LREC-COLING 2024

The 6th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT) with Shared Tasks on Arabic LLMs Hallucination and Dialect to MSA Machine Translation

Workshop Proceedings

Editors
Hend Al-Khalifa, Kareem Darwish,
Hamdy Mubarak, Mona Ali
and Tamer Elsayed

25 May, 2024
Torino, Italia

Proceedings of The 6th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT) with Shared Tasks on Arabic LLMs Hallucination and Dialect to MSA Machine Translation

ISBN 978-2-493814-36-4
ISSN 2951-2093 (COLING); 2522-2686 (LREC)

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

Preface

Following the success of five editions of the of Open-Source Arabic Corpora and Corpora Processing Tools (OSACT) workshop collocated with LREC 2014, LREC 2016, LREC 2018, LREC 2020, and LREC2022, the sixth workshop comes to enable researchers and practitioners of Arabic language technologies to present their research with associated data and tools and to push the boundaries of their work in computational linguistics (CL), natural language processing (NLP), and information retrieval (IR). The sixth iteration gives special attention to areas of timely interest to the community, namely Large Language Models (LLMs), Generative AI, and dialectal translation, with two dedicated shared tasks on detecting LLM hallucinations and dialects to Modern Standard Arabic (MSA) translation.

OSACT6 had an acceptance rate of 43%, where we received 23 regular papers from which 10 papers were accepted, in addition to 6 shared task papers. We believe that the accepted papers are of high quality and present a mixture of interesting topics.

This year, we introduced the Shared Task on Dialectal Arabic (DA) to Modern Standard Arabic (MSA) Machine Translation, which attracted many teams from different countries in the Middle East, Europe, and the US. For this shared task, 29 teams signed up, and six teams made submissions to the competition's leaderboard, with five of them submitting their system description papers.

The other shared task aimed to address hallucinations (generation of false or misleading content) in Arabic Large Language Models (LLMs), such as GPT-3.5 and GPT-4. It features a dataset of 10,000 sentences from these LLMs annotated for factuality and correctness. There were two subtasks: A) detecting if a given sentence is factually correct, incorrect, or non-factual without additional information; and B) detecting the accuracy using the model's name, input word, part-of-speech (POS), and readability level. Only one team signed up and submitted a system paper.

Finally, we would like to thank everyone who in one way or another helped in making this workshop a success. Our special thanks go to the members of the program committee, who did an excellent job in reviewing the submitted papers, and to the LREC-COLING-2024 organizers. Finally, we would like to thank our authors and the workshop participants.

This volume documents the Proceedings of the 6th Workshop on Open-Source Arabic Corpora and Processing Tools, held on 25 May 2024 as part of the LREC-COLING-2024 conference.

Hend Al-Khalifa, Kareem Darwish, Hamdy Mubarak,
Mona Ali and Tamer Elsayed
OSACT6 Organizing Committee

Organizing Committee

- Hend Al-Khalifa, King Saud University, KSA
- Hamdy Mubarak, Qatar Computing Research Institute, Qatar
- Kareem Darwish, aiXplain Inc., US
- Tamer Elsayed, Qatar University, Qatar
- Mona Ali, Northeastern University, Canada

Programme Committee

- Ganesh Jawahar, University of British Columbia, Canada
- Go Inoue, Mohamed bin Zayed University of Artificial Intelligence, UAE
- Bassam Haddad, University of Petra, Jordan
- Hamada Nayel, Banha University, Egypt
- Ibrahim Abu Farha, The University of Sheffield, UK
- Imed Zitouni, Google, USA
- Almoataz B. Al-Said, Cairo University, Egypt
- Mourad Abbas, Assistant Secretary-General of Al-Tnall Al-Arabi in Algeria
- Nada Ghneim, Arab International University, Syria
- Omar Trigui, University of Sousse, Tunisia
- Salima Harrat, École Normale Supérieure de Bouzaréah (ENSB), Algeria
- Salima Mdhaffar, Avignon University (LIA), France
- Kamel Smaili, University of Lorraine, France
- Violetta Cavalli-Sforza, Al Akhawayn University, Morocco
- Wassim El-Hajj, American University of Beirut, Lebanon
- Wissam Antoun, ALMAnaCH - INRIA Paris, France
- Nada Almarwani, Taibah University, KSA
- Samah Aloufi, Taibah University, KSA
- Imene Bensalem, Constantine 2 University, Algeria
- Abdelkader El Mahdaouy, Mohammed VI Polytechnic University, Morocco
- Amr Keleg, University of Edinburgh, UK
- Wajdi Zaghouani, Hamad Bin Khalifa University, Qatar
- Amr El-Gendy, Arab Academy, Egypt

- Maha Alamri, AlBaha University, KSA
- Saied Alshahrani, Clarkson University, USA
- Lubna Alhenaki, Majmaah University, KSA
- Fatimah Alqahtani, Jazan University, KSA
- Eman Albilali, King Saud Univeristy, KSA
- Ahmed Abdelali, SDAIA, KSA
- Mohamed Al-Badrashiny, aiXplain Inc., US
- Firoj Alam, QCRI, Qatar
- Norah Alzahrani, SDAIA, KSA
- Nadir Durrani, QCRI, Qatar
- Ashraf Elneima, aiXplain Inc., US
- Nizar Habash, NYU-AD, UAE
- Walid Magdy, University of Edinburgh, UK
- Zaid Alyafeai, KFUPM, KSA
- Injy Hamed, NYU-AD, UAE
- Fouzi Harrag, Ferhat Abbas University, Algeria

Table of Contents

Workshop Program

Saturday 25 May 2024

Session 1: Main Workshop

9:00–9:10 ***Workshop Opening***

9:10–9:50 *Keynote Talk: Towards Arab-Centric Large Language Models*
Muhammad Abdul-Mageed

9:50–10:10 *AraTar: A Corpus to Support the Fine-grained Detection of Hate Speech Targets in the Arabic Language*
Seham Alghamdi, Youcef Benkhedda, Basma Alharbi and Riza Batista-Navarro

10:10–10:30 *CLEANANERCorp: Identifying and Correcting Incorrect Labels in the ANERcorp Dataset*
Mashael AlDuwais, Hend Al-Khalifa and Abdulmalik AlSalman

Session 2: Main Workshop (Cont.)

11:00–11:20 *Munazarat 1.0: A Corpus of Arabic Competitive Debates*
Mohammad M. Khader, AbdulGabbar Al-Sharafi, Mohamad Hamza Al-Sioufy, Wajdi Zaghouani and Ali Al-Zawqari

11:20–11:40 *Leveraging Corpus Metadata to Detect Template-based Translation: An Exploratory Case Study of the Egyptian Arabic Wikipedia Edition*
Saied Alshahrani, Hesham Haroon Mohammed, Ali Elfilali, Mariama Njie and Jeanna Matthews

11:40–12:00 *A Novel Approach for Root Selection in the Dependency Parsing*
Sharefah Ahmed Al-Ghamdi, Hend Al-Khalifa and Abdulmalik AlSalman

12:00–12:20 *AraMed: Arabic Medical Question Answering using Pretrained Transformer Language Models*
Ashwag Alasmari, sarah alhumoud and Waad Alshammari

12:20–12:40 *The Multilingual Corpus of World's Constitutions (MCWC)*
Mo El-Haj and Saad Ezzini

12:40–13:00 *TafsirExtractor: Text Preprocessing Pipeline preparing Classical Arabic Literature for Machine Learning Applications*
Carl Kruse and Sajawel Ahmed

Saturday 25 May 2024 (continued)

Session 3: Main Workshop (Cont.)

14:00–14:20 *Advancing the Arabic WordNet: Elevating Content Quality*

Abed Alhakim Freihat, Hadi Mahmoud Khalilia, Gábor Bella and Fausto Giunchiglia

14:20–14:40 *Arabic Speech Recognition of zero-resourced Languages: A case of Shehri (Jibbali) Language*

Norah A. Alrashoudi, Omar Said Alshahri and Hend Al-Khalifa

Session 4: Shared Tasks

14:40–14:55 *OSACT6 Dialect to MSA Translation Shared Task Overview*

Ashraf Hatim Elneima, AhmedElmogtaba Abdelmoniem Ali Abdelaziz and Kareem Darwish

14:55–15:10 *OSACT 2024 Task 2: Arabic Dialect to MSA Translation*

hanin atwany, Nour Rabih, Ibrahim Mohammed, Abdul Waheed and Bhiksha Raj

15:10–15:25 *ASOS at OSACT6 Shared Task: Investigation of Data Augmentation in Arabic Dialect-MSA Translation*

Omer Nacar, Abdullah Alharbi, Serry Sibaee, Samar Ahmed, Lahouari Ghouti and Anis Koubaa

15:25–15:50 *LLM-based MT Data Creation: Dialectal to MSA Translation Shared Task*

AhmedElmogtaba Abdelmoniem Ali Abdelaziz, Ashraf Hatim Elneima and Kareem Darwish

15:50–16:00 *Sirius_Translators at OSACT6 2024 Shared Task: Fin-tuning Ara-T5 Models for Translating Arabic Dialectal Text to Modern Standard Arabic*

Salwa Saad Alahmari

Saturday 25 May 2024 (continued)

Session 5: Shared Tasks (Cont.)

16:30–16:45 *AraT5-MSAizer: Translating Dialectal Arabic to MSA*

Murhaf Fares

16:45–17:00 *ASOS at Arabic LLMs Hallucinations 2024: Can LLMs detect their Hallucinations :)*

Serry Taiseer Sibaee, Abdullah I. Alharbi, Samar Ahmed, Omar Nacar, Lahouri Ghouti and Anis Koubaa

17:00–17:05 ***Workshop Closing***

AraTar: A Corpus to Support the Fine-grained Detection of Hate Speech Targets in the Arabic Language

Seham Alghamdi[†,‡], **Youcef Benkhedda**[†], **Basma Alharbi**[◇] **and Riza Batista-Navarro**[†]
[†]Department of Computer Science, The University of Manchester, UK
[‡]Department of Information Systems, University of Jeddah, Saudi Arabia
[◇]Department of Computer Science and Artificial Intelligence, University of Jeddah, Saudi Arabia
{seham.alghamdi, youcef.benkhedda, riza.batista}@manchester.ac.uk, bmalharbi@uj.edu.sa

Abstract

We are currently witnessing a concerning surge in the spread of hate speech across various social media platforms, targeting individuals or groups based on their protected characteristics such as race, religion, nationality and gender. This paper focusses on the detection of hate type (Task 1) and hate target (Task 2) in the Arabic language. To comprehensively address this problem, we have combined and re-annotated hate speech tweets from existing publicly available corpora, resulting in the creation of AraTar, the first and largest Arabic corpus annotated with support for multi-label classification for both hate speech types and target detection with a high inter-annotator agreement. Additionally, we sought to determine the most effective machine learning-based approach for addressing this issue. To achieve this, we compare and evaluate different approaches, including: (1) traditional machine learning-based models, (2) deep learning-based models fed with contextual embeddings, and (3) fine-tuning language models (LMs). Our results demonstrate that fine-tuning LMs, specifically using AraBERTv0.2-twitter (base), achieved the highest performance, with a micro-averaged F1-score of 84.5% and 85.03%, and a macro-averaged F1-score of 77.46% and 73.15%, for Tasks 1 and 2, respectively.

Keywords: Hate speech detection, Arabic language models, Text classification, Annotated corpus

1. Introduction

The widespread propagation of hate speech messages on social media and the anonymity enjoyed by online users who post such messages have had an overwhelming negative impact on those targeted by hate speech (Alsafari et al., 2020a; Aluru et al., 2020). Moreover, hate speech can provoke dangerous reactions and online aggression amongst online users, which, in some cases, can spill over into physical harm to people (Aluru et al., 2020; Abu Farha and Magdy, 2020). Hate speech is defined as discriminating against, or insulting an individual or a group of people based on characteristics such as race, sexual orientation, ethnicity, religion, gender or nationality (ElSherief et al., 2018; Blaya, 2019). In addition to studying and detecting hate speech in general, it is imperative to identify the specific targets of hate speech, e.g., individuals or groups experiencing religious intolerance, racism and misogyny. Natural language processing (NLP) plays a critical role in detecting such content (Waseem and Hovy, 2016).

In this work, we cast Arabic Hate Speech and Target Detection (AHTD) as a text classification problem with two tasks. The first task (Task 1) is detecting hate speech within a message, classifying it according to pre-defined categories which are based on protected characteristics covered by the definition of hate speech: religion-hate (RH), ethnicity-hate (EH), nationality-hate (NH), gender-hate (GH), undefined-hate (UDH)[1] or clean (CL), with the last category pertaining to messages that do not contain hate according to the definition above. This task is considered to be a multi-label classification problem where any number of labels (i.e., the hate categories) can be assigned to a given message. The second task (Task 2) involves identifying the specific target of hate speech according to finer-grained categories under the above-mentioned hate categories. For example, targets for the religion-hate category could be Islam, Christianity or Judaism. This task is considered as a multi-label classification problem, as we cannot assume that every message is directed only towards one target; there are cases when there are multiple targets, hence approaches that assign only one label at a time are insufficient.

Targets are different in each hate category and are defined in this research as the individual or group of people possessing certain protected characteristics who are the subject of hate. The novelty of our work lies in addressing the second task, which thus far has been under-explored with respect to hate speech detection in Arabic. The main contributions[2] of this paper are:

- A new corpus, AraTar, with annotated hate

[1]Pertains to hate types different from RH, EH, NH and GH

[2]Our annotation guidelines, annotations and code are publicly available at https://github.com/SehamAlghamdi/AraTar.

OSACT 2024 Workshop, pages 1–12

types and hate targets, which supports the development of multi-label classification methods for automatically detecting types and targets of hate speech.

- A comparative study conducted to investigate different machine learning-based approaches, including: (1) traditional machine learning-based models, (2) deep learning-based models, and (3) fine-tuning language models (LMs).
- Comparative evaluation of the best performing model on our corpus and on other relevant corpora.

2. Related Work

Despite the abundance of Arabic corpora and approaches proposed for automatic hate speech detection, it is important to note that the number of such resources falls short in comparison to those available in English. While several efforts have been made to develop corpora and detection methods for Arabic hate speech, they primarily focus on distinguishing between hate and non-hate categories, or differentiating hate speech from offensive and abusive language. The development of resources specifically focussing on fine-grained hate speech detection and hate target identification remains limited.

Hate Type Detection (Task 1). Upon conducting a careful literature search, we noted that the majority of the corpora reported in the literature concentrated on detecting hate speech types and formalising the problem as either a multi-class classification problem whereby one out of multiple possible hate types is identified (Mubarak et al., 2023; Duwairi et al., 2021; Alsafari et al., 2020b; Al-Hassan and Al-Dossari, 2022; Anezi, 2022; Yadav et al., 2023), or a binary classification problem focussing on detecting whether a given input text contains a specific type of hate speech or not, e.g., religious hate (Albadi et al., 2018) and ethnicity hate (Alotaibi and Abul Hasanat, 2020). Only one study (Azzi and Zribi, 2022) developed a corpus and approaches compatible with multi-label classification, achieving a 79% micro-averaged F1-score. Seven classes were defined in their corpus to detect racism, sexism, religious hatred, xenophobia, violence, hate, pornography and LGBTQ hate (Azzi and Zribi, 2022).

Hate Target Identification (Task 2). A few studies have investigated the detection of specific targets of hate speech. Aref et al. (2020) and Alraddadi and Ghembaza (2021), for instance, focussed on anti-Islam or Islamophobic speech. They achieved varying levels of performance: F1-scores of 52% and 97% on their SSIT corpus and anti-Islamic corpus, respectively. In another work, the detection of anti-immigrant speech was explored by Mohdeb et al. (2022), obtaining an F1-score of 57% based on their own RED corpus. Speech containing sentiment against women (i.e., misogyny) was investigated in the Arabic Misogyny Identification (ArMI) shared task (Mulki and Ghanem, 2021). Six participating teams used the ArMI corpus, with the highest ranked team achieving a 91% macro-averaged F1-score (Mahdaouy et al., 2022). In a similar vein, the study by Guellil et al. (2022) focussed on women as hate targets, making use of their own Arabic_fr_en corpus. They obtained a macro-averaged F1-score of 86%. It is worth noting that all these studies formalised the detection of hate target as a binary classification problem.

We also noted common limitations among the existing corpora mentioned above. Firstly, the majority of them do not support multi-label classification, dealing with mutually exclusive classes only, thus ignoring the possibility that messages could pertain to multiple hate types or targets. Secondly, there is no standard labelling scheme for the types or targets of hate; each dataset follows a different set of hate types and targets. Furthermore, these existing corpora focussed on either only one type or one target of hate speech; therefore, there is no benchmark corpus for the task of fine-grained hate speech detection that covers multiple existing types and targets of hate speech in Arabic.

3. Data Collection and Annotation

We collected hate tweets from various available corpora and re-annotated them to facilitate a multi-label setting and to identify hate targets.

3.1. Data Collection

Five available corpora were used in collecting hate tweets, described as follows.

Arabic-Twitter corpus (Alsafari et al., 2020b). This is the first corpus that was constructed while considering the task of detecting different hate types. Specifically, four different hate types were explored: religion, ethnicity, gender and nationality hate, as well as offensive speech. It contains 5,340 tweets collected from Twitter where 1,423 tweets belong to the defined hate types. The tweets were obtained through robust search techniques using keywords, hashtags, user profiles, and phrases that defend groups with protected characteristics (as they are typically posted in response to hate-containing tweets which were retrieved to become part of the corpus). The researchers specifically included tweets written in the Gulf Arabic dialect and Modern Standard Arabic. The corpus was manually

annotated by native Arabic speakers, employing a three-level hierarchical annotation scheme for the binary classification of offensive and hate speech, ternary classification of offensive, hate speech and non-hate speech, and multi-class classification of different types of hate and offensive speech.

OSACT5 shared task corpus (Mubarak et al., 2023). The OSACT5 corpus was developed for the fine-grained hate speech detection shared task, consisting of 12,698 tweets where 1,339 tweets were labelled as containing hate.The tweets were collected from Twitter using an emoji-based method, where emojis that are known to often appear in offensive content were used. The annotation process incorporated a hierarchical annotation scheme to address three distinct sub-tasks: (1) offensiveness detection, treated as a binary classification task (offensive or non-offensive); (2) hate speech detection, also approached as a binary classification task (hate or non-hate); and (3) fine-grained hate detection, treated as a multi-class classification task with seven classes: hate based on nationality, race, and ethnicity, hate based on religion and belief, ideological hate, hate based on disability, hate based on social class, hate based on gender and non-hate speech.The tweets were written in both Modern Standard Arabic and various Arabic dialects and were annotated through crowd-sourcing.

Arabic hate-speech corpus (Al-Hassan and Al-Dossari, 2022). This corpus consists of 11K tweets, with 2,605 tweets labelled as containing hate. It was compiled by curating a list of hashtags associated with topics that are known to trigger hateful content. The annotation scheme employs multi-class classification, assigning one of five distinct classes to each tweet, namely religious hate, racial hate, sexism, general hate and no hate. The initial annotation was conducted by a volunteer, followed by a rigorous review process involving two additional volunteers to ensure the accuracy and consistency of the annotations.

Levantine Hate Speech and Abusive Language Dataset (L-HSAB) (Mulki et al., 2019). This corpus contains 5,846 tweets obtained through the Tweepy API and were written in the Lebanese and Syrian dialects. A lexicon-based approach was used to collect tweets from verified or popular political and social public figures' timelines, focussing on entities associated with hate, such as refugees. The annotated tweets in L-HSAB support multi-class classification, categorised into three classes: normal, hate, and abusive. The annotation was carried out by three annotators, who are Levantine native speakers.

3.2. Data Annotation Task

Our annotation task was carried out by three volunteer annotators, all of whom are native Arabic speakers and pursuing a higher education degree at that time. To ensure annotation quality, several meetings took place with the lead annotator (the first author of this paper who is also a native Arabic speaker) and volunteer annotators during the annotation stage. Firstly, a workshop was held with the annotators to describe the task and the process, including a training exercise with a number of example tweets from each category of both tasks. In the workshop, discussions about confusing and ambiguous cases were held. Then, a pilot study was conducted with the annotation team, who were given annotation guidelines and a set of 300 hate tweets that were previously annotated by the lead annotator. The annotators were asked to independently label the tweets using a hierarchical annotation scheme (described below). Their annotations were compared to those of the lead annotator in order to identify the most consistent annotator and to identify cases of disagreement. These cases were then discussed and clarified, and the annotation guidelines were revised accordingly.

Annotation Process. The process of annotation took six months and involved two stages. In Stage 1, 30% of the hate tweets in the corpus (1541 tweets) were annotated by all three annotators. Then, we evaluated the inter-annotator agreement (IAA) or reliability among the annotators. In Stage 2, the remaining 70% (3594 tweets) was divided among annotators for single annotation, each annotating 1198 tweets independently.

The annotation was performed using spreadsheets designed with drop-down lists that allow for multiple selections to support the annotators in annotating the tweets with one or more types or targets of hate.

Annotation Scheme. A hierarchical scheme formed the basis of the annotation of the corpus, shown in Figure 1. This scheme was designed for Task 1 based on the annotation scheme used in the Arabic Twitter corpus (Alsafari et al., 2020b), but refined to consider annotating one or more types of hate and non-predefined hate types, and extended to annotate targets of hate (Task 2).

In the proposed scheme, targets of hate were defined based on recently published work that highlighted the common targets of Arabic hate speech according to religion, nationality, and gender (Mubarak et al., 2023). For ethnicity hate, a pilot study was conducted on 30 tweets from the combined corpus to identify the most common ethnicity targets. Additionally, in the proposed scheme, the issue of annotating hate tweets that do not belong

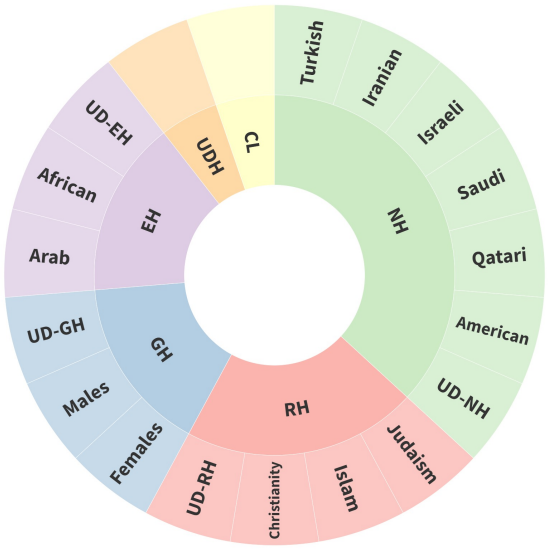


Figure 1: The AraTar Annotation Scheme. Key: RH = religion-hate, EH = ethnicity-hate, NH = nationality-hate, GH = gender hate, UDH = undefined-hate, CL = clean, UD = Undefined.

to the defined types and targets was addressed by defining an undefined-hate (UDH) category and undefined target categories, including undefined-RH (UD-RH), undefined-NH (UD-NH), undefined-EH (UD-EH), and undefined-GH (UD-GH). Figure 1 illustrates our taxonomy, i.e., the hate speech types and target categories in a hierarchical/sunburst form.

Annotation Guidelines. We have developed and validated annotation guidelines to provide our annotators with clear instructions for the tasks. Our annotation guidelines for Task 1 were inspired by the guidelines proposed by Alsafari et al. (2020b). However, we have extended these guidelines to include the annotation of hate types that are not covered in their annotation scheme, as well as the identification of hate targets. Furthermore, our guidelines take into account the annotation of implicit hate: when the type or target of hate is mentioned implicitly, either by using epithets or indirect references to the type or target of hate.

3.3. Annotation Results

As mentioned above, a common set consisting of 30% of the hate tweets in our corpus was independently annotated by the three annotators, thus allowing us to measure inter-annotator agreement (IAA). IAA was calculated using metrics that are suitable for multi-label scenarios such as F1-score (Hripcsak and Rothschild, 2005) and Krippendorff's α (Krippendorff, 1970, 2004), as they consider the distance/difference in annotations across all potential annotation units, regardless of the number of labels or annotators and the nature of annotation (including numeric, categorical and ordinal labels). The results, presented in Table1, show high agreement among the annotators in both Tasks 1 and 2. The average macro-averaged F1-scores are 97.21% and 97.18%, respectively, and the average micro-averaged F1-scores are 98.92% and 98.67% respectively. Similarly, Krippendorff's α is high, i.e., 98.76% in both tasks.

Metrics	Task1	Task2
Avg of Pairwise Macro-F1	97.21	97.18
Avg of Pairwise Micro-F1	98.92	98.67
Krippendorff's α	98.76	98.76

Table 1: IAA for Hate Type Detection (Task 1) and Hate Target Identification (Task 2).

Conflicting cases between the annotators were resolved by the lead annotator. At the end of the annotation process, 6124 tweets were added to the corpus for the clean (CL) category, drawn from the offensive and clean categories of the OSACT5 corpus. Additionally, 81 tweets from different datasets were manually labelled as CL, as closer inspection showed that they did not contain hate speech. Furthermore, 40 tweets were deleted due to duplication. The total number of tweets in AraTar is 11,219, spanning Modern Standard Arabic and a number of dialects including Gulf and Levantine. Figure 2 shows the label distribution according to hate type and hate target. Notably, in AraTar, 7% and 10% of hate tweets were annotated with more than one type of hate and more than one hate target, respectively.

4. Methodology

Upon completion of the annotation of the AraTar corpus, we set out to determine the performance of various classification models on Tasks 1 and 2. In this section, we describe the steps that we took towards this goal, including pre-processing of the tweets in the corpus, selection and design of three different types of classification models, and experimentation with the said models.

