements, especially for the TF-based systems (Kuparinen et al., 2023). This approach requires word-level data alignments which are not trivial to produce. Therefore we defer this to future work.

5 Discussion

5.1 Evaluation metrics

The BLEU score (Papineni et al., 2002), which was used as the official metric in this shared task, treats each word as an atomic unit and considers a word as wrong even if only one character is incorrect. However, in dialect-to-standard translation tasks, an large amount of differences is expected to concern changes of individual characters. It

[5] The samples used for byT5 and AraT5 differ due to computational time constraints.

Model		Overall	EGY	EMI	JOR	PAL
		BLEU				
2.2	LAI	**15.78**	14.87	**26.31**	12.75	**12.90**
2.3	SMT-mono	15.39	**15.91**	17.94	**14.84**	11.78
2.1	SMT-multi	15.62	15.01	25.74	12.52	12.64
		chrF				
2.2	LAI	45.02	46.56	49.47	40.85	44.10
2.3	SMT-mono	**46.96**	**49.11**	**51.11**	**43.81**	**44.69**
2.1	SMT-multi	44.96	46.60	49.37	40.81	43.94

Table 3: BLEU and chrF scores on the development set.

Model		NADI	MADAR3
2.2	LAI	14.28	3.48
2.1	SMT-multi	13.60	3.53
3.1	SMT-multi	17.69	10.22
3.4	AraT5 V2	17.46	17.85

Table 4: Overall BLEU scores for the NADI and MADAR3 test sets.

might therefore be interesting to consider metrics that reflects this better, for example the chrF score (Popović, 2015), which is based on the precision and recall of character n-grams.

Table 3 compares the development set BLEU scores with the chrF scores of our Subtask 2 submissions. According to BLEU, LAI is the best performing system, mostly thanks to its good performance on Emirati. SMT-mono is the worst of the three despite winning on two individual dialects. In contrast, according to chrF, SMT-mono outperforms all other systems on all four dialects. The large variation on Emirati has also disappeared.

This suggests that our SMT-mono system could in fact be perceived as better than the higher-ranked LAI baseline. It would be instructive to see which of the two evaluation metrics correlates better with human assessment on this particular task.

5.2 Test data domains

While the MADAR corpus contains relatively short and simple sentences from the travel domain, the development and test data provided for the NADI shared task comes from a different source and text domain. It can be interesting to see how the proposed translation models fare on both domains.

To this end, we extracted the test instances from the MADAR3 corpus (which were held out from model training) and evaluated some of our submissions on them. Table 4 provides a comparison of the results on the NADI test data and the MADAR3 test data.

There is a striking difference in terms of LAI BLEU between the two datasets: NADI seems to be much "easier" than MADAR, in the sense that fewer replacements are required. For both datasets, the closed-track SMT model does not do any better than the baseline. The two selected open-track models have very similar performances on the NADI test set, but differ greatly on their performance on MADAR. AraT5, presumably thanks to the large amount of pretraining data, generalizes much better to the more difficult MADAR test set.

6 Conclusions

In this paper, we described our participation in the NADI shared task, where we submitted seven systems to two tracks. Our submissions placed second on both tracks. Our strongest translation method was SMT in both tracks, but given the difficulty of the task, it was outperformed by an LAI baseline in the closed track. Neural models closed the gap to the SMT models only with large amounts of additional parallel data obtained through backtranslation.

We would like to note again that our open track submissions do not use any human-translated parallel training data besides MADAR, and that the total amount of training data is smaller than what was used for AraT5 pre-training. This makes our models, in particular the SMT ones, more data efficient than large pretrained models such as AraT5 or ByT5.

We also showed that the participating systems could have been ranked differently with a character-based evaluation metric, which underlines the importance of the selected metrics.

Acknowledgements

This work has been supported by the Academy of Finland through project No. 342859 "CorCoDial – Corpus-based computational dialectology".

The authors wish to acknowledge CSC – IT Center for Science, Finland, for computational resources.

References

Muhammad Abdul-Mageed, AbdelRahim Elmadany, Chiyu Zhang, ElMoatez Billah Nagoudi, Houda Bouamor, and Nizar Habash. 2023. NADI 2023: The Fourth Nuanced Arabic Dialect Identification

Shared Task. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*.

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, Barcelona, Spain (Online). Association for Computational Linguistics.

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, Houda Bouamor, and Nizar Habash. 2021. NADI 2021: The second nuanced Arabic dialect identification shared task. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 244–259, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, Houda Bouamor, and Nizar Habash. 2022. NADI 2022: The third nuanced Arabic dialect identification shared task. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 85–97, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Giuseppe Attardi. 2015. Wikiextractor. `https://github.com/attardi/wikiextractor`.

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).

Dirk Goldhahn, Thomas Eckart, and Uwe Quasthoff. 2012. Building large monolingual dictionaries at the leipzig corpora collection: From 100 to 200 languages. In *International Conference on Language Resources and Evaluation*.

Nizar Y Habash. 2022. *Introduction to Arabic natural language processing*. Springer Nature.

Vani Kanjirangat, Tanja Samardžić, Ljiljana Dolamic, and Fabio Rinaldi. 2023. Optimizing the size of subword vocabularies in dialect classification. In *Tenth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2023)*, pages 14–30, Dubrovnik, Croatia. Association for Computational Linguistics.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. 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.

Olli Kuparinen, Aleksandra Miletić, and Yves Scherrer. 2023. Dialect-to-Standard Normalization: A Large-Scale Multilingual Evaluation. In *Findings of EMNLP2023*. (accepted).

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.

El Moatez Billah Nagoudi, AbdelRahim Elmadany, Muhammad Abdul-Mageed, and Tariq Alhindi. 2020. Machine generation and detection of Arabic manipulated and fake news. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 69–84, Barcelona, Spain (Online). 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.

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.

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.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

David Samuel and Milan Straka. 2021. ÚFAL at MultiLexNorm 2021: Improving multilingual lexical normalization by fine-tuning ByT5. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 483–492, Online. Association for Computational Linguistics.

Yves Scherrer. 2023. Character alignment methods for dialect-to-standard normalization. In *Proceedings of the 20th SIGMORPHON workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 110–116, Toronto, Canada. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Toshiyuki Takezawa, Genichiro Kikui, Masahide Mizushima, and Eiichiro Sumita. 2007. Multilingual spoken language corpus development for communication research. In *International Journal of Computational Linguistics & Chinese Language Processing, Volume 12, Number 3, September 2007: Special Issue on Invited Papers from ISCSLP 2006*, pages 303–324.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Rob van der Goot, Alan Ramponi, Arkaitz Zubiaga, Barbara Plank, Benjamin Muller, Iñaki San Vicente Roncal, Nikola Ljubešić, Özlem Çetinoğlu, Rahmad Mahendra, Talha Çolakoğlu, Timothy Baldwin, Tommaso Caselli, and Wladimir Sidorenko. 2021. MultiLexNorm: A shared task on multilingual lexical normalization. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 493–509, 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.

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, Online. Association for Computational Linguistics.

Imad Zeroual, Dirk Goldhahn, Thomas Eckart, and Abdelhak Lakhouaja. 2019. OSIAN: Open source international Arabic news corpus - preparation and integration into the CLARIN-infrastructure. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 175–182, Florence, Italy. Association for Computational Linguistics.

A Experimental Details

We trained all neural models on a single NVIDIA V100 GPU. The SMT models were trained on a Xeon Gold 6230 CPU. We will make the training scripts and the additional data publicly available for the final submission.

Model	Parameter	Selected values	Considered alternatives
SMT	Subword tokenization	characters	words, unigram subwords
	Alignment tool	eflomal	— (Scherrer, 2023)
	Alignment symmetrization	grow-diag-final-and	—
	Language model n-gram size	10	—
	Maximum phrase length	10	—
	Distortion	disabled	—
	Tuning method	MERT	—
NMT	Subword tokenization	unigram subwords	characters
	Encoder + decoder layers	6 + 6	—
	Attention heads	8	—
	Embedding dimensions	512	—
	Hidden layer dimensions	512	—
	Position representation clipping	4	—
	Dropout	0.1	—
	Label smoothing	0.1	—
	Optimizer	Adam	—
	Adam $\beta2$	0.98	0.998
	Batch size / accumulate gradient	2 * 5000 tokens	—
	Initial learning rate	0.1	0.01, 2.0
	Decay	Noam, 10000 warmup steps	—
	Max. training sequence length	1000	—
	Max. prediction sequence length	1000	—
	Training time	100000 steps	—
ByT5	Foundation model	google/byt5-small	google/byt5-base
	Max. sequence length	512	—
	Batch size	8 sentences	—
	Early stopping	disabled	—
	Training time	5 epochs	—
	Model selection criterion	validation loss	—
AraT5	Foundation model	UBC-NLP/AraT5v2-base-1024	UBC-NLP/AraT5-base
	Max. sequence length	256	—
	Batch size	12 sentences	—
	Early stopping	5 epochs	—
	Max. training time	20 epochs	—
	Model selection criterion	validation loss	—

Table 5: Hyperparameter settings.

Mavericks at NADI 2023 Shared Task: Unravelling Regional Nuances through Dialect Identification using Transformer-based Approach

Vedant Deshpande [*], **Yash Patwardhan**[*], **Kshitij Deshpande**[*],
Sudeep Mangalvedhekar[*] and **Ravindra Murumkar**[*]
Pune Institute of Computer Technology, Pune
{vedantd41, yash23pat, kshitij.deshpande7, sudeepm117}@gmail.com,
rbmurumkar@pict.edu

Abstract

In this paper, we present our approach for the "Nuanced Arabic Dialect Identification (NADI) Shared Task 2023". We highlight our methodology for subtask 1 which deals with country-level dialect identification. Recognizing dialects plays an instrumental role in enhancing the performance of various downstream NLP tasks such as speech recognition and translation. The task uses the Twitter dataset (TWT-2023) that encompasses 18 dialects for the multi-class classification problem. Numerous transformer-based models, pre-trained on Arabic language, are employed for identifying country-level dialects. We fine-tune these state-of-the-art models on the provided dataset. The ensembling method is leveraged to yield improved performance of the system. We achieved an F_1-score of 76.65 (11th rank on the leaderboard) on the test dataset.

1 Introduction

Dialects, which are variations of a language, often differ in their vocabulary, grammar, pronunciation, and occasionally even cultural quirks. The practice of identifying the particular dialect or regional variety of a language that is used in a text or speech sample is known as dialect identification. The goal of dialect identification is to categorize a text or speech into one of the many dialects or regional adaptations that may exist. For many NLP applications, including language modeling, speech recognition, and data retrieval, this task may be vital.

Arabic, with its plethora of dialects, is a rich language. However, many of these dialects are not studied in depth because of a dearth of monetary backing and available datasets. Arabic dialect identification can assist in perpetuating linguistic diversity by acknowledging and valuing various dialects. It contributes to addressing the gap between existing NLP techniques and the rich fabric of regional dialectal differences in a globalized setting. Rule-based strategies for Arabic dialect identification have given way to data-driven techniques, with a focus on machine learning, deep learning, and the creation of corpora of languages and datasets. The accuracy of dialect detection has risen significantly with the use of multilingual pre-trained models such as BERT and its derivatives.

This paper presents our approach for subtask 1: Country-level Dialect Identification, which poses a multiclass classification problem (Abdul-Mageed et al., 2023). Multiclass classification is a form of statistical modeling or machine learning problem where the objective is to classify data into more than two unique classes or labels. We aim to classify the tweets and map them into their respective dialect labels. We have demonstrated the use of various transformer-based models on the given Arabic data. The ensembling method has been leveraged to enhance the performance of the proposed system.

2 Related Work

Dialect detection in Arabic is an arduous task due to several factors, including the lack of a consistent spelling system, the medium's characteristics, and the scarcity of data. Surveys on deep learning and Natural Language Processing methods for processing Arabic data were presented in 2015 (Shoufan and Alameri, 2015) and 2017 (Al-Ayyoub et al., 2018), focusing on the identification of Arabic dialects. However, only 6 Arabic dialect classes had been examined until that time. The MADAR project was launched in 2018 to provide a large corpus of 25 Arabic city dialects (Bouamor et al., 2018). A study on the classification of dialects in 25 Arab cities used multi-label classification methods and examined a wide range of features, yielding promising results (Salameh et al., 2018). Employing supervised machine learning methods on Arabic NLP tasks was found to be a difficult

[*] Equal contribution

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 678–682
December 7, 2023 ©2023 Association for Computational Linguistics

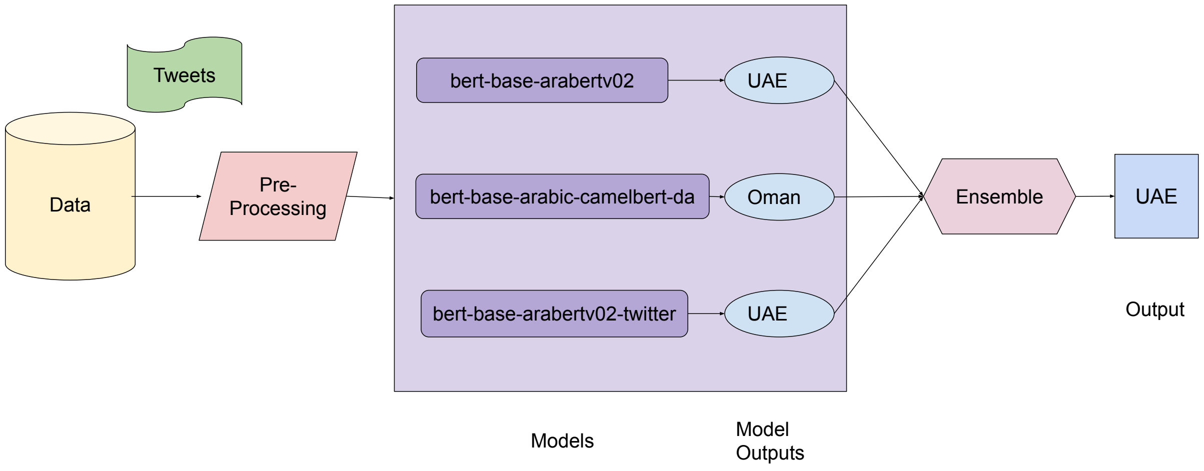


Figure 1: System architecture

feat because of the lack of resources in the Arabic language (El Mekki et al., 2020). As a result, scholars and researchers have introduced plenty of initiatives to make new datasets available and encourage more people to work in the field of Arabic NLP. One of the initiatives, Nuanced Arabic Dialect Identification (NADI) shared tasks, was started in 2020 which comprised country-level and province-level dialect detection (Abdul-Mageed et al., 2020). BERT (Bidirectional Encoder Representation from Transformers) (Devlin et al., 2019) models have been commonly used for these dialect detection tasks. A multilingual BERT model was pretrained on unlabeled tweets and fine-tuned for the classification task by Mansour et al. (2020). AraBERT was finetuned on an additional dataset produced by reverse translating the NADI dataset and employed for the dialect detection task by Tahssin et al. (2020). Furthermore, Gaanoun and Benelallam (2020) utilized ensembling methods and semi-supervised methods along with Arabic-BERT. A system comprising an ensembling of multiple models was created using MARBERT as the base model, which yielded promising results (AlKhamissi et al., 2021). In the NADI 2022 shared task, AlShenaifi and Azmi (2022) pretrained AraBERT model and BiLSTM model for dialect detection. Various models were combined and performance was enhanced using a combination of TF-IDF and n-grams. An ensembling of transformer-based models, predominantly using variations of MARBERT was employed for dialect detection as well as sentiment analysis, in the NADI 2022 shared tasks (Bayrak and Issifu, 2022), (Khered et al., 2022). (Oumar and Mrini, 2022) addressed the issue regarding an imbalance in the classes of the NADI dataset by using focal loss and employed various Arabic BERT-based models.

This paper proposes a system that employs an ensemble of transformer-based models, specifically variations of BERT for the classification task.

Dataset	Number of Samples
Training	18000
Development	1800
Testing	3600

Table 1: Dataset's training, development, and test split

3 Data

The dataset provided for subtask 1: Country-level dialect identification contains tweets. The given Twitter dataset comprises 18 dialects and a corpus of a total of 23400 tweets. The entire dataset is split into training (76.92%), development (7.69%), and test (15.38%). Additionally, datasets of previous years (Abdul-Mageed et al., 2020, 2021; Bouamor et al., 2018) are also provided for the training purpose. As shown in table 1, the training data has 18000 tweet samples, development data has 1800 samples and testing data has 3600 samples. The dataset contains features such as id, content, and label. Every sample's tweet content in the training dataset is labeled with its dialect. This subtask falls under the category of multi-class classification.

The provided dataset needed to be pre-processed before passing it to the model. We make use of regular expressions to remove "noisy" elements from the input texts. Texts like "USER", "NUM" and "URL" are removed from the input because they don't contribute additional information to the model's understanding.

4 System

The given subtask tackles the problem of country-level dialect identification. This comes under the umbrella of multi-class classification problems for which Language Models have been extensively used and have achieved impressive results. The models are trained for 10 epochs with a learning rate of 1e-5, a batch size of 32, and the AdamW optimizer. We experiment and use several language models and ensembling methods in our research, as shown in Figure 1.

4.1 AraBERT

Antoun et al. (2020) addresses how BERT models that have been pre-trained on a sizable corpus of a particular language, such as Arabic, do well on language comprehension tasks. They point out several such models, which are used in our study to help deliver cutting-edge outcomes for the Arabic language.

The 70 million phrases that make up the pre-training dataset, which is around 24 GB in size, are used to train the models. The news in the data covers a wide range of topics that is valuable for many downstream applications. The pre-training tasks that aid in the models' contextual knowledge of the input sequence include the Next Sentence Prediction Task and Masked Language Modelling Tasks. To demonstrate AraBERT's efficacy across diverse tasks and domains, it was tested on three NLP tasks: entity recognition, sentiment analysis, and question-answering.

Small adjustments have been made to the pre-training phases and parameters for the selected AraBERT model versions. AraBERT v1 or v0.1 are the original models, and v2 or v0.2 are the more recent versions with improved pre-processing and vocabulary. In addition to the dataset used for the other v0.2 models, AraBERTv0.2-Twitter-base is pre-trained with 60 million multi-dialect tweets. It possesses 136 parameters. Pre-trained examples for AraBERTv2-base include 207M instances with a sequence length of 512 and 420M examples with a sequence length of 128.

4.2 CAMeLBERT

Inoue et al. (2021) introduced the CAMelBERT model collection, which consists of more than eight pre-trained models for NLP tasks in Arabic. The parameters taken into consideration for the experiment were the task type, language variant, and size. Language models were provided in several variants, including classical Arabic (CA), dialectal Arabic (DA), and Modern Standard Arabic (MSA), with the DA variant being chosen for this study. The models were pretrained on variations of the MADAR dataset and NADI datasets for the Dialect Identification task. CAMelBERT was trained with the Adam optimizer and a learning rate of 1e-4. The pre-trained models are evaluated on five major tasks in NLP: Sentiment Analysis, Dialect Identification, POS tagging, Named Entity Recognition, and Poetry Classification.

5 Ensembling

Ensembling is a technique that integrates the output of multiple models to get the system's eventual outcome. For this, both statistical and non-statistical methods are employed. Ensembling is beneficial since it contributes to the production of results that are superior to those provided by the individual models.

We note that the "hard voting" ensemble strategy emerges as the most effective and precise among the many strategies used for ensembling. In hard voting, the final prediction is chosen based on the majority vote or the "mode" of all the predictions. It reduces the volatility in the outcomes and aids in strengthening the system's robustness.

Model	F_1 Score
AraBERTv02-Twitter-base	**77.03**
CAMeLBERT-DA	72.78
AraBERTv02-base	73.07
Ensemble - Hard Voting	**77.62**

Table 2: Results for Dialect Identification Task on the Development dataset

6 Results

This section discusses the results obtained by our system and analyses its performance. Table 2 and

Model	F_1 Score
AraBERTv02-Twitter-base	**75.17**
CAMeLBERT-DA	71.99
AraBERTv02-base	72.09
Ensemble - Hard Voting	**76.65**

Table 3: Results for Dialect Identification Task on the Test dataset

Table 3 depict our scores for the individual models used and the corresponding ensembled score on the development dataset and the test dataset respectively. The F_1 score is used as the official metric for scoring the systems.

AraBERTv02-Twitter-base outperforms the other models with an F_1 score of 77.03 on the development dataset and 75.17 on the test dataset. This performance demonstrates the benefits of utilizing a model that is pre-trained on a corpus similar to the one the task demands. AraBERTv02-Twitter-base is pre-trained on 60M multi-dialect tweets besides the usual datasets used for AraBERT models, giving it an edge over other models for this particular task. We select the ensemble-based system as our final approach since it produces outcomes with minimal variation and offers more stable predictions. This is justified by the superior performance of our system in the final evaluation stage. Our final system achieved an F_1 score of 76.65 on the test dataset.

7 Conclusion

This paper compares several transformer-based models on the task of Nuanced Arabic Dialect Identification (NADI). AraBERTv02-Twitter-base is found to outperform other models for this task. It achieves an F_1 score of 76.65. We use hard voting-based ensembling as the final approach for our system as it generates predictions that are stable while also improving the overall performance. With higher computational resources at hand, the performance of the system can be improved by training it for longer and by using bigger models for the system. Models that are specifically pretrained on data that is similar to the data used in the task at hand can help enhance understanding and in turn, give better performance. We can also experiment with other suitable ensembling methods and gauge their efficiency for our task.

Limitations

Models used for this task are computationally heavy and require significant computing resources for inference. As a result in certain real-world applications where there are compute constraints, using the system may pose a challenge. The data used for evaluation and pre-training of the models mentioned may have been biased even though the quality of the data used is high. Thus, it may not accurately represent real-world scenarios.

References

Muhammad Abdul-Mageed, AbdelRahim Elmadany, Chiyu Zhang, ElMoatez Billah Nagoudi, Houda Bouamor, and Nizar Habash. 2023. NADI 2023: The Fourth Nuanced Arabic Dialect Identification Shared Task. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*.

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, Barcelona, Spain (Online). Association for Computational Linguistics.

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, Houda Bouamor, and Nizar Habash. 2021. NADI 2021: The second nuanced Arabic dialect identification shared task. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 244–259, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Mahmoud Al-Ayyoub, Aya Nuseir, Kholoud Alsmearat, Yaser Jararweh, and Brij Gupta. 2018. Deep learning for arabic nlp: A survey. *Journal of Computational Science*, 26:522–531.

Badr AlKhamissi, Mohamed Gabr, Muhammad El-Nokrashy, and Khaled Essam. 2021. Adapting MARBERT for improved Arabic dialect identification: Submission to the NADI 2021 shared task. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 260–264, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Nouf AlShenaifi and Aqil Azmi. 2022. Arabic dialect identification using machine learning and transformer-based models: Submission to the NADI 2022 shared task. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 464–467, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. AraBERT: Transformer-based model for Arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France. European Language Resource Association.

Giyaseddin Bayrak and Abdul Majeed Issifu. 2022. Domain-adapted BERT-based models for nuanced Arabic dialect identification and tweet sentiment analysis. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 425–430, Abu Dhabi, United Arab Emirates (Hybrid). 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 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).

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.

Abdellah El Mekki, Ahmed Alami, Hamza Alami, Ahmed Khoumsi, and Ismail Berrada. 2020. Weighted combination of BERT and n-GRAM features for nuanced Arabic dialect identification. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 268–274, Barcelona, Spain (Online). Association for Computational Linguistics.

Kamel Gaanoun and Imade Benelallam. 2020. Arabic dialect identification: An Arabic-BERT model with data augmentation and ensembling strategy. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 275–281, Barcelona, Spain (Online). Association for Computational Linguistics.

Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2021. The interplay of variant, size, and task type in Arabic pre-trained language models. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 92–104, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Abdullah Salem Khered, Ingy Yasser Hassan Abdou Abdelhalim, and Riza Batista-Navarro. 2022. Building an ensemble of transformer models for Arabic dialect classification and sentiment analysis. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 479–484, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Moataz Mansour, Moustafa Tohamy, Zeyad Ezzat, and Marwan Torki. 2020. Arabic dialect identification using BERT fine-tuning. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 308–312, Barcelona, Spain (Online). Association for Computational Linguistics.

Ahmed Oumar and Khalil Mrini. 2022. Ahmed and khalil at NADI 2022: Transfer learning and addressing class imbalance for Arabic dialect identification and sentiment analysis. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 442–446, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Mohammad Salameh, Houda Bouamor, and Nizar Habash. 2018. Fine-grained Arabic dialect identification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1332–1344, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Abdulhadi Shoufan and Sumaya Alameri. 2015. Natural language processing for dialectical Arabic: A survey. In *Proceedings of the Second Workshop on Arabic Natural Language Processing*, pages 36–48, Beijing, China. Association for Computational Linguistics.

Rawan Tahssin, Youssef Kishk, and Marwan Torki. 2020. Identifying nuanced dialect for Arabic tweets with deep learning and reverse translation corpus extension system. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 288–294, Barcelona, Spain (Online). Association for Computational Linguistics.

ANLP-RG at NADI 2023 shared task: Machine Translation of Arabic Dialects: A Comparative Study of Transformer Models

Wiem Derouich and **Sameh Kchaou** and **Rahma Boujelbane**
ANLP Research Group, MIRACL Lab. FSEGS,
University of Sfax, Tunisia
wiemderwich123@gmail.com samehkchaou4@gmail.com rahma.boujelbane@gmail.com

Abstract

In this paper, we present our findings within the context of Subtask 2 of the NADI-2023 Shared Task. This task requires the exclusive utilization of the DIALECT-MSA MADAR Bouamor et al. (2018) corpus to develop sentence-level machine translations from Palestinian, Jordanian, Emirati, and Egyptian dialects to Modern Standard Arabic (MSA). However, MADAR lacks a parallel Emirati-MSA corpus. To address this challenge, we pre-trained the AraT5 transformer model using different configurations of the MADAR corpus and compared their performance results with those of existing transformer models. The best model achieved a BLEU score of 11.14% on the dev set and 10.02% on the test set.

1 Introduction

Arabic dialects (AD) represent a diverse range of informal languages spoken throughout Arab countries. The rise of social media has greatly amplified the growth of these dialects, which have become an integral part of everyday communication. Platforms such as Twitter, Facebook, and Instagram often feature user-generated content written in these dialects. Arabic dialects can indeed vary significantly from one region to another, and these vast regional differences make them challenging to understand and interpret. This linguistic diversity can be so pronounced that even within a single country, identical words might bear different meanings. As a result, due to the variation among these Arabic dialects (ADs), it becomes exceedingly challenging to create tools capable of accurately processing Arabic social media content. It can also be difficult to employ standard tools designed for Modern Standard Arabic (MSA), which serves as the mother language for these dialects. One solution to overcome this problem involves leveraging the richness of the MSA language by translating the dialect variants into it. Currently, there is a significant amount of work focused on translating dialects to MSA. However, most approaches treat each dialect separately, as seen in studies like Kchaou et al. (2022) for Tunisian and Al-Ibrahim and Duwairi (2020) for the Jordan dialect. Yet, it's important to acknowledge that these dialectal variations coexist in social networks. Therefore, it is important to develop models capable of handling the processing of all these dialects collectively. This work fits into this context by involving the development of a machine translation (MT) model to translate a subset of Arabic dialects, namely Palestinian and Levantine, into Modern Standard Arabic (MSA). As part of the competition offered by the NADI shared task, the challenge involves the development of a translation model for four dialects based solely on the MADAR corpus. It's worth noting that MADAR lacks a parallel corpus for the Emirati dialect. In this paper, we outline the experiments to build a dialect translation model. Specifically, we have compared the results of two methods: the first involves fine-tuning the AraT5 transformer model (Nagoudi et al., 2021) utilizing various corpus configurations from MADAR, while the second entails refining existing tools and employing the back-translation method. The rest of this paper is structured as follows: Section 2 outlines related works. Section 3 describes the dataset used. The fine-tuning of AraT5 models is presented in Section 4. We assess the benefits of leveraging tools to improve the translation process in Section 5. In Section 6, we discuss the results obtained. Finally, Section 7 provides a conclusive summary.

2 Related works

In the field of neural machine translation for Arabic dialects, the primary focus has been on translating these dialects into Modern Standard Arabic (MSA). However, most of these works typically concentrate on a single dialect, resulting in a lack of models that address the full spectrum of dialects. For instance, Al-Ibrahim and Duwairi (2020) conducted

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 683–689
December 7, 2023 ©2023 Association for Computational Linguistics

a study that focused on translating the Jordanian Arabic dialect into MSA using deep learning techniques and implemented an RNN encoder-decoder model. However, the size of the corpus limited their progress. Similarly, Baniata et al. (2018) tackled the challenge of translating Levantine dialects, including Jordanian, Syrian, and Palestinian, into MSA. They worked with a relatively small dataset of about 20,000 parallel sentences from the AD Applications and Resources (MADAR) and Parallel Arabic Dialect Corpus (PADIC) corpora. Their approach introduced an RNN-based multitask learning model in which the decoder was shared across language pairs, with each source language having its own encoder. A transductive transfer learning approach, introduced by Hamed et al. (2022), emerged in the context of low-resource neural machine translation for the Algerian Arabic dialect. This approach employed fine-tuned transfer learning to transfer knowledge from a parent model to a child model. The evaluation was carried out using the MADAR and PADIC corpus. This study applied the transductive transfer learning strategy with two NMT models: Seq2Seq and Attentional-Seq2Seq. Moreover, Nagoudi et al. (2022) introduced TURJUMAN, a versatile neural toolbox that can translate 20 languages into MSA. The TURJUMAN toolbox uses the power of the AraT5 model, renowned for its ability to decode Arabic. Notably, TURJUMAN allowed for flexibility in decoding methods, facilitating the creation of paraphrases for MSA translations. The tool was trained to use semantic similarity to collect publicly available parallel data samples to ensure data quality. This initiative resulted in the development and launch of AraOPUS-20, which establishes a new benchmark for machine translation. It encompasses a benchmark dataset (AraOPUS-20) and the translation toolkit (TURJUMAN). Another contribution comes from Kchaou et al. (2023), who introduced a hybrid approach to building a translation model for the Tunisian dialect. They proposed different augmentation methods to create a large corpus. Using this corpus, the authors tested different NMT models. The best model was obtained using JoeyNMT, achieving a BLEU score of 69.22

3 NADI-2023 Shared Task Subtask 2: DATASETS

In Subtask 2, we had access to three primary datasets: training (Train), development (Dev), and testing (Test), as outlined in Table 1. Our approach began with the utilization of the MADAR parallel corpus as our training set. During this phase, our model learned from the data and fine-tuned its parameters accordingly. Following the training, we evaluated our model's performance on the development set provided by the shared task. Finally, we generated translations using the best configuration on the test set. The subsequent sections will provide a detailed description of the contents within these three corpora.

Data set	**#Lines**
Train	111096
Dev	400
Test	2000

Table 1: Distribution of Different Sets.

3.1 Training SET

Subtask 2 allowed the usage of only the dataset from the MADAR parallel corpus for training. The statistics for the MADAR corpus for Subtask 2 are provided in Table 2.

Corpus	**#lines**	**#token**	**#vocabularies**
Tunisian	14k	87113	17102
Iraq	4k	2414	6466
Libya	4k	26209	6247
Morroco	14k	94289	18120
Syria	4k	6098	24363
SAUDI-ARABIA	4K	24751	6248
EGYPT	4k	26757	6239
JORDAN	42k	26074	6247
PALESTINIAN	2k	12574	3902
QATAR	12k	72878	12480
Yemen	2k	12823	4317
Algeria	2k	13198	4180
Lebanon	12k	72806	15531
Oman	2k	13201	4531
Sudan	2k	13352	4120

Table 2: Statistics of MADAR Subtask 2 Data Set.

3.2 Dev set

The development set comprises 400 sentences, with 100 sentences dedicated to each dialect. This dataset plays a crucial role in enhancing and evaluating translation systems, aiming for exceptional results. Each development tweet is accompanied by a unique identifier (#1_id) for each dialect, followed by the tweet's content (#2_content). The third column (#3_label) presents the tweet's gold label at the country-of-origin level.

3.3 Test set

The test set includes a total of 2,000 sentences, with an equal distribution across four different dialects: Egyptian, Emirati, Jordanian, and Palestinian. These tests have been thoughtfully designed to assess the capability of translation systems to effectively convert AD into MSA. Furthermore, each test tweet is accompanied by a unique identifier (#1_id) and specifies the dialect's name at the country-of-origin level (#2_dialect_id). [1]

4 Fine-Tuning AraT5 models

In light of the impressive performance showcased by the transformer architecture in Neural Machine Translation (NMT) of Arabic Dialects, as highlighted by Kchaou et al. (2023) and Nagoudi et al. (2022), our proposed strategy is to further harness this potential. Specifically, we intend to fine-tune the transformer AraT5 model using the MADAR corpus. This fine-tuning process is geared towards developing a specialized Machine Translation (MT) model capable of effectively handling the four dialects introduced for testing.

4.1 Architectures

In order to determine the most suitable AraT5 configuration for this task, we conducted fine-tuning on seven different architectures, including:

- The **AraT5 base** model by Abdul-Mageed et al. (2021): This model represents a modification of the T5 (Text-To-Text Transfer Transformer), finely tuned for the processing of Arabic text. It functions as a foundational model for various natural language processing tasks, encompassing text classification, text generation, and machine translation (MT). AraT5-base capitalizes on the Transformer architecture and pre-trained embeddings to effectively comprehend and generate Arabic text.

- The **AraT5v2-base-1024** model represents an enhanced iteration of AraT5-Base. In this version, the sequence length has been extended from 512 to 1024, denoted by the "1024". This expanded sequence length significantly augments the model's adaptability across various Natural Language Processing (NLP) tasks. Notably, the fine-tuning process of AraT5v2-base-1024 exhibits approximately 10 times faster convergence compared to its predecessor, AraT5-base. This accelerated convergence holds the potential to significantly expedite both the training and fine-tuning procedures, thereby enhancing overall efficiency. The selection of this model for our experiments was motivated by its exceptional performance, as demonstrated in Table 4, where it outperformed other models under the AraT5v2-Base category.

- The **Sultan-ArabicT5** model Alrowili and Shanker (2022) : It is another variant of the T5 model tailored for Arabic text processing. Similar to other T5-based models, Sultan-ArabicT5 is versatile and can be fine-tuned for a range of natural language processing (NLP) tasks. Specific features and details of this model may vary depending on the creator's objectives and training data.

- **AraT5-MSA-Small and AraT5-MSA-Base** Models Nagoudi et al. (2021): We evaluated two additional versions of the AraT5 model in our experiments: AraT5-MSA-Base and AraT5-MSA-Small, each tailored to meet specific requirements. The AraT5-MSA-Base is an upgraded AraT5 version that is well-equipped to handle a wide array of standard Arabic Natural Language Processing (NLP) tasks. It boasts a larger architecture and an increased number of parameters, making it particularly adept at intricate tasks that demand a deep understanding of the language. AraT5-MSA-Base is an excellent choice for research projects and applications that necessitate advanced linguistic modeling. AraT5-MSA-Small in contrast, is a streamlined iteration of the AraT5 model, optimized for efficient processing of MSA data. It operates at a faster pace and demands fewer computational resources compared to the "Base" version. This version is typically employed in applications where efficiency is a priority, without a significant loss in quality. The key distinction between these two models lies in their size and their suitability for various standard Arabic NLP tasks. AraT5-MSA-Small prioritizes speed and resource efficiency, while AraT5-MSA-Base excels in proficiency and

[1] https://github.com/Wiemder/Levantin-Dataset

versatility across a broader spectrum of standard Arabic NLP tasks.

- **AraT5-Tweet-Small and AraT5-Tweet-Base** Models: as presented by Nagoudi et al. (2021), AraT5-Tweet-Small and AraT5-Tweet-Base are specialized models meticulously crafted to tackle the unique linguistic challenges presented by social media content, particularly tweets and informal online discourse. These models are fine-tuned to specifically address the subtleties involved in translating Arabic dialects commonly found in user-generated content on platforms like Twitter. Their incorporation into our experiments equips us with the tools needed to effectively navigate the complexities associated with translating such content.

In specific scenarios, the transformer-based model "AraT5v2-base-1024" can indeed prove to be a valuable asset for traditional machine learning models. In our specific context, the proposed fine-tuning of AraT5 models offers several advantages. These pre-trained models can be further customized and optimized for specific Natural Language Processing (NLP) tasks, subsequently serving as input features or foundational models for various tasks within traditional machine learning. Transformer-based models, including AraT5, bring advanced capabilities for text preprocessing, encompassing tasks such as tokenization, embedding, and attention mechanisms. These preprocessing steps can be seamlessly integrated with traditional machine learning models that might lack such built-in capabilities. The fusion of predictions from a transformer-based model and a traditional machine learning model, often referred to as ensemble learning, frequently results in enhanced prediction accuracy. This is particularly valuable for tasks that necessitate the handling of both textual and structured data, creating a synergy that can lead to improved performance across a wide range of applications.

4.2 MADAR Configurations for AraT5 Fine-tuning

Our approach involved fine-tuning the aforementioned models with various configurations of the MADAR corpus. Initially, we conducted experiments using the entire corpus, and subsequently, we suggested the utilization of a subset of dialects from Palestine, Jordan, and Egypt. In a second phase, we incorporated dialects from geographically adjacent regions, namely Qatari and Saudi-Arabian dialects. All the models used in this research were sourced from the Hugging Face repository, and the experiments were designed and executed using the PyTorch Transformers library. To ensure consistency and comparability, we implemented the models with identical parameter settings, as outlined in Table 3. This standardized approach enabled us to make meaningful comparisons and draw reliable conclusions from our experiments. These parameters were carefully selected to achieve optimal performance while minimizing training time. They were carefully selected to achieve optimal performance while minimizing training time. The maximum length for the number is set at 128 characters, and the batch size parameter is configured for training with a value of 16. We carried out a single training epoch to compare the initial performance of the model across various experiments. The sequence length of 20 characters was determined based on the improvement of the results. A learning rate of 2e-5 was optimal to achieve fast convergence without the risk of overfitting. The weight decay is sustained at 0.01 to regulate model learning, and a save_total_limit of 3 is used to retain essential checkpoints during training. These parameters are pivotal in ensuring the reproducibility of our experiments and the accuracy of our results. Table 4 provides a comprehensive view of the BLEU scores obtained for diverse Arabic dialects (ADs) generated by a range of models and strategies. Notably, the MADAR corpus, in combination with the AraT5v2-base-1024 model, emerges as the top performer with an impressive overall BLEU score of 11.14. This underscores the critical importance of meticulous model selection in achieving optimal translation quality for specific Arabic dialects. Additionally, the variability in BLEU scores across different dialects suggests that certain models may exhibit superior performance for specific dialects, reinforcing the need for tailored approaches to enhance translation quality effectively.

5 Leveraging existing tools

To elevate the BLEU scores in our translation task, we pursued enhancements through the utilization of existing tools. Our approach unfolded in two key steps: Firstly, we showcased the efficacy of these tools in translating dialects into Modern Standard Arabic (MSA). Secondly, taking advantage of the

Parameters	Values
Max-length	128 Characters
Batch-size	16
Epoch	1
Seq-length	20
Learning-rate	2e5
Weight-decay	0.01
Sav-total-limit	3

Table 3: Parameters of the AraT5v2-base-1024 model

Corpus	Model	Overall	Egy	Emirate	Jord	Pales
TS MADAR	AraT5v2-base	11.14	10.58	8.11	10.04	11.38
TS MADAR	Sultan-ArabicT5	6.11	5.03	6.46	5.69	6.80
Egy-Pal-Jor	AraT5V2-Base	5.62	4.56	5.46	6.19	5.58
Egy-Pal-Jor-Qat-Ksa	AraT5V2-1024	7.02	6.51	6.16	8	6.38
Aug-ALE	AraT5V2-Base	9.52	10.66	6.88	8.76	8.95
Aug-ALE-ALX-JER	AraT5V2-Base	9.40	10.66	6.88	8.76	8.95
Aug-PaysGolf	AraT5V2-Base	6.09	5.40	7.28	4.88	5.88

Table 4: Bleu scores on the Dev set of the proposed configurations.

broader availability of Dialectal Arabic (DA) to English translation tools, we introduced the back translation method. This technique involves using English as an intermediary language between Dialectal Arabic and MSA, contributing to improved translation quality.

5.1 Direct-Translation

In our research, we leveraged the capabilities of TURJUMAN, a robust neural machine translation system designed not only for Modern Standard Arabic (MSA) but also for 20 other languages[2]. To optimize its performance, we carefully fine-tuned TURJUMAN with unique configurations. These included setting "search_method" to "beam," "seq_length" to 20, "num_beams" to 5, "no_repeat_ngram_size" to 3, and "max_outputs" to 1. These distinctive parameter choices allowed us to generate a fresh batch of MSA texts, resulting in a substantial improvement in BLEU scores, as depicted in table 5. Furthermore, we conducted experiments to explore the impact of increasing the value of max_outputs" to 3, thereby generating three distinct MSA texts. Remarkably, these experiments revealed no significant variation in BLEU scores among the different texts. Additionally, we experimented with the dl-translate 0.3.0 library[3], designed for text translation. Unfortunately, our evaluation using BLEU scores indicated that the quality of the generated texts fell below our expectations. These findings underscore the critical importance of tool selection and configuration in optimizing translation quality and ultimately enhancing BLEU scores in our research.

	Models	Overall	Egypt	Emirati	Jordan	Palestinian
Back translation	GoogleTranslator	10.98	12.12	7.73	9.54	12.18
	PonsTranslator	10.87	12.15	7.72	9.69	11.71
Direct-Translation	TURJUMAN	10.08	11.18	12.99	7.98	8.50

Table 5: Bleu Scores on the Dev set using existing transformer-based tools.

5.2 Back-Translation

In implementing the back-translation technique, following the approach by Hoang et al. (2018), we employed English as the intermediary language. This method was executed using the deep-translator library, which incorporates both the PonsTranslator and Google-translator models. As demonstrated in Table 5, this approach led to improvements in BLEU scores. Additionally, we re-utilized the dL-Translate 0.3.0 library for back-translation. This process entails the generation of English texts, followed by the subsequent back-translation into MSA. We applied this method to both transformer models offered in dl-translate 0.3.0: "nllb-200" and "m2m100." These efforts contributed to the enhancement of our translation quality and the corresponding BLEU scores, demonstrating the effectiveness of back-translation techniques in our research.

6 Discussion

The results we obtained exhibit a range of variations, encompassing both positive and negative outcomes. Our performance curve for the implemented strategies demonstrates fluctuations, highlighting the complexity of the translation task. It's important to note that the table of BLEU scores does not include certain results, such as those for AraT5-tweet-base, which received a NADI BLEU score of 0 and an overall BLEU score of 0.28. In contrast, the highest BLEU score of 11.14 on the dev set was achieved through the fine-tuning of AraT5V2-base-1024. Tis con

Moreover, upon analyzing the use of DeepL Translate and dl-translate 0.3.0, we observed that the models from the DeepL Translate library outperformed those from the dl-translate library. These models exhibited the potential to contribute to enhancing our corpus, resulting in higher MSA scores. In contrast, the results obtained from

[2] https://github.com/UBC-NLP/turjuman

[3] https://github.com/xhluca/dl-translate

the dl-translate library were significantly lower. Moreover, our experiments with data augmentation yielded limited benefits, potentially due to the extensive diversity of Arabic written forms. Table 6 provides examples generated using the augmentation method, offering insights into the outcomes of this approach. The relatively lower success rates observed in our experiments can be attributed to several factors. These include limitations within the corpus, notably the absence of the Emirati dialect in the parallel corpus. Additionally, the vocabulary and the quantity of comments available within the MADAR dataset may have played a role in influencing the results. These factors collectively contribute to the challenges associated with achieving higher success rates in dialect translation tasks. Addressing these limitations and enhancing the availability of diverse and comprehensive datasets could potentially lead to improved translation outcomes in the future.

ALX-MADAR-Corpus	Aug1-ALX	Aug2-ALX
مّكن بدف الِك متِا الي بميتين دولَار Can my $200 check be cashed?	دَا مّكن تكدر تدفء بقية اليك الِ بميتين ٥٠٠ دولَار dear Can my $500 check be 200 cashed	مّكن بس الي بَالّون ا اهمر الي بميتين دولَار please dear Can my $200 check red be cashed !

Table 6: Example of generated sentence using augmentation method.

7 Conclusion

Our work constitutes an integral contribution to the NADI2023 shared task, which centers around the machine translation of Arabic dialects (AD) into Modern Standard Arabic (MSA) using the MADAR corpus. Throughout our research, we explored a multitude of methods and strategies aimed at tackling this complex and challenging task. Our efforts yielded two notable strengths. Firstly, fine-tuning models, with a particular focus on the "AraT5v2-base-1024" model, emerged as an effective approach for enhancing translation quality. Additionally, we achieved commendable results by leveraging existing translation tools, especially the Google Translator model, coupled with back-translation methods. These outcomes underscore the relevance and practicality of these approaches for translating Arabic dialects. In fact, these results open up the possibility of utilizing these methods to automate the parallel corpus construction process. Furthermore, we are dedicated to furthering our research efforts by delving into additional fine-tuning techniques for transformer models.

References

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, Houda Bouamor, and Nizar Habash. 2021. NADI 2021: The second nuanced Arabic dialect identification shared task. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 244–259, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

R. Al-Ibrahim and R. M. Duwairi. 2020. Neural machine translation from jordanian dialect to modern standard arabic. In *2020 11th International Conference on Information and Communication Systems (ICICS)*.

Sultan Alrowili and Vijay Shanker. 2022. Generative approach for gender-rewriting task with ArabicT5. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 491–495, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Laith Baniata, Seyoung Park, and Seong-Bae Park. 2018. A neural machine translation model for arabic dialects that utilizes multitask learning (mtl). In *Computational Intelligence and Neuroscience.*

Houda Bouamor, Nizar Habash, Mohammad Salameh, Wajdi Zaghouani, Owen Rambow, Dana Abdulrahim, Ossama Obeid, Salam Khalifa, Fadhl Eryani, Alexander Erdmann, et al. 2018. The madar arabic dialect corpus and lexicon. In *Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018).*

Injy Hamed, Nizar Habash, Slim Abdennadher, and Ngoc Thang Vu. 2022. Investigating lexical replacements for arabic-english code-switched data augmentation.

Cong Duy Vu Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *2nd Workshop on Neural Machine Translation and Generation*, pages 18–24. Association for Computational Linguistics.

Saméh Kchaou, Rahma Boujelbane, Emna Fsih, and Lamia Hadrich-Belguith. 2022. Standardisation of dialect comments in social networks in view of sentiment analysis : Case of Tunisian dialect. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5436–5443, Marseille, France. European Language Resources Association.

Saméh Kchaou, Rahma Boujelbane, and Lamia Hadrich. 2023. Hybrid pipeline for building arabic tunisian dialect-standard arabic neural machine translation model from scratch. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 22(3).

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.*

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)*, Marseille, France. European Language Resource Association.

Qur'an QA 2023 Shared Task: Overview of Passage Retrieval and Reading Comprehension Tasks over the Holy Qur'an

Rana Malhas
Qatar University
rana.malhas@qu.edu.qa

Watheq Mansour
The University of Queensland*
w.mansour@uq.edu.au

Tamer Elsayed
Qatar University
telsayed@qu.edu.qa

Abstract

Motivated by the need for intelligent question answering (QA) systems on the Holy Qur'an and the success of the first Qur'an Question Answering shared task (Qur'an QA 2022 at OSACT 2022), we have organized the second version at ArabicNLP 2023. The Qur'an QA 2023 is composed of two sub-tasks: the passage retrieval (PR) task and the machine reading comprehension (MRC) task. The main aim of the shared task is to encourage state-of-the-art research on Arabic PR and MRC on the Holy Qur'an. Our shared task has attracted 9 teams to submit 22 runs for the PR task, and 6 teams to submit 17 runs for the MRC task. In this paper, we present an overview of the task and provide an outline of the approaches employed by the participating teams in both sub-tasks.

1 Introduction

The timeless and sacred Qur'an will never cease to attract the interest and inquisition of millions of Muslims and non-Muslims for its profound teachings, legislation, and fertile knowledge. Such inquisitions may be driven by learning, curiosity, or skepticism. The Qur'an is composed of 114 Surahs and 6,236 verses (Ayas) of different lengths, with a total of about 80k words. The words, revealed more than 1,400 years ago, are in Classical Arabic.

Extractive question answering (QA) approaches are being formulated in the literature as machine reading comprehension (MRC) tasks (Chen, 2018). Given a passage of text, a system is evaluated based on its ability to correctly answer a set of questions over the given text. We believe that the resurgence of the MRC field should be harnessed to address the timeless interest in the Holy Qur'an and the information needs of its inquisitors and knowledge seekers (Bashir et al., 2022). This has motivated the inception of the first Qur'an Question Answering shared task, Qur'an QA 2022 at OSACT 2022 Workshop (Malhas et al., 2022).

Although MRC systems are relieved from the task of passage retrieval (i.e., the task of retrieving candidate passages that potentially contain answers to a given question) to purely focus on inference and reasoning for answer extraction, the retriever component remains an integral contributor to the performance of end-to-end extractive QA systems that adopt a retriever-reader architecture (Zhu et al., 2021). Prevalent search/retrieval systems on the Holy Qur'an are either keyword-based, semantic-based, or a hybrid of both paradigms. Semantic-based approaches are predominantly ontology-based with almost no use of state-of-the-art approaches such as dense retrieval (Karpukhin et al., 2020), generative retrieval (Santos et al., 2020) and beyond (Malhas, 2023) to our knowledge.

To this end, and to build on the success of the first Qur'an QA 2022 shared task (Malhas et al., 2022), we have organized the second Qur'an QA shared task (Qur'an QA 2023) at ArabicNLP 2023. Qur'an QA 2023 comprises a Qur'anic Passage Retrieval (PR) task and a Machine Reading Comprehension (MRC) task. The PR task aims at finding all Qur'anic passages that have potential answers for a given question that is posed in Modern Standard Arabic (MSA). Whereas the MRC task targets the extraction of all answers to a given question from a given qur'anic passage. Each answer must be a *span* of text extracted from the given passage. To make both tasks more challenging, we include questions that have no answers in the Qur'an. Further details about the two tasks are provided in Sections 3 and 4, respectively.

To encourage quality participation in the task, we allotted five awards. The awards for the best and second-best teams in each task are \$300 and \$200, respectively, provided that their papers are accepted at the conference. The fifth award is \$150 allotted for the best paper.

*Part of the work on this paper was done while being at Qatar University.

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 690–701
December 7, 2023 ©2023 Association for Computational Linguistics

السؤال: من هم الملائكة المذكورون في القرآن؟

Question: Who are the angels mentioned in Qur'an?

Gold Qur'anic Passages الفقرات القرآنية الذهبية

وَلَقَدْ ءَاتَيْنَا مُوسَى ٱلْكِتَٰبَ وَقَفَّيْنَا مِنۢ بَعْدِهِۦ بِٱلرُّسُلِ وَءَاتَيْنَا عِيسَى ٱبْنَ مَرْيَمَ ٱلْبَيِّنَٰتِ وَأَيَّدْنَٰهُ بِرُوحِ ٱلْقُدُسِ أَفَكُلَّمَا
جَآءَكُمْ رَسُولٌۢ بِمَا لَا تَهْوَىٰٓ أَنفُسُكُمُ ٱسْتَكْبَرْتُمْ فَفَرِيقًا كَذَّبْتُمْ وَفَرِيقًا تَقْتُلُونَ. وَقَالُوا۟ قُلُوبُنَا غُلْفٌۢ بَل لَّعَنَهُمُ ٱللَّهُ
بِكُفْرِهِمْ فَقَلِيلًا مَّا يُؤْمِنُونَ.

قُلْ مَن كَانَ عَدُوًّا لِّجِبْرِيلَ فَإِنَّهُۥ نَزَّلَهُۥ عَلَىٰ قَلْبِكَ بِإِذْنِ ٱللَّهِ مُصَدِّقًا لِّمَا بَيْنَ يَدَيْهِ وَهُدًى وَبُشْرَىٰ لِلْمُؤْمِنِينَ. مَن كَانَ عَدُوًّا
لِّلَّهِ وَمَلَٰٓئِكَتِهِۦ وَرُسُلِهِۦ وَجِبْرِيلَ وَمِيكَىٰلَ فَإِنَّ ٱللَّهَ عَدُوٌّ لِّلْكَٰفِرِينَ. وَلَقَدْ أَنزَلْنَآ إِلَيْكَ ءَايَٰتٍۭ بَيِّنَٰتٍ وَمَا يَكْفُرُ بِهَآ إِلَّا
ٱلْفَٰسِقُونَ. أَوَكُلَّمَا عَٰهَدُوا۟ عَهْدًا نَّبَذَهُۥ فَرِيقٌ مِّنْهُم بَلْ أَكْثَرُهُمْ لَا يُؤْمِنُونَ. وَلَمَّا جَآءَهُمْ رَسُولٌ مِّنْ عِندِ ٱللَّهِ مُصَدِّقٌ لِّمَا
مَعَهُمْ نَبَذَ فَرِيقٌ مِّنَ ٱلَّذِينَ أُوتُوا۟ ٱلْكِتَٰبَ كِتَٰبَ ٱللَّهِ وَرَآءَ ظُهُورِهِمْ كَأَنَّهُمْ لَا يَعْلَمُونَ.

وَٱتَّبَعُوا۟ مَا تَتْلُوا۟ ٱلشَّيَٰطِينُ عَلَىٰ مُلْكِ سُلَيْمَٰنَ وَمَا كَفَرَ سُلَيْمَٰنُ وَلَٰكِنَّ ٱلشَّيَٰطِينَ كَفَرُوا۟ يُعَلِّمُونَ ٱلنَّاسَ ٱلسِّحْرَ وَمَآ
أُنزِلَ عَلَى ٱلْمَلَكَيْنِ بِبَابِلَ هَٰرُوتَ وَمَٰرُوتَ وَمَا يُعَلِّمَانِ مِنْ أَحَدٍ حَتَّىٰ يَقُولَآ إِنَّمَا نَحْنُ فِتْنَةٌ فَلَا تَكْفُرْ فَيَتَعَلَّمُونَ مِنْهُمَا مَا
يُفَرِّقُونَ بِهِۦ بَيْنَ ٱلْمَرْءِ وَزَوْجِهِۦ وَمَا هُم بِضَآرِّينَ بِهِۦ مِنْ أَحَدٍ إِلَّا بِإِذْنِ ٱللَّهِ وَيَتَعَلَّمُونَ مَا يَضُرُّهُمْ وَلَا يَنفَعُهُمْ وَلَقَدْ
عَلِمُوا۟ لَمَنِ ٱشْتَرَىٰهُ مَا لَهُۥ فِى ٱلْءَاخِرَةِ مِنْ خَلَٰقٍ وَلَبِئْسَ مَا شَرَوْا۟ بِهِۦٓ أَنفُسَهُمْ لَوْ كَانُوا۟ يَعْلَمُونَ. وَلَوْ أَنَّهُمْ ءَامَنُوا۟ وَٱتَّقَوْا۟
لَمَثُوبَةٌ مِّنْ عِندِ ٱللَّهِ خَيْرٌ لَّوْ كَانُوا۟ يَعْلَمُونَ.

...

Figure 1: An example for the PR task: a factoid question with some of its gold (answer-bearing) Qur'anic passages. Answers are highlighted in each passage.

Qur'an QA 2023[1] has attracted 38 and 29 teams to sign up for the PR Task and the MRC Task, respectively. In the final phase, 9 teams participated in the PR task with 22 run submissions, and 6 teams participated in the MRC task with 17 run submissions. Table 1 lists the participating teams per task with their affiliations and team size. Six of them have accepted system description papers as referenced in the table.

The rest of the paper is organized as follows. Section 2 outlines the first version of Qur'an QA. Sections 3 and 4 discuss the PR and MRC tasks, respectively, in detail including the task descriptions, datasets, evaluation setups, results, and analysis of approaches employed by the participating teams. We conclude with final thoughts in Section 5.

2 The Qur'an QA 2022 Shared Task

The Qur'an QA shared task in its first version in 2022[2] (Malhas et al., 2022) only comprised an MRC task that is similar to the MRC task proposed this year, but it was relatively simplified. It was defined as follows: given a Qur'anic passage that consists of consecutive verses in a specific Surah of the Holy Qur'an and a question posed in MSA over that passage, a system is required to extract *any* correct answer *span* to that question (regardless if the question had more than one answer in that passage or only one answer). As such, the main measure used in the performance evaluation of participating systems was partial Reciprocal Rank (pRR) (Malhas and Elsayed, 2020).

Qur'an QA 2022 has attracted 30 teams to sign up for the task. In the final phase, 13 teams participated, with a total of 30 submitted runs on the test set. Ten out of the thirteen teams submitted system description papers, which were peer-reviewed and published in OSACT 2022 (Al-Khalifa et al., 2022).

3 Task A: Passage Retrieval (PR)

In this section, we define the PR task, introduce the dataset, and elaborate on the evaluation setup and teams' results. We conclude this section with an overview of the main methods employed by the participating teams.

3.1 Task Description

The task is defined as follows: Given a free-text question posed in MSA and a collection of passages that cover the Holy Qur'an, the system is required to return a *ranked list* of up to 10 answer-bearing passages (i.e., passages that potentially enclose all the answers to the given question) from this collection. The question can be factoid or non-factoid. An example question is shown in Figure 1.

To make the task more realistic (thus challenging), some questions may not have an answer in the Holy Qur'an. We call them *zero-answer* questions. In such cases, the ideal system should return no

[1] https://sites.google.com/view/quran-qa-2023

[2] https://sites.google.com/view/quran-qa-2022

Team	Tasks	Size	Affiliations
Al-Jawaab (Zekiye and Amroush, 2023)	A, B	2	Koç University, Niuversity
AHJL (Alawwad et al., 2023)	A	4	King Abdulaziz University, Saudi Electronic University, Imam Mohammad Ibn Saud Islamic University (IMSIU), King Saud University
GYM (Mahmoudi and Morshedzadeh, 2023)	A, B	2	Iran University of Science and Technology, University of British Columbia
LKAU23 (Alnefaie et al., 2023)	A, B	5	University of Leeds, King Abdulaziz University
LowResContextQA (Veeramani and Roy, 2023)	B	2	University of California, Los Angeles (UCLA), UoSC
PSUT	A, B	5	Princess Sumaya University for Technology
sabran	A	1	Independent
SSZ	A	3	Qatar University
TCE (Elkomy and Sarhan, 2023)	A, B	2	Tanta University
TERROR	A	1	Helwan University

Table 1: Participating teams in Qur'an QA 2023.

Dataset	%	# Questions	QP Pairs
Training	70%	174	972
Development	10%	25	160
Test	20%	52	427
All	100%	251	1,599

Table 2: Distribution of questions and question-passage (QP) pairs in the PR dataset (AyaTECv1.2)

answers; otherwise, it returns a ranked list of the answer-bearing passages.

3.2 PR Dataset

In this section, we introduce the dataset/test collection used in the PR task. In general, a *test collection* is typically composed of a document collection[3] (the Holy Qur'an passages in our case), a set of queries (questions), and their relevance judgments (Lin and Katz, 2006) (i.e., the gold answers, or the passages that comprise them in our case).

For the PR task, an extended version of the *AyaTEC* dataset/test collection (Malhas and Elsayed, 2020) was used (AyaTECv1.2).[4] It is composed of the Qur'anic Passage Collection (QPC) (Malhas, 2023; Swar, 2007), an augmented set of AyaTEC's original questions (AyaTECv1.1), and their relevance judgments (i.e., the answer-bearing passages for each question).

The QPC was developed by topically segmenting the 114 Qur'anic Surahs of different lengths using the Thematic Holy Qur'an (Swar, 2007),[5] which is a printed edition that clusters the verses of each Surah into topics. This segmentation resulted in a total of 1,266 topical passages.

As for the set of questions, 199 out of the original 207 questions of the AyaTECv1.1 test collection (Malhas and Elsayed, 2020) were used. This set was augmented with 52 new questions for evaluating the systems in the PR task. Overall, we have included a total of 37 zero-answer questions (about 15%) that do not have an answer in the Holy Qur'an. The distribution of the training (70%), development (10%), and test (20%) splits are exhibited in Table 2.

For the additional 52 questions, we adopted the same verse-based answer extraction/annotation methodology used while developing the original AyaTEC dataset. The extraction of potential verse-based answers was conducted by two annotators who are knowledgeable about the Qur'an, while the annotation was conducted by three Qur'an specialists. Further details about the annotation process are provided in (Malhas and Elsayed, 2020). Developing the relevance judgments of the final set of questions over the QPC were generated automatically using the same methodology adopted by Malhas (2023). Each Qur'anic passage in the collection is considered relevant to the question if it happens to comprise any of the gold verse-based answer(s) completely or partially.

[3] In information retrieval, the term "document collection" or "collection" refers to a corpus or dataset (Yates et al., 2021); we use these terms interchangeably.

[4] https://gitlab.com/bigirqu/quran-qa-2023

[5] https://surahquran.com/tafseel-quran.html

3.3 Evaluation Setup

In this section, we shed light on the setup and methodology followed in evaluating the performance of participating systems.

3.3.1 Leaderboard and Repository

The leaderboard for both the PR and MRC tasks was hosted on CodaLab (Pavao et al., 2023) to allow participants to evaluate their runs and facilitate benchmarking. A participating team is required to submit their results/answers in one file, denoted as a “run file” or a “run” in short. The run should match TREC run format, i.e., having the following columns: ["question-id", "Q0", "passage-id", "rank", "score", "tag"]. Each team is allowed to submit 30 runs on the dev set, but up to 3 runs on the test set. Each run typically constitutes the results of a different system or a model.