4.1. Data pre-processing

To prepare the corpus for analysis, we applied the following text pre-processing steps: removing diacritics, punctuation, repeated characters, symbols, special characters, URLs, English tokens and emojis to reduce noise, and performing letter normalisation by converting the forms of three letters into one form: Alif (أ ، آ ، إ to ا), Hamza (ؤ ، ئ to و ، ى) and Ta Marbouta (ة to ه). Next, the AraTar dataset was

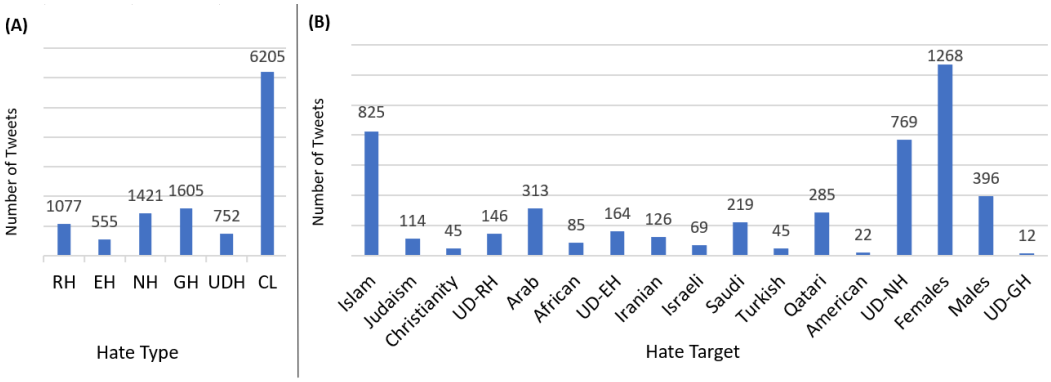


Figure 2: Label distribution in AraTar for the Hate Type Detection (A) and Hate Target Identification (B) tasks.

split using stratified sampling into three subsets: training, validation and test sets with proportions of 70%, 15% and 15% respectively.

4.2. Approaches and Models

We investigated the following machine learning-based approaches on our two tasks.

(1) Traditional Machine Learning-based Approach. We investigate support vector machine (SVM) models (Cortes and Vapnik, 1995) trained on features based on term frequency-inverse document frequency (TF-IDF) (Sparck Jones, 1972). In previous work, satisfactory results were achieved by employing SVM models fed with TF-IDF features, both in detecting hate type (Al-Hassan and Al-Dossari, 2022; Azzi and Zribi, 2022) and detecting hate target as a binary classification problem (Alraddadi and Ghembaza, 2021; Aref et al., 2020).

(2) Deep learning-based Model fed with Contextual Embeddings. We used a Long Short Term Memory (LSTM) model initialised with AraBERTv02-twitter embeddings. Apart from the ability of LSTM models to learn long-term dependencies between words, it has also proven its robustness in capturing and identifying multiple types of hate categories in the work of Al-Hassan and Al-Dossari (2022). LSTMs models have also shown good performance in the experiments conducted by Al-Hassan and Al-Dossari (2022) and Alsafari et al. (2020b) compared to other deep learning algorithms. Furthermore, the work of Alsafari et al. (2020b) demonstrated that the use of contextual word embeddings in LSTMs yields superior results compared to LSTM models with static word embeddings such as fastText and AraVec.

In our own work, we used the contextual word embeddings from AraBERTv02-twitter (base), a model variant of AraBERT (Antoun et al., 2020) that supports dialectal Arabic and is trained on Arabic tweets, as our data is from the Twitter platform. These embeddings were then fed as features to an LSTM model that was built upon the architecture proposed by Alsafari et al. (2020b).

(3) Fine-tuned Language Models. We used state-of-art transformer-based Arabic language models (LMs), namely, MARBERTV2 (Abdul-Mageed et al., 2021) and AraBERTv02-twitter (base and large) which are variants of AraBERTv2 (Antoun et al., 2020) since our data is from Twitter. These language models were used as they were pre-trained on dialectal Arabic tweets and are thus best-suited LMs for the downstream task of Arabic hate speech detection. In addition to that, models based on fine-tuning MARABERTv2 and AraBERTv2 achieved state-of-art results on hate speech detection cast as multi-class classification (AlKhamissi and Diab, 2022; Althobaiti, 2022; Bennessir et al., 2022; Shapiro et al., 2022) and binary classification (Abbes et al., 2021; Mahdaouy et al., 2022; Messaoudi et al., 2021; Mohdeb et al., 2022; Nwesri et al., 2021). On top of each pre-trained LM, we added a linear layer which computes a probability distribution based on the possible classes in the task at hand, i.e., either of Task 1 and Task 2.

4.3. Experimental Setup

Experiments on AraTar. We used identical training, validation and test sets across all five models: SVM, LSTM, MARBERT, AraBERT-base (AraBERT-b) and AraBERT-large (AraBERT-l). For the last four models, we employed the following hyperparameter settings: a maximum sequence length of 90, which considers the maximum sequence length in the corpus; the Adam optimiser; a learning rate of 5×10^{-5}; training for 50 epochs with early stopping based on validation loss; and for the fine-tuning of language models we used 16 as the batch size and binary cross-entropy as the loss function.

As mentioned above, our LSTM model was adopted from the architecture and implementations used in the study by Alsafari et al. (2020b). However, it was instead fed with contextual word embeddings, specifically AraBERTv02-twitter (both base and large variants) and evaluated on our corpus. The obtained results were disappointing, with low micro-averaged F1-scores of 25% and 2% for Task 1, and 43% and 11% for Task 2, using the

base and large models respectively. The macro-averaged F1-scores were also low, at 22% and 1% for Task 1, and 11% and 10% for Task 2 with the base and large models respectively. We thus optimised the hyperparameters used in training the LSTM model. Specifically, we set dropout and recurrent dropout to 0.2, and set batch size to 32.

Experiments on Other Corpora. To assess the performance of our top-performing model on other datasets, we conducted further fine-tuning using the OSACT5 (Mubarak et al., 2023) and Arabic Twitter datasets (Alsafari et al., 2020b), which were described in Section 3.1. The rationale behind choosing OSCAT5 and the Arabic Twitter dataset for comparison lies in their unique attributes. OSCAT5 stands out as the current benchmark corpus in the field, while the developed models using the Arabic Twitter dataset have demonstrated superior performance in previous literature. Furthermore, both datasets offer readily available training and test sets, ensuring the comparability of our experiments. It is worth noting that there are currently no other available corpora specifically focussed on the types of hate speech. Our experiments were conducted using the complete datasets and the original training and test sets provided by the authors. We however excluded Disability hate from OSACT5 due to its limited representation, with only two tweets in the entire corpus. Table 2 summarises the class frequencies in both datasets.

Arabic Twitter		OSACT5	
Classes	**Count**	**Classes**	**Count**
RH	321	Race-HS1	366
EH	382	Religion-HS2	38
NH	368	Ideology-HS3	190
GH	352	Social Class-HS5	101
OFF	437	Gender-HS6	641
Clean	3480	NOT_HS	11359
Total	**5340**	**Total**	**12695**

Table 2: Frequencies of hate types in the Arabic Twitter and OSACT5 corpora.

Additionally, since these corpora have a maximum sequence length close to that in AraTar, we kept the same hyperparameter value for model training to maintain consistency. We also used the same values as before, for the rest of the hyperparameters.[3]

Evaluation Metrics. Following standard practices, we calculated the precision, recall and F1-score to evaluate the performance of the classification models. Additionally, we report the exact match ratio metric (EMR) in our experiments on AraTar, which is commonly used for multi-label scenarios to measure the proportion of predicted outputs that exactly match the ground truth.

5. Evaluation Results

Tables 3 and 4 present the evaluation results in terms of F1-score (F1) for each label and model for Tasks 1 and 2, respectively.[4] Additionally, we provide the combined performance of each model in terms of micro-averaged F1-score (micro-F1), macro-averaged F1-score (macro-F1) and EMR. From the obtained results, it is noticeable that overall, the fine-tuned LMs, particularly the models that use AraBERTv2-twitter (i.e., AraBERT-b and AraBERT-l) obtained superior performance over the SVM and LSTM models in both tasks.

Classes	**SVM**	**LSTM**	**MARBERT**	**AraBERT-b**	**AraBERT-l**
	F1	F1	F1	F1	F1
RH	76.55	82.72	85.29	**86.15**	82.43
EH	63.16	68.70	**79.76**	79.53	78.69
NH	65.16	67.18	75.26	79.07	**80.00**
GH	72.98	77.10	**79.09**	76.28	78.75
UDH	27.27	36.16	48.09	**51.98**	39.53
CL	86.08	87.44	90.10	**91.76**	90.04
Micro-F1	77.78	79.37	83.55	**84.50**	83.56
Macro-F1	65.20	69.88	76.26	**77.46**	74.91
EMR	70.19	72.62	74.26	**81.83**	55.29

Table 3: Evaluation Results for Hate Type Detection (Task 1).

Classes	**SVM**	**LSTM**	**MARBERT**	**AraBERT-b**	**AraBERT-l**
	F1	F1	F1	F1	F1
Islam	82.16	90.91	90.27	90.20	**91.25**
Judaism	48.00	72.73	72.73	**75.86**	74.07
Christianity	22.22	**66.67**	00.00	**66.67**	54.55
UD-RH	23.53	74.07	69.23	**75.00**	71.43
Arab	65.75	79.52	**83.15**	78.65	79.55
African	64.00	69.23	78.57	**90.32**	86.67
UD-EH	48.48	80.00	84.44	82.93	**88.37**
Iranian	34.48	64.71	76.60	**80.95**	76.92
Israeli	00.00	50.00	**61.54**	55.56	50.00
Saudi	26.09	68.66	70.27	**72.46**	72.22
Turkish	00.00	71.43	61.54	**88.89**	87.50
Qatari	69.33	79.52	82.76	**91.67**	89.13
American	00.00	00.00	00.00	**40.00**	00.00
UD-NH	63.58	**85.45**	83.25	84.40	83.65
Females	84.42	**93.44**	91.78	90.34	92.91
Males	62.22	78.10	77.69	79.67	**85.71**
UD-GH	00.00	00.00	00.00	00.00	00.00
Micro-F1	68.87	84.20	83.66	85.03	**86.05**
Macro-F1	40.84	66.14	63.75	**73.15**	69.64
EMR	53.49	77.36	74.26	**77.52**	72.56

Table 4: Evaluation Results for Hate Target Identification (Task 2).

Hate Type Detection (Task 1). The results in Table 3 show that AraBERT-b obtained the highest micro-averaged F1-score of 84.50%, followed by AraBERT-l which obtained 83.56%. Notably, AraBERT-b consistently outperformed other models in Task 1, according to the three metrics for combined performance (micro-F1, macro-F1 and EMR) with a significant margin in terms of EMR.

[3] Implementation details including the hardware and software frameworks that were used in our experiments are provided in Appendix B.

[4] Precision and recall values are reported in Appendix C.

The highest score for EMR in Task 1 is 81.83% (AraBERT-b) followed by 74.26% (MARBERT), resulting in an improvement of 7.5 percentage points. This indicates that AraBERT-b is the most dependable model for accurately identifying various forms of hate in Arabic tweets.

AraBERT-b displayed superior performance in accurately classifying religious hate, undefined hate and clean (CL) categories compared to the other models. However, ethnicity and gender hate were better identified by MARBERT by a margin of 0.23 and 2.81 percentage points, respectively, compared to AraBERT-b.

It is also worth noting that, although not directly comparable, our best model (AraBERT-b) outperformed the model proposed by Azzi and Zribi (2022). Even though a direct comparison might not be entirely apt, their model, often regarded as the state-of-the-art in the literature, obtained a micro-averaged F1-score of 79%. In contrast, our model achieved 84.50% for the same metric. Moreover, to best of our knowledge, in terms of macro-averaged F1-score, AraBERT-b achieved a higher score compared to the majority of the existing models reported in the literature (Alsafari et al., 2020a,b; Al-Hassan and Al-Dossari, 2022; Duwairi et al., 2021; Althobaiti, 2022; Bennessir et al., 2022; Magnossão de Paula et al., 2022; Shapiro et al., 2022; AlKhamissi and Diab, 2022; Albadi et al., 2018, 2019).

Hate Target Identification (Task 2). The best performance was obtained by AraBERT-l, with a micro-averaged F1-score of 86.05%, gaining a 1 percentage point improvement over AraBERT-b which obtained 85.03% on the same metric. For the remaining two overall metrics, AraBERT-b achieved the highest scores of 73.15% and 77.52% in terms of macro-averaged F1-score and EMR. For these two metrics, the next best scores were 69.64% (AraBERT-l) and 77.36% (LSTM). This indicates that AraBERT-b obtained a 4 and 0.16 percentage point improvement on macro-averaged F1-score and EMR, respectively. AraBERT-b demonstrated superior performance in learning nine hate targets: Judaism, Christianity, undefined religious targets, African, Iranian, Saudi, Turkish, Qatari and American. In contrast, for categories like Islam, undefined ethnicity and males, AraBERT-l outperformed AraBERT-b, with improvements of 1.25, 5.44 and 6 percentage points, respectively. For other categories such as Arab, Israeli, undefined nationality and females, either MARBERT or LSTM proved to be superior, showing gains of 4.5, 5.98, 1.05 and 3.1 percentage points over AraBERT-b, respectively.

A notable limitation of the classification models is their difficulty in accurately identifying undefined gender hate targets. AraBERT-b, along with all other models, did not effectively learn to identify this target. This could be attributed to the low number of samples in the training set.

Furthermore, when comparing our best model (AraBERT-b) with those reported in the literature, there is an absence of reporting the micro-averaged F1-score of the published models and a lack of studies that have developed generalised detection models that consider different targets in a multi-label classification task. However, when looking at the macro-averaged F1-score, AraBERT-b performed lower than the majority of published models. This may be attributed to the imbalanced distribution in our dataset and the more complex nature of the multi-label classification task compared to binary classification.

6. Discussion

Error Analysis. We conducted error analysis by inspecting some of the misclassified cases produced by the best model in each task. A total number of 306 samples and 145 samples were misclassified in Tasks 1 and 2, respectively, with 82 overlapping samples. We have four main observations, outlined below.

Disclaimer: Due to the nature of this work, our examples contain hate speech which some readers might find offensive. These do not in any way reflect the researchers' own views or opinions.

(1) Mention of hate targets in a neutral context might mislead the trained classifier: We identified instances where mentions of potential hate targets were used in a neutral context, thus misleading the classifier. For instance, "Houthis" in the tweet *"By God, show us at the borders with the Houthis, O Mas'ood. Two states are with you. Seriously, they are besieged and you couldn't handle them, O'Utaibi, O effeminate"*

واله ورينا في الحدود مع الحوثي يا مسعود ٢ دوله معاكم
علي جماعه محاصره وما قدرتوا عليهم يا عتيبي يا خنيث

(2) Implicit hate: We recognised that in some cases the classification model fails to detect implicit hate, as in the following post with implicit gender hate towards women. *"O [vomiting emoji], Do not believe themselves, butterflies and dancers"*

يع لايصدقون نفسهم قسم الفراشات والغوازي

In the context of this tweet, "butterflies" and "dancers" are allegorically used to refer to women. Such coded language presents challenges for our classifier due to its inherent subtlety. This inability to predict such coded language can be addressed by employing a dataset that captures many examples of such cases.

At times, epithets are mentioned in the content that refers to a hate target. An example is the post: *"I'm tired from cursing and insulting the Be*uins.*

AraTar		Arabic Twitter		OSACT5	
Classes	**F1**	**Classes**	**F1**	**Classes**	**F1**
RH	86.15	RH	81.32	Race-HS1	43.24
EH	79.53	EH	82.03	Religion-HS2	0.00
NH	79.07	NH	83.70	Ideology-HS3	21.62
GH	76.28	GH	76.62	Social Class-HS5	0.00
UDH	51.98	OFF	80.32	Gender-HS6	64.20
CL	91.76	Clean	94.83	NOT_HS	96.16
Micro-F1	84.50	Micro-F1	91.14	Micro-F1	92.44
Macro-F1	77.46	Macro-F1	83.14	Macro-F1	37.54

Table 5: Results of AraBERT-b on the AraTar, Arabic Twitter and OSCAT5 corpora.

They don't know that this sound could explode a child's ear and make them deaf due to this ignorance. Please, Mohammed, find a solution to this drifting"

تعبت وانا العن و اسب الب* و مايدرون ان هالصوت ممكن يفجر اذن الطفل و يصير اصم بسبب هالتخلف تكفى يامحمد شف حل للطعوس

The tweet combines a personal feeling of exhaustion with a negative generalisation about the Bedouins, suggesting ignorance. However, it does not mention any explicit derogatory terms. It mentions "drifting", an epithet used for Bedouins. It is worth noting that the use of an asterisk (*) to mask some characters in the word *"Bedouins"* was likely a means for avoiding detection by Twitter's automatic moderation tools.

(3) Correlation between less presented subtargets and contents might lead to misclassification: For instance, the Islam hate target, Houthi (an extremist Islamic group), was misclassified as nationality hate and as an undefined nationality hate target in Tasks 1 and 2, respectively. We can interpret the reason for this behaviour as the correlation between Houthi and Yaman in the content.

(4) Offensive tweets that were predicted as having a hate type: For example, the following offensive tweet was classified as gender hate: *"We are on time, has many frivolous people, they are disgusting [face with medical mask emoji]."*

نحنُ في زمن كَثُر فيه الخفيفون و الخفيفات، مثيرين للاشمئزاز

Comparison with Other Corpora. Table 5 presents the results of applying AraBERT-b on the other corpora with multi-class classification settings. The motivation for conducting this comparison is two-fold:

(1) Highlight the extent to which AraTar can enable a model to learn the hate type classification task. Upon closer examination of the F1-score for each label, it becomes apparent that performance on AraTar is better than on OSACT5. Moreover, it is noticeable that performance on the social hate and religion hate classes is 0. This can be attributed to their under-representation in OSACT5, making it challenging for the model to learn and generalise effectively to these specific classes. Furthermore, even though the Arabic Twitter corpus has a more balanced distribution, AraTar was able to provide comparable results.

The reason for the reduction in the overall performance on AraTar is the score for the UDH class which is one of the minority classes and has a diversity of tweets that convey different hate types that do not belong to the other categories. These empirical findings lead to the conclusion that classification based on AraTar yields satisfactory results although UDH is difficult to detect. Unlike the Arabic Twitter dataset that supports the detection of only one hate type at a time, AraTar supports the detection of messages containing general hate as well as any number of defined hate types where they exist.

(2) Assess whether our best performing model (AraBERT-b) obtains competitive performance on multi-class classification of hate type, when compared with the state-of-the-art models previously reported for the other corpora. The obtained results demonstrate a significant improvement in the detection performance on the Arabic Twitter dataset achieved by AraBERT-b in terms of the reported macro-averaged F1-score of state-of-art models. Alsafari et al. (2020a) used the Arabic Twitter dataset and achieved the highest macro-averaged F1-score at 80.23% using an ensemble model that employed the BiLSTM architecture with AraBERTv1 embeddings and applying the average value method for aggregation. Our model outperformed this performance by 2.91 percentage points. However, AraBERT-b exhibits a 15.3 percentage point decrease when compared to the state-of-the-art model on OSACT5, which achieved a macro-averaged F1 score of 52.8. This model, which ranked first in the OSACT5 competition (Mubarak et al., 2022), was designed using multi-task learning techniques. Specifically, its architecture consists of a hard parameter-sharing layer composed of AraBERTv2 contextualised text representation models and subtask-specific layers. These subtask-specific layers were fine-tuned using quasi-recurrent neural networks (QRNNs) for each subtask. The model was trained on two tasks: the detection of offensive speech and general hate speech (Magnossão de Paula et al., 2022).

7. Conclusion

We present AraTar, a corpus to support the fine-grained detection of Arabic hate speech targets. It addresses the previously limited scale of Arabic hate speech detection and the lack of unified annotation in previous datasets. Our experiments show that fine-tuning language models, especially AraBERTv2-twitter, yields favourable results for both the Hate Type Detection and Hate Target

Identification tasks. An AraBERT model trained on AraTar also fares well in comparison with the same model architecture trained on other corpora.

Limitations

The main limitations of AraTar lie in the fact that not all Arabic dialects are covered, and that the corpus is confined to tweets. Furthermore, specific targets are under-represented, thus affecting classification performance for these targets. Future work will focus on broadening the scope of the corpus to include diverse dialects and platforms, and on employing data augmentation methods to generate synthetic data to improve the representation of minority hate targets. Another future direction is the enhancement of the capability of models in detecting hate targets by developing a stronger model using techniques such as parameter-efficient tuning (Yang et al., 2022) or ensemble methods as described in the study by Alsafari and Sadaoui (2021).

Bibliographical References

Istabrak Abbes, Eya Nakache, and Moez BenHajHmida. 2021. Context-aware language modeling for arabic misogyny identification. In *Working Notes of the 2021 Forum for Information Retrieval Evaluation (FIRE 2021)*, pages 847–851.

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. Association for Computational Linguistics.

Ibrahim Abu Farha and Walid Magdy. 2020. Multitask Learning for Arabic Offensive Language and Hate-Speech Detection. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 86–90, Marseille, France. European Language Resource Association.

Areej Al-Hassan and Hmood Al-Dossari. 2022. Detection of hate speech in Arabic tweets using deep learning. *Multimedia systems*, 28(6):1963–1974.

Nuha Albadi, Maram Kurdi, and Shivakant Mishra. 2018. Are they Our Brothers? Analysis and Detection of Religious Hate Speech in the Arabic Twittersphere. In *Proceedings of the 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 69–76.

Nuha Albadi, Maram Kurdi, and Shivakant Mishra. 2019. Investigating the effect of combining GRU neural networks with handcrafted features for religious hatred detection on Arabic Twitter space. *Social Network Analysis and Mining*, 9(1):41.

Badr AlKhamissi and Mona Diab. 2022. Meta AI at Arabic Hate Speech 2022: MultiTask Learning with Self-Correction for Hate Speech Classification. In *Proceedings of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools with Shared Tasks on Qur'an QA and Fine-Grained Hate Speech Detection*, pages 186–193, Marseille, France. European Language Resources Association.

Afaf Alotaibi and Mozaherul Hoque Abul Hasanat. 2020. Racism Detection in Twitter Using Deep Learning and Text Mining Techniques for the Arabic Language. In *Proceedings of the 1st International Conference of Smart Systems and Emerging Technologies (SMARTTECH)*, pages 161–164.

Rawan Abdullah Alraddadi and Moulay Ibrahim El-Khalil Ghembaza. 2021. Anti-Islamic Arabic Text Categorization using Text Mining and Sentiment Analysis Techniques. *International Journal of Advanced Computer Science and Applications*, 12(8).

Safa Alsafari and Samira Sadaoui. 2021. Ensemble-based semi-supervised learning for hate speech detection. In *The International FLAIRS Conference Proceedings*, volume 34.

Safa Alsafari, Samira Sadaoui, and Malek Mouhoub. 2020a. Deep Learning Ensembles for Hate Speech Detection. In *Proceedings of the 32nd IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 526–531.

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

Maha Jarallah Althobaiti. 2022. BERT-based Approach to Arabic Hate Speech and Offensive Language Detection in Twitter: Exploiting Emojis and Sentiment Analysis. *International Journal of Advanced Computer Science and Applications*, 13(5).

Sai Saketh Aluru, Binny Mathew, Punyajoy Saha, and Animesh Mukherjee. 2020. Deep learning models for multilingual hate speech detection. *arXiv preprint arXiv:2004.06465*.

Faisal Yousif Al Anezi. 2022. Arabic Hate Speech Detection Using Deep Recurrent Neural Networks. *Applied Sciences*, 12(12):6010.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. AraBERT: Transformer-based Model for Arabic Language Understanding. 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.

Abdullah Aref, Rana Husni Al Mahmoud, Khaled Taha, and Mahmoud Al-Sharif. 2020. Hate Speech Detection of Arabic Shorttext. In *Proceedings of the 9th International Conference on Information Technology Convergence and Services (ITCSE 2020)*, pages 81–94. AIRCC Publishing Corporation.

Salma Azzi and Chiraz Zribi. 2022. Comparing Deep Learning Models for Multi-label Classification of Arabic Abusive Texts in Social Media. In *Proceedings of the 17th International Conference on Software Technologies*, pages 374–381, Lisbon, Portugal. SCITEPRESS - Science and Technology Publications.

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 *Proceedings of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools with Shared Tasks on Qur'an QA and Fine-Grained Hate Speech Detection*, pages 176–180, Marseille, France. European Language Resources Association.

Catherine Blaya. 2019. Cyberhate: A review and content analysis of intervention strategies. *Aggression and Violent Behavior*, 45:163–172.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning*, 20(3):273–297.

Rehab Duwairi, Amena Hayajneh, and Muhannad Quwaider. 2021. A Deep Learning Framework for Automatic Detection of Hate Speech Embedded in Arabic Tweets. *Arabian Journal for Science and Engineering*, 46(4):4001–4014.

Mai ElSherief, Vivek Kulkarni, Dana Nguyen, William Yang Wang, and Elizabeth Belding. 2018. Hate Lingo: A Target-Based Linguistic Analysis of Hate Speech in Social Media. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 12.

Imane Guellil, Ahsan Adeel, Faical Azouaou, Mohamed Boubred, Yousra Houichi, and Akram Abdelhaq Moumna. 2022. Ara-Women-Hate: An Annotated Corpus Dedicated to Hate Speech Detection against Women in the Arabic Community. In *Proceedings of the Workshop on Dataset Creation for Lower-Resourced Languages within the 13th Language Resources and Evaluation Conference*, pages 68–75, Marseille, France. European Language Resources Association.

George Hripcsak and Adam S. Rothschild. 2005. Agreement, the F-Measure, and Reliability in Information Retrieval. *Journal of the American Medical Informatics Association: JAMIA*, 12(3):296–298.

Klaus Krippendorff. 1970. Bivariate agreement coefficients for reliability of data. *Sociological Methodology*, 2:139–150.

Klaus Krippendorff. 2004. Measuring the reliability of qualitative text analysis data. *Quality and quantity*, 38:787–800.

Angel Felipe Magnossão de Paula, Paolo Rosso, Imene Bensalem, and Wajdi Zaghouani. 2022. UPV at the Arabic Hate Speech 2022 Shared Task: Offensive Language and Hate Speech Detection using Transformers and Ensemble Models. In *Proceedings of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools with Shared Tasks on Qur'an QA and Fine-Grained Hate Speech Detection*, pages 181–185, Marseille, France. European Language Resources Association.

Abdelkader El Mahdaouy, Abdellah El Mekki, Ahmed Oumar, Hajar Mousannif, and Ismail Berrada. 2022. Deep Multi-Task Models for Misogyny Identification and Categorization on Arabic Social Media. *arXiv preprint arXiv:2206.08407*.

Abir Messaoudi, Chayma Fourati, Mayssa Kchaou, and Hatem Haddad. 2021. iCompass Working Notes for Arabic Misogyny Identification. In *Working Notes of the 2021 Forum for Information Retrieval Evaluation (FIRE 2021)*, page 5.

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.

Hamdy Mubarak, Hend Al-Khalifa, and Abdulmohsen Al-Thubaity. 2022. Overview of OSACT5 shared task on Arabic offensive language and hate speech detection. In *Proceedings of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools with Shared Tasks on Qur'an QA and Fine-Grained Hate Speech Detection*, pages 162–166, Marseille, France. European Language Resources Association.

Hamdy Mubarak, Sabit Hassan, and Shammur Absar Chowdhury. 2023. Emojis as anchors to detect Arabic offensive language and hate speech. *Natural Language Engineering*, 29(6):1436–1457.

Hala Mulki and Bilal Ghanem. 2021. ArMI at FIRE 2021: Overview of the First Shared Task on Arabic Misogyny Identification. In *Working Notes of the 2021 Forum for Information Retrieval Evaluation (FIRE 2021)*, pages 820–830.

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 3rd Workshop on Abusive Language Online*, pages 111–118, Florence, Italy. Association for Computational Linguistics.

Abdusalam Nwesri, Stephen Wu, and Harmain Harmain. 2021. Detecting Misogyny in Arabic Tweets. In *Working Notes of the 2021 Forum for Information Retrieval Evaluation (FIRE 2021)*, page 6.

Ahmad Shapiro, Ayman Khalafallah, and Marwan Torki. 2022. AlexU-AIC at Arabic Hate Speech 2022: Contrast to Classify. In *Proceedings of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools with Shared Tasks on Qur'an QA and Fine-Grained Hate Speech Detection*, pages 200–208, Marseille, France. European Language Resources Association.

Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21.

Zeerak Waseem and Dirk Hovy. 2016. Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter. In *Proceedings of the NAACL Student Research Workshop*, pages 88–93, San Diego, California. Association for Computational Linguistics.

Ankit Yadav, Shubham Chandel, Sushant Chatufale, and Anil Bandhakavi. 2023. LAHM: Large Annotated Dataset for Multi-Domain and Multilingual Hate Speech Identification. *arXiv preprint arXiv:2304.00913*.

Zhuoyi Yang, Ming Ding, Yanhui Guo, Qingsong Lv, and Jie Tang. 2022. Parameter-Efficient Tuning Makes a Good Classification Head. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7576–7586. Association for Computational Linguistics.

Appendix

A. Annotation Guidelines

The annotation guidelines can be downloaded from our Github repository.[5]

B. Implementation Details

B.1. Hardware

For both Tasks 1 and 2, we ran the SVM and LSTM experiments on a single Tesla V100 GPU with 51 GB RAM using the Google Colab Pro+ platform.[6] Also, we used a single NVIDIA A100 with 84 GB RAM to run the fine-tuning experiments with MARBERT, AraBERT-b and AraBERT-l.

B.2. Software Frameworks

Python 3.10.12 was used in implementing all models and experiments. Different machine learning frameworks were used. Firstly, the scikit-learn toolkit[7] was used in developing the SVM model. Additionally, we employed the skmultilearn library[8] which applies the binary relevance technique to a multi-label classification problem. For our LSTM model, Keras[9] was used. Lastly, we utilised Hugging Face's Transformers library[10] to fine-tune the pre-trained MARBERTv2 and AraBERT-twitter (base and large) language models for our multi-label classification tasks. Specifically, we loaded them and built our models using the `AutoModelForSequenceClassification` class, leveraging Hugging Face's Trainer API.

For evaluation, we used the metrics implemented in the scikit-learn toolkit.

For reproducibility, we set the seed parameter to 42 in all AraTar experiments.

[5] https://github.com/SehamAlghamdi/AraTar
[6] https://colab.research.google.com/
[7] https://scikit-learn.org/stable/
[8] http://scikit.ml/
[9] https://keras.io/
[10] https://huggingface.co/docs/transformers/index

C. Detailed Results

For both Tasks 1 and 2, we report the results of a single run trained for 50 epochs with early stopping based on validation loss. Tables 6, 7 and 8 present detailed results, including precision and recall scores, to complement Tables 3, 4 and 5 in the paper.

	SVM			**LSTM**			**MARBERT**			**AraBERT-b**			**AraBERT-l**		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
RH	**86.05**	68.94	76.55	82.21	83.23	82.72	82.56	**88.20**	85.29	85.37	86.96	**86.15**	84.87	80.12	82.43
EH	85.71	50.00	63.16	**95.74**	53.57	68.70	79.76	79.76	**79.76**	78.16	80.95	79.53	72.73	**85.71**	78.69
NH	83.33	53.49	65.16	74.86	60.93	67.18	**84.39**	67.91	75.26	79.07	**79.07**	79.07	80.95	**79.07**	**80.00**
GH	82.29	65.56	72.98	72.96	**81.74**	77.10	78.93	79.25	**79.09**	**86.77**	68.05	76.28	85.44	73.03	78.75
UDH	**75.00**	16.67	27.27	46.38	29.63	36.16	58.67	40.74	48.09	49.58	**54.63**	**51.98**	53.12	31.48	39.53
CL	85.81	86.36	86.08	**95.31**	80.77	87.44	92.33	87.97	90.10	91.46	92.05	**91.76**	87.06	**93.23**	90.04
Micro	84.96	71.72	77.78	85.33	74.20	79.37	**86.28**	80.98	83.55	85.21	**83.79**	**84.50**	83.85	83.28	83.56
Macro	**83.03**	56.84	65.20	77.91	64.98	69.88	79.44	73.97	76.26	78.40	**76.95**	**77.46**	77.36	73.78	74.91

Table 6: Complete Results for Hate Type Detection (Task 1).

	SVM			**LSTM**			**MARBERT**			**AraBERT-b**			**AraBERT-l**		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Islam	90.83	75.00	82.16	90.91	**90.91**	90.91	92.80	87.88	90.27	**93.50**	87.12	90.20	91.60	**90.91**	**91.25**
Judaism	75.00	35.29	48.00	75.00	70.59	72.73	75.00	70.59	72.73	91.67	64.71	**75.86**	**1.00**	58.82	74.07
Christian.	**1.00**	12.50	22.22	**1.00**	**50.00**	**66.67**	00.00	00.00	00.00	**1.00**	**50.00**	**66.67**	**1.00**	37.50	54.55
UD-RH	66.67	14.29	23.53	**76.92**	71.43	74.07	75.00	64.29	69.23	66.67	**85.71**	**75.00**	71.43	71.43	71.43
Arab	85.71	53.33	65.75	**86.84**	73.33	79.52	84.09	**82.22**	**83.15**	79.55	77.78	78.65	81.40	77.78	79.55
African	**1.00**	47.06	64.00	**1.00**	52.94	69.23	**1.00**	64.71	78.57	**1.00**	**82.35**	**90.32**	**1.00**	76.47	86.67
UD-EH	88.89	33.33	48.48	**1.00**	66.67	80.00	90.48	**79.17**	84.44	**1.00**	70.83	82.93	**1.00**	**79.17**	**88.37**
Iranian	55.56	25.00	34.48	78.57	55.00	64.71	66.67	**90.00**	76.60	77.27	85.00	**80.95**	**78.95**	75.00	76.92
Israeli	00.00	00.00	00.00	75.00	37.50	50.00	**80.00**	50.00	**61.54**	50.00	**62.50**	55.56	75.00	37.50	50.00
Saudi	75.00	15.79	26.09	79.31	60.53	68.66	72.22	**68.42**	70.27	**80.65**	65.79	**72.46**	76.47	**68.42**	72.22
Turkish	00.00	00.00	00.00	**1.00**	55.56	71.43	**1.00**	44.44	61.54	88.89	**88.89**	**88.89**	**1.00**	77.78	87.50
Qatari	**89.66**	56.52	69.33	89.19	71.74	79.52	87.80	78.26	82.76	88.00	**95.65**	**91.67**	89.13	89.13	89.13
Amer.	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	50.00	33.33	**40.00**	00.00	00.00	00.00
UD-NH	87.30	50.00	63.58	88.35	**82.73**	**85.45**	87.88	79.09	83.25	85.19	83.64	84.40	**88.78**	79.09	83.65
Females	89.22	80.11	84.42	**91.28**	**95.70**	**93.44**	90.58	93.01	91.78	87.82	93.01	90.34	90.77	95.16	92.91
Males	**96.55**	45.90	62.22	93.18	67.21	78.10	78.33	77.05	77.69	79.03	**80.33**	79.67	94.12	78.69	**85.71**
UD-GH	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00
Micro	88.54	56.35	68.87	**89.38**	79.59	84.20	86.56	80.95	83.66	86.03	**84.05**	85.03	89.37	82.97	**86.05**
Macro	64.73	32.01	40.84	77.92	58.93	66.14	69.46	60.54	63.75	77.54	70.98	**73.15**	**78.68**	64.29	69.64

Table 7: Complete Results for Hate Target Identification (Task 2).

AraTar				**Arabic Twitter**				**OSACT5**			
	AraBERT-b				AraBERT-b				AraBERT-b		
	P	R	F1		P	R	F1		P	R	F1
RH	85.37	86.96	86.15	RH	86.05	77.08	81.32	Race-HS1	72.73	30.77	43.24
EH	78.16	80.95	79.53	EH	87.25	77.39	82.03	Religion-HS2	0.00	0.00	0.00
NH	79.07	79.07	79.07	NH	81.20	86.36	83.70	Ideology-HS3	80.00	12.50	21.62
GH	86.77	68.05	76.28	GH	81.05	72.64	76.62	Social Class-HS5	0.00	0.00	0.00
UDH	49.58	54.63	51.98	OFF	84.75	76.34	80.32	Gender-HS6	70.91	58.65	64.20
CL	91.46	92.05	91.76	Clean	93.08	96.65	94.83	NOT_HS	93.73	98.72	96.16
Micro	85.21	83.79	84.50	Micro			91.14	Micro			92.44
Macro	78.40	76.95	77.46	Macro	85.56	81.08	83.14	Macro	52.89	33.44	37.54

Table 8: Complete Results of AraBERT-b on the AraTar, Arabic Twitter and OSCAT5 corpora.

*CLEAN*ANERCorp: Identifying and Correcting Incorrect Labels in the ANERcorp Dataset

Mashael Al-Duwais, Hend Al-Khalifa and Abdulmalik Al-Salman
King Saud University
Riyadh, Saudi Arabia
{malduwais, hendk, salman}@ksu.edu.sa

Abstract

Label errors are a common issue in machine learning datasets, particularly for tasks such as Named Entity Recognition. Such label errors might hurt model training, affect evaluation results, and lead to an inaccurate assessment of model performance. In this study, we dived deep into one of the widely adopted Arabic NER benchmark datasets (ANERcorp) and found a significant number of annotation errors, missing labels, and inconsistencies. Therefore, in this study, we conducted empirical research to understand these errors, correct them and propose a cleaner version of the dataset named CLEANANERCorp. CLEANANERCorp will serve the research community as a more accurate and consistent benchmark.

Keywords: Arabic NER, Label Error, Dataset.

1. Introduction

Named Entity Recognition (NER) is the task of identifying both spans and types of named entities in text. It is a fundamental task in the natural language processing pipeline.

The ANERcorp dataset is the most well-known and utilized dataset for Arabic NER (Benajiba et al., 2007), and is a crucial benchmark for evaluating Arabic NER approaches. ANERcorp consists of 316 manually annotated articles from the news domain.

Deep Learning approaches have achieved state-of-the-art performance in the ANERcorp dataset with F1-score (0.84, 0.88, 0.89, 0.91, 0.92) (Antoun et al., 2021a, 2021b; Khalifa & Shaalan, 2019; Al-Qurishi & Souissi, 2021; Alsaaran & Alrabiah, 2021) respectively.

While researchers have relied heavily on the ANERcorp as a benchmark dataset to evaluate Arabic NER models, none has considered the dataset quality. Label errors and inconsistency can have significant impact on evaluating machine learning algorithms. Detecting and correcting these errors is crucial for training accurate NER models, as the quality of the training data directly impacts the model's performance.

Moreover, previous experiments did not consider all tags during their experiments and used different data splits. This poses challenges in comparing NER approaches and analyzing their errors.

To address this issue, we present a thorough re-annotation effort that corrects **6.34%** of the label mistakes in the ANERcorp dataset and produces a cleaner version of the dataset named (CLEANANERCorp) that significantly improves annotation quality and consistency.

To the best of our knowledge, this is the first study that systematically handles label mistakes in the ANERcorp dataset. We conducted extensive experiments on both the original ANERcorp dataset and our corrected dataset CLEANANERCorp and achieved superior results.

The contributions of this study are as follows:

- We present CLEANANERCorp, a clean version of ANERcorp that includes corrected, consistent and reliable NER annotations in both splits, where (6.45%) of the training set and (6.16%) of the test set of the ANERcorp have been updated.
- We re-evaluated the popular Arabic NER models with CLEANANERCorp and achieved a marginally high increase with the F1 score results, which is about (7.23%).
- We re-evaluated the popular Cross-lingual NER models that achieved state-of-the-art performance with the corrected test set and achieved higher results.

CLEANANERCorp is publicly available to encourage the community to use it and to improve its quality further[1].

2. Related Work

The process of identifying incorrect labels in Named Entity Recognition (NER) dataset is common in the literature. These errors can occur due to human annotator mistakes or inconsistencies in the labeling guidelines. Previous studies have addressed label quality in NER datasets (Helgadóttir, Loftsson and Rögnvaldsson, 2014; Abudukelimu *et al.*, 2018; Stanislawek *et al.*, 2019; Wang *et al.*, 2019; Reiss *et al.*, 2020; Rücker and Akbik, 2023).

(Wang *et al.*, 2019) proposed a manually corrected test set of CoNLL2003 called (CoNLL++) were they identified label mistakes in about 5.38% test sentences. Likewise, (Reiss *et al.*, 2020) proposed a more error-free version of the CoNLL2003 dataset, were they identified errors in about 3.7% of the dataset. Recently, (Rücker and Akbik, 2023) proposed CLEANCONLL, where they corrected 7.0% of all labels in the English CoNLL2003 dataset using manual re-annotation and cross checking.

[1] Github link: https://github.com/iwan-rg/CLEANANERCorp

OSACT 2024 Workshop, pages 13–19

To the best of our knowledge, there is no previous attempt to investigate the quality and label errors in Arabic NER dataset (ANERcorp).

3. ANERcorp Overview

ANERcorp is one of the earliest and most widely adopted NER corpora for Arabic. It was published in 2007 and has since become the standard in the Arabic NER literature. ANERcorp comprises two corpora for training and one for testing. The total number of articles included 316 from different newspapers.

The dataset annotation guidelines followed in the ANERcorp dataset were based on MUC Conventions (Sang & De Meulder, 2003). Following this guideline, the dataset was tagged with four entities: *person (PER), location (LOC), organization (ORG), and miscellaneous (MISC)*. The tagging scheme is the inside–outside–beginning (IOB) scheme originally proposed by (Ramshaw and Marcus, 1999). Therefore, any word on the text should be annotated as one of the following tags:

- B-PER: The Beginning of the name of a person.[2]
- I-PER: The continuation (Inside) of the name of a person.
- B-LOC: The Beginning of the name of a location.
- I-LOC: The Inside of the name of a location.
- B-ORG: The Beginning of the name of an organization.
- I-ORG: The Inside of the name of an organization.
- B-MISC: The Beginning of the name of an entity that does not belong to any of the previous classes (miscellaneous).
- I-MISC: The Inside of the name of an entity that does not belong to any of the previous classes.
- O: The word is not a named entity (Other).

The dataset contains (150,286) tokens and (32,114) types, which makes the ratio of tokens to types is (4.67). The distributions of the different tags are listed in Table 1.

Class	Ratio
PER	39%
LOC	30.4%
ORG	20.6%
MISC	10%

Table 1 Ratio of phrases by classes

In 2020, the CAMeL Lab (Obeid *et al.*, 2020) released a new version of ANERcorp, where they split the data and performed minor corrections agreed upon with the original author.

[2] The original dataset used B-PERS instead of B-PER and I-PERS instead of I-PER in the annotation. We re-annotate the dataset with the same original tags in the dataset but refer to them as B-PER and I-PER in this paper.

The changes from the original dataset include the following:

- Correct minor tag spelling errors.
- Convert the middle periods (·) and bold periods (•) to regular periods (.).
- Remove the blank Unicode character (\u200F).
- Add sentence boundaries after sequences of one or more periods.
- Split the dataset sequentially. The sentences containing the first 5/6 of the words go to training, and the rest go to testing. The training split had 125,102 words, and the test split had 25,008 words.

However, no previous efforts have been made to correct tagging errors and mislabeling in the ANERcorp dataset. We have carefully reviewed the original ANERcorp and identified the different types of labeling errors. They are listed below with examples:

A. Label Inconsistency

Some tokens were tagged differently for each sentence. For example, (الدولارات، جنيه استرليني) has been tagged sometimes as MISC and sometimes as O. Also, (الضفة الغربية) has been tagged as LOC and O in different sentences.

B. Wrong Labels

In Figure 1, the word "المتحدة" has been tagged as B-ORG while it should be tagged as I-LOC.

Figure 1 An Example of a Wrong Label

C. MISC tag Ambiguity

As the dataset follows the same classes that were defined in the MUC-6 Conventions (Sang and De Meulder, 2003) (Organization, Location, Person, and Miscellaneous), the MISC tag was not covered correctly and many MISC entities were tagged as O.

D. Sentence Beginning Ambiguity

We noticed an ambiguity in the first words of many sentences where the correct label was not clear. Figure 2 shows an example of such a sentence where the word (برند) has been tagged as (B-PER) and the meaning of the word is not clear.

Figure 2 Sentence Beginning Ambiguity Example

E. Typographical Errors

In addition to tagging errors, we noticed some typographical errors in the dataset. The dataset was written in two columns, where each word was placed on a separate line with its tag. We encountered two words attached to each other in one line without space. For example: (فيهاالبلدان، التفسيرالنصى، ولمافشلت، وراءوالدهم، المصادرالتاريخية، إنسبعةعراقيين، أكبرمحافظة).

4. Reannotation Process

The reannotation process was conducted in four distinct phases.

4.1 Annotation Guideline Definition

The ANERcorp annotations are based on MUC-6 Conventions (Sang and De Meulder, 2003) guidelines. Following these guidelines, the dataset is tagged with four entities: Person (PER), Location (LOC), Organization (ORG), and Miscellaneous (MISC). As there are no clear documentation of the ANERcorp annotation guidelines, we have defined our own guidelines that follow MUC-6 published guidelines and suit Arabic language. For example, we consider prefixes to be part of the entity names. For example: (منظمة الامم المتحدة), (شركة النفط النيجيرية), (بورصة نيويورك).

We developed a special handling for ambiguities in the guidelines to resolve cases that were not clear during the revision. In most cases, we assigned a tag that matched the context of the sentence. Following (Rücker and Akbik, 2023), we decided to tag the national sport team with ORG instead of LOC (المنتخب المصري، المنتخب السعودي). Political houses were also tagged as LOC (البيت الأبيض، الكرملين).

We have noticed inconsistency in tagging the currency, sometimes as MISC and sometimes as O or LOC. Following CoNLL tagging, we decided to label the currency and physical units as O instead of MISC.

4.2 Automatic Error Detection with CLEANLAB

CLEANLAB[3] is a framework that automatically detects label issues in a machine learning dataset using confident learning (Wang and Mueller, 2022). This framework uses existing models to detect dataset problems that can be fixed to train even better models. We utilized CLEANLAB as a first round to check the number of issues in the dataset. We detected (1945) issues. These issues have been manually investigated and corrected.

4.3 Manual Re-annotation

An annotator was hired to manually re-annotate all the entities in the dataset. The annotator was provided with guidelines and encouraged to use search engines and Wikipedia for suspicious token spans. The dataset was split into nine files for ease of handling.

4.4 Final Revision

After the re-annotating process of all the tokens, a final round of revision has been conducted by the annotator and the author of the paper to resolve any ambiguity and inconsistency in the updated tags.

Finally, we corrected a total of **9518** label mistakes, which is approximately **6.34%** of the dataset.

5. Evaluation

5.1 Dataset Statistics

All labeling errors and typographical errors detected were resolved. The following subsections present some statistics on the data.

A. Label Distribution

Tables 2 and 3 compare the total count of annotated named entities and the distribution across the four classes for CLEANANERCorp and the original ANERcorp. We observe that CLEANANERCorp has a slightly higher number of ORG and MISC entities than the base version. This originates from a more consistent use of ORG labels in organization names and MISC labels for adjectives and entity types, such as sports leagues and events.

	ANERcorp		**CLEANANERCorp**	
Class	#	%	#	%
PER	1499	5.99%	1504	6.01%
LOC	751	3.00%	813	3.25%
ORG	725	2.90%	1006	4.02%
MISC	400	1.60%	1081	4.32%
O	21633	86.50%	20604	82.39%
Total	25008	100%	25008	100%

Table 2 Statistics of test set entities in ANERcorp vs. CLEANANERCorp datasets.

	ANERcorp		**CLEANANERCorp**	
Class	#	%	#	%
PER	4926	3.94%	4906	3.92%
LOC	4301	3.44%	4649	3.72%
ORG	2691	2.15%	4254	3.40%
MISC	1263	1.01%	5278	4.22%
O	111921	89.46%	106015	84.74%
Total	125102	100%	125102	100%

Table 3 Statistics of entities of the training set in ANERcorp vs. CLEANANERCorp datasets.

B. Labels Changed

Table 4 shows the extent of the label updates introduced compared to the original dataset. A total of (**9518**) labels were modified from the original dataset, which is (**6.34%**) of the total dataset. Tables 5 and 6 further examine the update details for each data split.

[3] https://github.com/cleanlab

	TRAIN		TEST		TOTAL	
	#	%	#	%	#	%
Changed	7974	**6.37**	1544	**6.17**	9518	**6.34**
Unchanged	117128	93.6	23464	93.83	140592	93.66
Total	125102	100	25008	100	150110	100

Table 4 NER labels updated in CLEANANERCorp datasets.

	CLEANANERCorp Train Set	
	#	%
Label Corrected	1664	1.33%
Label Added	6310	5.04%
Label Unchanged	117128	93.63%
#Entities	125102	100%

Table 5 NER labels in the CLEANANERCorp train set according to the type of change.

	CLEANANERCorp Test Set	
	#	%
Label Corrected	392	1.57%
Label Added	1152	4.61%
Label Unchanged	23464	93.83%
#Entities	25008	100%

Table 6 NER labels in the CLEANANERCorp test set according to the type of change.

6. Experiments

To determine the extent to which our relabeling effort affects model performance, we re-evaluated a set of NER models on CLEANANERCorp and ANERcorp in two different settings: monolingual and cross-lingual transfer.

Currently, fine-tuning large pre-trained language models has achieved state-of-the-art performance on both monolinguals (Antoun, Baly and Hajj, 2021a, 2021b) and cross-lingual NER (Hu *et al.*, 2020; Lan *et al.*, 2020). Therefore, we selected pre-trained language models from the literature that report state-of-the-art results on Arabic and English-Arabic cross-lingual transfer and re-evaluated them on different dataset versions for the NER task.

For the cross-lingual transfer, we experimented with a zero-shot cross-lingual transfer from English to Arabic, where the model was trained on English data and tested on Arabic. We used the CoNLL2003 dataset (Sang and De Meulder, 2003) for training and validation.

Although there are other published results (Abdul-Mageed et al., 2021; Khalifa & Shaalan, 2019) with higher SOTA, they reported the results on different data splits and tested the models without the MISC tag, focusing only on three tags: person (PER), location (LOC), and organization (ORG), while setting other labels to the unnamed entity (O).

6.1 Reference Models

We re-evaluated state-of-the-art Arabic and multilingual language models on the CLEANANERCorp and ANERcorp datasets.