To facilitate checking and evaluating runs before their submittal to the leaderboard, we made the submission-checker and evaluation scripts publicly available through the official repository of the shared task.[6] Furthermore, to give participants a reference point over the leaderboard, we opted for BM25 (a simple, yet very common, classical lexical-based retrieval model) as a baseline, and released the code to the same repository.

3.3.2 Evaluation Measures

As the PR task is a classical *ranked retrieval* task, we adopt Mean Average Precision at depth 10 (MAP@10) as the main official evaluation measure. We also report the Mean Reciprocal Rank at depth 10 (MRR@10) to measure the performance of retrieving *any* answer-bearing passage. The no-answer cases are handled simply by giving full credit to “no answer” system output, and zero otherwise, in both measures.

3.4 Results

Thirty eight teams registered for the PR task. Among these teams, nine participated in the final (test) phase and submitted 22 runs. The teams are officially ranked based on their best performing submitted run. Table 3 demonstrates the performance of all submitted runs in the test phase ranked by MAP@10.

We note that 8 runs from 3 teams outperformed the baseline, whereas the rest were below it. The highest scores of MAP@10 and MRR@10 are *0.2506* and *0.4610*; both were achieved by the *TCE* team (Elkomy and Sarhan, 2023). Figure 4 (in the Appendix) shows the boxplots for all submitted runs on the test queries (questions) to illustrate the performance distribution. The boxplots reveal the diverse performance across the questions for most of the runs.

Team	Run	MAP@10	MRR@10
TCE	M00	**0.2506**	**0.4610**
TCE	A00	0.2464	0.4940
TCE	C00	0.2302	0.4706
AHJL	SG2	0.1995	0.3889
AHJL	SWOP3	0.1318	0.3021
LKAU23	run63	0.1242	0.3750
AHJL	SS1	0.1202	0.2907
LKAU23	run61	0.1166	0.3632
Baseline	*BM25*	0.0904	0.2260
SSZ	run02	0.0804	0.2177
TERROR	new01	0.0789	0.1608
SSZ	un01	0.0784	0.2206
TERROR	new03	0.0739	0.1566
LKAU23	run62	0.0701	0.2047
Al-Jawaab	test	0.0643	0.1609
GYM	GRun1	0.0545	0.1581
TERROR	new02	0.0327	0.0737
GYM	Run0	0.0315	0.1023
PSUT	run3	0.0214	0.0752
GYM	Run2	0.0116	0.0356
PSUT	run2	0.0114	0.0523
sabran	vers01	0.0000	0.0000
Al-Jawaab	trem02	0.0000	0.0000

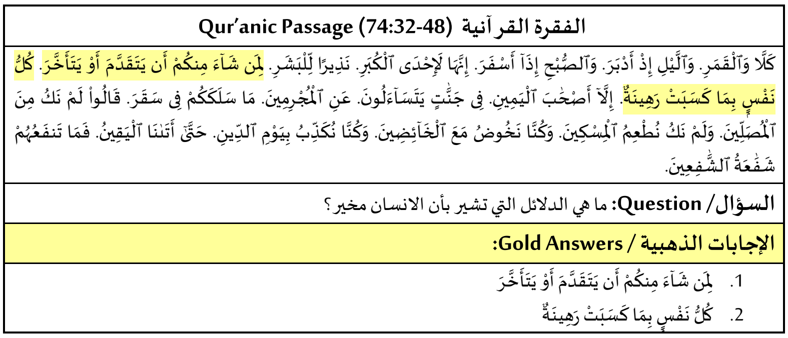
Table 3: PR evaluation results of all submitted runs ranked by MAP. The team name is removed from the run name to save space. The underlined rows are the median runs.

3.5 Methods and Analysis

In this section, we give an overview of the main approaches adopted by the 9 participating teams in their submitted runs on the test set. We do that in the context of highlighting some of our perceptions and general trends that characterize the participating systems and their submitted runs.

As expected, all systems utilized pre-trained transformer-based Language Models (LMs), two of which used generative (decoder-only) LMs (e.g., GPT), while the remaining systems employed encoder-only LMs (e.g., BERT). The majority of the semantic search/retrieval systems used bi-encoder and cross-encoder architectures either in-

[6] https://gitlab.com/bigirqu/quran-qa-2023



Qur'anic Passage (74:32-48) الفقرة القرآنية

كَلَّا وَٱلْقَمَرِ. وَٱلَّيْلِ إِذْ أَدْبَرَ. وَٱلصُّبْحِ إِذَآ أَسْفَرَ. إِنَّهَا لَإِحْدَى ٱلْكُبَرِ. نَذِيرًا لِّلْبَشَرِ. لِمَن شَآءَ مِنكُمْ أَن يَتَقَدَّمَ أَوْ يَتَأَخَّرَ. كُلُّ
نَفْسٍ بِمَا كَسَبَتْ رَهِينَةٌ. إِلَّآ أَصْحَٰبَ ٱلْيَمِينِ. فِى جَنَّٰتٍ يَتَسَآءَلُونَ. عَنِ ٱلْمُجْرِمِينَ. مَا سَلَكَكُمْ فِى سَقَرَ. قَالُوا۟ لَمْ نَكُ مِنَ
ٱلْمُصَلِّينَ. وَلَمْ نَكُ نُطْعِمُ ٱلْمِسْكِينَ. وَكُنَّا نَخُوضُ مَعَ ٱلْخَآئِضِينَ. وَكُنَّا نُكَذِّبُ بِيَوْمِ ٱلدِّينِ. حَتَّىٰٓ أَتَىٰنَا ٱلْيَقِينُ. فَمَا تَنفَعُهُمْ
شَفَٰعَةُ ٱلشَّٰفِعِينَ.

السؤال/ Question: ما هي الدلائل التي تشير بأن الانسان مخير؟

الإجابات الذهبية / Gold Answers:

1. لِمَن شَآءَ مِنكُمْ أَن يَتَقَدَّمَ أَوْ يَتَأَخَّرَ
2. كُلُّ نَفْسٍ بِمَا كَسَبَتْ رَهِينَةٌ

Figure 2: An example of the MRC task: a non-factoid question with the answers highlighted in the given passage.

dependently or jointly. Also, ensemble and self-ensemble approaches were employed by many systems to stabilize prediction fluctuations across runs and/or to enhance prediction accuracy through the wisdom of the crowd. For zero-answer questions, the majority of the systems did not explicitly address this challenge.

The three run submissions of the TCE team (Elkomy and Sarhan, 2023) outperformed all other submissions to the PR task (Table 3). TCE's three systems leveraged transfer learning and ensemble learning while training their dual-encoders (bi-encoders) and cross-encoders for ad hoc search (Yates et al., 2021). Their top performing system (with a MAP score of 0.2505) employed an ensemble of CAMel_BERT-CA (Inoue et al., 2021) and AraBERTv0.2-base (Antoun et al., 2020) dual encoder. Each of these encoders was self-ensembled and fine-tuned using three datasets; namely, the Arabic part of the multilingual TyDiQA dataset (Ar_TyDiQA) (Clark et al., 2020), followed by a Tafseer dataset [7] and finally the Task A dataset (AyaTECv1.2). Their second and third best systems employed the self-ensembled AraBERTv0.2-base and CAMel_BERT-CA encoders, respectively. For zero-answer questions, TCE adopted a thresholding mechanism to identify questions with a low cumulative likelihood of having answers in the Holy Qur'an. However, the threshold value should have been tuned rather than being set to approximately equal the percentage of zero-answer questions in AyaTECv1.2 training and development datasets.

The second-best ranked team (AHJL) (Alawwad et al., 2023) employed two semantic search models that were equipped with a translation module to translate a given question to English prior to performing the search. As such, an English translation of the meanings of the Qur'an was used. Given a translated question, the retriever module employs a bi-encoder to retrieve relevant passage candidates, then a cross-encoder is employed as a re-ranker. A zero-shot training setting was adopted. The OpenAI embeddings-based (OpenAI, 2023c) semantic search model (with translation) was their best-performing system (attaining a MAP score of 0.1995) while being able to successfully identify more than half of the zero-answer questions in the test set. OpenAI's best embeddings 'text-embedding-ada-002' model (OpenAI, 2023b) was employed as the bi-encoder (for primary search) and OpenAI's 'text-davinci-003' model as the cross-encoder (for re-ranking). Their second best performing system adopted the SBERT API (Reimers, 2023) that adopts the Sentence-BERT architecture (Reimers and Gurevych, 2019) with translation as well. It employed the 'msmarco-distilbert-base-tas-b' (Reimers, 2023) sentence transformer model as the bi-encoder and 'msmarco-MiniLM-L-6-v2' (Reimers, 2023) as the cross-encoder. We note that Al-Jawaab team also employed a bi-encoder architecture using the same OpenAI's embeddings used by the AHJL team for their bi-encoder, but their MAP score attained a below-median score of 0.0643.

The third ranked team (LKAU23) (Alnefaie et al., 2023) also adopted the Sentence-BERT architecture for their four Arabic pre-trained LMs (bi-encoders) fine-tuned using AyaTECv1.2 and QRCDv1.2 datasets, respectively. Their best-performing model was CL-AraBERT (Malhas and Elsayed, 2022) which attained a better MAP score (0.1242) than that of ArabicBERT (Safaya et al., 2020), CAMeL-BERT (Inoue et al., 2021), and AraBERT (Antoun et al., 2020). Their second performing model was an ensemble of ArabicBERT

[7] Interpretation resources (Tafseer) from Al-Muyassar and Al-Jalalayn were obtained from Tanzil `https://tanzil.net/docs/resources`.

and CL-AraBERT that attained a MAP score of 0.1166, which is still better than the median and baseline scores (Table 3).

Among the remaining submitted runs that attained near-median MAP scores (but below the baseline score) belonged to the SSZ and the TERROR teams. The GYM team (Mahmoudi and Morshedzadeh, 2023) attained a below-median MAP score of 0.0545 despite their deployment of an interesting approach that leveraged *unsupervised* fine-tuning of sentence bi-encoders using Transformer-based Sequential Denoising Auto Encoders (TSDAE) (Wang et al., 2021) and Simple Contrastive Learning of Sentence Embeddings (SimCSE) (Gao et al., 2021). The bi-encoder is then fine-tuned using a multi-task learning approach. Their best-performing run employed an AraBERT bi-encoder fine-tuned using the QPC dataset with the TSDAE unsupervised method. Then, it was fine-tuned using Mr. TyDi's Arabic dataset (Zhang et al., 2021) and the Qur'an-passage pairs of AyaTECv1.2 with a multi-task learning approach.

4 Task B: Machine Reading Comprehension (MRC)

In this section, we define the MRC task, present the dataset, and detail the evaluation methodology and results. We conclude with an overview of the main methods employed by the participating teams.

4.1 Task Description

The task is defined as follows: Given a Qur'anic passage that consists of consecutive verses in a specific Surah of the Holy Qur'an, and a free-text question posed in MSA over that passage, a system is required to extract *all* answers to that question that are stated in the given passage (rather than *any* answer as in Qur'an QA 2022). Each answer must be a *span* of text extracted from the given passage. If a question has only one answer in the given passage, it is considered a *single-answer* question, whereas if the question's answer is composed of more than one component/span in the accompanying passage, then the question is considered a *multi-answer* question. We note that the question can be a factoid or non-factoid question. An example is shown in Figure 2.

To make the task more realistic (thus challenging), some questions may not have an answer in the given passage. In such cases, the ideal system should return no answers; otherwise, it returns a ranked list of up to 10 answer spans.

Question	QP Pairs				QPA
Type	Train	Dev	Test	All	Triplets
Multi-answer	134 (14%)	29 (18%)	62 (15%)	224 (14%)	552 (29%)
Single-Answer	806 (81%)	124 (76%)	331 (81%)	1,261 (81%)	1,261 (67%)
Zero-Answer	52 (5%)	10 (6%)	14 (4%)	76 (5%)	76 (4%)
All	992	163	407	1,562	1,889

Table 4: Distribution of question-passage (QP) pairs and question-passage-answer (QPA) triplets by question type in the dataset of Task B (QRCDv1.2)

4.2 MRC Dataset

For the MRC task, an extended version of the Qur'anic Reading Comprehension Dataset (QRCD) (Malhas and Elsayed, 2022) was used (QRCDv1.2). It is composed of the original 1,093 question-passage (QP) pairs in QRCDv1.1, and an augmented set of 62 QP pairs whose questions have no answer in the accompanying passages (nor in the Holy Qur'an). These additional zero-answer questions were paired with hard negative passages retrieved using a BM25 retrieval model. We chose not to pair hard negative passages with the original (single-answer and multi-answer) questions so as not to contaminate the QRCD dataset with non-answer-bearing passages to questions which the Holy Qur'an does have an answer for.

To evaluate the systems in the MRC task, 407 additional QP pairs were included in QRCD, whose questions are the same new questions introduced to the dataset of the PR task (AyaTECv1.2) in Section 3.2 above. Fourteen (14) out of the 407 QP pairs have no answer in the Holy Qur'an; thus, they were also paired with hard negatives. The distribution of the training, development, and test sets are shown in Table 4.

For the additional QP pairs, we adopted the same span-based answer extraction methodology utilized while developing the original QRCD dataset. One Qur'an specialist and two annotators (who are knowledgeable about the Qur'an), extracted the specific answer spans from their respective direct (gold) verse-based answers given by AyaTEC.[8]

[8] Only Qur'an specialists can decide if a verse-based answer represents a *direct* or *indirect* answer to a given question. For a formal definition of a *direct* and *indirect* answer, refer to Malhas and Elsayed (2020).

4.3 Evaluation Setup

In this section, we demonstrate the approach applied to evaluate the performance of participating systems in the MRC task.

4.3.1 Leaderboard and Baseline

As mentioned previously, the leaderboard for the MRC task was hosted on CodaLab (Pavao et al., 2023) with the same conditions over the number of allowed runs. The run file should be in JSON format as in Qur'an QA 2022. However, its format is slightly different. Every answer to each question is a dictionary containing the answer text span, rank, score, start token position, and end token position. The latter two key-value pairs are newly introduced for the task this year.

The baseline for this task is a simple system that gives the whole passage as an answer to the corresponding question. We denote this baseline as *Whole Passage*. Similar to the PR task, we made the baseline code along with submission-checker and evaluation scripts publicly available through the official repository of the shared task.[9]

4.3.2 Evaluation Measures

We chose *partial Average Precision* (pAP) as the main evaluation measure. It is a rank-based measure that integrates *partial matching* to give credit to a QA system that may retrieve an answer that is not necessarily at the first rank and/or *partially* match one of the gold answers (Malhas and Elsayed, 2022). Moreover, pAP is capable of evaluating questions that may have one or more answers in the accompanying passage. This makes pAP more suitable to the MRC task of Qur'an QA 2023 than *partial Reciprocal Rank* (pRR), which was the main evaluation measure for the MRC task in Qur'an QA 2022. Participating systems in the latter task were only required to return *any* answer to a given question even if it has more than one answer in the given passage. Similar to the PR task, the no-answer cases are handled simply by giving full credit to "no answers" system output and zero otherwise. To get an overall evaluation score, the measure is averaged over all questions.

Since the MRC task in Qur'an QA 2023 is different and more challenging than that in Qur'an QA 2022, performance comparisons between the two are not meaningful.

4.4 Results

Twenty nine teams registered for the task. Among these teams, six participated in the final (test) phase with 17 run submissions. The teams are officially ranked based on their best performing submitted run. The performance on the test set of all submitted runs is shown in Table 5, where the runs are ranked by pAP.

It is evident that all teams but one showed superiority over the baseline. The highest pAP score is 0.5711, which was achieved by the TCE (Elkomy and Sarhan, 2023) team. The performance distribution of submitted runs is captured in Figure 5 (in the Appendix). We observed diverse performance across the questions for most of the runs. More details about the teams' approaches are provided next.

Team	Run	pAP
TCE	4dfb8d601	0.5711
TCE	dac0bdf4b	0.5643
Al-Jawaab	tpgp4	0.5457
Al-Jawaab	tgp4	0.5393
TCE	ccc877dca	0.5311
LKAU23	run03	0.5008
LKAU23	run02	0.4989
LowResContextQA	run01	0.4745
LowResContextQA	run02	0.4745
LowResContextQA	run03	0.4745
GYM	run0	0.4613
GYM	ensemble	0.4588
LKAU23	run01	0.4541
GYM	test1	0.4304
Baseline	*WholePassage*	0.3268
PSUT	run2	0.2630
PSUT	RUN3	0.2396
PSUT	RUN1	0.0000

Table 5: MRC evaluation results of all submitted runs ranked by pAP. The team name is removed from the run name to save space. The underlined row is the median run.

4.5 Methods and Analysis

In this section, we provide an overview of the main approaches employed by the 6 participating teams in their submitted runs on the MRC test set. We do that with a focus on the methods employed to address the additional challenges in the MRC task (in its second version); namely, zero-answer questions and multi-answer questions.

[9] https://gitlab.com/bigirqu/quran-qa-2023

Except for Al-Jawaab team (Zekiye and Amroush, 2023) that leveraged generative pre-trained Large Language Models (LLMs) with zero-shot learning setups, all systems of the remaining teams employed encoder-only pre-trained LMs. With the relatively modest size of the QRCDv1.2 dataset, almost all systems leveraged transfer learning by using Arabic pre-trained LMs fine-tuned using large Arabic MRC resources (before fine-tuning using QRCDv1.2) to better perform on the downstream MRC task. Leveraging transfer learning, in the same way, was also heavily witnessed among most of the above-median performing teams in Qur'an QA 2022 (Ahmed et al., 2022; Mostafa and Mohamed, 2022; Wasfey et al., 2022; Premasiri et al., 2022). Interestingly, AraELECTRA (Antoun et al., 2021) and AraBERT (Antoun et al., 2020) LMs maintained their leading performance in both Qur'an QA 2022 and Qur'an QA 2023.

The majority of the systems used one (or more) of the following large Arabic MRC resources for fine-tuning. Ar_TyDiQA (Clark et al., 2020) was used by the TCE (Elkomy and Sarhan, 2023), LowResContextQA (Veeramani and Roy, 2023) and GYM (Mahmoudi and Morshedzadeh, 2023) teams; Arabic SQuADv2.0 (Ahmed, 2023) was used by the GYM and the PSUT teams; and ARCD (Mozannar et al., 2019) and AQQAC (Alqahtani and Atwell, 2018) were used by the LKAU23 (Alnefaie et al., 2023) team. Ensemble and/or self-ensemble learning approaches were also employed by the TCE, LKAU23, LowResContextQA, and GYM teams.

To address the challenge of the zero-answer questions, the TCE, GYM and PSUT teams utilized SQuADv2.0-like fine-tuning (Rajpurkar et al., 2018; Devlin et al., 2019) that uses the [CLS] token to predict the likelihood/probability of a given question to have an answer in the accompanying passage. Interestingly, Al-Jawaab team utilized a carefully hand-crafted prompt (shown in Figure 3) to address the challenge of zero-answer as well as multi-answer questions. The prompt was phrased to instruct their two generative (GPT-4) pre-trained LLMs to answer a given question from its accompanying passage with one *or more* answers, such that they must be extracted from the given passage. The prompt also instructs the model to generate a "no answer" if the given passage does not include an answer to the given question.

ءجب على السؤال التالي من النص المرفق فقط .
لا تقم بإضافة أية شرح أو أية إجابة من خارج
النص. اكتب الإجابة أو الإجابات فقط، إن وجدت
أكثر من إجابة اكتبها على شكل تعدادات. الاجابة
يجب أن تكون فقط المقطع أو المقاطع التي تحوي
الجواب بدون أية زيادة. اجعل كل مقطع في سطر
منفصل. إن لم توجد إجابة، اكتب: "No Answer".

Figure 3: The handcrafted prompt used by the Al-Jawaab team with their employed generative models.

As for multi-answer questions, the TCE team employed Maximum Marginal Likelihood (MML) fine-tuning to address this challenge in the MRC task. MML is a form of Bayesian fine-tuning that incorporates uncertainty to preclude the trained systems from being overly confident in a single answer span; thus, distributing the probability among more than one answer span in the accompanying passage of a given question. MML fine-tuning seems to be among the main contributors to the leading performance achieved by TCE (Table 5). No other team addressed this challenge explicitly, other than Al-Jawaab team which used prompt engineering with its generative models (as mentioned above).

An important finding by Al-Jawaab team, is that despite their careful prompt engineering scenarios to instruct their generative GPT-4-based models (OpenAI, 2023a; Schreiner, 2023), not to provide out-of-passage answers to a given question, the models sporadically succeeded in providing answer spans strictly from the accompanying passages. Among the main problems was "prompt injection", where parts of the textual prompt instruction/question given to the model are injected back into the generated answer. As such, they applied some post-processing heuristics to the answers obtained by their top performing model.

5 Conclusion

With prevalent *semantic* search approaches on the Holy Qur'an being predominantly ontology-based, we believe that recent neural dense and generative retrieval approaches coupled with the resurgence of the MRC field would pave the way for more intelligent state-of-the-art QA systems on the Holy Qur'an.

To this end, we organized Qur'an QA 2023 shared task, which witnessed the participation of 27 team members from 17 different institutes representing 10 teams. Our shared task in its second version comprised two subtasks; a passage retrieval

(PR) task and a machine reading comprehension (MRC) task. It attracted 9 teams to submit 22 runs for the PR task, and 6 teams to submit 17 runs for the MRC task.

As anticipated, recent transformer-based neural retrieval and reading comprehension approaches were heavily employed by all the participating systems. The majority of the systems deployed encoder-based BERT-like models, whereas generative (decoder-based) GPT-like models were used more sparingly in both tasks. The performance of the systems on the test sets in both tasks indicates that encoder-based transformer models are still taking the lead over generative transformer models. Interestingly, AraELECTRA and AraBERT fine-tuned using large external task-related resources pioneered the Arabic transformers scene. These two models were employed by the best-performing team in each task with self-ensemble. The second best-performing teams in both tasks leveraged generative transformer models (LLMs) using zero-shot learning setups. Though in the PR task, the second ranked team utilized an Arabic-to-English translation module with their retrieval module. The majority of the semantic search/retrieval systems used bi-encoder and cross-encoder architectures independently or jointly. Also, ensemble and self-ensemble approaches were employed by many systems to stabilize prediction fluctuations across runs and/or to enhance prediction accuracy through the wisdom of the crowd.

For zero-answer questions, the best system adopted a thresholding mechanism to identify questions with a low predicted likelihood of having answers in the Holy Qur'an (for Task A), or in the accompanying passage (for Task B). The majority of the teams did not address this challenge *explicitly* in both tasks, other than the second ranked team adopting a naive handcrafted prompt, engineered to instruct their generative GPT-4-based models to return a "no answer" for the MRC task.

As for multi-answer questions in the MRC task, the best performing system employed MML Bayesian fine-tuning to address this challenge. No other team addressed this challenge *explicitly*, other than the second ranked team which used prompt engineering with its generative-based models (as mentioned above). We note that multi-answer (or multi-span) extraction in the literature is an active area of research in the extractive MRC/QA scene that would benefit Qur'anic QA research.

Our prospects towards the third version of the shared task are to aim at including an end-to-end QA task on the Holy Qur'an.

Limitations

The sizes of the AyaTEC and QRCD datasets are relatively modest. This is mainly attributed to the sensitivity of dealing with the sacred Holy Qur'an, for which we have adopted a rigorous and strict process for extracting and annotating the verse-based and span-based answers to the questions of the datasets. Nevertheless, we have foreseen the opportunity to leverage transfer learning and/or model adaptation among other state-of-the-art neural approaches to overcome size-related concerns by question answering systems.

Acknowledgements

We would like to thank all the Qur'an specialists who contributed to extracting/annotating the verse-based and span-based answers to the additional test questions in the datasets; especially Dr. Ahmad Shukri, Professor of Tafseer and Qur'anic Sciences at Qatar University, for his scholarly advice throughout the annotation process of the answers extracted from the Holy Qur'an.

References

Basem H. Ahmed, Motaz K. Saad, and Eshrag A. Refaee. 2022. QQATeam at Quran QA 2022: Fine-Tunning Arabic QA Models for Quran QA Task. In *Proceedings of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT5) at the 13th Language Resources and Evaluation Conference (LREC 2022)*.

Zeyad Ahmed. 2023. Arabic squad v2.0 dataset based on the popular squadv2.0 with unanswered questions for more challenging task. https://huggingface.co/datasets/ZeyadAhmed/Arabic-SQuADv2.0. Accessed: September 28, 2023.

Hend Al-Khalifa, Tamer Elsayed, Hamdy Mubarak, Abdulmohsen Al-Thubaity, Walid Magdy, and Kareem Darwish. 2022. Proceedinsg of the 5th workshop on osact with shared tasks on qur'an qa and fine-grained hate speech detection. In *Proceedinsg of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools with Shared Tasks on Qur'an QA and Fine-Grained Hate Speech Detection*.

Hessa A. Alawwad, Lujain A. Alawwad, Jamilah Alharbi, and Abdullah I. Alharbi. 2023. AHJL at Qur'an QA 2023 Shared Task: Enhancing Passage

Retrieval using Sentence Transformer and Translation. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore.

Sarah Alnefaie, Abdullah N. Alsaleh, Eric Atwell, Mohammad Ammar Alsalka, and Abdulrahman Altahhan. 2023. LKAU23 at Qur'an QA 2023: Using Transformer Models for Retrieving Passages and Finding Answers to Questions from the Qur'an. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore.

Mohammad Alqahtani and Eric Atwell. 2018. Annotated corpus of Arabic al-quran question and answer.

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

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

Muhammad Huzaifa Bashir, Aqil M Azmi, Haq Nawaz, Wajdi Zaghouani, Mona Diab, Ala Al-Fuqaha, and Junaid Qadir. 2022. Arabic natural language processing for Qur'anic research: a systematic review. *Artificial Intelligence Review*, pages 1–54.

Danqi Chen. 2018. *Neural Reading Comprehension and Beyond*. Ph.D. thesis, Stanford University.

Jonathan H Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. Tydi qa: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

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)*, page 4171–4186. Association for Computational Linguistics.

Mohammed Alaa Elkomy and Amany Sarhan. 2023. TCE at Qur'an QA 2023 Shared Task: Low Resource Enhanced Transformer-based Ensemble Approach for Qur'anic QA. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

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.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Jimmy Lin and Boris Katz. 2006. Building a reusable test collection for question answering. *Journal of the American Society for Information Science and Technology*, 57(7):851–861.

Ghazaleh Mahmoudi and Yeganeh Morshedzadeh. 2023. GYM at Qur'an QA 2023 Shared Task: Multi-Task Transfer Learning for Quranic Passage Retrieval and Question Answering with Large Language Models. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore.

Rana Malhas. 2023. *Arabic Question Answering on the Holy Qur'an*. Ph.D. thesis, Qatar University.

Rana Malhas and Tamer Elsayed. 2020. AyaTEC: Building a Reusable Verse-based Test Collection for Arabic Question Answering on the Holy Qur'an. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 19(6).

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 *Proceedings of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT5) at the 13th Language Resources and Evaluation Conference (LREC 2022)*, pages 79–87.

Aly Mostafa and Omar Mohamed. 2022. GOF at Qur'an QA 2022: Towards an Efficient Question Answering For The Holy Qu'ran In The Arabic Language Using Deep Learning-Based Approach. In *Proceedings of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT5) at the 13th Language Resources and Evaluation Conference (LREC 2022)*.

Hussein Mozannar, Elie Maamary, Karl El Hajal, and Hazem Hajj. 2019. Neural arabic question answering. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, page 108–118. Association for Computational Linguistics.

OpenAI. 2023a. GPT-4 Technical Report. Technical report.

OpenAI. 2023b. New and improved embedding model. https://openai.com/blog/new-and-improved-embedding-model. Accessed: September 27, 2023.

OpenAI. 2023c. OpenAI Embeddings API. https://platform.openai.com/docs/guides/embeddings/. Accessed: September 27, 2023.

Adrien Pavao, Isabelle Guyon, Anne-Catherine Letournel, Dinh-Tuan Tran, Xavier Baro, Hugo Jair Escalante, Sergio Escalera, Tyler Thomas, and Zhen Xu. 2023. Codalab competitions: An open source platform to organize scientific challenges. *Journal of Machine Learning Research*, 24(198):1–6.

Damith Premasiri, Tharindu Ranasinghe, Wajdi Zaghouani, Ruslan Mitkov, Jamal Berrich, and Toumi Bouchentouf. 2022. DTW at Qur'an QA 2022: Utilising Transfer Learning with Transformers for Question Answering in a Low-resource Domain . In *Proceedings of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT5) at the 13th Language Resources and Evaluation Conference (LREC 2022)*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*.

Nils Reimers. 2023. SBERT Semantic Search. https://www.sbert.net/examples/applications/semantic-search/README.html. Accessed: September 27, 2023.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Ali Safaya, Moutasem Abdullatif, and Deniz Yuret. 2020. Kuisail at semeval-2020 task 12: Bert-cnn for offensive speech identification in social media. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 2054–2059.

Cicero Nogueira dos Santos, Xiaofei Ma, Ramesh Nallapati, Zhiheng Huang, and Bing Xiang. 2020. Beyond [cls] through ranking by generation. *arXiv preprint arXiv:2010.03073*.

Maximilian Schreiner. 2023. Gpt-4 architecture, datasets, costs and more leaked. https://the-decoder.com/gpt-4-architecture-datasets-costs-and-more-leaked/. Accessed: September 28, 2023.

Marwan N. Swar. 2007. *Mushaf Al-Tafseel Al-Mawdoo'ee*. Dar Al-Fajr Al-Islami, Damascus.

Hariram Veeramani and Kaushik Roy. 2023. LowResContextQA at Qur'an QA 2023 Shared Task: Temporal and Sequential Representation Augmented Question Answering Span Detection. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore.

Kexin Wang, Nils Reimers, and Iryna Gurevych. 2021. TSDAE: Using transformer-based sequential denoising auto-encoderfor unsupervised sentence embedding learning. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 671–688, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ahmed Wasfey, Eman Elrefai, Muhammad Marwa, and Nawaz Haq. 2022. Stars at Qur'an QA 2022: Building Automatic Extractive Question Answering Systems for the Holy Qur'an with Transformer Models and Releasing a New Dataset. In *Proceedings of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT5) at the 13th Language Resources and Evaluation Conference (LREC 2022)*.

Andrew Yates, Rodrigo Nogueira, and Jimmy Lin. 2021. Pretrained transformers for text ranking: BERT and beyond. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Tutorials*, pages 1–4, Online. Association for Computational Linguistics.

Abdulrezzak Zekiye and Fadi Amroush. 2023. Al-jawaab at Qur'an QA 2023 shared task: Exploring Embeddings and GPT Models for Passage Retrieval and Reading Comprehension. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore.

Xinyu Zhang, Xueguang Ma, Peng Shi, and Jimmy Lin. 2021. Mr. TyDi: A multi-lingual benchmark for dense retrieval. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 127–137, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Fengbin Zhu, Wenqiang Lei, Chao Wang, Jianming Zheng, Soujanya Poria, and Tat-Seng Chua. 2021. Retrieving and reading: A comprehensive survey on open-domain question answering. *arXiv preprint arXiv:2101.00774*.

A Appendix

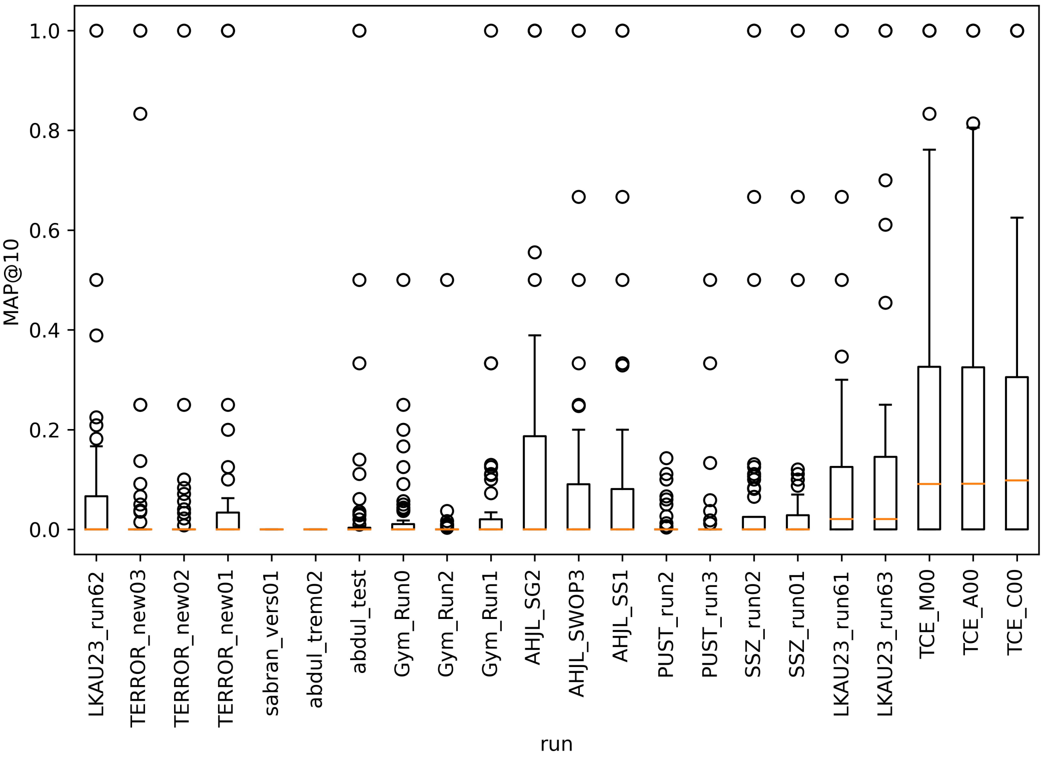


Figure 4: Boxplots for the MAP@10 metric for all submitted runs on the PR task. The plot illustrates the median and inter-quartile distance across questions.

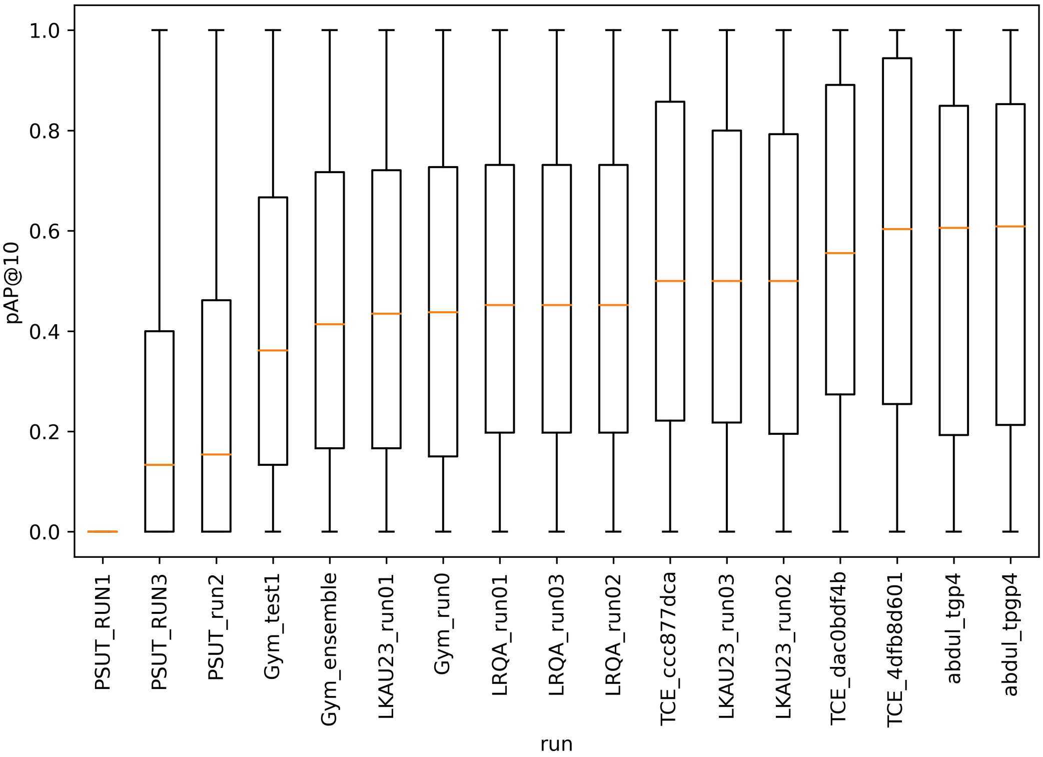


Figure 5: Boxplots for the pAP metric of all submitted runs on task-B. The plot illustrates the median and inter-quartile distance across questions. LRQA is shortened from LowResContextQA.

AHJL at Qur'an QA 2023 Shared Task: Enhancing Passage Retrieval using Sentence Transformer and Translation

Hessa A. Alawwad
Imam Mohammad Ibn Saud Islamic
University (IMSIU)
haalawwad@imamu.edu.sa

Lujain A. Alawwad
Saudi Electronic University
L.alawwad@seu.edu.sa

Jamilah Alharbi
King Abdulaziz University
eng.jamilah@gmail.com

Abdullah I. Alharbi
King Abdulaziz University
aamalharbe@kau.edu.sa

Abstract

The Holy Qur'an is central to Islam, influencing around two billion Muslims globally, and is known for its linguistic richness and complexity. This article discusses our involvement in the PR task (Task A) of the Qur'an QA 2023 Shared Task. We used two models: one employing the Sentence Transformer and the other using OpenAI's embeddings for document retrieval. Both models, equipped with a translation feature, help interpret and understand Arabic language queries by translating them, executing the search, and then reverting the results to Arabic. Our results show that incorporating translation functionalities improves the performance in Arabic Question-Answering systems. The model with translation enhancement performed notably better in all metrics compared to the non-translation model.

1 Introduction

The Holy Qur'an holds significant relevance as it serves as the central holy book in Islam, guiding the beliefs and practices of over 1.9 billion Muslims worldwide. It provides essential spiritual guidance, imparts moral values, and establishes rules for living, exerting a profound influence on the lives of Muslims and their communities. Comprising 114 chapters (Suras) and 6236 verses (Ayas) of varying lengths, totaling approximately 80,000 Arabic words, the Qur'an, revealed over 1,400 years ago, is written in classical Arabic (Atwell et al., 2011). is considered to be linguistically complex because it uses a rich vocabulary, intricate sentence structures, and rhetorical devices like metaphors and allegories. Its verses can have multiple meanings depending on the context, allowing for various interpretations (Alasmari, 2020). Various studies have explored the Holy Qur'an for different NLP tasks, such as creating datasets, question answering (QA), retrieving related information, and and identifying topics (Adeleke et al., 2019; Mohd et al., 2021; Mohamed and Shokry, 2022; Malhas and Elsayed, 2022).

One recent study on applying NLP to the Qur'an relies on the Qur'an QA shared task (2022) (Malhas et al., 2022). They propose a task defined as giving a group of verses from a particular part of the Holy Qur'an and a question about those verses; a system needs to find the answer to the provided question. The organizers continued to provide this shared task, Qur'an QA 2023 Shared Task. However, they added a new task called the Qur'anic passage retrieval (PR) task. PR is defined as participants will be given a question in Modern Standard Arabic and a set of Qur'anic passages that cover the entire Holy Qur'an. The system is required to return a list of these passages, ranked in order of how likely they are to contain the answer to the provided question. The question may vary in complexity, ranging from simple and direct to more intricate and nuanced. However, some questions might not have an answer in the Holy Qur'an to make the task more realistic and challenging. In such cases, an adequate system should recognize that there is no answer. Otherwise, it should return a list of the top ten passages likely to contain the answer.

This paper describes our participation in the PR task (Task A) provided by the Qur'an QA 2023 Shared Task. Our proposed method is to translate the Arabic Questions into English and incorporate a paraphrasing module to enhance the retrieving process. The rest of the paper is organized as follows: Section 2 discusses the related work. Section 3 explains the data we used for our tests. Section 4 provides details of our experiments. Section 5

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 702–707
December 7, 2023 ©2023 Association for Computational Linguistics

presents the results tied to our research queries. Finally, we discuss potential next steps and conclude the paper.

2 Related Work

In the domains of Natural Language Processing (NLP) and Information Retrieval (IR), the task of Question Answering (QA) involves finding accurate answers to questions within a body of text. QA combines these two fields by requiring an understanding of language, as in NLP, and the ability to find the proper documents, as in IR (Alami et al., 2023). A typical QA system consists of several steps, including understanding questions, finding relevant text passages, and extracting answers to deliver precise responses from extensive textual sources (Alwaneen et al., 2022).

In the field of information retrieval, using language models to rank documents based on their relevance to a query has been a popular method (Ponte and Croft, 2017). Earlier methods used count-based language models for each document to determine its likelihood of being relevant to a query (Zhai and Lafferty, 2004). Sentence similarity involves assessing the likeness between two texts, where each sentence pair is judged based on the notion that they have identical meanings (Achananuparp et al., 2008). Models for sentence similarity transform input texts into embeddings that capture the overall meaning and then compute their proximity according to some specific measure, such as cosine-similarity or dot product. In the Al-Bayan system by Abdelnasser et al. (2014), the researchers utilized the Holy Qur'an and Tafseer to identify verses with similar meanings using semantic analysis. They developed a semantic interpreter with machine learning to transform text into vectors representing Qur'anic concepts. These vectors, built from terms in the relevant documents, are weighted using the TF-IDF method. The system calculates the similarity between the vectors of a given question and terms in the Qur'an, and then highlights the most relevant terms to that question. These methods had challenges, like dealing with limited data.

Using commercial search engines as external sources for paragraph retrieval is one of the methods used in the literature. The EWAQ system, introduced by AL-Khawaldeh (2015), presents a novel passage retrieval (PR) method. This method fetches passages from search engines and calculates their relevance to a query based on "entailment similarity", employing cosine directional similarity as a metric. A similar method for passage retrieval was suggested by Bakari and Neji (2022). Initially, passages related to the query are fetched from Google using the question's keywords. These passages are then refined, standardized, and divided. Next, the questions and passages are examined linguistically, including identifying named entities, analyzing syntax, and assessing morphology. In the end, the main ideas of the question and the passage are presented logically.

More modern techniques use advanced language models like BERT to determine query relevance. Such methods have an advantage over older sparse retrieval techniques because they recognize word-based and more profound meaning similarities rather than just looking for exact keyword matches (Nogueira dos Santos et al., 2020). Karpukhin et al. (2020) aimed to develop an effective dense embedding model by merging the BERT pre-trained model with a dual-encoder setup. This model transforms text into a specific vector format and then indexes every passage for retrieval. They found that their model surpassed several other models in question-answer tests on various datasets like SQuAD (Rajpurkar et al., 2016) and TriviaQA (Joshi et al., 2017).

3 Methodology

3.1 Dataset Description

The dataset used consists of three main components: the Qur'anic Passage Collection (QPC), questions from the AyaTEC dataset, and relevance assessments for these questions against the QPC passages. The QPC was created by organizing the 114 Qur'anic chapters into topic-based segments using the Thematic Holy Qur'an (Swar, 2007), resulting in 1,266 distinct passages. The dataset was split into 70% training with 174 questions including 25 no-answer question, 10% development with 25 question including 4 no-answer question, and 20% testing sets with 52 question. 15% of the total questions are designed to have no corresponding answer in the Qur'an, termed as zero-answer questions to raise the challenge of the model's understanding. The Query Relevance Judgements (QRels) dataset includes 1,132 'gold standard' passage IDs from the Qur'an, each associated with a specific question from the AyaTEC dataset (Malhas and Elsayed, 2020) (Malhas, 2023). For questions that have no

Dataset	Structure	Preprocessing Applied
QPC	<passage-id> <passage-text>	None
Training, Dev, and Test	<question-id> <question-text>	None
QRels Gold	<question-id> Q0 <passage-id> <relevance>	None
Qur'an English Translation	<sura-id> <aya-id> <translation>	Cleaning
Questions (Post-Augmentation)	<question-id> <question-text> <question-en> <question-versions>	Translation and Paraphrasing

Table 1: Dataset Formatting and Structure

answer in the Qur'an, a placeholder value of "-1" is assigned as the passage ID.

Our datasets employ tab-delimited formatting and undergo different types of preprocessing. The architecture of these datasets is described in Table 1. We applied two primary components in our system for question preprocessing: translation and paraphrasing. The resulting structure of the question file post-augmentation is also outlined in Table 1.

We also used the English translation of the meanings of the Qur'an dataset from the Rowwad Translation Center (qur, 2023). It has a total of 6236 records, which represent the translation of every verse in the Quraan. The Ruwwad Centre for Translation has carefully examined each Arabic verse, consulting multiple sources of Arabic Tafseer and grammar. They have opted for modern phrasing and strived to maintain an arrangement that mirrors the original Arabic sequence as closely as possible.

3.2 Model Setup

The proposed cross-lingual model architecture is depicted in Figure 1, and its components are explained in detail. The general components of the model are the English translation module, paraphrasing module, and information retrieval module which is based on the sentence-transformer model.

For the translation and paraphrasing, we used OpenAI ChatCompletion API, gpt-3.5-turbo model, and the prompts: "You will be provided with a sentence in Arabic, and your task is to translate it into English." And "You will be provided with an English question, and your task is to paraphrase it." Respectively for each task. The temperature of the model is 0.9, with 150 maximum tokens. The translation process was proposed to enhance the quality of the processing of the used models, as they performed poorly in Arabic directly. The paraphrasing was proposed to enhance further the accuracy of the answer retrieved.

The retrieved documents of different paraphrases are aggregated and sorted according to their similarity scores, eliminating duplicate documents in case the same document is retrieved from multiple paraphrases. The model handles no-answer questions by setting a threshold value of similarity score in an attempt to eliminate irrelevant documents. such that a document is accepted as an answer if its score exceeds the threshold value. The threshold value was determined according to the analysis conducted during the model experimentation.

The information retrieval model was built using a semantic search (Reimers, 2022). It is also known as dense retrieval, which transforms the search query into a vector representation and identifies document embeddings that are proximate in the vector space. The lexical search seeks exact word-for-word matches of the query terms within the set of documents, failing to account for synonyms and acronyms. Semantic search, on the other hand, converts the search query into a vector format and fetches document embeddings that are close to that vector space.

The initial retrieval system could fetch documents that may not be highly relevant to the search query. To address this, a second-layer re-ranker is employed, which uses a cross-encoder to evaluate and score the relevance of all candidate documents in relation to the specified search query as shown in Figure 1.

In our study, we employ two distinct models to assess the efficacy of document retrieval in a question-answering context. Model A which is a Semantic Search that employs 'msmarco-distilbert-base-tas-b'[1] sentence Transformer model as the bi-encoder, 'cross-encoder/ms-marco-MiniLM-L-6-v2'[2] model as the cross encoder. Model B which is a semantic search that employs OpenAI's best embeddings 'text-embedding-ada-002' engine as the bi-encoder and OpenAI's 'text-davinci-003" engine[3] as the cross encoder. The two models serve

[1] https://huggingface.co/sentence-transformers/msmarco-distilbert-base-tas-b

[2] https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-6-v2

[3] https://platform.openai.com/docs/models/overview

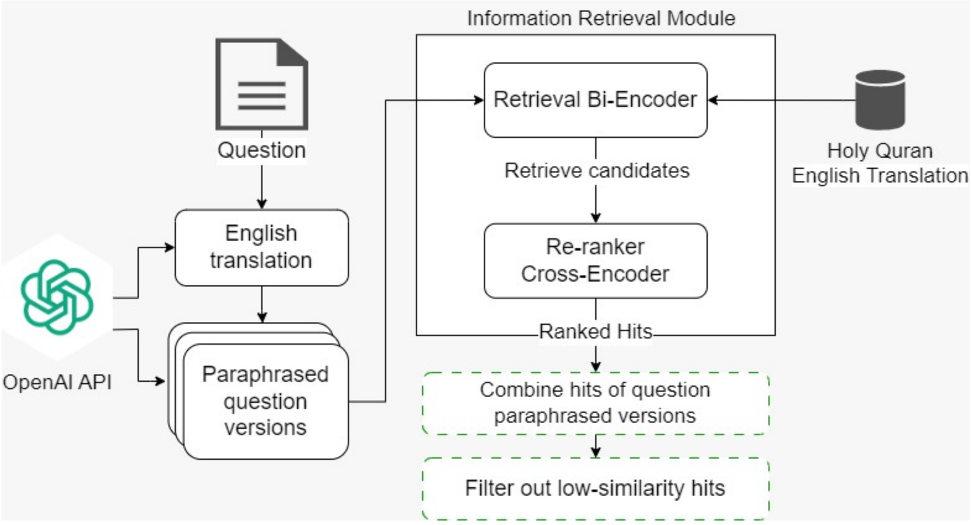

Information Retrieval Module
Question
English translation
OpenAI API
Paraphrased question versions
Retrieval Bi-Encoder
Retrieve candidates
Re-ranker Cross-Encoder
Ranked Hits
Holy Quran English Translation
Combine hits of question paraphrased versions
Filter out low-similarity hits

Figure 1: Model Architecture of the Passage Retrieval.

as a comprehensive setup incorporating a translation component to facilitate multilingual query processing. Built on an advanced neural network architecture, those models with Translation are capable of understanding and interpreting queries in the Arabic language. The translation feature allows it to translate the queries into a common language, perform the search, and then translate the results back into the original language, if necessary. In this work, we elaborate on the three setups used in our experiments: model A with translation and paraphrasing. model A with translation and no paraphrasing, and model B with translation and no paraphrasing.

3.3 Experiments Setup

All the pre-trained models were used in a zero-shot manner. With no fine-tuning on the dataset explained in Dataset Description. The primary metric for evaluation is the Mean Average Precision (MAP) and Mean Reciprocal Rank (MRR). All the experiments were carried out on a single T4 GPU and implemented using Google Collaboratory. We will use our built model with the translated question and no paraphrasing as the baseline for comparison.

4 Result and Discussion

In our research, we initially focused on assessing the capabilities of the semantic search integrated with a translation component. The metrics used for performance evaluation included MRR and MAP. The results of the three proposed models are shown in Table 2. According to the scores on the dev set, the SBERT semantic search without paraphrasing was the best-performing model, with a MAP score of 0.343 and an MRR score of 0.413. When it comes to the test set, the SBERT semantic search without paraphrasing had the highest MAP score of 0.132, while the OpenAI semantic search without paraphrasing had the highest MRR score of 0.389.

Model	Metric	Dev	Test
SBERT with paraphrasing	MAP MRR	0.284 0.408	0.120 0.291
SBERT without paraphrasing	MAP MRR	0.343 0.413	0.132 0.302
OpenAI without paraphrasing	MAP MRR	0.221 0.369	0.199 0.389

Table 2: Performance results for the three proposed models: SBERT semantic search with paraphrasing, SBERT semantic search without paraphrasing and OpenAI semantic search without paraphrasing.

Our findings indicate that the translation-augmented version exhibited significant improvements across all metrics when compared to the model without translation. For instance, the MAP score witnessed an increase from 0.003 using an Arabic sentence transformer model 'medmediani/Arabic-KW-Mdel'[4] to 0.343 using the English sentence transformer model 'msmarco-distilbert-base-tas-b' on the dev set, suggesting that the translation component greatly enhanced the model's ability to retrieve more relevant documents. Overall, integrating translation into the system substantially improved its performance, validating our

[4] https://huggingface.co/medmediani/Arabic-KW-Mdel

hypothesis that translation is a crucial element for improving retrieval quality in the Arabic question-answering (QA) environment.

Building on this, we also introduced a second model that involved multiple paraphrased versions of the input question for even more precise retrieval. The results of the versions of the question produced by the paraphrasing component were sorted, and the duplication in the retrieved answers was deleted. The result of both T-test (Semenick, 1990) and Mann-Whitney U test (McKnight and Najab, 2010) shows no significant difference in MAP and MRR scores with adding the paraphrasing component to the base model.

In the case of questions with no answer, the test set contained 7 questions with no answers, the best model was able to correctly say 'No answer' to four questions, 0.57 of the questions. The threshold value for eliminating irrelevant documents is set to -5, where documents with a score of -6 and below are considered irrelevant.

The test set has in total 7 questions that did not have corresponding answers (no-answer questions). Interestingly, out of these 7 questions, our best-performing model accurately identified 'No answer' for 4 of them, giving us a 57% accuracy rate in this specific context. In order to filter out irrelevant documents, we established a threshold value of -5, which means that any documents scoring -6 or lower were considered irrelevant.

Some questions in the test set are not direct and cannot be solved with similarity measures but rather require some inference methodology to infer the question from the given context.

Certain questions within the test set are indirect and present challenges when addressed through similarity measures. To effectively tackle these questions, a more nuanced approach, specifically an inference methodology, is necessary in order to ascertain the intention of the question from the given context.

5 Conclusion

In this study, we explored the linguistic complexity of the Holy Qur'an, which holds profound influence over approximately two billion Muslims worldwide. Our engagement in the Qur'an QA 2023 Shared Task's PR task (Task A) led us to employ two distinct models: the Sentence Transformer and OpenAI's embeddings, both aimed at effective document retrieval. A significant feature of our approach was the integration of a translation mechanism to facilitate the interpretation of Arabic queries. Upon evaluation, the translation-enhanced model showcased superior performance across all metrics in comparison to its non-translation counterpart.

References

2023. Rowwad translation center, the noble qur'an encyclopedia.

Heba Abdelnasser, Maha Ragab, Reham Mohamed, Alaa Mohamed, Bassant Farouk, Nagwa M El-Makky, and Marwan Torki. 2014. Al-bayan: an arabic question answering system for the holy quran. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 57–64.

Palakorn Achananuparp, Xiaohua Hu, and Xiajiong Shen. 2008. The evaluation of sentence similarity measures. In *Data Warehousing and Knowledge Discovery: 10th International Conference, DaWaK 2008 Turin, Italy, September 2-5, 2008 Proceedings 10*, pages 305–316. Springer.

A Adeleke, NA Samsudin, ZA Othman, and SK Ahmad Khalid. 2019. A two-step feature selection method for quranic text classification. *Indonesian Journal of Electrical Engineering and Computer Science*, 16(2):730–736.

Fatima T AL-Khawaldeh. 2015. Answer extraction for why arabic questions answering systems: Ewaq. *World of Computer Science & Information Technology Journal*, 5(5).

Hamza Alami, Abdelkader Mahdaouy, Abdessamad Benlahbib, Noureddine En-Nahnahi, Ismail Berrada, and Said El Alaoui Ouatik. 2023. Daqas: Deep arabic question answering system based on duplicate question detection and machine reading comprehension. *Journal of King Saud University-Computer and Information Sciences*, page 101709.

Jawharah Saeed N Alasmari. 2020. *A Comparative Analysis of The Arabic and English Verb Systems Using the Qur'an Arabic Corpus [A corpus-based study]*. Ph.D. thesis, University of Leeds.

Tahani H Alwaneen, Aqil M Azmi, Hatim A Aboalsamh, Erik Cambria, and Amir Hussain. 2022. Arabic question answering system: a survey. *Artificial Intelligence Review*, pages 1–47.

Eric Atwell, Claire Brierley, Kais Dukes, Majdi Sawalha, and Abdul-Baquee Sharaf. 2011. An artificial intelligence approach to arabic and islamic content on the internet. In *Proceedings of NITS 3rd National Information Technology Symposium*, pages 1–8. Leeds.

Wided Bakari and Mahmoud Neji. 2022. A novel semantic and logical-based approach integrating rte technique in the arabic question–answering. *International Journal of Speech Technology*, 25(1):1–17.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781.

Rana Malhas. 2023. *Arabic Question Answering on the Holy Qur'an*. Doctoral dissertation.

Rana Malhas and Tamer Elsayed. 2020. Ayatec: building a reusable verse-based test collection for arabic question answering on the holy qur'an. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 19(6):1–21.

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.

Patrick E McKnight and Julius Najab. 2010. Mann-whitney u test. *The Corsini encyclopedia of psychology*, pages 1–1.

Ensaf Hussein Mohamed and Eyad Mohamed Shokry. 2022. Qsst: A quranic semantic search tool based on word embedding. *Journal of King Saud University-Computer and Information Sciences*, 34(3):934–945.

Masnizah Mohd, Faizan Qamar, Idris Al-Sheikh, and Ramzi Salah. 2021. Quranic optical text recognition using deep learning models. *IEEE Access*, 9:38318–38330.

Cicero Nogueira dos Santos, Xiaofei Ma, Ramesh Nallapati, Zhiheng Huang, and Bing Xiang. 2020. Beyond [CLS] through ranking by generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1722–1727, Online. Association for Computational Linguistics.

Jay M Ponte and W Bruce Croft. 2017. A language modeling approach to information retrieval. In *ACM SIGIR Forum*, volume 51, pages 202–208. ACM New York, NY, USA.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Nils Reimers. 2022. Sbert semantic searchs.

Doug Semenick. 1990. Tests and measurements: The t-test. *Strength & Conditioning Journal*, 12(1):36–37.

M. N. Swar. 2007. *Mushaf Al-Tafseel Al-Mawdoo'ee*. Dar Al-Fajr Al-Islami.

Chengxiang Zhai and John Lafferty. 2004. A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems (TOIS)*, 22(2):179–214.

LowResContextQA at Qur'an QA 2023 Shared Task: Temporal and Sequential Representation Augmented Question Answering Span Detection in Arabic

Hariram Veeramani
Department of Electrical
and Computer Engineering,
UCLA, USA
hariram@ucla.edu

Surendrabikram Thapa
Department of Computer
Science, Virginia Tech,
Blacksburg, USA
sbt@vt.edu

Usman Naseem
College of Science and
Engineering, James Cook
University, Australia
usman.naseem@jcu.edu.au

Abstract

The Qur'an holds immense theological and historical significance, and developing a technology-driven solution for answering questions from this sacred text is of paramount importance. This paper presents our approach to task B of Qur'an QA 2023, part of EMNLP 2023, addressing this challenge by proposing a robust method for extracting answers from Qur'anic passages. Leveraging the Qur'anic Reading Comprehension Dataset (QRCD) v1.2, we employ innovative techniques and advanced models to improve the precision and contextuality of answers derived from Qur'anic passages. Our methodology encompasses the utilization of start and end logits, Long Short-Term Memory (LSTM) networks, and fusion mechanisms, contributing to the ongoing dialogue at the intersection of technology and spirituality.

1 Introduction

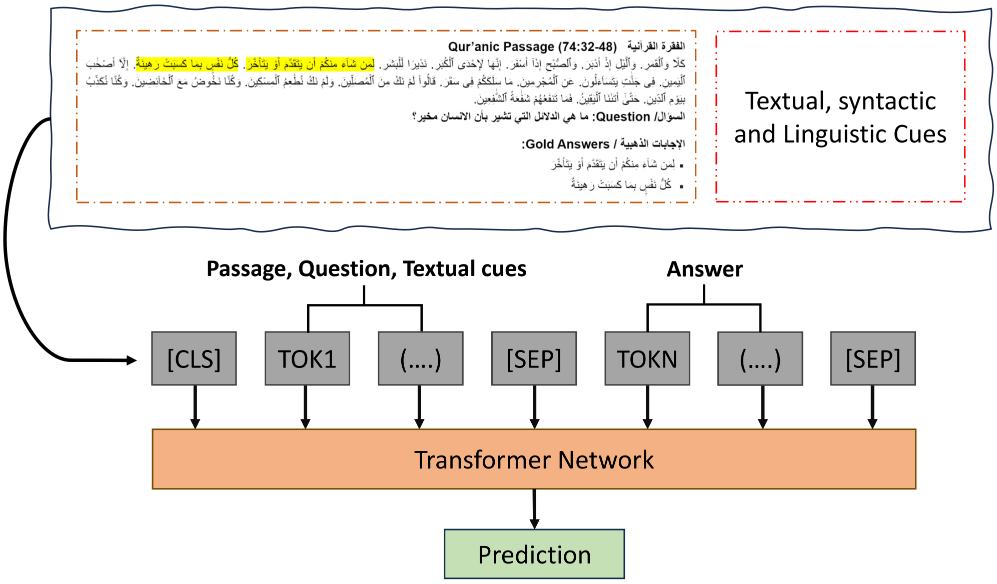
The Holy Qur'an considered the central religious text of Islam, is a source of profound wisdom, guidance, and spiritual insight for millions of people around the world (Touati-Hamad et al., 2020). Its rich and complex content spans a wide range of topics, encompassing historical narratives, moral teachings, legal principles, and metaphysical concepts (Ahmed and Atwell, 2016). For devout Muslims, seeking knowledge and understanding from the Qur'an is a fundamental aspect of their faith, and it serves as a cornerstone for theological, ethical, and philosophical discourse (Malhas et al., 2022).

In the age of information technology, the quest for a deeper comprehension of the Qur'an has extended beyond traditional exegesis, embracing digital tools and computational approaches (Bashir et al., 2023; Malhas and Elsayed, 2022; Ahmed and Atwell, 2016; Mohamed and El-Behaidy, 2021; Veeramani et al., 2023b,d,e). One such critical task in this domain is Qur'anic question-answering (QA), which bridges the sacred text with modern technology and linguistic analysis (Malhas et al., 2022). The goal of Qur'anic QA is to enable the retrieval of specific, contextually relevant answers (Malhas et al., 2022; Malhas and Elsayed, 2022, 2020) to a wide range of questions from the Qur'an's voluminous text.