For the Arabic pretrained language models, we re-evaluated the following:

- **ARABERT**v0.2 base (Antoun, Baly and Hajj, 2021a): The state-of-the-art Arabic-specific BERT model for various Arabic IE tasks. The model contained 24 layers of encoders stacked on top of each other, 16 self-attention heads, and a hidden size of 1024.
- **ARBERT** (Abdul-Mageed, Elmadany and Nagoudi, 2021): Arabic-specific Transformer LMs pre-trained on very large and diverse datasets, including MSA as well as Arabic dialects.
- **AraELECTRA** (Antoun, Baly and Hajj, 2021b): A pretrained ELECTRA model on a large-scale Arabic dataset.

For the cross-lingual experiments, we re-evaluated

- **mBERT** (Devlin *et al.*, 2019): Multilingual BERT pretrained on Wikipedia of 104 languages using masked language modelling (MLM).
- **XLM-RoBERT** (XLM-R) (Conneau *et al.*, 2020): A transformer-based multilingual masked language model pre-trained on text in 100 languages that obtains state-of-the-art performance on different cross-lingual tasks.
- **GigaBERT** (Lan *et al.*, 2020): A bilingual BERT for English-to-Arabic cross-lingual transfer trained on newswire English and Arabic text from the Gigaword dataset in addition to Wikipedia and Web crawl data.

Hyperparameter: For monolingual fine-tuning experiments, we followed the same hyperparameter reported by (Antoun et al., 2021b), where all the models were fine-tuned with batch size set to (32), maximum sequence length of (256), and learning rates (5e-5). For cross-lingual fine-tuning, we followed the same hyperparameters reported by (Hu et al., 2020), where mBERT was fine-tuned for two epochs, with a training batch size of (32) and a learning rate of (2e-5), and XLM-R was fine-tuned for two epochs with a learning rate of 3e-5 and size of 16. All hyperparameter tuning for the cross-lingual experiment was performed on the English validation data.

6.2 Monolingual Results

The experimental results of the tested models for the different dataset versions are listed in Table 7. F1-score was averaged over three runs with different seeds for each experimental setting.

Model	Train/Test : **ANERcorp**	Train/Test : **CLEANANERCorp**
AraBERT v2	0.83	**0.89**
ARBERT	0.83	**0.89**
AraELECTRA	0.82	**0.87**

Table 7 Average F1 score of fine-tuning Arabic LMs on ANERcorp vs. CLEANANERCorp datasets.

The results show that CLEANANERCorp achieved marginally higher performance on all tested models compared to the original dataset, which indicates that our relabeling effort successfully improved label quality and consistency.

AraBERT F1 score has increased by (7.23%) from (0.83) to (0.89) after re-annotation. Table 8 shows a detailed comparison of each entity type in terms of Precision, Recall and F1-score for the AraBERT model on the two versions of the datasets.

We can see that all the F1 scores increased after correction, and the highest gain in entity F1 score was from the MISC and ORG labels, where the F1 score increased by (26.47%) and (16%), respectively.

	ANERcorp			**CLEANANERCorp**		
	Prec	**Rec**	**F1**	**Prec**	**Rec**	**F1**
LOC	0.89	0.93	0.91	0.94	0.92	0.93
MISC	0.73	0.63	0.68	0.85	0.86	0.86
ORG	0.76	0.73	0.75	0.85	0.87	0.86
PER	0.88	0.84	0.86	0.93	0.90	0.92
Overall	0.84	0.82	0.83	0.89	0.89	0.89

Table 8 Entity-based precision, recall, and F1 score of fine-tuned AraBERT on ANERcorp vs. CLEANANERCorp datasets.

6.3 Cross-Lingual Zero-Shot Transfer Results

Table 9 reports the average F1 scores over three runs with different seeds for each experimental setting.

From the results in Table 9, we can observe a high increase in F1 scores when transferring to the corrected dataset compared to those on the original test set.

Model	**Train: Conll2003 Test: ANERcorp**	**Train: Conll2003 Test: CLEANANERCorp**
mBERT-base	0.46	**0.48**
XLM-R-base	0.52	**0.62**
XLM-R-Large	0.53	**0.62**
GigaBERT	0.61	**0.72**

Table 9 Average F1 Scroe of Cross-lingual transfer on the ANERcorp vs. CLEANANERCorp datasets.

For example, fine-tuning XLM-R-base achieved (19.23%) increase from the (0.52) to (0.62) F1-score. Table 10 shows the F1 score per entity type, where we can see a high increase in the MISC label F1 score from (0.08) to (0.57), which justifies the increase in the overall score.

	ANERcorp			**CLEANANERCorp**		
	Prec	**Rec**	**F1**	**Prec**	**Rec**	**F1**
LOC	0.63	0.72	0.68	0.61	0.70	0.65
MISC	0.05	0.19	0.08	0.59	0.56	0.57
ORG	0.41	0.54	0.46	0.44	0.53	0.48
PERS	0.61	0.71	0.66	0.70	0.70	0.70
Overall	0.43	0.63	0.51	0.59	0.63	0.61

Table 10 Entity-based precision, recall, and F1 score of the Cross-lingual Transfer of XLM-R on ANERcorp vs. CLEANANERCorp dataset.

The above results indicate that CLEANANERCorp is more consistent with the CONLL2003 dataset and can be used to reflect the accuracy of the Cross-lingual Zero-Shot models more stably.

To get further insight into the label quality of the corrected and original dataset. We analyze the best model performance on cross-lingual zero-shot experiment using GigaBERT model.

Figures 3 and 4 show the classification report for the cross-lingual transfer on GigaBERT using the original and corrected version dataset. We noticed an improvement in the F1 score of all the tags specially for the MISC, PER and ORG were they have been mainly corrected in the new dataset version.

	precision	recall	f1-score	support
LOC	0.72	0.84	0.78	676
MISC	0.06	0.21	0.10	243
ORG	0.58	0.63	0.60	459
PER	0.75	0.76	0.75	906
micro avg	0.53	0.70	0.60	2284
macro avg	0.53	0.61	0.56	2284
weighted avg	0.64	0.70	0.66	2284

Figure 3 Classification Report for the Cross-lingual Zero Shot transfer using the original ANERcorp

	precision	recall	f1-score	support
LOC	0.73	0.84	0.78	682
MISC	0.70	0.68	0.69	835
ORG	0.69	0.57	0.63	590
PER	0.79	0.81	0.80	902
micro avg	0.74	0.73	0.73	3009
macro avg	0.73	0.72	0.72	3009
weighted avg	0.73	0.73	0.73	3009

Figure 4 Classification Report for the Cross-lingual Zero Shot transfer using CLEANANERCorp

Figure 5 shows the confusion matrix of the original dataset, where we see that the beginning of a person and location is often confused with the inside of MISC token. Figure 6 shows the confusion matrix of the corrected dataset with major improvements.

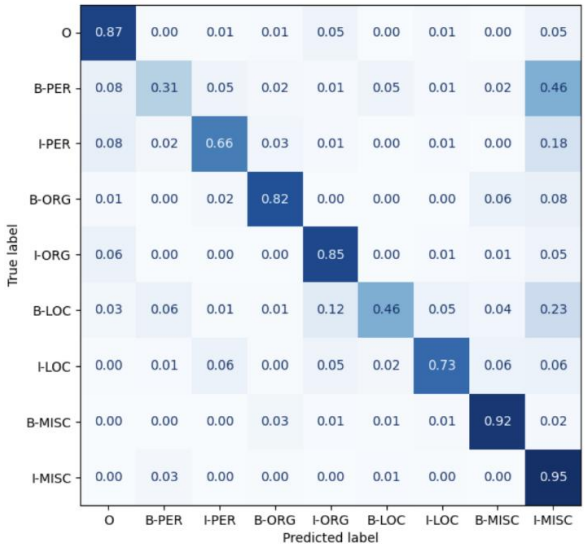


Figure 5 Confusion Matrix for Zero-shot Cross-lingual transfer using ANERCrop

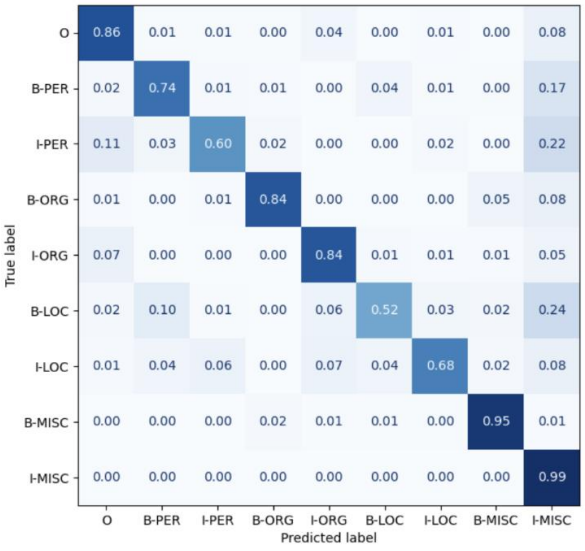


Figure 6 Confusion Matrix for Zero-shot Cross-lingual transfer using CLEANANERCorp

7. Conclusion

We presented CLEANANERCorp, a corrected and cleaner version of the widely adopted Arabic NER benchmark dataset ANERcorp. Our re-annotation updated (6.34%) the labels in the original dataset. Our evaluation of monolingual and cross-lingual NER language models achieved higher performance and strongly indicated that the overall annotation quality and consistency were significantly improved. Therefore, we contribute to improving the quality of the public Arabic NER datasets with updated and more consistent NER labels.

8. References

Abdul-Mageed, M., Elmadany, A. and Nagoudi, E.M.B. (2021) 'ARBERT & MARBERT: Deep Bidirectional Transformers for Arabic', in C. Zong et al. (eds) *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)*. *ACL-IJCNLP 2021*, Online: Association for Computational Linguistics, pp. 7088–7105. Available at: https://doi.org/10.18653/v1/2021.acl-long.551.

Abudukelimu, H. *et al.* (2018) 'Error Analysis of Uyghur Name Tagging: Language-specific Techniques and Remaining Challenges', in N. Calzolari et al. (eds) *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. *LREC 2018*, Miyazaki, Japan: European Language Resources Association (ELRA). Available at: https://aclanthology.org/L18-1700 (Accessed: 30 March 2024).

Al-Qurishi, M.S. and Souissi, R. (2021) 'Arabic Named Entity Recognition Using Transformer-based-CRF Model', in M. Abbas and A.A. Freihat (eds) *Proceedings of the 4th International Conference on Natural Language and Speech Processing (ICNLSP 2021)*. *ICNLSP 2021*, Trento, Italy: Association for Computational Linguistics, pp. 262–271. Available at: https://aclanthology.org/2021.icnlsp-1.31 (Accessed: 25 February 2024).

Alsaaran, N. and Alrabiah, M. (2021) 'Arabic named entity recognition: A BERT-BGRU approach', *Comput. Mater. Contin*, 68, pp. 471–485.

Antoun, W., Baly, F. and Hajj, H. (2021a) 'AraBERT: Transformer-based Model for Arabic Language Understanding'. arXiv. Available at: https://doi.org/10.48550/arXiv.2003.00104.

Antoun, W., Baly, F. and Hajj, H. (2021b) 'AraELECTRA: Pre-Training Text Discriminators for Arabic Language Understanding'. arXiv. Available at: http://arxiv.org/abs/2012.15516 (Accessed: 25 February 2024).

Benajiba, Y., Rosso, P. and BenedíRuiz, J.M. (2007) 'ANERsys: An Arabic Named Entity Recognition System Based on Maximum Entropy', in A. Gelbukh (ed.) *Computational Linguistics and Intelligent Text*

Processing. Berlin, Heidelberg: Springer (Lecture Notes in Computer Science), pp. 143–153. Available at: https://doi.org/10.1007/978-3-540-70939-8_13.

Conneau, A. *et al.* (2020) 'Unsupervised Cross-lingual Representation Learning at Scale'. arXiv. Available at: https://doi.org/10.48550/arXiv.1911.02116.

Devlin, J. *et al.* (2019) 'BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding'. arXiv. Available at: https://doi.org/10.48550/arXiv.1810.04805.

Helgadóttir, S., Loftsson, H. and Rögnvaldsson, E. (2014) 'Correcting Errors in a New Gold Standard for Tagging Icelandic Text', in N. Calzolari et al. (eds) *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. *LREC 2014*, Reykjavik, Iceland: European Language Resources Association (ELRA), pp. 2944–2948. Available at: http://www.lrec-conf.org/proceedings/lrec2014/pdf/677_Paper.pdf (Accessed: 30 March 2024).

Hu, J. *et al.* (2020) 'XTREME: A Massively Multilingual Multi-task Benchmark for Evaluating Cross-lingual Generalisation', in *Proceedings of the 37th International Conference on Machine Learning*. *International Conference on Machine Learning*, PMLR, pp. 4411–4421. Available at: https://proceedings.mlr.press/v119/hu20b.html (Accessed: 20 December 2023).

Khalifa, M. and Shaalan, K. (2019) 'Character convolutions for Arabic Named Entity Recognition with Long Short-Term Memory Networks', *Computer Speech & Language*, 58, pp. 335–346. Available at: https://doi.org/10.1016/j.csl.2019.05.003.

Lan, W. *et al.* (2020) 'An Empirical Study of Pre-trained Transformers for Arabic Information Extraction', in B. Webber et al. (eds) *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. *EMNLP 2020*, Online: Association for Computational Linguistics, pp. 4727–4734. Available at: https://doi.org/10.18653/v1/2020.emnlp-main.382.

Obeid, O. *et al.* (2020) 'CAMeL Tools: An Open Source Python Toolkit for Arabic Natural Language Processing', in N. Calzolari et al. (eds) *Proceedings of the Twelfth Language Resources and Evaluation Conference*. *LREC 2020*, Marseille, France: European Language Resources Association, pp. 7022–7032. Available at: https://aclanthology.org/2020.lrec-1.868 (Accessed: 19 February 2024).

Ramshaw, L.A. and Marcus, M.P. (1999) 'Text Chunking Using Transformation-Based Learning', in S. Armstrong et al. (eds) *Natural Language Processing Using Very Large Corpora*. Dordrecht: Springer Netherlands (Text, Speech and Language Technology), pp. 157–176. Available at: https://doi.org/10.1007/978-94-017-2390-9_10.

Reiss, F. *et al.* (2020) 'Identifying Incorrect Labels in the CoNLL-2003 Corpus', in R. Fernández and T. Linzen (eds) *Proceedings of the 24th Conference on Computational Natural Language Learning*. *CoNLL 2020*, Online: Association for Computational Linguistics, pp. 215–226. Available at: https://doi.org/10.18653/v1/2020.conll-1.16.

Rücker, S. and Akbik, A. (2023) 'CleanCoNLL: A Nearly Noise-Free Named Entity Recognition Dataset'. arXiv. Available at: http://arxiv.org/abs/2310.16225 (Accessed: 28 November 2023).

Sang, E.F.T.K. and De Meulder, F. (2003) 'Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition'. arXiv. Available at: http://arxiv.org/abs/cs/0306050 (Accessed: 20 February 2024).

Stanislawek, T. *et al.* (2019) 'Named Entity Recognition - Is There a Glass Ceiling?', in M. Bansal and A. Villavicencio (eds) *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*. *CoNLL 2019*, Hong Kong, China: Association for Computational Linguistics, pp. 624–633. Available at: https://doi.org/10.18653/v1/K19-1058.

Wang, W.-C. and Mueller, J. (2022) 'Detecting Label Errors in Token Classification Data'. arXiv. Available at: http://arxiv.org/abs/2210.03920 (Accessed: 26 February 2024).

Wang, Z. *et al.* (2019) 'CrossWeigh: Training Named Entity Tagger from Imperfect Annotations', in K. Inui et al. (eds) *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. *EMNLP-IJCNLP 2019*, Hong Kong, China: Association for Computational Linguistics, pp. 5154–5163. Available at: https://doi.org/10.18653/v1/D19-1519.

Munazarat 1.0: A Corpus of Arabic Competitive Debates

Mohammad Majed Khader[1], **Abdul Gabbar Al-Sharafi**[2],
Mohamad Hamza Al-Sioufy[3], **Wajdi Zaghouani**[4], **Ali Al-Zawqari**[5]
[1]QatarDebate Center, [2]Sultan Qaboos University,
[3]Georgetwon University in Qatar, [4]Hamad Bin Khalifa University,
[5]Department of Fundamental Electricity and Instrumentation, Vrije Universiteit Brussel
[1]mkhader@qatardebate.org, [2]alsharaf@squ.edu.om, [3]ma2052@georgetown.edu
[4]wzaghouani@hbku.edu.qa, [5]aalzawqa@vub.be

Abstract

This paper introduces the Corpus of Arabic Competitive Debates, Munazarat. Despite the significance of competitive debating in fostering critical thinking and promoting dialogue, researchers in the fields of Arabic Natural Language Processing (NLP), linguistics, argumentation studies, and education have limited access to datasets on competitive debating. At this stage of the study, we introduce Munazarat 1.0, which combines transcribed recordings of approximately 50 hours from 73 debates at QatarDebate-recognized tournaments, all available on YouTube. Munazarat is a novel specialized Arabic speech corpus, predominantly in Modern Standard Arabic (MSA), covering diverse debating topics and accompanied by metadata for each debate. The transcription of debates was performed using Fenek, a speech-to-text Kanari AI tool, and reviewed by three native Arabic speakers to enhance quality. The Munazarat 1.0 dataset can serve as a valuable resource for training Arabic NLP tools, developing argumentation mining machines, and analyzing Arabic argumentation and rhetoric styles.

Keywords: Arabic Speech Corpus, Modern Standard Arabic, Debates

1. Introduction

Arabic is the sixth most spoken language in the world. As a Semitic language, Arabic distinguishes itself from the Indo-European linguistic family in several dimensions: phonetically, morphologically, syntactically, and semantically. Thus, the development and research of Arabic NLP applications face various challenges based on the language's linguistic structure (Shaalan et al., 2019). Furthermore, an additional challenge is that Arabic exists today in three forms: (i) Classical Arabic, (ii) Modern Standard Arabic (MSA), and (iii) Dialectical Arabic, which varies significantly based on geographical regions. The Arabic language is suffering from a scarcity of available open datasets compared to English and other languages like Chinese, German, and French. In Papers With Code (pap), a repository showed results of open text datasets in March 2024: 1446 for English, 205 for Chinese, 126 for German, and only 54 for Arabic. While Hugging Face repository (hug) showed results of only 446 Arabic datasets out of 126,088 open text datasets in comparison to 8,826 for English, 1005 for Chinese, and 667 for German.

Competitive debating, an intellectually rigorous oral argumentative discourse activity governed by specific rules and regulations, typically takes place in the context of large tournaments. Thousands of university and school students from different geographical regions around the world participate in local and international Arabic debating tournaments. For Arabic debating, QatarDebate Center (www.qatardebate.org) is considered the leading debate institution, organizing major international Arabic debating tournaments and publishing the recordings of debates on YouTube. QatarDebate's 3 vs 3 debate format, as shown in Figure 1, a modified format of the World Schools Debating Championship (www.wsdcdebating.org), is dominant in Arabic competitive debating activities. A motion is presented for every debate in this format, and two opposing teams compete against one another. Every team consists of three speakers, and each one is allowed to talk for a total of 6-7 minutes, beginning with the first proposition speaker, followed by the first opposition speaker, and so on till the last opposition speaker. Then, each team delivers a three-minute technical speech called the "Reply Speech" that does not include any new argument. Due to the competitive nature of these debates, an adjudication panel of an odd number of judges votes for the most persuasive team to win and assign individual speakers' scores. The effectiveness of the offered argumentation and refutation is the primary criterion for judging debate presentations. This type of debate is very structured and follows specific rules and regulations that govern the flow of the debate and its evaluation.

The significance of creating an Arabic debate corpus comes from the fact that debates are rich in argumentative and sentimental speeches that can help study Arabic argumentation and rhetoric styles. It can also be used to study various linguistic features of the spoken MSA among native and non-native debaters. In addition, it provides raw data for developing Arabic NLP tools for argumen-

OSACT 2024 Workshop, pages 20–30

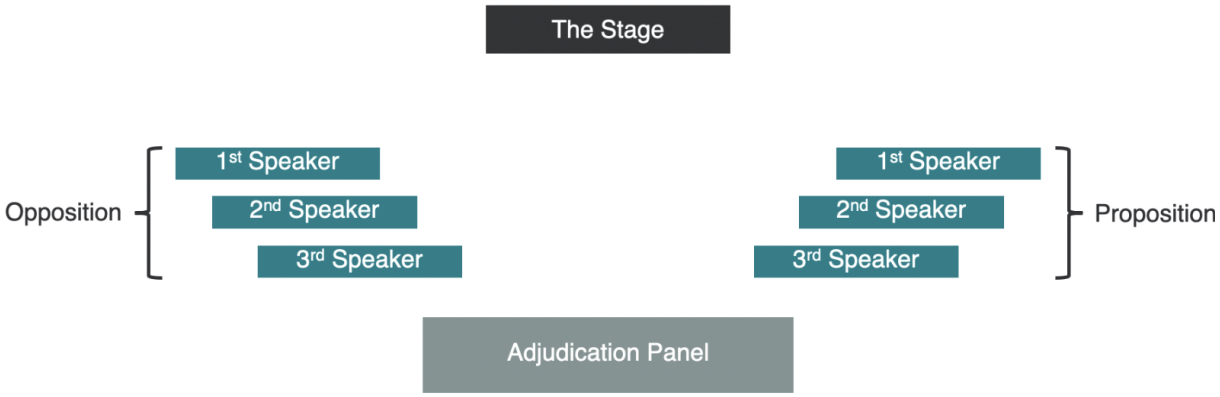


Figure 1: Illustration of 3 vs 3 Debate Format

tation mining, speech recognition, etc. Unlike other datasets, Munazarat 1.0 stands today as the specialized corpus of Arabic competitive debate and the largest corpus of argumentative Arabic content.

2. Related Work

Dataset availability is an essential key to developing NLP applications. However, the cost of acquiring corpora represents a major challenge, especially in Arabic NLP with all of its variations (Ahmed et al., 2022c; Zaghouani, 2014). After a survey of available Arabic resources today (Al-sulaiti and Atwell, 2006; El-Khair, 2016; Al-Twairesh et al., 2018; Graja et al., 2010; Almeman et al., 2013; Alrabiah et al., 2013; Ahmed et al., 2022a; Mubarak et al., 2021; Khader, 2020; Al-Fetyani et al., 2023; Bouamor et al., 2018), and despite the recent efforts in the field of Arabic NLP (Darwish et al., 2021), the available specialized Arabic corpora remain in shortage. Datasets of relevance to our study manifest as either Arabic speech corpora or compilations encompassing discourse of a debating or argumentative nature.

The development of the Arabic PropBank has been instrumental in the semantic analysis of Arabic texts. These efforts have laid the groundwork for parsing argument structures in sentences (Palmer et al., 2008; Diab et al., 2008; Zaghouani et al., 2010) while Error annotation is essential for the accuracy and reliability of language resources. Studies focusing on large-scale Arabic error annotation and non-native text correction have significantly contributed to the field (Zaghouani et al., 2014) and (Zaghouani et al., 2015). Furthermore, Dialectal variation in Arabic poses unique challenges for argumentation analysis. The MADAR project and the DIACT corpus have addressed this by focusing on dialect-specific expressions and the use of rhetorical devices such as irony (Bouamor et al., 2018; Abbes et al., 2020).

By situating our work alongside these significant contributions, we aim to address the gap in resources specific to argumentation within the Arabic language, building on the robust foundations laid by these earlier works. Each cited resource provides a unique perspective on the intricacies of argumentative discourse, from structural annotations to the subtleties of linguistic diversity.

2.1. Speech Corpora

Lately, two Arabic speech corpora were introduced: the Massive Arabic Speech Corpus (MASC) (Al-Fetyani et al., 2023), which contains 1,000 hours from over 700 YouTube channels, and QCRI Aljazeera Speech Resource (QASR) (Mubarak et al., 2021) which is the largest Arabic speech corpus to date and consists of 2,000 hours from Aljazeera TV channel shows. Recently, a digital corpus of the Australian Parliamentary Debates was published (Katz and Alexander, 2023) following the lead of the Canadian Parliamentary Debates (Beelen et al., 2017). Those two studies show the recent interest in collecting and publishing specialized debate corpora, namely political debates. The availability of English debate corpora highlights the gap for an equivalent Arabic debate collection we seek to address in providing a source for Arabic competitive debates.

2.2. Debate & Argumentation Corpora

Many corpora were found to be interested in studying debates and argumentation models in English (Hautli-Janisz et al., 2022; Serban et al., 2015; Fisas et al., 2016; Peldszus and Stede, 2015). Several studies have compiled corpora to advance research in argument mining and related tasks. Walker et al. (2012) introduced a corpus of English language debates annotated with argumentative discourse units to facilitate computational argumentation research. Zhang et al. (2021) presented a corpus of Wikipedia talk page conversations annotated for conversational failure, enabling

the study of breakdowns in cooperative discussion. Lawrence and Reed (2020) surveyed datasets for argument mining, reviewing annotation approaches across key tasks.

Other efforts have focused on particular argumentation genres and languages. Al Khatib et al. (2018) annotated German Wikipedia articles with argument strategies, like evidence types, to analyze deliberative argumentation. Bar-Haim et al. (2006) overviewed textual entailment challenges involving argumentation data. Orăsan and Evans (2007) developed a corpus of noun phrase animacy annotations to assist anaphora resolution with potential dialogue applications. Some datasets have annotated the persuasiveness of arguments. Habernal and Gurevych (2016) presented a corpus of web argument pairs annotated for comparative convincingness to predict persuasiveness. Hidey and McKeown (2018) annotated student essays for argument persuasiveness and sequencing.

Other studies have advanced annotation methodologies. For instance, Musi et al. (2018) performed an annotation study of argument schemes like expert opinion to provide guidelines. On the other hand, Aharoni et al. (2014) annotated claims and evidence in controversial topics for automatic detection. There are also argument-mining efforts in other languages like Italian (Durmus et al., 2021) and argument relation annotations from multilingual social media like X (Twitter previously) (Bosc et al., 2016).

For the purpose of this study, the most notable previous work is QT30 corpus (Hautli-Janisz et al., 2022), which contains public debates from the BBC's show 'Question Time'. However, it is limited to only 30 episodes and focuses solely on political debates. Yet, to the best of our knowledge, there is no work focusing on building a corpus in Arabic for argumentation or debating, except for two recent projects. The first one is a project of (Khader, 2020), which introduced a small corpus containing only 12 debate recordings. The other one is the Qatari Corpus of Argumentative Writing (QCAW) (Ahmed et al., 2024; Zaghouani et al., 2024; Ahmed et al., 2022a), which targets bilingual (Arabic/English) argumentative texts by students, providing a novel resource for cross-linguistic argumentation studies with 195 texts in Arabic and 195 texts in English. This corpus facilitates a deeper understanding of argumentative structures within an educational context, contributing to the field of discourse analysis. Yet, QCAW does not incorporate any spoken argumentative content.

The availability of rich resources for argumentative and persuasive Arabic speech is nonexistent. Yet, argument mining from spoken content could enable studies on rhetoric, reasoning, and dialectics across the language's breadth. Competitive debating generates valuable linguistic data - structured speeches rich in argumentation, sentiment, and diverse vocabulary spanning different topics. Debates capture authentic goal-oriented argumentation between experts, unlike other dialogues (Serban et al., 2015). The lack of argumentative and conversational Arabic speech data poses challenges for speech recognition, dialect studies, and MSA research (Al-Fetyani et al., 2023). Applications like argument mining also require substantial training corpora (Lawrence and Reed, 2019).

Munazarat 1.0 data can facilitate Arabic research on linguistics, reasoning, debating, and NLP applications through this resource. Our work addresses the key limitations of scarce available Arabic corpora compared to other languages, very minimal argumentative or conversational Arabic data, lack of large-scale Arabic speech resources for training models, and the absence of a dedicated corpus for the rich Arabic debating domain.

3. Methods

3.1. Debate Collection

Munazarat 1.0 consists of approximately 50 hours of transcribed Arabic competitive debates that QatarDebate Center hosted in several tournaments. The corpus is created using 73 debates prerecorded and already published online by the host, as we collected them from YouTube without disclosing any extra private information about the debaters. The collected debates comprise a combination of university and school debates held between 2013 and 2023. The corpus will be expanded, with an expected goal of reaching 120 debates by the end of 2024.

3.2. Transcription and Human Review

All debates were transcribed using Fenek, a multilingual Arabic/English speech-to-text tool from Kanari AI (www.kanari.ai) (Khurana and Ali, 2016). After that, all debate transcripts were cleaned, briefly annotated, and reviewed in three stages, as described below, to ensure the transcription quality. We also published more details on the human reviewing guidelines that we used in this project with the dataset for public usage. It is important to mention that 10 transcripts were taken from publicly available previous work of (Khader, 2020) and those transcripts went only through stages two and three of the human review.

- **Stage One:** During the first review, the reviewer listens to the debate from the YouTube link while reading the transcription in order to eliminate any mistakes made by the Artificial Intelligence (AI) tool in the transcription. We iden-

tified four types of transcription mistakes for the reviewer to correct: added words (highlighted in red), missed words (highlighted in yellow), spelling mistakes (highlighted in green), and language detection mistakes (highlighted in blue). The reviewer also deletes any side talks that happen in the recording that are not part of the six essential debate speeches, such as deleting the chair's organizational remark. We also decided to remove the "Reply Speeches" from the script since they are not essential to the debate and are not standard in all debating tournaments. The reviewer cleans any repetitions during the speeches if they were caused by unintentional stammering. By the end of this stage, the reviewer produces a clean file ready for the next reviewer with three marking steps: (i) making a brief annotation by marking the beginning of each debater's speech by (#) symbol and indicating the speaker's order and position; (ii) mentioning the gender of the speaker at the beginning of each speech as illustrated in Figure 2; (iii) marking any Point of Information (POI) from the opponent team as shown in Figure 3.

- **Stage Two:** In the second review, another reviewer reviews the script. However, this time, the reviewer only reads the text and does not listen to the debate. This stage was meant to account for any typos, grammatical and spelling mistakes, etc., that the first reviewer did not catch. In rare cases, the second reviewer revisits the debate video to cross-check the transcript.

- **Stage Three:** This stage is for quality control, where the third reviewer eliminates any mistakes that were left by the previous two reviewers and provides feedback for them during the periodic reviewers' meetings. In addition, the third reviewer tries to organize the transcript in the form of paragraphs to produce a better readable file.

4. Data Records & Analysis

Munazarat 1.0 is a unique resource for researchers interested in various aspects of Arabic competitive debating, Arabic linguistics studies, argumentation studies, education, and Arabic NLP. Munazarat 1.0 is available for public download as a ZIP file containing 73 debate files in TXT format to facilitate its effective use. Researchers can access and download this dataset via an open access github[1]. Each file is named descriptively, incorporating information about the debate, including the serial number, tournament, year, gender, and whether the speakers are native or non-native Arabic speakers. For example, 028-IUDC-2017-MFMFMF-AA represents a debate with serial number 028, from the International Universities Debating Championship (IUDC), featuring three male speakers in the proposition team and three female speakers in the opposition team, all of whom are Arabic native speakers. Each TXT file includes basic annotations that indicate the order and the gender of the speaker as well as any POI from the opponent speakers.

The corpus represents a diverse collection in several aspects, as shown in Table 1. The demographic representation in this corpus is rich. A list of 27 countries in the corpus is shown in Table 2, and the higher occurrences are relevant to the country's history of participation in the Arabic debate activity. The corpus is inclusive of 51 debates between native Arabic speakers, 22 debates between native and non-native speakers, 51 university-level debates from international tournaments, 11 school debates from international tournaments, and 11 school debates from Qatar.

Munazarat 1.0 also displays a well-balanced male-to-female ratio of (M:223, F:215) since some studies pointed out the differences in speech patterns among genders in English debates (Shaw, 2000; Hargrave and Langengen, 2021), which needs to be examined against an Arabic dataset. The debate motions are diverse and wide-ranging, from politics and philosophy to sports. Table 3 provides the overall topic distribution of the debates. For each debate, we have the following: a video recording with a YouTube link, a transcribed text (TXT) file of the debate's script, and some metadata, explained later in the Data Records section.

Table 1: Diversity Representation

Category	**Count**
Tournament Level - Debate Count	
University Level	51
School Level	22
Geographic Representation - Debate Count	
Local (Qatar)	11
International	62
Language Proficiency - Debate Count	
All Native Arab Speakers	51
Natives and Non-Natives Speakers	22
Gender Representation - Speakers Count	
Male Debaters	223
Female Debaters	215

[1] https://github.com/moh72y/Munazarat1.0/

المتحدث الأول موالاة: (أنثى)/

بسم الله الرحمن الرحيم، اللجنة الكريمة زملائي و زميلاتي في فريقي الموالاة و المعارضة السلام عليكم و رحمة الله، جئنا اليوم لنناقش النص التالي نص يقول يفضل هذا المجلس نمط حياة الرحالة الرقمي. و قد أتى نص القضية بتعريف للرحالة الرقميين و هم الأشخاص الذين يحصلون على دخلهم من خلال العمل عبر الإنترنت أثناء السفر و التنقل

Figure 2: Beginning of Speech Annotation: Debater's Role, Gender, and # Symbol.
English Translation: *First speaker Poposition (Female) In the name of God, the Most Gracious, the Most Merciful, the honorable committee, my colleagues in the proposition and opposition teams, may God's peace and mercy be upon you. We have come today to discuss the following motion, the motion says that, This house prefers the digital nomad's lifestyle. The motion came with a definition of digital nomads, who are the people who obtain their income through working online while traveling*

مداخلة: أليس الرحالة الرقميون هو عمل تنطبق عليه مشاكل العمل التقليدي/

هو طبعا عمل نحن نتكلم عن شخص يجني دخل اقتصادي هذا أصلا تعريفه عمل و لكن كيف يجني هذا الدخل
الاقتصادي بطريقة تختلف عن طريقة العمل التقليدية، طريقة العمل التقليدية أنت ملتزمة بدوام تأتينا مثلا الساعة ثمانية
الصبح للساعة ثلاثة بعد الظهر أنت ملتزمة في مكان معين محاطة بأشخاص معينين مجبورة أنت على البقاء معهم سواء

Figure 3: Point of Information (POI) Annotation.
English Translation: *Point of Information: Isn't digital nomads a job to which the problems of traditional work apply/ It is, of course, work. We are talking about a person who earns economic income. This is basically the definition of work, but how does he earn this economic income in a way that differs from the traditional method of work. The traditional method of work, You are committed to a shift. You come, for example, at eight in the morning until three in the afternoon. You are committed to a specific place surrounded by specific people that you are forced to stay with them.*

Table 2: Country Distribution

Country	No. of Teams	University Level	School Level
Qatar	35	16	19
Jordan	13	12	1
Sudan	12	12	0
Oman	12	11	1
Tunisia	10	10	0
Malaysia	9	5	4
Kuwait	8	7	1
Palestine	6	3	3
Libya	6	6	0
Lebanon	6	3	3
Türkiye	5	2	3
USA	5	5	0
Indonesia	3	2	1
Syria	3	0	3
Algeria	1	1	0
Iraq	1	1	0
Somalia	1	1	0
Bahrain	1	1	0
Norway	1	1	0
Canada	1	1	0
Poland	1	1	0
Morocco	1	0	1
Pakistan	1	0	1
Singapore	1	0	1
Yemen	1	0	1
Côte d'Ivoire	1	1	0
Australia	1	0	1
Total	146	102	44

Table 3: Topic Distribution

Topic	No. of Debates
Politics	16
Ethics/Philosophy	16
Human Rights	10
Media	6
Education	5
Technology	5
Culture	3
Environment	3
Law	3
Sports	3
Economy	2
Lifestyle	1
Total	73

5. Technical Validation

5.1. AI Transcription Accuracy Report

This human validation process was done fully on 63 newly transcribed debates out of 73, and partially on the other 10 transcripts that were taken from previous work by (Khader, 2020) as the transcripts were available for public use online. During the first human review stage mentioned above, while listening to and editing the debates, the reviewer identified transcription mistakes in four categories. The red category is used to highlight any additional words that the tool added but were not originally spoken by the speaker during their speech. The yellow category is used to highlight any words that

were added by the reviewer and were missed by the tool. The green category is used with the words caught wrongly by the tool and thus modified by the reviewer. Finally, language detection mistakes in the blue category to highlight words that were in a different language, as the debates were conducted originally in Arabic, but some terminologies in English might appear and were written in a wrong way by the tool. Figure 4 and Figure 5 show a sample of the color coding process. After that, the reviewer stores the data from each debate regarding the number of mistakes in each category and the total number of mistakes in the whole debate. Table 4 demonstrates the Mean and Median of mistake count per debate for each category reported by the human reviewer.

Following the transcription of the 63 debates using the speech-to-text tool from Kanari AI, we report an average accuracy rate of 96% per debate. Approximately 40% of the tool's mistakes fall under the 'Missed Word' category, which we attribute to microphone quality and the fast speaking pace of some debaters. Conversely, the tool effectively detected language switches when debaters used English for certain terminologies.

Table 4: Mean (M) and Median (Mdn) Transcription Accuracy Report

Category	M per Debate	Mdn per Debate
Word Count	4546	4458
Added Words	49	37
Missed Words	76	28
Spelling Mistake	58	45
Language Detection Mistake	1	0
Total Mistakes	185	140
Accuracy Rate	96%	97%

5.2. Keyword Analysis

Keyword analysis is a vital aspect of corpus studies, helping unveil a corpus's underlying themes and domain. In exploring Munazarat 1.0, a diverse debate corpus, we employed AntConc software (Anthony, 2023) to conduct a comprehensive keyword analysis. The keyness function in AntConc generates the keyword list of the studied corpus compared to a reference, usually a much larger and generic one. These keywords are not merely the most frequent words in the corpus; they represent statistically significant words that shed light on the corpus's domain. This function helps filter out stopwords, insignificant words, and letters, allowing us to recognize the corpus's domain and key themes. For this analysis, we utilized QASR (Mubarak et al., 2021), one of the largest available Arabic speech corpora, as our reference. In Table 5, we present the keywords from various categories within Munazarat 1.0, both in comparison to the corpus itself and against QASR.

A preliminary review of the keyword list from a complete or partial corpus analysis, in comparison to QASR, reveals the distinctive nature of Munazarat 1.0 as a debate corpus, with terms like "proposition", "team", "speaker". and "this house" stand out. However, it is important to note that the initial analysis of keywords within specific portions, category-based, of the corpus against Munazarat 1.0 primarily reflects the debated topics within that portion rather than providing insights regarding the characteristics of the studied category. Still, a dedicated study among various categories in Munazrart 1.0 might reveal some linguistic styles that can be associated with non-native debaters, school debaters, Qatari debaters, etc.

Table 5: Top Five Keywords per Category

Category	Against Munazarat 1.0	Against QASR
Native Speakers - University Level	العمال	الموالاة
	Labours	Proposition
	الحرب	فريق
	War	Team
	سادتي	التحدث
	Gentlemen	Speaker
	الأندية	سادتي
	Clubs	Gentlemen
	الحقوق	سوف
	Rights	Will
General - Schools Level	تحليل	الموالاة
	Analysis	Proposition
	الطلاب	التحدث
	Students	Speaker
	العربية	أيها
	Arabic	Hey
	الجمهور	فريق
	Audience	Team
	الأيام	التحدثة
	Days	Speaker - female
Native Students from Qatari Schools	تحليل	الموالاة
	Analysis	Proposition
	الأيام	التحدث
	Days	Speaker
	التواصل	تحليل
	Communication	Analysis
	يوم	التحدثة
	Day	Speaker - female
	مواقع	بركاته
	Sites	His Blessings

Table 6 demonstrates a sample from the keyword analysis per theme. Debates were selected from three themes: Politics, Ethics/Philosophy, and Technology. The results show that generic debate terms appeared, as expected, against QASR (Mubarak et al., 2021) for both politics and ethics. However, theme-specific words related to AI most notably appeared for the technology theme, telling us that QASR is most probably poor for AI terms despite its length and diversity. On the other hand, the theme-based keyword analysis reflected the themes when run against Munazarat 1.0. The words "Intelligence" and "Artificial" were the most highlighted keywords which reflects the fact that four debate transcripts out of five in the technology

المتحدث الأول موالاة: (انثي)/

بسم الله الرحمن الرحيم، اللجنة الكريمة زملائي و زميلاتي في فريقي الموالاة و المعارضة السلام عليكم و رحمة الله، اه
جئنا اليوم لنناقش النص التالي نص يقول بفضل هذا المجلس نمط حياة الرحالة الرقمي و قد أتى نص القضية بتعريف
للرحالة الرقميين و هم الأشخاص الذين يحصلون على دخلهم من خلال العمل عبر الإنترنت أثناء السفر و التنقل

Figure 4: Sample of Color Coding Transcription Mistakes
English Translation: *First speaker poposition (Female) In the name of God, the Most Gracious, the Most Merciful, the honorable committee, my colleagues in the proposition and opposition teams, may God's peace and mercy be upon you. We have come today to discuss the following motion, the motion says that, This house prefers the digital nomad's lifestyle. The motion of the issue came with a definition of digital nomads, who are the people who obtain their income through working online while traveling*

27	كلمات زائدة يجب حذفها Extra words that require deletion
28	كلمات ناقصة تمت إضافتها Missing words that have been added
45	كلمات خاطئة تم تعديلها Misspelled words that have been corrected
4	خطأ في التعرف على اللغة تم تعديله Language detection mistakes that have been edited

Figure 5: Sample of the Mistakes Table in the First Human Review Process

category are debates about AI.

6. Usage Notes

Along with the debate transcript files, we offer a detailed Excel sheet that provides metadata for each debate. This metadata includes information such as the tournament, university or school level, debate motion, proposition and opposition teams, the number of male and female debaters, word count, YouTube link, and the debate topic genre (e.g., Politics, Economy, Human Rights, Law, etc.). Researchers can use this metadata for various analytical purposes and to filter debates based on specific criteria.

The dataset provided in this study is the largest available Arabic argumentative transcribed text to date, which makes it suitable for several applications including but not limited to the three following suggestions: (i) using UBIAI (www.ubiai.tools) text annotation online software to annotate the speeches' argument scheme since it is compatible with the Arabic text; (ii) applying sentimental analysis on the corpus using tools such as Repustate (www.repustate.com); and (iii) running more linguistic analysis through AntConc (www.laurenceanthony.net/software/antconc/).

The dataset is currently provided in a separate TXT file for each debate. However, it can be easily converted to other formats as per the researchers' requirements. It can also be easily segmented into separate files per speech for extra gender or demographical-based analysis. To facilitate the segmentation process, the beginning of each speech is marked by a (#) symbol.

While Munazarat 1.0 serves as a substantial raw corpus, it currently lacks standard splits into training and test sets to enable benchmarking of AI models. Creating such splits by partitioning the data while maintaining balance across dialects, speaker demographics, topics, and other variables is an important area we aim to pursue in future work. We plan to take measures to avoid speaker overlap between the splits. The speaker metadata captured in our annotations will assist in creating speaker-independent partitions. Providing standardized training and test splits will allow Munazarat 1.0 to serve as a rigorous benchmark dataset for developing and evaluating Arabic argument mining and related NLP models. We will make the splits available along with the corpus.

7. Limitations

Munazarat 1.0 has some limitations to highlight. In the current version, only competitive debating content is included. Adding other genres, like talk shows, could improve the diversity of the dataset. Moreover, the metadata currently captures basic attributes. More fine-grained speakers and sociolinguistic metadata could enable deeper analyses. The semi-automated transcription allows some errors; therefore, periodic human checks on newer data may help enhance quality. Finally, the release

Table 6: Top Five Keywords per Selected Themes

Theme	Against Munazarat 1.0	Against QASR
Politics	الدول	الموالاة
	Countries	Proposition
	المساعدات	الدول
	Aids	Countries
	الصين	سادتي
	China	Gentlemen
	التدخل	فريق
	Intervention	Team
	روسيا	المتحدث
	Russia	Speaker
Ethics & Philosophy	الطبيب	الموالاة
	Physician	Proposition
	الرقابة	أيها
	Surveillance	Hey
	الأيام	فريق
	Days	Team
	الآثار	المتحدث
	Monuments	Speaker
	الإسلام	السادة
	Islam	Gentlemen
Technology	الذكاء	الذكاء
	Intelligence	Intelligence
	الصناعي	الصناعي
	Artificial	Artificial
	الاصطناعي	الاصطناعي
	Artificial	Artificial
	التقدم	المولاة
	Progress	Proposition
	الإنسان	سادتي
	Human	Gentlemen

rights limit sharing some video recordings publicly, and getting broader rights could increase accessibility. Addressing these limitations through corpus expansion, increased metadata, transcription quality checks, and enhanced accessibility can make Munazarat 1.0 an even more impactful community resource. We aim to pursue these improvements in ongoing and future work.

8. Conclusion

We have introduced Munazarat 1.0, the first large-scale corpus of transcribed Arabic competitive debates. Spanning 50 hours of content across 73 university and school-level debates, Munazarat 1.0 represents a valuable linguistic resource for Arabic NLP and related fields. We described the rigorous process of collecting high-quality video recordings and machine transcribing debates using speech recognition, followed by extensive human reviews.

With the provided metadata, including speaker demographics and debate topics, Munazarat 1.0 enables multifaceted analyses of argumentation, rhetoric, dialectal variations, and other phenomena in Arabic debates. Our validation demonstrates the accuracy of the AI-generated transcripts. Keyword analyses reveal the corpus's core themes like argumentation and specific debate motions. Munazarat 1.0 provides Arabic researchers with a substantial dataset to train computational models and drive advancements for impactful applications in education, linguistics, and reasoning analysis. Currently, two works in the literature are introduced to take advantage of Munazarart 1.0, namely in (Al-Sharafi et al., accepted 2024; Al-Zawqari et al., accepted 2024). The first one is developing an annotation model for argumentation in competitive debates, and the second is focusing on the classification of persuasion modes in Arabic debates according to Aristotle's rhetoric.

9. Ethical Statement

In compiling and releasing Munazarat 1.0, rigorous procedures were followed to protect user privacy and obtain consent. The included debates were exclusively sourced from publicly accessible YouTube videos released by participating institutions with debaters' consent. The corpus does not reveal any extra personal data that was not already published publicly. The textual transcripts contain no direct user IDs or handles. Furthermore, the educational institutions that originally published the footage were contacted regarding the research use of this content. Only recordings that we received consent to share in Munazarat 1.0 were included. Those rigorous procedures ensure that, while maximizing the data's research utility, we maintain participant privacy and ethics in compiling and releasing this corpus.

10. Acknowledgements

This work was made possible by two QD Fellowship awards [QDRF-2022-01-003] and [QDRF-2022-01-005] from QatarDebate Center. We would like also to thank the group of native Arab students who contributed to this project by carrying on the task of reviewing text and human validation: Besher Al-Sioufy, Hilmi AbuAlyyan, Moaz Jemmieh, Abdullah Al-Shaar, Jenen Al-Hanai, and Abdullah Al-Kubaisi.

11. Bibliographical References

Datasets | hugging face the ai community building the future. https://huggingface.co/datasets?sort=trending. Accessed: 2024-03-30.

The latest in machine learning | papers with code. https://paperswithcode.com/. Accessed: 2024-03-30.

Ines Abbes, Wajdi Zaghouani, Omaima El-Hardlo, and Faten Ashour. 2020. Daict: A dialectal arabic irony corpus extracted from twitter. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6265–6271.

Ehud Aharoni, Anatoly Polnarov, Tamar Lavee, Daniel Hershcovich, Ran Levy, Ruty Rinott, Dan Gutfreund, and Noam Slonim. 2014. A benchmark dataset for automatic detection of claims and evidence in the context of controversial topics. In *Proceedings of the first workshop on argumentation mining*, pages 64–68.

Abdelhamid Ahmed, Debra Myhill, Esmaeel Abdollahzadeh, Lee McCallum, Wajdi Zaghouani, Lameya Rezk, Anissa Jrad, and Xiao Zhang. 2022a. Qatari corpus of argumentative writing.

Abdelhamid M Ahmed, Xiao Zhang, Lameya M Rezk, and Wajdi Zaghouani. 2024. Building an annotated l1 arabic/l2 english bilingual writer corpus: The qatari corpus of argumentative writing (qcaw). *Corpus-based Studies across Humanities*.

Arfan Ahmed, Nashva Ali, Mahmood Alzubaidi, Wajdi Zaghouani, Alaa Abd-alrazaq, and Mowafa Househ. 2022b. Arabic chatbot technologies: A scoping review. *Computer Methods and Programs in Biomedicine Update*, 2(100057).

Arfan Ahmed, Nashva Ali, Mahmood Alzubaidi, Wajdi Zaghouani, Alaa A Abd-alrazaq, and Mowafa Househ. 2022c. Freely available arabic corpora: A scoping review. *Computer Methods and Programs in Biomedicine Update*, 2(100049).

Mohammad Al-Fetyani, Muhammad Al-Barham, Gheith Abandah, Adham Alsharkawi, and Maha Dawas. 2023. Masc: Massive arabic speech corpus. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 1006–1013. IEEE.

Khalid Al Khatib, Henning Wachsmuth, Kevin Lang, Jakob Herpel, Matthias Hagen, and Benno Stein. 2018. Modeling deliberative argumentation strategies on wikipedia. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2545–2555.

Abdul Gabbar Al-Sharafi, Mohammad Majed Khader, Mohamad Hamza Al-Sioufy, Wajdi Zaghouani, and Ali Al-Zawqari. accepted 2024. A hybrid annotation model for arabic argumentative debate corpus. In *The Eighth International Conference on Arabic Language Processing, ICALP 2023, Rabat, Morocco, April 19–20, 2024*. Springer.

Latifa Al-sulaiti and Eric Atwell. 2006. The design of a corpus of contemporary arabic. *International Journal of Corpus Linguistics*, 11:135–171.

Nora Al-Twairesh, Rawan Al-Matham, Nora Madi, Nada Almugren, Al-Hanouf Al-Aljmi, Shahad Alshalan, Raghad Alshalan, Nafla Alrumayyan, Shams Al-Manea, Sumayah Bawazeer, et al. 2018. Suar: Towards building a corpus for the saudi dialect. *Procedia computer science*, 142:72–82.

Ali Al-Zawqari, Abdul Gabbar Al-Sharafi, Mohamed Ahmed, Mohammad Majed Khader, and Gerd Vandersteen. accepted 2024. Classifying persuasion modes in arabic debates: A preliminary language model-based analysis. In *The Eighth International Conference on Arabic Language Processing, ICALP 2023, Rabat, Morocco, April 19–20, 2024*. Springer.

Khalid Almeman, Mark Lee, and Ali Abdulrahman Almiman. 2013. Multi dialect arabic speech parallel corpora. In *2013 1st International Conference on Communications, Signal Processing, and their Applications (ICCSPA)*, pages 1–6. IEEE.

Maha Alrabiah, A Al-Salman, and ES Atwell. 2013. The design and construction of the 50 million words ksucca. In *Proceedings of WACL'2 Second Workshop on Arabic Corpus Linguistics*, pages 5–8. The University of Leeds.