This paper addresses the pressing need to develop and refine QA systems tailored for Qur'anic texts. In this paper, we provide a detailed description of our system for task B of the Qur'an QA 2023 shared task (Malhas et al., 2023). The task at hand involves providing accurate, contextually appropriate answers to questions posed in Modern Standard Arabic (MSA) regarding specific Qur'anic passages. These passages consist of consecutive verses from a particular Surah (chapter) of the Qur'an. The complexity of this task arises from the need to extract precise answers directly from the provided passage, ensuring that the responses are contextually relevant and adhere to the theological and linguistic nuances of the Qur'an.

Our model uses start and end logits, augmented by employing two model variants. Using two separate question-answering models enables us to explore different aspects of the task, capitalizing on the strengths of each model to ensure comprehensive coverage and accuracy in answer extraction. To further enhance the accuracy and relevance of our system, we pick the best start-end logits with Long Short-Term Memory (LSTM) networks (Hochreiter and Schmidhuber, 1997) and multilayer perception. In this endeavor, we aim to advance both Natural Language Processing (NLP) and the accessibility of the Qur'an's profound wisdom. Our work not only provides a bridge between technology and spirituality but also strives to make the wealth of knowledge contained within the Qur'an more accessible to individuals seeking answers to a wide array of inquiries, whether they be of a religious, historical, or ethical nature. Addi-

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 708–713
December 7, 2023 ©2023 Association for Computational Linguistics



الفقرة القرآنية Qur'anic Passage (74:32-48)
كَلَّا وَٱلْقَمَرِ. وَٱلَّيْلِ إِذْ أَدْبَرَ. وَٱلصُّبْحِ إِذَآ أَسْفَرَ. إِنَّهَا لَإِحْدَى ٱلْكُبَرِ. نَذِيرًا لِّلْبَشَرِ. لِمَن شَآءَ مِنكُمْ أَن يَتَقَدَّمَ أَوْ يَتَأَخَّرَ. كُلُّ نَفْسٍۭ بِمَا كَسَبَتْ رَهِينَةٌ. إِلَّآ أَصْحَٰبَ
ٱلْيَمِينِ. فِى جَنَّٰتٍ يَتَسَآءَلُونَ. عَنِ ٱلْمُجْرِمِينَ. مَا سَلَكَكُمْ فِى سَقَرَ. قَالُوا۟ لَمْ نَكُ مِنَ ٱلْمُصَلِّينَ. وَلَمْ نَكُ نُطْعِمُ ٱلْمِسْكِينَ. وَكُنَّا نَخُوضُ مَعَ ٱلْخَآئِضِينَ. وَكُنَّا نُكَذِّبُ
بِيَوْمِ ٱلدِّينِ. حَتَّىٰٓ أَتَىٰنَا ٱلْيَقِينُ. فَمَا تَنفَعُهُمْ شَفَٰعَةُ ٱلشَّٰفِعِينَ.
السؤال/ Question: ما هي الدلائل التي تشير بأن الانسان مخير؟
الإجابات الذهبية / Gold Answers:
لِمَن شَآءَ مِنكُمْ أَن يَتَقَدَّمَ أَوْ يَتَأَخَّرَ
كُلُّ نَفْسٍۭ بِمَا كَسَبَتْ رَهِينَةٌ
Textual, syntactic and Linguistic Cues
Passage, Question, Textual cues
Answer
[CLS]
TOK1
(....)
[SEP]
TOKN
(....)
[SEP]
Transformer Network
Prediction

Figure 1: A brief overview of a question-answering model for extracting answers from Qur'anic passages.

tionally, it offers a valuable resource for scholars, educators, and researchers engaged in Qur'anic studies, empowering them to navigate the text efficiently and extract pertinent information.

2 Task Description

We only participated in Task B of Qur'an QA 2023. For task B, given a specific passage from the Qur'an consisting of consecutive verses within a particular Surah, along with a free-text question posed in MSA regarding that passage, the system's objective is to extract all answers to the question that are explicitly stated within the provided passage. The answers extracted must be in the form of text spans directly sourced from the given passage. In order to enhance the task's realism and difficulty level, some questions may not have a corresponding answer within the provided passage. In such instances, the ideal system should return no answers. Conversely, when there are answers present in the passage, the system should return a ranked list of up to 10 answer spans that are relevant to the question.

The evaluation measure utilized for this task is partial Average Precision (pAP) (Malhas and Elsayed, 2022). This metric plays a central role in assessing the performance of Question-Answering (QA) systems by incorporating partial matching. It acknowledges and rewards QA systems that retrieve answers that may not necessarily occupy the top rank and may only partially match one of the gold-standard answers. Additionally, pAP is particularly well-suited for evaluating questions that may have one or more valid answers within the accompanying passage. For questions where no answer exists within the provided passage, the evaluation approach is straightforward. A "no-answer" system output is granted full credit, while any other response is assigned a score of zero. To arrive at an overall evaluation score, the pAP measure is calculated and averaged across all questions, providing a comprehensive assessment of the QA system's performance. This metric is designed to capture the system's effectiveness in terms of accuracy and ranking relevance, offering a holistic view of its capabilities in the context of Qur'anic text-based question-answering.

3 Dataset

Task B utilizes the QRCD (Qur'anic Reading Comprehension Dataset) v1.2. This dataset (Malhas and Elsayed, 2020, 2022) currently consists of 1,155 question-passage pairs, forming 1,399 question-passage-answer triplets. The data split for training, development, and test sets is targeted at 70%, 10%, and 20%, respectively. A unique aspect of this dataset is the inclusion of "zero-answer questions", which make up 15% of the questions and are questions without answers in the Holy Qur'an. This addition aims to provide a more realistic and challenging reading comprehension task.

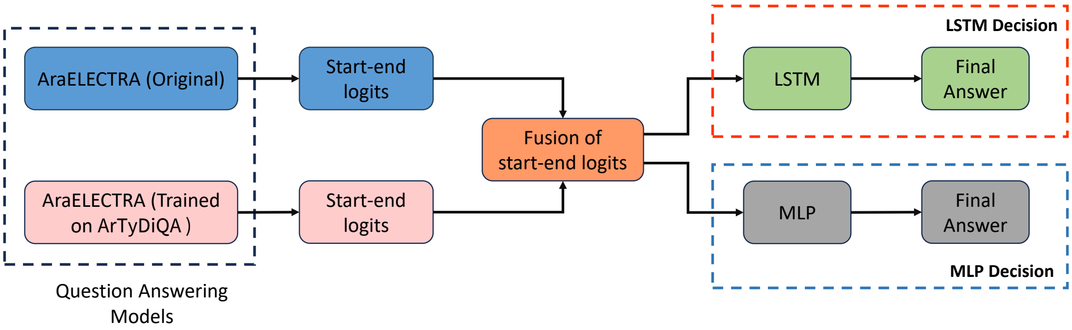


Figure 2: Framework of our methodology for extracting answers from Qur'anic passages.

4 System Description

In this section, we describe two different question-answering models we use along with our methodology. Figure 2 represents our approach for the Qur'an question-answering task.

4.1 Question Answering Models

QA models are a subset of NLP models designed to answer questions in human language automatically. They employ machine learning and deep learning techniques to understand questions and relevant text to extract suitable answers. In the context of Qur'an QA, Figure 1 provides an overview of our approach to extracting answers from Qur'anic passages.

AraELECTRA (Original)–Model 1: AraELECTRA (Antoun et al., 2021) is an Arabic language representation model that is pre-trained using the replaced token detection (RTD) objective. This objective is similar to the masked language modeling (MLM) objective used by other pre-trained language models, such as BERT and RoBERTa (Liu et al., 2019; Gururangan et al., 2020; Veeramani et al., 2023c,a,f). However, instead of masking tokens in the input sequence, the RTD objective replaces some tokens with a special [MASK] token and then trains the model to distinguish the original tokens from the replaced tokens. It was pre-trained on a large corpus of Arabic text, including news articles, books, and social media posts. AraELECTRA has been shown to outperform previous Arabic language representation models on various natural language processing (NLP) tasks, including question answering, sentiment analysis, and named entity recognition. It is also smaller and faster than previous models, making it more suitable for deployment on resource-constrained platforms. In addition to the original model, we also test AraElectra-ARCD[1].

AraELECTRA-ArTyDiQA–Model 2: This version of AraELECTRA is trained on the extensive ArTyDiQA dataset, which offers several advantages for question answering. Firstly, its pre-training on ArTyDiQA, a substantial Arabic question-answering dataset, equips it with a strong grasp of the Arabic language's nuances and its usage in the context of question answering. This enhanced language understanding enables AraELECTRA-ArTyDiQA to comprehend the intent of questions better and effectively extract relevant information from the corpus. Additionally, as AraELECTRA is built upon the ELECTRA architecture (Clark et al., 2019), it benefits from rapid and effective learning facilitated by the ArTyDiQA dataset, which adeptly captures the intricacies of Arabic question answering.

4.2 Answer Span Start-End Logits

In QA models, start and end logits are critical components that facilitate the extraction of answers from a given passage. These logits are computed for each token in the passage when the model analyzes a question and a text. They represent the likelihood that a token serves as the start or end point of the answer. By comparing these logits, the model identifies potential answer spans by selecting tokens with the highest combined scores. The final answer span is determined by choosing a continuous sequence of tokens with the highest joint likelihood based on the start and end logits. In some cases, QA models may further enhance answer selection by scoring and ranking multiple possible spans, ultimately presenting the span with the high-

[1] https://huggingface.co/salti/AraElectra-base-finetuned-ARCD

est overall score, which usually includes contextual information beyond just the logits. This mechanism ensures the model provides accurate and contextually relevant answers to the posed questions. In our case, we take start and end logits from both the models we used.

We employ two distinct QA model settings in our approach. In the first setting, we utilize start and end logits independently. These logits are processed by passing them through a Multi-Layer Perceptron (MLP) layer. This configuration allows each model to make individual predictions based on its understanding of the input, ensuring a level of independence in their responses. In the second setting, we introduce a fusion process for the start and end logits obtained from two separate models. These fused logits are then fed into the MLP layer. This fusion mechanism enables the models to collaboratively refine their predictions, potentially benefiting from the diverse insights each model offers.

We also utilize Long Short-Term Memory (LSTM) networks and MLP in our experiment. The LSTM component enhances the models' ability to capture temporal dependencies across passages/answers along with the sequential representation of the input data. It promotes local context understanding and global features, further optimizing the models' performance in delivering accurate and contextually relevant answers to the posed questions.

4.3 Decision Mechanism

We adopt MLP (Rosenthal et al., 2017; Kanagasabai et al., 2023)and LSTM to extract finer features to reinforce the confidence in picking the right start-end logit pair from one of the above-mentioned models. In both our Multi-Layer Perceptron (MLP) and Long Short-Term Memory (LSTM) components, we apply specific mechanisms to refine the models' predictions. For the LSTM, we utilize a softmax function. Softmax is employed to transform the LSTM output into a probability distribution over possible answer spans. This ensures that the model assigns a probability score to each pair of start-end logits, indicating its likelihood of being part of the answer span.

On the other hand, in the MLP layer, we employ an argmax of the computed class probabilities/classes to identify the answer's starting and ending points with the highest probability score. This is done along with the logits processed through the MLP. The argmax function selects the start-end logit pair with the highest predicted probability as the start of the answer span and the token with the highest predicted probability as the end of the answer span.

5 Results

In Table 1 showcasing results for Quran question answering, models are evaluated based on their performance measured in partial Average Precision (pAP). Model 1 and Model 2 achieve pAP scores of 0.367 and 0.406, respectively. Model 2 achieves a pAP score of 0.474 on the test set. A configuration combining both models (fused logits) using the LSTM branch achieves a pAP score of 0.411 during evaluation. The model configuration involving fusing logits with the MLP layer excels with a pAP score of 0.435 during evaluation, and we expect even better performance on the test dataset. Because of the unavailability of the test dataset, we only report the best submission score. Similar to the previous model, using the MLP branch but this time with AraElectra-ARCD instead of AraELECTRA (original), we achieved the highest pAP score of 0.442 on the evaluation dataset. These scores reflect the efficacy of different model configurations in Quranic question answering, with fused logits using the MLP branch displaying the highest overall performance, particularly on the test dataset.

Models	Eval	Test
Model 1	0.367	-
Model 2	0.406	0.474
Fused Logits (LSTM branch)	0.411	-
Fused Logits (MLP branch)	0.435	-
Fused Logits (AraElectra-ARCD + AraElectra) MLP Branch	**0.442**	-

Table 1: Results for various models with the dataset provided. All values are given with the pAP metric.

6 Conclusion

In summary, our paper contributes to developing precise Question-Answering (QA) systems for Qur'anic texts. By employing advanced techniques and models, we significantly improve answer accuracy and contextuality. Notably, certain model configurations, particularly those incorporating fused logits with the MLP branch, excel in achieving high partial Average Precision (pAP) scores across both evaluation and test datasets. This research not only advances the field of Natural Language Processing (NLP) but also offers an invaluable resource for a diverse audience, ranging from scholars and educators to individuals seeking a deeper understanding of the Qur'an. It bridges technology and spirituality, promoting the harmonious integration of ancient wisdom with modern technology.

Limitations

This work exhibits several limitations. Firstly, the modest size of the QRCD dataset may restrict the models' full potential, warranting consideration for larger and more diverse Qur'anic text datasets. Furthermore, while our models aim for contextuality, capturing the intricate theological and linguistic nuances of the Qur'an remains an ongoing challenge. Addressing these limitations is essential to enhance the versatility and robustness of Question-answer models for Qur'anic texts and potentially expand their utility to broader NLP applications.

Ethics Statement

The Qur'anic text, being a sacred and religious source, is treated with the utmost respect and sensitivity. We have taken measures to ensure that our research and models align with cultural and religious considerations, and we do not engage in any activities that may cause harm or disrespect to any community or belief system. Additionally, we adhere to guidelines on data usage, compliance with applicable laws and regulations, and ethical conduct in research. We aim to contribute positively to the field of Natural Language Processing while promoting inclusivity, respect, and responsible use of technology.

References

Rasha Ahmed and ES Atwell. 2016. Developing an ontology of concepts in the qur'an. *International Journal on Islamic Applications in Computer Science and Technology*, 4(4):1–8.

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

Muhammad Huzaifa Bashir, Aqil M Azmi, Haq Nawaz, Wajdi Zaghouani, Mona Diab, Ala Al-Fuqaha, and Junaid Qadir. 2023. Arabic natural language processing for qur'anic research: A systematic review. *Artificial Intelligence Review*, 56(7):6801–6854.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2019. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Rajaraman Kanagasabai, Saravanan Rajamanickam, Hariram Veeramani, Adam Westerski, and Kim Jung Jae. 2023. Semantists at ImageArg-2023: Exploring Cross-modal Contrastive and Ensemble Models for Multimodal Stance and Persuasiveness Classification. In *Proceedings of the 10th Workshop on Argument Mining*, Online and in Singapore. Association for Computational Linguistics.

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

Rana Malhas and Tamer Elsayed. 2020. Ayatec: building a reusable verse-based test collection for arabic question answering on the holy qur'an. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 19(6):1–21.

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 the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore.

Ensaf Hussein Mohamed and Wessam H El-Behaidy. 2021. An ensemble multi-label themes-based classification for holy qur'an verses using word2vec embedding. *Arabian Journal for Science and Engineering*, 46:3519–3529.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. SemEval-2017 task 4: Sentiment analysis in Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, Vancouver, Canada. Association for Computational Linguistics.

Zineb Touati-Hamad, Mohamed Ridda Laouar, and Issam Bendib. 2020. Quran content representation in nlp. In *Proceedings of the 10th International Conference on Information Systems and Technologies*, pages 1–6.

Hariram Veeramani, Surendrabikram Thapa, and Usman Naseem. 2023a. Automated Citation Function Classification and Context Extraction in Astrophysics: Leveraging Paraphrasing and Question Answering. In *Proceedings of the second Workshop on Information Extraction from Scientific Publications*, Online. Association for Computational Linguistics.

Hariram Veeramani, Surendrabikram Thapa, and Usman Naseem. 2023b. DialectNLU at NADI 2023 Shared Task: Transformer Based Multitask Approach Jointly Integrating Dialect and Machine Translation Tasks in Arabic. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore. Association for Computational Linguistics.

Hariram Veeramani, Surendrabikram Thapa, and Usman Naseem. 2023c. Enhancing ESG Impact Type Identification through Early Fusion and Multilingual Models. In *Proceedings of the Sixth Workshop on Financial Technology and Natural Language Processing (FinNLP)*. Association for Computational Linguistics.

Hariram Veeramani, Surendrabikram Thapa, and Usman Naseem. 2023d. KnowTellConvince at ArAIEval Shared Task: Disinformation and Persuasion Detection in Arabic using Similar and Contrastive Representation Alignment. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore. Association for Computational Linguistics.

Hariram Veeramani, Surendrabikram Thapa, and Usman Naseem. 2023e. LowResourceNLU at BLP-2023 Task 1 2: Enhancing Sentiment Classification and Violence Incitement Detection in Bangla Through Aggregated Language Models. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Hariram Veeramani, Surendrabikram Thapa, and Usman Naseem. 2023f. Temporally Dynamic Session-Keyword Aware Sequential Recommendation system. In *2023 International Conference on Data Mining Workshops (ICDMW)*.

GYM at Qur'an QA 2023 Shared Task: Multi-Task Transfer Learning for Quranic Passage Retrieval and Question Answering with Large Language Models

Ghazaleh Mahmoudi *[1], **Yeganeh Morshedzadeh***[2], **Sauleh Eetemadi**[1]
[1] School of Computer Engineering, Iran University of Science and Technology, Iran
[2] School of Engineering, The University of British Columbia, Canada
gh_mahmoodi@comp.iust.ac.ir, yeganeh.morshedzadeh@ubc.ca, sauleh@iust.ac.ir

Abstract

This work addresses the challenges of question answering for vintage texts like the Quran. It introduces two tasks: passage retrieval and reading comprehension. For passage retrieval, it employs unsupervised fine-tuning sentence encoders and supervised multi-task learning. In reading comprehension, it fine-tunes an Electra-based model, demonstrating significant improvements over baseline models. Our best AraElectra model achieves 46.1% partial Average Precision (pAP) on the unseen test set, outperforming the baseline by 23%.

1 Introduction

Question Answering (QA) for vintage, religious texts like the Quran presents unique challenges for natural language understanding systems. Understanding the concepts and connections in the Quran requires deep semantic reasoning to map questions to relevant passages and surface correct answers. To advance research in this domain, the Qur'anQA 2023 Shared Task [1] proposes two sub-tasks focused on machine comprehension of the Quran (Malhas et al., 2023).

Task A on passage retrieval requires matching Modern Standard Arabic (MSA) free-text questions to Quran verses potentially containing the answer. This tests semantic similarity between questions and passages. We propose using sentence encoders (Reimers and Gurevych, 2019) to derive dense vector representations for questions and passages. These vectors can be indexed and searched efficiently to find relevant matches.

Task B on reading comprehension focuses on extracting span answers from a given passage. This is framed as a machine reading comprehension task. However, given its literary Arabic and frequent need for theological reasoning, it is especially difficult for the Quran. We formulate the task as extractive QA and experiment with span prediction models like AraElectra (Antoun et al., 2021).

The Qur'anQA 2023 shared task includes two sub-tasks that form an end-to-end QA pipeline. Task A retrieves candidate passages potentially containing answers. This narrows the search space from the entire Quran to a small set of relevant verses. Task B then extracts answer spans from these candidates. The tasks work sequentially: passage retrieval provides context to reading comprehension, which verifies answers. Together they comprise an end-to-end QA system over the Quran.

Our key contribution to this work is utilizing transfer learning and model adaptation techniques to develop customized QA models for the limited Qur'anQA 2023 shared task dataset. After experimenting with several Arabic and multi-lingual language models (LMs) we choose AraElectra and AraBERT (Antoun et al.) as strong candidates. These models provide contextual representations of Arabic text learned from broad domains. In this work, we aim to address these research questions:

- How can we adapt LMs for Qur'anQA with limited task data?
- Which methods (e.g., transfer learning, data augmentation, unsupervised pretraining, etc.) improve the accuracy of the Quranic domain?

Through experiments, we analyze different strategies for unsupervised sentence embeddings and supervised task-specific fine-tuning. Despite the scarce training data, this allows the model to learn specialized embeddings for Quranic comprehension. Our work provides insights into adapting pretrained language models to new domains with limited labeled data. By combining broad pre-trained knowledge with targeted fine-tuning, we develop customized QA models capable of reasoning about the Quran's abstract concepts and archaic language. The source code is available at GitHub[2].

[*] These authors contributed equally.

[1] https://sites.google.com/view/quran-qa-2023/home

[2] github.com/ghazaleh-mahmoodi/Quran-QA_2023_Shared-Task

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 714–719
December 7, 2023 ©2023 Association for Computational Linguistics

2 Task A: Passage Retrieval

For a free-text question in MSA, the system must retrieve and rank Quranic passages that potentially contain answers to the question from a corpus covering the entire Quran.

3 Data

For this work, we utilize the training and development datasets provided by the Qur'anQA 2023 organizers (Malhas and Elsayed, 2020; Swar, 2007; Malhas, 2023), a summary of which is provided in Table 1. Across both train and development splits, there are 30 zero-answer questions, meaning that they have no answers in the Quran passages.

To augment the limited size of the Quran-specific data, we incorporate additional datasets during fine-tuning. For this passage retrieval task, we leverage the multi-lingual Mr. TyDi dataset, which contains monolingual question-passage pairs for information retrieval in 11 different languages (Zhang et al., 2021). We utilized the Arabic portion to fine-tune our proposed model.

Split	# Question	# Question-Passage Pairs
Training	174	972
Development	25	160
Test	52	-
All	251	1132

Table 1: Task A Dataset Distribution

3.1 System

Our implementation leverages the Sentence-Transformers framework (SBERT) (Reimers and Gurevych, 2019) to derive question and passage embeddings optimized for semantic similarity search. This provides an efficient method to match questions to relevant passages based on learned representations. SBERT provides a Siamese BERT network architecture optimized for semantic textual similarity. We used AraBERT, a BERT variant pre-trained on Arabic Wikipedia and news corpus.

To derive semantic vector representations of questions and Quran verses, the proposed passage retrieval approach trains a sentence embedding model, also known as a bi-encoder model. In order to achieve this, first, using **unsupervised** methods, AraBERT is fine-tuned on Quran passages to get sentence embedding. In the second step, the bi-encoder is trained on Mr. TyDi's Arabic dataset and Quran question-passage pairs using **supervised multi-task learning**.

3.1.1 Unsupervised Fine-Tuning: Learning Sentence Embedding

We experiment with TSDAE (Wang et al., 2021) and SimCSE (Gao et al., 2021) as the unsupervised training approach for encoding questions and passages.

TSDAE (Transformer-based Denoising Auto-Encoder) is a denoising Auto-Encoder trained to reconstruct corrupted passages, learning robust representations that capture semantic meaning.

SimCSE (Simple Contrastive Learning of Sentence Embeddings) is a contrastive self-supervised learning approach to derive passage embeddings. SimCSE is trained to predict a passage from itself, using only standard dropout as noise for data augmentation.

By transfer learning, these models learn robust passage representations that capture semantic meaning without the need for labeled data.

3.1.2 Supervised Fine-Tuning: Training Bi-Encoder using Question-Passage Pairs

After unsupervised fine-tuning convergence, a mean pooling and dense layer are added to the last layers of the bi-encoder. This bi-encoder is then fine-tuned end-to-end on Mr. TyDi and our question-passage pairs dataset. More specifically, the bi-encoder takes paired question and passage embeddings as input to predict relevance in a multi-task approach.

3.1.3 Model Specific Preparation

Models are trained with a combination of multiple negative ranking (Henderson et al., 2017), contrastive (Hadsell et al., 2006), and triplet (Dong and Shen, 2018) losses. As the models are trained in a multi-task manner, different loss functions are used for each dataset. A summary of the models is deprecated in Table 3. These three models were trained for 3 epochs with a batch size of 64, taking approximately 48 minutes in total on Nvidia GeForce RTX 3090 GPU.

As for the **Quranic question-passage** pairs, either a contrastive loss or triplet loss was incorporated:

- When using **contrastive loss**, we benefited from BM25 retrieval over the full corpus to mine negative passages for contrastive learning. More specifically, for each question in the training data, we first retrieve the top 1000 most relevant passages using BM25. We then label the ground truth

Model Name	Train Set		Development Set		Test Set	
	MAP	MRR	MAP	MRR	MAP	MRR
AraBERT-TSDAE-Contrastive	0.1502	0.3206	0.1365	**0.2613**	**0.0545**	**0.1581**
AraBERT-SimCSE-Contrastive	**0.6522**	**0.7646**	**0.1459**	0.2573	0.0315	0.1023
AraBERT-SimCSE-Triplet	0.5243	0.6580	0.1082	0.1693	0.0116	0.0356

Table 2: Task A MAP@10 and MRR@10 Results

passage associated with the question as positive examples (label 1). The BM25 retrieved passages that do not match any ground truth passages are used as hard negatives (label 0). Each <*question, positive passage, label=1*> and <*question, negative passage, label=0*> is added as a training example. By learning attempts to maximize similarity for positive pairs and minimize it for mined negatives.

• For **triplet loss**, similarly, BM25 is used to mine negatives but used in a different format and structure. Specifically, for each question, the top 100 BM25 retrieved passages are obtained. Then for each positive passage, negative passages are sampled to be used in forming of <*question, positive passage, negative passage*> triplets. Finally, for all of the question-passage pairs, multiple such triplets are created by pairing them with each possible negative passage from the BM25 results. Triplet loss optimizes the model to ensure the positive passage embedding is closer to the question than the negative passage.

For **Mr. TyDi**, the samples follow a format <*question, positive passage, negative passage*> and accordingly, multiple negative ranking loss function is used.

3.2 Results

To evaluate system performance, we report the official metrics of Mean Average Precision (MAP) and Mean Reciprocal Rank (MRR) on train, development, and test splits.

On the training set, our best-performing model is AraBERT-SimCSE-Contrastive, achieving a MAP@10 of 0.6522 and MRR@10 of 0.7646. Contrastive learning approaches generally outperform the triplet loss in our experiments. On the development set, AraBERT-SimCSE-Contrastive obtains the best MAP@10 of 0.1459 while AraBERT-TSDAE-Contrastive achieves the highest MRR@10 of 0.2613 score. Our top-performing model on the official test set is AraBERT-TSDAE-Contrastive, with a MAP@10 of 0.0545 and MRR@10 of 0.1581. Table 2 summarizes the full results on dataset distributions for top-10.

	ATC [3]	ASC [4]	AST [5]
TSDAE	✓	-	-
SimCSE	-	✓	✓
Denoising AE [6]	✓	-	-
Contrastive	✓	✓	-
Triplet	-	-	✓
Multiple Negative	✓	✓	✓
Quran Q-P [7]	✓	✓	✓
Mr. TyDi	✓	✓	✓

Table 3: Task A Models Summery

3.3 Discussion

Overall, our results demonstrate performance for passage retrieval on the Qur'anQA dataset. Observing the results of the development set indicates that the models are effective at retrieving all relevant passages containing name entities, which appeared in both the question and the passage. However, performance suffers for questions that are only relevant to a single obscure passage.

The unsupervised learning approaches of TSDAE and SimCSE both improve results compared to other methods we experimented with Arabic LMs. TSDAE in particular excels at ranking the relevant passages higher, leading to better MRR. This shows the value of its robust representations learned by reconstructing passages. The unsupervised fine-tuning allows the model to generalize better despite the limited size of the Quranic dataset.

4 Task B: Reading Comprehension

Given a Quranic passage that consists of consecutive verses in a specific Surah [8] of the Quran and a

[3] AraBERT-TSDAE-Contrastive (GYM_Run1)

[4] AraBERT-SimCSE-Contrastive (GYM_Run0)

[5] AraBERT-SimCSE-Triplet (GYM_Run2)

[6] Auto-Encoders

[7] Question-Passage

[8] A surah is a chapter in the Quran consisting of a set of verses revealed to the Islamic prophet Muhammad. There are 114 surahs in the Quran.

free-text question posed in MSA over that passage, a system is required to extract all answers to that question that is stated in the given passage.

4.1 Data

For Task B, we use Qur'anic Reading Comprehension Dataset (QRCD v1.2) (Malhas and Elsayed, 2022, 2020; Malhas et al., 2022) which consists of question-passage pairs combined with one or more annotated answers (15% of the questions have no answers). The dataset distribution is illustrated in Table 4.

Split	%	#Q	#Q-P	#Q-P-A
Training	70%	174	992	1179
Development	10%	25	163	220
Test	20%	51	431	-
All	100%	250	1586	1399

Table 4: Task B dataset distribution. #Q shows the number of questions. #Q-P shows the number of question-passage pairs. #Q-P-A shows the number of question-passage-answer triplets in the dataset.

4.2 System

Our solution for Task B is using the AraElectra-based model (Antoun et al., 2021) that is pre-trained on general domain Arabic language data. We propose two strategies for fine-tuning this model on the QRCD v1.2 dataset in addition to other complimentary datasets. The description of each model's training settings is summarized in Table 5. The hardware used is a GPU.1080Ti.xlarge with 31.3GB RAM. In the following sections, we briefly explain how we train each model.

4.3 Models Specifications

We chose **AraElectra-SQuADv2** (Ahmed, 2023a) model which is fine-tuned using the Arabic-SQuADv2.0 (Ahmed, 2023b) dataset. Specifically, AraElectra-SQuADv2 is the AutoModelForQuestionAnswering model from the transformers library in HuggingFace initialized with AraElectra model (Antoun et al., 2021). This model was trained on question-answer pairs, including unanswerable questions targeting QA task. We further fine-tuned this model using the QRCD v1.2 dataset (submitted as *GYM_Run0*).

We select **AraElectra-TyDiQA** (Ahmed et al., 2022) which fine-tuned on TyDi QA (Clark et al., 2020) dataset. Similarly, we fine-tuned this model on the QRCD v1.2 (submitted as *GYM_Run1*).

We incorporated **ensemble modeling** which is a machine-learning technique for combining multiple models in the prediction process. More specifically, by finding the top 10 answers using both AraElectra-SQuADv2 and AraElectra-TyDiQA, we can aggregate the given scores for all specified spans that are common among these runs/models (submitted as *GYM_ensemble*). The aggregation process works as follows:

I. We consider the output results of both AraElectra-SQuADv2 and AraElectra-TyDiQA models for each given question.
   - If the answers are the same, we sum the model's output scores.
   - Otherwise, we keep the answer without changing the score.

II. Finally, based on the newly calculated scores, we sort the output results of the two models and consider the top 10 outputs as the final output of the ensemble model.

	AraElec-SQuADv2	AraElec-TyDiQA
SQuADv2	✓	-
TyDiQA	-	✓
QRCD v1.2	✓	✓
Epoch	30	1
Batch Size	4	8
Max Seq Len [9]	256	256
Doc Strid [10]	64	64

Table 5: Task B train setting

4.4 Results

Reading Comprehension is evaluated with partial Average Precision (pAP), which accounts for partial matches and multiple answers. Our best configuration, AraElectra-SQuADv2, beats the task's baseline by 23.0% and reaches 48.5% pAP@10 on the dev set and 13.5% while achieving 46.1% pAP@10 on the test set (Table 6). Our experiments indicate that in comparison with other models, including an AraBERT, the AraElectra model gives better results on the Qur'anQA Task. Also, the use of the Arabic-SQuADv2.0 dataset, which is similar to QRCD v1.2, significantly improves the result.

4.5 Discussion

The results demonstrate that transfer learning from large Arabic NLP datasets (TyDiQA and SQuADv2) is an effective strategy for adapting models to Qur'anQA despite limited task-specific

[9] The maximum length of a feature.

[10] The authorized overlap between two part of the context when splitting is needed.

Model	Dev	Test
AraElectra-SQuADv2	**0.485**	**0.461**
Ensemble	0.481	0.458
AraElectra-TyDiQA	0.431	0.430
Baseline	0.255	0.326

Table 6: Task B pAP@10 result

training data. Pre-training on broad domains equips models like AraElectra with useful linguistic and semantic knowledge of Arabic that transfers well to Qur'anQA. Fine-tuning on the small QRCD v1.2 dataset provides the final layer of adaptation to handle Quranic syntax, terminology, and reasoning requirements.

Our best approach leverages Arabic SQuADv2 and is able to effectively identify questions with multiple answers and specify the start and end tokens of each answer. Among the answers, there were cases where the predicted answers overlap; hence, having a mechanism to handle overlapping predictions could improve the results. Additionally, it would be beneficial to optimize the model's confidence scores for predicting start and end tokens, such that falling below a threshold indicates no answer.

Overall, our results demonstrate promising multi-span extraction capabilities gained via pre-training on SQuADv2 data. However, enhancements to prediction post-processing and confidence modeling could further improve the handling of overlap and no-answer cases. This would move towards more human-like discernment of when extracted snippets represent valid or invalid answers.

Conclusion

This work demonstrates adapting LMs to Qur'anQA with limited data. Key techniques include unsupervised fine-tuning, negative sample extracting with BM25, multi-tasking, and transfer learning. For passage retrieval, unsupervised strategies like TSDAE and SimCSE improve ranking over training from scratch. In reading comprehension, leveraging Arabic SQuAD allows AraElectra to excel at span prediction despite scarce Quran annotations. Overall, leveraging additional datasets benefited models in both sub-tasks. We provide insights into tailoring state-of-the-art NLP techniques to learn specialized behavior for machine comprehension of the Quran's semantics given modest labeled data.

Limitations

The main constraint we faced was the lack of labeled data. To overcome this, we used similar non-Quranic datasets. While this affected the model's quality during training, it improved its ability to perform well on unseen data.

An important aspect to consider in the context of this research is the wealth of Tafsirs[11] available for the Quran, authored by religious scholars spanning different time periods and languages. These Tafsirs provide invaluable insights into the interpretations and nuances of the Quranic text, shedding light on the historical, linguistic, and cultural contexts in which the verses were revealed. The Quran, being a deeply layered and intricate scripture, often carries layers of meaning that extend beyond the literal words and Tafsirs help unravel these layers. Incorporating Tafsirs into the model's training data could enable it to better capture these nuanced interpretations and subtle connections, potentially leading to more accurate and contextually informed question-answering for vintage texts like the Quran.

Another challenge in passage retrieval we encountered was when the input question had no corresponding answer in the Quranic passages. In these cases, the model's performance suffered because we had to apply a threshold to the output scores, which were not fine-tuned specifically for this task. Additionally, the difference between the questions in Modern Standard Arabic (MSA) and the diverse variations of Quranic texts presented another challenge. One additional challenge we faced in this task was the lack of negative passages. To address this, we had to generate a set of negative passages using the BM25 method, as previously explained in detail. However, the quality of these negative passages plays a crucial role in the model's training. One approach we considered was to treat all passages, except the positive ones, as negatives. However, due to the imbalance between positive and negative samples and GPU limitations, we decided not to pursue this approach. But this approach can be examined in future work.

Acknowledgements

We'd like to thank the organizers for introducing this task. We are glad that we had the opportunity to engage more in the challenges and progress in the information retrieval task.

[11] Tafsir is Quranic exegesis that explains, interprets, contextualizes, or comments on Quran verses.

References

Basem Ahmed, Motaz Saad, and Eshrag A. Refaee. 2022. QQATeam at qur'an QA 2022: Fine-tunning Arabic QA models for qur'an QA task. 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 130–135, Marseille, France. European Language Resources Association.

Zeyad Ahmed. 2023a. Arabic Machine Reading Comprehension: Effective Models and Introducing Arabic-SQuAD v2.0. https://github.com/zeyadahmed10/Arabic-MRC, https://huggingface.co/ZeyadAhmed/AraElectra-Arabic-SQuADv2-QA. Original-date: 2021-11-04T18:03:17Z.

Zeyad Ahmed. 2023b. Arabic SQuAD v2.0 Dataset based on the popular SQuADv2.0 with unanswered questions for more challenging task. https://huggingface.co/datasets/ZeyadAhmed/Arabic-SQuADv2.0. Original-date: 2022-06-29T18:03:17Z.

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

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

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*.

Xingping Dong and Jianbing Shen. 2018. Triplet loss in siamese network for object tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Empirical Methods in Natural Language Processing (EMNLP)*.

R. Hadsell, S. Chopra, and Y. LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742.

Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun-Hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient natural language response suggestion for smart reply. *ArXiv*, abs/1705.00652.

Rana Malhas and Tamer Elsayed. 2020. Ayatec: Building a reusable verse-based test collection for arabic question answering on the holy qur'an. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 19(6).

Rana Malhas 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 *Proceedings of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT5) at the 13th Language Resources and Evaluation Conference (LREC 2022)*, pages 79–87.

Rana Malhas, Watheq Mansour, and Tamer Elsayed. 2023. Qur'an QA 2023 Shared Task: Overview of Passage Retrieval and Reading Comprehension Tasks over the Holy Qur'an. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore.

Rana R Malhas. 2023. *ARABIC QUESTION ANSWERING ON THE HOLY QUR'AN*. Ph.D. thesis.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Marwan N. Swar. 2007. *Mushaf Al-Tafseel Al-Mawdoo'ee*. Dar Al-Fajr Al-Islami, Damascus.

Kexin Wang, Nils Reimers, and Iryna Gurevych. 2021. TSDAE: Using transformer-based sequential denoising auto-encoderfor unsupervised sentence embedding learning. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 671–688, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xinyu Zhang, Xueguang Ma, Peng Shi, and Jimmy Lin. 2021. Mr. TyDi: A multi-lingual benchmark for dense retrieval. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 127–137, Punta Cana, Dominican Republic. Association for Computational Linguistics.

LKAU23 at Qur'an QA 2023: Using Transformer Models for Retrieving Passages and Finding Answers to Questions from the Qur'an

Sarah Alnefaie[1][2], **Abdullah N. Alsaleh**[1][2],
Eric Atwell[2], **Mohammed Ammar Alsalka**[2], **Abdulrahman Altahhan**[2]
King Abdulaziz University[1]
University of Leeds[2]
{scsaln, scanaa, e.s.atwell, m.a.alsalka, a.altahhan}@leeds.ac.uk

Abstract

The Qur'an QA 2023 shared task has two sub tasks: Passage Retrieval (PR) task and Machine Reading Comprehension (MRC) task. Our participation in the PR task was to further train several Arabic pre-trained models using a Sentence-Transformers architecture and to ensemble the best performing models. The results of the test set did not reflect the results of the development set. CL-AraBERT achieved the best results, with a 0.124 MAP. We also participate in the MRC task by further fine-tuning the base and large variants of AraBERT using Classical Arabic and Modern Standard Arabic datasets. Base AraBERT achieved the best result with the development set with a partial average precision (pAP) of 0.49, while it achieved 0.5 with the test set. In addition, we applied the ensemble approach of best performing models and post-processing steps to the final results. Our experiments with the development set showed that our proposed model achieved a 0.537 pAP. On the test set, our system obtained a pAP score of 0.49.

1 Introduction

The Arabic language poses many challenges in Natural Language Processing (NLP), including in the areas of Machine Reading Comprehension (MRC) and Passage Retrieval (PR). One of the most prominent recent NLP techniques applied to MRC and PR tasks in the Arabic language is pre-trained transformer-based models, which can achieve state-of-the-art performance (Alsubhi et al., 2021, 2022).

There are PR studies that use a dense approach based on pre-trained models (Karpukhin et al., 2020). This approach has outperformed traditional information retrieval, such as TF-IDF (Sammut and Webb, 2010) with Modern Standard Arabic (MSA; Alsubhi et al., 2022). To our knowledge, the dense approach has not been researched with Classical Arabic (CA). Therefore, our proposed system for Task A of the Qur'an QA 2023 shared tasks uses the dense approach by fine-tuning the Arabic pre-trained models and then ensemble the best scores. The idea of Task A is to build a system to return a list of Qur'anic passages that contain answers to a posed question/query (Malhas et al., 2023). However, the challenging aspect of this task is that there are some questions that do not have an answer in the Qur'an. The first research question **RQ1**: Does using the Arabic pre-trained models in PR for CA outperform the traditional approach such as BM25?

Most recent studies on the Qur'an MRC task have tended to use transformers-based models along with Qur'anic Reading Comprehension Dataset (QRCD) (Malhas et al., 2022). We noticed that they improved the performance of the systems using three approaches: (1) using an additional MSA and/or CA datasets in fine-tuning (Mostafa and Mohamed, 2022; Aftab and Malik, 2022), (2) constructing an ensemble of different BERT models (3) applying appropriate post-processing steps on the result of the final ranked list (ElKomy and Sarhan, 2022). To the best of our knowledge, no studies have combined these three approaches. Therefore, we applied the combination of those approaches for Task B of the Qur'an QA 2023 shared task. The goal of Task B was to build a model that took a Qur'anic passage and MSA question as input and extracted a ranked list of up to 10 answer spans to that question from the passage as output (Malhas et al., 2023). The new challenge in the second version of this task was that there were no answers to some questions. The second research questions **RQ2** in this paper is: Does the combination of fine-tuning the models with a large CA dataset and/or MSA dataset, ensembling these models and then applying post-processing steps improve the results?

The paper's structure is as follows: In Section 2, related work is presented. Section 3 describes the datasets. This is followed by Section 4, which

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 720–727
December 7, 2023 ©2023 Association for Computational Linguistics

explains the proposed models. In Section 5, the results are discussed. Finally, the paper provides a conclusion.

2 Related Work

2.1 Task A: Passage Retrieval

Karpukhin et al. (2020) proposed their dense passage retrieval (DPR) system using BERT base and uncased models. Their system applies dual encoders for the passages to be transformed into dimensional real-valued vectors and then applies an index for all passages for retrieval. The input query is then encoded and mapped into the dimensional vector space and passages are retrieved that are near the query vector. Their approach outperformed other multiple open-domain QA techniques on several QA datasets such as TriviaQA and SQuAD. Sachan et al. (2022) proposed the unsupervised passage re-ranker (UPR), in which the system utilizes zero-shot question generation for re-ranking passages in order to improve passage retrieval. It then computes the relevance scores over the generated question and sort the results. Their approach outperformed DPR (Karpukhin et al., 2020) on several datasets, such as SQuAD and TriviaQA. Finally, Alsubhi et al. (2022) proposed a multilingual DPR model that was fine-tuned on Arabic datasets. Their model outperformed TF-IDF on Arabic datasets, which were ARCD (Mozannar et al., 2019) and TyDiQA-GoldP (Clark et al., 2020).

2.2 Task B: Machine Reading Comprehension

Recently, several researchers have built an MRC system to answer questions about the Qur'an. All these studies used QRCD_v1.1 in the fine-tuning and evaluation phases (Malhas et al., 2022; Malhas and Elsayed, 2022). Some studies have proposed further fine-tuning the model using MSA datasets (Mostafa and Mohamed, 2022; Malhas and Elsayed, 2022). Mostafa and Mohamed (2022) developed the Arabic Qur'an MRC model by fine-tuning the AraELECTRA model using three MSA datasets: Ar-TyDi, Arabic-SQuAD and Arabic Reading Comprehension Dataset (ARCD). Their model achieved a 0.54 pRR, 0.52 F1@1 and 0.23 EM. Other studies have proposed fine-tuning the model using the CA dataset. Sleem et al. (2022) fine-tuned AraBERTv02 using the Arabic Al-Qur'an Question and Answer Corpus (AQQAC) (Alqahtani, 2019). This model achieved scores of 0.52 pRR, 0.5 F1@1 and 0.25 EM.

ElKomy and Sarhan (2022) recommends using the training and development sets of QRCD_v1.1 to fine-tune five different Arabic BERT models. They then used these five models individually to find the answers for the QRCD test set. To obtain good results, they implemented an ensemble approach for the results of these models. Finally, post-processing was applied to enhance the results. The results showed a pRR of 56.6, an EM of 26.8 and F1@1 of 0.50.

To the best of our knowledge, no study has been conducted on the impact of the combination of the following three factors in building the Arabic Qur'an MRC model: First, Arabic pre-trained models are fine-tuned using CA and MSA datasets. Second, the ensembling approach was applied to the results using the majority vote. Finally, the final list was refined through several post-processing steps.

3 Datasets

3.1 Task A: Passage Retrieval

The data were comprised of the Qur'anic passage collection (QPC) and questions from AyaTEC (Malhas and Elsayed, 2020). The QPC was developed by segmenting the Qur'an passages into topics, which resulted in 1,266 passages. There were 199 questions that were derived from the AyaTEC dataset. The Query Relevance Judgements (QRels) dataset contained 1,132 gold (answer-bearing) Qur'anic passages that answered the questions; these data were used in training and development sets. Finally, the distribution of the dataset was 70%, 10% and 20% for training, development and testing sets respectively.

3.2 Task B: Machine Reading Comprehension

In this study, we used three different datasets, as follows:

QRCD: QRCD_v1.2 consists of 1,399 question–passage–answer triplets in the training and development splits, as shown in Table 6. It was split 70%, 10%, and 20% for the training, development and test sets respectively (Malhas and Elsayed, 2022, 2020).

ARCD: It consists of 1,395 question–passage–answer triplets for Wikipedia passages (Mozannar et al., 2019).

Quran Question–Answer pairs (QUQA): It consists of 3,382 question–passage–answer triplets regarding the Arabic Holy Qur'an. This dataset was built using the available Qur'an AQQAC dataset

Model	Encoder	MAP	MRR
BM25 (Robertson and Zaragoza, 2009)	-	0.17	0.313
ArabicBERT (Safaya et al., 2020)	bi-encoder	**0.511**	**0.687**
	cross-encoder	0.292	0.452
CL-AraBERT (Malhas and Elsayed, 2022)	bi-encoder	0.489	0.7
	cross-encoder	0.318	0.481
AraBERT (Antoun et al.)	bi-encoder	0.461	0.662
	cross-encoder	0.351	0.54
CAMeL-BERT (Inoue et al., 2021)	bi-encoder	0.455	0.606
	cross-encoder	0.351	0.505
Ensemble ArabicBERT & CL-AraBERT	bi-encoder	**0.534**	**0.73**
Ensemble ArabicBERT & CL-AraBERT & CAMeL-BERT	bi-encoder	0.487	0.688
Ensemble ArabicBERT & CL-AraBERT & AraBERT	bi-encoder	0.485	0.682

Table 1: The results of the development set by BM25, individual Arabic pre-trained models and the ensemble method. MAP is the official evaluation metric. The cross-encoder is used for re-ranking the list of answers output by the bi-encoder.

(Alqahtani, 2019) and five available books. It is available in the Github repository. [1]

4 Proposed Models

4.1 Task A: Passage Retrieval

Sentence transformers, also known as Sentence-BERT (SBERT), introduced a bi-encoder that transforms a pair of sentences independently and maps them to a dense vector for efficient comparison when performing an information retrieval task (Thakur et al., 2021). Our proposed system uses a bi-encoder method for a semantic search task by further training Arabic pre-trained models with the QRCD_v1.1 (Malhas et al., 2022). We also used the cross-encoder “mmarco-mMiniLMv2-L12-H384-v1” [2] for re-ranking; however, it did not improve the performance of the individual models.

Training the Models: We trained a set of four models using the SBERT architecture with Arabic pre-trained models: ArabicBERT (Safaya et al., 2020), CAMeL-BERT (Inoue et al., 2021), AraBERT (Antoun et al.) and CL-AraBERT (Malhas and Elsayed, 2022). Two datasets were used for training the models: the training set of Task A and the QRCD_v1.1. Since most of the data were duplicated between the QRCD_v1.1 and the training set of PR task, we used the NoDuplicatesDataLoader function to remove any copies prior to training. We used the MultipleNegativesRankingLoss (MNRL) loss function, as it allowed for two similar or positive sentences without labels to be computed. Finally, the QPC dataset were encoded for each model. All the models used the following parameters: 5 epochs, a learning rate of 2e-5, max length 512 and batch size of 16.

Ensemble Approach: The ensemble method used for this task was to retrieve the top 20 answers from the Arabic pre-trained models. If the answer was found in all outputs, we then summed up the scores and divided by the number of models to obtain the average score. These answers were then put at a top-ranked list by descending order of averaged score. If there were remaining places in the ranked list, we added answers that had the highest scores out of all the models. Finally, we capped the ranked list at 10 answers [3].

4.2 Task B: Machine Reading Comprehension

The pre-trained transformer-based models were the basis of our methodology. As a first step, we fine-tuned all available Arabic pre-trained models with the QRCD_v1.2 training set. There were nine Arabic pre-trained models: AraBERT base, AraBERT large, CAMeL-BERT, ArabicBERT, CL-AraBERT, AraELECTRA (Antoun et al., 2021), MARBERT, ARBERT (Abdul-Mageed et al., 2021) and QARiB (Abdelali et al., 2021). When we conducted our experiments, we set the batch size to 8 for AraBERT large and 16 for the rest of the models, the number of epochs to 4, and the learning rate to 1e-4. We

[1] http://https://github.com/scsaln/HAQA-and-QUQA

[2] https://huggingface.co/nreimers/mmarco-mMiniLMv2-L12-H384-v1

[3] The code can be accessed here https://github.com/AlsalehAbdullah/Quran_PR_Task

Model	**QRCD**	**QRCD +QUQA**	**QRCD +ARCD**	**QRCD +QUQA +ARCD**
AraBERT Large	0.165	**0.482**	0.162	-
AraBERT Base	0.402	**0.458**	**0.433**	**0.49**
MARBERT	**0.326**	0.089	0.291	-
ARBERT	0.357	**0.38**	0.343	-
QARiB	**0.307**	0.301	0.278	-
CAMeL-BERT	0.401	**0.406**	0.362	-
ArabicBERT	**0.332**	0.330	0.313	-
AraELECTRA	**0.332**	0.248	0.218	-
CL-AraBERT	0.373	**0.383**	0.358	-

Table 2: The pAP@10 result of fine-tuned different Arabic pre-trained models by using different combinations of the datasets.

attempted to improve the performance using the following three optimisation approaches [4]:

Transfer Learning: We conducted three experiments using this approach. We further fine-tuned the models using different datasets. In the first experiment, we used the CA dataset QUQA. Second, the MSA ARCD was used. Finally, a combination of the QUQA dataset and ARCD was used only for the models that showed an improvement in performance when using one of these two datasets individually.

Ensemble Approach: We used majority voting among the models to produce the final ranked-list results. We took the top 20 answers with their scores for each question from each model. We then computed the total score for each answer. The total score was the sum of the scores obtained from the answers from all models. After that, we sorted the answers for each question based on the total score. Finally, we adopted the top 10 answers as the final ranked list.

Post-Processing: There were two issues when producing the ranked list: uninformative answers (as shown in Figure 1) and overlapping answers (as shown in Figure 2). The first issue was solved by removing these answers from the list. The second was overcome by applying a redundancy elimination algorithm (ElKomy and Sarhan, 2022).

5 Results and Discussion

5.1 Task A: Passage Retrieval

The official evaluation metric used for this task was mean average precision (MAP), but the mean reciprocal rank (MRR) was also reported.

Model	**pAP@10**
Ensemble **Vanilla** (All)	0.466
Ensemble **Vanilla** (Best)	0.517
Ensemble **POST** (Best)	**0.537**

Table 3: The results of the ensemble approach. Ensemble **Vanilla** (All) refers to applying the ensemble approach to all models. Ensemble **Vanilla** (Best) represents applying the ensemble approach to the best two performed models (the bert-large-arabertv02 and the bert-base-arabertv02). Ensemble **POST** (Best) refers to the **Vanilla** (Best) after applying the postprocessing step.

Validation: As for the validation results, the BM25 scored the lowest, with a 0.17 MAP. As for the pre-trained models, ArabicBERT performed the best of the individual models using a bi-encoder with a 0.511 MAP, while the ensemble of ArabicBERT and CL-AraBERT performed the best with the validation set with 0.534. Therefore, to address **RQ1**, the Arabic pre-trained models outperformed BM25 (See Table 1).

Testing: For the test set, we chose three methods based on their performances with the validation set. They were: ArabicBERT, CL-AraBERT and the ensemble of ArabicBERT and CL-AraBERT. The test set results did not reflect the performances on the validation set, as it can be seen in Table 4. CL-AraBERT performed the best with a 0.124 MAP while the performance of the ensemble method was a close second with a 0.117 MAP. The ensemble method and CL-AraBERT have successfully answered two questions with a perfect score of 1 MAP while 21 questions scored a 0 MAP. Some

[4] The code can be accessed here `https://github.com/scsaln/RC`

of these happened to be a no-answer, which the models have failed to identify.

5.2 Task B: Machine Reading Comprehension

The evaluation metric for Task B was partial average precision (pAP) (Malhas and Elsayed, 2022, 2020).

Validation: Column QRCD in Table 2 presents the results of the models when they were fine-tuned using only the QRCD dataset. The AraBET base model outperformed the other models with a 0.402 pAP@10.

First, we addressed **RQ2**, which was related to whether the combination of the three factors enhanced the performance of the Qur'an MRC models. The first factor further fine-tuned the models using the CA dataset QUQA and/or MSA ARCD. The results are shown in columns 'QRCD + QUQA', 'QRCD + ARCD' and 'QRCD + QUQA + ARCD' in Table 2. There are three interesting observations in the results. First, using the QUQA dataset led to improvements in more than half of the models. The best score was the pAP@10 of 0.482, obtained by AraBERT large. Second, when we trained the model using the ARCD dataset it enhanced the performance of the AraBERT base model only with 0.433 pAP@10. Third, using QUQA and ARCD at the same time to train the AraBERT base improved results with 0.49 pAP@10 compared to using QUQA and ARCD separately. For the second factor, we used the ensemble method for all the models; however, this approach did not yield the best performance with a result of 0.466 pAP@10. We then ensembled two of the best performing individual models, which were AraBERT base and AraBERT large. The results outperformed the other models with 0.517. For the third factor, we note that the post-processing step improved the results based on the Ensemble '**POST** (Best)' row shown in Table 3.

Testing: For the test set, we chose two methods based on the performance of the development set. They were (1) the ensemble of AraBERT base and AraBERT large with post-processing and (2) the AraBERT base model. The ensemble with the post-processing approach achieved a 0.498 pAP@10, while the AraBERT base model achieved the best performance with a 0.5 pAP@10, as it can be seen in Table 5.

Model	**MAP**	**MRR**
Ensemble	0.117	0.36
ArabicBERT	0.07	0.20
CL-AraBERT	**0.124**	0.375

Table 4: Test set results of Task A.

Model	**pAP@10**
Ensemble **POST** (Best)	0.498
AraBERT Base	0.5

Table 5: Test set results of Task B.

When we analysed the model answers to questions from the development set, we identified the following: The model worked as a simple match model. When part of the passage contained words from the question, it retrieved this part as an answer to the question, even though the meaning of this part did not answer the question (see Figure 3). Therefore, the system failed to predict the correct answer when the answer has semantically similar words to the question (see Figure 4).

6 Conclusion

This paper presented our contributions to Task A: PR and Task B: MRC of the Qur'an QA 2023 shared task. Our proposed PR method was to train several Arabic pre-trained models with QRCD dataset using SBERT architecture and then ensemble the combination of these models. The ensemble method did not yield the best performance with the test set, although it had the best score with the development set. CL-AraBERT achieved the best results with a 0.124 MAP. Our proposed MRC system is based on combining the transfer learning and ensemble approaches for the best-performing models. Initially, we fine-tuned nine different Arabic pre-trained models using different data collections. We then applied the ensemble approach to the two best-performing models. Finally, we implemented appropriate post-processing steps. The combination of the base and large variants of AraBERT achieved the best results on the development set, with a 0.537 pAP@10. The second-highest score was achieved by base AraBERT with a 0.49 pAP@10. The results of applying these two models to the test set showed that the base AraBERT model was the best with a score of 0.5 pAP@10, while the ensemble model achieved a score of 0.49 pAP@10.

Limitations

One of the most important factors affecting the performance of pretraining models is the size of the dataset. The size of the dataset used in the training in this study is miniscule compared to the size of the data available in the English language. Therefore, we noticed weak performance of the models in Arabic. There is an urgent need to build large data collections in Arabic.

Acknowledgement

The authors would like to express their deepest gratitude to King Abdulaziz University for the support. We thank the reviewers for their constructive comments.

References

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

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2021. ARBERT & MARBERT: Deep bidirectional transformers for Arabic. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7088–7105, Online. Association for Computational Linguistics.

Esha Aftab and Muhammad Kamran Malik. 2022. erock at qur'an qa 2022: Contemporary deep neural networks for qur'an based reading comprehension question answers. 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 96–103.

Mohammad Mushabbab A Alqahtani. 2019. *Quranic Arabic semantic search model based on ontology of concepts*. Ph.D. thesis, University of Leeds.

Kholoud Alsubhi, Amani Jamal, and Areej Alhothali. 2021. Pre-trained transformer-based approach for arabic question answering: A comparative study. *arXiv preprint arXiv:2111.05671*.

Kholoud Alsubhi, Amani Jamal, and Areej M. Alhothali. 2022. Deep learning-based approach for arabic open domain question answering. *PeerJ Computer Science*, 8.

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

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

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. TyDi QA: A Benchmark for Information-Seeking Question Answering in Typologically Diverse Languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

Mohammed ElKomy and Amany M Sarhan. 2022. Tce at qur'an qa 2022: Arabic language question answering over holy qur'an using a post-processed ensemble of bert-based models. *arXiv preprint arXiv:2206.01550*.

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.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Rana Malhas and Tamer Elsayed. 2020. Ayatec: Building a reusable verse-based test collection for arabic question answering on the holy qur'an. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 19(6).

Rana Malhas 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 the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore.

Ali Mostafa and Omar Mohamed. 2022. Gof at qur'an qa 2022: Towards an efficient question answering for the holy qu'ran in the arabic language using deep learning-based approach. In *Proceedinsg of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools with Shared Tasks on Qur'an QA and Fine-Grained Hate Speech Detection*, pages 104–111.

Hussein Mozannar, Karl El Hajal, Elie Maamary, and Hazem Hajj. 2019. Neural arabic question answering. *arXiv preprint arXiv:1906.05394*.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Devendra Sachan, Mike Lewis, Mandar Joshi, Armen Aghajanyan, Wen-tau Yih, Joelle Pineau, and Luke Zettlemoyer. 2022. Improving passage retrieval with zero-shot question generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3781–3797, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ali Safaya, Moutasem Abdullatif, and Deniz Yuret. 2020. KUISAIL at SemEval-2020 task 12: BERT-CNN for offensive speech identification in social media. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 2054–2059, Barcelona (online). International Committee for Computational Linguistics.

Claude Sammut and Geoffrey I. Webb, editors. 2010. *TF–IDF*, pages 986–987. Springer US, Boston, MA.

Ahmed Sleem, Eman Mohammed lotfy Elrefai, Marwa Mohammed Matar, and Haq Nawaz. 2022. Stars at qur'an qa 2022: Building automatic extractive question answering systems for the holy qur'an with transformer models and releasing a new dataset. In *Proceedinsg of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools with Shared Tasks on Qur'an QA and Fine-Grained Hate Speech Detection*, pages 146–153.

Nandan Thakur, Nils Reimers, Johannes Daxenberger, and Iryna Gurevych. 2021. Augmented SBERT: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 296–310, Online. Association for Computational Linguistics.

A QRCD Dataset Distribution

In this appendix, Table 6 presents the distribution of the dataset.

Dataset	**# Q**	**# Q-P Pairs**	**# Q-P-A Triplets**
Training	174	992	1179
Development	25	163	220

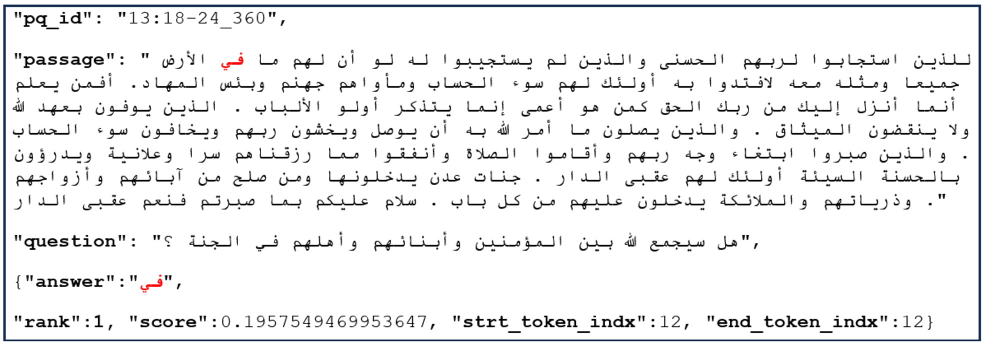
Table 6: QRCD distribution. # Q shows the number of the questions, # Q-P Pairs show the number of the questions–passage pairs and # Q-P-A Triplets show number of questions–passage–answers triplets.

B The problems of the list of answers

In this appendix, Figure 1 and Figure 2 present the problems we encountered in the list of answers.

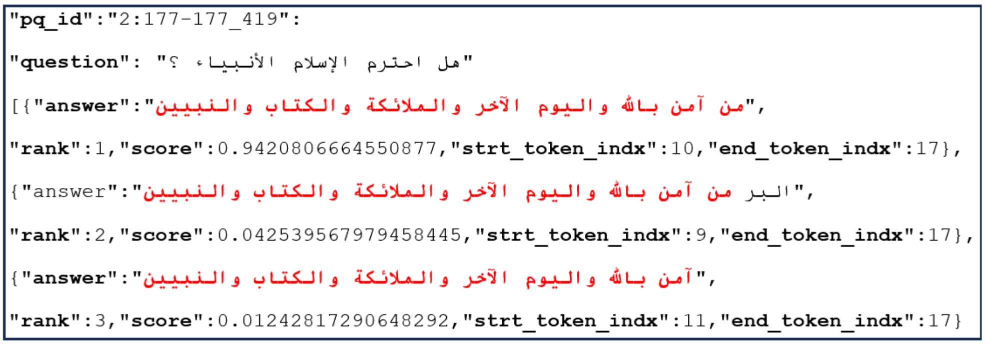
C The Analysis and Discussion of Task B

In this appendix, Figure 3 and Figure 4 present the discussion of Task B Machine Reading Comprehension.



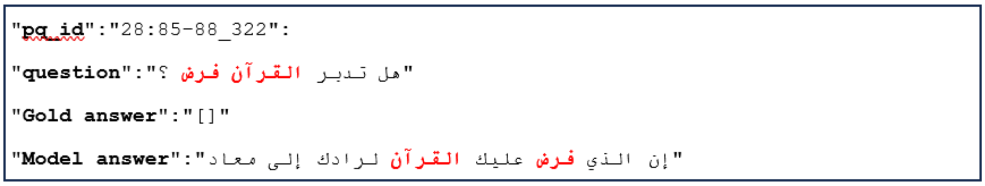
```
"pq_id": "13:18-24_360",

"passage": " للذين استجابوا لربهم الحسنى والذين لم يستجيبوا له لو أن لهم ما في الأرض
جميعا ومثله معه لافتدوا به أولئك لهم سوء الحساب ومأواهم جهنم وبئس المهاد. أفمن يعلم
أنما أنزل إليك من ربك الحق كمن هو أعمى إنما يتذكر أولو الألباب . الذين يوفون بعهد الله
ولا ينقضون الميثاق . والذين يصلون ما أمر الله به أن يوصل ويخشون ربهم ويخافون سوء الحساب
. والذين صبروا ابتغاء وجه ربهم وأقاموا الصلاة وأنفقوا مما رزقناهم سرا وعلانية ويدرؤون
بالحسنة السيئة أولئك لهم عقبى الدار . جنات عدن يدخلونها ومن صلح من آبائهم وأزواجهم
وذرياتهم والملائكة يدخلون عليهم من كل باب . سلام عليكم بما صبرتم فنعم عقبى الدار ."

"question": "هل سيجمع الله بين المؤمنين وأبنائهم وأهلهم في الجنة ؟",

{"answer":"في",

"rank":1, "score":0.1957549469953647, "strt_token_indx":12, "end_token_indx":12}
```

Figure 1: Example of an uninformative answer.



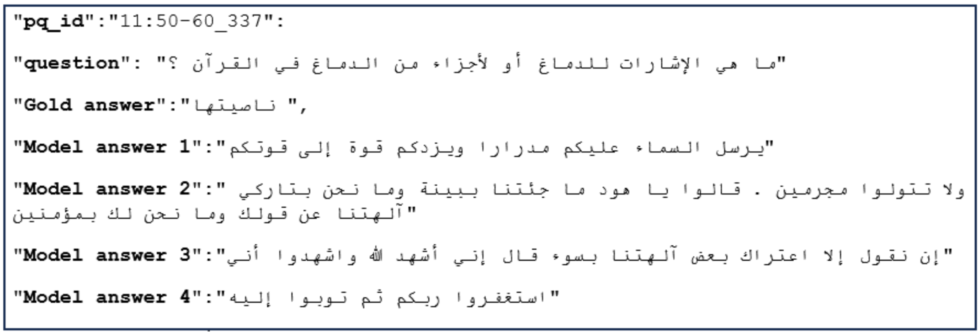
```
"pq_id":"2:177-177_419":

"question": "هل احترم الإسلام الأنبياء ؟"

[{"answer":"من آمن بالله واليوم الآخر والملائكة والكتاب والنبيين",

"rank":1,"score":0.9420806664550877,"strt_token_indx":10,"end_token_indx":17},

{"answer":"البر من آمن بالله واليوم الآخر والملائكة والكتاب والنبيين",

"rank":2,"score":0.042539567979458445,"strt_token_indx":9,"end_token_indx":17},

{"answer":"آمن بالله واليوم الآخر والملائكة والكتاب والنبيين",

"rank":3,"score":0.01242817290648292,"strt_token_indx":11,"end_token_indx":17}
```

Figure 2: Example of repeated answers.



```
"pq_id":"28:85-88_322":

"question":"هل تدبر القرآن فرض ؟"

"Gold answer":"[]"

"Model answer":"إن الذي فرض عليك القرآن لرادك إلى معاد"
```

Figure 3: Example 1 of an incorrect answer.



```
"pq_id":"11:50-60_337":

"question": "ما هي الإشارات للدماغ أو لأجزاء من الدماغ في القرآن ؟"

"Gold answer":"ناصيتها ",

"Model answer 1":"يرسل السماء عليكم مدرارا ويزدكم قوة إلى قوتكم"

"Model answer 2":" ولا تتولوا مجرمين . قالوا يا هود ما جئتنا ببينة وما نحن بتاركي
آلهتنا عن قولك وما نحن لك بمؤمنين"

"Model answer 3":"إن نقول إلا اعتراك بعض آلهتنا بسوء قال إني أشهد الله واشهدوا أني"

"Model answer 4":"استغفروا ربكم ثم توبوا إليه"
```

Figure 4: Example 2 of an incorrect answer.

TCE at Qur'an QA 2023 Shared Task: Low Resource Enhanced Transformer-based Ensemble Approach for Qur'anic QA

Mohammed ElKomy, Amany Sarhan
Department of Computer Engineering, Faculty of Engineering
Tanta University, Egypt
{mohammed.a.elkomy,amany_sarhan}@f-eng.tanta.edu.eg

Abstract

In this paper, we present our approach to tackle Qur'an QA 2023 shared tasks **A** and **B**. To address the challenge of low-resourced training data, we rely on transfer learning together with a voting ensemble to improve prediction stability across multiple runs. Additionally, we employ different architectures and learning mechanisms for a range of Arabic pre-trained transformer-based models for both tasks. To identify unanswerable questions, we propose using a thresholding mechanism. Our top-performing systems greatly surpass the baseline performance on the hidden split, achieving a MAP score of 25.05% for task **A** and a partial Average Precision (pAP) of 57.11% for task **B**.

1 Introduction

Ad hoc search is a fundamental task in Information Retrieval (IR) and serves as the foundation for numerous Question Answering (QA) systems and search engines. Machine Reading Comprehension (MRC) is a long-standing endeavor in Natural language processing (NLP) and plays a significant role in the framework of text-based QA systems. The emergence of **B**idirectional **E**ncoder **R**epresentations from **T**ransformers (BERT) and its family of transformer-based pre-trained language models (LM) have revolutionized the landscape of transfer learning systems for NLP and IR as a whole (Yates et al., 2021; Bashir et al., 2021).

Arabic is widely spoken in the Middle East and North Africa, and among Muslims worldwide. Arabic is known for its extensive inflectional and derivational features. It has three main variants: Classical Arabic (CA), Modern standard Arabic (MSA), and Dialectal Arabic (DA).

Qur'an QA 2023 shared task **A** is a passage retrieval task organized to engage the community in conducting ad hoc search over the Holy Qur'an (MALHAS, 2023; Malhas and Elsayed, 2020). While Qur'an QA 2023 shared task **B** is a ranking-based MRC over the Holy Qur'an, which is the second version of Qur'an QA 2022 shared task (Malhas et al., 2022; MALHAS, 2023).

This paper presents our approaches to solve the two tasks **A** and **B**. For task **A**, we explore both dual-encoders and cross-encoders for ad hoc search (Yates et al., 2021). For task **B**, we investigate LMs for extractive QA using two learning methods (Devlin et al., 2019). For both tasks, we utilize various pre-trained Arabic LM variants. Moreover, we adopt external Arabic resources in our fine-tuning setups (MALHAS, 2023). Finally, we employ an ensemble-based approach to account for inconsistencies among multiple runs. We contribute to the NLP community by releasing our experiment codes and trained LMs to GitHub [1].

In this work, we address the following research questions [2]:
RQ1: What is the impact of using external resources to perform pipelined fine-tuning?
RQ2: How does ensemble learning improve the performance obtained?
RQ3: What is the effect of thresholding on zero-answer questions?
RQ4^{A}: What is the impact of hard negatives on the dual-encoders approach?
RQ5^{B}: What is the impact of multi answer loss method on multi-answer cases?
RQ6^{B}: How is post-processing essential for ranking-based extractive question answering?

The structure of our paper is as follows: Sections 2 and 3 provide an overview of the datasets used in our study. In Section 4, we present the system design and implementation details for both tasks. The main results for both tasks are presented in Section 5. Section 6 focuses on the analysis and discussion of our research questions **RQ**s. Finally, Section 7 concludes our work.

[1] https://github.com/mohammed-elkomy/quran-qa

[2] A superscript at the end of a RQ refers to one of the tasks. No superscript means the RQ applies for both tasks.

December 7, 2023 ©2023 Association for Computational Linguistics

Split		Training	Development
# Question-passage relevance pairs		972	160
# Questions	Multi-answer	105 (60%)	15 (60%)
	Single-answer	43 (25%)	6 (24%)
	Zero-answer	26 (15%)	4 (16%)
	Total	174	25

Table 1: Task **A** dataset relevance pairs distribution across training and development splits. We also include the distribution of answer types per split.

Split		Training	Development
# Question-passage-answer Triplets		1179	220
# Question-passage Pairs	Multi-answer	134 (14%)	29 (18%)
	Single-answer	806 (81%)	124 (76%)
	Zero-answer	52 (5%)	10 (6%)
	Total	992	163

Table 2: Task **B** dataset pairs and triplets distribution across training and development splits. For questions-passage pairs, we show the distribution of answer types.

2 Task A Dataset Details

Qur'an QA 2023 shared task **A** serves as a test collection for the ad hoc retrieval task. The divine text is divided into segments known as the Thematic Qur'an Passage Collection (QPC), where logical segments are formed based on common themes found among consecutive Qur'anic verses (Malhas et al., 2023; Swar, 2007). In this task, systems are required to provide responses to user questions in MSA by retrieving relevant passages from the QPC when possible. This suggests there is a language gap between the questions and the passages, as the passages are in CA. Table 1 presents the distribution of the dataset across the training and development splits. The majority of questions in the dataset are multi-answer questions, meaning that systems can only receive full credit if they are able to identify all relevant passages for these queries. Additionally, Table 1 provides information on zero-answer questions, which are unanswerable questions from the entire Qur'an. (More information about the dataset distribution of topics in Appendix A.1)

Task **A** is evaluated as a ranking task using the standard mean Average Precision (MAP) metric. (Additional information about the evaluation process including zero-answers cases can be found in Appendix A.2)

3 Task B Dataset Details

Qur'an QA 2023 shared task **B** is a ranking-based SQuADv2.0-like MRC over the Holy Qur'an, which extends to the Qur'an QA 2022 (Malhas et al., 2022; Rajpurkar et al., 2016). The dataset is also referred to as Qur'an reading comprehension dataset v1.2 (QRCDv1.2). The same questions from task **A** are organized as answer span extraction task from relevant passages (Malhas and Elsayed, 2020; Malhas et al., 2022). (See the dataset distribution of topics in Appendix A.1)

Table 2 depicts the distribution of dataset pairs and triplets across the training and development splits. In addition, the table presents the distribution of answer types for the dataset pairs.
Although zero-answer questions account for 15% of the questions in task **A** test collection, they only contribute to 5% of the question-passage pairs in task **B**. Furthermore, task **B** has a limited number of unique questions in comparison to their corresponding question-passage pairs as seen from Tables 1 and 2, respectively. As a consequence, task **B** can have repeated questions and passages among different samples and can be even *leaked* among training and development splits (Keleg and Magdy, 2022). Keleg and Magdy (2022) analyzed this phenomenon and identified sources of *leakage* in Qur'an reading comprehension dataset v1.1 (QRCDv1.1). In QRCDv1.1, leakage is defined as the presence of passages, questions, or answers that are shared among multiple samples (Keleg and Magdy, 2022). This can lead to LMs memorizing or overfitting leaked samples (Keleg and Magdy, 2022). Keleg and Magdy (2022) categorized QRCDv1.1 into four distinct and mutually exclusive categories based on the type of leakage: pairs of passage-question, passage-answer, or just questions. (For more information about leakage in task **B**, please refer to Appendix A.4)

We extend the analysis made by Keleg and Magdy (2022) for QRCDv1.2. Our main observation is that 90% of the samples with no answer belong to the trivial leakage group called $D_{(1)}$. This group refers to samples with duplicate passage-answer or question-answer pairs. This indicates that zero-answer questions are not just less prevalent in task **B** but also present a greater challenge in terms of generalization. Given the four groups defined by Keleg and Magdy (2022), they proposed a data re-splitting mechanism for QRCDv1.1 called *faithful* splits. In this work, we extend their re-splitting approach and create faithful splits for QRCDv1.2. (Please refer to Appendix A.4 for more details about faithful splitting)

Task **B** is evaluated as a ranking task as well, using a recently proposed measure called pAP (Malhas and Elsayed, 2020; MALHAS, 2023). (More details about this measure and zero-answer sample evaluation can be found in Appendix A.3)

4 System Design

In this work, we fine-tune a variety of pre-trained Arabic LMs, namely AraBERTv0.2-base (Antoun et al., 2020), CAMeLBERT-CA (Inoue et al., 2021), and AraELECTRA (Antoun et al., 2021). We utilize transfer learning and ensemble learning for both tasks. To determine zero-answer cases, we apply a thresholding mechanism. (Additional information on transfer learning and ensemble learning can be found in Appendices B and C, respectively)

4.1 Task A Architecture

We examine two distinct approaches for neural ranking in ad-hoc search: dual-encoders and cross-encoders approaches (Yates et al., 2021).

In dual-encoders, documents and queries are encoded separately into dense vectors, which are then compared using a metric learning function, such as cosine distance. We utilize **S**table **T**raining **A**lgorithm for dense **R**etrieval (STAR) with a batch size of 16 queries to train our dense retrievers (Zhan et al., 2021; Yates et al., 2021).

In contrast cross-encoders involve encoding positive and negative pairs of documents and questions, assigning a relevance score. This method packs a document and a question into a single input for a sentence similarity LM (Yates et al., 2021). Both methods require negative relevance signals during training. (Please refer to Figures 4a and 4b in Appendix for both approaches. Additionally, see Appendix D for more details about negative selection criteria and zero-answer prediction)

Although cross-encoders have a higher computational overhead compared to dual-encoders when used for ranking, the former has a quadratic complexity while the latter has a linear complexity. However, both methods are still feasible for low-resource datasets (Yates et al., 2021). In both approaches, we utilize the cumulative predicted scores of the top K documents to calculate the likelihood of each question having an answer. We then apply a threshold ζ to identify zero-answer questions.

4.2 Task B Architecture

We fine-tune pre-trained LMs for span prediction as in SQuADv2.0 (Rajpurkar et al., 2018; Devlin et al., 2019). We use two different fine-tuning methods: First answer loss (FAL) and Multi answer loss (MAL). The FAL method focuses on optimizing for the first answer in the ground truth answers, which is the default approach in standard span prediction implementations for SQuAD (Devlin et al., 2019; Wolf et al., 2019). In contrast, MAL optimizes for multiple answers simultaneously for the multi-answer samples in QRCDv1.2. This helps prevent the trained systems from being overly confident in a single span and distributes the predicted probability among different spans. (Refer to Appendix E for more information about these learning methods)

It is worth noting that raw predictions from span prediction LMs are suboptimal for ranking MRC, as many of them have overlapping content. To address this, we follow a post-processing mechanism proposed by Elkomy and Sarhan (2022). (See Appendix E.1 for implementation details)

Similar to task **A**, we perform thresholding by a hyperparameter ζ to determine zero-answer samples using LM null answer **[CLS]** token probability (Rajpurkar et al., 2018; Devlin et al., 2019). (See Appendix E.2 for more details on zero-answer cases)

5 Results

The results tables for both tasks use the following notational format: We use short forms to refer to combinations of LMs and their fine-tuning approaches using superscripts and subscripts.

The subscripts $\sim$ and $\approx$ denote direct fine-tuning and pipelined fine-tuning, respectively. Additionally, the arrows in model names subscripts indicate the stages of pipelined fine-tuning, with the learning resources names listed. Superscripts are used to denote the architectures employed for task **A** and the learning methods for task **B**.

Tables 3 and 4 present our detailed results on the development split for both tasks for single and self-ensemble models. Table 3 shows the results for cross encoder and dual-encoders for task **A**. Our best single model, (ARB$^{\otimes}_{\approx}$), achieved a MAP of 34.83% and an MRR of 47.09%. (ARB$^{\otimes}_{\approx}$) self-ensemble achieved the best MAP of 36.70%. Table 3 also presents the R@10 and R@100 metrics. This represents the upper bound on the reranking

Short Form	Systems	Single Model								Self Ensemble	
		MAP	**MRR**	**R@10**	**R@100**	$\textbf{MAP}^{\star}_{\zeta}$	**MAP (Question Type)**			**MAP**	$\textbf{MAP}^{\star}_{\zeta}$
							Zero	**Single**	**Multi**		
Lexical Baseline											
BM$_{\sim}$	BM25	18.43	26.40	19.98	19.98	26.40	25.00	16.67	17.39	N/A	N/A
Dual-encoder											
ARB$^{\circledcirc}_{\sim}$	AraBERTv0.2-base$_{\textit{TASK A+ Random Neg}}$	20.02	42.87	29.72	48.23	20.02	0.00	35.42	19.20	N/A	N/A
ARB$^{\circledcirc}_{\approx}$	AraBERTv0.2-base$_{\textit{TASK A+ Hard Neg}}$	**24.44**	35.17	36.09	43.96	24.44	0.00	45.00	22.73	N/A	N/A
Cross Encoder											
ELC$^{\otimes}_{\sim}$	AraELECTRA$_{\textit{TASK A}}$	8.96	16.51	19.13	42.49	16.48	3.00	10.32	10.01	12.18	16.18
ELC$^{\otimes}_{\approx}$	AraELECTRA$_{\textit{TyDi QA}_{\text{AR}}\rightarrow \textit{Tafseer}\rightarrow \textit{TASK A}}$	26.60	41.61	38.52	59.19	31.91	19.00	38.31	23.94	29.13	36.56
CAM$^{\otimes}_{\sim}$	CAMeLBERT-CA$_{\textit{TASK A}}$	23.16	33.52	37.06	55.12	27.45	13.00	36.92	20.36	27.57	32.02
CAM$^{\otimes}_{\approx}$	CAMeLBERT-CA$_{\textit{TyDi QA}_{\text{AR}}\rightarrow \textit{Tafseer}\rightarrow \textit{TASK A}}$	29.34	42.17	39.93	57.23	33.81	18.00	**51.40**	23.54	32.77	36.77
ARB$^{\otimes}_{\sim}$	AraBERTv0.2-base$_{\textit{TASK A}}$	31.76	41.93	**46.55**	**62.71**	34.27	**46.00**	28.16	**29.41**	36.09	36.87
ARB$^{\otimes}_{\approx}$	AraBERTv0.2-base$_{\textit{TyDi QA}_{\text{AR}}\rightarrow \textit{Tafseer}\rightarrow \textit{TASK A}}$	**34.83**	**47.09**	39.99	60.82	**37.55**	43.00	46.22	28.10	**36.70**	**40.70**

Table 3: Dev split evaluation results for task **A**. **MAP** means ζ is set to mark 15% of questions as unanswerable. ★ accompanied by ζ refers to applying the best ζ (see Appendix F). Average performance is reported for multiple runs of single models. Superscripts $\circledcirc$ and $\otimes$ in short form refer to dual-encoder and cross encoder, respectively. Subscripts $\sim$ and $\approx$ denote direct fine-tuning and pipelined fine-tuning, respectively.

stage performance that we can obtain (Yates et al., 2021).

Table 4 summarizes the results for task **B**. Our best performing model over the standard split, (ELC$^{M}_{\approx}$), attained a pAP of 53.36% and 55.21% for single model and self-ensemble models, respectively. Table 4 also presents results for the faithful validation split we defined previously. (ARB$^{M}_{\approx}$) is our best performing single model for the faithful split, achieving a pAP score of 54.19%.

Both tables present comprehensive results for different question types, as well as the outcomes for a manually set threshold ζ and $\zeta^{\star}$, i.e., the threshold that yields the best performance.
(See Appendix F for more details about ζ and optimal ζ selection)

Considering the question types , experiments of (ARB$^{\otimes}_{\sim}$) and (ARB$^{\otimes}_{\approx}$) obtains the best MAP performance for zero-answer and multi-answer questions for task **A**.

With regard to the hidden split, Tables 5 and 6 provide a summary of our official submissions.

In task **A**, as shown in Table 5, we made 3 cross-encoder submissions: MIX$^{\otimes}_{\approx}$, which is an ensemble combining runs from CAM$^{\otimes}_{\approx}$ and ARB$^{\otimes}_{\approx}$ cross encoders. MIX$^{\otimes}_{\approx}$ achieved a MAP of 25.05%. In comparison, the TF-IDF baseline only achieved a MAP of 9.03%.

On the other hand, in task **B**, we experimented with our two best performing models in Table 4. As shown in Table 6, (ARB$^{M}_{\approx}$) outperformed (ELC$^{M}_{\approx}$) with a pAP of 57.11%. This result is consistent with the findings from the faithful validation split (Keleg and Magdy, 2022) in Table 4 for (ARB$^{M}_{\approx}$) and (ELC$^{M}_{\approx}$). Specifically, the MAL method outperformed FAL for all of our models in the faithful validation split (underlined in Table 4).

6 Analysis and Discussion

Regarding **RQ1**, external resources always bring significant improvements to the same LM for both tasks. For task **A**, we have three stages of fine-tuning as indicated by arrows in Table 3. For example, when (ELC$^{\otimes}_{\sim}$) is fine-tuned with external resources into (ELC$^{\otimes}_{\approx}$) the MAP performance improves from 8.96% to 26.60% for single models as in Table 3. In similar fashion for task **B**, (ELC$^{M}_{\approx}$) outperforms (ELC$^{M}_{\sim}$) by almost 13% for the standard split in Table 4.

To answer our **RQ2**, ensemble learning consistently outperforms single models for both tasks. For instance, (CAM$^{\otimes}_{\approx}$) ensemble surpasses its single model by 3.5% for the MAP metric for task **A**. Similarly, (ELC$^{M}_{\approx}$) ensemble outperforms its corresponding single model by almost a pAP of 2% for task **B**.

With regard to **RQ3**, the hyperparameter ζ affects the zero answer type evaluation scores for both tasks. We make best use of the available data by employing a quantile method to determine the threshold ζ for both tasks. However, (ARB$^{\otimes}_{\approx}$) model MAP performance improves by 3% when the optimal $\zeta^{\star}$ is employed for task **A**. This suggests that there is a room for improvement for the

Short Form	Systems		Single Model								Self Ensemble Model		
	Model	Method	Faithful		Standard Development Split								
			pAP	pAP$_{Post}$	pAP	pAP$_{Post}$	pAP$^{\star}_{\zeta}$	pAP (Sample Type) Zero	 Single	 Multi	pAP	pAP$_{Post}$	pAP$^{\star}_{\zeta}$
ELC$^{F}_{\sim}$	AraELECTRA *TASK B*	FAL	34.97	41.23	38.27	44.40	39.26	18.67	41.51	31.18	41.16	46.50	41.72
ELC$^{M}_{\sim}$		MAL	$\underline{37.44}$	$\underline{42.63}$	$\underline{40.55}$	$\underline{45.56}$	41.48	14.67	43.69	$\underline{36.04}$	42.01	47.21	43.90
ELC$^{F}_{\approx}$	AraELECTRA *TyDi QA*$_{\text{AR}}$→*TASK B*	FAL	52.76	$\underline{55.45}$	49.76	53.70	51.99	10.33	54.36	43.69	50.66	55.35	52.75
ELC$^{M}_{\approx}$		MAL	$\underline{53.15}$	55.43	$\underline{\mathbf{53.36}}$	$\underline{\mathbf{56.42}}$	**55.10**	18.33	**56.61**	$\underline{\mathbf{51.55}}$	**55.21**	**58.38**	**57.05**
CAM$^{F}_{\sim}$	CAMeLBERT-CA *TASK B*	FAL	41.45	45.76	37.63	42.04	38.36	11.00	40.83	33.13	42.51	45.50	43.18
CAM$^{M}_{\sim}$		MAL	$\underline{43.54}$	$\underline{47.36}$	$\underline{38.57}$	$\underline{43.38}$	39.38	12.67	40.52	$\underline{39.20}$	41.66	45.39	43.80
CAM$^{F}_{\approx}$	CAMeLBERT-CA *TyDi QA*$_{\text{AR}}$→*TASK B*	FAL	50.64	53.12	$\underline{41.59}$	$\underline{46.50}$	42.39	13.67	44.36	39.39	47.03	49.37	47.12
CAM$^{M}_{\approx}$		MAL	$\underline{52.14}$	$\underline{54.01}$	40.08	44.80	41.30	15.00	41.61	$\underline{42.18}$	42.75	46.87	44.23
ARB$^{F}_{\sim}$	AraBERTv0.2-base *TASK B*	FAL	44.81	48.93	45.66	$\underline{49.34}$	46.60	23.67	49.29	37.74	49.38	53.05	50.01
ARB$^{M}_{\sim}$		MAL	$\underline{47.41}$	$\underline{50.62}$	$\underline{45.71}$	47.69	46.85	25.67	48.43	$\underline{41.03}$	49.69	52.03	51.28
ARB$^{F}_{\approx}$	AraBERTv0.2-base *TyDi QA*$_{\text{AR}}$→*TASK B*	FAL	52.97	55.86	$\underline{50.62}$	$\underline{54.43}$	51.28	**35.33**	53.78	42.39	52.20	55.77	53.45
ARB$^{M}_{\approx}$		MAL	$\underline{\mathbf{54.19}}$	$\underline{\mathbf{56.55}}$	50.51	53.32	51.35	31.33	53.22	$\underline{45.54}$	52.13	54.94	52.94

Table 4: Dev split evaluation results for task **B**. **pAP** means fixing ζ to 0.8. ***Post*** subscript identifies post-processing. ★ accompanied by ζ refers to applying the best ζ (see Appendix F). Average performance is reported for multiple runs of single models. Superscripts F and M in short form indicate FAL and MAL methods, respectively. Subscripts $\sim$ and $\approx$ denote direct fine-tuning and pipelined fine-tuning, respectively. Underlined values refer to the higher performance when comparing the two learning methods.

Short Form	Self Ensemble Model	MAP	MRR
TF-IDF Baseline		9.03	22.60
CAM$^{\otimes}_{\approx}$	CAMeLBERT-CA *TyDi QA*$_{\text{AR}}$→*Tafseer*→*TASK A*	23.02	47.06
ARB$^{\otimes}_{\approx}$	AraBERTv0.2-base *TyDi QA*$_{\text{AR}}$→*Tafseer*→*TASK A*	24.64	**49.39**
MIX$^{\otimes}_{\approx}$	CAM$^{\otimes}_{\approx}$ + ARB$^{\otimes}_{\approx}$	**25.05**	46.10

Table 5: Results on the hidden split for task **A**. ζ is set to mark 15% of questions as unanswerable.

Short Form	Method	Self Ensemble Model	pAP@10
Full-passage Baseline			32.68
ELC$^{M}_{\approx}$	MAL	AraELECTRA *TyDi QA*$_{\text{AR}}$→*TASK B*	53.10
ARB$^{M}_{\approx}$		AraBERTv0.2-base *TyDi QA*$_{\text{AR}}$→*TASK B*	**57.11**
MIX$^{M}_{\approx}$		ELC$^{M}_{\approx}$ + ARB$^{M}_{\approx}$	56.43

Table 6: Results on the hidden split for task **B**. ζ is set to mark 5% of pairs as unanswerable.

ζ parameter. (Please refer to Appendix F for more details about ζ selection and **RQ3**).

In Table 3, we experimented with dual-encoders using both random and hard negatives (Zhan et al., 2021) to address **RQ4**. (ARB$^{\circledcirc}_{\approx}$) outperforms (ARB$^{\circledcirc}_{\sim}$) by almost 4.5% when we perform hard negatives mining using a fine-tuned checkpoint (ARB$^{\circledcirc}_{\sim}$).

In Table 4, MAL learning method consistently brings significant improvements to the final performance for all models over the faithful split. Moreover, it consistently outperforms FAL learning method for the multi-answer type of samples. For instance, (ELC$^{M}_{\approx}$) performs better than (ELC$^{F}_{\approx}$), achieving a pAP score of 51.55% compared to 43.69% achieved by (ELC$^{F}_{\approx}$) for the subset of multi-answer samples. However, due to the fact that multi-answer samples make up only 18% of the development samples in the standard split (Table 2), MAL does not always outperform FAL for the standard split overall performance. This finding addresses **RQ5**.

With regard to **RQ6**, the post-processing approach proposed by Elkomy and Sarhan (2022) always surpasses the raw prediction score for both single and ensemble models. This is represented by ***Post*** subscript in Table 4. For example, post-processing improves (ARB$^{M}_{\approx}$) both single model and self-ensemble pAP performance by almost 3%.

7 Conclusion

In this paper, we have presented our solution for both task **A** and task **B** of Qur'an QA 2023 shared tasks. We explored various Arabic LMs using different training approaches and architectures. Our best performing systems are ensemble-based, enhanced with transfer learning using external learning resources. Lastly, we addressed a set of **RQ**s that highlight the main strengths of our work.

Limitations

In this paper, we have adapted conventional learning-based architectures for Arabic QA tasks, specifically for MRC and ad hoc search. However, we faced several challenges throughout our study. One significant challenge was the scarcity of training resources, along with the imbalanced distribution of topics and question types. This was particularly evident in the zero-answer cases. As a consequence, our zero-answer thresholding mechanism demonstrated high sensitivity to each individual model.

Additionally, we noticed significant performance variations due to the small size of the datasets. In order to tackle the problem of variations and noisy predictions, we investigated an ensemble approach. However, we still suggest that the results we obtained during the development phase may not accurately reflect the actual performance of learning systems. Despite the effectiveness of faithful splits for task **B**, we still suggest exploring n-fold cross-validation for both tasks. However, our computation resources were significantly limited during the competition phase.

For task **B**, our models trained for MRC were found to be suboptimal for ranking tasks. Although our post-processing technique improved the raw predictions, this indicates the necessity for other ranking-based MRC approaches. Furthermore, we would like to explore the performance of large LMs on this particular task.

Ethics Statement

The paper contains facts and beliefs that do not necessarily reflect the views or opinions of the authors. The information presented is based on objective analysis and does not aim to promote or endorse any particular religious interpretation.

Acknowledgements

We would like to extend our heartfelt appreciation to Dr. Moustafa El Zantout for his invaluable support and insights during the course of this work. We would also like to express our deep gratitude to the organizers of Qur'an QA for their efforts in promoting research in Arabic in general, and the most significant Arabic text, the Holy Qur'an.

References

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. pages 9–15. European Language Resource Association.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2021. Araelectra: Pre-training text discriminators for arabic language understanding. pages 191–195. Association for Computational Linguistics.

Muhammad Huzaifa Bashir, Aqil M Azmi, Haq Nawaz, Wajdi Zaghouani, Mona Diab, Ala Al-Fuqaha, and Junaid Qadir. 2021. Arabic natural language processing for qur'anic research: A systematic review.

Jonathan H Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. Tydi qa: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. pages 4171–4186. Association for Computational Linguistics.

Mohamemd Elkomy and Amany M Sarhan. 2022. Tce at qur'an qa 2022: Arabic language question answering over holy qur'an using a post-processed ensemble of bert-based models. pages 154–161. European Language Resources Association.

Siddhant Garg, Thuy Vu, and Alessandro Moschitti. 2020. Tanda: Transfer and adapt pre-trained transformer models for answer sentence selection. volume 34, pages 7780–7788.

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. Association for Computational Linguistics.

Amr Keleg and Walid Magdy. 2022. Smash at qur'an qa 2022: Creating better faithful data splits for low-resourced question answering scenarios. pages 136–145. European Language Resources Association.

Rana Malhas and Tamer Elsayed. 2020. AyaTEC: Building a Reusable Verse-based Test Collection for Arabic Question Answering on the Holy Qur'an. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 19(6):1–21.

Rana Malhas 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 *Proceedings of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT5) at the 13th Language Resources and Evaluation Conference (LREC 2022)*, pages 79–87.

Rana Malhas, Watheq Mansour, and Tamer Elsayed. 2023. Qur'an QA 2023 Shared Task: Overview of Passage Retrieval and Reading Comprehension Tasks over the Holy Qur'an. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore.

RANA R MALHAS. 2023. *ARABIC QUESTION ANSWERING ON THE HOLY QUR'AN*. Ph.D. thesis.

Ali Mostafa and Omar Mohamed. 2022. Gof at qur'an qa 2022: Towards an efficient question answering for the holy qu'ran in the arabic language using deep learning-based approach. pages 104–111. European Language Resources Association.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. pages 2383–2392. Association for Computational Linguistics.

Omer Sagi and Lior Rokach. 2018. Ensemble learning: A survey. *WIREs Data Mining and Knowledge Discovery*, 8(4):e1249.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*.

Ahmed Sleem, Eman Mohammed lotfy Elrefai, Marwa Mohammed Matar, and Haq Nawaz. 2022. Stars at qur'an qa 2022: Building automatic extractive question answering systems for the holy qur'an with transformer models and releasing a new dataset. pages 146–153. European Language Resources Association.

Marwan N. Swar. 2007. *Mushaf Al-Tafseel Al-Mawdoo'ee*. Dar Al-Fajr Al-Islami, Damascus.

Tanzil. 2007-2023. Tanzil - quran translations. `https://tanzil.net/trans/`. Electronic Quranic Resources.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Andrew Yates, Rodrigo Nogueira, and Jimmy Lin. 2021. Pretrained transformers for text ranking: BERT and beyond. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Tutorials*, pages 1–4, Online. Association for Computational Linguistics.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing dense retrieval model training with hard negatives. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 1503–1512, New York, NY, USA. Association for Computing Machinery.

Appendix

A Dataset Additional Details

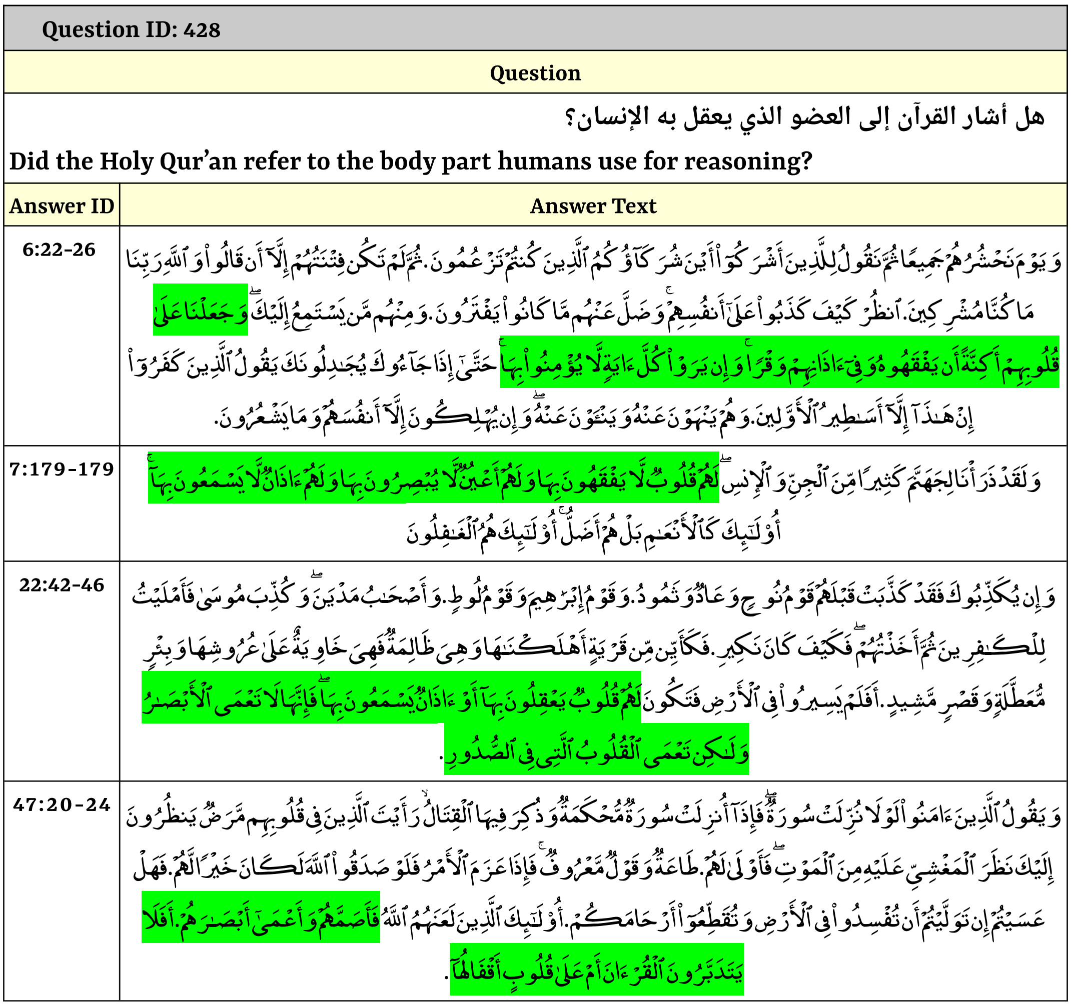
AyaTEC is a dataset designed to evaluate the performance of retrieval-based Arabic QA systems over the Holy Qur'an. It contains 207 questions and 1,762 corresponding answers, which are categorized into 11 topics covering different aspects of the Qur'an. The dataset caters to the information needs of two types of users: skeptical and curious (Malhas and Elsayed, 2020). The dataset includes single-answer and multi-answer questions, as well as questions that have no answer. Both Qur'an QA 2023 shared tasks are primarily based on an adapted version of AyaTEC (MALHAS, 2023; Malhas et al., 2022). Figure 1 illustrates an example from task **A**. The question asks whether there is a reference in the Qur'an to the body part used for reasoning. Four relevant Qur'anic segments are annotated to have an answer for this question. Figure 2 depicts a question-passage-answer triplet from task **B**. The question in this case is about creatures capable of praising God, within the context of the given passage.

A.1 Topic Distribution for tasks

AyaTEC covers 11 diverse topics referenced in the Holy Qur'an. Figure 3 illustrates the imbalanced nature of those different topics. Furthermore, the representation of unique questions is significantly limited in comparison to question-passage-answer triplets. Additionally, it is evident that the ratio of triplets to unique questions varies for each respective topic. In task **B**, these factors give rise to common questions across various passages. Consequently, they result in data leakage between the training and development splits (Keleg and Magdy, 2022). (Further information regarding this can be found in Appendix A.4)

A.2 Task A Evaluation Measures

For this ranking task, systems are expected to return up to 10 Qur'anic passages for each question when possible. If the system determines that the question is unanswerable from the entire Qur'an, a null document is only returned, indicated by -1. The primary measure for the task is MAP, which gives full credit only if all relevant documents are retrieved at the top of the ranked answer list. For



Question ID: 428

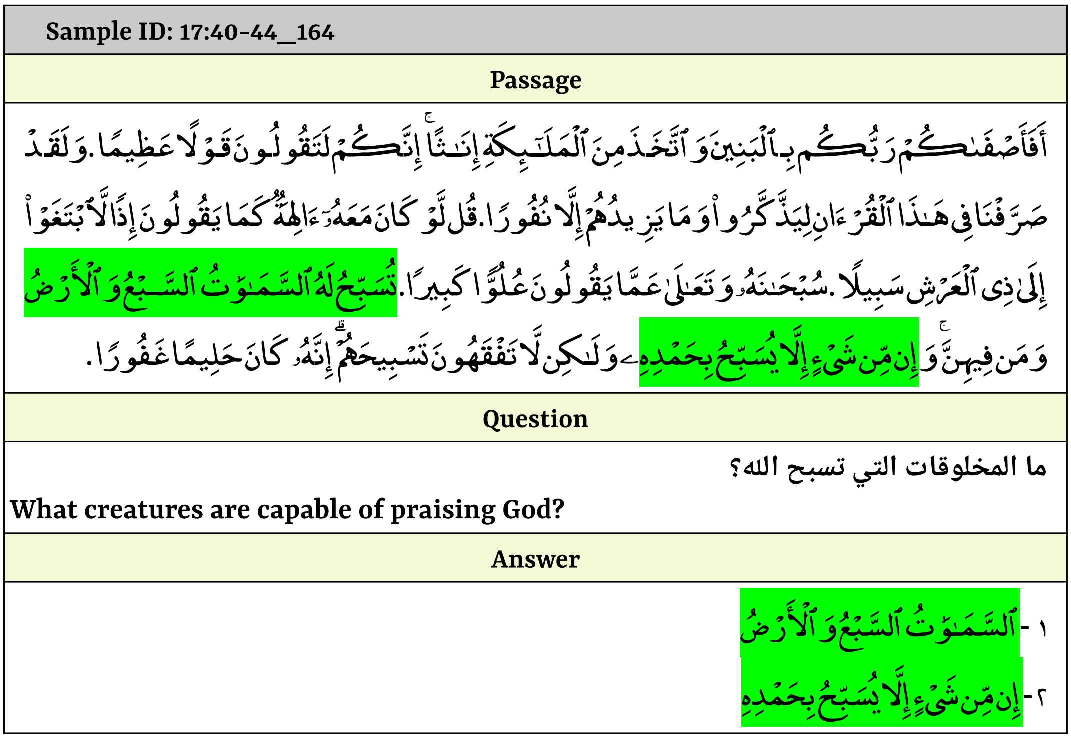
Question

هل أشار القرآن إلى العضو الذي يعقل به الإنسان؟

Did the Holy Qur'an refer to the body part humans use for reasoning?

Answer ID	Answer Text
6:22-26	وَيَوْمَ نَحْشُرُهُمْ جَمِيعًا ثُمَّ نَقُولُ لِلَّذِينَ أَشْرَكُوٓا۟ أَيْنَ شُرَكَآؤُكُمُ ٱلَّذِينَ كُنتُمْ تَزْعُمُونَ. ثُمَّ لَمْ تَكُن فِتْنَتُهُمْ إِلَّآ أَن قَالُوا۟ وَٱللَّهِ رَبِّنَا مَا كُنَّا مُشْرِكِينَ. ٱنظُرْ كَيْفَ كَذَبُوا۟ عَلَىٰٓ أَنفُسِهِمْ ۚ وَضَلَّ عَنْهُم مَّا كَانُوا۟ يَفْتَرُونَ. وَمِنْهُم مَّن يَسْتَمِعُ إِلَيْكَ ۖ وَجَعَلْنَا عَلَىٰ قُلُوبِهِمْ أَكِنَّةً أَن يَفْقَهُوهُ وَفِىٓ ءَاذَانِهِمْ وَقْرًا ۚ وَإِن يَرَوْا۟ كُلَّ ءَايَةٍ لَّا يُؤْمِنُوا۟ بِهَا ۚ حَتَّىٰٓ إِذَا جَآءُوكَ يُجَٰدِلُونَكَ يَقُولُ ٱلَّذِينَ كَفَرُوٓا۟ إِنْ هَٰذَآ إِلَّآ أَسَٰطِيرُ ٱلْأَوَّلِينَ. وَهُمْ يَنْهَوْنَ عَنْهُ وَيَنْـَٔوْنَ عَنْهُ ۖ وَإِن يُهْلِكُونَ إِلَّآ أَنفُسَهُمْ وَمَا يَشْعُرُونَ.
7:179-179	وَلَقَدْ ذَرَأْنَا لِجَهَنَّمَ كَثِيرًا مِّنَ ٱلْجِنِّ وَٱلْإِنسِ ۖ لَهُمْ قُلُوبٌ لَّا يَفْقَهُونَ بِهَا وَلَهُمْ أَعْيُنٌ لَّا يُبْصِرُونَ بِهَا وَلَهُمْ ءَاذَانٌ لَّا يَسْمَعُونَ بِهَآ ۚ أُو۟لَٰٓئِكَ كَٱلْأَنْعَٰمِ بَلْ هُمْ أَضَلُّ ۚ أُو۟لَٰٓئِكَ هُمُ ٱلْغَٰفِلُونَ
22:42-46	وَإِن يُكَذِّبُوكَ فَقَدْ كَذَّبَتْ قَبْلَهُمْ قَوْمُ نُوحٍ وَعَادٌ وَثَمُودُ. وَقَوْمُ إِبْرَٰهِيمَ وَقَوْمُ لُوطٍ. وَأَصْحَٰبُ مَدْيَنَ ۖ وَكُذِّبَ مُوسَىٰ فَأَمْلَيْتُ لِلْكَٰفِرِينَ ثُمَّ أَخَذْتُهُمْ ۖ فَكَيْفَ كَانَ نَكِيرِ. فَكَأَيِّن مِّن قَرْيَةٍ أَهْلَكْنَٰهَا وَهِىَ ظَالِمَةٌ فَهِىَ خَاوِيَةٌ عَلَىٰ عُرُوشِهَا وَبِئْرٍ مُّعَطَّلَةٍ وَقَصْرٍ مَّشِيدٍ. أَفَلَمْ يَسِيرُوا۟ فِى ٱلْأَرْضِ فَتَكُونَ لَهُمْ قُلُوبٌ يَعْقِلُونَ بِهَآ أَوْ ءَاذَانٌ يَسْمَعُونَ بِهَا ۖ فَإِنَّهَا لَا تَعْمَى ٱلْأَبْصَٰرُ وَلَٰكِن تَعْمَى ٱلْقُلُوبُ ٱلَّتِى فِى ٱلصُّدُورِ.
47:20-24	وَيَقُولُ ٱلَّذِينَ ءَامَنُوا۟ لَوْلَا نُزِّلَتْ سُورَةٌ ۖ فَإِذَآ أُنزِلَتْ سُورَةٌ مُّحْكَمَةٌ وَذُكِرَ فِيهَا ٱلْقِتَالُ ۙ رَأَيْتَ ٱلَّذِينَ فِى قُلُوبِهِم مَّرَضٌ يَنظُرُونَ إِلَيْكَ نَظَرَ ٱلْمَغْشِىِّ عَلَيْهِ مِنَ ٱلْمَوْتِ ۖ فَأَوْلَىٰ لَهُمْ. طَاعَةٌ وَقَوْلٌ مَّعْرُوفٌ ۚ فَإِذَا عَزَمَ ٱلْأَمْرُ فَلَوْ صَدَقُوا۟ ٱللَّهَ لَكَانَ خَيْرًا لَّهُمْ. فَهَلْ عَسَيْتُمْ إِن تَوَلَّيْتُمْ أَن تُفْسِدُوا۟ فِى ٱلْأَرْضِ وَتُقَطِّعُوٓا۟ أَرْحَامَكُمْ. أُو۟لَٰٓئِكَ ٱلَّذِينَ لَعَنَهُمُ ٱللَّهُ فَأَصَمَّهُمْ وَأَعْمَىٰٓ أَبْصَٰرَهُمْ. أَفَلَا يَتَدَبَّرُونَ ٱلْقُرْءَانَ أَمْ عَلَىٰ قُلُوبٍ أَقْفَالُهَآ.

Figure 1: A sample from shared task **A**. We highlight the most relevant part in each Qur'anic segment.



Sample ID: 17:40-44_164

Passage

أَفَأَصْفَىٰكُمْ رَبُّكُم بِٱلْبَنِينَ وَٱتَّخَذَ مِنَ ٱلْمَلَٰٓئِكَةِ إِنَٰثًا ۚ إِنَّكُمْ لَتَقُولُونَ قَوْلًا عَظِيمًا. وَلَقَدْ صَرَّفْنَا فِى هَٰذَا ٱلْقُرْءَانِ لِيَذَّكَّرُوا۟ وَمَا يَزِيدُهُمْ إِلَّا نُفُورًا. قُل لَّوْ كَانَ مَعَهُۥٓ ءَالِهَةٌ كَمَا يَقُولُونَ إِذًا لَّٱبْتَغَوْا۟ إِلَىٰ ذِى ٱلْعَرْشِ سَبِيلًا. سُبْحَٰنَهُۥ وَتَعَٰلَىٰ عَمَّا يَقُولُونَ عُلُوًّا كَبِيرًا. تُسَبِّحُ لَهُ ٱلسَّمَٰوَٰتُ ٱلسَّبْعُ وَٱلْأَرْضُ وَمَن فِيهِنَّ ۚ وَإِن مِّن شَىْءٍ إِلَّا يُسَبِّحُ بِحَمْدِهِۦ وَلَٰكِن لَّا تَفْقَهُونَ تَسْبِيحَهُمْ ۗ إِنَّهُۥ كَانَ حَلِيمًا غَفُورًا.

Question

ما المخلوقات التي تسبح الله؟

What creatures are capable of praising God?

Answer

١- ٱلسَّمَٰوَٰتُ ٱلسَّبْعُ وَٱلْأَرْضُ

٢- إِن مِّن شَىْءٍ إِلَّا يُسَبِّحُ بِحَمْدِهِۦ

Figure 2: A sample from shared task **B**. We highlight the ground truth answers in the Qur'anic passage.

the zero-answer questions, full credit is given to successful systems only when they are unable to find any relevant Qur'anic passage to answer the question, and return the null document. In addition to MAP, mean Reciprocal Rank (MRR) is also reported, which gives credit just for the first relevant document from the ranked list (Yates et al., 2021).

In formal notation, we begin by defining the function $\alpha(q,p)$, which is a binary relevance function that indicates whether a passage p is annotated as relevant to a question q in the test collection. Equ.(1) represents the function that calculates the total number of relevant Qur'anic passages from the QPC to q.

$$\psi(q) = \sum_{p \in \mathcal{QPC}} \alpha(q,p) \tag{1}$$

Zero-answer questions have a zero value for the function ψ, and their MAP score is calculated in a different way. Equ.(2) shows the evaluation measure for MAP for answerable questions. For a ranked list R, we calculate the precision at each possible cutoff $@i$ at which a relevant document is present (Yates et al., 2021).

$$\mathrm{MAP}(R,q) = \frac{\sum_{(i,p)\in R} \mathrm{Prec}\,@i(R,q)\cdot \alpha(q,p)}{\psi(q)}, \tag{2}$$

Equ.(3) illustrates the combined MAP evaluation measure for task **A**. In this measure, zero-answer questions are given full credit only when R is the null document, represented by -1 in the official evaluation script [3] (MALHAS, 2023).

$$\mathrm{MAP_A}(R,q) = \begin{cases} \mathbb{1}_{R\equiv[-1]} & \text{if } \psi(q)=0 \\ \mathrm{MAP}(R,q) & \text{Otherwise} \end{cases} \tag{3}$$

$\mathbb{1}_C$ is an *indicator* function, which returns 1 if the binary condition C holds and 0 otherwise.

A.3 Task B Evaluation Measures

Standard MRC tasks, like SQuADv2.0, are evaluated based only on the first prediction. In contrast, task **B** is evaluated as a ranking task against a ranked list, rather than relying solely on the top prediction. As in task **A**, systems are expected to return up to 10 answer spans from a given Qur'anic passage to answer a question when possible. The primary evaluation metric for this task is pAP (Malhas and Elsayed, 2020; MALHAS, 2023). This metric incorporates partial matching with the traditional rank-based Average Precision measure, i.e., MAP. In the case of unanswerable samples, the system receives a full score if it only returns and empty ranked list.

Formally, partial matching is performed over token indexes of two substrings extracted from a given supporting passage. Based on Malhas and Elsayed (2020), F_1 is used to calculate the similarity between the two substrings R_k and g. R_k represents the k^{th} answer from a ranked list R, and g refers to any ground truth answer from the set of ground truth answers G.

$$\mathcal{F}_{1k}^{R} = \max_{g\in G}\{F_1(R_k,g)\} \tag{4}$$

In terms of Equ.(4), we can define a partial matching version of precision at cutoff K, i.e., pPrec (Malhas and Elsayed, 2020; MALHAS, 2023).

$$\mathrm{pPrec}\,@K(R) = \frac{1}{K}\sum_{i=1}^{K}\mathcal{F}_{1i}^{R} \tag{5}$$

In their study, MALHAS (2023) introduced a method for handling multi-answer samples. They proposed a string splitting mechanism that ensures only one correct answer is matched in each entry of R. Equ.(6) presents the pAP evaluation metric for multi-answer ranking MRC in terms of pPrec (Malhas and Elsayed, 2022), which stands as a token-level partial matching version of Equ(2).

$$\mathrm{pAP}(R) = \frac{\sum_{i\in R}\mathrm{pPrec}\,@i(R)\cdot\beta(R,i)}{|G|}, \tag{6}$$

$\beta(R,i)$ is a binary function that returns one if R_i is a partially relevant answer. More specifically,

$$\beta(R,k) = \mathbb{1}_{\mathcal{F}_{1k}^{R}>0} \tag{7}$$

In similar fashion, Equ.(8) presents the complete pAP evaluation measure for task **B**. In this measure, zero-answer samples are given full credit only when R is an empty list (MALHAS, 2023).

$$\mathrm{pAP_B}(R) = \begin{cases} \mathbb{1}_{R\equiv[\,]} & \text{if } |G|=0 \\ \mathrm{pAP}(R) & \text{Otherwise} \end{cases} \tag{8}$$

[3] The symbol $\equiv$ signifies the equivalence operator between two lists.

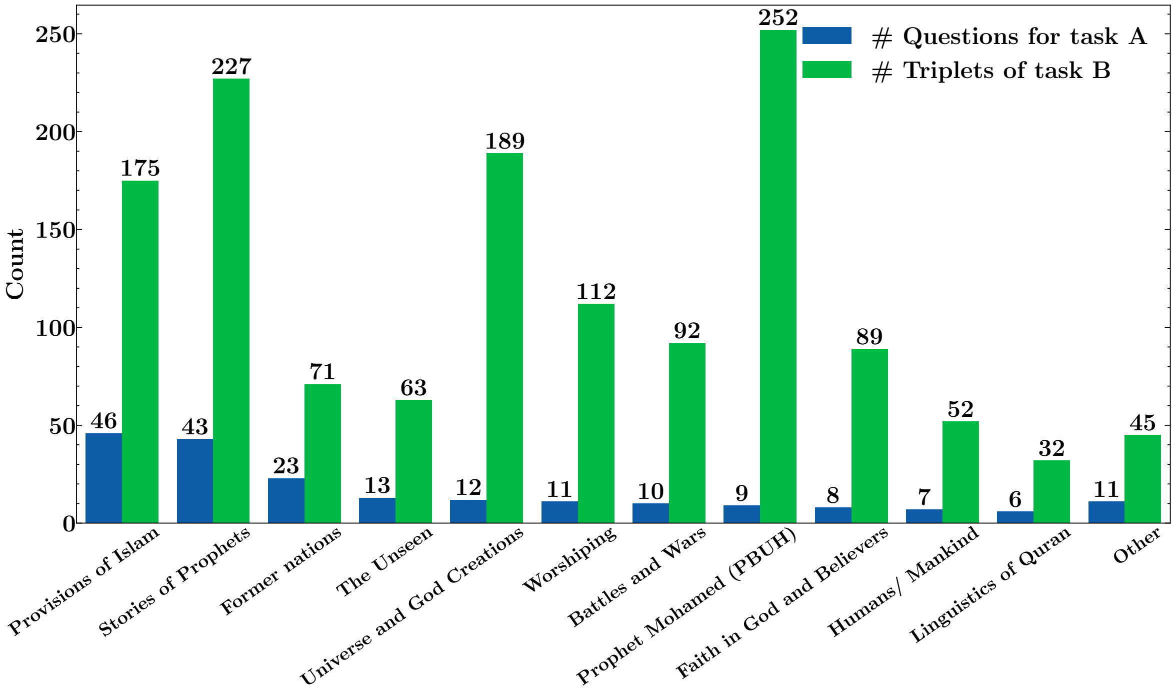


Figure 3: Distribution of QRCDv1.2 over the 11 topics for task **A** questions and task **B** triplets.

A.4 Leakage in QRCDv1.2

Keleg and Magdy (2022) analyzed QRCDv1.1 and identified instances where passages and questions were repeated. They classified QRCDv1.1 into four logical mutually-exclusive categories according to their complexity. Table 7 provides a summary of the criteria used and the expected behavior of trained LMs for each category. Additionally, symbols are employed to indicate the levels of complexity within each category, as determined by performance scores obtained by Keleg and Magdy (2022). Based on their analysis, Keleg and Magdy (2022) solely utilized $D_{(3)\ \text{ood + hard}}$ for their final development split for QRCDv1.1.

In this work, we extend their approach for QRCDv1.2. We slightly modify this by considering both $D_{(2)}$ and $D_{(3)}$ for the development split. In addition, we employ disjoint set algorithm to find all leakage groups in $D_{(1)}$. We use those groups to balance the zero-answer questions ratio in the development split. This is because 90% of zero-answer questions belong to the trivial leakage group $D_{(1)}$.

In their work, Keleg and Magdy (2022) also proposed a resplitting approach for QRCDv1.1. They reorganized training and development splits using the four logical groups to create what they called *faithful* splits for QRCDv1.1. Faithful splits aim to create more representative evaluations for QRCDv1.1 dataset. Table 8 summarizes the modifications we made for performing evaluation using faithful splits. Table 9 presents the distribution of our faithful split for QRCDv1.2 based on our modified splitting strategy outlined in Table 8. It also includes the distribution of zero-answer samples within each group. As in Table 9, we preserve the original ratio of training to development splits. Additionally, the percentage of zero-answer samples within each split is preserved compared to the original distribution in Table 2.

A.5 External Learning Resources

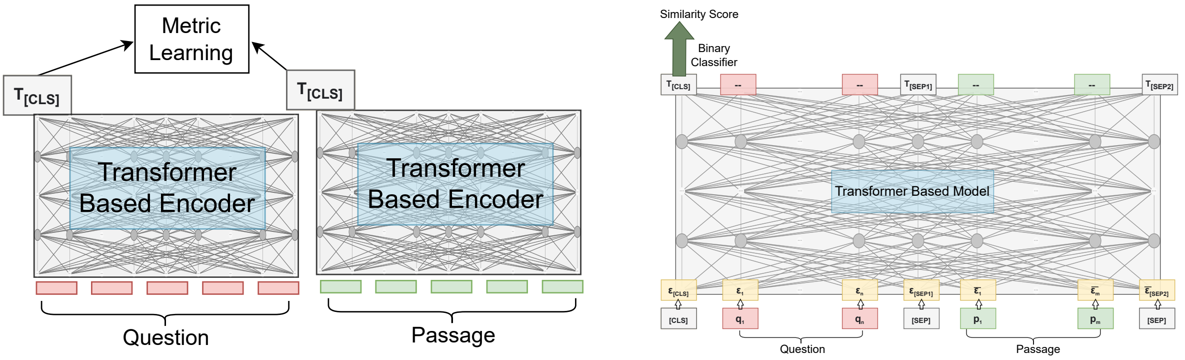
We leverage external resources to perform pipelined fine-tuning for both tasks **A** and **B**. For task **A**, we utilized interpretation resources (tafseer) from both Muyassar and Jalalayn, obtained from Tanzil (2007-2023). We created pairs of QPC Qur'anic passages and their corresponding interpretations, resulting in approximately 2.5K relevant pairs. Additionally, we used the Arabic TyDI-QA Gold**P** dataset (Clark et al., 2020) to generate pairs of relevant questions and their supporting evidence passages, resulting in 15K relevant pairs. For task **B**, we solely relied on the Arabic subset of the TyDI-QA Gold**P** MRC dataset (Clark et al., 2020). This dataset consists of approximately 15K question-passage-answer triplets.

Category	**Criteria**	**Expected LM behavior**
$D_{(1)}$ in+leakage	Samples with repeated passage-answer or question-answer pairs	Memorize answers and overfit to training data ∼
$D_{(2)}$ in+no leakage	Samples with repeated passages but having unique answers which are different from $D_{(1)}$ answers	Reasoning is required to find the right answer ≋
$D_{(3)}$ ood + hard	Samples with unique passages but having rarely repeated questions (appearing 3 times or less)	Some reasoning is required to find the right answer for rare questions ≋
$D_{(4)}$ ood + easy	Samples with unique passages but having commonly repeated questions (more than 3 times)	Lexical matching guides trained LMs to find similar answers ≈

Table 7: Description of the four categories introduced by Keleg and Magdy (2022) over QRCDv1.1 dataset. We show the criteria for identifying each category and the expected behavior for a fine-tuned LM. We denote the complexity of each category using symbols. For instance, ≋ represents the most challenging set for learning systems, while ∼ refers the least challenging set.

Category	**Splitting Strategy by Keleg and Magdy (2022)**	**Our Modified Splitting Strategy**
$D_{(1)}$ in+leakage	For duplicate question-answer or passage-answer pairs, choose only one sample for training and leave the rest for the development set.	Use it entirely for training, this is due to the fact that $D_{(1)}$ is trivial for development. To balance the zero-answer questions ratio, we take entire zero-answer leakage groups into the development set. We employ disjoint-set algorithm for this purpose.
$D_{(2)}$ in+no leakage	Split randomly with a splitting ratio of 86.7% for training and 13.3% for development, which corresponds to the original ratio of the data.	Split them into two overlapping sets, as such, confusing examples with the same passages are distributed among training and development with different answers.
$D_{(3)}$ ood + hard	Only use it for the development set (removed from training).	Same as Keleg and Magdy (2022)
$D_{(4)}$ ood + easy	Split randomly with a splitting ratio of 86.7% for training and 13.3% for development, which corresponds to the original ratio of the data.	Use it entirely for training, this is due to the fact that $D_{(4)}$ is trivial for development.