L Anthony. 2023. Antcconc. https://www.laurenceanthony.net/software/antconc/.

Roy Bar-Haim, Ido Dagan, Bill Dolan, Lisa Ferro, and Danilo Giampiccolo. 2006. The second pascal recognizing textual entailment challenge. In *Proceedings of the second PASCAL challenges workshop on recognizing textual entailment*, volume 7, pages 785–794.

Kaspar Beelen, Timothy Alberdingk Thijm, Christopher Cochrane, Kees Halvemaan, Graeme Hirst, Michael Kimmins, Sander Lijbrink, Maarten Marx, Nona Naderi, Ludovic Rheault, Roman Polyanovsky, and Tanya Whyte. 2017. Digitization of the canadian parliamentary debates. *Canadian Journal of Political Science / Revue canadienne de science politique*, 50(3):pp. 849–864.

Tom Bosc, Elena Cabrio, and Serena Villata. 2016. Dart: A dataset of arguments and their relations on twitter. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1258–1263.

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)*.

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.

Mona Diab, Aous Mansouri, Martha Palmer, Olga Babko-Malaya, Wajdi Zaghouani, Ann Bies, and Mohammed Maamouri. 2008. A pilot arabic propbank. In *Proceedings of the 7th International Conference on Language Resources and Evaluation*.

Esin Durmus, Marco Lippi, and Paolo Torroni. 2021. Argumentation mining on news editorials and blog posts in italian. In *Proceedings of the Eighth Italian Conference on Computational Linguistics (CLiC-it 2021)*, pages 1–6.

Ibrahim Abu El-Khair. 2016. 1.5 billion words arabic corpus. *arXiv preprint arXiv:1611.04033*.

Beatriz Fisas, Francesco Ronzano, and Horacio Saggion. 2016. A multi-layered annotated corpus of scientific papers. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3081–3088, Portorož, Slovenia. European Language Resources Association (ELRA).

Marwa Graja, Maher Jaoua, and L Hadrich Belguith. 2010. Lexical study of a spoken dialogue corpus in tunisian dialect. In *The international arab conference on information technology (acit), benghazi–libya*.

Ivan Habernal and Iryna Gurevych. 2016. Which argument is more convincing? analyzing and predicting convincingness of web arguments using bidirectional lstm. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1589–1599.

Lotte Hargrave and Tone Langengen. 2021. The gendered debate: Do men and women communicate differently in the house of commons? *Politics & Gender*, 17(4):580–606.

Annette Hautli-Janisz, Zlata Kikteva, Wassiliki Siskou, Kamila Gorska, Ray Becker, and Chris Reed. 2022. QT30: A corpus of argument and conflict in broadcast debate. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3291–3300, Marseille, France. European Language Resources Association.

Christopher Hidey and Kathy McKeown. 2018. Persuasive influence detection: The role of argument sequencing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Lindsay Katz and Rohan Alexander. 2023. Digitization of the australian parliamentary debates, 1998-2022. *Scientific Data*, 10.

Mohammad Majed Khader. 2020. A digital study on public speaking: Nlp arguments analysis of the first corpus of arabic debates. Master's thesis, Hamad Bin Khalifa University (Qatar).

Sameer Khurana and Ahmed Ali. 2016. Qcri advanced transcription system (qats) for the arabic multi-dialect broadcast media recognition: Mgb-2 challenge. In *2016 IEEE Spoken Language Technology Workshop (SLT)*, pages 292–298. IEEE.

John Lawrence and Chris Reed. 2019. Argument mining: A survey. *Computational Linguistics*, 45(4):765–818.

John Lawrence and Chris Reed. 2020. Argument mining: A survey. *Computational Linguistics*, 45(4):765–818.

Hamdy Mubarak, Amir Hussein, Shammur Absar Chowdhury, and Ahmed Ali. 2021. Qasr: Qcri aljazeera speech resource–a large scale annotated arabic speech corpus. *arXiv preprint arXiv:2106.13000*.

Elena Musi, Debanjan Ghosh, and Smaranda Muresan. 2018. Towards feasible guidelines for the annotation of argument schemes. In *Proceedings of the First Workshop on Argument Mining*, pages 154–163.

Constantin Orăsan and Richard Evans. 2007. Np animacy identification for anaphora resolution. *Journal of Artificial Intelligence Research*, 29:79–103.

Martha Palmer, Olga Babko-Malaya, Ann Bies, Mona T Diab, Mohamed Maamouri, Aous Mansouri, and Wajdi Zaghouani. 2008. A pilot arabic propbank. In *International Conference on Language Resources and Evaluation*.

Andreas Peldszus and Manfred Stede. 2015. An annotated corpus of argumentative microtexts. In *Argumentation and Reasoned Action: Proceedings of the 1st European Conference on Argumentation, Lisbon*, volume 2, pages 801–815.

Iulian Vlad Serban, Ryan Lowe, Peter Henderson, Laurent Charlin, and Joelle Pineau. 2015. A survey of available corpora for building data-driven dialogue systems. *arXiv preprint arXiv:1512.05742*.

Khaled Shaalan, Sanjeera Siddiqui, Manar Alkhatib, and Azza Abdel Monem. 2019. Challenges in arabic natural language processing. In *Computational linguistics, speech and image processing for arabic language*, pages 59–83. World Scientific.

Sylvia Shaw. 2000. Language, gender and floor apportionment in political debates. *Discourse & society*, 11(3):401–418.

Jacky Visser, Bartosz Konat, Rory Duthie, Marcin Koszowy, Katarzyna Budzynska, and Chris Reed. 2020. Argumentation in the 2016 us presidential elections: Annotated corpora of television debates and social media reaction. *Language Resources and Evaluation*, 54(1):123–154.

Marilyn A Walker, Jean E Fox Tree, Pranav Anand, Rob Abbott, and Joseph King. 2012. A corpus for research on deliberation and debate. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, volume 12, pages 812–817. Istanbul, Turkey.

Wajdi Zaghouani. 2014. Critical survey of the freely available arabic corpora. *OSACT, Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*.

Wajdi Zaghouani, Abdelhamid Ahmed, Xiao Zhang, and Lameya Rezk. 2024. Qcaw 1.0: Building a qatari corpus of student argumentative writing. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING)*.

Wajdi Zaghouani, Mona Diab, Aous Mansouri, Sameer Pradhan, and Martha Palmer. 2010. The revised arabic propbank. In *Proceedings of the fourth linguistic annotation workshop*, pages 222–226.

Wajdi Zaghouani, Nizar Habash, Houda Bouamor, Alla Rozovskaya, Behrang Mohit, Abeer Heider, and Kemal Oflazer. 2015. Correction annotation for non-native arabic texts: Guidelines and corpus. In *Proceedings of The 9th Linguistic Annotation Workshop*, pages 129–139.

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 Language Resources and Evaluation Conference*.

Amy Zhang, Cristian Danescu-Niculescu-Mizil, Jure Lee, Jilin Chen, Tianze Hua, and Dario Taraborelli. 2021. Conversations gone awry: Detecting early signs of conversational failure. *arXiv preprint arXiv:2101.06814*.

Leveraging Corpus Metadata to Detect Template-based Translation: An Exploratory Case Study of the Egyptian Arabic Wikipedia Edition

Saied Alshahrani[1] **Hesham Haroon**[2] **Ali Elfilali**[3] **Mariama Njie**[4] **Jeanna Matthews**[1]
[1]Clarkson University, USA [2]Sesame Labs, Egypt [3]Cadi Ayyad University, Morocco [4]M&T Bank, USA
{saied, jnm}@clarkson.edu, hesham@smsm.ai, a.elfilali9805@uca.ac.ma, mnjie@mtb.com

Abstract

Wikipedia articles (content pages) are commonly used corpora in Natural Language Processing (NLP) research, especially in low-resource languages other than English. Yet, a few research studies have studied the three Arabic Wikipedia editions, Arabic Wikipedia (AR), Egyptian Arabic Wikipedia (ARZ), and Moroccan Arabic Wikipedia (ARY), and documented issues in the Egyptian Arabic Wikipedia edition regarding the massive automatic creation of its articles using template-based translation from English to Arabic without human involvement, overwhelming the Egyptian Arabic Wikipedia with articles that do not only have low-quality content but also with articles that do not represent the Egyptian people, their culture, and their dialect. In this paper, we aim to mitigate the problem of template translation that occurred in the Egyptian Arabic Wikipedia by identifying these template-translated articles and their characteristics through exploratory analysis and building automatic detection systems. We first explore the content of the three Arabic Wikipedia editions in terms of density, quality, and human contributions and utilize the resulting insights to build multivariate machine learning classifiers leveraging articles' metadata to detect the template-translated articles automatically. We then publicly deploy and host the best-performing classifier, XGBoost, as an online application called Egyptian Wikipedia Scanner♣ and release the extracted, filtered, and labeled datasets to the research community to benefit from our datasets and the online, web-based detection system.

Keywords: Arabic, Egyptian, Moroccan, Wikipedia, Template Translation, Multivariate Classification

1. Introduction

Wikipedia articles are widely used as pre-training datasets for many Natural Language Processing (NLP) tasks like language modeling (language models) and word representation (word embedding models) tasks, especially for low-resource languages like Arabic, due to its large collection of multilingual content and its vast array of metadata that can be quantified and compared (Mittermeier et al., 2021). However, not all Wikipedia articles are organically produced by native speakers of those languages; while humans have naturally generated some articles in those languages, many others have been automatically generated using bots or automatically translated from high-resourced languages like English without human revision using off-the-shelf automatic translation tools like Google Translate[1] (Hautasaari, 2013; Nisioi et al., 2016; Baker, 2022; Alshahrani et al., 2022; Johnson and Lescak, 2022; Bhattacharjee and Giner, 2022; Wikimedia Foundation, 2022).

A few researchers have addressed this issue and highlighted its implications for NLP systems and tasks. For example, Alshahrani et al. (2022) have studied the three Arabic Wikipedia editions, Arabic Wikipedia (AR), Egyptian Arabic Wikipedia (ARZ), and Moroccan Arabic Wikipedia (ARY), and documented issues in the Egyptian Wikipedia with automatic creation/generation and translation of content pages from English without human supervision. They stressed that these issues could substantially affect the performance and accuracy of Large Language Models (LLMs) trained from these corpora, producing models that lack the cultural richness and meaningful representation of native speakers. In another research work by the same authors, they investigated the performance implications of using inorganic, unrepresentative corpora, mainly generated through automated techniques such as bot generation or automated template-based translation, to train a few masked language models and word embedding models. They found that models trained on bot-generated or template-translated articles underperformed the models trained on human-generated articles and underscored that, for good NLP performance, researchers need both large and organic corpora (Alshahrani et al., 2023a).

In this paper, we solely focus on the problem of template translation that took place in the Egyptian Arabic Wikipedia edition, where a few registered users employed simple templates to translate more than one million content pages (articles) from English to Arabic using Google Translate, all without translation error checking or culture misrepresentation verification, disregarding the consequences of using such poor articles (Baker, 2022; Das, 2020; Alshahrani et al., 2022; Agrawal et al., 2023; Al-Khalifa et al., 2024; Thompson et al., 2024). We first explore the three Arabic Wikipedia editions' content in terms of density, quality, and human contributions, highlighting how the template-based

♣https://hf.co/spaces/Egyptian-Wikipedia-Scanner.
[1]Google Translate: https://translate.google.com.

OSACT 2024 Workshop, pages 31–45

translation occurred on the Egyptian Wikipedia produces unrepresentative content. We second, attempt to build powerful multivariate machine learning classifiers leveraging corpus/articles' metadata to detect the template-translated articles automatically. We then deploy and publicly host the best-performing classifier, XGBoost, so researchers, practitioners, and other users can benefit from our online, web-based detection system. We lastly argue that practices such as template translations could not only impact the performance of models trained on these template-translated articles but also could misrepresent the native speakers and their culture and do not echo their views, beliefs, opinions, or perspectives.

2. Exploratory Analysis

We explore, in the following subsections, the three Arabic Wikipedia editions, Arabic Wikipedia (AR), Egyptian Arabic Wikipedia (ARZ), and Moroccan Arabic Wikipedia (ARY), regarding their articles' content in terms of density, quality, and human contributions.

2.1. Analysis Setup

We follow the same methodology Alshahrani et al. (2023a) used to quantify the bot-generated articles, but we, here, utilize the Wikimedia `XTools` API[2] to collect all Arabic Wikipedia editions' articles' metadata; specifically, we collect the total edits, total editors, top editors, total bytes, total characters, total words, creator name, and creation date for each article. We use the complete Wikipedia dumps of each Arabic Wikipedia edition, downloaded on the 1st of January 2024 (Wikimedia, 2024) and processed using the `Gensim` Python library (Řehůřek and Sojka, 2010). We also use Wikipedia's "List Users" service[3] to retrieve the full list of bots in each Arabic Wikipedia edition to measure the bot and human contributions to each article.

2.2. Shallow Content

We, in this subsection, study the density of the content of the three Arabic Wikipedia editions, highlighting general statistics and token/character length distributions per Arabic Wikipedia edition.

2.2.1. Summary Statistics

We shed light on a few general statistics of the three Arabic Wikipedia editions regarding their total articles, total extracted articles, corpus size, total bytes, total characters, and total tokens, highlighting the minimum, maximum, and mean values of the three articles' metadata: total bytes, total characters, and total tokens.[4] From Table 1, it is notable that the Egyptian Arabic Wikipedia has a greater number of total articles than the Arabic Wikipedia (which is generally believed to be more organically generated), with almost 400K articles, yet as we will discuss later in Table 3, this number of total articles does not reflect true measurements of organically generated contributions since all three Arabic Wikipedia editions include substantial bot generation and template translation activities (Baker, 2022; Alshahrani et al., 2022, 2023b). We employ the `Gensim` Python library to parse and extract the textual content (articles) from each Wikipedia dump file. However, since the library discards any articles with less than 50 tokens/words, all three Arabic Wikipedia editions lost many articles. For example, the Egyptian Wikipedia lost nearly 741K (46%) of its articles, whereas the Moroccan Wikipedia and the Arabic Wikipedia lost 2.9K (30%) and 346K (28%) of their articles, respectively. This loss of articles exhibits how the Egyptian Arabic Wikipedia contains almost half of its total articles under 50 tokens per article, indicating that it has more limited and shallow content and reflecting the template translation that occurred on its articles.

2.2.2. Token/Character Length Distribution

We visualize, in Figure 1, the token and character distributions for each Arabic Wikipedia edition by plotting the tokens per article and characters per article with the mean lines for each Arabic Wikipedia edition. We observe that the Egyptian Wikipedia length distributions (token and character) are less dense than the Arabic Wikipedia and Moroccan Wikipedia, and a notable number of articles in the Egyptian Wikipedia are below the mean line/threshold, exhibiting that the Egyptian Wikipedia has unusually smaller and shorter articles than other Arabic Wikipedia editions. Surely, the Egyptian Wikipedia has more articles than the other Arabic Wikipedia editions, but it does have the lowest mean values of the total of characters and total of tokens/words, 610 and 100, respectively, compared to the mean values of the Arabic Wikipedia and the Moroccan Wikipedia, as shown in Table 1. These observations signal that the template translation that happened on its articles does not produce rich, dense, and long content but only produces poor, limited, and shallow content.

[2]XTools API: https://www.mediawiki.org/wiki/XTools.
[3]https://{WIKI}.wikipedia.org/wiki/Special:ListUsers.
[4]We use the Wikimedia Statistics service, https://stats.wikimedia.org, to retrieve the total articles (content pages) for each Arabic Wikipedia edition, whereas the other statistics are generated from the extracted articles from each Arabic Wikipedia edition.

Wikipedia	Total Articles	Extracted Articles	Corpus Size	Total Bytes			Total Characters			Total Tokens		
				Min	Max	Mean	Min	Max	Mean	Min★	Max	Mean
Arabic (AR)	1,226,784	880,334	2.6GB	6,424,572,842			1,564,243,778			264,761,062		
				488	1,419,547	7,297	200	334,464	1,776	50	56,395	300
Egyptian (ARZ)	1,621,745	736,158	766MB	1,525,938,072			449,449,693			74,277,188		
				515	1,217,036	2,072	233	399,641	610	50	74,009	100
Moroccan (ARY)	9,659	6,754	11MB	25,109,824			6,802,694			1,153,946		
				646	105,009	3,717	248	32,853	1,007	50	5,635	170

Table 1: General statistics of the three Arabic Wikipedia editions in terms of total articles, total extracted articles, corpus/articles size, total bytes, total characters, and total tokens. ★As a result of the `Gensim` Python library discarding articles with tokens/words less than 50, all minimum tokens of articles are 50.

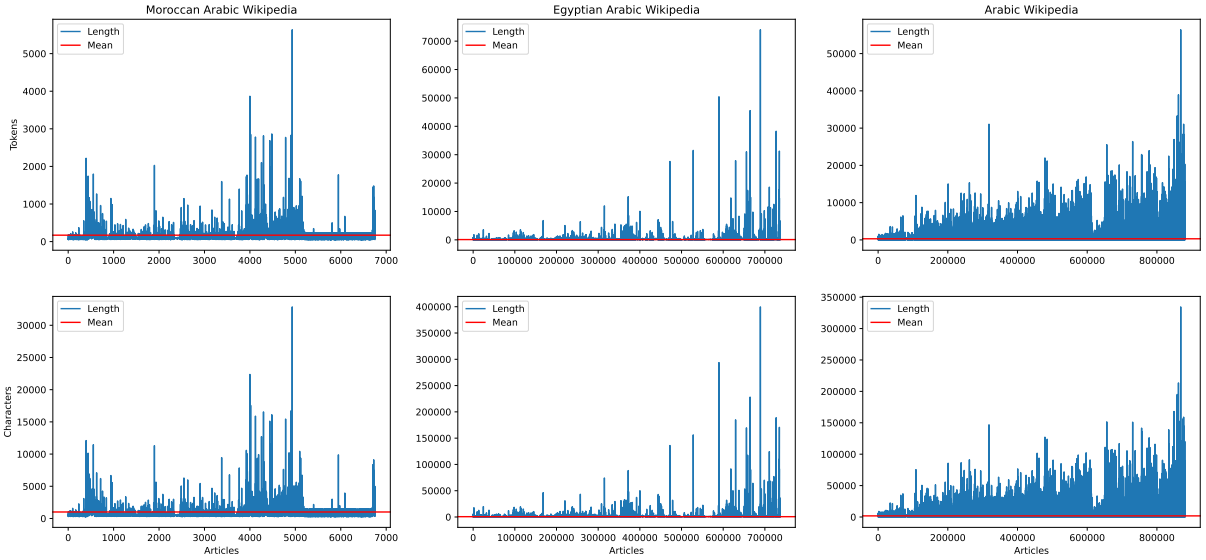


Figure 1: Visualizations of tokens and characters per article for each Arabic Wikipedia edition, displaying the total tokens and characters on the y-axes and articles on the x-axes, with plotting the mean lines.

2.3. Poor Quality Content

We study the quality of the Arabic Wikipedia editions' content regarding lexical richness and diversity and the most common and duplicate n-grams.

2.3.1. Lexical Richness/Diversity

We use the terms lexical richness and lexical diversity equivalently and interchangeably in this study, as Daller et al. (2003) suggested. To measure the lexical richness and diversity, we first compute the total tokens and unique tokens per Arabic Wikipedia edition, and second, we utilize three simple but widely used lexical richness metrics: Type-Token Ratio (TTR) (Chotlos, 1944; Templin, 1957), Root Type-Token Ratio (RTTR) (Guiraud, 1954, 1959), and Corrected Type-Token Ratio (CTTR) (Carroll, 1964). Yet, as many have emphasized, like McCarthy (2005), we find that these metrics are not often precise and sometimes erroneous and do not reflect the true lexical richness and diversity of a corpus. For example, we observe that the TTRs of Arabic Wikipedia and Egyptian Wikipedia are identical, and the RTTRs and CTTRs of Egyptian Wikipedia and Moroccan Wikipedia are similar, despite the massive difference between the Arabic Wikipedia editions' corpora in terms of the lexicon size and vocabulary size, as shown in Table 2. Therefore, we adopt an advanced metric to measure the lexical richness and diversity called 'Measure of Textual Lexical Diversity (MTLD)', introduced by Mccarthy and Jarvis (2010). We utilize the `LexicalRichness` Python library's implementation of the MTLD metric with a default factor size of 0.720 (Shen, 2022). We find that the results are consistent with the other metrics, as reported in Table 2, in that the Moroccan Wikipedia has the best lexical richness and diversity among the three Arabic Wikipedia editions, where the Arabic Wikipedia comes second, and Egyptian Wikipedia comes in last, documenting the Egyptian Arabic Wikipedia corpus is not lexically rich and diverse, which we attribute to the template-based translation took place on its articles (content pages).

2.3.2. Most Common/Duplicate N-Grams

We generate n-grams from each Arabic Wikipedia corpus, where n={1, 2, 3, 5, 10, 50}, to highlight the common and duplicate n-grams. We hypothesize that the higher the count of n-grams in an Arabic Wikipedia corpus, especially when n={5, 10, 50}, the more we can detect templates used in the template translation activities in the Arabic Wikipedia

Wikipedia	Total Tokens	Unique Tokens	Type-Token Ratio (TTR)	Root Type Token Ratio (RTTR)	Corrected Type Token Ratio (CTTR)	Measure of Textual Lexical Diversity (MTLD)
Arabic (AR)	264,777,392	2,867,782	0.010	176.24	124.62	71.20
Egyptian (ARZ)	74,278,320	759,519	0.010	88.12	62.31	45.69
Moroccan (ARY)	1,154,058	94,827	0.082	88.27	62.41	89.77

Table 2: Calculations of four lexical richness and diversity metrics, TTR, RTTR, CTTR, and MTLD, accompanied with total tokens (lexicon) and unique tokens (vocabulary) for each Arabic Wikipedia edition.

editions, specifically in the Egyptian Wikipedia. We notice that n-grams in the Egyptian Wikipedia have very large counts compared to the Arabic and Moroccan Wikipedia editions, as shown in Tables 9 and 10 in Appendix A.[5] In Figure 2, we visualize the log values of the top K=1 counts of common and duplicate n-grams generated from each Arabic Wikipedia corpus, where n={1, 2, 3, . . . , 50}, and we observe that all the n-grams in all the Arabic Wikipedia editions exhibit exponential decay, drastically (like Arabic Wikipedia) or gradually (like Egyptian Wikipedia and Moroccan Wikipedia). Yet, the large counts of Egyptian Wikipedia's n-grams when n>={5} do not decline exponentially but linearly, suggesting that there is an anomaly in the Egyptian Wikipedia corpus, where the n-grams of the normally generated corpus by humans usually factorially decreases, as the n value increases. We believe the template-based translation on the Egyptian Wikipedia creates such an anomaly, as many parts/grams/phrases of templates used in the translation are duplicated repeatedly in its corpus.

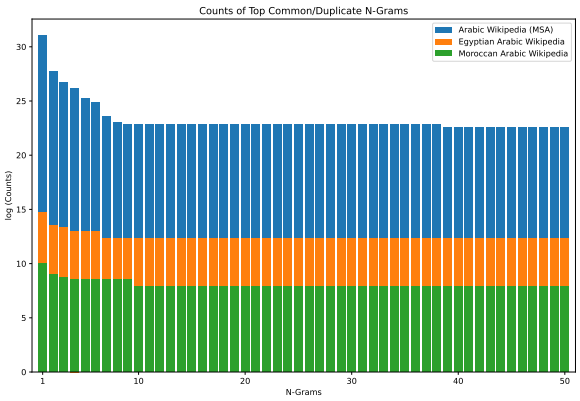


Figure 2: Counts of top common/duplicate n-grams of each Arabic Wikipedia edition; log values/counts are only for top K=1 common/duplicate n-grams.

2.4. Misleading Human Involvement

We shed light on the human involvement across the three Arabic Wikipedia editions, specifically the type of page creators and editors, debating how the template translation activities could produce misleading metadata regarding human involvement.

[5]We further analyze the 5-grams and 10-grams of each Arabic Wikipedia edition in Appendix A.

2.4.1. List of Contributors

We collect all the page creators for each article in the Arabic Wikipedia editions, count the number of their contributions (article creations), and categorize them into bots and humans. As shown in Table 3, it is clear that the Arabic Wikipedia and Moroccan Wikipedia suffer from mass auto-creation of articles by bots, especially by the 'JarBot', which has created nearly 260K articles (29.31%) in the Arabic Wikipedia, and the 'DarijaBot', which has created nearly 3.2K articles (34%) in the Moroccan Wikipedia.[6] However, the worst of all is the unguided, unreviewed, unsupervised template translation of articles from English in the Egyptian Wikipedia by registered users, largely by two registered users, 'HitomiAkane' and 'Al-Dandoon', who have created more than 1.4M articles (88.57%) and 113K articles (6.99%), respectively.[7]

2.4.2. Type of Contributors

We calculate the percentage of creators and editors of articles (bots and humans) in each Arabic Wikipedia edition. We use the absolute count of page creators and classify the creators based on their types, bots or humans, while with the page editors, we calculate the percentage using the total number of editors on each article and set a threshold of 50%, where if an article was edited by more than 50% by bots, we then consider this article a bot-edited, and vice versa. As shown in Figure 3, we see bots often create articles side-by-side with humans in the Arabic Wikipedia (31.5%) and Moroccan Wikipedia (22.30%) editions, which is normal and permitted to a certain degree according to Wikipedia's bot policy (Wikipedia, 2024b). However, in the Egyptian Wikipedia edition, we observe that its articles are 100% created by humans, i.e., registered users, and this percentage is misleading given that 42.72% of its articles are

[6]These two bots, 'JarBot' and 'DarijaBot', have approval from Wikimedia to operate on the Arabic Wikipedia and the Moroccan Wikipedia (Wikidata, 2024b,a).