Table 8: Description of our modified *faithful* splitting for QRCDv1.2 dataset over the four categories introduced by Keleg and Magdy (2022). We also show their proposed splitting approach (Keleg and Magdy, 2022). Check Table 7 for more details and reasons behind such splitting strategies.



(a) Dual-encoder generic architecture with metric learning for neural ranking.

(b) Cross-encoder generic architecture for an input pair of a question and a passage with a predicted similarity score.

Figure 4: Diagrams for model architectures for task **A**.

Category	Train	Development	Total
$D_{(1)\text{ in+leakage}}$	405 (49)	7 (7)	412 (56)
$D_{(2)\text{ in+no leakage}}$	290 (2)	95 (1)	385 (3)
$D_{(3)\text{ ood + hard}}$	0 (0)	62 (3)	62 (3)
$D_{(4)\text{ ood + easy}}$	296 (0)	0 (0)	296 (0)
Total	991 (51)	164 (11)	1155 (62)
Zero-answer %	5.15 %	6.71 %	5.37 %

Table 9: QRCDv1.2 dataset distribution of pairs for our *faithful* splitting over the four categories introduced by (Keleg and Magdy, 2022). Parenthesized values refer to the number of zero-answer samples within each category for each split.

B Transfer Learning

In order to overcome the limited training resources for both tasks, we incorporate external QA and interpretation resources (tafseer) (Tanzil, 2007-2023). External resources enhance our learning systems in general by leveraging transfer learning across multiple fine-tuning stages (Garg et al., 2020; MALHAS, 2023). We use arrows in subscripts in Tables 3, 4, 5, and 6 to refer to stages of fine-tuning. (More details about external learning resources and their construction in Appendix A.5)

C Ensemble Learning

We utilize a voting self-ensemble technique for a group of fine-tuned models trained with different seeds (Sagi and Rokach, 2018). We use the raw predictions without applying a zero-answer threshold.

In task **A**, for an ensemble $\mathcal{E}$ we aggregate the relevance scores for a Qur'anic passage p and a question q assigned by a model φ. The ensemble relevance score $\mathcal{S}$ between p and q is as follows:

$$\mathcal{S}(q, p) = \sum_{\varphi \in \mathcal{E}} \varphi(q, p) \tag{9}$$

In similar fashion for task **B**, we leverage a span voting ensemble (Elkomy and Sarhan, 2022). For each sample, we aggregate span scores for each span s made by each predictor φ.

$$\mathcal{S}(s) = \sum_{\varphi \in \mathcal{E}} \varphi(s) \tag{10}$$

After that, we apply zero-answer thresholding to the aggregated result.

D Additional System Details for task A

We summarize both architectures for task **A** in Figures 4a and 4b for dual-encoders and cross-encoders, respectively.

D.1 Implementation Details

In our STAR training process, we incorporate both random in-batch negatives and hard negatives. Random negatives involve randomly selecting irrelevant documents for each query, providing positive and negative signals for learning systems (Yates et al., 2021). On the other hand, hard negatives refer to the most offending irrelevant examples predicted by an encoder similarity score (Zhan et al., 2021). In a batch of size 16, we encode 16 different queries with their corresponding positive documents; in addition, in-batch negatives are used for all other queries. These negatives can be chosen randomly or through STAR hard negative mining. We use a learning rate of 5×10^{-5} for all of our dual-encoder experiments. In the case of cross-encoders, we generate question-document pairs. These pairs have a ratio of one positive pair and three randomly selected negative pairs. For all of our cross-encoders, we use a learning rate of 1×10^{-6} with a batch size of 16.

D.2 Zero-answer Prediction

We assign a likelihood for each question q to be answerable using the total relevance scores for the top returned passages R. φ refers to a general relevance predictor between q and a passage p.

$$\gamma(q) = -\sum_{p \in R} \varphi(q, p) \tag{11}$$

The negative sign corresponds to the inverse proportional relationship between high relevance scores and the likelihood of unanswerability. We then normalize those scores for all questions into $\bar{\gamma}(q)$ and apply a no answer threshold ζ. We define a binary threshold function, σ, which applies the threshold to identify unanswerable questions.

$$\sigma(q) = \mathbb{1}_{\bar{\gamma}(q) > \zeta} \tag{12}$$

E Additional System Details for task B

In this work, we fine-tune LMs for extractive MRC as span predictors (Devlin et al., 2019). The fine-tuning process involves packing each question-passage pair x together and feeding it to a LM to predict the start and end token indices from the passage, as shown in Figure 5. To achieve this, a trainable randomly initialized start vector S and end vector E are stacked on top of the LM, having the i^{th} token hidden-representation T_i. The final model with the newly stacked layers has learnable parameters θ.

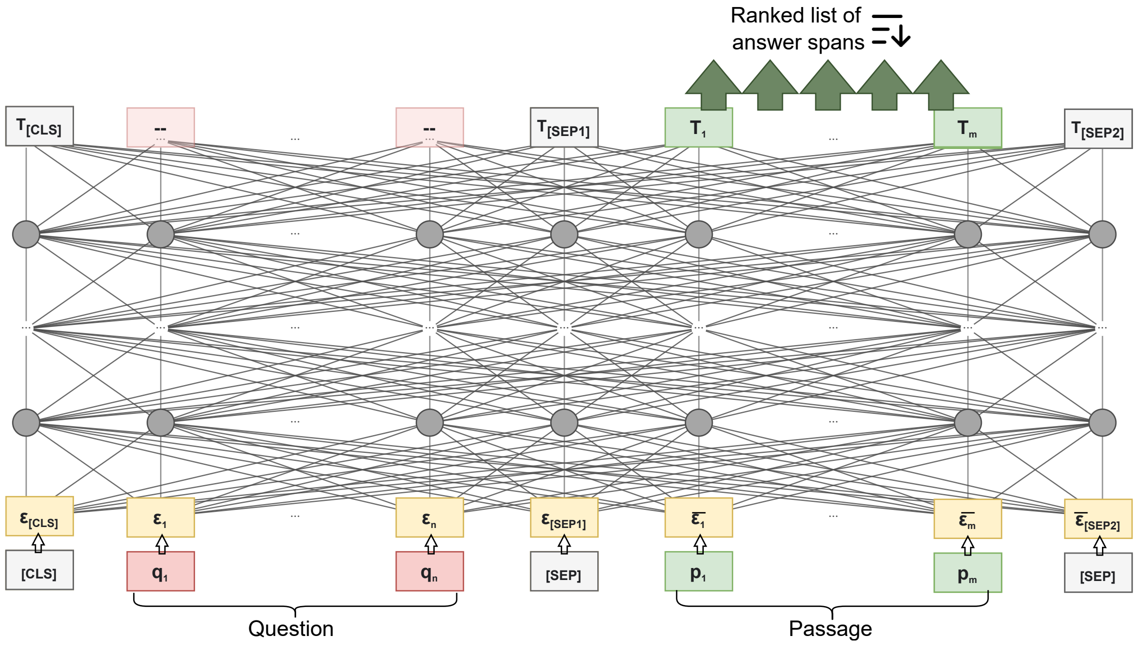
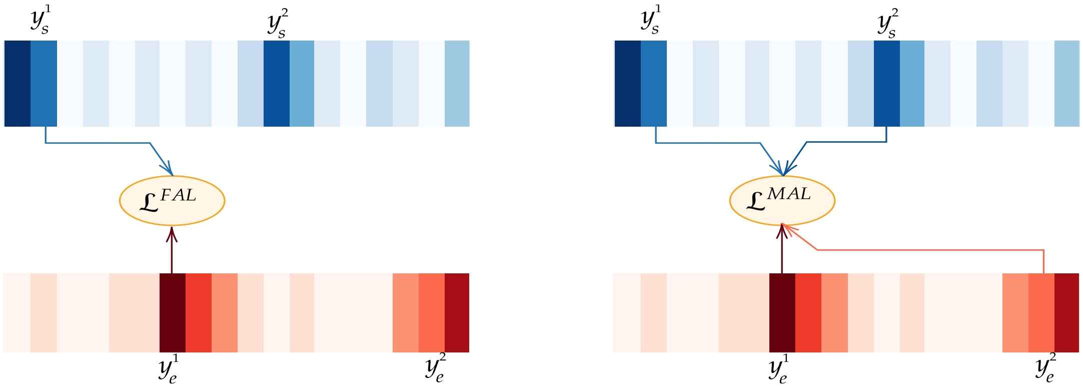


Figure 5: Generic architecture illustration of a LM for ranking MRC.



(a) Standard (FAL)

(b) MAL

Figure 6: Illustration of Learning Methods.

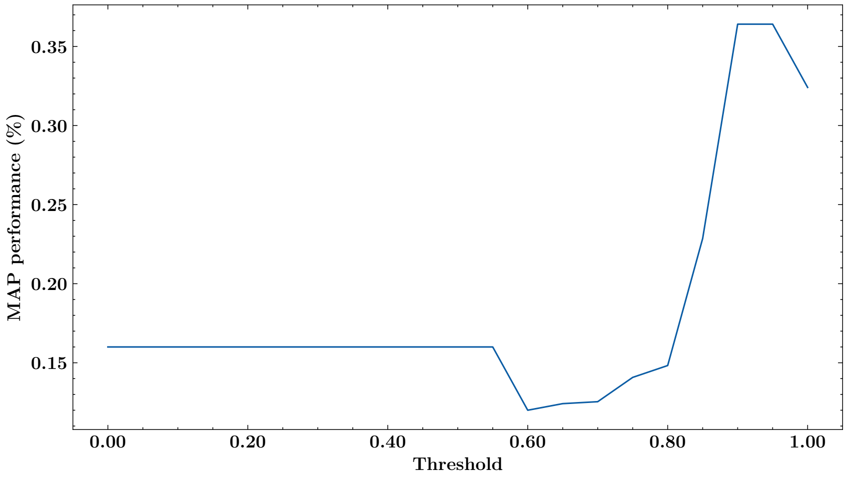


Figure 7: Thresholding effect against MAP performance for one of our fine-tuned models.

The dot product between S and T_i is chosen to determine the score that the i^{th} token is the start of the answer span. These scores for all passage tokens are followed by a softmax layer that produces the probabilities for individual tokens being the start of the answer span (Seo et al., 2016; Devlin et al., 2019). Equ.(13) depicts the probability that the i^{th} token is the start of the answer span.

$$\mathbb{P}(i \mid x; \theta) = \frac{e^{S \cdot T_i}}{\sum_j^{|T|} e^{S \cdot T_j}} \tag{13}$$

Under full-supervision, the training objective is to optimize the log-likelihoods for both the ground truth start and end positions. For a model with learnable θ, an input x, and a single ground truth answer span y, the log likelihood for the start token position is as follows:

$$\mathcal{L}_{\text{start}}(\theta \mid x, y) = -\log \mathbb{P}\left(y_s \mid x; \theta\right) \tag{14}$$

where the subscript s in y_s refers to the start position of the answer span y.

If there are multiple answers for a sample x, we rather have a set of plausible answer spans $\mathcal{Y}$. Elkomy and Sarhan (2022); Sleem et al. (2022); Mostafa and Mohamed (2022) in Qur'an QA 2022 tackled this by considering any answer span from $\mathcal{Y}$ by taking one at random or the first answer span, namely, y^1. We denote the i^{th} answer from $\mathcal{Y}$ as y^i. We call this learning method First answer loss (FAL). This can be formulated in terms of $\mathcal{Y}$ as denoted below:

$$\mathcal{L}_{\text{start}}^{\text{FAL}}(\theta \mid x, \mathcal{Y}) = -\log \mathbb{P}\left(y_s^1 \mid x; \theta\right) \tag{15}$$

Figure 6a illustrates this learning method. However, QRCDv1.2 task **B** considers a multi-answer MRC scenario, this leads to discrepancy between training and testing when FAL learning method is employed for fine-tuning. Towards this end, we define MAL learning method. This learning method takes the multi-answer cases in consideration by optimizing for all answers altogether. Mathematically, this generalizes to any y^i from the set $\mathcal{Y}$ and takes the sum of the log likelihood losses for multiple answers as shown in Equ.(16):

$$\mathcal{L}_{\text{start}}^{\text{MAL}}(\theta \mid x, \mathcal{Y}) = -\sum_{y^i \in \mathcal{Y}} \log \mathbb{P}\left(y_s^i \mid x; \theta\right) \tag{16}$$

We show the MAL learning method in Figure 6b.

E.1 Implementation Details

To enhance LMs predictions, we employ a post-processing approach. Elkomy and Sarhan (2022) proposed an effective non-maximum suppression post-processing approach at Qur'an QA 2022 (Malhas et al., 2022). They also proposed some operations for rejecting uninformative short answers. For all of our models, we used a learning rate of 2×10^{-5} and a batch size of 16.

E.2 Zero-answer Prediction

MRC for SQuADv2.0-like datasets uses null answer **[CLS]** token probability to give a likelihood for a question to have an answer within the supporting passage (Rajpurkar et al., 2018; Devlin et al., 2019). This works by finding the difference between the null answer score of **[CLS]** token and the non-empty answer span with the highest score. φ is a general span extractor that operates on a question q and a passage p.

$$\gamma(q,p) = \varphi(q,p)_{\textit{[CLS]}} - \varphi(q,p)_{\textit{MAX}} \tag{17}$$

Upon calculating scores for all samples, we proceed to normalize them into $\bar{\gamma}(q)$ and then apply a threshold value ζ to determine if there is no answer. To identify unanswerable questions, we use a binary threshold function σ,

$$\sigma(q) = \mathbb{1}_{\bar{\gamma}(q) > \zeta} \tag{18}$$

F ζ Selection and $\zeta^\star$

In this work, we defined ζ hyperparameter for zero-answer thresholding. This hyperparameter controls the proportion of samples that are considered to be zero-answer. Due to the small size of the dataset, we used a quantile method to set ζ. This method marks a proportion of the samples according to the statistics of the dataset. Task **B** is less sensitive to this parameter because almost 5% of the samples are zero-answer. In contrast, task **A** is highly sensitive to this parameter because of the larger proportion of zero-answer cases compared to task **A**. Additionally, We are interested in finding the theoretical upperbound performance for ζ; this is addressed by **RQ3**.

In Tables 3 and 4 we use ★ accompanied by ζ to refer to the optimal performance of the binary classification problem of has-answer vs. has-no-answer, as explained in Appendices D.2 and E.2. Figure 7 illustrates the thresholding effect against fine-tuned model performance for task **A**; this answers **RQ3**. As we can see, the ζ hyperparameter can not be set arbitrarily. Instead, we can adjust it by considering the outcomes obtained from trained models on the training data. To find the optimal threshold $\zeta^{\star}$ for both tasks, we implemented a greedy optimization algorithm for all possible levels of thresholds made by a given model; check the code for more details [4].

[4] In both code bases, this is performed by function *find_best_thresh*. You may find this function under *metrics* directory in *compute_score_qrcd.py* and *helpers.py* scripts for tasks **A** and **B**, respectively.

Al-Jawaab at Qur’an QA 2023 Shared Task: Exploring Embeddings and GPT Models for Passage Retrieval and Reading Comprehension

Abdulrezzak Zekiye
Istanbul, Türkiye
abdalrazak.zk@gmail.com

Fadi Amroush
Niuversity / Berlin, Germany
fadi.amr@niuversity.com

Abstract

This paper introduces a comprehensive system designed to address two natural language processing tasks: Passage Retrieval (Task A) and Reading Comprehension (Task B), applied to datasets related to the Holy Qur’an. Task A was treated as a measurement of a textual similarity problem where the system leverages OpenAI’s "text-embedding-ada-002" embedding model to transform textual content into numerical representations, with cosine similarity serving as the proximity metric. Task B focuses on the extraction of answers from Qur’anic passages, employing the Generative Pre-trained Transformer-4 (GPT-4) language model. In Task A, the system is evaluated using the Mean Average Precision (MAP) metric, achieving MAP scores of 0.109438 and 0.06426543057 on the development and test datasets with an optimal similarity threshold set at 0.85. Task B evaluation employs partial Average Precision (pAP), where our system surpasses a baseline whole-passage retriever with pAP scores of 0.470 and 0.5393130538 on the development and test datasets, respectively.

Holy Qur’an, passage retrieval, reading comprehensive, GPT-4, embeddings

1 Introduction

Establishing a dependable method for providing accurate responses and citing relevant passages from the Holy Qur’an within the framework of natural language processing represents a crucial and challenging endeavor. The creation of a reliable model capable of delivering precise answers to inquiries about Islam and the Holy Qur’an holds substantial potential. It not only serves as a valuable resource for facilitating accurate information retrieval but also as a potent tool for automatically detecting and countering the dissemination of false information on the internet and social media platforms. Qur’an QA 2023 Shared Task (Malhas et al., 2023) encourages researchers to work on two important tasks, Task A: Passage Retrieval and Task B: Reading Comprehension.

Task A: Passage Retrieval. This task involves providing a ranked list of passages from the Holy Qur’an that potentially contain answers to a given free-text question in Modern Standard Arabic (MSA). The task encompasses both factoid and non-factoid questions where factoid questions have short answers such as names and numerical values, and non-factoid questions need explanations, reasoning, or opinions to provide an answer (Surdeanu et al., 2011). This task also includes certain questions within the dataset that lack corresponding answers in the Holy Qur’an. The system should return a ranked list containing up to 10 Qur’anic passages believed to contain the answer(s) to the given question if any, and "no answers." in case there is no answer from the Holy Qur’an.

Task B: Reading Comprehension. The task involves working with a particular Qur’anic passage, which comprises consecutive verses from a specific Surah in the Holy Qur’an, along with a free-text question presented in MSA pertaining to that passage. The primary objective is for a system to identify and extract all answers to a question that are explicitly mentioned within the corresponding passage. This approach differs from the previous Qur’an QA 2022 task, where the system was required to return any answer (Malhas et al., 2022). These answers are expected to be contiguous spans of text within the passage. Similar to Task A, the questions themselves can encompass both factoid and non-factoid types and the system should return up to 10 answers out of the provided passage, or an empty set representing a "no answer" case.

The structure of this paper is as follows: In Section 2, we provide an overview of the datasets employed for Tasks A and B. Section 3 details the methodologies utilized to address both tasks. Our results are presented in Section 4. Lastly, Section 5 offers a discussion of the results.

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 743–747
December 7, 2023 ©2023 Association for Computational Linguistics

2 Data

2.1 Task A: Passage Retrieval

Task A dataset (Malhas et al., 2023; Malhas, 2023; Malhas and Elsayed, 2020; Swar, 2007) comprises three main components: the Qur'anic passage collection (QPC), the questions from the AyaTEC dataset, and query relevance judgements (QRels) as the assessments of how relevant these questions are to the passages within the QPC. The QPC was created by categorizing the 114 Qur'anic chapters, each of varying lengths, based on thematic divisions as outlined in the Thematic Holy Qur'an (Swar, 2007). This process led to a total of 1,266 distinct passages. The AyaTEC dataset has 199 questions and the QRels dataset consists of 1,132 gold Qur'anic passage-ids that are deemed relevant to each question. The output format of the system that solves task A should be as follows, where *tag* is used to indicate a human-readable model name: '<question-id>' Q0 '<passage-id>' '<rank>' '<relevance-score>' '<tag>'. The dataset was split as 70% for training, 10% for development, and 20% for testing, yielding 174 questions for the training, 25 for the development, and 52 for testing. From question-passage pairs point-of-view, the dataset had 972 pairs for training, 160 for development, and 427 for testing.

2.2 Task B: Reading Comprehension

In Task B, the used dataset is taken out of the Qur'anic Reading Comprehension Dataset (QRCD) v1.2 (Malhas et al., 2023, 2022; Malhas and Elsayed, 2022, 2020). QRCD v1.2 consists of 1,399 triplets of questions and corresponding passages, along with their extracted answers. The questions with "no answer" constitute 15% of the questions in the QRCD v1.2 dataset. The dataset was split as 64% for training, 10% for development, and 26% for testing. In other words, this task's dataset had 992 question-passage pairs for the training, 163 for the development, and 407 for testing.

3 System

3.1 Task A: Passage Retrieval

To solve this task, we measured the similarity between the question and all Qur'anic passages and then selected the most similar passages, up to 10. We put a threshold to indicate whether the question and a passage are similar or not. If no passage has a similarity score of more than the threshold, then a "no answer" case is indicated by the system. Similarity cannot be measured directly between two passages (the question and passage in our task). However, we can convert the passages to numerical representations and then measure the similarity between the resulting representations. Embedding models, such as BERT (Devlin et al., 2018), Word2Vec (Church, 2017), and GloVe (Pennington et al., 2014), can be used to convert a given text into a numerical space. In this work, we used OpenAI's embedding model which is called "text-embedding-ada-002" (OpenAI, 2023a). According to (OpenAI, 2023b), "text-embedding-ada-002" converts a given text into a 1536-dimension embedding vector with an 81.5% performance score on SenEval, a tool designed to assess the effectiveness of sentence embeddings (Conneau and Kiela, 2018). To measure the distance between two embedding vectors, we used the cosine similarity (Rahutomo et al., 2012).

3.2 Task B: Reading Comprehension

To solve this task, we utilized a handcrafted prompt with Generative Pre-trained Transformer-3 (GPT-3.5) and Generative Pre-trained Transformer-4 (GPT-4) language models in order to retrieve the answers to a question out of the corresponding passage, if any. GPT-3.5 is based on GPT-3 which is an autoregressive model with 175 billion parameters where it exhibits remarkable proficiency across a diverse range of natural language processing tasks (Brown et al., 2020). GPT-4 is a language model much larger than GPT-3.5 with about 1.7 trillion parameters (Schreiner, 2023). GPT-4 demonstrates performance comparable to that of humans with enhanced performance in terms of accuracy and adherence to desired behavioral criteria (Team, 2023).

In Task B, the system is supposed to return all the sections that contain an answer to a question out of a passage. While dealing with GPT models, we can think of the following scenarios:

1. **Scenario 1: Asking GPT model a direct question.** If we ask GPT-3.5 or GPT-4 to give us the answers to a question without a passage, it would provide us an answer where it might or might not be true, with a more accurate answer to be provided by GPT-4.

2. **Scenario 2: Asking GPT model a question with a passage to extract answers from.** Providing the passage to GPT and asking it to give

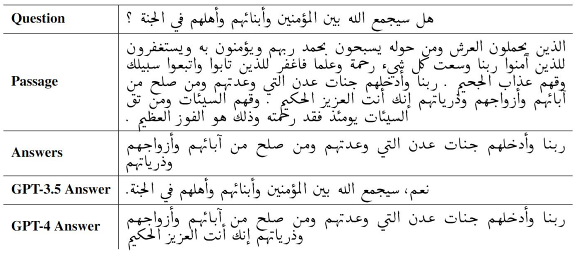
us answers to a question out of the provided passage would provide us with more reliable answers compared to scenario 1. However, there is still a chance for both models, GPT-3.5 and GPT-4, to provide us answers out of the provided passage.

3. **Scenario 3: The scenario is like scenario 2 but with making the model more determined.** When dealing with GPT-3.5 and GPT-4 models' APIs, we can control the *temperature* parameter to have lower values to get more determined answers. In other words, if we set this parameter to a value near zero, we will probably not get an answer out of the provided passage.

In our system, we followed the third scenario where we provided the GTP model with the prompt followed by the question and then the passage, along with setting the *temperature* parameter to zero. The *temperature* parameter varies between 0 and 2. Higher values yield more random output and lower values enhance the output determinism. The result of the model is not determined or fixed in every call where it sometimes returns an answer with double quotations, sometimes returned as a list with a special character in front of each answer, and so on. For that reason, we included a step that cleans the result by deleting special characters and white spaces out of the answer. The final step we have in the system is finding the corresponding start and end indices for each answer out of the passage as required by the task. If the provided answer is not in the passage, then we discard the answer since it means that the model has given an out-of-passage answer. We prompted the GPT model to return "no answer" in case the passage contains no answers to the provided question. As a result, our system returns "no answer" either if the GPT model gave a "no result" or all provided answers are out-of-passage. The prompt we used before is as follows:

ءجب على السؤال التالي من النص المرفق فقط .
لا تقم بإضافة أية شرح أو أية إجابة من خارج
النص. اكتب الإجابة أو الإجابات فقط، إن وجدت
أكثر من إجابة اكتبها على شكل تعدادات. الاجابة
يجب أن تكون فقط المقطع أو المقاطع التي تحوي
الجواب بدون أية زيادة. اجعل كل مقطع في سطر
منفصل. إن لم توجد إجابة، اكتب: "No Answer".

Fig. 1 shows an example from the dev dataset that consists of a question, a passage, and answers, along with the corresponding answers we obtained from GPT-3.5 and GPT-4.



Question	هل سيجمع الله بين المؤمنين وأبنائهم وأهلهم في الجنة ؟
Passage	الذين يحملون العرش ومن حوله يسبحون بحمد ربهم ويؤمنون به ويستغفرون للذين آمنوا ربنا وسعت كل شيء رحمة وعلما فاغفر للذين تابوا واتبعوا سبيلك وقهم عذاب الجحيم . ربنا وأدخلهم جنات عدن التي وعدتهم ومن صلح من آبائهم وأزواجهم وذرياتهم إنك أنت العزيز الحكيم . وقهم السيئات ومن تق السيئات يومئذ فقد رحمته وذلك هو الفوز العظيم .
Answers	ربنا وأدخلهم جنات عدن التي وعدتهم ومن صلح من آبائهم وأزواجهم وذرياتهم
GPT-3.5 Answer	نعم، سيجمع الله بين المؤمنين وأبنائهم وأهلهم في الجنة.
GPT-4 Answer	ربنا وأدخلهم جنات عدن التي وعدتهم ومن صلح من آبائهم وأزواجهم وذرياتهم إنك أنت العزيز الحكيم

Figure 1: Answers obtained from GPT-3.5 and GPT-4 for an example of Task B's dev dataset

4 Results

In this section, we present the results of our two models for Task A and Task B along with comparing them to the base model in each task.

4.1 Task A: Passage Retrieval

In the context of the information retrieval task, which follows a traditional ranked retrieval paradigm, the evaluation metric employed was the Mean Average Precision (MAP). Mean Average Precision (MAP) is a widely employed metric that is calculated across the entirety of a ranking(Voorhees, 2001). Instances where no answers are available were addressed by assigning complete credit to the system's "no answers" output and zero credit to all other responses. We have not trained the system since there is no method for fine-tuning the "text-embedding-ada-002" embedding model. With a threshold ranging between 0.4 and 0.95 with a 0.5 step, we found the best threshold to be 0.85 on the dev dataset with a 0.109438 MAP score and 0.267974 MRR score. The base model in this task is the BM25 model, which depends on the bag-of-words representation of the text (Amati, 2009). The BM25 model MAP and MRR scores for the dev dataset were 0.170291 and 0.313333 respectively. Using the test dataset, the BM25 model had a MAP score of 0.09036485 and an MRR score of 0.22603485 while our system

Task	Model	Score
Task A	BM25	0.090
	Similarity measurement with "ext-embedding-ada-002" embeddings	0.064
Task B	NWPR	0.326
	GPT-4-based Model	0.545

Table 1: Comparision between our methods and base models on the test dataset

achieved a MAP score of 0.06426543057 and an MRR score of 0.1608621226.

4.2 Task B: Reading Comprehension

The evaluation metric for Task B was the partial Average Precision (pAP) (Kishida, 2005), a rank-based measure designed to account for partial matching and assess the performance of a QA system in scenarios where the retrieved answer may not necessarily occupy the top rank and may only partially match one of the gold answers. Furthermore, pAP is well-suited for evaluating questions that may have one or more correct answers within the accompanying passage. This attribute makes pAP a more appropriate choice for assessing Task B compared to partial Reciprocal Rank (pRR) (Malhas and Elsayed, 2022). The baseline model to compare with is a naive whole passage retriever (NWPR) that returns the whole passage as an answer and has 0.255 and 0.3267900357 for the dev and test datasets respectively. Our GPT-4-based model scored better than the base model with pAP scores of 0.470 and 0.5393130538 for dev and test datasets respectively. Processing the results of GPT-4 gave a slice increase in performance when we tested it on the test dataset and got a pAP score of 0.5456830602. The GPT-3.5-based model yielded an exceedingly low score on the development dataset; consequently, we opted to exclude it from our comparative analysis.

Table 1 shows the results of our proposed methods compared to the corresponding base models.

5 Discussion

The results demonstrate that the OpenAI models utilized in this work provide a reasonable starting point for addressing the Qur'an QA tasks. However, there is substantial room for improvement to achieve state-of-the-art performance.

Regarding Task A, we initially attributed the low MAP score to a potential deficiency in Arabic language support. To investigate this, we employed Google Translate to render both the questions and passages into English. Subsequently, we applied the same methodology as described in Section 3.1. Surprisingly, the outcome proved to be notably inferior to the results obtained using the original Arabic dataset. We attribute this disparity to the inherent limitations of translation, which struggle to convey the precise nuances of Quranic passages accurately. Unfortunately, since "text-embedding-ada-002" embedding model is not open-sourced, it cannot be fine-tuned to fit our task.

In the context of the reading comprehension task, it is noteworthy that the GPT-3.5 prompt engineering approach performs notably worse than a naive baseline model. Conversely, the GPT-4 prompt engineering approach exhibits a significant performance improvement, surpassing the naive baseline by a considerable margin. However, it is essential to recognize that while GPT-4 demonstrates superior adherence to prompts compared to GPT-3.5, its behavior is not entirely deterministic, and variations can occur. Additionally, we must address the issue of "Prompt Injection", wherein a prompt could be introduced after the initial prompt, potentially altering the model's behavior. While this behavior was more prevalent in GPT-3.5, it is less pronounced in GPT-4. For instance, when applying the GPT-4-based model to the test set, we encountered very few cases like the question من هو المؤمن بتعريف القرآن ؟, which yielded the answer المؤمن بتعريف القرآن هو الذين آمنوا indicating that GPT-4 ignored entirely the prompt we mentioned in Section 3.2 and was appended before the question.

Conclusion

In this paper, we presented our methods for solving the two tasks of Qur'an QA 2023 Shared Task. We solved the passage retrieval task by (1) using "text-embedding-ada-002" embeddings to convert the questions and passages into a numerical representation, (2) calculating the cosine distances between the questions and answers, and then (3) selecting the top 10 similar passages. This method achieved a score lower than the baseline BM25 model with a MAP score equals to 0.06426543057. The reading comprehension task was solved using a handcrafted prompt along with GPT-4 with the *temperature* parameter equals to zero. Our method achieved a pAP score equals to 0.5456830602, approximately a 67% increase in performance compared to the baseline model.

Limitations

One of the limitations is the usage cost of ChatGpt APIs, especially GPT-4 which is approximately 10x the cost of using GPT-3.5. Another limitation is the explainability of the results. Providing explanations to answers is a challenging task and could be achieved partially by several methods as in (Zakieh and Alpkocak, 2021). However, the methods used in (Zakieh and Alpkocak, 2021) cannot be applied to the methods we used in this work.

References

Giambattista Amati. 2009. *BM25*, pages 257–260. Springer US, Boston, MA.

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

Kenneth Ward Church. 2017. Word2vec. *Natural Language Engineering*, 23(1):155–162.

Alexis Conneau and Douwe Kiela. 2018. Senteval: An evaluation toolkit for universal sentence representations. *arXiv preprint arXiv:1803.05449*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Kazuaki Kishida. 2005. *Property of average precision and its generalization: An examination of evaluation indicator for information retrieval experiments*. National Institute of Informatics Tokyo, Japan.

Rana Malhas and Tamer Elsayed. 2020. AyaTEC: Building a Reusable Verse-based Test Collection for Arabic Question Answering on the Holy Qur'an. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 19(6):1–21.

Rana Malhas 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 *Proceedings of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT5) at the 13th Language Resources and Evaluation Conference (LREC 2022)*, pages 79–87.

Rana Malhas, Watheq Mansour, and Tamer Elsayed. 2023. Qur'an QA 2023 Shared Task: Overview of Passage Retrieval and Reading Comprehension Tasks over the Holy Qur'an. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore.

Rana R Malhas. 2023. *ARABIC QUESTION ANSWERING ON THE HOLY QUR'AN*. Ph.D. thesis.

OpenAI. 2023a. Embeddings - openai api. September 12, 2023.

OpenAI. 2023b. New and improved embedding model. September 12, 2023.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Faisal Rahutomo, Teruaki Kitasuka, and Masayoshi Aritsugi. 2012. Semantic cosine similarity. In *The 7th international student conference on advanced science and technology ICAST*, volume 4, page 1.

Maximilian Schreiner. 2023. Gpt-4 architecture, datasets, costs and more leaked. September 12, 2023.

Mihai Surdeanu, Massimiliano Ciaramita, and Hugo Zaragoza. 2011. Learning to rank answers to non-factoid questions from web collections. *Computational linguistics*, 37(2):351–383.

Marwan N. Swar. 2007. *Mushaf Al-Tafseel Al-Mawdoo'ee*. Dar Al-Fajr Al-Islami, Damascus.

GPT-4 Research Team. 2023. Gpt-4 technical report. Technical report.

Ellen M Voorhees. 2001. Evaluation by highly relevant documents. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 74–82.

Abdul Razak Zakieh and Adil Alpkocak. 2021. Classification of medical transcriptions with explanations. Technical report, EasyChair.

WojoodNER 2023:
The First Arabic Named Entity Recognition Shared Task

Mustafa Jarrar[1] **Muhammad Abdul-Mageed**[2,3] **Mohammed Khalilia**[1] **Bashar Talafha**[2]
AbdelRahim Elmadany[2] **Nagham Hamad**[1] **Alaa' Omar**[1]
[1]Birzeit University, Palestine
[2]Deep Learning & Natural Language Processing Group, The University of British Columbia
[3]Department of Natural Language Processing & Department of Machine Learning, MBZUAI
`mjarrar@birzeit.edu` `muhammad.mageed@ubc.ca`

Abstract

We present WojoodNER-2023, the first Arabic Named Entity Recognition (NER) Shared Task. The primary focus of WojoodNER 2023 is on Arabic NER, offering novel NER datasets (i.e., Wojood) and the definition of subtasks designed to facilitate meaningful comparisons between different NER approaches. WojoodNER-2023 encompassed two Subtasks: FlatNER and NestedNER. A total of 45 unique teams registered for this shared task, with 11 of them actively participating in the test phase. Specifically, 11 teams participated in FlatNER, while 8 teams tackled NestedNER. The winning teams achieved F_1 scores of 91.96 and 93.73 in FlatNER and NestedNER, respectively.

1 Introduction

NER is a fundamental task in Natural Language Processing (NLP), especially in information extraction and language understanding (Jarrar et al., 2023a). The objective of NER is to identify and classify named entities in a given text into predefined categories, such as "*person*", "*location*", "*organization*", "*event*", and "*occupation*". NER is also a critical task for many NLP applications, such as question-answering systems (Shaheen and Ezzeldin, 2014), knowledge graphs (James, 1991), and semantic search (Guha et al., 2003), interoperability (Jarrar et al., 2011) among others. Named entities can either be flat or nested. For instance, in the sentence "`Cairo Bank announces its profit in` 2023", there are two flat entities: "`Cairo Bank`" is tagged as `ORG` (i.e., organization) and "`2023`" as `DATE`. In nested NER, entity mentions contained inside other entity mentions are also considered named entities. In this case, "`Cairo`", is tagged as `GPE` (i.e., geopolitical entity). Section 3 illustrates more examples. As will be discussed in Section 2, research in Arabic NER is currently limited, particularly in the context of nested entities. This limitation is not exclusive to Modern Standard Arabic (MSA) but extends to various Arabic dialects across diverse domains and NER subtypes. The majority of existing research on Arabic NER primarily emphasizes flat entities to cover a limited set of entity types, mainly "person", "organization", and "location".

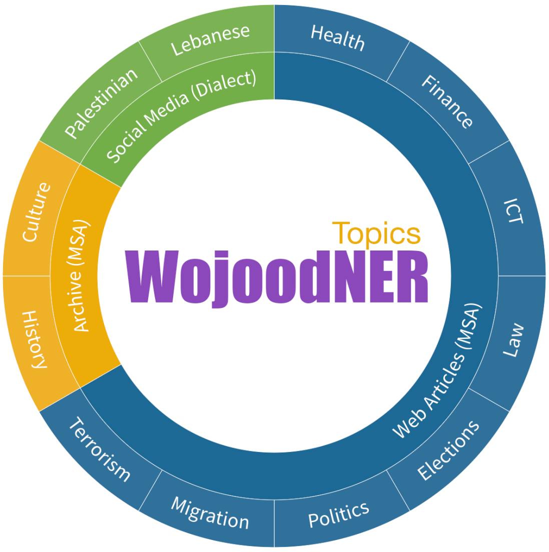


Figure 1: Topics in the Wojood NER corpus.

In this paper, we provide an overview of the WojoodNER-2023 Shared Task[1], which represents a significant step forward in advancing NER research in the Arabic language. The shared task encompasses subtask1 (FlatNER) and subtask2 (NestedNER). For this competition, we grant participants access to the Wojood corpus (Jarrar et al., 2022)[2], a substantial and diverse Arabic NER dataset known as Wojood. As shown in Figure 1, Wojood is particularly notable for its scale, containing approximately 550K tokens. About 12% of the corpus was collected from social media in *Palestinian* and *Lebanese* dialects Curras and Baladi corpora (Haff et al., 2022). The remaining $\sim$ 88% is in MSA, covering multiple domains, including

[1] SharedTask Call: `https://dlnlp.ai/st/wojood/`

[2] Wojood Corpus: `https://sina.birzeit.edu/wojood/`

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 748–758
December 7, 2023 ©2023 Association for Computational Linguistics

health, *finance*, *politics*, *ICT*, *terrorism*, *migration*, *history and culture*, and *law and elections*, making it a rich resource for various research purposes. Wojood was annotated manually using 21 entity types, offering a rich Arabic NER corpus.

The primary objective of this shared task is to encourage participants to explore different NER methodologies. Teams were invited to experiment with various approaches, ranging from classical machine learning to advanced deep learning and transformer-based techniques, among others. The shared task generated a remarkably diverse array of submissions. A total of 45 teams registered to participate in the shared task. Among these, 11 teams successfully submitted their models for evaluation on the blind test set during the final phase of the competition. As a result, we received 11 papers that provide detailed insights into the results achieved by these teams for either one or both of the subtasks.

The rest of the paper is organized as follows: Section 2 provides a brief overview of Arabic NER. We describe the two subtasks and WojoodNER-2023 restrictions in Section 3. Section 4 introduces shared task datasets and evaluation setup. We present participating teams and shared task results and provide a high-level description of submitted systems in Section 5. We conclude in Section 6.

2 Literature Review

NER has been a long-standing research area, with significant advances made in recent years. As will be discussed in this section, early NER approaches focused on identifying and classifying flat named entities, and recent research has focused on nested NER. In this section, our primary focus is exclusively on Arabic NER research, encompassing corpora, methodologies and shared tasks.

Corpora. Most of the available Arabic NER corpora are annotated as flat NER. `ANERCorp` (Benajiba et al., 2007), sourced from the news domain (MSA text), comprises $\sim$ 150k tokens. Its main emphasis is directed towards four distinct entity types. `CANERCorpus` (Salah and Zakaria, 2018) is dedicated to Classical Arabic (CA) and encompasses a dataset of 258K tokens. This corpus is annotated for a total of 14 entity types, all of which pertain to religious entities. `ACE2005` (Walker et al., 2005) is a multilingual corpus that incorporates Arabic text encompassing *five* distinct types of entities. `Ontonotes5` (Weischedel et al., 2013) dataset consists of approximately 300K tokens, meticulously annotated with 18 distinct entity types. Nevertheless, these corpora were collected a long time ago and mainly cover the media and politics domains; hence, may not be representative of the current state of Arabic language use. This is especially the case since language models are known to be sensitive to temporal and domain shifts. Recently, Jarrar et al. (2022) proposed `Wojood`, the largest Arabic NER corpus. It is distinctive for its support of both flat and nested entity annotations, making it a crucial resource utilized in this shared task. It comprises roughly 550K tokens encompassing a diverse range of 21 unique entity types, spanning both MSA and two dialectal Arabic forms (the Palestinian Curras2 and Lebanese Baladi corpora (Haff et al., 2022).

Methodologies. Various studies explore Arabic NER by employing various approaches, with some researchers focusing on rule-based (Shaalan and Raza, 2007; Jaber and Zaraket, 2017) and machine learning (Settles, 2004; Abdul-Hamid and Darwish, 2010; Zirikly and Diab, 2014; Dahan et al., 2015; Darwish et al., 2021) strategies. Recent researches embrace deep learning methodologies including character and word embeddings with Long-Short Term Memory (LSTM) networks (Ali et al., 2018), BiLSTM followed by Conditional Random Field (CRF) models (El Bazi and Laachfoubi, 2019; Khalifa and Shaalan, 2019), Deep Neural Networks (DNN) (Gridach, 2018), and pretrained Language Models (LM) (Jarrar et al., 2022; Liqreina et al., 2023). Wang et al. (2022) proposed a survey that extensively explores different approaches to nested entity recognition, encompassing rule-based, layered-based, region-based, hypergraph-based, and transition-based methodologies. Fei et al. (2020) proposed a multitask learning approach for nested NER that employs a dispatched attention model. Ouchi et al. (2020) proposed an approach for nested NER that involves enumerating all region representations from the contextual encoding sequence and then assigning a category label to each of them.

Shared tasks. While there are multiple shared tasks for NER in various languages and domains, such as the `MultiCoNER` for multilingual complex NER (Malmasi et al., 2022), the `HIPE-2022` for NER and linking in multilingual historical documents (Ehrmann et al., 2022), the `RuNNE-2022` for nested NER in Russian (Artemova et al., 2022), and the NLPCC2022 for extracting entities in the material science domain (Cai et al., 2022). To the best of our knowledge, there has been no dedicated shared task for Arabic NER. Therefore, we initiate this shared task with the aim of being the inaugural event in this specific domain.

3 Task Description

To the best of our knowledge, WojoodNER-2023 is recognized as the inaugural shared task in Arabic NER. In this competition, we present two distinct subtasks—one for “**FlatNER**” and the other for “**NestedNER**”. These subtasks are of paramount importance in addressing the challenges inherent in Arabic NER processing. We now describe each subtask in detail.

3.1 Subtask1 – FlatNER

In FlatNER, each token in the data is labeled with only one tag. The participants in this subtask are expected to develop models to classify each token as a multi-class classification problem. An example of the FlatNER data is shown in Figure 2. The Wojood annotation guidelines were designed for nested entities only, therefore, the flat entities were derived from the nested entities by taking the top-level entity mentions (i.e., topmost tags).

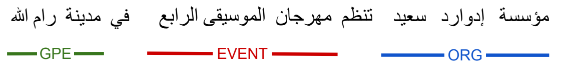

مؤسسة إدوارد سعيد تنظم مهرجان الموسيقى الرابع في مدينة رام الله
ORG
EVENT
GPE

Figure 2: Flat NER example

3.2 Subtask2 – NestedNER

In the NestedNER subtask, each token can have one or more tags. In this data, we will find entity mentions inside other entity mentions as demonstrated in Figure 3. For instance, the phrase “مؤسسة إدوارد سعيد” is annotated as ORG, which is the same as the flat annotation in Figure 2. However, in nested NER, it contains another entity mention “إدوارد سعيد” tagged with PERS.

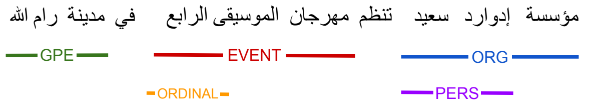

مؤسسة إدوارد سعيد تنظم مهرجان الموسيقى الرابع في مدينة رام الله
ORG
EVENT
GPE
PERS
ORDINAL

Figure 3: Nested NER example

3.3 Restrictions

This section outlines the stipulations and directives that govern participants’ engagement in the WojoodNER 2023 Shared Task. These regulatory directives and guidelines establish an equitable competitive environment for all participants, ensuring transparency and impartiality throughout the duration of the WojoodNER 2023 Shared Task. They also ensure the credibility of the task’s assessment procedure, which was published on the shared task official website frequently asked question page.

External data. Participants are strictly prohibited from using external data from previously labeled datasets or employing taggers that have been previously trained to predict named entities. The use of any resources with prior knowledge related to NER is not allowed.

Data format constraints. The submission to the task consists of one file containing the model prediction in CoNLL format. The CoNLL format should include multiple columns space-separated. The first column is reserved for the tokens, while all subsequent columns are used for the tags. In the case of nested NER, the tag columns have a predefined order, which we specified on the shared task webpage[3]. The IOB2 (Sang and Veenstra, 1999) scheme is used for the submission, which is the same format used in the Wojood dataset. Finally, text segments are separated by a blank line.

Pretrained models. The participants are allowed to utilize pretrained transformer models such as “*BERT*” (Devlin et al., 2018) and word representations like “*Word2Vec*” (Church, 2017) and “*ELMo*” (Peters et al., 2018) for the purpose of transfer learning. It is worth noting that our baseline model is based on BERT.

Linguistic features. When considering the incorporation of linguistic features to enhance the dataset, participants are permitted to include part-of-speech tagging and syntactic layers within their code.

4 Shared Task Datasets and Evaluation

This section presents the dataset, evaluation metrics, and the submission process.

Datasets. WojoodNER-2023 shared task employs the Wojood corpus as its primary dataset (Jarrar et al., 2022). The Wojood corpus encompasses approximately 550K tokens, spanning both MSA and two Arabic dialects, annotated using 21 entity

[3] https://dlnlp.ai/st/wojood/

Entity Name	NER Tag	FlatNER				NestedNER			
		TRAIN	DEV	TEST	Total	TRAIN	DEV	TEST	Total
Person	PERS	4,496	650	1,409	**6,555**	4,994	730	1,562	**7,286**
Group of people	NORP	3,505	488	948	**4,941**	3747	520	1006	**5273**
Occupation	OCC	3,774	544	1,058	**5,376**	3,887	551	1,95	**5,533**
Organization	ORG	10,731	1,566	3,047	**15,344**	13,174	1,869	3,738	**18,781**
GeoPolitical Entity	GPE	8,133	1,132	2,281	**11,546**	15,300	2,163	4,315	**21,778**
Geographical location	LOC	510	63	168	**741**	619	76	204	**899**
Facility (e.g., landmarks)	FAC	689	85	165	**939**	880	111	224	**1,215**
Product	PRODUCT	36	5	13	**54**	36	5	14	**55**
Event	EVENT	1,863	253	556	**2,672**	1,934	267	577	**2,778**
Date	DATE	10,667	1,567	3,091	**15,325**	11,290	1,656	3,288	**1,6234**
Time	TIME	286	55	84	**425**	288	55	84	**427**
Language	LANGUAGE	131	15	51	**197**	132	15	51	**198**
Website	WEBSITE	434	45	128	**607**	434	45	128	**607**
Law	LAW	374	44	78	**496**	374	44	78	**496**
Cardinal	CARDINAL	1,245	182	360	**1,787**	1,263	183	363	**1,809**
Ordinal	ORDINAL	2,805	410	858	**4,073**	3,488	504	1,070	**5,062**
Percent	PERCENT	105	13	19	**137**	105	13	19	**137**
Quantity	QUANTITY	44	3	7	**54**	46	3	8	**57**
Unit	UNIT	7	0	2	**9**	48	3	9	**60**
Money	MONEY	171	20	36	**227**	171	20	36	**227**
Currency	CURR	19	1	5	**25**	179	21	41	**241**
	Total	**50,025**	**7,141**	**14,364**	**71,530**	**62,389**	**8,854**	**17,910**	**89,153**

Table 1: Distribution of NER tags in WojoodNER-2023 Subtask1 (i.e., FlatNER) and Subtask2 (i.e., NestedNER) across the training (i.e., TRAIN) , development (i.e., DEV), and test (i.e., TEST) splits for the WojoodNER-2023.

types. Wojood annotation guidelines are optimized for nested Arabic NER annotations. However, for the purposes of the shared task, we generate a flat NER dataset by reducing the nested NER annotation to the top level only as demonstrated in Figure 2 and 3. For both subtasks, we split the data 70/10/20 for training, development, and test dataset respectively at the domain level. This split ensures similar data distribution across the three datasets. Table 1 present the statistics and characteristics of WojoodNER-2023's subtask1 and subtask2 training, development, and test datasets.

Evaluation metrics. The official evaluation metric for subtask1 and subtask2 is the macro-averaged $\mathsf{F_1}$ score. In addition to this metric, we also report system performance in terms of `Precision`, `Recall`, and `Accuracy` for submissions to both subtasks.

Submission roles. We allowed participant teams to submit up to *four* runs for each test set, for both subtasks. In each one, we strictly retain only the submission with the highest score from each participating team. Although the official results were solely derived from the blind test set. To streamline the evaluation of participant systems, we have set up two separate CodaLab (Pavao et al., 2023) competitions for scoring each subtask.[4] We are keeping the CodaLab (Pavao et al., 2023) for each subtask active even after the official competition has concluded. This is aimed at facilitating researchers who wish to continue training models and evaluating systems with the shared task's blind test sets. As a result, we will not disclose the labels for the test sets in any of the subtasks.

5 Shared Task Teams & Results

5.1 Participating Teams

In total, we received 45 unique team registrations. At the testing phase, a total of 57 valid entries were submitted by 12 unique teams. We received 35 submissions for FlatNER from *eleven* teams and 22 submissions for NestedNER from *eight* teams. Table 2 lists the teams, their affiliation, and the tasks they participated in (Subtask1 – FlatNER and Subtask2 – NestedNER). From 12 teams we received 11 description papers from which we accepted 8 for publication and 3 were rejected (for quality or not adhering to the shared task guidelines).

[4] The different CodaLab competitions are available at the following links: `subtask-1` and `subtask-2`.

Team	Affiliation	Task
Alex-U 2023 NLP (Hussein et al., 2023)	Alexandria University	1,2
AlexU-AIC (Elkordi et al., 2023)	Alexandria University	1,2
AlphaBrains (Ehsan et al., 2023)	University of Gujrat, Pakistan	1,2
ARATAL	IPSA	1
El-Kawaref (Elkaref and Elkaref, 2023)	German University in Cairo	1
ELYADATA (Laouirine et al., 2023)	ELYADATA	1,2
Fraunhofer IAIS	Fraunhofer IAIS	1
LIPN (El Khbir et al., 2023)	LIPN, Université Paris 13	1,2
Lotus (Li et al., 2023)	MBZUAI	1,2
R00	Jordan University of Science and Technology	1,2
Think NER	Ulm University	1,2
UM6P & UL (El Mahdaouy et al., 2023)	Mohammed VI Polytechnic University	1,2

Table 2: List of teams that participated in either one or both subtasks. Teams with accepted papers are cited.

5.2 Baselines

For both subtasks, we fine-tune the AraBERT$_{v2}$ (Antoun et al., 2020) and ARBERT$_{v2}$ (Abdul-Mageed et al., 2021) pre-trained models using the training data that is specific to each subtask for 20 epochs and employed a learning rate of $1e-5$, along with a batch size of 16. To ensure model optimization, we incorporate early stopping with a patience setting of 5. After each epoch, we evaluated the model's performance and selected the best-performing checkpoints based on their performance on the respective development set. Subsequently, we present the performance metrics of the best-performing model on the test datasets.

Rank	Team	F1	Pre.	Rec.
1	LIPN	**91.96**	92.56	91.36
2	El-Kawaref	91.95	91.43	92.48
3	ELYADATA	91.92	91.88	91.96
4	Alex-U 2023 NLP	91.80	91.61	92.00
5	Think NER	91.25	90.76	91.73
6	ARATAL	91.13	90.49	91.77
7	UM6P & UL	91.13	90.70	91.57
8	AlexU-AIC	91.13	91.33	90.92
	Baseline-I (ARBERT$_{v2}$)	89.20	88.32	90.09
	Baseline-II (AraBERT$_{v2}$)	87.33	86.00	88.00
9	AlphaBrains	87.15	87.45	87.58
10	Lotus	83.39	80.90	86.04
11	R00	76.99	76.67	77.31
12	Fraunhofer IAIS	64.45	65.53	63.40

Table 3: Results of Subtask1 – FlatNER.

5.3 Results

Table 3 and Table 4 present the leaderboards of Subtask1 – FlatNER and Subtask2 – NestedNER, respectively, sorted by macro-F_1 in descending order. The macro-F_1 score for each team represents the highest score among the four allowed submissions for each task.

Rank	Team	F1	Pre.	Rec.
1	Elyadata	93.73	93.99	93.48
2	UM6P & UL	93.03	92.46	93.61
3	AlexU-AIC	92.61	92.10	93.13
4	LIPN	92.45	92.31	92.59
	Baseline-I (ArBERT$_{v2}$)	91.68	91.01	92.35
5	Think NER	91.4	90.03	92.82
	Baseline-II (AraBERT$_{v2}$)	91.06	90.74	91.38
6	Alex-U 2023 NLP	90.01	89.39	90.63
7	AlphaBrains	88.84	88.45	89.23
8	Lotus	76.02	82.19	70.72

Table 4: Results of Subtask2 – NestedNER.

For FlatNER, LIPN team (El Khbir et al., 2023) achieved the highest F_1 score of 91.96, while El-Kawaref (Elkaref and Elkaref, 2023) came in second place with 91.95 and Elyadata in third place with 91.92. Notably, on FlatNER, *eight* teams surpass our two baselines performance, as seen in Table 3. Moreover, the winning team (i.e, LIPN (El Khbir et al., 2023)) outperforms the Baseline-I by 2.76%. *Three* teams underperform Baseline-I and Baseline-II. However, the gap between the baseline-I and the worst-performing model is about 24.75%. We also notice that the difference in the F_1 score among the top *eight* teams is marginal ($\sigma = 0.41$).

We also analyzed the performance at the entity-type level in FlatNER and we noticed that certain entity types are more challenging to learn by all submitted models, including the baseline. The main reason for their low performance is the rarity of those entities in the dataset, with frequency reaching as low as 9 for UNIT and 54 for both PRODUCT

Team Name	F_1	*Preprocessing*	Features			Techniques						
			TF-IDF	*Word Embeds*	*Resampling*	*Neural Nets*	*Contrast. L*	*Ensemble*	*Adapter*	*Multitask*	*PLM*	*Hie. Cls*
FlatNER												
LIPN	91.96					✓		✓			✓	
El-Kawaref	91.95	✓									✓	
Elyadata	91.92	✓			✓	✓					✓	
Alex-U 2023 NLP	91.80						✓				✓	
ThinkNER	91.25											
UM6P & UL	91.13									✓	✓	
AlexU-AIC	91.13	✓									✓	✓
ARATAL	91.13							✓			✓	
AlphaBrains	87.51			✓		✓				✓		
Lotus	83.39	✓	✓						✓		✓	
Fraunhofer IAIS	64.45										✓	
NestedNER												
Elyadata	93.73	✓			✓	✓					✓	
UM6P & UL	93.03									✓	✓	
AlexU-AIC	92.61	✓									✓	✓
LIPN	92.45					✓		✓			✓	
ThinkNER	91.40											
Alex-U 2023 NLP	76.02						✓				✓	
AlphaBrains	88.84			✓		✓				✓		
Lotus	76.02	✓	✓						✓		✓	

Table 5: Summary of approaches used by participating teams in subtask1 (i.e., FlatNER) and subtask2 (i.e., NestedNER). Teams are sorted by their performance on the official metric, Macro-F_1 score. The term “Neural Nets" refers to any model based on neural networks (e.g., FFNN, RNN, CNN, and Transformer) trained from scratch. PLM refers to neural networks pretrained with unlabeled data such as ARBERT$_{v2}$. (Hie. Cls, hierarchical classification approach); (Contrast. L, contrastive learning).

and QUANTITY. The highest F_1 for PRODUCT is 61.54 (Hussein et al., 2023), for QUANTITY 50.00 (Elkaref and Elkaref, 2023) and for UNIT 50.00 (Elkaref and Elkaref, 2023; Hussein et al., 2023; Laouirine et al., 2023). CURR also achieved low performance among all participants ($F_1 \leq 66.67$) with exception to (Elkaref and Elkaref, 2023), which reported an $F_1 = 88.89$, despite its low frequency in the data of 25 occurrences. Our Baseline-II achieved low performance on the three entities mentioned above, but outperformed all submitted models on QUANTITY with an $F_1 = 75.00$.

For NestedNER, the ELYADATA team (Laouirine et al., 2023) ranks in the first position with an F_1 score of 93.73, followed by UM6P & UL team (El Mahdaouy et al., 2023) with a score of 93.09 and in third place AlexU-AIC with a score of 92.61. Notably, there are *four* teams that outperform baseline-I with F_1 score gap between the baseline and the best model of 2.05%. Whereas, the gap between baseline-I and the worst-performing model is about 15.66%. The difference in the F_1 score among the top four teams is $\sigma = 0.57$.

The performance at the entity level for NestedNER is analyzed to explain the challenge for all submitted models. As previously mentioned, the scarcity of some entities in the dataset influences the performance of some entity types in FlatNER. This scarcity influences the results on NestedNER, too. The product, quantity, and website obtained the lowest performance in all models. The highest performance for the product is 66.67% which is obtained by ThinkNER team. For the quantity, the 63.16% F1-score is obtained by (El Mahdaouy et al., 2023). For website, the best performance is 69.26% F1-score. The unit entity also achieved a low performance among all teams except (Elkordi et al., 2023) which obtained 80% F1-score.

The final observation we will highlight is the pattern of scores across the two subtasks, where all scores (micro-F1, precision, and recall) are higher in NestedtNER compared to FlatNER. This was also observed in the baseline (Jarrar et al., 2022).

It may seem counter-intuitive, but in fact, FlatNER is harder than NestedtNER. Recall that the Wojood annotation guideline was optimized for nested NER and the flat annotations are simply the top-level tags found in the nested annotations. This conversion from nested to flat annotations caused some tokens to have conflicting tags in the dataset, which breaks the high annotation consistency found in the nested dataset. Another reason for this pattern is the co-occurrence among nested tags. For instance, an entity mention tagged with OCC is more likely to have nested entity mentions tagged as ORG or PERS, rather than entity mentions tagged with PRODUCT, EVENT or DATE.

5.4 General Description of Submitted Systems

All the models submitted to the shared task adopt the transfer learning approach, leveraging pre-trained models trained on various data sources. Generally, we observe that the top-performing models addressed the challenge of identifying nested entities of the same type, a limitation described by Jarrar et al. (2022).

Table 5 summarizes the techniques employed by the participating teams in the WojoodNER-2023 shared task. The common theme is the use of pre-trained models by all participants. The choice of models include AraBERT (Antoun et al., 2020), MARBERT (Abdul-Mageed et al., 2021), ARBERT (Abdul-Mageed et al., 2021), XLM-R (Conneau et al., 2019), and CAMelBERT (Inoue et al., 2021). AraBART$_{v2}$ is the pre-trained language model used the most in the shared task, where it was utilized by seven teams in FlatNER and five teams in NestedNER. MARBERT comes in second place in terms of usage, where six teams used it in both subtasks (Figure 4).

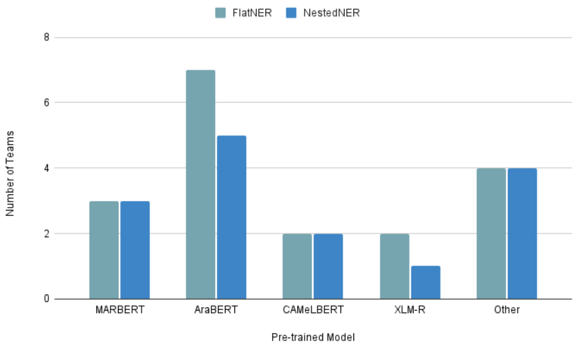


Figure 4: Distribution of pre-trained models across teams.

It was observed in the submissions that compare AraBERT with MARBERT and CAMeLBERT that the AraBERT transformer consistently outperformed the others. This is noteworthy, especially considering that AraBERT is pre-trained solely on MSA data and has a smaller size than both MARBERT and CAMeLBERT.

Other transformer-based pre-trained models were also utilized. For instance, Elyadata fine-tuned BioBERT (Lee et al., 2020), but the results were much worse than the baseline, which is expected since BioBERT is trained on English biomedical corpus. In a comparative study, the UM6P & UL (El Mahdaouy et al., 2023) team explored the capabilities of QARiB (Abdelali et al., 2021), a model pre-trained specifically on Arabic tweets, against ARBERT$_{v2}$ (Abdul-Mageed et al., 2021), which is trained on an expansive and diverse Arabic datasets. Their finding shows ARBERT$_{v2}$'s superiority over other models. The rest of this section will discuss the systems submitted by each team in more details.

We start by LIPN (El Khbir et al., 2023) team, who relies on converting the task from sequence labeling to span classification task. Their approach classifies all possible spans within a sequence. For FlatNER, they employ a two-step decoding process: 1) non-entity spans are filtered out, and 2) for the remaining spans, a maximum independent set algorithm is employed to get the optimal set of entity spans. This fusion of algorithmic techniques with machine learning, coupled with the task's reformation, achieved state-of-the-art results for FlatNER and enabled the LIPN (El Khbir et al., 2023) team to secure first place in FlatNER and fourth place in the NestedNER.

UM6P & UL (El Mahdaouy et al., 2023) utilized multi-task learning similar to (Jarrar et al., 2022). The sequence is encoded using a transformer encoder and each entity type has one multi-class classification head to predict the IOB2 tag for each token. The model is trained with multiple objectives including cross-entropy loss, dice loss to handle class imbalance, Tversky loss to balance false positives and false negatives, and focal loss to down-weight easy examples. All four objectives are combined as a weighted sum, the authors refer to the unified loss. Additionally, the authors used variance penalty loss that computes the variance across all task losses. The authors experimented with different loss configurations and pre-trained models, using the unified loss and variance loss with ARBERT$_{v2}$ provided the best performance, ranking the team seventh in FlastNER and second in NestedNER.

ELYADATA (Laouirine et al., 2023) team developed the best-performing NestedNER system. They reformulated the task as a denoising problem. DiffusionNER model architecture (Shen et al., 2023) is used with AraBERT, which introduces noise spans to the gold entity boundaries and is trained to reconstruct the entity boundaries. During the inference phase, it picks noisy spans from a standard Gaussian distribution and then produces named entities by leveraging the learned reverse diffusion process. This novel approach enabled the ELYADATA (Laouirine et al., 2023) team to get first place and achieve state-of-the-art outcomes in NestedNER.

AlexU-AIC (Elkordi et al., 2023) technique relies on machine reading comprehension. In their approach, they formulate a query for each entity type, totaling 21 queries, one for each entity type. Based on the query, the model extracts the answer span from the sequence. Their architecture consists of a transformer encoder followed by two binary classifiers, one classifies if the token is the start of the answer span and another classifies if the token is the end of the answer span. The authors also adopted the stochastic weight averaging technique, in which they average the weights of the four best-performing checkpoints. The team is ranked eighth in FlatNER and third in NestedNER.

AlphaBrains (Ehsan et al., 2023) developed a multi-task learning technique that is similar to (Jarrar et al., 2022), but it employes BiLSTM encoder instead of a transformer. The input to the BiLSTM is a concatenation of learned word embeddings and ELMo representations. The team is ranked ninth in FlatNER and seventh in NestedNER.

El-Kawaref (Elkaref and Elkaref, 2023) proposes StagedNER for FlatNER. In the first stage, the transformer encoder is fine-tuned based IOB2 classification task. In that stage, the authors also used part-of-speech (POS) tagging to improve model performance. The second stage also fine-tunes the transformer encoder on entity type classification task and it takes IOB2 tags as an additional input. During training the authors use the ground truth IOB2 tags and in inference, they use the predicted tags. The team is ranked second in FlatNER.

Alex-U 2023 NLP (Hussein et al., 2023) developed AraBINDER. The approach relies on a contrastive learning objective, where the goal is to maximize the similarity between the entity mention span and its entity type and minimize the similarity with the negative classes. To do that, the authors use a bi-encoder, one for encoding the named entity type and another for encoding the named entity mention. The team is ranked fourth in FlatNER and sixth in NestedNER.

Lotus (Li et al., 2023) proposes a model also inspired by (Jarrar et al., 2022). Their model is based on XLM-R with 21 classification heads, one classifier for each entity type and each classifier is a multi-class that outputs one of the IOB2 tags. The team is ranked tenth in the FlatNER and eighth in the NestedNER.

6 Conclusion and Future Work

In this paper, we present the outcomes of WojoodNER-2023, the inaugural shared task dedicated to both flat and nested NER challenges in the Arabic language. The results obtained from the participating teams underscore the persistent challenges associated with NER. However, it is promising to observe that various innovative approaches, often harnessing the capabilities of language models, have demonstrated their effectiveness in addressing this complex task. As we move forward, we remain committed to further advancing research in this domain. Our vision includes ongoing efforts to enhance the field of Arabic NER, incorporating the valuable insights gained from WojoodNER-2023 and continuing to explore innovative solutions. We plan to extend the Wojood corpus to include more dialects. We plan to include the Syrian Nabra dialects (Nayouf et al., 2023) as well as the four dialects in the Lisan (Jarrar et al., 2023b) corpus.

Acknowledgment

We would like to thank Sana Ghanem for helping us with data annotations, and Tymaa Hammouda for her technical support during the organization of the task.

7 Limitations

While our aim was to achieve the broadest possible coverage, it is essential to acknowledge that WojoodNER-2023 primarily concentrated on MSA data, with only a limited representation of dialects, specifically covering two dialects, Palestinian and Lebanese.

References

Ahmed Abdelali, Sabit Hassan, Hamdy Mubarak, Kareem Darwish, and Younes Samih. 2021. Pretraining bert on arabic tweets: Practical considerations. *arXiv preprint arXiv:2102.10684*.

Ahmed Abdul-Hamid and Kareem Darwish. 2010. Simplified feature set for Arabic named entity recognition. In *Proceedings of the 2010 Named Entities Workshop*, Uppsala, Sweden. Association for Computational Linguistics.

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2021. Arbert & marbert: Deep bidirectional transformers for arabic.

Mohammed NA Ali, Guanzheng Tan, and Aamir Hussain. 2018. Bidirectional recurrent neural network approach for arabic named entity recognition. *Future Internet*, 10(12):123.

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

Ekaterina Artemova, Maxim Zmeev, Natalia Loukachevitch, Igor Rozhkov, Tatiana Batura, Vladimir Ivanov, and Elena Tutubalina. 2022. Runne-2022 shared task: Recognizing nested named entities.

Yassine Benajiba, Paolo Rosso, and José Miguel Benedíruiz. 2007. Anersys: An arabic named entity recognition system based on maximum entropy. In *Computational Linguistics and Intelligent Text Processing: 8th International Conference, CICLing 2007, Mexico City, Mexico, February 18-24, 2007. Proceedings 8*, pages 143–153. Springer.

Borui Cai, He Zhang, Fenghong Liu, Ming Liu, Tianrui Zong, Zhe Chen, and Yunfeng Li. 2022. Overview of nlpcc2022 shared task 5 track 2: Named entity recognition. In *Natural Language Processing and Chinese Computing*, pages 336–341, Cham. Springer Nature Switzerland.

Kenneth Ward Church. 2017. Word2vec. *Natural Language Engineering*, 23(1):155–162.

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

Fadl Dahan, Ameur Touir, and Hassan Mathkour. 2015. First order hidden markov model for automatic arabic name entity recognition. *International Journal of Computer Applications*, 123(7).

Kareem Darwish, Nizar Habash, Mourad Abbas, Hend Al-Khalifa, Huseein T. Al-Natsheh, Houda Bouamor, Karim Bouzoubaa, Violetta Cavalli-Sforza, Samhaa R. El-Beltagy, Wassim El-Hajj, Mustafa Jarrar, and Hamdy Mubarak. 2021. A panoramic survey of natural language processing in the arab worlds. *Commun. ACM*, 64(4):72–81.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Maud Ehrmann, Matteo Romanello, Sven Najem-Meyer, Antoine Doucet, and Simon Clematide. 2022. Overview of hipe-2022: Named entity recognition and linking in multilingual historical documents. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 423–446, Cham. Springer International Publishing.

Toqeer Ehsan, Amjad Ali, and Ala Al-Fuqaha. 2023. Alphabrains at wojoodner shared task: Arabic named entity recognition by using character-based context-sensitive word representations. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*.

Ismail El Bazi and Nabil Laachfoubi. 2019. Arabic named entity recognition using deep learning approach. *International Journal of Electrical & Computer Engineering (2088-8708)*, 9(3).

Niama El Khbir, Urchade Zaratiana, Nadi Tomeh, and Thierry Charnois. 2023. Lipn at wojoodner shared task: A span-based approach for flat and nested arabic named entity recognition. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*.

Abdelkader El Mahdaouy, Salima Lamsiyah, Hamza Alami, Christoph Schommer, and Ismail Berrada. 2023. UM6P & UL at wojoodner shared task: Improving multi-task learning for flat and nested arabic named entity recognition. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*.

Elkaref and Elkaref. 2023. El-kawaref at wojoodner shared task: Stagedner for arabic named entity recognition. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*.

Shereen Elkordi, Noha Adly, and Marwan Torki. 2023. Alexu-aic at wojoodner shared task: Sequence labeling vs mrc and swa for arabic named entity recognition. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*.

Hao Fei, Yafeng Ren, and Donghong Ji. 2020. Dispatched attention with multi-task learning for nested mention recognition. *Information Sciences*, 513:241–251.

Mourad Gridach. 2018. Deep learning approach for arabic named entity recognition. In *Computational Linguistics and Intelligent Text Processing: 17th International Conference, CICLing 2016, Konya, Turkey, April 3–9, 2016, Revised Selected Papers, Part I 17*, pages 439–451. Springer.

Ramanathan Guha, Rob McCool, and Eric Miller. 2003. Semantic search. In *Proceedings of the 12th international conference on World Wide Web*, pages 700–709.

Karim El Haff, Mustafa Jarrar, Tymaa Hammouda, and Fadi Zaraket. 2022. Curras + baladi: Towards a levantine corpus. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2022)*, Marseille, France.

Mariam Hussein, Sarah Khaled, Marwan Torki, and Nagwa Elmakky. 2023. Alex-u 2023 nlp at wojoodner shared task: Arabinder (bi-encoder for arabic named entity recognition). In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*.

Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2021. The interplay of variant, size, and task type in arabic pre-trained language models. *arXiv preprint arXiv:2103.06678*.

Amin Jaber and Fadi A Zaraket. 2017. Morphology-based entity and relational entity extraction framework for arabic. *arXiv preprint arXiv:1709.05700*.

P. James. 1991. *Knowledge graphs*. Number 945 in Memorandum Faculty of Applied Mathematics. University of Twente, Faculty of Applied Mathematics.

Mustafa Jarrar, Anton Deik, and Bilal Faraj. 2011. Ontology-based data and process governance framework -the case of e-government interoperability in palestine. In *Proceedings of the IFIP International Symposium on Data-Driven Process Discovery and Analysis (SIMPDA'11)*, pages 83–98.

Mustafa Jarrar, Mohammed Khalilia, and Sana Ghanem. 2022. Wojood: Nested arabic named entity corpus and recognition using bert. Marseille, France.

Mustafa Jarrar, Sanad Malaysha, Tymaa Hammouda, and Mohammad Khalilia. 2023a. Salma: Arabic sense-annotated corpus and wsd benchmarks. In *Proceedings of the 1st Arabic Natural Language Processing Conference (ArabicNLP), Part of the EMNLP 2023*. ACL.

Mustafa Jarrar, Fadi Zaraket, Tymaa Hammouda, Daanish Masood Alavi, and Martin Waahlisch. 2023b. Lisan: Yemeni, irqi, libyan, and sudanese arabic dialect copora with morphological annotations. In *The 20th IEEE/ACS International Conference on Computer Systems and Applications (AICCSA)*. IEEE.

Muhammad Khalifa and Khaled Shaalan. 2019. Character convolutions for arabic named entity recognition with long short-term memory networks. *Computer Speech & Language*, 58:335–346.

Imen Laouirine, Haroun Elleuch, and Fethi Bougares. 2023. Elyadata at wojoodner shared task: Data and model-centric approaches for arabic flat and nested ner. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Jiyong Li, Dilshod Azizov, Hilal AlQuabeh, and Shangsong Liang. 2023. Lotus at wojoodner shared task: Multilingual transformers: Unveiling flat and nested entity recognition. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*.

Haneen Liqreina, Mustafa Jarrar, Mohammed Khalilia, Ahmed Oumar El-Shangiti, and Muhammad Abdul-Mageed. 2023. Arabic fine-grained entity recognition. In *Proceedings of the 1st Arabic Natural Language Processing Conference (ArabicNLP), Part of the EMNLP 2023*. ACL.

Shervin Malmasi, Anjie Fang, Besnik Fetahu, Sudipta Kar, and Oleg Rokhlenko. 2022. SemEval-2022 task 11: Multilingual complex named entity recognition (MultiCoNER). In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1412–1437, Seattle, United States. Association for Computational Linguistics.

Amal Nayouf, Mustafa Jarrar, Fadi zaraket, Tymaa Hammouda, and Mohamad-Bassam Kurdy. 2023. Nâbra: Syrian arabic dialects with morphological annotations. In *Proceedings of the 1st Arabic Natural Language Processing Conference (ArabicNLP), Part of the EMNLP 2023*. ACL.

Hiroki Ouchi, Jun Suzuki, Sosuke Kobayashi, Sho Yokoi, Tatsuki Kuribayashi, Ryuto Konno, and Kentaro Inui. 2020. Instance-based learning of span representations: A case study through named entity recognition. *arXiv preprint arXiv:2004.14514*.

Adrien Pavao, Isabelle Guyon, Anne-Catherine Letournel, Dinh-Tuan Tran, Xavier Baro, Hugo Jair Escalante, Sergio Escalera, Tyler Thomas, and Zhen Xu. 2023. Codalab competitions: An open source platform to organize scientific challenges. *Journal of Machine Learning Research*, 24(198):1–6.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Ramzi Esmail Salah and Lailatul Qadri Binti Zakaria. 2018. Building the classical arabic named entity recognition corpus (canercorpus). In *2018 Fourth International Conference on Information Retrieval and Knowledge Management (CAMP)*, pages 1–8. IEEE.

Erik F Sang and Jorn Veenstra. 1999. Representing text chunks. *arXiv preprint cs/9907006*.

Burr Settles. 2004. Biomedical named entity recognition using conditional random fields and rich feature sets. In *Proceedings of the international joint workshop on natural language processing in biomedicine and its applications (NLPBA/BioNLP)*, pages 107–110.

Khaled Shaalan and Hafsa Raza. 2007. Person name entity recognition for arabic. In *Proceedings of the 2007 workshop on computational approaches to semitic languages: common issues and resources*, pages 17–24.

Mohamed Shaheen and Ahmed Magdy Ezzeldin. 2014. Arabic question answering: systems, resources, tools, and future trends. *Arabian Journal for Science and Engineering*, 39:4541–4564.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. Diffusionner: Boundary diffusion for named entity recognition. *arXiv preprint arXiv:2305.13298*.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2005. Ace 2005 multilingual training corpus-linguistic data consortium. *URL: https://catalog. ldc. upenn. edu/LDC2006T06*.

Yu Wang, Hanghang Tong, Ziye Zhu, and Yun Li. 2022. Nested named entity recognition: a survey. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 16(6):1–29.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. 2013. Ontonotes release 5.0 ldc2013t19. *Linguistic Data Consortium, Philadelphia, PA*, 23:170.

Ayah Zirikly and Mona Diab. 2014. Named entity recognition system for dialectal Arabic. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 78–86, Doha, Qatar. Association for Computational Linguistics.

ELYADATA at WojoodNER Shared Task:
Data and Model-centric Approaches for Arabic Flat and Nested NER

Imen Laouirine† and **Haroun Elleuch**† and **Fethi Bougares**
firstname.lastname@elyadata.com

Abstract

This paper describes our submissions to the WojoodNER shared task organized during the first ArabicNLP conference. We participated in the two proposed sub-tasks of flat and nested Named Entity Recognition (NER). Our systems were ranked first over eight and third over eleven in the Nested NER and Flat NER, respectively. All our primary submissions are based on DiffusionNER models, where the NER task is formulated as a boundary-denoising diffusion process. Experiments on nested WojoodNER achieves the best results with a micro F1-score of **93.73%**. For the flat sub-task, our primary system was the third-best system, with a micro F1-score of **91.92%**[1].

Keywords: nested NER, flat NER, DiffusionNER, PIQN, data re-sampling.

1 Introduction

Named Entity Recognition is the task of locating a word or a phrase that references a particular entity within a given text. It is among the most prominent challenges in Natural Language Processing (NLP). NER has been an active research area with growing interest and significant development over the past twenty years. This increased interest has led to the organization of multiple NER shared tasks and evaluation campaigns, especially for English (Tjong Kim Sang, 2002) and some other languages (Tsygankova et al., 2019) (Benikova et al., 2014) (Nguyen et al., 2019).

As for the used techniques and methods applied to NER, there are mainly four approaches according to Jehangir et al. (2023): rule-based algorithms, supervised and unsupervised machine learning algorithms, and deep-learning algorithms. The rule based approach relies on predefined grammatical rules and dictionaries to identify named entities. These systems are precise, but do not generalize well. Supervised learning approaches can achieve high accuracy, yet demand labelled data and may give poor results for unseen domains. In contrast, unsupervised learning techniques retrieve named entities without requiring labelled data. However, the absence of labels makes it challenging to assess the performance of unsupervised models effectively (Jehangir et al., 2023). The advent of deep learning models like Transformers (Vaswani et al., 2017) which excel across domains and languages, enabled researches to make bigger strides towards better performance.

Multiple studies have addressed the challenges of identifying and classifying named entities in Arabic text. Notable initiatives include the work by Benajiba et al. (2007a), which introduced a NER system relying only on n-grams and maximum entropy, and it achieved an F1-score of 54.11% on ANERcorp dataset (Benajiba et al., 2007b). Additionally, Gridach (2016) has used deep learning methodologies to enhance the performance of Arabic NER with an F1-score 88.64% on the same dataset (Qu et al., 2023). Furthermore, multiple pretrained models such as AraBERT, MARBERT, and JABER were fine-tuned to achieve respectively 90.51%, 80.5% and 84.20% F1-score on ANERcorp (Qu et al., 2023).

Despite these previous efforts and progress made for Arabic NER, nested NER still constitutes a challenging task not well studied in Arabic. Nested NER refers to the particular case of NER where entities are nested within each other, possibly with different tags. Another common challenge in Arabic NER lies the specific nature of the language itself: Arabic is a morphologically rich and highly ambiguous language with numerous dialectal variants and a significant amount of code-switching (Jarrar et al., 2022; Darwish et al., 2021).

These factors, combined, make NER in its nested or flat variants, a particularly challenging task when

[0]†Equal contribution

[1]Our code is available at `https://github.com/elyadata/NER_shared_task_2023`

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 759–764
December 7, 2023 ©2023 Association for Computational Linguistics

performed on Arabic and dialectal Arabic. Hence, the motivation for the 2023 NER Shared Task (Jarrar et al., 2023) which aims to produce models that can overcome the aforementioned challenges. The main contribution of this work can be summarized as follows:

- Experimenting with re-sampling techniques for dataset unbalance alleviation.
- Fine-tuning state-of-the-art-models for both the nested and flat NER sub-tasks.
- Achieving best results for the nested NER model submission.

This paper is organized as follows. Section 2 describes the Wojood dataset. Section 3 presents the implemented flat and nested tasks, and Section 4 details the experiments and the obtained results. The overall results are discussed in Section 5 before concluding the paper in section 6.

2 Dataset

The NER Shared Task dataset "Wojood" (Jarrar et al., 2022) consists of 16817 sentences with an average sentence length of 23.45 words and a vocabulary size of 44881 for training, 3133 sentences with an average length of 17.8 and a 13134 vocabulary size for validation. The test set has 5990 sentences with an average length of 18.68 and a vocabulary size of 20920. It comprises both Modern Standard Arabic (MSA) and dialectal Arabic sentences. However, as detailed in Table 3 of Jarrar et al. (2022), the Wojood dataset is unbalanced, with the most common class being GPE (Geopolitical Entity) in the nested train set and ORG (Organization) in the flat train set, and the least frequent class being PRODUCT for Nested and UNIT for Flat.

3 NER systems

In this work, we approached the NER task with two different methods: a data-centric approach encompassing dataset re-sampling and data preprocessing and a model-centric approach using several model architectures.

3.1 Data cleaning

The data cleaning process involved several steps. Firstly, a definition of 974 stop-words was established. Then, stop words were removed from the train set. Additionally, both exclamation marks (!) and question marks (?) were removed from the text, enabling the model to focus only on sentence context. However, full-stops(.) and commas (,) were kept to avoid offsetting numerical values. Moreover, each occurrence of two or more dots were replaced with just one to maintain text clarity.

3.2 Data-centric approach

Based on the key observation of the unbalanced nature of the Wojood data-set, we decided to experiment with a data-centric approach using various re-sampling methods.

NER datasets are typically unbalanced, with an over-representation of the *Outside* <*O*> tag. To mitigate this unbalance, Wang and Wang (2022) proposed 4 different re-sampling methods in order to increase the occurrences of sentences including sequences tagged with an under-represented class. The following is an outline of these methods.

Smoothed Count (sC) re-sampling: This is the re-sampling method upon which the following are built and the most simplistic. It works by re-sampling sentences that contain the most named entities not classified as *Outside* <*O*>.

Smoothed re-sampling incorporating Count and Rareness (sCR): This method iterates on *sC* by incorporating a rarity factor. Using this technique, sentences with rarer tokens are more likely to be re-sampled. This method incorporates a rarity component in addition to the smoothed count.

Smoothed re-sampling incorporating Count, Rareness, and Density (sCRD): In order to focus on sentences that have a higher number of entity tokens per sentence length, the entity density is added in this method, while still considering the rarity of the tokens.

Normalized and Smoothed re-sampling incorporating Count, Rareness, and Density (nsCRD): This method adds a utility factor to the sCRD re-sampling function, to model the usefulness or the pertinence of a token. Thus, the more varied the tags in a sentence, the higher its marginal utility factor, and the higher its chances of re-sampling when compared to a less varied sentence.

3.3 Model-centric approach

Two models were used. A Parallel Instance Query Networks model (PIQN) (Shen et al., 2022) and a DiffusionNER model (Shen et al., 2023). PIQN extracts entities from a sentence concurrently using a parallel approach. Each individual query instance predicts a single entity. By concurrently processing all of these query instances, multiple entities can be retrieved in parallel. This model contains three main components: the encoder module, the entity prediction module that conducts entity localization and entity classification and the dynamic label assignment module that assigns the ground truth entities to the instance queries.

DiffusionNER is a state-of-the-art model in the field of NER. It is a diffusion model (Ho et al., 2020) that adds noises spans to the ground truth entity boundaries and learns to reverse this process to reconstruct correctly the entity boundaries during training. During inference, it randomly selects noisy spans from a standard Gaussian distribution, then it generates named entities by applying a denoising operation using the acquired reverse diffusion process. For each of the two mentioned models above, several BERT (Devlin et al., 2019) encoder derivatives were used, namely MARBERT (Abdul-Mageed et al., 2021), AraBERT (Antoun et al., 2020), and BioBERT (Lee et al., 2019).

4 Experiments and results

In order to reproduce the results obtained in Jarrar et al. (2022), we started by training a multitask-based baseline model for each evaluation condition (flat and nested) using AraBERT (Antoun et al., 2020). Table 1 compare our baseline results to the ones obtained in (Jarrar et al., 2022). The scores are calculated on the *development* set provided by the shared task organizers.

Model	sub-task	Micro F1-score
(Jarrar et al., 2022)	flat	86.81
Our baseline	flat	**87.55**
(Jarrar et al., 2022)	nested	90.47
Our baseline	nested	**90.53**

Table 1: Reproducing results of Jarrar et al. (2022). Results are measured on the development set related to each version of the models (flat or nested).

As stated in section 3, we have opted for two main axes of experimentation: data-centric and model-centric approaches. These are covered in the following sections.

4.1 Dataset re-sampling

To assess the benefits of data re-sampling, we started from the baseline of the flat sub-task and re-trained a new model for each re-sampling method presented above. This results in four different new models, each trained on a resampled version of the data. As is shown in table 2, compared to the baseline, an improvement of 0.37 was obtained with *sC* re-sampling whereas a slight improvement when using *sCR* method. However, a drop in F1-score has been observed with entity density-based re-sampling methods *sCRD* and *nsCRD*.

Model	Micro F1-score
baseline	87.55
baseline with sC	**87.92**
baseline with sCR	87.62
baseline with sCRD	87.37
baseline with nsCRD	87.42

Table 2: Comparing re-sampling methods on flat Wojood. All scores are obtained using the development set of flat Wojood.

In order to assess the effectiveness of the data cleaning process, we applied the pre-processing steps described in section 3.1 and re-trained the best system from the table above (*baseline with sC*). Unfortunately, data pre-processing affects the system improvement and reduces the F1-score of the *baseline with sC* model to 80.71%. Given this, we have decided to proceed without data pre-processing.

4.2 Model fine-tuning

Regarding the model-centric approaches, we trained several PIQN and DiffusionNER models using multiple BERT encoders.

We started by applying the default PIQN configuration with *biobert-large* encoder on the nested Wojood sub-task. The obtained results were worse than the baseline, which could be explained by the language and domain mismatch. In fact, BioBERT was trained using English Wikipedia, BooksCorpus and several large-scale biomedical corpora.

To remedy this mismatch situation, we replaced the BioBERT encoder by AraBERT (Antoun et al., 2020) which is trained on Arabic and hence more suitable for the Wojood shared task. As shown in table 3, we observed that AraBERT is quite effective. Compared to the baseline, an F1 score improvement of 3.07 and 2.01 are obtained for flat and nested sub-tasks respectively.

As dataset re-sampling is shown to improve the model performance (see Table 2), we tried by training the PIQN model on top of each re-sampling method. This experiment was performed using the flat Wojood data-set. As we can see in Table 3, all the implemented re-sampling method did not yield the desired results when used with PIQN models trained using AraBERT encoder. Given the results obtained with PIQN, we trained DiffusionNER model without any re-sampling. As listed in Table 3, fine-tuning DiffusionNER with AraBERT encoder module resulted in the highest obtained F1-scores of **91.50%** and **93.19%** for both flat and nested NER respectively.

As Wojood data is a mix of MSA and dialectal Arabic, we have also tried to train DiffusionNER using MARBERT (Abdul-Mageed et al., 2021) encoder. Since MARBERT was trained using a mixture of MSA and dialectal Arabic, we hypothesize that it could improve the NER results. Unfortunately, that was not the case since the usage of MARBERT resulted in lower F1 score compared to the DiffusionNER model trained with AraBERT encoder (Using MARBERT reduces the F1 score of the DiffusionNER model from 93.19 to 90.39).

Model	Flat micro F1-score	Nested micro F1-score
PIQN_AraBERT	90.59	92.54
PIQN_AraBERT_sC	88.98	–
PIQN_AraBERT_sCR	88.63	–
PIQN_AraBERT_sCRD	80.98	–
PIQN_AraBERT_nsCRD	90.04	–
DiffusionNER_AraBERT	**91.50**	**93.19**

Table 3: Results of fine-tuning PIQN and DiffusionNER on flat and nested Wojood development set. Best results are in bold.

All our models were implemented using PyTorch (Paszke et al., 2019) and trained on one Nvidia Quadro RTX 6000 GPU using Adam optimizer (Kingma and Ba, 2017) for a number of epochs ranging between 100 and 150 with a batch size ranging from 8 to 32.

4.3 System submissions

For both flat and nested sub-tasks, we submitted the top two performing systems on the development set.

Table 4 shows the scores obtained on the official test set. Our primary submission for nested NER was ranked first among all participants with an F1-score of **93.73%**. As for our flat NER primary submission, it was ranked third with an F1-score of 91.92%.

Model	sub-task	micro F1-score
DiffusionNER_AraBERT	nested	**93.73**
PIQN_AraBERT	nested	91.86
DiffusionNER_AraBERT	flat	**91.92**
PIQN_AraBERT	flat	90.87

Table 4: Official evaluation results. DiffusionNER and PIQN are respectively the primary and auxiliary submissions.

5 Discussion

We started our experiments by re-sampling the data set in order to alleviate the data imbalance problem. Our experiments show that not all the tested re-sampling improves the F1 score of the Wojood data-set. We also tested model-centric methods using two encoder-only models, namely PIQN and DiffusionNER. Both models have been tested using various pretrained encoders: BioBERT, AraBERT and MARBERT.

Using BioBERT has decreased the model performance. This is not surprising, given the fact that BioBERT is mainly trained on biomedical English text, which does not fit our use-case.

We also noted that the results of PIQN and DiffusionNER are better without dataset re-sampling. We observed this when we trained PIQN and DiffusionNER models after re-sampling the flat NER dataset. However, due to time constraints, we didn't run the same experiments for nested NER.

Another observation made during the experiments is that the use of an AraBERT encoder yields better results compared to MARBERT despite the presence of dialectal sentences in the latter. This can be attributed to the pretraining of MARBERT being exclusively on tweets, or to the possibility that the Wojood dataset has more MSA content than dialectal.

6 Conclusion

This paper presents results obtained on two NER sub-tasks of the Wojood shared task, namely flat and nested NER. Our submission relies on the usage of data-centric and model-centric approaches. Data-centric consists of a set of re-sampling methods intended to mitigate the unbalanced nature of Wojood data-set. Various re-sampling method are implemented and led to only limited success. Model-centric approaches, for their part, are designed to train the best model for a given dataset. We experimented with PIQN and DiffusionNER models trained using various pre-trained encoder. Remarkably, the DiffusionNER fine-tuned with an AraBERT sentence encoder module without any re-sampling or pre-processing, yielded to significant improvements over the baseline results for both nested and flat NER. This allows us to be ranked first out of eight for the nested NER sub-task with a **93.93%** F1-score and third out of eleven for the flat NER sub-task with a **91.92%** F1-score.

Limitations

Despite the high ranking of our submitted system, there are still a number of limitations that should be mentioned and addressed in the future. Those can be summarized into the following categories:

(1) Complexity: These systems are comprised of multiple tightly coupled modules and components, which is relatively complex when compared to the baseline model. Moreover, this complexity results in higher hardware requirements in terms of GPU memory and computing power.

(2) Maximum sequence length: Both models are not able to train or perform inference on sequences longer than 512 characters tokens. A workaround consisting of splitting longer sentences into two smaller ones has been adopted for this work, but that is not an elegant solution for long-term or industrial usage.

(3) Annotated data requirements: Both systems require annotated data, which can be more costly and harder to source. Self-supervised, unsupervised or few-shot learning alternatives like the work of Das et al. (2022) should be explored in order to mitigate the need for high amounts of labelled data.

Ethics Statement

All models used for this work are open-source and publicly available. All results are reproducible, given the Wojood dataset. The authors obtained the dataset by submitting a formal request.

References

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2021. ARBERT & MARBERT: Deep bidirectional transformers for Arabic. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7088–7105, Online. Association for Computational Linguistics.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. AraBERT: Transformer-based model for Arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France. European Language Resource Association.

Yassine Benajiba, Paolo Rosso, and José Miguel Benedíruiz. 2007a. Anersys: An arabic named entity recognition system based on maximum entropy. In *Computational Linguistics and Intelligent Text Processing: 8th International Conference, CICLing 2007, Mexico City, Mexico, February 18-24, 2007. Proceedings 8*, pages 143–153. Springer.

Yassine Benajiba, Paolo Rosso, and José Miguel BenedíRuiz. 2007b. Anersys: An arabic named entity recognition system based on maximum entropy. In *Computational Linguistics and Intelligent Text Processing*, pages 143–153, Berlin, Heidelberg. Springer Berlin Heidelberg.

Darina Benikova, Chris Biemann, Max Kisselew, and Sebastian Padó. 2014. Germeval 2014 named entity recognition shared task.

Kareem Darwish, Nizar Habash, Mourad Abbas, Hend Al-Khalifa, Huseein T. Al-Natsheh, Houda Bouamor, Karim Bouzoubaa, Violetta Cavalli-Sforza, Samhaa R. El-Beltagy, Wassim El-Hajj, Mustafa Jarrar, and Hamdy Mubarak. 2021. A panoramic survey of natural language processing in the arab world. *Commun. ACM*, 64(4):72–81.

Sarkar Snigdha Sarathi Das, Arzoo Katiyar, Rebecca Passonneau, and Rui Zhang. 2022. CONTaiNER: Few-shot named entity recognition via contrastive learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6338–6353, Dublin, Ireland. 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.

Mourad Gridach. 2016. Deep learning approach for arabic named entity recognition. In *Computational Linguistics and Intelligent Text Processing: 17th International Conference, CICLing 2016, Konya, Turkey, April 3–9, 2016, Revised Selected Papers, Part I 17*, pages 439–451. Springer.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc.

Mustafa Jarrar, Muhammad Abdul-Mageed, Mohammed Khalilia, Bashar Talafha, AbdelRahim Elmadany, Nagham Hamad, and Alaa Omer. 2023. WojoodNER: The Arabic Named Entity Recognition Shared Task. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*.

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.

Basra Jehangir, Saravanan Radhakrishnan, and Rahul Agarwal. 2023. A survey on named entity recognition – datasets, tools, and methodologies. *Natural Language Processing Journal*, 3:100017.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Huyen Thi Minh Nguyen, Quyen The Ngo, Luong X Vu, Vu Mai Tran, and Hien T. Nguyen. 2019. Vlsp shared task: Named entity recognition. *Journal of Computer Science and Cybernetics*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Xiaoye Qu, Yingjie Gu, Qingrong Xia, Zechang Li, Zhefeng Wang, and Baoxing Huai. 2023. A survey on arabic named entity recognition: Past, recent advances, and future trends. *arXiv preprint arXiv:2302.03512*.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. DiffusionNER: Boundary diffusion for named entity recognition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3875–3890, Toronto, Canada. Association for Computational Linguistics.

Yongliang Shen, Xiaobin Wang, Zeqi Tan, Guangwei Xu, Pengjun Xie, Fei Huang, Weiming Lu, and Yueting Zhuang. 2022. Parallel instance query network for named entity recognition. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 947–961, Dublin, Ireland. Association for Computational Linguistics.

Erik F. Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*.

Tatiana Tsygankova, Stephen Mayhew, and Dan Roth. 2019. BSNLP2019 shared task submission: Multisource neural NER transfer. In *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*, pages 75–82, Florence, Italy. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Xiaochen Wang and Yue Wang. 2022. Sentence-level resampling for named entity recognition. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2151–2165, Seattle, United States. Association for Computational Linguistics.

Lotus at WojoodNER Shared Task: Multilingual Transformers: Unveiling Flat and Nested Entity Recognition

Jiyong Li[1], Dilshod Azizov[2], Hilal AlQuabeh[2], Shangsong Liang[1,2,*]
[1]Sun Yat-sen University
[2]Mohamed bin Zayed University of Artificial Intelligence
lijy373@mail2.sysu.edu.cn,{dilshod.azizov, hilal.alquabeh}@mbzuai.ac.ae
[*]Corresponding author, liangshangsong@gmail.com

Abstract

We introduce our systems developed for two subtasks in the shared task "WOJOOD" on Arabic NER detection, part of ARABICNLP 2023. For Subtask 1, we employ the XLM-R model to predict Flat NER labels for given tokens using a single classifier capable of categorizing all labels. For Subtask 2, we use the XLM-R encoder by building 21 individual classifiers. Each classifier corresponds to a specific label and is designed to determine the presence of its respective label. In terms of performance, our systems achieved competitive *micro-F1* scores of **0.83** for Subtask 1 and **0.76** for Subtask 2, according to the leaderboard scores.

1 Introduction

Named Entity Recognition (NER) is crucial for Natural Language Processing (NLP), enabling the extraction of entities like names and locations from texts. Given the rich linguistic diversity and varied dialects of Arabic, NER becomes especially challenging (Guellil et al., 2021).

Arabic, spoken by 420 million natives, is one of the top ten global languages (Guellil et al., 2021; Cheng et al., 2021; Qu et al., 2023). Its lack of capital letters amplifies its morphological complexity, contrasting NER ease in English due to its varied dialects and rich history.

Arabic can be broadly classified into three distinct forms (Benajiba and Rosso, 2008): Classical Arabic (CA), Modern Standard Arabic (MSA), and Dialectal Arabic (DA) (Elgibali, 2005). Although CA is the esteemed language of most religious Arabic texts, MSA, recognized as an official language by the United Nations, finds its presence in contemporary media, formal correspondences, and the academic sphere. DA, on the contrary, dominates informal day-to-day communications (Qu et al., 2023).

The increasing volume of Arabic content on digital platforms, driven by the proliferation of social media, has led to a surge in the demand for Arabic NER. Beyond its general applications, NER serves specialized domains, enabling tasks such as relation extraction (Cheng et al., 2021), entity linking (Gu et al., 2021), event extraction (Zhu et al., 2021), coreference resolution (Clark and Manning, 2016), and machine translation (Ugawa et al., 2018),

Historically, most Arabic NER research has focused on direct, flat entity recognition techniques. However, the introduction of the Wojood corpus (Jarrar et al., 2022b) marks a pivotal shift. This corpus, which forms the foundation of the ARABICNLP 2023 WOJOOD (Jarrar et al., 2023) shared task, stands out for its extensive reach, encompassing more than 550k tokens from MSA and its respective dialects. All of these are carefully annotated across a spectrum of 21 different entity types.

The shared task (Jarrar et al., 2023) highlights two principal NER challenges:

(i) Wojood-Flat: This traditional method assigns each token to a single well-defined entity type.

(ii) Wojood-Nested: A more complex approach where tokens can be linked to multiple overlapping entity labels, highlighting the intricacies of languages as depicted in Figure 1.

Two significant challenges arise in this context. First, despite progress in NLP, Nested NER (Wang et al., 2022; Jarrar et al., 2022a; Straková et al., 2019) remains relatively uncharted compared to its flat NER counterpart, which has been deeply explored through cutting-edge linguistic, statistical, and neural techniques (Li et al., 2019; Zirikly and Diab, 2015; Shaalan and Raza, 2009). Second, there exists a conspicuous lack of detailed and expansive datasets designed specifically for nested NER. Consequently, addressing the demands of Subtask 2 poses a more significant challenge compared to Subtask 1.

Our paper offers the following contributions:

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 765–770
December 7, 2023 ©2023 Association for Computational Linguistics

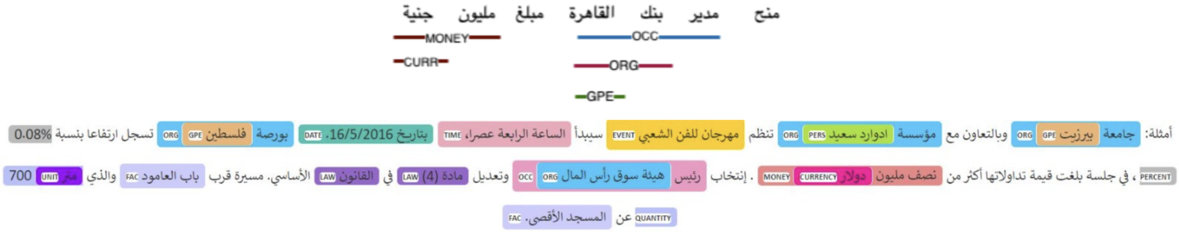

منح مدير بنك القاهرة مبلغ مليون جنية
MONEY
CURR
OCC
ORG
GPE
أمثلة: جامعة بيرزيت GPE ORG وبالتعاون مع مؤسسة ادوارد سعيد PERS ORG تنظم مهرجان للفن الشعبي EVENT سيبدأ الساعة الرابعة عصراء TIME بتاريخ 16/5/2016. DATE بورصة فلسطين GPE ORG تسجل ارتفاعا بنسبة 0.08% PERCENT ، في جلسة بلغت قيمة تداولاتها أكثر من نصف مليون دولار CURRENCY MONEY . إنتخاب رئيس هيئة سوق رأس المال ORG OCC وتعديل مادة (4) LAW في القانون LAW الأساسي. مسيرة قرب باب العامود FAC والذي 700 متر UNIT
QUANTITY عن المسجد الأقصى. FAC

Figure 1: An example of a nested NER scheme, where some tokens have more than one entity type assigned. Source: www.sina.birzeit.edu

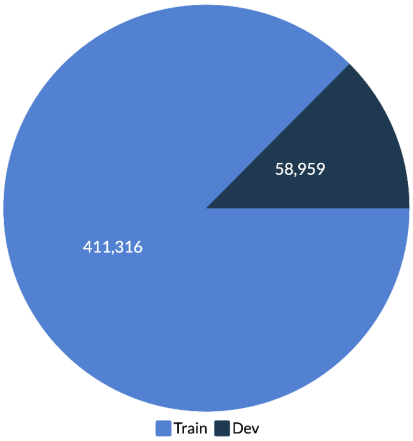


Figure 2: Statistics about tokens distribution in train and development sets.

- We introduce an automated system that uses the XLM-R (Conneau et al., 2019) architecture for both subtasks, but with distinct number of classifiers.
- We discuss and compare the performance of XLM-R, AraBERT(Antoun et al., 2020), and MARBERT (Abdul-Mageed et al., 2020) in our datasets.

The organization of this paper is as follows. In Section 2, we present prior and recent research on arabic NER. Section 3 presents a comprehensive analysis of the dataset. Section 4 describes our proposed system, experimental setup, and results. In Section 5, we conclude and point out ideas for future research.

2 Related Work

Arabic NER has seen notable advancements with various corpora, such as Ontonotes 5, which features 18 entity types from MSA (Weischedel et al., 2011), and others such as ANERCorp, CANERCorpus, and the expansive AQMAR corpus (Benajiba et al., 2007; Salah and Zakaria, 2018; Mohit et al., 2012). The Wojood corpus stands out, supporting named entities over 21 types and spanning both MSA and dialects (Jarrar et al., 2022b, 2023). Although NER methods have transitioned from rule-based to deep learning, the integration of LSTM-CRF (Lample et al., 2016) and pre-trained embeddings has been transformative. The introduction of BERT highlighted the potential of transformers in NER (Devlin et al., 2018). Nested NER challenges persist, but innovations such as multi-layer BiLSTM and pyramid architectures signal progress (Katiyar and Cardie, 2018; Ju et al., 2018; Wang et al., 2020).

3 Data

In this section, we provide a detailed description of the dataset released by the WojoodNER organizers, which comprises Arabic tokens, where less than half of the dataset consists of named entities.

The Wojood corpus encompasses a comprehensive and diverse array of flat and nested named entities, representing a new split distinct from the established Wojood paper (Jarrar et al., 2022b).

Data Attributes:

Each token in the Wojood corpus is associated with one of the predefined named entities.

- Token: single-word or sub-word unit.
- Entity types: 21 predefined entity types (e.g., location, organization, event).

Data Size:

The set Wojood corpus comprises a significant number of tokens for each named entity. In total, the dataset has around 550k tokens. Figure 2 illustrates the distribution of tokens in the train and development sets. Figure 3 shows the distribution of the named entities in the train and development sets. We observe that the majority, which constitute almost 60% of the dataset, are "Not named entities". The second most common

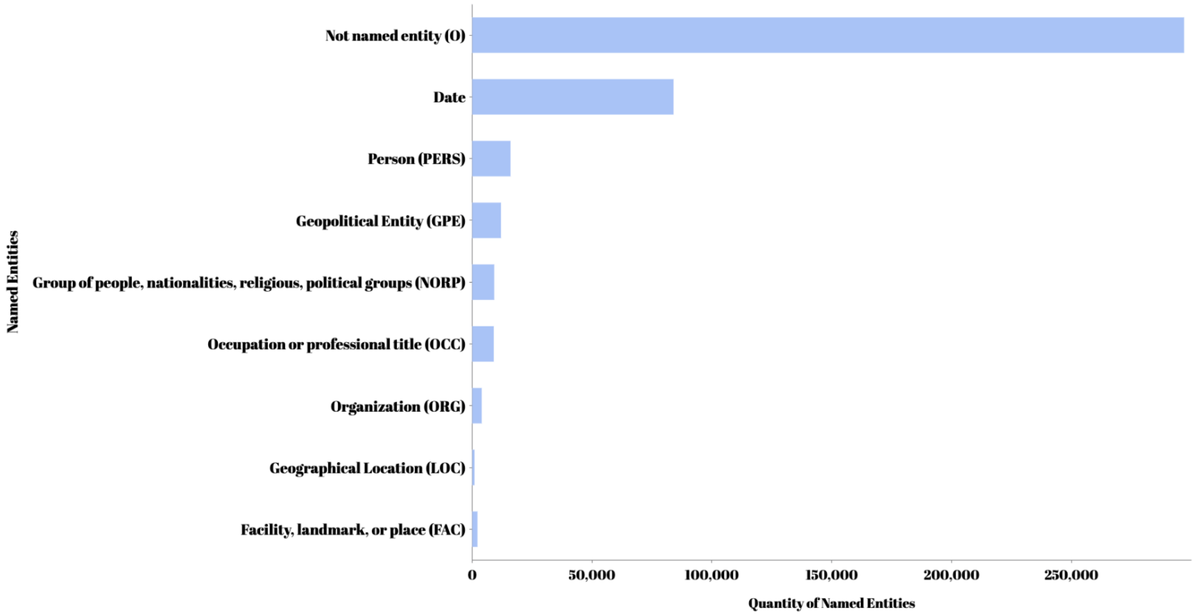


Figure 3: Namedy entities distribution over the train and development from our Wojood corpus.

entity is “Date”, followed by “Person (PERS)”, “Geographical Entity (GPE)”. The least frequent entities in the dataset are “Organization”, “Facility, landmark or place (FAC)”, and “Geographical Location (LOC)”. The rest of the entities named not shown in Figure 3, their numbers are significantly low (e.g., “Currency”).

4 Systems Description and Results

4.1 System Description

For evaluation, we use the official evaluation scorers provided for the shared task. The primary measure for both subtasks is the micro-F1 score. However, the scorers also provide data on precision and recall. Our model training was executed on 2 NVIDIA Tesla T4 (16GB) GPU.

Subtask 1. For the Subtask 1 system, we used a configuration with a sequence length of 256 and a batch size of 8. The model was trained for 5 epochs, which is the optimal duration to prevent overfitting and with a learning rate of 2e-5. Measures are captured every 500 steps, and gradients are limited to a maximum norm of 1.0. The ADAMW variant was chosen for optimization. Model checkpoints are saved every 500 steps and at the end of each epoch. We did not employ a warm-up phase, as indicated by both the warm-up ratio and the step count set to zero. To counteract overfitting, we applied a weight decay of 0.01.

Subtask 2. For Subtask 2, we trained the XLM-R model on Wojood corpus with parameters including a batch size of 16 and a learning rate of 1e-5, and the rest of the hyperparameters are similar to Subtask 1.

4.2 Results

During the initial stages, we experimented with AraBERT, MARBERT, and XLM-R with the default parameters. We experimented with the development set, since we used it as a test set, and from the train set we cut 10% out of the total tweets for the development set.

The comparative evaluation of the three frameworks, MARBERT, AraBERT, and XLM-R, on two distinct subtasks showcased varied performances. For Subtask 1, XLM-R emerged as the leading model with the highest micro-F1 score of 0.829, precision of 0.803, and recall of 0.857. AraBERT was followed with a micro-F1 of 0.713, and Precision and Recall values of 0.695 and 0.731, respectively. MARBERT, on the other hand, demonstrated a comparatively lower performance, recording a micro-F1 of 0.563. In Subtask 2, XLM-R maintained its superior performance, achieving the highest micro-F1 of 0.879 and a precision of 0.882. However, in terms of recall, MARBERT led with a score of 0.884. AraBERT showed decent performance with a micro-F1 of 0.848, a precision of 0.826, and a recall of 0.871.

In addition, in Subtask 1, data processing emerged as a critical component, dictating how

	Subtask 1			Subtask 2		
	Micro F1	**Precision**	**Recall**	**Micro F1**	**Precision**	**Recall**
MARBERT	0.663	0.675	0.610	0.870	0.857	**0.884**
AraBERT	0.713	0.695	0.731	0.848	0.826	0.871
XLM-R	**0.829**	**0.803**	**0.857**	**0.879**	**0.882**	0.877

Table 1: Experimental results of MARBERT, AraBERT, and XLM-R on the development sets for *Subtask 1* and *Subtask 2*.

well the subsequent stages would proceed. Meanwhile, in Subtask 2, the structure of the classifier piqued interest. Specifically, a simplistic approach to the nested NER – treating it as a standard classification problem and differentiating "I-XX" and "B-XX" as separate labels – would likely lead to suboptimal results.

Subtask 1. For our Named Entity Recognition (NER) task, we employed a careful preprocessing approach on our Wojood corpus using the AraBERT preprocessor and tokenizer. A key component in this process is ensuring accurate label alignment. To handle the challenge posed by tokenization splitting words into fragments, we introduced a strategy: any token resulting from either padding or representing a fragment of a word is assigned a label of -100. For instance, the word "responding" would be tokenized into two parts: "respond" and "-ing". In this case, "-ing" would be assigned the label -100.

Subsequently, our NER task was framed as a multi-class classification problem. It is important to note that we treat "I-XX" and "B-XX" as separate labels. We used the XLM-R model equipped with a single classifier capable of categorizing all labels. Based on the leaderboard scores, our system achieved a competitive micro-F1 score of 0.83.

Subtask 2. For this subtask, we use official scripts for processing, resulting in slight procedural variations compared to Subtask 1. Similar data processing methods were employed; however, in this case, padding tokens were assigned tags corresponding to the "O" label index. We conceptualized this nested NER task as a two-tier classification. After initial input processing, the system generates 21 different classifiers, each specifically related to a unique label, such as "CARDINAL", etc.

Each of these classifiers has the role of categorizing input tokens into one of three categories: "I-", "B-", or "O". To elucidate with an example: should an input be classified as "B-" by the "CARDINAL" classifier, it would translate into a prediction "B-CARDINAL". The performance measures on the leaderboard indicate that our system achieved a micro-F1 score of 0.76.

5 Conclusion and Future Work

In this paper, we detail our XLM-R based systems for two subtasks in the ARABICNLP 2023 Wojood NER shared task. Subtask 1 utilized a single classifier, while Subtask 2 developed 21 label-specific classifiers. Our models achieved micro-F1 scores of **0.83** and **0.76** for Subtasks 1 and 2, respectively, according to official leaderboard scores. We also compared our systems with state-of-the-art AraBERT and MARBERT models.

In future work, we plan to incorporate data augmentation methods, including sentence mixing and back-translation. Additionally, we would adopt a Meta-Ensembling approach, integrating models such as AraBERT, MARBERT, XLM-R, and ARBERT, to enhance performance on the unique and diverse Wojood corpus.

References

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

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

Yassine Benajiba and Paolo Rosso. 2008. Arabic named entity recognition using conditional random fields. In *Proc. of Workshop on HLT & NLP within the Arabic World, LREC*, volume 8, pages 143–153.

Yassine Benajiba, Paolo Rosso, and José Miguel Benedíruiz. 2007. Anersys: An arabic named entity recognition system based on maximum entropy. In *Computational Linguistics and Intelligent Text Processing: 8th International Conference, CICLing 2007, Mexico City, Mexico, February 18-24, 2007. Proceedings 8*, pages 143–153. Springer.

Qiao Cheng, Juntao Liu, Xiaoye Qu, Jin Zhao, Jiaqing Liang, Zhefeng Wang, Baoxing Huai, Nicholas Jing Yuan, and Yanghua Xiao. 2021. Hacred: A large-scale relation extraction dataset toward hard cases in practical applications. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2819–2831.

Kevin Clark and Christopher D Manning. 2016. Improving coreference resolution by learning entity-level distributed representations. *arXiv preprint arXiv:1606.01323*.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Alaa Elgibali. 2005. *Investigating Arabic: Current parameters in analysis and learning*, volume 42. Brill.

Yingjie Gu, Xiaoye Qu, Zhefeng Wang, Baoxing Huai, Nicholas Jing Yuan, and Xiaolin Gui. 2021. Read, retrospect, select: An mrc framework to short text entity linking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12920–12928.

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

Mustafa Jarrar, Muhammad Abdul-Mageed, Mohammed Khalilia, Bashar Talafha, AbdelRahim Elmadany, Nagham Hamad, and Alaa Omer. 2023. WojoodNER: The Arabic Named Entity Recognition Shared Task. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*.

Mustafa Jarrar, Mohammed Khalilia, and Sana Ghanem. 2022a. Named entity recognition, multi-task learning, nested entities, bert, arabic ner corpus. *arXiv preprint arXiv:2205.09651*.

Mustafa Jarrar, Mohammed Khalilia, and Sana Ghanem. 2022b. Wojood: Nested arabic named entity corpus and recognition using bert. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3626–3636.

Meizhi Ju, Makoto Miwa, and Sophia Ananiadou. 2018. A neural layered model for nested named entity recognition. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1446–1459.

Arzoo Katiyar and Claire Cardie. 2018. Nested named entity recognition revisited. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 1.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2019. A unified mrc framework for named entity recognition. *arXiv preprint arXiv:1910.11476*.

Behrang Mohit, Nathan Schneider, Rishav Bhowmick, Kemal Oflazer, and Noah A Smith. 2012. Recall-oriented learning of named entities in arabic wikipedia. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 162–173.

Xiaoye Qu, Yingjie Gu, Qingrong Xia, Zechang Li, Zhefeng Wang, and Baoxing Huai. 2023. A survey on arabic named entity recognition: Past, recent advances, and future trends. *arXiv preprint arXiv:2302.03512*.

Ramzi Esmail Salah and Lailatul Qadri Binti Zakaria. 2018. Building the classical arabic named entity recognition corpus (canercorpus). In *2018 Fourth International Conference on Information Retrieval and Knowledge Management (CAMP)*, pages 1–8. IEEE.

Khaled Shaalan and Hafsa Raza. 2009. Nera: Named entity recognition for arabic. *Journal of the American Society for Information Science and Technology*, 60(8):1652–1663.

Jana Straková, Milan Straka, and Jan Hajič. 2019. Neural architectures for nested ner through linearization. *arXiv preprint arXiv:1908.06926*.

Arata Ugawa, Akihiro Tamura, Takashi Ninomiya, Hiroya Takamura, and Manabu Okumura. 2018. Neural machine translation incorporating named entity. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3240–3250.

Yu Wang, Hanghang Tong, Ziye Zhu, and Yun Li. 2022. Nested named entity recognition: a survey. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 16(6):1–29.

Yuesong Wang, Tao Guan, Zhuo Chen, Yawei Luo, Keyang Luo, and Lili Ju. 2020. Mesh-guided multi-view stereo with pyramid architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2039–2048.

Ralph Weischedel, Sameer Pradhan, Lance Ramshaw, Martha Palmer, Nianwen Xue, Mitchell Marcus, Ann Taylor, Craig Greenberg, Eduard Hovy, Robert Belvin, et al. 2011. Ontonotes release 4.0.

LDC2011T03, Philadelphia, Penn.: Linguistic Data Consortium.

Tong Zhu, Xiaoye Qu, Wenliang Chen, Zhefeng Wang, Baoxing Huai, Nicholas Jing Yuan, and Min Zhang. 2021. Efficient document-level event extraction via pseudo-trigger-aware pruned complete graph. *arXiv preprint arXiv:2112.06013*.

Ayah Zirikly and Mona Diab. 2015. Named entity recognition for arabic social media. In *Proceedings of the 1st workshop on vector space modeling for natural language processing*, pages 176–185.

AlexU-AIC at WojoodNER shared task: Sequence Labeling vs MRC and SWA for Arabic Named Entity Recognition

Shereen Elkordi
Alexandria University
es-shereen.elkordi2018@alexu.edu.eg
Applied Innovation Center
selkordi@aic.gov.eg

Noha Adly
Alexandria University
nadly@alexu.edu.eg
Applied Innovation Center
nadly@aic.gov.eg

Marwan Torki
Alexandria University
mtorki@alexu.edu.eg

Abstract

Named entity recognition (NER) is one of many challenging tasks in Arabic Natural Language Processing. It is also the base of many critical downstream tasks to help understand the source of major trends and public opinions. In this paper, we will describe our submission in the Wojood NER Shared Task of Arabic-NLP 2023. We used a simple machine reading comprehension-based technique in the Flat NER Subtask ranking eighth on the leaderboard with a 91.13% F1-score. For the Nested NER Subtask, we fine-tuned a pre-trained language model and got a 92.61% F1 score ranking third on the leaderboard.

1 Introduction

Arabic internet content has witnessed a leap in the past years which encourages the community to explore a large spectrum of tasks. Named Entity Recognition (NER) is one of the fundamental tasks that can be included in many applications. It uses semantic text features to identify names, organizations, locations, and many other mentions in a given text. This information can be used to identify social media trends (Li et al., 2022), summarize articles (Nan et al., 2021) or as a component in question answering (Mollá et al., 2006) and machine translation (Nowakowski et al., 2022).

Many techniques to solve the NER problem have emerged and can be classified into three categories: sequence labeling, span-based classification, and sequence-to-sequence generation. Sequence labeling mainly classifies the entity type of each word or token. This category has been investigated widely in high and low-resource languages (Yang et al., 2018; Katiyar and Cardie, 2018).

For the span-based models, They depend on generating all possible spans in the input and classifying each span (Yu et al., 2020). For the sequence-to-sequence models, a decoder is required to start generating the tag for each token (Zhu et al., 2020; Straková et al., 2019), or generate all found tags with their span indices (Yan et al., 2021). Apart from the mentioned categories, other methods have been proposed that include contrastive learning as in (Huang et al., 2022; Das et al., 2022; Zhang et al., 2022; Hussein et al., 2023).

Lots of challenges exist for the Arabic NER problem, i.e. the lack of a large well-annotated dataset or language-dependent problems (Shaalan, 2014). These issues may have restricted the exploration of all the mentioned techniques. The sequence labeling technique has been the most investigated (Qu et al., 2023). Many encoders have been deployed starting from recurrent neural networks till the transfer learning from pre-trained language models like AraBERT (Antoun et al., 2020) and its variants. Recently, there were attempts to explore the multitasking track as in (Jarrar et al., 2022).

In this paper, we are trying to explore the machine reading comprehension method (MRC) (Li et al., 2020) and compare it to the sequence labeling technique with a pre-trained language model as a baseline. MRC injects a prompt alongside the input text to help the model better exploit the features that will aid it in answering the prompt. The model is guided not just to perform sequence labeling but to understand the meaning behind it and maybe better generalize to uncommon cases.

We describe our submission, to the Wojood NER Shared Task (Jarrar et al., 2023), which covers using the pre-trained model JABER (Ghaddar et al., 2021) in a sequence labeling technique, and formulating the Arabic NER task as a machine reading comprehension task following (Li et al., 2020). Further, we followed (Izmailov et al., 2018) on averaging the best checkpoints of the Flat NER model producing our best result.

2 Data

For our experiments, we used the Wojood dataset (Jarrar et al., 2022) which contains 21 entity labels.

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 771–776
December 7, 2023 ©2023 Association for Computational Linguistics

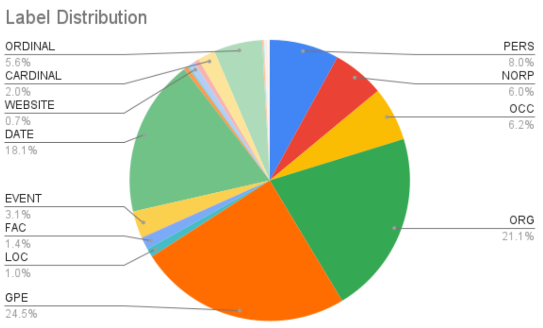


Figure 1: Label Distribution For the Nested NER Train Split

It contains three splits: train, validation, and test with 16,817, 3,133, and 5,990 sentences respectively. The train entities distribution can be found in Figure 1. To use this dataset in an MRC model, It needed some preparation. We created a new dataset where each sample includes the following fields: the context (the input text), the query (the entity type), and the start and the end positions of the answer to the queried type. These positions are indicated using the index of the start and end entity word in the context respectively.

The sizes of the dataset splits went up to 353,157, 65,793, and 125,790 since there are 21 new data samples for each sentence. We tried two different types of queries: the Arabic translation of the labels (keywords) using Google Translate [1] and the annotation guidelines of each entity as mentioned in (Jarrar et al., 2022). Examples of the dataset using the annotation guidelines queries can be shown in Table 1

For the flat NER task, we found 39,724 and 5,799 answered queries in the train and validation sets respectively. These numbers increased to 47,457 and 6,973 in the nested NER task. We can notice that the Geopolitical entity (GPE), Date, and Organization categories comprise most of the dataset with more than 11K occurrences each. In contrast, Percentage, Quantity, and Unit categories have less than 50 occurrences each.

3 System

This section will describe our two approaches: the sequence labeling technique and the problem formulation as a machine reading comprehension problem.

3.1 Sequence labeling Models

We conducted several experiments using the same codebase as (Jarrar et al., 2022). The flat NER model is composed of a pre-trained language model with a classifier layer of 43 classes following the IOB2 scheme (I-tag and B-tag for each category + O-class). The nested NER model uses 21 parallel classification layers for each category, where the output number of classes for each layer is 3 (B, I, and O). Several backbones (PLM) were explored using the sample data provided here [2]. For training the model, we used the cross entropy loss between the predicted index and the ground truth class index.

We used AraBERTv2, AraBERTv2-Twitter (Antoun et al., 2020), MARBERT, MARBERTv2 (Abdul-Mageed et al., 2021), AraElectra (Antoun et al., 2021), CamelBERT (Inoue et al., 2021) and JABER(Ghaddar et al., 2021). The best-performing encoders were JABER followed by AraBERTv2. Therefore we used JABER for training our sequence labeling models on both flat and nested tasks. JABER (Ghaddar et al., 2021) is a pre-trained language model that uses a byte-level byte pair encoding (BBPE) with data cleaning tricks, leveraging better representation of the input text.

3.2 Machine Reading Comprehension (MRC)

We decided to explore the effect of MRC by applying the method mentioned in (Li et al., 2020). It starts by creating a query for each category in the dataset. We created 21 queries for each data sample. The model's role is to extract the answer span to the query from the context (the data sample). The input to the model is the concatenation of the query and the context.

The model consists of a pre-trained encoder followed by two binary classifiers for which a token embedding is an input. The first binary classifier detects whether the provided context token represents the start of the query answer span. The second classifier predicts if the token is the end of an answer span.

There is another binary classifier whose role is to predict whether a token i and a token j from the same sentence can represent an answer span (start and end respectively). This is to match the end index with its start in case multiple start and end indices are found for the same query. This classifier

[1] https://translate.google.com/?sl=ar&tl=en&op=translate

[2] https://github.com/SinaLab/ArabicNER

Query	Start Position	End Position
Geopolitical like countries, cities, and states الجيوسياسية مثل البلدان والمدن والدول	[6, 13]	[7, 13]
Legal or social bodies like institutions, companies, agencies, teams, parties, armies, and governments. الهيئات القانونية أو الاجتماعية مثل المؤسسات والشركات والوكالات والفرق والأحزاب والجيوش والحكومات	[1]	[4]

Table 1: Example of data samples for the context: Message from the Makassed Islamic Charity Association in Jerusalem to the Acting Prime Minister in Jerusalem.
رسالة جمعية المقاصد الخيرية الإسلامية في مدينة القدس إلى رئيس الوزراء بالوكالة في القدس.
13 12 11 10 9 8 7 6 5 4 3 2 1 0

works with the two binary classifiers to filter the spans and produce the answer.