[7]These two registered users were local admins of the Egyptian Arabic Wikipedia edition until their permissions were revoked in May 2020 by the Stewards, the global admins of the Wikipedia project, for their abuse of admin permissions and their massive unsupervised and unauthorized creation of articles (Wikipedia, 2020).

[8]Wikiscan Statistics service: https://wikiscan.org.

Wikipedia \ Rank (percentage)		1st (%)	2nd (%)	3rd (%)	4th (%)	5th (%)
Arabic (AR)	**Name**	JarBot	Mr. Ibrahem	جار الله	ElphiBot	Majed
	Count	359,677 (29.31%)	52,222 (4.25%)	43,691 (3.56%)	42,669 (3.47%)	26,228 (2.13%)
	Type	Bot	Human	Human	Bot	Human
Egyptian (ARZ)	**Name**	HitomiAkane	Al-Dandoon	Raafat	Ghaly	حمدى10
	Count	1,436,430 (88.57%)	113,468 (6.99%)	18,334 (1.13%)	7,212 (0.44%)	2,720 (0.16%)
	Type	Human	Human	Human	Human	Human
Moroccan (ARY)	**Name**	DarijaBot	Tifratin	Ideophagous	Sedrati	Rachidourkia
	Count	3,285 (34%)	1,302 (13.47%)	1,231 (12.74%)	765 (7.92%)	540 (5.59%)
	Type	Bot	Human	Human	Human	Human

Table 3: Top five page creators in the Arabic Wikipedia editions, highlighting their types (bots or humans) and how many articles they have created until March 1st, 2024, according to Wikiscan Statistics service.[8]

automatically template-translated from English to Arabic using templates without human supervision or intervention, as documented by Baker (2022) and Alshahrani et al. (2022).

3. Experimental Setup

We, here, attempt to build classifiers to identify and mitigate the impacts of the template-translated articles on the Egyptian Wikipedia edition since it particularly suffers from template translations, as documented by Alshahrani et al. (2022). We first extract all articles with their metadata, split the articles into two categories: before and after the template-based translation occurred, and lastly, label, preprocess, and encode these categorized articles using Arabic pre-trained models.

3.1. Dataset Filtrating and Labeling

We follow a few heuristic rules to classify Egyptian Wikipedia into articles created before and after the massive template-based translation activities related to creation dates, total edits, and types of creators and editors. We take insights from our exploratory analysis, section 2, the Wikimedia Statistics service, and the previous research works that documented the template translation activities in the Egyptian Wikipedia (Baker, 2022; Alshahrani et al., 2022; Wikimedia Statistics, 2024), to craft these rules, specifically when selecting the dates.

Category	Total
Total Articles (both categories)	736,107
Articles Before Template Translation	11,126
Articles After Template Translation	155,275
Uncategorized Articles	569,706

Table 4: Statistics of filtered articles after applying our heuristic filtration rules, displaying the totals.

We list the heuristic rules for filtering the articles created *before* and *after* the translations in Appendix B, where we employ more rigorous heuristic rules to filter the articles created *after* the template translation appeared on the Egyptian Wikipedia. In Table 4, we show the statistics of our rule-based filtration process. We then randomly select 10K articles from each category to train a multivariate machine learning classifier to detect the template-based translations automatically. We lastly label the articles *before* translation as 'human-generated' articles since all articles are created by registered users and label the articles *after* translation as 'template-translated' articles.

3.2. Dataset Preprocessing

We lightly preprocess the filtered articles by replacing all non-alphanumeric and non-Arabic characters with white spaces and normalizing the extra unnecessary whitespaces to one whitespace. We do not apply stemming, lemmatization, or any Arabic text normalization on the articles to have organic content (articles) as much as possible.

3.3. Dataset Encoding

We use two different types of embedding techniques to encode the randomly selected 20K articles separately: pre-trained Egyptian Arabic context-independent word embeddings (Word2Vec) of the size of 300 dimensions from `Spark-NLP` Python library[9] and context-dependent word embeddings (contextual) of the size of 768 dimensions produced by utilizing the pre-trained CAMeLBERT-Mix POS-EGY model[10] (Inoue et al., 2021) as our feature extraction model. The goal is to test with different embedding techniques to maximize the performance of our multivariate machine learning classifiers and investigate how the type and size of the word embeddings would affect their performance.

4. Template Translation Detection

We experiment with a few supervised classification algorithms and unsupervised clustering algorithms

[9]Word2Vec Embeddings in Egyptian Arabic (300d): https://sparknlp.org/2022/03/14/w2v_cc_300d_arz_3_0.

[10]CAMeLBERT-Mix POS-EGY model: https://huggingface.co/CAMeL-Lab/bert-base-arabic-camelbert-mix-pos-egy.

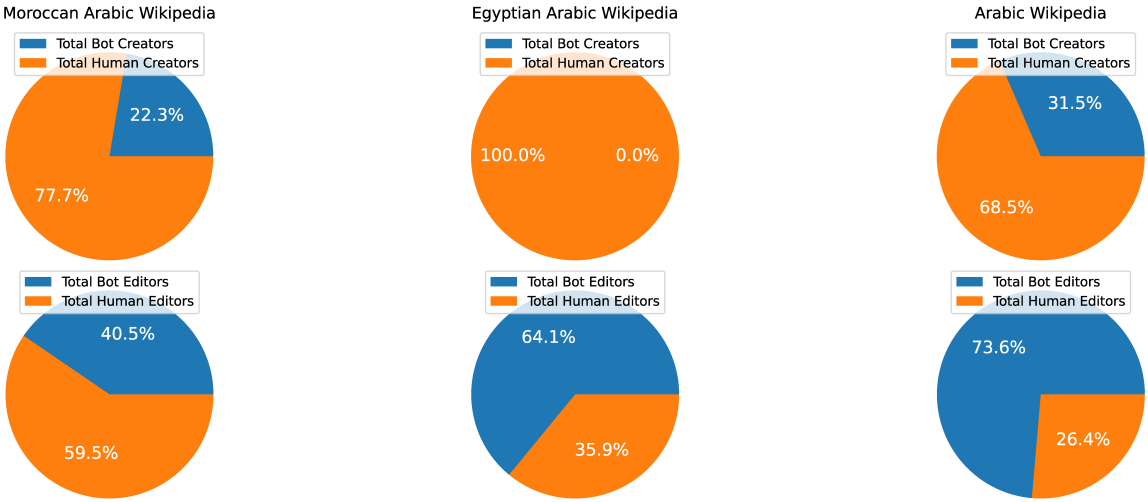


Figure 3: Visualizations displaying the percentage of article creators and editors in terms of their types, bots, and humans, and their number of contributions (article creations) in each Arabic Wikipedia edition.

to determine which approach and algorithm will best solve our template-based translation problem.

4.1. Input Features Extraction

We aim to leverage the metadata of corpus, i.e., articles, collected using Wikimedia services to detect the template-translated articles in the Egyptian Wikipedia edition. Besides utilizing pre-trained Word2Vec and CAMeLBERT word embeddings as input features, we also include the metadata we collect about every article: total edits, total editors, total bytes, total characters (charts), and total words. Overall, we test the machine learning algorithms' performance using three input features: only embeddings, only metadata, or both (metadata and embeddings), as illustrated in Figure 4.

4.2. Metadata Ablation Studies

We perform two ablation studies for each machine learning algorithm (classification and clustering) to determine the best metadata features to include in the input features. We first test each metadata feature's performance individually and then combine two, three, and all metadata features consecutively.

4.3. Classification Algorithms

We select five supervised classification algorithms to solve our multivariate classification problem: Logistic Regression (LR) (Fan et al., 2008), Support Vector Machine (SVM) (Chang and Lin, 2011), Gaussian Naive Bayes (GNB) (Pedregosa et al., 2011), Random Forests (RF) (Breiman, 2001), and XGBoost (eXtreme Gradient Boosting) (XGBoost, 2024). We, in the next subsections, discuss the experimental setups and the performance results of these supervised machine learning classifiers.

4.3.1. Classification Experimental Setup

We split the randomly selected 20K articles into training (80%) and testing (20%) splits with data shuffling and stratification enabled to ensure that the training and test splits are randomized and have the same proportion of each class. We further evaluate our classifiers using the accuracy metric with the Stratified K-Folds Cross-Validation technique, where we set the number of folds K=5, ensuring every fold has a representative class distribution.

4.3.2. Results of Classification Ablations

We report, in Table 5, the evaluation accuracy results on the testing splits of our metadata ablations. We can see that all machine learning classifiers achieve excellent (100%) to very good performance (100%>accuracy>90%) with the total edits and total editors separated or combined. In contrast, metadata features like the total bytes, total characters, and total words perform from fairly to poorly and, unfortunately, decrease the overall performance of all metadata features combined with some classifiers like SVM. Generally, we observe that the ensemble classifiers (RF and XGBoost) outperform the other classifiers even with the metadata features that contribute less to the classifiers' learning.

4.3.3. Results of Classification Algorithms

We show, in Table 6, the evaluation accuracy scores on the testing splits of the multivariate machine learning classifiers studied, demonstrating how the classifiers would perform with three input features: two embedding styles (Word2Vec or CAMeLBERT), corpus/articles metadata, and both embeddings and metadata combined. Here, we decided to include all the articles' metadata, not only

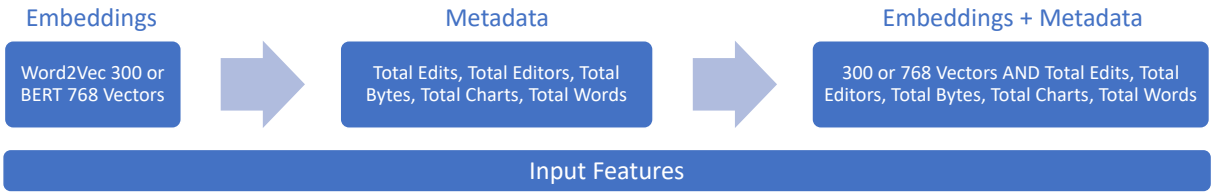


Figure 4: A basic process chart demonstrating the studied input features: embeddings (two word embeddings of sizes 300 or 768), metadata (five metadata of articles), or both (embeddings + metadata).

Classifier	Metadata							
	A	B	C	D	E	A+B	C+D+E	All
Logistic Regression	100	100	88.30	83.85	84.67	100	89.03	98.42
Support Vector Machine	90.30	100	87.95	83.60	83.95	99.78	87.62	87.75
Naive Bayes	100	100	82.00	74.28	78.00	100	80.50	99.60
Random Forest	100	100	86.17	82.23	84.80	100	91.25	100
XGBoost	100	100	88.60	84.52	84.70	100	90.53	100

Table 5: Accuracies of metadata ablations of the studied classifiers. Encoded columns denote metadata features as follows: A) total edits, B) total editors, C) total bytes, D) total characters, and E) total words.

Classifier	Embeddings		Metadata	Both (Embeddings + Metadata)	
	Word2Vec	CAMeLBERT		Word2Vec	CAMeLBERT
Logistic Regression	91.22	99.30	98.42	99.40	100
Support Vector Machine	99.02	98.45	87.75	87.90	87.90
Naive Bayes	88.90	95.17	99.60	99.60	99.52
Random Forest	98.08	98.17	100	100	99.95
XGBoost	98.28	98.78	100	100	100

Table 6: Accuracies of the machine learning classifiers studied, showing their performance with different input features: two embedding styles, corpus metadata, and both embeddings and metadata combined.

the features that performed well in our ablation studies, to diversify the classifiers' learning and ensure that each category of the Egyptian Wikipedia articles (human-generated and template-translated) is well-represented. We report, again, that the SVM classification algorithm underperforms all the other algorithms and find that the metadata features present a bottleneck performance for it (i.e., highly variable features). We attribute the poor performance to the complex, multivariate nature of the dataset, specifically, the high variability of the metadata features like the total bytes, words, and characters, as seen in Table 5.[11] On the other bright side, we find that ensemble classification algorithms like RF and XGBoost excel and outperform the traditional, single classification algorithms due to their ability to overcome noise, bias, and variance; the RF algorithm uses the bagging technique, and XGBoost algorithm uses boosting technique to handle such technical challenges.[12]

[11]We handled the dataset noise through our filtration process and the bias by balancing the dataset classes, yet the dataset variance is challenging due to the high dispersion in metadata features collected.

[12]As an online application, we deploy our best classifier, XGBoost, with input features of metadata and CAMeLBERT embeddings. See Appendix C for details.

4.4. Clustering Algorithms

We explore three different unsupervised clustering algorithms to solve the template-based translation problem: K-Means (Wu, 2012), Hierarchical Agglomerative (Zepeda-Mendoza and Resendis-Antonio, 2013), and DBSCAN (Density-Based Spatial Clustering of Applications with Noise) (Ester et al., 1996). We, in the following, discuss the experimental setups and the performance results of these unsupervised machine learning clusterers.

4.4.1. Clustering Experimental Setup

We feed the unsupervised clustering algorithms all the randomly selected 20K articles after removing the labels without splitting them due to their nature. We set the number of clusters to K=2 since our dataset only has two categories (human-generated and template-translated). We further evaluate our clusterers using the Silhouette coefficient with the Euclidean distance, a widely used internal evaluation metric to measure how cohesive and separated the clusters are, based on the distances or similarities between the data points, i.g., articles.[13]

[13]Values of the Silhouette coefficient are always between 1 and -1. We apply a percentage normalization

4.4.2. Results of Clustering Ablations

We report the Silhouette scores of our metadata ablation studies in Table 7. We can see that all machine learning clusterers achieve great performance with the total bytes, total characters, and total words, separated or combined, except for the DBSCAN algorithm. In contrast, metadata features like the total edits and total editors perform from fairly to poorly with K-Means and Hierarchical clustering algorithms, except for the DBSCAN algorithm. The results of these metadata ablations indicate an opposite behavior from those discussed in subsection 4.3.2, where the previously weak metadata features for the classification algorithms, like the total bytes, words, and characters, became strong metadata features for the clustering algorithms instead of the total edits and editors, which were previously strong. Generally, the K-Means and Hierarchical clustering algorithms outperform the DBSCAN algorithm even with the metadata features that contribute more to the clusterers' learning.

4.4.3. Results of Clustering Algorithms

We show, in Table 8, the Silhouette scores of the machine learning clusterers studied, demonstrating how the unsupervised clusterers would perform with three input features: two embedding styles (Word2Vec or CAMeLBERT), corpus/articles metadata, and both embeddings and metadata combined. We, here, fit all the articles' metadata, not only the features that performed well in our ablation studies, to diversify the clusterers' learning and ensure that each class of the Egyptian Arabic Wikipedia articles (human-generated and template-translated) is included. We report that all the clustering algorithms perform poorly with the word embeddings as features, whereas the metadata features present a performance improvement. We assume clustering the word embeddings is challenging, especially with their large dimensionality; Word2Vec's size is 300, and CAMeLBERT's is 768. Overall, the unsupervised clustering algorithms underperform the supervised classification algorithms, yet we can confirm that the clustering algorithms do better with low-dimensionality features like articles' metadata, even though they introduce high-variable and dispersed features.

5. Discussion

We discuss three negative implications of the unguided, unreviewed, unsupervised template-based translation from English to Arabic on the Egyptian Wikipedia articles: societal, representation, and performance implications. On the societal implications, we argue that using off-the-shelf-translation tools like Google Translate, which is widely known for its social problems like gender, cultural, and religious biases and stereotypes, could not only cause linguistic and grammatical errors but also amplify these social risks like biases and stereotypes (Prates et al., 2020; Ullmann and Saunders, 2021; Lopez-Medel, 2021; Naik et al., 2023; Al-Khalifa et al., 2024). Many researchers have emphasized how unsupervised translations are prone to serious gender bias issues, like producing translations with inaccurate gender, that could impact native speakers. For example, Stanovsky et al. (2019) have automatically evaluated the gender bias for eight highly-gendered languages like Arabic and found that a few popular industrial and academic machine translation systems (like Google Translate and Microsoft Translator[14]) were significantly prone to gender-biased translation errors for all tested target languages. We believe those machine translation systems are greatly beneficial tools, yet they should not be used to naively, directly, or automatically translate content without human review, especially if the content is related to the societal representation of Arabic native speakers.

On the representation implications, we argue that such automatic template-based translations without humans in the loop could misrepresent the Egyptian Arabic native speakers, where instead of the Egyptian people enriching the content of Wikipedia by sharing their voices, opinions, knowledge, perspectives, and experiences, a couple of registered users automated the creation and translation of more than a million and a half million articles (95.56%) from English on their behalf without supervision or revision of the translated articles, disregarding that the main goal of Wikipedia is to be written by the people to the people (Cohen, 2008). Another troubling drawback of such a practice is the cultural misrepresentation of the Egyptian people and their culture, where the unfiltered and unsupervised translation from English could introduce content that is not representative of the culture of native speakers. Lastly, we argue that including culturally unrepresentative articles from the Egyptian Arabic Wikipedia in pre-training corpora for language models could present cultural implications and generate culturally misaligned outputs from these models, where the majority of Arabic and multilingual language models have been fundamentally pre-trained on Wikipedia dumps like Jais and Jais-chat (Sengupta et al., 2023), AraMUS (Alghamdi et al., 2023), and JASMINE (Nagoudi et al., 2023). We believe research works, like ours, that automatically identify these template-translated articles could promote data transparency and help

(multiply values by 100) when reporting the values to draw a head-to-head comparison between algorithms.

[14] Microsoft Bing: https://www.bing.com/translator.

Clusterer	Metadata							
	A	B	C	D	E	A+B	C+D+E	All
K-Means	82.68	78.32	97.10	96.46	96.39	81.77	96.89	96.89
Hierarchical	86.85	81.42	97.10	97.37	97.32	82.28	96.08	97.52
DBSCAN	97.80	99.62	37.20	67.58	89.79	77.11	68.35	68.33

Table 7: Silhouette scores of the metadata ablations of the studied clusterers. Encoded columns denote metadata features: A) total edits, B) total editors, C) total bytes, D) total characters, and E) total words.

Clusterer	Embeddings		Metadata	Both (Embeddings + Metadata)	
	Word2Vec	CAMeLBERT		Word2Vec	CAMeLBERT
K-Means	12.50	14.95	96.89	96.89	96.89
Hierarchica	11.79	10.82	97.52	96.77	96.77
DBSCAN	61.64	8.43	68.33	68.34	68.68

Table 8: Silhouette scores of the machine learning clusterers studied, showing their performance with different features: two embedding styles, corpus/articles metadata, and both embeddings and metadata.

researchers make an informed decision about what to include in their pre-training datasets/corpora.

On the performance implications, we argue that the template-based translations that occurred on the Egyptian Wikipedia produce not only short and shallow articles, where we have reported that nearly 46% of the Egyptian Wikipedia articles are less than 50 tokens/words and recognized a large number of duplicate n-grams due to the templates used in translations, but also articles that lack lexical richness and diversity, where we have found that the Egyptian Wikipedia scored the worst among other Arabic Wikipedia editions in the MTLD metric. These poorly translated articles could negatively impact the performance of language models and NLP tasks that are trained on them. One research that supports our claim is the recent work of Alshahrani et al. (2023a), where they documented that models trained on the template-translated articles of the Egyptian Wikipedia performed the worst when compared with the models trained on the Arabic Wikipedia articles. Finally, we recommend excluding the unfiltered template-translated articles from Egyptian Wikipedia from training datasets to mitigate their negative societal, representation, and performance implications and encourage using automatic detection systems, like ours, to identify such articles that are not only mispicturing the Egyptian people and their culture but also affecting the performance of language models and NLP tasks.

6. Limitations

We leverage five metadata of articles of different sizes (total edits, total editors, total bytes, total characters, and total words) and then append them to two types of word embeddings (Word2Vec and CAMeLBERT) of sizes of 300 or 768 vectors to build powerful classifiers, yet concatenating all these different features could produce highly variable features due to the high dispersion between the extracted input features, which could present a performance challenge for our proposed automatic detection system and could increase the non-deterministic behavior of its classifiers.

7. Conclusion

We attempt to mitigate the template translations on the Egyptian Arabic Wikipedia by identifying these template-translated articles and their characteristics through exploratory analysis and developing automatic detection systems. We first investigate the content of the three Arabic Wikipedia editions in terms of density, quality, and human contributions and use such insights to build powerful multivariate machine learning classifiers leveraging articles' metadata to detect template-translated articles automatically; we find that the supervised classification algorithms are better than the unsupervised clustering algorithms. We then publicly deploy the best-performing classifier, XGBoost, as an application and release the extracted, filtered, labeled, and preprocessed datasets to the community to benefit from our datasets and the online detection system.

Reproducibility

We share our labeled datasets, code and scripts of the exploratory analysis, and the multivariate machine learning classifiers on GitHub at `https://github.com/SaiedAlshahrani/leveraging-corpus-metadata`.

Acknowledgments

We would like to thank Clarkson University and the Office of Information Technology (OIT) for providing computational resources. We also would like to thank Norah Alshahrani for her valuable feedback.

Bibliographical References

Ameeta Agrawal, Lisa Singh, Elizabeth Jacobs, Yaguang Liu, Gwyneth Dunlevy, Rhitabrat Pokharel, and Varun Uppala. 2023. All Translation Tools Are Not Equal: Investigating the Quality of Language Translation for Forced Migration. In *2023 IEEE 10th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–10.

Hend Al-Khalifa, Khaloud Al-Khalefah, and Hesham Haroon. 2024. Error Analysis of Pretrained Language Models (PLMs) in English-to-Arabic Machine Translation. *Human-Centric Intelligent Systems*.

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. *arXiv preprint arXiv:2306.06800*.

Saied Alshahrani, Norah Alshahrani, Soumyabrata Dey, and Jeanna Matthews. 2023a. Performance Implications of Using Unrepresentative Corpora in Arabic Natural Language Processing. In *Proceedings of ArabicNLP 2023*, pages 218–231, Singapore (Hybrid). Association for Computational Linguistics.

Saied Alshahrani, Norah Alshahrani, and Jeanna Matthews. 2023b. 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 (Hybrid). Association for Computational Linguistics.

Maher Asaad Baker. 2022. *How I Wrote a Million Wikipedia Articles*, 2 edition. BookRix GmbH & Co. KG., Munich, Germany.

Runa Bhattacharjee and Pau Giner. 2022. You Can Now Use Google Translate to Translate Articles on Wikipedia. Last accessed on 2024-03-01.

Leo Breiman. 2001. Random Forests. *Machine Learning*, 45:5–32.

John Bissell Carroll. 1964. *Language and Thought*. Prentice-Hall.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A Library for Support Vector Machines. In *ACM Transactions on Intelligent Systems and Technology*, volume 2, New York, NY, USA. Association for Computing Machinery.

John W Chotlos. 1944. IV. A Statistical and Comparative Analysis of Individual Written Language Samples. *Psychological Monographs*, 56(2):75.

Noam Cohen. 2008. Open-Source Troubles in Wiki World. The New York Times. Last accessed on 2024-03-01.

Helmut Daller, Roeland van Hout, and Jeanine Treffers-Daller. 2003. Lexical Richness in the Spontaneous Speech of Bilinguals. *Applied Linguistics*, 24(2):197–222.

Alok Das. 2020. Neural Machine Translation (NMT): Inherent Inadequacy, Misrepresentation, and Cultural Bias. *International Journal of Translation*, 32:115–145.

M Ester, H P Kriegel, J Sander, and Xu Xiaowei. 1996. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases With Noise. *U.S. Department of Energy Office of Scientific and Technical Information*.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A Library for Large Linear Classification. *the Journal of Machine Learning Research*, 9:1871–1874.

Pierre Guiraud. 1954. *Les Caractères Statistiques du Vocabulaire: Essai de Méthodologie*. Presses universitaires de France, Paris, France.

Pierre Guiraud. 1959. *Problèmes et Méthodes de la Statistique Linguistique*. D. Reidel, Dordrecht, Holland.

Ari Hautasaari. 2013. "Could Someone Please Translate This?": Activity Analysis of Wikipedia Article Translation by Non-experts. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*, CSCW '13, page 945–954, New York, NY, USA. Association for Computing Machinery.

Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2021. The Interplay of Variant, Size, and Task Type in Arabic Pre-trained Language Models. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 92–104, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Isaac Johnson and Emily Lescak. 2022. Considerations for Multilingual Wikipedia Research. *arXiv preprint arXiv:2204.02483*.

Maria Lopez-Medel. 2021. Gender bias in machine translation: an analysis of Google Translate in English and Spanish. *Academia.edu*.

Philip Mccarthy and Scott Jarvis. 2010. MTLD, vocd-D, and HD-D: A validation study of sophisticated approaches to lexical diversity assessment. *Behavior research methods*, 42:381–92.

Philip M McCarthy. 2005. *An Assessment of the Range and Usefulness of Lexical Diversity Measures and the Potential of the Measure of Textual, Lexical Diversity (MTLD)*. Ph.D. thesis, The University of Memphis.

John Mittermeier, Ricardo Correia, Rich Grenyer, Tuuli Toivonen, and Uri Roll. 2021. Using Wikipedia to Measure Public Interest in Biodiversity and Conservation. *Conservation Biology*, 35.