The ground truth labels consist of two lists of length N and a matrix of size NxN, where N is the number of tokens. The first list indicates if the token is the start of an answer span, while the other indicates the end. The matrix entry indicates if the token i and a token j is an answer span. The model is trained using the binary cross entropy loss.

3.3 Stochastic Weighted Average (SWA)

To improve the results, we adopted the technique mentioned in (Izmailov et al., 2018). They show that averaging multiple checkpoints of the model can improve the performance. Due to the large size of the created dataset, this choice is more convenient than an ensemble. It leads to a better usage of the computational power and decreases the inference time. Hence, we averaged the weights of the best four checkpoints of the MRC model in the flat NER Subtask.

3.4 Model Evaluation and Post processing

For the flat model inference, each sentence will be queried for every tag. The answer is returned as a list of start and a list of end positions. The answers for all 21 queries are gathered so that each word is given only one tag with the IOB2 scheme. We face a challenge here where there could be words that are included in many answer spans i.e. given two or more different labels. This can be summarized in three cases:

1. A word given B-tag1 and B-tag2
2. A word given I-tag1 and B-tag2
3. A word given I-tag1 and I-tag2

We solve this problem for the flat NER by assigning priorities to labels. These priorities are based on the frequency of the label in the training set. The more the label exists in the train set, the higher priority it gets (we are counting the B-tags only). We also make sure that the label of the word matches that of its previous in the case of I-tags. In this way, the longest named entity streak is preserved and the priority selection happens mainly in case of conflicting B-tags only.

3.5 Training Details

All models were trained on a V100 GPU. For the submitted nested model we used JABER encoder in the sequence labeling technique with a batch size of 8, a learning rate of 1e-5, and a maximum sequence length of 512. The model achieves its best result at epoch 40 and is trained for 24 hours. As For MRC models in the tasks, several experiments were done while varying the learning rate between 3e-5, 3e-6, and 2e-5. We also tried using a maximum sequence length of 200 and 256.

For the submitted flat model, we used an AraBERT-based MRC model that is trained with a batch size of 10, a learning rate of 3e-5, and a 256 maximum sequence length. The model stabilizes at epoch 10 and is trained for 48 hours. Our implementation is based on the MRC official code[3].

4 Results

We started with the sequence labeling technique in both tasks. The results with JABER on the validation set are higher in both flat and nested tasks hence we used them as our first test submission.

We tried to enhance the results by employing the MRC technique. We tried the two backbones AraBERT and JABER for both tasks. In the flat NER task, the results improved, unlike the nested

[3] https://github.com/ShannonAI/mrc-for-flat-nested-ner/

task. To further improve the results we tried performing the SWA technique which gave us the best results on the flat NER task. A table of the conducted experiments and results can be shown in Table 2

	F1-Score	Precision	Recall
	Flat NER Subtask		
Seq. Lab. (AraBERT)	0.8688	0.8558	0.8822
Seq. Lab. (JABER)	0.9052	0.90	0.9106
MRC (AraBERT)	0.9065	0.9192	0.8942
MRC (JABER)	0.9086	0.9207	0.8969
MRC (AraBERT) + keywords	0.9038	0.9208	0.8875
MRC (JABER) + keywords	0.9037	0.9249	0.8836
MRC (AraBERT) + SWA	**0.9113**	0.9133	0.9092
MRC (JABER) + SWA	0.9095	0.9152	0.9039
	Nested NER Subtask		
Seq. Lab. (AraBERT)	0.8929	0.8832	0.9028
Seq. Lab. (JABER)	**0.9261**	0.921	0.9313
MRC (AraBERT)	0.9124	0.9214	0.9036
MRC (JABER)	0.9203	0.926	0.9146
MRC (AraBERT) + keywords	0.9177	0.9188	0.9167
MRC (JABER) + keywords	0.9138	0.9241	0.9039
MRC (JABER) + SWA	0.9219	0.9226	0.9212

Table 2: Results on the test set using Sequence labeling and MRC techniques Associated with SWA.

5 Discussion

By inspecting the model performance on the validation set. We found that the flat and nested models perform poorly in the quantity, website and product classes. This is due to the insufficient number of data samples as well as the inconsistency in the annotations. An example for the inconsistency: 'Vodafone Cash and Orange Cash', these are two equivalent entities but the ground truth label for 'Vodafone Cash' is Organisation while the label for 'Orange' is Product.

For the flat NER task, the two best-performing models are MRC (AraBERT) and MRC(JABER) with stochastic weighted averaging. We analyzed the output to find the cases mentioned in Section 3.4. We found 100 words with different B-tag and I-tag labels amongst them 51 words with different I-tag-only labels and 12 words with different B-tag-only labels in the AraBERT-based model. An example of the B-tag confusion is the word 'Google' where it is assigned the labels B-ORG and B-WEBSITE. The JABER-based model has 163 words with conflicting B-tag and I-tag labels, amongst them 68 with conflicting I-tags only and 38 with conflicting B-tags only.

We wanted to analyze the efficiency of our priority-based selection scheme. We compared it with choosing randomly the B-tag label amongst the conflicting ones. We conduct 5 runs, calculate the validation F1-score at each time, and average them. For the AraBERT-based model, we find the priority scheme to score 0.90642 and the random scheme to score 0.90675. For the JABER-based model, the priority scheme produces 0.90173 while the random scheme scores 0.90155.

We notice that the more confusion in the model output, the more the random scheme fails. The first model had 12 conflicting B-tag words while the second had 38. Hence, to ensure determinism and reproducibility, we decided to follow the priority scheme. As a plan, we can choose a better scheme that would keep the model confidence scores for all 21 inferences for the sentence and compare conflicting ones to choose the B-tag with the highest score.

The Flat NER results show that the effect of adding SWA to the AraBERT-based MRC model is greater than adding it to the JABER-based model. We investigated the F1 score of each class for all the checkpoints involved in SWA. For the JABER-based models, no checkpoint could have enhanced greatly the scores of the best checkpoint.

On the other hand, other checkpoints included in the AraBERT-SWA model perform better in the cardinal, GPE, money, time, and website classes which corrected the labels on 32 samples. Meanwhile, there was a slight degradation in language, law, location, occupation, product, and quantity classes which yielded the mislabeling of 9 samples. The degradation is not effective though due to the sparsity of these classes in the dataset. In total, there was an improvement in the performance over the best checkpoint.

6 Conclusion

Arabic NER has been an underexplored problem, the lack of a large dataset can be one of the reasons. In this work, we investigate the effect of applying the machine reading comprehension technique to the Arabic NER problem. We tried two different types of prompts and concluded that the label description is more beneficial than inserting keywords as queries. We compared MRC and the sequence labeling technique. We also investigated the effectiveness of applying the stochastic weighted averaging technique. We found that the results are comparable between the sequence labeling and MRC and either of them can be used in NER. Many other methods still exist and can be tackled and finetuned for Arabic usage.

7 Limitations

MRC suffers from low scalability and long inference time. For every sentence, the required number of inferences is equal to the number of categories in the dataset. Also, the created training dataset is very sparse, many queries have no answer. Future trials can include training with a balanced set of answered and unanswered queries.

Moreover, another limitation that would affect the model performance is the absence of a considerable amount of samples for some of the classes in the dataset, i.e. the Unit class. There is no occurrence of this class in the Flat validation set which makes us unable to judge the model performance.

References

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2021. ARBERT & MARBERT: Deep bidirectional transformers for Arabic. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7088–7105, Online. Association for Computational Linguistics.

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

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

Sarkar Snigdha Sarathi Das, Arzoo Katiyar, Rebecca Passonneau, and Rui Zhang. 2022. CONTaiNER: Few-shot named entity recognition via contrastive learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6338–6353, Dublin, Ireland. Association for Computational Linguistics.

Abbas Ghaddar, Yimeng Wu, Ahmad Rashid, Khalil Bibi, Mehdi Rezagholizadeh, Chao Xing, Yasheng Wang, Duan Xinyu, Zhefeng Wang, Baoxing Huai, et al. 2021. Jaber and saber: Junior and senior arabic bert. *arXiv preprint arXiv:2112.04329*.

Yucheng Huang, Kai He, Yige Wang, Xianli Zhang, Tieliang Gong, Rui Mao, and Chen Li. 2022. COPNER: Contrastive learning with prompt guiding for few-shot named entity recognition. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2515–2527, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Mariam Hussein, Sarah Khaled, Marwan Torki, and Nagwa Elmakky. 2023. Alex-u 2023 nlp at wojoodner shared task: Arabinder (bi-encoder for arabic named entity recognition). In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*.

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.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. 2018. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*.

Mustafa Jarrar, Muhammad Abdul-Mageed, Mohammed Khalilia, Bashar Talafha, AbdelRahim Elmadany, Nagham Hamad, and Alaa Omer. 2023. WojoodNER 2023: The First Arabic Named Entity Recognition Shared Task. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*.

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.

Arzoo Katiyar and Claire Cardie. 2018. Nested named entity recognition revisited. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 861–871, New Orleans, Louisiana. Association for Computational Linguistics.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2022. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 34(1):50–70.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020. A unified MRC framework for named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5849–5859, Online. Association for Computational Linguistics.

Diego Mollá, Menno Van Zaanen, and Daniel Smith. 2006. Named entity recognition for question answering. In *Proceedings of the Australasian language technology workshop 2006*, pages 51–58.

Feng Nan, Ramesh Nallapati, Zhiguo Wang, Cicero Nogueira dos Santos, Henghui Zhu, Dejiao Zhang, Kathleen McKeown, and Bing Xiang. 2021. Entity-level factual consistency of abstractive text summarization. *arXiv preprint arXiv:2102.09130*.

Artur Nowakowski, Gabriela Pałka, Kamil Guttmann, and Mikołaj Pokrywka. 2022. Adam mickiewicz university at wmt 2022: Ner-assisted and quality-aware neural machine translation. *arXiv preprint arXiv:2209.02962*.

Xiaoye Qu, Yingjie Gu, Qingrong Xia, Zechang Li, Zhefeng Wang, and Baoxing Huai. 2023. A survey on arabic named entity recognition: Past, recent advances, and future trends. *arXiv preprint arXiv:2302.03512*.

Khaled Shaalan. 2014. A survey of arabic named entity recognition and classification. *Computational Linguistics*, 40(2):469–510.

Jana Straková, Milan Straka, and Jan Hajic. 2019. Neural architectures for nested NER through linearization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5326–5331, Florence, Italy. Association for Computational Linguistics.

Hang Yan, Tao Gui, Junqi Dai, Qipeng Guo, Zheng Zhang, and Xipeng Qiu. 2021. A unified generative framework for various NER subtasks. 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 5808–5822, Online. Association for Computational Linguistics.

Jie Yang, Shuailong Liang, and Yue Zhang. 2018. Design challenges and misconceptions in neural sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3879–3889, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Juntao Yu, Bernd Bohnet, and Massimo Poesio. 2020. Named entity recognition as dependency parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6470–6476, Online. Association for Computational Linguistics.

Sheng Zhang, Hao Cheng, Jianfeng Gao, and Hoifung Poon. 2022. Optimizing bi-encoder for named entity recognition via contrastive learning. *arXiv preprint arXiv:2208.14565*.

Huiming Zhu, Chunhui He, Yang Fang, and Weidong Xiao. 2020. Fine grained named entity recognition via seq2seq framework. *IEEE Access*, 8:53953–53961.

UM6P & UL at WojoodNER shared task: Improving Multi-Task Learning for Flat and Nested Arabic Named Entity Recognition

Abdelkader El Mahdaouy[1], **Salima Lamsiyah**[2], **Hamza Alami**[1]
Christoph Schommer[2] and **Ismail Berrada**[1]
[1]College of Computing, Mohammed VI Polytechnic University, Morocco
[2]Dept. of Computer Science, Faculty of Science, Technology and Medicine,
University of Luxembourg, Luxembourg
{firstname.lastname}@{um6p.ma[1], uni.lu[2]}

Abstract

In this paper, we present our submitted system for the WojoodNER Shared Task, addressing both flat and nested Arabic Named Entity Recognition (NER). Our system is based on a BERT-based multi-task learning model that leverages the existing Arabic Pretrained Language Models (PLMs) to encode the input sentences. To enhance the performance of our model, we have employed a multi-task loss variance penalty and combined several training objectives, including the Cross-Entropy loss, the Dice loss, the Tversky loss, and the Focal loss. Besides, we have studied the performance of three existing Arabic PLMs for sentence encoding. On the official test set, our system has obtained a micro-F1 score of 0.9113 and 0.9303 for Flat (Sub-Task 1) and Nested (Sub-Task 2) NER, respectively. It has been ranked in the 6th and the 2nd positions among all participating systems in Sub-Task 1 and Sub-Task 2, respectively.

1 Introduction

Named Entity Recognition (NER) is a fundamental component for many Natural Language Processing (NLP) applications, including Information Extraction, Information Retrieval, Question-Answering, and Text Summarization, among others (Yadav and Bethard, 2018; Li et al., 2022). NER is a sequence labeling task that involves identifying and assigning predefined class labels to named entity mentions (individual words or spans of words), such as names of persons, locations, organizations, and more. Based on the structure of named entities, they can be categorized as either flat or nested entities. Flat named entities consist of contiguous word spans with non-overlapping structures. In contrast, nested named entities exhibit a more complex structure where a named entity encompasses or is part of other named entities (Wang et al., 2022). Therefore, several tools, models, and datasets have been introduced to address both flat and nested NER tasks (Finkel and Manning, 2009; Katiyar and Cardie, 2018; Yadav and Bethard, 2018; Li et al., 2022; Wang et al., 2022). However, most existing research in this field has primarily focused on languages with high resources, such as English.

The Arabic language encompasses three distinct language varieties: Modern Standard Arabic (MSA), Classical Arabic, and Dialectal Arabic. The latter refers to the diverse spoken dialects of Arabic across the Arab World. Over the past two decades, significant attention has been paid to the Arabic NER task, where several models, tools, and datasets have been proposed (Shaalan, 2014; Liu et al., 2019; Qu et al., 2023). However, it is important to note that most available resources have primarily focused on the Modern Standard Arabic, including ANERCorp[1], ACE2004[2], ACE2005[3], Ontonotes5[4], and AQMAR (Benajiba et al., 2007; Mohit et al., 2012) datasets. For Dialectal Arabic NER, Darwish (2013) have introduced a dataset sourced from Twitter, covering both MSA and Dialectal Arabic. Similarly, Salah and Binti Zakaria (2018) have compiled a NER corpus from religious texts specifically for Classical Arabic.

Most of the previously mentioned datasets have been introduced for the flat NER task and are limited to a single Arabic language variety. To overcome these limitations, Jarrar et al. (2022) have presented the Wojood dataset, specifically created for both flat and nested Arabic NER tasks. This dataset has been collected from diverse sources, spanning various domains and topics. Moreover, it is considered the largest available multi-domain and multi-dialectal Arabic NER corpus.

In this paper, we introduce our participating system to WojoodNER shared task (Jarrar et al., 2023). Our system is built upon a BERT-based multi-task

[1] http://curtis.ml.cmu.edu/w/courses/index.php/ANERcorp
[2] https://catalog.ldc.upenn.edu/LDC2005T09
[3] https://catalog.ldc.upenn.edu/LDC2006T06
[4] https://catalog.ldc.upenn.edu/LDC2013T19

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 777–782
December 7, 2023 ©2023 Association for Computational Linguistics

learning model, where each entity type is associated with a multi-class classification head that predicts the IOB2 tag of a given input token. We have employed the same model for both the flat and nested NER sub-tasks. To encode input sentences, we have explored three Arabic Pretrained Language Models (PLMs): QARiB (Abdelali et al., 2021), CAMeLBERT-Mix (Inoue et al., 2021), and ARBERTv2 (Abdul-Mageed et al., 2021; Elmadany et al., 2022). Our final model is trained to minimize a multi-task variance loss penalty and loss function that combines the Cross-Entropy loss, the Dice loss (Li et al., 2020), the Tversky loss (Salehi et al., 2017), and the Focal Loss (Lin et al., 2020). Our system is evaluated using the micro-average Precision, Recall, and F1 score. It has achieved a micro-F1 score of 0.9113 and 0.9303 on the test sets of Flat (Sub-Task 1) and Nested (Sub-Task 2) NER, respectively. Our system achieved the 6th and 2nd positions, respectively, among all participating systems in Sub-Task 1 and Sub-Task 2 of the WojoodNER shared task. It is worth mentioning that the best results were obtained using the ARBERTv2 sentence encoder (Abdul-Mageed et al., 2021; Elmadany et al., 2022) for both sub-tasks.

2 Data

WojoodNER shared task organizers provide a rich and large dataset for Arabic NER (Jarrar et al., 2023). The shared task organizers propose two sub-tasks: one for flat NER (Sub-Task 1) and one for nested NER (Sub-Task 2) in Arabic. The provided dataset, namely Wojood (Jarrar et al., 2022), is collected from various sources and covers several domains and topics. It consists of approximately 550k tokens, comprising sentences from MSA and Dialectal Arabic. The authors have followed the LDC's OntoNotes 5 annotation guidelines (Weischedel et al., 2013) to label the Wojood dataset. The dataset tokens are labeled using 21 entity types. Additionally, they provided labels for both flat (Wojood-Flat) and nested (Wojood-Nested) Arabic NER. To evaluate the annotation quality of the Wojood dataset, the authors measured inter-annotator agreement using Cohen's *Kappa*. They have reported a macro Kappa of 0.98 and 0.979, with and without the 'O' entity tag, respectively. Both Wojood-Flat and Wojood-Nested have been split into 70%, 10%, and 20% for model training, development, and evaluation, respectively.

It is worth mentioning that we have trained, validated, and evaluated our models using the officially provided splits for training, validation, and development, respectively. Furthermore, we do not employ any text preprocessing or normalization technique.

3 System Overview

In this section, we present the overall architecture of our system's model and the employed training objectives.

3.1 Model Architecture

Following the work of Jarrar et al. (2022), we have employed a transformer-based multi-task learning model for both flat and nested Arabic NER tasks. Our model comprises a BERT-based Pre-trained Language Model (PLM) for the Arabic language, along with one classification head for each entity type. Specifically, each entity type has a multi-class classification head that predicts the IOB2 tag for a given input token. Each of these heads consists of a linear layer followed by a softmax activation function. Thus, the model can be effortlessly employed for both flat NER (Sub-Task 1) and nested NER (Sub-Task 2). Figure 1 illustrates the overall architecture of our model for both flat and nested Arabic NER.

For the input sentences encoding, we have leveraged the potential of three existing BERT-based Arabic PLMs, including QARiB (Abdelali et al., 2021), CAMeLBERT-Mix (Inoue et al., 2021) and ARBERTv2 (Abdul-Mageed et al., 2021; Elmadany et al., 2022). These PLMs have been pretrained on large Arabic text corpora.

As depicted in figure 1, given an input sentence of length m, the PLM's tokenizer splits it into n sub-words and append the $[CLS]$ and $[SEP]$ special tokens, representing the start and end of the input sequence, to the tokenized sentence ($[CLS], w_1, w_2, w_3, ..., w_n, [SEP]$). Then, the latter is passed to the PLM encoder which generates the contextualized word embedding $h_{[CLS]}, h_1, h_2, h_3, ..., h_n, h_{[SEP]}$ of the input sentence. Afterward, the contextualized word embeddings $\{h_i\}_{i=1}^{n}$ are fed to each classification head to predict the tag of each entity type.

3.2 Training objectives

To enhance the performance of our model, we have utilized a multi-task loss variance penalty and combined several training objectives, including the Cross-Entropy loss, the Dice loss, the Tversky loss,

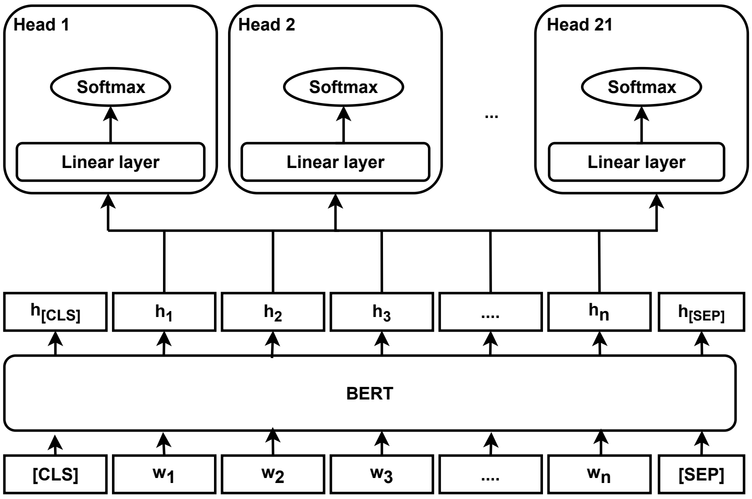


Figure 1: Overall Model Architecture

and the Focal Loss, described as follows:

- $\mathcal{L}_{CE}$ denotes the cross-entropy loss;
- $\mathcal{L}_{DI}$ denotes the dice loss. This loss is used to handle the class imbalance problem (Li et al., 2020);
- $\mathcal{L}_{TV}$ denotes the Tversky loss function. This loss is a generalization of the dice loss and allows to control the balance between false positives and false negatives using hyper-parameter α (Salehi et al., 2017);
- $\mathcal{L}_{FL}$ denotes the focal loss. This loss addresses class imbalance by down-weighting easy well-classified examples during training. It puts more emphasis on hard examples to improve overall performance (Lin et al., 2020);
- $\mathcal{L}_{VAR}$ is the multi-task loss variance penalty which consists of computing the variance of all task losses. This loss function encourages the model to minimize all task losses.

To leverage the strengths of the aforementioned loss functions, we have employed a Unified Loss function that combines them as follows:

$$\mathcal{L}_{UL} = \lambda_1 \cdot \mathcal{L}_{CE} + \lambda_2 \cdot \mathcal{L}_{DI} + \lambda_3 \cdot \mathcal{L}_{TV} + \lambda_4 \cdot \mathcal{L}_{FL} \tag{1}$$

where $\{\lambda_i\}_{i=1}^{4}$ are hyper-parameters that control the contribution of loss function. In our experiments, we have assessed the performance of the following training objectives: $\mathcal{L}_{CE}$, $\mathcal{L}_{CE} + p \cdot \mathcal{L}_{VAR}$, $\mathcal{L}_{UL}$, and $\mathcal{L}_{UL} + p \cdot \mathcal{L}_{VAR}$. Where p is a hyper-parameter that weights the multi-task loss variance penalty.

4 Experiments and Results

In this section, we present the experimental settings and discuss the obtained results.

4.1 Experiment Settings

We have implemented our model using Pytorch[5] framework and Hugging Face Transformers[6] library. Additionally, we have utilized parts of Wojood's baseline source code, namely ArabiNER[7] (Jarrar et al., 2022), for model training and evaluation. Our experiments have been performed using a Dell PowerEdge XE8545 server, having 2 AMD EPYC 7713 64-Core Processor 1.9GHz, 1TB of RAM, and 4 NVIDIA A100-SXM4-80GB GPUs.

For both Sub-Task 1 and Sub-Task 2, our models are trained using 15 epochs with a batch size of 16 examples and a learning rate of 2×10^{-5}. Moreover, weight decay is applied to all the layers of the model weights except biases and Layer Normalization (LayerNorm) and is fixed to 1×10^{-3}. Based on our preliminary experiments, we set the hyper-parameters λ_1, λ_2, λ_3, and λ_4 of the $\mathcal{L}_{UL}$ to 0.4, 0.2, 0.2, 0.2, respectively. The variance loss penalty (hyper-parameter p) is fixed at 5. The hyper-parameter α that balances the weight importance of false positives and false negatives in the Tversky loss is set to 0.5. Whereas, the hyper-parameter γ of the focal loss is fixed to 2. It is worth mentioning that we did not perform hyper-parameters tuning and we have fixed them based on our preliminary experiments.

[5] https://pytorch.org/
[6] https://github.com/huggingface/transformers
[7] https://github.com/SinaLab/ArabicNER

		Dev			Test		
Loss	**Encoder**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
$\mathcal{L}_{CE}$	QARiB	0.8571	0.8863	0.8715	0.8642	0.8882	0.876
	CAMeLBERT-Mix	0.8717	0.8934	0.8824	0.8825	0.9013	0.8918
	ARBERTv2	0.8593	0.8911	0.8749	0.8686	0.8993	0.8837
$\mathcal{L}_{CE} + p \cdot \mathcal{L}_{VAR}$	QARiB	0.8624	0.8911	0.8765	0.8617	0.8896	0.8754
	CAMeLBERT-Mix	0.8725	0.8922	0.8822	0.8838	0.8997	0.8917
	ARBERTv2	0.8578	0.8974	0.8771	0.8703	0.9039	0.8868
$\mathcal{L}_{UL}$	QARiB	0.8771	0.9013	0.889	0.88	0.9005	0.8901
	CAMeLBERT-Mix	0.8869	0.9056	0.8961	0.8988	0.9079	0.9033
	ARBERTv2	0.8963	0.91	0.9031	0.9057	0.9133	0.9095
$\mathcal{L}_{UL} + p \cdot \mathcal{L}_{VAR}$	QARiB	0.8749	0.8988	0.8866	0.8832	0.9008	0.8919
	CAMeLBERT-Mix	0.8886	0.9055	0.8969	0.8963	0.9087	0.9025
	ARBERTv2	**0.8984**	**0.9125**	**0.9054**	**0.907**	**0.9157**	**0.9113**

Table 1: The obtained results of our system on Sub-Task 1 (Wojood-Flat). Our official submission results are highlighted in bold font.

		Dev			Test		
Loss	**Encoder**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
$\mathcal{L}_{CE}$	QARiB	0.8797	0.9161	0.8976	0.8836	0.9156	0.8993
	CAMeLBERT-Mix	0.8862	0.9082	0.8971	0.897	0.9089	0.9029
	ARBERTv2	0.8982	0.928	0.9129	0.9063	0.9311	0.9185
$\mathcal{L}_{CE} + p \cdot \mathcal{L}_{VAR}$	QARiB	0.8773	0.9169	0.8967	0.8749	0.9143	0.8942
	CAMeLBERT-Mix	0.8979	0.9111	0.9044	0.9071	0.9176	0.9123
	ARBERTv2	0.903	0.9245	0.9136	0.9116	0.9285	0.92
$\mathcal{L}_{UL}$	QARiB	0.8931	0.9221	0.9074	0.904	0.9254	0.9146
	CAMeLBERT-Mix	0.9077	0.9253	0.9164	0.9162	0.9267	0.9214
	ARBERTv2	0.9181	0.9309	0.9245	0.9238	0.9336	0.9287
$\mathcal{L}_{UL} + p \cdot \mathcal{L}_{VAR}$	QARiB	0.8951	0.9214	0.9081	0.9	0.9235	0.9116
	CAMeLBERT-Mix	0.9106	0.9273	0.9189	0.9161	0.9295	0.9228
	ARBERTv2	**0.9172**	**0.933**	**0.925**	**0.9246**	**0.9361**	**0.9303**

Table 2: The obtained results of our system on Sub-Task 2 (Wojood-Nested). Our official submission results are highlighted in bold font.

We have trained, validated, and evaluated our models on the officially provided splits for training, validation, and development, respectively. For evaluation purposes, we have followed the shared task guidelines and utilized the micro average Precision, Recall, and F1 score.

4.2 Results

In this section, we present the obtained results of our model for Wojood Ner Sub-tasks.

4.2.1 Sub-Task 1

Table 1 summarizes the obtained results of our system for the flat NER subtask. Our official submission results are highlighted in bold font.

For the cross-entropy loss ($\mathcal{L}_{CE}$), the best results are obtained using the CAMeLBERT-Mix encoder. Besides, the combination of the multi-task variance loss and the cross-entropy loss ($\mathcal{L}_{CE} + p \cdot \mathcal{L}_{VAR}$) have slightly improved the Recall and the F1 score when ARBERTv2 encoder is used to encode the input sentences. Nevertheless, for the unified loss, the best performances are achieved by employing the ARBERTv2 encoder.

In accordance with the results of the cross-entropy loss, the combination of the unified loss and the multi-task variance loss ($\mathcal{L}_{UL} + p \cdot \mathcal{L}_{VAR}$) have enhanced the performance of our model when ARBERTv2 encoder is utilized. The overall obtained results show that using the unified loss leads to far better performance than the cross-entropy loss. Finally, the best performance is achieved by the combination of unified loss and the multi-task variance loss ($\mathcal{L}_{UL} + p \cdot \mathcal{L}_{VAR}$), and ARBERTv2 encoder.

Rank	Team	F1
1	LIPN	0.9196
2	El-Kawaref	0.9195
3	ELYADATA	0.9192
4	Alex-U 2023 NLP	0.918
5	tdink NER	0.9125
6	**Our team**	**0.9113**
7	AlexU-AIC	0.9113
8	ARATAL	0.9113
9	AlphaBrains	0.8751
10	Lotus	0.8339
11	Fraunhofer IAIS	0.6445

Table 3: Official leaderbord of Sub-Task 1

Table 3 shows the ranking of participating teams in the official leaderbord of Sub-Task 1. Our system is ranked at the 6th position. The top-ranked system outperformed ours by a micro-F1 score increment of 0.0083.

4.2.2 Sub-Task 2

Table 1 presents the obtained results of our system on the Nested NER subtask. Our official submission results are highlighted in bold font.

In contrast to Sub-Task 1 results, the ARBERTv2 encoder surpasses both QARiB and CAMeLBERT-Mix PLMs on all our nested experiments. The incorporation of the multi-task variance loss to the cross-entropy has slightly enhanced the performance of our model when QARiB and ARBERTv2 encoders are utilized.

The combination of unified loss and the multi-task variance loss ($\mathcal{L}_{UL}+p\cdot\mathcal{L}_{VAR}$) have enhanced the performance of our model when ARBERTv2 and QARiB encoders are employed. In line with Sub-Task 1 overall results, the unified loss improved the performance of our system using the three encoders. Finally, the best results are obtained using the combination of the unified loss and the multi-task variance loss ($\mathcal{L}_{UL} + p \cdot \mathcal{L}_{VAR}$), and ARBERTv2 encoder.

Table 4 shows the ranking of participating teams in the official leaderbord of Sub-Task 2. Our system is ranked at the 2nd position. The top-ranked system outperformed ours by a micro-F1 score increment of 0.007.

5 Discussion

The results have shown that combining loss functions that deal with the class imbalance problem improves the results. A straightforward path of future research work is to explore other training objectives that deal with the aforementioned problem. Besides, we have evaluated three existing Arabic PLMs. Thus, investigating the other state-of-the-art Arabic PLMs might improve the results.

Rank	Team	F1
1	ELYADATA	0.9373
2	**Our team**	**0.9303**
3	AlexU-AIC	0.9261
4	LIPN	0.9245
5	tdink NER	0.914
6	Alex-U 2023 NLP	0.9001
7	AlphaBrains	0.8884
8	Lotus	0.7602

Table 4: Official leaderbord of Sub-Task 2

6 Conclusion

In this paper, we have presented our participating system to WojoodNER shared task. Our system relies on a BERT-based multi-task learning model for both flat and nested Arabic NER. For the input sentence encoding, we have assessed the performance of three Arabic PLMs: QARiB, CAMeLBERT-Mix, and ARBERTv2. Our best model is trained to minimize a multi-task variance loss penalty and loss function that linearly combines the Cross-Entropy loss, the Dice loss, the Tversky loss, and the Focal Loss. The proposed system is evaluated using the micro-average Precision, Recall, and F1 score. It has achieved a micro-F1 score of 0.9113 and 0.9303 on the test sets of Flat (Sub-Task 1) and Nested (Sub-Task 2) NER, respectively. Besides, it has been ranked 6th and 2nd out of all participating systems in Sub-Task 1 and Sub-Task 2, respectively. Besides, our best results are obtained using the ARBERTv2 sentence encoder for both sub-tasks.

References

Ahmed Abdelali, Sabit Hassan, Hamdy Mubarak, Kareem Darwish, and Younes Samih. 2021. Pre-training bert on arabic tweets: Practical considerations. *ArXiv*, abs/2102.10684.

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.

Yassine Benajiba, Paolo Rosso, and José Miguel BenedíRuiz. 2007. Anersys: An arabic named entity recognition system based on maximum entropy. In *Computational Linguistics and Intelligent Text Processing*, pages 143–153, Berlin, Heidelberg. Springer Berlin Heidelberg.

Kareem Darwish. 2013. Named entity recognition using cross-lingual resources: Arabic as an example. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1558–1567, Sofia, Bulgaria. Association for Computational Linguistics.

AbdelRahim Elmadany, El Moatez Billah Nagoudi, and Muhammad Abdul-Mageed. 2022. Orca: A challenging benchmark for arabic language understanding. *arXiv preprint arXiv:2212.10758*.

Jenny Rose Finkel and Christopher D. Manning. 2009. Nested named entity recognition. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 141–150, Singapore. Association for Computational Linguistics.

Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2021. The interplay of variant, size, and task type in Arabic pre-trained language models. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 92–104, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Mustafa Jarrar, Muhammad Abdul-Mageed, Mohammed Khalilia, Bashar Talafha, AbdelRahim Elmadany, Nagham Hamad, and Alaa Omer. 2023. WojoodNER: The Arabic Named Entity Recognition Shared Task. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*.

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.

Arzoo Katiyar and Claire Cardie. 2018. Nested named entity recognition revisited. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 861–871, New Orleans, Louisiana. Association for Computational Linguistics.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2022. A survey on deep learning for named entity recognition. *IEEE Trans. Knowl. Data Eng.*, 34(1):50–70.

Xiaoya Li, Xiaofei Sun, Yuxian Meng, Junjun Liang, Fei Wu, and Jiwei Li. 2020. Dice loss for data-imbalanced NLP tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 465–476, Online. Association for Computational Linguistics.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2020. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):318–327.

Liyuan Liu, Jingbo Shang, and Jiawei Han. 2019. Arabic named entity recognition: What works and what's next. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 60–67, Florence, Italy. Association for Computational Linguistics.

Behrang Mohit, Nathan Schneider, Rishav Bhowmick, Kemal Oflazer, and Noah A. Smith. 2012. Recall-oriented learning of named entities in Arabic Wikipedia. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 162–173, Avignon, France. Association for Computational Linguistics.

Xiaoye Qu, Yingjie Gu, Qingrong Xia, Zechang Li, Zhefeng Wang, and Baoxing Huai. 2023. A survey on arabic named entity recognition: Past, recent advances, and future trends.

Ramzi Esmail Salah and Lailatul Qadri Binti Zakaria. 2018. Building the classical arabic named entity recognition corpus (canercorpus). In *2018 Fourth International Conference on Information Retrieval and Knowledge Management (CAMP)*, pages 1–8.

Seyed Sadegh Mohseni Salehi, Deniz Erdogmus, and Ali Gholipour. 2017. Tversky loss function for image segmentation using 3d fully convolutional deep networks. In *Machine Learning in Medical Imaging*, pages 379–387, Cham. Springer International Publishing.

Khaled Shaalan. 2014. A survey of arabic named entity recognition and classification. *Comput. Linguist.*, 40(2):469–510.

Yu Wang, Hanghang Tong, Ziye Zhu, and Yun Li. 2022. Nested named entity recognition: A survey. *ACM Trans. Knowl. Discov. Data*, 16(6).

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. 2013. Ontonotes release 5.0 LDC2013t19. *Linguistic Data Consortium, Philadelphia, PA*, 23:170.

Vikas Yadav and Steven Bethard. 2018. A survey on recent advances in named entity recognition from deep learning models. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2145–2158, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

AlphaBrains at WojoodNER shared task: Arabic Named Entity Recognition by Using Character-based Context-Sensitive Word Representations

Toqeer Ehsan[1], **Amjad Ali**[2], **Ala Al-Fuqaha**[2]
[1]Department of Computer Science, University of Gujrat, Gujrat, Pakistan
[2]Information and Computing Technology (ICT) Division, College of Science and Engineering (CSE),
Hamad Bin Khalifa University, Doha, Qatar
toqeer.ehsan@uog.edu.pk, amsali@hbku.edu.qa, aalfuqaha@hbku.edu.qa

Abstract

This paper presents Arabic named entity recognition models by employing single-task and multi-task learning paradigms. The models were developed by using character-based contextualized Embeddings from Language Model (ELMo) in the input layers of the Bidirectional Long-Short Term Memory (BiLSTM) networks. The ELMo embeddings are quite capable of learning the morphology and contextual information of tokens in word sequences. The single-task learning model outperformed the multi-task learning model, achieving micro F_1-scores of 0.8751 and 0.8884, respectively, ranking 10^{th} and 7^{th} in the shared task for flat and nested NER.

1 Introduction

Named Entity Recognition (NER) is a Natural Language Processing (NLP) task which aims at identifying and extracting sub-sequences of the text associated with Named Entities (NEs). These NEs are subsequently categorized into different semantic groups, such as names, places, organizations, events and dates, etc. NER is considered a crucial preliminary task for the development of different applications, such as, information retrieval (Popovski et al., 2020), text summarization (Khademi and Fakhredanesh, 2020), machine translation (Vu et al., 2020), topic modeling and event discovery (Feng et al., 2018), word-sense disambiguation (Al-Hajj and Jarrar, 2022) and others. NER is a typical sequence labeling token classification task where each token is assigned a tag. IOB labeling is a common method employed for annotating datasets for NER.

Different machine and deep learning techniques have been used to perform NER, such as, Conditional Random Fields (CRF) (Patil et al., 2020; Bhumireddypalli et al., 2023), Support Vector Machines (SVM) (Mady et al., 2022), template-based (Cui et al., 2021), Recurrent Neural Networks (RNN) (Ahmad et al., 2020), Bidirectional LSTM (Tehseen et al., 2023), Transformer-based Models (e.g. BERT) (Jarrar et al., 2022; Agrawal et al., 2022) and others. On the other hands, the nested NER has also been performed by employing LSTM with CRF inference (Dadas and Protasiewicz, 2020), LSTM-based hierarchical layering model along with contextual word representations (Wang et al., 2020), bidirectional LSTMs with exhaustive representations (Sohrab and Miwa, 2018), BERT embeddings based LSTM-CRF (Straková et al., 2019), fine-tuning pre-trained BERT model (Jarrar et al., 2022) and others.

This paper presents model development and results of a shared task for Arabic NER (Jarrar et al., 2023). The shared task has been divided into two sub-tasks, flat NER[1] and nested NER[2]. The flat NER uses a conventional annotation scheme, however, the nested scheme provides a hierarchical annotation within the NEs. For the shared task, a different version of the Wojood dataset (Jarrar et al., 2022) has been used which has 70% data for training, 10% for development and 20% for evaluation purposes. The nested NEs are challenging to predict as multiple output layers are required to train. However, the nested annotation provides a deeper insight of overlapping NEs.

We developed two models which are based on single and multi-task learning. Both models are based on long-short term memory networks. Furthermore, transfer learning has been used to enhance the models' learning capability. The contextualized pre-trained ELMo embeddings have been incorporated with word embeddings at the input layers of the models. The ELMo embeddings significantly enhanced the results as compared to the Word2Vec and part of speech(POS) tagging. The POS tags were used as encoding vectors which were concatenated with token encoding vectors.

[1] https://codalab.lisn.upsaclay.fr/competitions/11740
[2] https://codalab.lisn.upsaclay.fr/competitions/11750

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 783–788
December 7, 2023 ©2023 Association for Computational Linguistics

Both single and multi-task learning models used *softmax* non-linearity for multi-class token classification. The details of the proposed models are discussed in the Section 3. The single task learning model performed better than the multi-task learning model and produced competitive results as compared to the baseline provided in the shared task. Rest of the paper describes the dataset, proposed single task and multi-task learning NER models, results and conclusion.

2 Data

The shared task released a version of the dataset from Jarrar et al. (2022). The training and development sets have IOB labels whereas the test set has been released without labels for evaluation purposes. Table 1 shows the label-wise distribution of NEs for training and development sets. Table 2 further presents the sentence and token distribution among all three sets.

3 System

We developed neural models by using Bidirectional Long-Short Term Memory (BiLSTM) networks. The BiLSTM model has the ability to learn context within token sequences for the token classification tasks (e.g. named entity recognition). A bidirectional model has two LSTM layers, the first layer reads the tokens in the forward direction whereas the second layer scans the tokens in the backward direction. The two way scanning is helpful to attain the contextual information within the token sequences. The input sequence of N words *x_1, x_2,..., x_n* is given as the input. Equation 1 shows the BiLSTM($x_{1:n}$,i) function which demonstrates union of the forward and backward layers.

$$BiLSTM(x_{1:n}, i) = LSTM_f(x_{1:i}) \circ LSTM_r(x_{n:i}) \quad (1)$$

The function shows the representation to a vector *i* by conditioning the previous context $x_{1:i}$ and the forthcoming sequence $x_{n:i}$. The models are based on two implementation paradigms; i) Single Task Learning (STL) and ii) Multi-Task Learning (MTL).

3.1 The Proposed Single Task Learning Model

The proposed STL-based model is comprised of word encodings, word embeddings, pre-trained word representations, BiLSTM-based hidden layers, and a single output layer. Figure 1 shows the architecture of our proposed STL model. The training and development samples have been converted to word encodings which are concatenated with embedding vectors at the input layer. The input layer contains embedding layers along with pre-trained ELMo embeddings vectors. Both embedding vectors are concatenated and fed to the hidden BiLSTM layers. The hidden layers produce contextual representations which are used to perform multi-class classification by employing *softmax* non-linearity function as shown in Equation 2.

$$o_i = Softmax(Xh_i + b) \quad (2)$$

Where o_i represents the output for ith instance, h_i shows hidden state of ith instance in the sequence along with the weights X and the bias b. The model has a single output layer to produce one label for each input token. The STL model has been trained for both flat and nested NER. The flat NEs are trained just like a standard sequence labeling task. However, for nested NER, we combined the NE labels with a delimiter to make it a single label. Section 4 presents the results of STL model for flat and nested labeling.

We experimented with three hidden BiLSTM layers. A *Dropout* layer is added after each hidden layer. The *keras* library has been used with *Tensorflow* back-end in Python-3 for the implementation of both models. The dimensions of the internal embeddings are set to 256 whereas the pre-trained ELMo embeddings have 1024 projection dimensions. Section 3.3 further describes the ELMo embeddings and transfer learning. Each hidden LSTM layer has 256 units with a dropout value of 0.2(20%). Root Mean Squared Propagation (RMSprop) optimizer has been used with a learning rate of 0.001. The loss function was the *categorical cross-entropy* for all the experiments. The sequence length has been set to have 256 tokens for each sentence. The models are trained for 15 epochs with a batch size of 128 samples. All the models have been trained using GPU servers available at the Scientific Compute Cluster (SCCKN)[3].

3.2 The Proposed Multi-task Learning Model

For the nested NER, a single entity can be annotated to have multiple layers of tags. Therefore, the multi-task learning is a suitable method. The MTL models hold a prominent position in the realm of research for conducting various NLP tasks including

[3] https://www.scc.uni-konstanz.de

IOB label	**Train set**			**Dev set**		
	$Count_{Flat}$	$Count_{Nested}$	$Total$	$Count_{Flat}$	$Count_{Nested}$	$Total$
CARDINAL	1,245	18	1,263	182	1	183
CURR	19	160	179	1	20	21
DATE	10,667	623	11,290	1,567	89	1,656
EVENT	1,863	71	1,934	253	14	267
FAC	689	191	880	85	26	111
GPE	8,133	7,167	15,300	1,132	1,031	2,163
LANGUAGE	131	1	132	15	0	15
LAW	374	0	374	44	0	44
LOC	510	109	619	63	13	76
MONEY	171	0	171	20	0	20
NORP	3,505	242	3,747	488	32	520
OCC	3,774	113	3,887	544	7	551
ORDINAL	2,805	683	3,488	410	94	504
ORG	10,731	2,444	13,175	1,566	303	1,869
PERCENT	105	0	105	13	0	13
PERS	4,496	498	4,994	650	80	730
PRODUCT	36	0	36	5	0	5
QUANTITY	44	2	46	3	0	3
TIME	286	2	288	55	0	55
UNIT	7	41	48	0	3	3
WEBSITE	434	0	434	45	0	45
Total	**50,025**	**12,365**	**62,390**	**7,141**	**1,713**	**8,854**

Table 1: Entity-wise statistics of train and development sets.

Category	**No. of Sentences**	**No. of Tokens**
Train set	16,817	394,499
Dev set	3,133	55,826
Test set	5,989	111,951
Total	**25,939**	**562,276**

Table 2: Number of sentences and tokens in train, development and test sets.

NER (Jarrar et al., 2022; Yan et al., 2023; Du et al., 2022; Fang et al., 2023). Figure 2 shows the architecture of the proposed MTL model. The proposed MTL model has 21 output layers associated with each NE label. The *softmax* non-linearity function is used for each output layer. The *softmax* function performs multi-class classification to predict an NE label or ‘O’ label. The MTL model has been trained for both flat and nested NER. The model performed better for the nested dataset because a single token may have multiple NE labels due to the nested nature of the text. We further performed MTL for flat NER by converting the flat dataset into 21 columns. The outputs from multiple output layers were then combined into a single label for each token. However, for flat NEs, it is challenging to find a single most appropriate label because the MTL model can predict multiple labels for a single token. The model setup and hyper-parameters are similar to the STL model.

3.3 Transfer Learning

Deep learning based models require larger datasets to produce state-of-the-art results. Mostly, the annotation of large datasets is not feasible. Therefore, the transfer learning is a suitable approach by training word embeddings on huge unannotated datasets. We have used ELMo embeddings which have been pre-trained on a large Arabic textual data (Che et al., 2018; Fares et al., 2017)[4]. Context-free word embeddings (Pennington et al., 2014; Mikolov et al., 2013; Bojanowski et al., 2017) provide a single word vector for each token irrespective of the context. However, contextual ELMo word embeddings (Peters et al., 2018) generate the vectors with respect to the character-based contextual information in a sentence. The ELMo model contains three neural network layers. First character-based convo-

[4] https://github.com/HIT-SCIR/ELMoForManyLangs

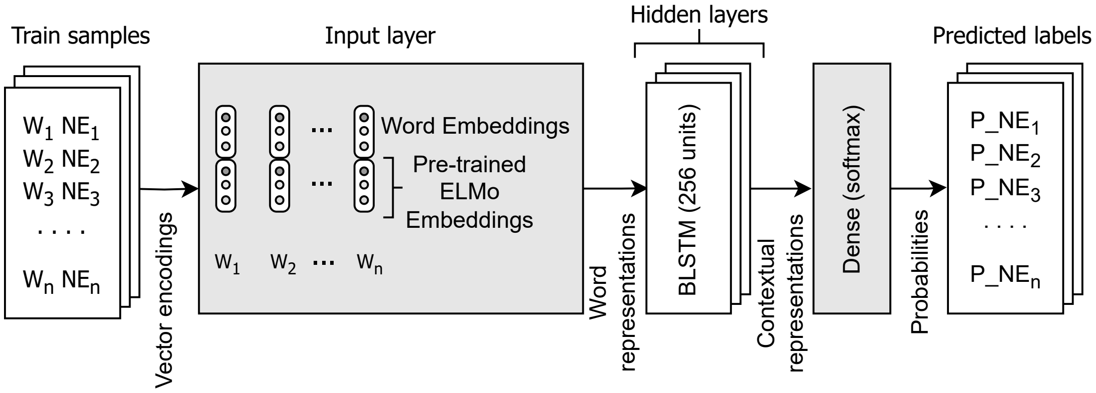


Figure 1: Architecture of the single task learning-based model.

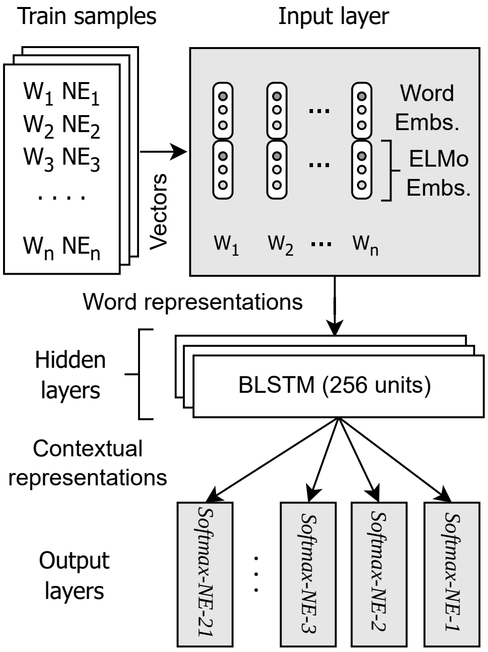


Figure 2: Architecture of multi-task learning-based model.

lutional layer, the second and the third layers are bidirectional LSTM networks to learn the contextual representations. Due to the character convolutions, the ELMo embeddings are quite capable to produce vectors for Out-Of-Vocabulary words. Both STL and MTL models have been trained by incorporating the ELMo vectors achieved from the third layer of the model showing significant improvements in the NER results.

4 Results and Discussion

The micro-F_1 score has been computed for the evaluation of the models by using *seqeval* Python package[5]. The results for the flat and nested NER on the 20% test set are shown in Tables 3 and 4.

Models	**Pre.**	**Rec.**	$\mathbf{F_1}$
Baseline	–	–	0.8681
Our STL model	0.8745	0.8758	0.8751
Our MTL model	0.8647	0.8806	0.8726

Table 3: Flat NER results (micro F_1-score).

Table 3 shows the comparison of the proposed single and multi-task learning models with the baseline score for flat NER. Our proposed STL model performed better than the MTL and the baseline. However, there is a subtle difference in our models due to the nested nature of the dataset as a single token can have multiple IOB labels. The MTL model may produce multiple labels for flat NER against a single token therefore, for the selection of a single label, a naive approach has been used which selects the left-most label among multiple NE labels.

Models	**Pre.**	**Rec.**	$\mathbf{F_1}$
Baseline	–	–	0.9047
Jarrar et al. (2022)	0.8772	0.8909	0.8840
Our STL model	0.8845	0.8923	0.8884
Our MTL model	0.8900	0.8793	0.8846

Table 4: Nested NER results (micro F_1-score).

Table 4 shows the results for nested NER from the proposed STL and MTL models and compares with the baseline and the F_1-score from Jarrar et al. (2022). While our results fall short of the baseline model, which is a transformer-based model, they outperform Jarrar et al. (2022). The STL model performs better than the MTL model for the nested

[5] https://pypi.org/project/seqeval

NER. For the nested NER to be trained on the STL model, we combined the labels by using a delimiter (∼) and trained the dataset like flat labels. This label combination resulted in a total of 298 distinct labels. Beside the contextualized word embeddings, we also experimented by incorporating part of speech(POS) tags and Word2Vec embeddings. POS tagging has not shown any improvements for NER (Tehseen et al., 2022, 2023) and the F_1-score remained around ∼0.78. We used the Stanford POS tagger (Toutanova et al., 2003) to tag the Wojood NER dataset and concatenated the POS encoding vectors with the word encoding vectors at the input layers of the models. The Arabic Word2Vec (Soliman et al., 2017) improved the results but the F_1-scores still remained under 0.82. The ELMo emeddings showed significant improvements by producing competitive results for Arabic NER.

5 Conclusion

This paper presents the description of the models and their performances for two shared tasks; i) flat NER and ii) nested NER for Arabic. We proposed Bidirectional LSTM-based single and multi-task learning models for both types of datasets. The incorporation of character-based contextualized word embeddings produced competitive results as compared to the baseline provided in the shared task.

References

Ankit Agrawal, Sarsij Tripathi, Manu Vardhan, Vikas Sihag, Gaurav Choudhary, and Nicola Dragoni. 2022. BERT-based transfer-learning approach for nested named-entity recognition using joint labeling. *Applied Sciences*, 12(3):976.

Muhammad Tayyab Ahmad, Muhammad Kamran Malik, Khurram Shahzad, Faisal Aslam, Asif Iqbal, Zubair Nawaz, and Faisal Bukhari. 2020. Named entity recognition and classification for Punjabi Shahmukhi. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 19(4):1–13.

Moustafa Al-Hajj and Mustafa Jarrar. 2022. ArabGlossBert: Fine-tuning BERT on Context-Gloss Pairs for WSD. *arXiv preprint arXiv:2205.09685*.

Veera Sekhar Reddy Bhumireddypalli, Srinivas Rao Koppula, and Neeraja Koppula. 2023. Enhanced conditional random field-long short-term memory for name entity recognition in English texts. *Concurrency and Computation: Practice and Experience*, 35(9):e7640.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146.

Wanxiang Che, Yijia Liu, Yuxuan Wang, Bo Zheng, and Ting Liu. 2018. Towards better UD parsing: Deep contextualized word embeddings, ensemble, and treebank concatenation. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 55–64, Brussels, Belgium. Association for Computational Linguistics.

Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. 2021. Template-based Named Entity Recognition using BART. *arXiv preprint arXiv:2106.01760*.

Sławomir Dadas and Jarosław Protasiewicz. 2020. A bidirectional iterative algorithm for nested named entity recognition. *IEEE Access*, 8:135091–135102.

Xiaojing Du, Yuxiang Jia, and Hongying Zan. 2022. MRC-based Medical NER with Multi-task Learning and Multi-strategies. In *China National Conference on Chinese Computational Linguistics*, pages 149–162. Springer.

Qin Fang, Yane Li, Hailin Feng, and Yaoping Ruan. 2023. Chinese Named Entity Recognition Model Based on Multi-Task Learning. *Applied Sciences*, 13(8):4770.

Murhaf Fares, Andrey Kutuzov, Stephan Oepen, and Erik Velldal. 2017. Word vectors, reuse, and replicability: Towards a community repository of large-text resources. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 271–276, Gothenburg, Sweden. Association for Computational Linguistics.

Xiaocheng Feng, Bing Qin, and Ting Liu. 2018. A language-independent neural network for event detection. *Science China Information Sciences*, 61:1–12.

Mustafa Jarrar, Muhammad Abdul-Mageed, Mohammed Khalilia, Bashar Talafha, AbdelRahim Elmadany, Nagham Hamad, and Alaa Omer. 2023. WojoodNER 2023: The First Arabic Named Entity Recognition Shared Task. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*.

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.

Mohammad Ebrahim Khademi and Mohammad Fakhredanesh. 2020. Persian Automatic Text Summarization based on Named Entity Recognition. *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, pages 1–12.

Lobna Ahmed Mady, Yasmine A Afify, and Nagwa Badr. 2022. Nested Biomedical Named Entity Recognition. *International Journal of Intelligent Computing and Information Sciences*, 22(1):98–107.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

Nita Patil, Ajay Patil, and BV Pawar. 2020. Named entity recognition using conditional random fields. *Procedia Computer Science*, 167:1181–1188.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. *arXiv preprint arXiv:1802.05365*.

Gorjan Popovski, Barbara Koroušić Seljak, and Tome Eftimov. 2020. A survey of named-entity recognition methods for food information extraction. *IEEE Access*, 8:31586–31594.

Mohammad Golam Sohrab and Makoto Miwa. 2018. Deep exhaustive model for nested named entity recognition. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2843–2849.

Abu Bakr Soliman, Kareem Eissa, and Samhaa R El-Beltagy. 2017. AraVec: A set of Arabic Word Embedding Models for use in Arabic NLP. *Procedia Computer Science*, 117:256–265.

Jana Straková, Milan Straka, and Jan Hajič. 2019. Neural architectures for nested NER through linearization. *arXiv preprint arXiv:1908.06926*.

Amina Tehseen, Toqeer Ehsan, Hannan Bin Liaqat, Amjad Ali, and Ala Al-Fuqaha. 2022. Neural POS Tagging of Shahmukhi by Using Contextualized Word Representations. *Journal of King Saud University-Computer and Information Sciences*.

Amina Tehseen, Toqeer Ehsan, Hannan Bin Liaqat, Xiangjie Kong, Amjad Ali, and Ala Al-Fuqaha. 2023. Shahmukhi Named Entity Recognition by using Contextualized Word Embeddings. *Expert Systems with Applications*, 229:120489.

Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. 2003. Feature-rich Part-of-Speech tagging with a cyclic dependency network. In *Proceedings of the 2003 human language technology conference of the north american chapter of the association for computational linguistics*, pages 252–259.

Van-Hai Vu, Quang-Phuoc Nguyen, Kiem-Hieu Nguyen, Joon-Choul Shin, and Cheol-Young Ock. 2020. Korean-vietnamese neural machine translation with named entity recognition and part-of-speech tags. *IEICE Transactions on Information and Systems*, 103(4):866–873.

Jue Wang, Lidan Shou, Ke Chen, and Gang Chen. 2020. Pyramid: A layered model for nested named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5918–5928.

Yibo Yan, Peng Zhu, Dawei Cheng, Fangzhou Yang, and Yifeng Luo. 2023. Adversarial Multi-Task Learning for Efficient Chinese Named Entity Recognition. *ACM Transactions on Asian and Low-Resource Language Information Processing*.

LIPN at WojoodNER shared task: A Span-Based Approach for Flat and Nested Arabic Named Entity Recognition

Niama Elkhbir[†⋆], **Urchade Zaratiana**[†×⋆], **Nadi Tomeh**[†], **Thierry Charnois**[†]
[×] FI Group, [†] LIPN, CNRS UMR 7030, France
{elkhbir,zaratiana,tomeh,charnois}@lipn.fr
[*]: Equal contribution

Abstract

The Wojood Named Entity Recognition (NER) shared task introduces a comprehensive Arabic NER dataset encompassing both flat and nested entity tasks, addressing the challenge of limited Arabic resources. In this paper, we present our team **LIPN** approach to addressing the two subtasks of WojoodNER Shared-Task. We frame NER as a span classification problem. We employ a pretrained language model for token representations and neural network classifiers. We use global decoding for flat NER and a greedy strategy for nested NER. Our model secured the first position in flat NER and the fourth position in nested NER during the competition, with an F-score of 91.96 and 92.45 respectively. Our code is publicly available (https://github.com/niamaelkhbir/LIPN-at-WojoodSharedTask).

1 Introduction

Named Entity Recognition (NER) plays a crucial role in various Natural Language Processing (NLP) applications, enabling the extraction and classification of entities from unstructured text. These entities span a wide range of categories, including individuals, organizations, locations, and dates, among others. While NER has witnessed significant progress, challenges persist, particularly in contexts marked by resource scarcity and linguistic complexity, such as the Arabic language.

In this context, the focus of Arabic NLP has predominantly revolved around flat entities (Liu et al., 2019; Helwe et al., 2020; Al-Qurishi and Souissi, 2021; El Khbir et al., 2022; Affi and Latiri, 2022), and the exploration of nested entity recognition in Arabic NLP has been relatively limited, primarily due to the scarcity of suitable nested Arabic datasets.

To address these limitations, the WojoodNER SharedTask 2023 (Jarrar et al., 2023) initiative was launched with the goal of overcoming these challenges. This initiative introduces the Wojood corpus (Jarrar et al., 2022), an extensively annotated Arabic NER dataset comprising approximately 550,000 tokens. It includes annotations for 21 distinct entity types, covering both Modern Standard Arabic (MSA) and dialectal variations, as well as flat and nested entity annotations.

The shared task objective is twofold: firstly, to encourage innovative solutions in flat NER, and secondly, to tackle nested NER. For both tasks, the aim is to develop models that can effectively identify and classify entities while accounting for complexities.

This paper outlines our strategy for tackling these subtasks. Our approach relies on a span-based methodology, employing token encoding, span enumeration, and subsequent classification. During inference, we employ global decoding for flat NER and a greedy decoding strategy for nested NER. Our contributions led us to achieve the top position in flat NER and the fourth position in nested NER during the WojoodNER SharedTask 2023.

In the following sections, we provide detailed insights into our methodology, experimentation, and the results achieved, highlighting the efficacy of our approach within the WojoodNER SharedTask 2023.

2 Related Work

Evolution of NER Approaches Early efforts in NER relied on handcrafted rules and lexicons for both flat (Zhou and Su, 2002) and nested entities (Shen et al., 2003; Zhang et al., 2004). Then, machine learning techniques gained prominence. Many studies focused on statistical models, such as Conditional Random Fields (CRFs) and Support Vector Machines (SVMs). These models demonstrated improved performance in identifying entities by capturing contextual dependencies and patterns within the data (McCallum and Li, 2003; Takeuchi and Collier, 2002). Deep learning

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 789–796
December 7, 2023 ©2023 Association for Computational Linguistics

techniques, particularly recurrent neural networks (RNNs) and recently, transformer-based architectures like BERT (Devlin et al., 2019) and GPT (Radford et al., 2019), revolutionized NER. These models leverage contextual embeddings to capture intricate relationships and dependencies, achieving state-of-the-art results in various languages and domains for both flat (Xia et al., 2019; Zheng et al., 2019; Arkhipov et al., 2019; Lothritz et al., 2020; Yu et al., 2020; Yang et al., 2021) and nested (Sohrab and Miwa, 2018; Katiyar and Cardie, 2018; Dadas and Protasiewicz, 2020; Wang et al., 2020) entities.

Approaches for NER Traditionally, NER tasks have been framed as sequence labeling (Lample et al., 2016; Akbik et al., 2018), i.e., token-level classification. Recently, innovative approaches have extended beyond token-level prediction. Some methods have treated NER as a question-answering problem (Li et al., 2020), while others have employed sequence-to-sequence models (Yan et al., 2021; Yang and Tu, 2022). In this work, we focus on span-based methods (Liu et al., 2016; Sohrab and Miwa, 2018; Fu et al., 2021; Zaratiana et al., 2022b), which involve enumerating all possible spans and then classifying them into specific entity types.

3 Data

	#Sentences	#Tokens	#F-Ent	#N-Ent
Train	16817	394500	50032	62403
Valid	3133	55827	7141	8854

Table 1: Statistics on Train and Validation Splits of Wojood Corpus.

The Wojood corpus is annotated for 21 different entity types, and it offers two versions: Wojood Flat and Wojood Nested. Both versions share identical training, validation, and test splits, differing only in the way entities are labeled. In Wojood Flat, each token receives a label corresponding to the first high-level label assigned to that token in Wojood Nested. Table 1 presents an overview of the statistics for the train and validation splits, including the number of sentences, tokens, flat entities (#F-Ent), and nested entities (#N-Ent).

Furthermore, Table 2 provides a breakdown of entity label counts for both flat and nested versions within the train and validation splits.

	Flat		Nested	
Label	**Train**	**Val**	**Train**	**Val**
CARDINAL	1245	182	1263	183
CURR	19	1	179	21
DATE	10667	1567	11291	1656
EVENT	1864	253	1935	267
FAC	689	85	882	111
GPE	8133	1132	15300	2163
LANGUAGE	131	15	132	15
LAW	374	44	374	44
LOC	510	63	619	76
MONEY	171	20	171	20
NORP	3505	488	3748	520
OCC	3774	544	3887	551
ORDINAL	2805	410	3488	504
ORG	10737	1566	10737	1566
PERCENT	105	13	105	13
PERS	4496	650	4996	730
PRODUCT	36	5	36	5
QUANTITY	44	3	46	3
TIME	286	55	288	55
UNIT	7	-	48	3
WEBSITE	434	45	434	45

Table 2: Entity Label Statistics in Wojood Corpus.

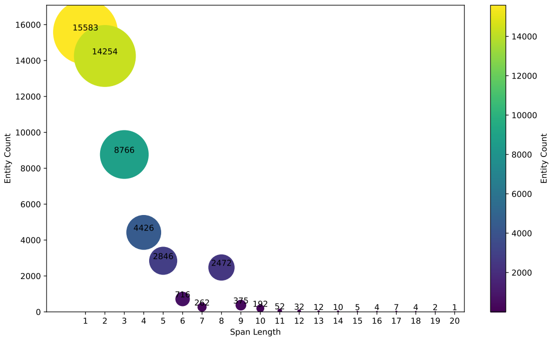


Figure 1: Entity count distribution by span length in the Flat Wojood training data.

To offer insights into the entity distribution based on span lengths, Figure 1 displays the entity count distribution concerning span lengths within the Flat Wojood training data. Note that for the sake of clarity in visualization, we have excluded entity counts for span lengths of 27, 29, 39, 43, and 124, each of which occurs either once or twice. We have established a maximum entity span length of 10 for our span-based model. Any entities surpassing this threshold are automatically excluded. Specifically, the training set includes 140 such entities, predominantly categorized as Website, Date, and Event. Similarly, in the validation set, 19 entities exceed the 10-span limit.

4 System

In this paper, we approach the named entity recognition task as a span classification problem. Given an input sequence $\boldsymbol{x} = \{x_i\}_{i=1}^{L}$, our goal is to classify all possible spans within the sequence, which can be defined as:

$$\boldsymbol{y} = \bigcup_{i=1}^{L} \bigcup_{j=i}^{L} s_{ijc} \tag{1}$$

where i, j, and c represent the start position, end position, and span type respectively. The probability of a specific span classification $\boldsymbol{y}$ given the input sequence $\boldsymbol{x}$ can be expressed as:

$$p_\theta(\boldsymbol{y}|\boldsymbol{x}) = \frac{\exp \sum_{s_{ijc} \in \boldsymbol{y}} \phi_\theta(s_{ijc}|\boldsymbol{x})}{\mathcal{Z}_\theta(\boldsymbol{x})} \tag{2}$$

where $\phi_\theta(.)$ is the span scoring function and $\mathcal{Z}_\theta(\boldsymbol{x})$ is the partition function. During training, our objective is to minimize the negative log-likelihood of the gold span classifications.

Decoding During inference, our aim is to determine:

$$\boldsymbol{y}^* = \arg\max_{\boldsymbol{y} \in \mathcal{Y}} \sum_{s_{ijc} \in \boldsymbol{y}} \phi_\theta(s_{ijc}|\boldsymbol{x}) \tag{3}$$

In other words, we seek to identify the span labeling configuration ($\boldsymbol{y} \in \mathcal{Y}$) that achieves the highest score (sum of individual span ($s_{ijc} \in \boldsymbol{y}$)). For unconstrained span classification, a straightforward approach is to assign the label with the highest score to each individual span, as follows:

$$s_{ijc^*} = \arg\max_{c} \phi_\theta(s_{ijc}|\boldsymbol{x}) \tag{4}$$

However, for both flat and nested NER, such a decoding strategy is suboptimal as it can lead to violations of structural constraints. For **flat NER**, where overlapping entity spans are not allowed, an efficient solution has been proposed in our previous works (Zaratiana et al., 2022c,a)[1]. This approach involves a two-stage decoding process: first, spans predicted as non-entities are filtered out, and then a maximum independent set algorithm is applied to the remaining spans to obtain the optimal set of entity spans. In contrast, for **nested NER**, where nesting is permitted but conflicting boundaries are prohibited, we employ a greedy algorithm to achieve a valid span classification. This algorithm iteratively selects the highest-scoring span that does not conflict with already selected spans.

	Flat NER		
TEAM	**P**	**R**	**F1**
LIPN (*Ours*)	92.56	91.36	91.96
El-Kawaref	91.43	92.48	91.95
ELYADATA	91.88	91.96	91.92
Alex-U 2023 NLP	91.61	92.00	91.80
tdink NER	90.76	91.73	91.25
	Nested NER		
TEAM	**P**	**R**	**F1**
ELYADATA	93.99	93.48	93.73
UM6P	92.46	93.61	93.03
AlexU-AIC	92.10	93.13	92.61
LIPN (*Ours*)	92.31	92.59	92.45
tdink NER	90.03	92.82	91.40

Table 3: Top 5 results for the Wojood flat/nested ner shared task.

Token and Span Representations In our approach, the span score $\phi_\theta(s_{ijc}|\boldsymbol{x})$ is computed as a linear projection of the span representation, obtained through a $1D$ convolution of token representations from a BERT-based model:

$$\boldsymbol{s}_{ijc} := w_c^T \texttt{Conv1D}_k([\boldsymbol{h}_i; \boldsymbol{h}_{i+1}; \ldots; \boldsymbol{h}_j]) \tag{5}$$

where $h_i \in \mathbb{R}^D$ is the token representation at position i, k is the size of the convolutional filter (j-i), and $w_c \in \mathbb{R}^D$ is a learned weight matrix for the span label c.

5 Results

Evaluation Metrics Following the shared task guidelines, we assess the performance of our model using precision, recall, and F1-score.

Settings and Hyperparameters For token representation, we use `bert-base-arabert` (Antoun et al., 2020) as a pretrained language model. Subsequently, we process the encoded tokens through a bidirectional Long Short-Term Memory (bi-LSTM) encoder to obtain the final representations. We set a maximum span length of 10 for enumerating all possible spans, which is a good balance between recall and training speed (Refer to Limitations Section).

Our model is trained with a batch size of 12 and evaluated with a batch size of 32. We set a learning rate of 5e-6 for BERT and 1e-3 for other model

[1] https://github.com/urchade/Filtered-Semi-Markov-CRF

parameters. We use the Adam optimizer and train our model for 50,000 steps, conducting evaluations every 250 steps.

We ran our experiments on a server equipped with v100 GPUs, and we estimated the needed computational budget for training to be 50 GPU hours.

Main results Eleven teams took part in the shared task, but due to space limitations, we present the results of the top 5 teams from the official leaderboard, which includes our own, in Table 3. The main results highlight the performance of our model in both the flat and nested Named Entity Recognition (NER) tasks. Our model achieved a good balance between precision and recall in both tasks, with a higher F-score in nested NER compared to flat NER.

Results by Class Table 4 presents the F1-scores associated with each label for both flat NER and nested NER on the validation set. Our model demonstrates high performance across both tasks for various entity types, including CURR, DATE, GPE, LAW, MONEY, ORDINAl, ORG, PERCENT, and PERS, all of which achieve an F-score exceeding 92.00.

The worst performance is observed for PRODUCT and WEBSITE, with F1-scores of 60.00 and 63.77, respectively. We provide further insights into this performance in section 6.2.

6 Discussion

6.1 Class Imbalance

One of the problems encountered in the Wojood dataset is class imbalance, where certain classes are significantly underrepresented in the training set. For example, the classes CURR, PRODUCT, QUANTITY, and UNIT constitute only 0.04%, 0.07%, 0.8%, and 0.01% of the training data, respectively. In contrast to dominant classes like DATE (21.23%), GPE (16.25%), and ORG (21.46%).

Such class imbalance can potentially skew evaluation results, especially when based solely on F-scores for these minority classes. Further work may involve sampling or data augmentation techniques to rebalance the dataset and provide more equitable representation and accurate assessment of the performance on these underrepresented classes.

6.2 Analysis of Model Errors

In this section, we analyze the remaining errors of our model in the validation set for flat NER.

Label	Flat	Nested
CARDINAL	89.44	87.98
CURR	100	100
DATE	96.20	96.46
EVENT	85.05	84.98
FAC	78.05	82.73
GPE	92.21	96.93
LANGUAGE	83.87	87.50
LAW	95.35	93.18
LOC	81.60	87.25
MONEY	95.00	91.89
NORP	79.25	79.20
OCC	89.66	89.99
ORDINAL	94.59	96.04
ORG	93.57	94.24
PERCENT	96.30	96.30
PERS	95.35	95.62
PRODUCT	60.00	66.67
QUANTITY	80.00	100
TIME	78.35	74.00
UNIT	-	80.00
WEBSITE	63.77	66.67

Table 4: F1-Scores by Entity Labels

Correct Span Offsets, but Incorrect Label Within this category, our model correctly identifies the span offsets but assigns incorrect labels to these spans. We identified a total of 68 instances where the model demonstrated this behavior.

To gain deeper insights into these errors, we provide in Figure 2 a visual representation of the confusion matrix for entity labels.

Approximately 45% of these errors arise from the ambiguity associated with certain entity labels, notably LOC, ORG and GPE. These errors often concern country or city names, such as الولايات المتحدة or السعودية, which, depending on the context, may belong to any of these categories.

Similarly, ambiguity between CARDINAL and ORDINAL labels accounts for 7% of this error category, while WEBSITE and ORG labels contribute approximately 6%. Also, NORP and ORG labels account for 7%. The remaining errors on labels can be found in Figure 2.

We observe comparable error patterns in the nested NER task. In Figure 3, we provide the confusion matrix for nested NER.

Span Boundary Errors with Correct Label Within this category, our model correctly predicts the entity label but fails to accurately identify the

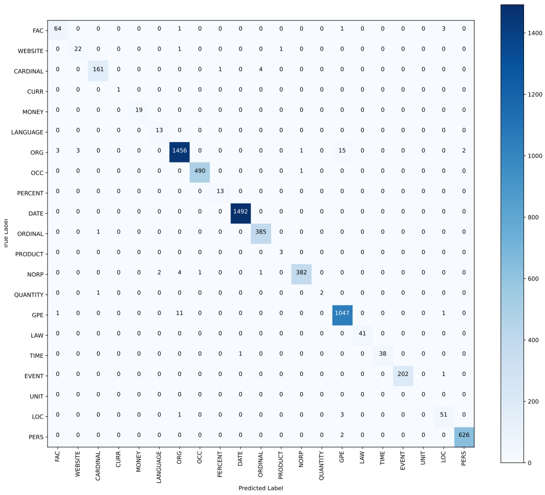


Figure 2: Confusion Matrix of Entity labels for flat NER.

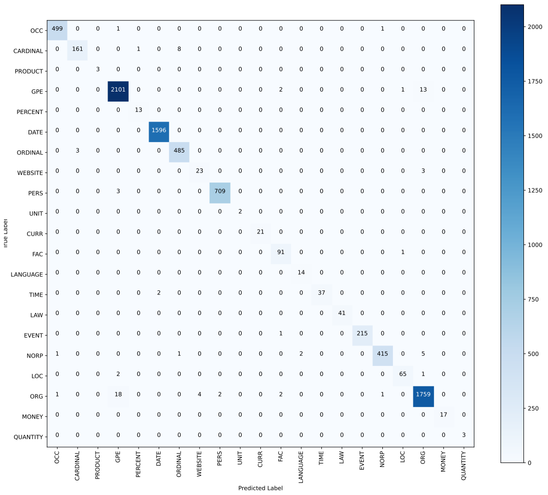


Figure 3: Confusion Matrix of Entity labels for nested NER.

start and end positions (span boundaries) of the entity within the text. We identified 167 instances where the model demonstrated this behavior. This category can be further broken down into two subtypes: (1) Span Start Error: The span start position is correct but the end position is incorrect; and (2) Span End Error: The span end position is correct but the start position is incorrect. Some of these errors seem to be annotation errors. See Table 5 for concrete examples.

False Negatives with Novel Entities Another type of error occurs when our model predicts spans that are not included in the gold annotations. We identified 305 instances where the model demonstrated this behavior. Although we did not conduct a precise quantification, a notable subset of these errors can be categorized as "false negatives". These false negatives are not part of the gold standard annotations, but they may have legitimacy as valid entities, thus the term "Novel Entities". Table 6 in the Appendix provides some illustrative examples of these errors.

7 Conclusion

Our approach to Arabic Named Entity Recognition in the WojoodNER Shared Task 2023 yielded competitive results, securing first place in flat NER and fourth in nested NER. This success highlights the potential of span-based methods and advanced decoding strategies. Moreover, we identified areas for improvement, including addressing class imbalance and refining span boundary predictions.

Limitations

Span Length Limitation Errors: In addition to the errors mentioned in Section 6.2, another type of errors is due to the span length limitation. As mentioned in Section 3, we have set a predefined limit of 10 tokens for span lengths, thus excluding all entities above this threshold. This decision was made to strike a balance between model complexity and computational efficiency. Due to this imposed constraint, our model cannot predict spans that surpass the 10-token threshold resulting in a reduced recall score. Particularly, with 140 and 19 spans surpassing the threshold in the training and validation set respectively, the maximum attainable recall score is 99.72 and 99.73 for the training and validation set respectively.

Acknowledgements

This work is partially supported by a public grantoverseen by the French National Research Agency (ANR) as part of the program Investissements d'Avenir (ANR-10-LABX-0083). This work was granted access to the HPC/AI resources of [CINES/IDRIS/TGCC] under the allocation 2023AD011013096R1 made by GENCI.

References

Manel Affi and Chiraz Latiri. 2022. Arabic named entity recognition using variant deep neural network architectures and combinatorial feature embedding based on cnn, lstm and bert. In *Proceedings of the 36th Pacific Asia Conference on Language, Information and Computation*, pages 302–312.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Muhammad Saleh Al-Qurishi and Riad Souissi. 2021. Arabic named entity recognition using transformer-based-crf model. In *Proceedings of the 4th International Conference on Natural Language and Speech Processing (ICNLSP 2021)*, pages 262–271.

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.

Mikhail Arkhipov, Maria Trofimova, Yuri Kuratov, and Alexey Sorokin. 2019. Tuning multilingual transformers for language-specific named entity recognition. In *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*, pages 89–93, Florence, Italy. Association for Computational Linguistics.

Sławomir Dadas and Jarosław Protasiewicz. 2020. A bidirectional iterative algorithm for nested named entity recognition. *IEEE Access*, 8:135091–135102.

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.

Niama El Khbir, Nadi Tomeh, and Thierry Charnois. 2022. Arabie: Joint entity, relation and event extraction for arabic. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 331–345.

Jinlan Fu, Xuanjing Huang, and Pengfei Liu. 2021. SpanNER: Named entity re-/recognition as span prediction. 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 7183–7195, Online. Association for Computational Linguistics.

Chadi Helwe, Ghassan Dib, Mohsen Shamas, and Shady Elbassuoni. 2020. A semi-supervised BERT approach for Arabic named entity recognition. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 49–57, Barcelona, Spain (Online). Association for Computational Linguistics.

Mustafa Jarrar, Muhammad Abdul-Mageed, Mohammed Khalilia, Bashar Talafha, AbdelRahim Elmadany, Nagham Hamad, and Alaa Omer. 2023. WojoodNER: The Arabic Named Entity Recognition Shared Task. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*.

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.

Arzoo Katiyar and Claire Cardie. 2018. Nested named entity recognition revisited. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 861–871, New Orleans, Louisiana. Association for Computational Linguistics.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020. A unified mrc framework for named entity recognition.

Liyuan Liu, Jingbo Shang, and Jiawei Han. 2019. Arabic named entity recognition: What works and what's next. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 60–67, Florence, Italy. Association for Computational Linguistics.

Yijia Liu, Wanxiang Che, Jiang Guo, Bing Qin, and Ting Liu. 2016. Exploring segment representations for neural segmentation models. In *IJCAI*.

Cedric Lothritz, Kevin Allix, Lisa Veiber, Tegawendé F. Bissyandé, and Jacques Klein. 2020. Evaluating pretrained transformer-based models on the task of fine-grained named entity recognition. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3750–3760, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Andrew McCallum and Wei Li. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Dan Shen, Jie Zhang, Guodong Zhou, Jian Su, and Chew-Lim Tan. 2003. Effective adaptation of hidden Markov model-based named entity recognizer for biomedical domain. In *Proceedings of the ACL 2003 Workshop on Natural Language Processing in Biomedicine*, pages 49–56, Sapporo, Japan. Association for Computational Linguistics.

Mohammad Golam Sohrab and Makoto Miwa. 2018. Deep exhaustive model for nested named entity recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2843–2849, Brussels, Belgium. Association for Computational Linguistics.

Koichi Takeuchi and Nigel Collier. 2002. Use of support vector machines in extended named entity recognition. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*.

Jue Wang, Lidan Shou, Ke Chen, and Gang Chen. 2020. Pyramid: A layered model for nested named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5918–5928, Online. Association for Computational Linguistics.

Congying Xia, Chenwei Zhang, Tao Yang, Yaliang Li, Nan Du, Xian Wu, Wei Fan, Fenglong Ma, and Philip Yu. 2019. Multi-grained named entity recognition. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1430–1440, Florence, Italy. Association for Computational Linguistics.

Hang Yan, Tao Gui, Junqi Dai, Qipeng Guo, Zheng Zhang, and Xipeng Qiu. 2021. A unified generative framework for various ner subtasks. In *ACL*.

Songlin Yang and Kewei Tu. 2022. Bottom-up constituency parsing and nested named entity recognition with pointer networks. In *ACL*.

Zhiwei Yang, Jing Ma, Hechang Chen, Yunke Zhang, and Yi Chang. 2021. HiTRANS: A hierarchical transformer network for nested named entity recognition. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 124–132, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jianfei Yu, Jing Jiang, Li Yang, and Rui Xia. 2020. Improving multimodal named entity recognition via entity span detection with unified multimodal transformer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3342–3352, Online. Association for Computational Linguistics.

Urchade Zaratiana, Niama Elkhbir, Pierre Holat, Nadi Tomeh, and Thierry Charnois. 2022a. Global span selection for named entity recognition. In *Proceedings of the Workshop on Unimodal and Multimodal Induction of Linguistic Structures (UM-IoS)*, pages 11–17, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. 2022b. GNNer: Reducing overlapping in span-based NER using graph neural networks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 97–103, Dublin, Ireland. Association for Computational Linguistics.

Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. 2022c. Named entity recognition as structured span prediction. In *Proceedings of the Workshop on Unimodal and Multimodal Induction of Linguistic Structures (UM-IoS)*, pages 1–10, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Jie Zhang, Dan Shen, Guodong Zhou, Jian Su, and Chew-Lim Tan. 2004. Enhancing hmm-based biomedical named entity recognition by studying special phenomena. *Journal of Biomedical Informatics*, 37(6):411–422. Named Entity Recognition in Biomedicine.

Changmeng Zheng, Yi Cai, Jingyun Xu, Ho-fung Leung, and Guandong Xu. 2019. A boundary-aware neural model for nested named entity recognition. 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 357–366, Hong Kong, China. Association for Computational Linguistics.

GuoDong Zhou and Jian Su. 2002. Named entity recognition using an hmm-based chunk tagger. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 473–480, USA. Association for Computational Linguistics.

A Example of Remaining Errors

Table 5 and 6 present examples of errors related to span boundaries and examples where the model predicts spans that are not part of the gold standard annotations, respectively.

Sentence	Gold Span	Predicted Span	Label
ضوابط حماية صغار المستثمرين	المستثمرين	صغار المستثمرين	NORP
لقد خلقت هده الأزمة انشقاقا في منطقة الشرق الأوسط وشمال افريقيا ...	الشرق الأوسط	منطقة الشرق الأوسط	LOC
شملت هذه القضايا سكانا سابقين في الغوطة، وحلب، وريف دمشق ...	دمشق	وريف دمشق	GPE
عندما التقينا في مدينة عدن الساحلية ...	مدينة عدن الساحلية	مدينة عدن	GPE
كوفئ اليمنيون، بمن فيهم المتظاهرون الذين شجعوا التغيير، باستفتاء في فبراير ...	المتظاهرون الذين شجعوا التغيير	المتظاهرون	NORP
بعد أسابيع قليلة في نيويورك، ...	بعد أسابيع قليلة	بعد أسابيع	DATE
إفراجات وغرامات مالية ضد المعتقلين الفلسطينيين (١٩٨٦)	الفلسطينيين	المعتقلين الفلسطينيين	NORP
برنامج خاص حول قمة واشنطن الرباعية تم إذاعته	قمة واشنطن	قمة واشنطن الرباعية	EVENT

Table 5: Example of Span Boundary Errors from the validation set for flat NER. The model predicts the correct label but fails to capture the gold span offsets.

Sentence	Predicted Span	Predicted Label
معرض المال والأعمال الأول	الأول	ORDINAL
ساعاتا بتزكّر نوزيا تبع سارتر	سارتر	PERS
اهتم بالاثنين فكلاهم مهم	بالاثنين	NORP
اهل قرية بيت نبالا اليوم	اليوم	DATE
في تخصصك: لنقل أنك مصمم	مصمم	OCC
المهم هسا الدكتور يرقص بقناة غنوه	الدكتور	OCC
welcome to geeksforgeeks	geeksforgeeks	WEBSITE
... والقرضاوي عم بيحرم الصلاة بالمسجد الاقصى	والقرضاوي	PERS
انا مسلم (رائع جدا)	مسلم	NORP
protect yourself from hackers	hackers	NORP
كيف أتابع ناروتو ؟	ناروتو	PERS
كان يا ما كان في قديم الزمان، ملك عنده ثلاث بنات	ثلاث	CARDINAL
الجميع يضغطون على الجرس ما عدا الخليجي	الخليجي	NORP
تتهي الاغنيه يرن هاتف المدير	المدير	OCC
بحسب نيرما جيلاسيتش، ممثلة اللجنة، تجمع اللجنة أدلة على الجرائم	ممثلة اللجنة	OCC
... فإن المجتمع الإيزيدي رفض الادعاءات	المجتمع الإيزيدي	NORP

Table 6: Example of False Negatives with Novel Entities from the validation set for flat NER. These are entities predicted by the model but not annotated in the dataset. All reported entities do not manifest any overlap with gold ones.

Alex-U 2023 NLP at WojoodNER shared task: AraBINDER (Bi-Encoder for Arabic Named Entity Recognition)

Mariam Hussein[*], **Sarah Khaled**[*], **Marwan Torki** and **Nagwa El-Makky**
Computer and Systems Engineering Department
Alexandria University
es-mariam99.mf, es-sara.khaled2019, mtorki, nagwamakky@alexu.edu.eg

Abstract

Named Entity Recognition (NER) is a crucial task in natural language processing that facilitates the extraction of vital information from text. However, NER for Arabic presents a significant challenge due to the language's unique characteristics. In this paper, we introduce AraBINDER, our submission to the Wojood NER Shared Task 2023 (ArabicNLP 2023). The shared task comprises two sub-tasks: sub-task 1 focuses on Flat NER, while sub-task 2 centers on Nested NER. We have participated in both sub-tasks. The Bi-Encoder has proven its efficiency for NER in English. We employ AraBINDER (Arabic Bi-Encoder for Named Entity Recognition), which uses the power of two transformer encoders and employs contrastive learning to map candidate text spans and entity types into the same vector representation space. This approach frames NER as a representation learning problem that maximizes the similarity between the vector representations of an entity mention and its type. Our experiments reveal that AraBINDER achieves a micro F-1 score of 0.918 for Flat NER and 0.9 for Nested NER on the Wojood dataset.

1 Introduction

Named Entity Recognition (NER) is a fundamental task in natural language processing that involves identifying and classifying named entities, such as person names, locations, organizations, and temporal expressions, within text. In recent years, deep learning models, particularly transformer-based architectures (Hanslo, 2022), have revolutionized NER by capturing contextual information effectively. However, applying these models to Arabic NER presents several difficulties. One of the major challenges is the lack of comprehensive and annotated Arabic NER data, which hinders the fair evaluation of Arabic NER models (Qu et al., 2023). Other previous work addressed Named entity Recognition as a Sequence labeling problem (Affi and Latiri, 2022), Span-based classification (Yu et al., 2020) or Seq-to-seq generation (Wang et al., 2019). There have been some approaches that dealt with the problem as a machine reading comprehension problem (MRC) (Li et al., 2020)(Elkordi et al., 2023). Meanwhile BINDER (Zhang et al., 2022) deals with NER as a representation learning problem that maximizes the similarity between the vector representations of an entity mention and its type. This makes it easy to handle Nested and Flat NER alike, and can better leverage noisy self-supervision signals. Moreover, it demonstrates superiority over past approaches in terms of speed and efficiency.

The use of dual networks dates back to (Bromley et al., 1993) for signature verification and (Chopra et al., 2005) for face verification. Moreover, (Humeau et al., 2019) conducted a comparison of three distinct architectures Bi-Encoder, Poly-Encoder, and Cross-Encoder all employing deep pre-trained transformers as encoders. In our solution, we use the Bi-Encoder architecture that has also been used in various tasks, such as information retrieval (Gillick et al., 2018), open-domain question answering (Karpukhin et al., 2020), and entity linking (Wu et al., 2020) and proved to achieve state of the art results.

Furthermore, in recent work, all tokens or spans that do not represent entities (non-entities) were categorized under a single class called "Outside" (O). Notably, our solution diverges from this conventional method since we use the proposed dynamic thresholding loss within the context of contrastive learning. This approach involves learning dynamic thresholds specific to candidates, aiding in the differentiation of entity spans from non-entity ones. While contrastive learning (CL) has considerably advanced numerous natural language processing (NLP) tasks, its application within the Arabic con-

[*] Equal contribution

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 797–802
December 7, 2023 ©2023 Association for Computational Linguistics

text has been somewhat limited(Qu et al., 2023). Recently, There has been a focus on this area as (Shapiro et al., 2022) demonstrated the efficacy of CL for Arabic hate speech detection, resulting in significant improvements over baselines. In a similar vein, (Abdul-Mageed and Lakshmanan, 2022) conducted experiments applying CL to diverse Arabic NLP tasks including dialect identification, emotion classification, sarcasm detection, and the identification of abusive and adult content.

In this paper, we bridge the gap by introducing AraBINDER, a novel approach to address these challenges. BINDER (Zhang et al., 2022), learns to differentiate between entities and non-entities, even when confronted with limited annotated data. This capability enhances its generalization potential, rendering it applicable to both Nested and Flat NER paradigms. For Sub-task 1 and Sub-task 2, we apply AraBINDER using our best model achieving micro F1 scores of 0.918 and 0.90 respectively.

2 Data

We conducted our work on the Wojood (Jarrar et al., 2022) dataset provided by the shared task (Jarrar et al., 2023). The shared task focuses on identifying named entity mentions in unstructured text and classifying them into predefined classes this is divided into two sub-tasks, sub-task 1 focuses on Flat NER while sub-task 2 centers on Nested NER. The data for sub-task 2 differed in the manner that in the Nested scheme some tokens had more than one entity type assigned to it.

The corpus of Wojood consists of about 27K sentences and 550K tokens and is manually annotated covering both Modern Standard Arabic (MSA) and Dialect Arabic (DA) in multiple domains. It contains about 75K entities, out of which 22.5% are Nested. The data was annotated for 21 entity types with IOB tags. The dataset introduced four new tags which are occupation, website, unit, and currency.

We follow the data split provided by the shared task: 70% of the data for training, 10% for development, and 20% for testing. Table 1 shows the label distribution of both Flat and Nested entities of the training and development sets. Since the provided data was IOB tagged, we have modified it by removing the tags and labeling each sentence with a unique ID. Also, the model uses the start and end of each span to modify the loss objective which is explained further on the paper for this purpose we extract the word's start and end characters for each sentence along with the start and end characters for entities in that sentence.

Tags	%Train	%Validation
PERS	8	8.24
NORP	6.01	5.87
OCC	6.23	6.22
ORG	21.12	21.11
GPE	24.52	24.43
LOC	0.99	0.86
FAC	1.41	1.25
PRODUCT	0.06	0.06
EVENT	3.1	3.02
DATE	18.1	18.7
TIME	0.46	0.62
LANGUAGE	0.21	0.17
WEBSITE	0.7	0.51
LAW	0.6	0.5
CARDINAL	2.02	2.07
ORDINAL	5.59	5.69
PERCENT	0.17	0.15
QUANTITY	0.07	0.03
UNIT	0.08	0.03
MONEY	0.27	0.23
CURR	0.29	0.24

Table 1: The distribution for Entity types in the train and validation sets of Wojood.

3 Method

In this section, we introduce the methodology of AraBINDER, which utilizes the Bi-Encoder architecture first introduced in (Zhang et al., 2022) for Arabic-named entity recognition (NER). The foundation of our model is the Bi-Encoder framework, which involves encoding both entity types and text using the Transformer-based architecture. To provide a comprehensive understanding, we begin by explaining the background of this Bi-Encoder framework. By leveraging the Bi-Encoder architecture and incorporating contrastive learning objectives, AraBINDER presents a robust and effective approach for Arabic NER. In the following sections, we will elaborate on the implementation details and experimental results to validate the performance of our proposed methodology.

3.1 Bi-Encoder for NER

The architecture of AraBINDER, as depicted in Figure 1, is based on a Bi-Encoder framework that has primarily been explored in the context of dense retrieval(Karpukhin et al., 2020). It has been put to the test in the case of NER for English and Chinese languages and demonstrated superior performance so we employ it to the Arabic language. The Bi-Encoder comprises two Transformer models, namely the entity type encoder and the text encoder, which are isomorphic and fully decoupled. For the task of NER, our model takes two types of inputs: entity-type descriptions and text containing potential named entities. At a high level, the entity type encoder generates representations for each entity of interest (e.g., "person" in Figure 1), while the text encoder produces representations for each input token in the given text where named entities may appear (e.g., "ميركل" in Figure 1). Based on these representations, we generate a set of span candidates and match them with each entity type in the vector space. As illustrated in Figure 1, the model aims to maximize the similarity between the entity type and positive spans while minimizing the similarity with negative spans. The introduction of Bidirectional Encoder Representation from Transformers (BERT) (Kenton and Toutanova, 2019) led to a revolution in the NLP world, as BERT-based models achieved state-of-the-art results in many tasks such as Machine Translation (Ghazvininejad et al., 2019), Question Answering (Yang et al., 2019), Text summarization (Zhang et al., 2019) and many more tasks. We utilize a pre-trained language model and fine-tune it for our NER task. In our experiments, we experimented with several pre-trained BERT-based models on Arabic such as CAMeLBERT (Inoue et al., 2021) and AraBERT with both versions (Antoun et al., 2020), but AraBERTv2 produced better results so we continued our experiments using it. AraBERTv2 has two different models that differ in the training data whereas the second one contains the same training data but in addition to 60M Multi-Dialect Tweets, most of its training data is MSA instead of DA. They also use Farasa (Darwish and Mubarak, 2016) Arabic morphological segmentation in the text preprocessing and we believe that this is beneficial to our task at hand based on the nature of the provided data, which contained some MSA.

3.2 Contrastive Learning

The primary goal of NER contrastive learning, illustrated in Figure 1, is to bring the representations of entity mention spans near their corresponding entity type embeddings (positive instances) and distant from irrelevant types (negative) in vector space. For instance, we aim to position the entity type "Person" closer to the mentioned span "ميركل " while maintaining a notable distance from any other word.

To accomplish this, we applied the multi-objective formulation in (Zhang et al., 2022) that comprises two distinct objectives based on the span and token embedding spaces, respectively. These objectives work together to guide the model in learning meaningful representations that capture the relationships between entity types and their associated mentions, enabling accurate and effective NER. Recognizing that the span-based objective in isolation might fall short, we enhance it with a position-based objective. The latter addresses a limitation where all negative spans receive equal penalties, even if they partially correspond to correct spans, for example, spans that share a common start or end token with the gold entity span. To address the challenge of predicting partially accurate spans, we introduce supplementary position-based contrastive learning objectives, which have the potential to enhance the model's ability to predict start and end positions more accurately.

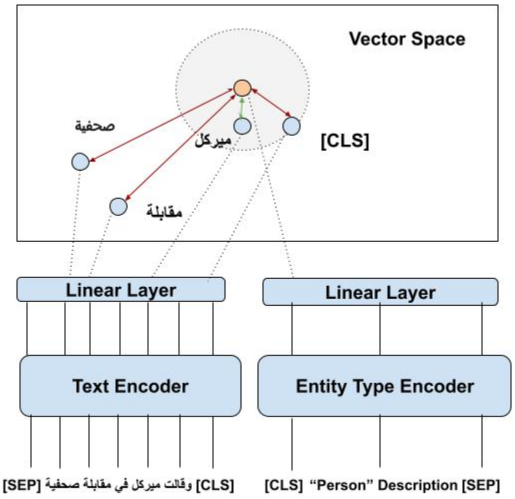


Figure 1: AraBINDER Architecture.

In the case of handling non-entities, the model, using the previously mentioned objectives may be able to distinguish between entities of different types, but it may fail to push away from non-entities to address this issue, we use the similarity between the special token [CLS] and the entity type as a dynamic threshold, as shown in Figure 1. Intuitively, the representation of [CLS] reads the entire input text and summarizes the contextual information, which could make it a good choice to estimate the threshold to separate entity spans from non-entity spans. In simpler terms, the final equation for the loss in Eq. (1) consists of three main parts, start loss, end loss, and span loss following the overall training objective in (Zhang et al., 2022). The equations of the three loss functions are given in (Zhang et al., 2022) and are not included here due to space limitations.

$$L = \alpha \ell_{\text{start}} + \gamma \ell_{\text{end}} + \lambda \ell_{\text{span}} \quad (1)$$

4 Experiments

4.1 Experiment Setting

All experiments were conducted using a single v100 GPU. We utilized the given training dataset for training our model and exploited the validation dataset to choose the hyper-parameters. A maximum input sequence length is set to 128, sequences greater than this length would be truncated and sequences less than this length would be padded to obtain the same length. For all experiments, we ignore sentence boundaries and tokenize and split text into sequences with a stride of 16. All base models are trained for 20 epochs with a learning rate of 3e-5 and a batch size of 8 sequences with a maximum token length of 128. For evaluation We follow the standard evaluation protocol and use micro F1, which indicates that a predicted entity span is considered correct if its span boundaries and the predicted entity type are both correct, we also include precision and recall in our results.

5 Results

In all our experiments, we exploit the AraBERTv2-Twitter base that is trained on MSA in addition to Multi-Dialect Arabic Tweets, since our data contain both MSA and DA in multiple domains. We demonstrate our results on the development set for Flat NER and Nested NER in Table 2 and Table 3 respectively, while Table 4 and Table 5 show Flat NER results and Nested NER results on the test set respectively.

6 Discussion

As can be shown from tables 2 and 3, the model performs better for Nested NER than for Flat NER on the development set. We noticed this behavior in several experiments. However, as can be seen from tables 4 and 5, it performs better for Flat NER on the test set, This indicates that the model may have failed to generalize. In the path forward, our focus will revolve around enhancing the performance of underperforming Nested experiments while delving into the exploration of alternative encoders for Arabic, such as JABER (Ghaddar et al., 2022), which could potentially enhance our results. Moreover, we are dedicated to further refine our data pre-processing strategies to tackle the unique challenges posed by Arabic, rectifying annotation errors, and addressing the scarcity of precise data.

Model	Recall	Precision	F1
AraBINDER(ours)	0.918	0.913	0.916

Table 2: Results of Flat NER on the development set.

Model	Recall	Precision	F1
AraBINDER(ours)	0.94	0.918	0.929

Table 3: Results of Nested NER on the development set.

Model	Recall	Precision	F1
(Jarrar et al., 2022)	-	-	87.33
AraBINDER(ours)	0.924	0.914	0.918

Table 4: Results of Flat NER on the test set.

Model	Recall	Precision	F1
(Jarrar et al., 2022)	-	-	0.91
AraBINDER(ours)	0.906	0.893	0.90

Table 5: Results of Nested NER on the test set.

7 Conclusion

In this paper, our approach revolves around the application of AraBINDER to tackle both Flat and Nested Named Entity Recognition (NER) tasks within the shared context. This methodology involves using a Bi-Encoder architecture, proficiently encoding both entity types and textual content. The infusion of contrastive learning into this framework serves to maximize the similarity between individual entity types and their corresponding mention spans.

Our evaluation revolved around BERT-based models trained on Arabic corpora, with a special focus on AraBERT. Through assessment, we observed that the AraBERTv2-Twitter base, pretrained on Arabic data encompassing Modern Standard Arabic (MSA) and Twitter data, performed the best. Notably, it performed better for the Flat NER task, outperforming its Nested NER counterpart.

Limitations

As shown during our experiments, The Nested NER results were not as good as expected and we believe that most of the mistakes were due to the challenge in the nature of Arabic data, and this is a problem for low-resource languages. We notice that some words that use conjunctions as person names may be confused with team names as in news reporting. For instance, "John and Johns" are two names for separate persons in English, while in Arabic, we find that the "و" is often linked to the following name "محمود واحمد" since there is no clear separation between them. This can be classified, at inference time, as a single-person entity with first and last names, instead of two separate person entities and this may lead to confusion.

References

Muhammad Abdul-Mageed and Laks VS Lakshmanan. 2022. A benchmark study of contrastive learning for arabic social meaning. *WANLP 2022*, page 63.

Manel Affi and Chiraz Latiri. 2022. Arabic named entity recognition using variant deep neural network architectures and combinatorial feature embedding based on cnn, lstm and bert. In *Proceedings of the 36th Pacific Asia Conference on Language, Information and Computation*, pages 302–312.

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.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1993. Signature verification using a" siamese" time delay neural network. *Advances in neural information processing systems*, 6.

Sumit Chopra, Raia Hadsell, and Yann LeCun. 2005. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546. IEEE.

Kareem Darwish and Hamdy Mubarak. 2016. Farasa: A new fast and accurate arabic word segmenter. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1070–1074.

Shereen Elkordi, Noha Adly, and Marwan Torki. 2023. Alexu-aic at wojoodner shared task: Sequence labeling vs mrc and swa for arabic named entity recognition. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023).*

Abbas Ghaddar, Yimeng Wu, Sunyam Bagga, Ahmad Rashid, Khalil Bibi, Mehdi Rezagholizadeh, Chao Xing, Yasheng Wang, Xinyu Duan, Zhefeng Wang, et al. 2022. Revisiting pre-trained language models and their evaluation for arabic natural language processing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3135–3151.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-predict: Parallel decoding of conditional masked language models. 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 6112–6121.

Daniel Gillick, Alessandro Presta, and Gaurav Singh Tomar. 2018. End-to-end retrieval in continuous space. *arXiv preprint arXiv:1811.08008.*

Ridewaan Hanslo. 2022. Deep learning transformer architecture for named-entity recognition on low-resourced languages: State of the art results. In *2022 17th Conference on Computer Science and Intelligence Systems (FedCSIS)*, pages 53–60. IEEE.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2019. Poly-encoders: Architectures and pre-training strategies for fast and accurate multi-sentence scoring. In *International Conference on Learning Representations.*

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.

Mustafa Jarrar, Muhammad Abdul-Mageed, Mohammed Khalilia, Bashar Talafha, AbdelRahim Elmadany, Nagham Hamad, and Alaa Omer. 2023. WojoodNER: The Arabic Named Entity Recognition Shared Task. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023).*

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.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1, page 2.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020. A unified mrc framework for named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5849–5859.

Xiaoye Qu, Yingjie Gu, Qingrong Xia, Zechang Li, Zhefeng Wang, and Baoxing Huai. 2023. A survey on arabic named entity recognition: Past, recent advances, and future trends. *arXiv preprint arXiv:2302.03512*.

Ahmad Shapiro, Ayman Khalafallah, and Marwan Torki. 2022. Alexu-aic at arabic hate speech 2022: Contrast to classify. 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 200–208.

Yu Wang, Yun Li, Ziye Zhu, Bin Xia, and Zheng Liu. 2019. Sc-ner: A sequence-to-sequence model with sentence classification for named entity recognition. In *Advances in Knowledge Discovery and Data Mining: 23rd Pacific-Asia Conference, PAKDD 2019, Macau, China, April 14-17, 2019, Proceedings, Part I 23*, pages 198–209. Springer.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020. Scalable zero-shot entity linking with dense entity retrieval. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6397–6407.

Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. End-to-end open-domain question answering with bertserini. *NAACL HLT 2019*, page 72.

Juntao Yu, Bernd Bohnet, and Massimo Poesio. 2020. Named entity recognition as dependency parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6470–6476.

Haoyu Zhang, Jingjing Cai, Jianjun Xu, and Ji Wang. 2019. Pretraining-based natural language generation for text summarization. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*. Association for Computational Linguistics.

Sheng Zhang, Hao Cheng, Jianfeng Gao, and Hoifung Poon. 2022. Optimizing bi-encoder for named entity recognition via contrastive learning. In *The Eleventh International Conference on Learning Representations*.

El-Kawaref at WojoodNER shared task: StagedNER for Arabic Named Entity Recognition

Nehal Elkaref
German University in Cairo
Cairo, Egypt
nehal.elkaref@student.guc.edu.eg

Mohab Elkaref
IBM Research Europe
Daresbury, United Kingdom
mohab.elkaref@ibm.com

Abstract

Named Entity Recognition (NER) is the task of identifying word-units that correspond to mentions as location, organization, person, or currency. In this shared task (Jarrar et al., 2023) we tackle flat-entity classification for Arabic, where for each word-unit a single entity should be identified. To resolve the classification problem we apply *StagedNER* as proposed by (Elkaref et al., 2023), which involves fine-tuning NER downstream tasks that divides the learning process of a transformer-model into two phases, where a model is tasked to learn sequence tags and then entity tags rather than learn both together simultaneously for an input sequence. We create an ensemble of two base models using this method that yield a score of F1 performance of 90.03% on the validation set and 91.95% on the test. The submitted model has ranked second for its F1 score, fourth in precision and ranked first scoring the highest recall.

1 Introduction

Named Entity Recognition (NER) is a vital sub-task for a plethora of NLP applications, those of which include machine translation (Ugawa et al., 2018), co-reference resolution(Clark and Manning, 2016) and information extraction (Cheng et al., 2021). The sub-task exhibits challenges when addressed from the lens of Arabic data, this comes back to the fact that the language is one of the richest in morphological inflections. To add more, attributes that typically help in locating entities such as capitalisation is not featured in the language. Arabic is also agglutinative in nature where one word could be combination of lemma, prefixes and suffixes (AbdelRahman et al., 2010) (Qu et al., 2023).

Arabic NER (ANER) has been approached using a wide spectrum of methods through the years however, more recently development of pre-trained language models (PLMs) specifically transformer-based models that learn context-aware representations has elevated the performance on ANER datasets. These models include MARBERT and ARBERT (Abdul-Mageed et al., 2021), AraBERT(Antoun et al., 2020a).

The architecture of these PLMs has been extended and equipped with different networks. To exemplify, (Al-Qurishi and Souissi, 2021) utilized a range of transformer based models namely AraBERT, XLM-Roberta (Conneau et al., 2019) and AraElectra (Antoun et al., 2020b) coupled with Conditional Random Field (CRF) to fine-tune an ANER downstream task revealing that AraBERT exhibited the highest scores. BiLSTM and BiGRU-CRF models have also been fine-tuned on Arabic BERT in an attempt to classify entities based on classical Arabic. (Alsaaran and Alrabiah, 2021). In similar vein, we leverage transformer based models to classify flat entities. However, we employ an alternative technique to fine-tuning PLMs on NER tasks where the learning regiment for a model is distributed over two stages for better learning (Elkaref et al., 2023).

In the next sections we begin by describing the data purposed for this shared task (Jarrar et al., 2023) in section 2 and highlight how we re-purposed train, validation and test sets to perform a two-staged fine-tuning process. Next, we give an extensive explanation of our adopted fine-tuning method in section 3. In sections 4 and 5 we present results of the submitted system, and discuss and analyse system performance on the validation set. Finally, we summarize and recap the proposed system and re-highlight performance scores and findings in section 6.

2 Data

Data utilised was from Wojood corpus (Jarrar et al., 2022), a rich and substantial corpus for Arabic NER that encompasses a wide range of entity types.

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 803–808
December 7, 2023 ©2023 Association for Computational Linguistics

The corpus is also further extended to include annotation for nested entities, however for the scope of this shared task paper only annotations purposed for flat entities are used. The total number of tokens amounts to over 550K of Modern Standard Arabic (MSA) and dialectical Arabic tokens. To add more, MSA tokens are more frequent, where about 86% of tokens are MSA and the rest come in the Levant dialect. The corpus covers a different domains for each Arabic class; MSA tokens were acquired from two resources, the Birzeit University digital Palestinian archive, "Awraq", and online articles[1]. The former covers cultural heritage and modern history of Palestine while the latter includes web articles of health, law, finance, politics, migration, terrorism, ICT and elections. Meanwhile, dialectical tokens were obtained from supplementary Lebanese and Palestinian corpora (Haff et al., 2022) (Jarrar et al., 2014) (Jarrar et al., 2017) and other additional Levant resources collectively discussing general topics. Train, development and test splits were provided; as depicted in figure 1 Out of word vocabulary (O) instances exceed other entity classes, where there were about 258K and 36K in the train and validation sets of O instances, for that reason figure 1 brings focus to other less dominant meaningful entities; whereby Date, Organization, Geopolitical, Occupation and Person are recurring throughout the data in comparison to Language, Product, Quantity, Currency and Unit which are rarely present.

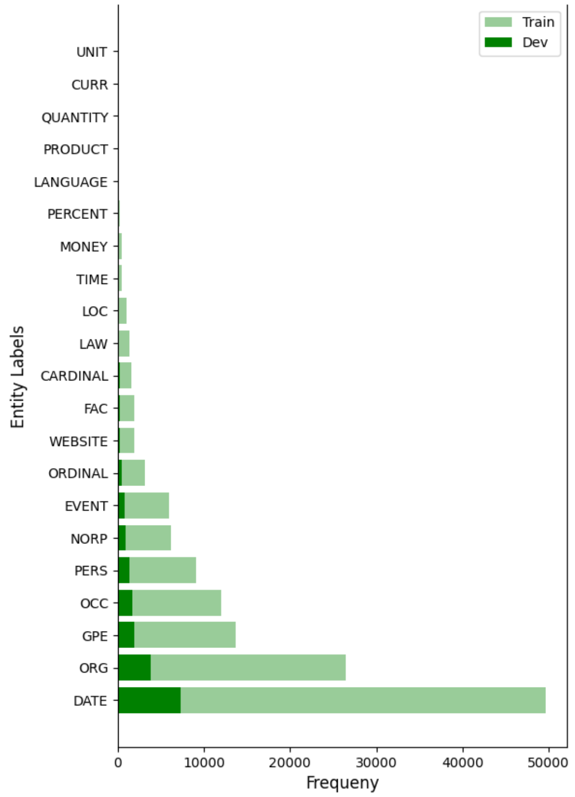


Figure 1: Entity distribution of train and validation sets of Wojood NER corpus without out-of-word vocabulary tag

As briefly mentioned before, the core idea of the proposed learning technique relies on separation of learning of sequence labels (BIO) tags and entity classes, hence the data goes through a separation of sequence labels and entities. Moreover, we rely on AraBERT's pre-processor[2] whereby diacritics and elongations are removed by default.

3 System Description

The backbone of the submitted system relies on fine-tuning a language model based on BERT's transformer architecture (Devlin et al., 2018). Typically, data utilised in the fine-tuning process for NER tasks follows BIO format, whereby at word-level, each entity is accompanied by an appropriate B (beginning) or I (inside), or O (outside) tag, hence the model is tasked to learn a position of an entity and the entity itself altogether. Meanwhile, we adopt the *StagedNER* approach (Elkaref et al., 2023) whereby the learning process is split into two sub-tasks, the first mimics a sequence-labelling problem where the model learns to assign appropriate BIO tags for each input, and the second sub-task is the original entity classification task. We note that this method is not analogous to sequential learning, as two separate instances of a transformer are leveraged in this method, thus each instance is assigned to exclusively learn either a BIO tag or an entity class.

Classifying BIO tags The first stage entails fine-tuning transformer on simply BIO tags of input sequences. To strengthen the transformer's learning at this stage we supply it with Part-of-Speech (POS) tags as an additional feature to help identify class spaces better, where representations from the model are pooled and summed with its appropriate POS tag.

Classifying Entity types In the second stage, a second untrained instance of the same transformer is utilized and is fine-tuned to predict

[1] un.org,hrw.org,msf.org,who.org,mipa.institute,elections.ps,sa.usembassy.gov,diplo-matie.ma,quora.com

[2] https://github.com/aub-mind/arabert

entity classes. Additionally, BIO labels predicted from the first stage are passed to the model, in doing so, we ensure during entity prediction time the transformer is aware of the boundaries of entity.

Overall Framework In figure 2 we illustrate StagedNER's framework bottom to top, the input sequence is passed on to the first transformer instance, where resulting representations for sub-word tokens are summed then fed to the classification layer to predict output BIO tags. Additionally, to incorporate POS tags, they are firstly added to the tokenizer as special tokens and then inserted between input token sequences. Next, the original input sequence is given to the second transformer where once again sub-word tokens are summed. When summing sub-word tokens, BIO tags from the first stage are taken and leveraged in-order to pool vectors representing the beginning and end of an entity. The pooled vectors are finally passed onto the classification layer to predict entity types. We note that during training and validation BIO tags utilised were the GOLD BIO tags, while for the test set we relied on predicted BIO tags from the first stage transformer.

3.1 Experimental Setup

We utilise AraBERTV02 (Antoun et al., 2020a) a transformer-based language model pre-trained on a collection of Arabic corpora[3] majority of which is in MSA. POS tags are generated for train, development and test sets using CAMel Tools Part-of-Speech tagger (Obeid et al., 2020) for MSA and Levant (LEV) each exclusively. Leveraging POS tags of different classes of Arabic was motivated by the nature of that data being a mix of dialectical (Levant) and MSA. (Jarrar et al., 2022)

Two instances of AraBERT are fine-tuned according to hyperparameters mentioned in table 1. We experimented with the same range of learning rates for both AraBERT transformers but we found 5e-5 to work best along with a batch size of 8 while the same dropout rate was used consistently in all experiments. Additional information about infrastructure are given in table 2. A span of experiments is conducted to yield three models, each of which utilises either Levant or MSA POS tags or no tags at all. In doing so, we produce four ensembles using different combinations of these three models.

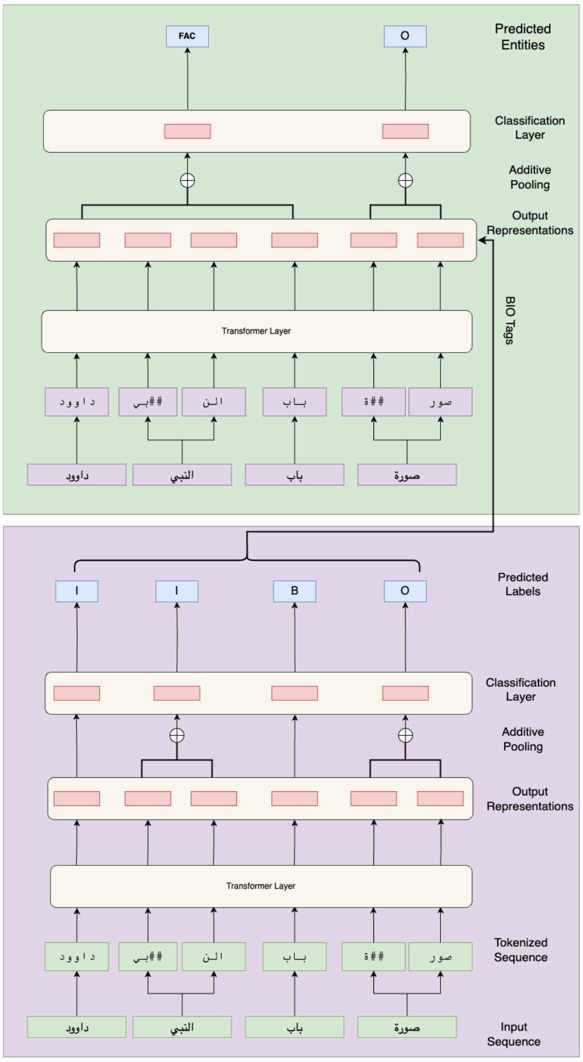


Figure 2: Bottom-up illustration of StagedNER framework, starting with BIO tag identification stage and up to entity classification stage

Hyperparameter	Value
learning rate	5e-6, 2e-6, 5e-5
dropout	0.1
epochs	8, 16

Table 1: Hyperparameter experimented with

Infrastructure	
GPU	A100 80GB
training time (mins/epoch)	11

Table 2: Infrastructure utilized

We submit four ensembles, three of which are comprised of two models that use MSA or Levant

[3] https://huggingface.co/aubmindlab/bert-large-arabertv02#dataset

POS tags or none at all, and a final variant that ensembles systems that uses MSA, LEV and no tags at all.

4 Results

We report micro F1, precision and recall scores for development and test sets in tables 4 and 5 for every ensemble and their unique POS combination. Additionally we showcase our performance compared to other teams in table 3 Scores show that the best performing model in-terms of F1 and recall is the forth ensemble for both validation and test sets that combined two base models, the first utilised MSA POS tags while the second relied only on representations learned during training. However for precision scores, the ensemble falls behind by 0.01% to an ensemble that leverages Levant and MSA POS tags.

Team	F1	P	R	Rank
LIPN	**91.96%**	**92.56%**	91.36%	1
ELYADATA	91.92%	91.88%	91.96%	3
Alex-U 2023 NLP	91.80%	91.61%	92.00%	4
tdink NER	91.25%	90.76%	91.73%	5
Our System	91.95%	91.43%	**92.48%**	2

Table 3: Shared task leaderboard and F1, precision and recall scores on the test set

Ensemble	DEV		
	F1	P	R
Baseline	86.81%	-	-
LEV + MSA + No POS	89.94%	88.92%	90.98%
LEV + MSA	89.95%	89.08%	90.84%
LEV + No POS	89.16%	89.90%	90.8%
MSA + No POS	**90.03%**	**88.92%**	**91.12%**

Table 4: F1, precision and recall scores for the validation set

5 Discussion

By inspecting tables 4 and 5, we can see that ensembles that incorporated MSA POS tags has had the highest F1 scores, this is analogous with the Arabic class distribution within the dataset, where majority of the data is curated and collected from MSA resources. We report additional F1, percision and recall below.

Ensemble	TEST		
	F1	P	R
LEV + MSA + No POS	91.88%	91.33%	92.44%
LEV + MSA	91.92%	**91.44%**	92.40%
LEV + No POS	91.78%	91.11%	92.45%
MSA + No POS	**91.95%**	91.43%	**92.48%**

Table 5: F1, precision and recall scores for test set

Entity	P	R	F1
CURR	00.00%	00.00%	00.00%
DATE	94.37%	95.21%	94.79%
EVENT	73.78%	77.87%	75.78%
FAC	72.41%	74.12%	73.26%
GPE	90.01%	91.52%	90.76%
LANGUAGE	85.71%	80.00%	82.76%
LAW	82.98%	88.64%	85.71%
LOC	71.64%	76.190%	73.85%
MONEY	95.00%	95.00%	95.00%
NORP	73.53%	79.88%	76.58%
OCC	85.89%	89.52%	87.67%
ORDINAL	95.134%	95.60%	95.36%
ORG	91.08%	93.29%	92.17%
PERCENT	00.00%	00.00%	00.00%
PERS	93.15%	96.31%	94.70%
PRODUCT	50.00%	40.00%	44.44%
QUANTITY	50.00%	66.67%	57.14%
TIME	75.00%	65.45%	69.90%
WEBSITE	54.54%	53.33%	53.93%

Table 6: Scores per entity class

By inspecting table 6, we find that ensemble had no problem classifying regularly occurring entities such as Date, GPE, ORG and PERS and managed to perform competitively on less occurring entities such as ORDINAL. The ensemble however falls behind on WEBSITE and PRODUCT. When examining instances belonging to such entities we found them to be either dialectical or even non-Arabic such as **AK** or واوروخ . This in-turn suggests relying on MSA POS tags is not enough, and using a model that was not exposed to non-Arabic data during pre-training might not be the ideal choice when dealing with non MSA data, therefore stronger POS for dialectical data is required that has been trained on a diverse range of topics. Moreover, we hypothesise that a model pre-trained on dialectical data such as MARBERT if it was part of the ensemble we would have witnessed stronger results.

6 Conclusion

In this shared task, we tackled flat entity classification on Wojood corpus, a predominantly MSA dataset, where we applied an alternative fine-tuning method, where one model is used to learn BIO tags and another separate model is used to learn entity classes, instead of a single model that learns to perform both tasks jointly. The motivation behind this was to lessen the number of classes a model had to learn; where instead of learning sequence variations of one entity such as I-ORG, B-ORG, the model simply learns to identify ORG and another model is tasked to learn BIO sequence tags. To strengthen the learning of BIO tags we equip the model with MSA and Levant POS tags and created four ensembles based on different combinations of them. Results show that having MSA POS tags made a difference in performance where the best performing ensemble that include MSA POS tags scored 90.03% and 91.95% on the development and test sets respectively. Our best performing model can be demonstrated on HuggingFace Spaces[4].

References

Samir AbdelRahman, Mohamed Elarnaoty, Marwa Magdy, and Aly Fahmy. 2010. Integrated machine learning techniques for arabic named entity recognition. *IJCSI*, 7(4):27–36.

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.

Muhammad Saleh Al-Qurishi and Riad Souissi. 2021. Arabic named entity recognition using transformer-based-crf model. In *Proceedings of the 4th International Conference on Natural Language and Speech Processing (ICNLSP 2021)*, pages 262–271.

Norah Alsaaran and Maha Alrabiah. 2021. Classical arabic named entity recognition using variant deep neural network architectures and bert. *IEEE Access*, 9:91537–91547.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020a. AraBERT: Transformer-based model for Arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France. European Language Resource Association.

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

Qiao Cheng, Juntao Liu, Xiaoye Qu, Jin Zhao, Jiaqing Liang, Zhefeng Wang, Baoxing Huai, Nicholas Jing Yuan, and Yanghua Xiao. 2021. Hacred: A large-scale relation extraction dataset toward hard cases in practical applications. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2819–2831.

Kevin Clark and Christopher D Manning. 2016. Improving coreference resolution by learning entity-level distributed representations. *arXiv preprint arXiv:1606.01323*.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Mohab Elkaref, Nathan Herr, Shinnosuke Tanaka, and Geeth De Mel. 2023. NLPeople at SemEval-2023 task 2: A staged approach for multilingual named entity recognition. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1148–1153, Toronto, Canada. Association for Computational Linguistics.

Karim El Haff, Mustafa Jarrar, Tymaa Hammouda, and Fadi Zaraket. 2022. Curras+ baladi: Towards a levantine corpus. *arXiv preprint arXiv:2205.09692*.

Mustafa Jarrar, Muhammad Abdul-Mageed, Mohammed Khalilia, Bashar Talafha, AbdelRahim Elmadany, Nagham Hamad, and Alaa Omer. 2023. WojoodNER 2023: The First Arabic Named Entity Recognition Shared Task. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*.

[4] https://huggingface.co/spaces/nehalelkaref/flat-arabic-entity-classification

Mustafa Jarrar, Nizar Habash, Diyam Akra, and Nasser Zalmout. 2014. Building a corpus for palestinian arabic: a preliminary study. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 18–27.

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*, 51:745–775.

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.

Ossama Obeid, Nasser Zalmout, Salam Khalifa, Dima Taji, Mai Oudah, Bashar Alhafni, Go Inoue, Fadhl Eryani, Alexander Erdmann, and Nizar Habash. 2020. CAMeL tools: An open source python toolkit for Arabic natural language processing. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 7022–7032, Marseille, France. European Language Resources Association.

Xiaoye Qu, Yingjie Gu, Qingrong Xia, Zechang Li, Zhefeng Wang, and Baoxing Huai. 2023. A survey on arabic named entity recognition: Past, recent advances, and future trends. *arXiv preprint arXiv:2302.03512*.

Arata Ugawa, Akihiro Tamura, Takashi Ninomiya, Hiroya Takamura, and Manabu Okumura. 2018. Neural machine translation incorporating named entity. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3240–3250.

Author Index