El Moatez Billah Nagoudi, Muhammad Abdul-Mageed, AbdelRahim Elmadany, Alcides Alcoba Inciarte, and Md Tawkat Islam Khondaker. 2023. JASMINE: Arabic GPT Models for Few-Shot Learning. *arXiv preprint arXiv:2212.10755*.

Ranjita Naik, Spencer Rarrick, and Vishal Chowdhary. 2023. Reducing Gender Bias in Machine Translation through Counterfactual Data Generation. *arXiv preprint arXiv:2311.16362*.

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 (LREC'16)*, pages 4197–4201, 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.

Marcelo Prates, Pedro Avelar, and Luís Lamb. 2020. Assessing Gender Bias in Machine Translation: A Case Study With Google Translate. *Neural Computing and Applications*, 32.

Motaz Saad and Basem Alijla. 2017. WikiDocsAligner: An Off-the-Shelf Wikipedia Documents Alignment Tool. In *Palestinian International Conference on Information and Communication Technology (PICICT)*.

Neha Sengupta, Sunil Kumar Sahu, Bokang Jia, Satheesh Katipomu, Haonan Li, Fajri Koto, William Marshall, Gurpreet Gosal, Cynthia Liu, Zhiming Chen, Osama Mohammed Afzal, Samta Kamboj, Onkar Pandit, Rahul Pal, Lalit Pradhan, Zain Muhammad Mujahid, Massa Baali, Xudong Han, Sondos Mahmoud Bsharat, Alham Fikri Aji, Zhiqiang Shen, Zhengzhong Liu, Natalia Vassilieva, Joel Hestness, Andy Hock, Andrew Feldman, Jonathan Lee, Andrew Jackson, Hector Xuguang Ren, Preslav Nakov, Timothy Baldwin, and Eric Xing. 2023. Jais and Jais-chat: Arabic-Centric Foundation and Instruction-Tuned Open Generative Large Language Models. *arXiv preprint arXiv:2308.16149*.

Lucas Shen. 2022. LexicalRichness: A Small Module to Compute Textual Lexical Richness. Last accessed on 2024-03-01.

Gabriel Stanovsky, Noah A. Smith, and Luke Zettlemoyer. 2019. Evaluating Gender Bias in Machine Translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1679–1684, Florence, Italy. Association for Computational Linguistics.

Mildred C. Templin. 1957. *Certain Language Skills in Children: Their Development and Interrelationships*, NED–New edition, volume 26. University of Minnesota Press.

Brian Thompson, Mehak Preet Dhaliwal, Peter Frisch, Tobias Domhan, and Marcello Federico. 2024. A Shocking Amount of the Web is Machine Translated: Insights from Multi-Way Parallelism. *arXiv preprint arXiv:2401.05749*.

Stefanie Ullmann and Danielle Saunders. 2021. Google Translate is sexist. What it needs is a little gender-sensitivity training. Last accessed on 2024-03-01.

Wikidata. 2024a. Wikidata: Requests For Permissions/Bot/DarijaBot. Last accessed on 2024-03-01.

Wikidata. 2024b. Wikidata: Requests For Permissions/Bot/JarBot. Last accessed on 2024-03-01.

Wikimedia. 2024. Wikimedia Downloads. Last accessed on 2024-03-01.

Wikimedia Foundation. 2022. Content Translation – Mediawiki. Last accessed on 2024-03-01.

Wikimedia Statistics. 2024. New Pages: Egyptian Arabic Wikipedia. Last accessed on 2024-03-01.

Wikipedia. 2020. Steward Removal of Flags on ARZWiki. Last accessed on 2024-03-01.

Wikipedia. 2024a. Wiki Markup. Last accessed on 2024-03-01.

Wikipedia. 2024b. Wikipedia: Bot Policy. Last accessed on 2024-03-01.

Junjie Wu. 2012. *Advances in K-means Clustering: A Data Mining Thinking*. Springer Science & Business Media.

XGBoost. 2024. XGBoost Documentation. Last accessed on 2024-03-01.

Marie Lisandra Zepeda-Mendoza and Osbaldo Resendis-Antonio. 2013. *Hierarchical Agglomerative Clustering*, pages 886–887. Springer New York, New York, NY.

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. Analysis of N-Grams

We analyze the 5-grams and 10-grams closely since they are suitable, not long or short. The n-grams in the Egyptian Wikipedia are very large compared to the Arabic and Moroccan Wikipedia editions, as indicated in Tables 9 and 10. Plus, it is noticeable that these counts do not decay exponentially as they normally should (the larger the n-gram size, the smaller the n-grams' count) but linearly and slowly (all near 222K even with different sizes of n-grams), suggesting this abnormal decay is a symptom of the template translations that Egyptian Wikipedia suffered from, where some grams/parts/phrases from the used templates are frequently and constantly repeated.

We additionally observe that most of the top ten 5-grams and 10-grams of the Moroccan Wikipedia edition are predominantly non-Arabic grams, which seems in a format of the Wikitext Markup Language (Wikipedia, 2024a), as exhibited in Tables 9, 10, and 11. We further investigate this issue by testing our parsing code scripts and find that it does not occur when parsing articles from the other two Arabic Wikipedia editions, Arabic (AR) and Egyptian (ARZ), using the same code scripts; it only surfaces when parsing the Moroccan Wikipedia articles. We attribute this issue to either leaking Wikipedia templates used to create articles or insert images into articles or an issue with the method used to dump and compress Moroccan Wikipedia articles. We urge the global and local admins of the Moroccan Wikipedia edition to investigate this issue, which could affect not only the Moroccan Wikipedia content but also the performance of perspective NLP models and tasks trained on such content.

B. Heuristic Filtration Rules

We list the heuristic filtration rules used to filter the articles *before* and *after* the template-based translation in the Egyptian Wikipedia edition and further shed light on the effectiveness of each enforced rule. We demonstrate, in Figures 5 and 6, the effectiveness of the implemented rule-based filtration. We can see that our heuristic filtration rules are practical, as each rule consecutively and rigorously filters out unfit articles that do not meet the heuristic filtration rules.

∗ Heuristic filtration rules for *before* the translation:

1. Include articles created before 2019-12-01.
2. Include articles with more than five edits.
3. Include articles with more than three editors.
4. Include articles with greater than or equal to 50% human editors.

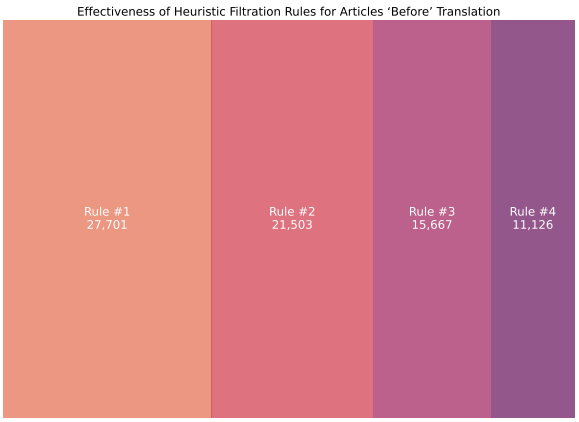


Figure 5: A treemap showing the effectiveness of the heuristic rules for articles *before* the template-based translation in Egyptian Wikipedia, highlighting the number of articles filtered out by each rule.

∗ Heuristic filtration rules for *after* the translation:

1. Include articles created between 2019-12-1 and 2023-12-01 and discard young articles with an age of less than 30 days (2023-12-1 and 2024-1-1).
2. Include articles with less than five edits.
3. Include articles with less than three editors.
4. Include articles with greater than or equal to 50% bot editors.
5. Include articles created by these registered users, 'HitomiAkane' and 'Al-Dandoon', who overwhelmed the Egyptian Arabic Wikipedia with massive auto-generated and template-translated articles without human supervision.

Wikipedia	Count	5-Gram
Arabic (AR)	141,880	تصنيف أشخاص على قيد الحياة Classification of surviving people
	80,460	لاعبو كرة قدم رجالية مغتربون Men's football players expatriate
	38,793	تعداد عام وبلغ عدد الأسر A general census, and the number of families reached
Egyptian (ARZ)	222,964	صوره هيا مجال الكره السماويه It is a picture of the celestial sphere
	222,961	الكره السماويه اللى المجره جزء The celestial sphere, of which the galaxy is a part
	222,939	مجموعه من النجوم اللى بتكون A collection of stars that forms
Moroccan (ARY)	5,172	width text textcolor black fontsize
	2,057	لعاطلين اللي سبق ليهوم خدمو For unemployed people, who have previously served
	1,483	على حساب لإحصاء الرسمي عام According to the official census of the year

Table 9: Selected top three 5-grams from each Arabic Wikipedia edition with their English translations.

Wikipedia	Count	10-Gram
Arabic (AR)	38,790	تعداد عام وبلغ عدد الأسر أسرة وعدد العائلات عائلة مقيمة A general census, the number of families was one family, and the number of families was one resident family
	38,710	وبلغت نسبة الأزواج القاطنين مع بعضهم البعض من أصل المجموع The percentage of couples living together was out of the total
	38,524	نسبة منها لديها أطفال تحت سن الثامنة عشر تعيش معهم A percentage of them have children under the age of eighteen living with them
Egyptian (ARZ)	222,935	صوره هيا مجال الكره السماويه اللى المجره جزء منها الانزياح A picture of the celestial sphere, of which the galaxy is a part of the displacement
	222,935	المطلع المستقيم هوا الزاويه المحصوره بين الدايره الساعيه لجرم سماوى The right ascension is the angle enclosed between the hourly circle of a celestial body
	221,251	أو صوره هيا مجال الكره السماويه اللى المجره جزء منها Or a picture of the celestial sphere, of which the galaxy is a part
Moroccan (ARY)	2,586	imagesize width height plotarea left right top bottom timeaxis orientation
	1,483	لعاداد كان ديالو واصل شخص على حساب لإحصاء الرسمي عام The number of people was counted up to according to the official census of the year
	1,348	ما كايعرفوش يقراو ولا يكتبو نسبة كان قاريين فوق انوي They did not know how to read or write the percentage of literate was above

Table 10: Selected top three 10-grams from each Arabic Wikipedia edition with their English translations.

توريرت (سيدي أحمد وعبدالله): أرمد هو دوار مجمع كاين جماعة أسني دائرة أسني إقليم لحوز جهة مراكش آسفي لمغريب هاد وار كينتامي مشيخة إمليل لعاداد كان ديالو واصل شخص على حساب لإحصاء الرسمي عام هو دوار لي كاين الجبل السلسلة ديال لأطلس الكبير الغربي الجغرافيا دوار أرمد بعيد كلم على مدينة مراكش على ارتفاع حوالي ميترو على البحر فالجبال ديال الأطلس الكبير هاد الدوار مشهور بلفلاحة خصوصا التفاح والكَرْكَاع فيه بزاف لوبيرجات والمحلات ولعشاش دالتجارة السياحية والبيع دالمنتجات التقليدية والماكلة السكان إحصائيات عامة عدد السكان ديال أرمد تزاد عدد لفاميلات تزاد مابين عدد لبالغين كان واحد منهوم دكور imagesize width height plotarea left right top bottom timeaxis orientation vertical alignbars justify colors id gray value gray dateformat yyyy نتوات period from till scalemajor unit year increment start gridcolor gray plotdata bar color green width from till width text textcolor black fontsize px bar color red width from till width text textcolor black fontsize px imagesize width height plotarea left right top bottom timeaxis orientation vertical alignbars justify colors id gray value gray dateformat yyyy period from till scalemajor unit year increment start gridcolor gray plotdata bar color width from till width text textcolor black fontsize px bar color red width from till width text textcolor black fontsize px green لفاميلات الجواج لولادة نسبة الجواج أرمد واصلة لموعد ال لعمر عند الجواج اللولاني هو عام عند الرجال عند لعيالات لخصوبة عند لعيالات واصلة لخصوبة لكاملة التسكويل نسبة التسكويل واصلة نسبة لأمية واصلة لخدمة نسبة الناس النشيطين دوار أرمد واصلة نسبة الشوماج واصلة نوطات عيون لكلام تصنيف جهة مراكش آسفي تصنيف دوار لمغريب تصنيف دوار إقليم لحوز تصنيف مقالات فيها مصدر بايت.

Table 11: A sample of a parsed article from Moroccan Wikipedia, showing the embedded Wiki markups.

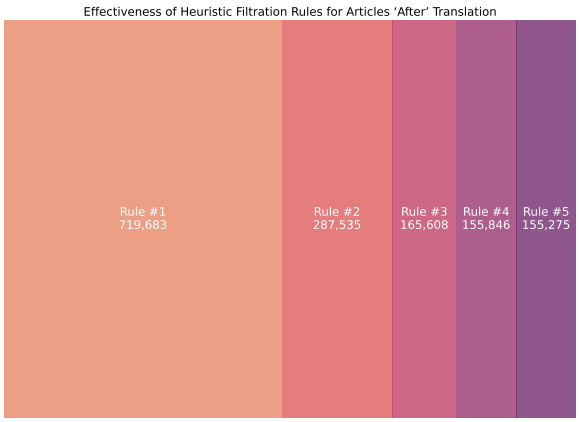


Figure 6: A treemap showing the effectiveness of the heuristic rules for articles *after* the template-based translation in Egyptian Wikipedia, highlighting the number of articles filtered out by each rule.

C. Egyptian Wikipedia Scanner

We evaluate our multivariate supervised machine learning classifiers using metrics like accuracy and ROC-AUC (Receiver Operating Characteristic Area Under Curve). We then publicly deploy and host our best classifier, XGBoost, which takes input features of articles' metadata and CAMeLBERT embeddings, as illustrated in Figures 7 and 8. We include the articles' metadata because we find that, from our two ablation studies, metadata could be practical and encode features useful for the classifier's learning. We also choose CAMeLBERT over Word2Vec word embeddings because CAMeLBERT's embeddings take the context into account, and Word2Vec's embeddings are context-free and need to be retrieved word by word and then averaged for the whole article; this is not ideal.

We call this online application Egyptian Wikipedia Scanner, where users can search for an article directly or select a suggested article retrieved using fuzzy search from the Egyptian Arabic Wikipedia edition.The application automatically fetches the article's metadata (using the Wikimedia `XTools` API), displays the fetched metadata in a table, and automatically classifies the article as 'human-generated' or 'template-translated'. The application also dynamically displays the full summary of the article and provides the URL to the article to read the full text, as shown in Figure 9.

We utilize the Streamlit Framework[15] to design, host, and deploy the application on the free Streamlit Community Cloud[16] service, making it publicly accessible to everyone at https://egyptian-wikipedia-scanner.streamlit.app. We also host the application on Hugging Face Spaces to avoid running out of Streamlit Cloud free, limited resources: https://huggingface.co/spaces/SaiedAlshahrani/Egyptian-Wikipedia-Scanner. This online application, Egyptian Wikipedia Scanner, is open-sourced on GitHub with an MIT license, here: https://github.com/SaiedAlshahrani/Egyptian-Wikipedia-Scanner.

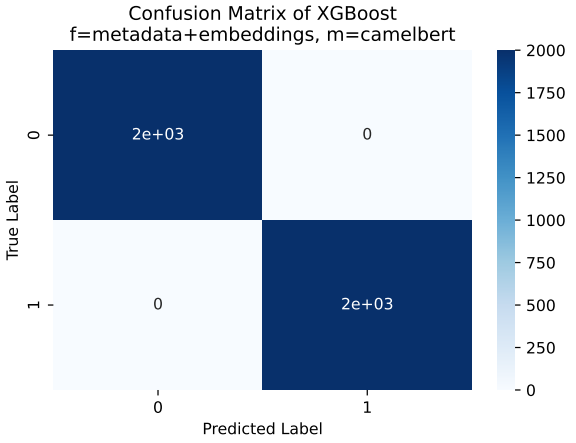


Figure 7: Confusion matrix of the best, deployed classifier, XGBoost, which takes input features of articles' metadata combined with CAMeLBERT's embeddings, showing the excellent performance of this multivariate ensemble classifier.

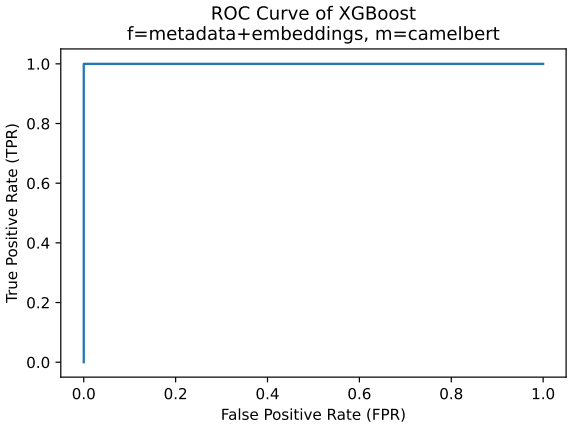


Figure 8: ROC curve of the best, deployed classifier, XGBoost, which takes input features of articles' metadata combined with CAMeLBERT's embeddings, showing the excellent performance of this multivariate ensemble classifier.

[15]Streamlit Framework: https://streamlit.io.
[16]Streamlit Cloud: https://streamlit.io/cloud.

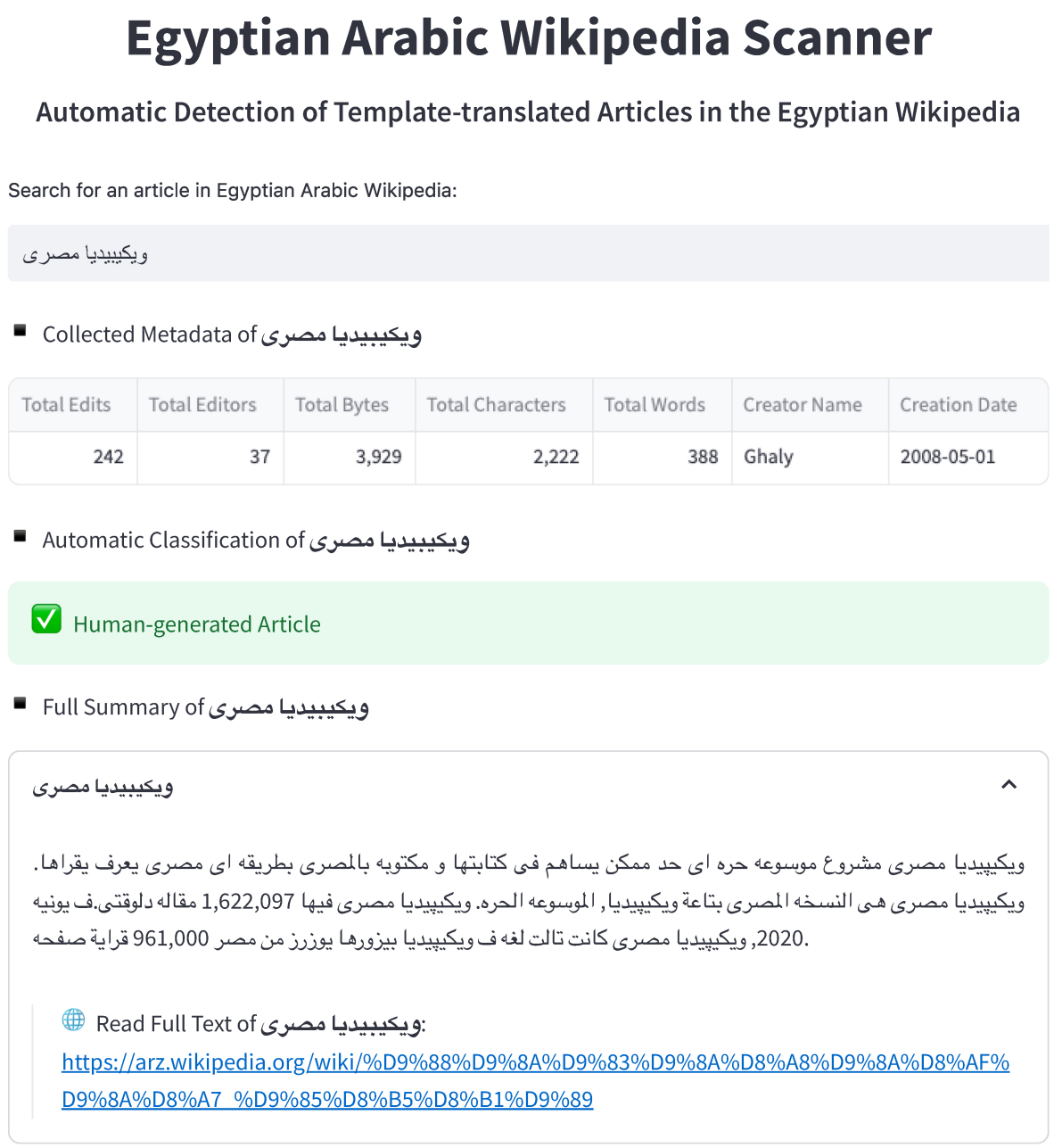


Figure 9: A screenshot of the Egyptian Wikipedia Scanner, illustrating its capabilities and features.

A Novel Approach for Root Selection in the Dependency Parsing

Sharefah AL-Ghamdi, Hend Al-Khalifa, Abdulmalik Al-Salman
College of Computer and Information Sciences,
King Saud University, P.O. Box 145111, Riyadh 4545, Saudi Arabia
{sharefah, hendk, salman}@ksu.edu.sa

Abstract

Although syntactic analysis using the sequence labeling method is promising, it can be problematic when the labels sequence does not contain a root label. This can result in errors in the final parse tree when the postprocessing method assumes the first word as the root. In this paper, we present a novel postprocessing method for BERT-based dependency parsing as sequence labeling. Our method leverages the root's part of speech tag to select a more suitable root for the dependency tree, instead of using the default first token. We conducted experiments on nine dependency treebanks from different languages and domains, and demonstrated that our technique consistently improves the labeled attachment score (LAS) on most of them.

Keywords: Dependency Parsing, Sequence Labeling, Natural Language Processing, BERT, Transformers

1. Introduction

Dependency parsing is the task of identifying the syntactic structure of a sentence by assigning a head (parent) and a label to each word (child). Traditionally, this task has been approached using transition or graph-based methods, which rely on explicit parsing algorithms or auxiliary structures. However, it has been shown that dependency parsing can also be performed as a sequence-labeling problem (Lacroix, 2019; Strzyz et al., 2019), where each word is associated with a label that encodes its head and syntactic information. Strzyz et al. (2019) show that this approach offers a good trade-off between parsing accuracy and speed as it leverages the efficiency of deep learning frameworks running on GPUs.

One of the challenges of dependency parsing as sequence labeling is the postprocessing stage, which can introduce errors in the syntactic analysis. A common flaw in this stage is that if the parser does not assign any word a label to be the root of the parse tree, it selects the first word in the sentence as the head of the syntactic tree and updates the rest of the labels accordingly (Vilares et al. 2020). Al-Ghamdi et al. (2023) highlighted this issue and showed that it was the reason for some errors in the final syntactic results.

In this work, we aim to reduce the effect of postprocessing on dependency parsing as sequence labeling. We propose a root part-of-speech (POS) identification as postprocessing method that predicts the root POS tag for a given sentence. Instead of choosing the first token of the sentence as the root when the parser fails to label a root, we choose the first token that has the predicted POS tag as the root. This way avoids some errors in the syntactic analysis.

To apply the proposed method, a root POS identifier (RPI) was built by fine-tuning a BERT pretrained model for text classification to perform the proposed solution of identifying the root POS for a sentence. This work is an enhancement of a previous exploratory work that also attempted to build a root index identifier model, yet the model was not accurate enough to find the correct roots. The details of that exploration are beyond the scope of this paper and will be reported elsewhere.

Our method was evaluated on nine different treebanks, and we showed that it could improve the Label attachment scores (LAS) and unlabeled attachment scores (UAS) of BERT-based dependency parsing for most parsers.

The rest of this paper is organized as follows: Section 2 reviews the postprocessing for the parse trees and presents the RPI used in the proposed method. Section 3 describes the experiments setup, including the data sets and the baseline parser. Section 4 reports and analyzes the results of our method and compares it with the baseline. Finally, Section 5 concludes the paper with limitations and future work.

2. Postprocessing with root POS identification

The root's POS tag is the tag of the word that serves as the syntactic head of the parse tree for the sentence. It also can provide useful information for downstream tasks, such as parsing and semantic analysis. In this section, we propose a novel root POS identification method for postprocessing in the sequence labeling dependency parsing. The following two sub-sections explain the postprocessing steps of the parse tree and present the proposed RPI.

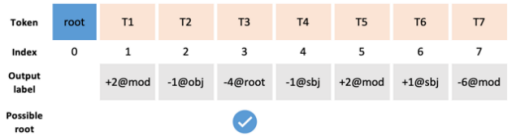
2.1 The Postprocessing for Parse Trees

The postprocessing implementation of Vilares et al. (2020) shows how to construct a parse tree from a labeled sequence, which is the output of the model. If the output labels do not include a root or a possible root, the first token in the sequence is assigned as the root node. The root is a token that has a head index of 0 and a relation label of the root. In contrast, the possible root is a token that has a head index that is not the root index but has a relation label of the root. Figure 1 shows an example of an output label sequence that does not include a root token but includes a possible root token. Token number 3 has a head index of -4, which means the fourth token to the left, but there are only three tokens to the left.

OSACT 2024 Workshop, pages 46–49

However, it has a relation label @root. Therefore, it is selected as the root node of the parse tree.

Figure 1: An example of the output of labels that



includes possible root.

Instead of selecting the first token as the root node, we apply our novel postprocessing step that identifies the root POS tag for a given sequence. Then, we select the first token that has the predicted POS tag as the root node. For instance, in the example shown in Figure 2, the third token in the sequence was selected as the root node, because it was the first verb token, and the RPI predicted the root POS tag as a verb.

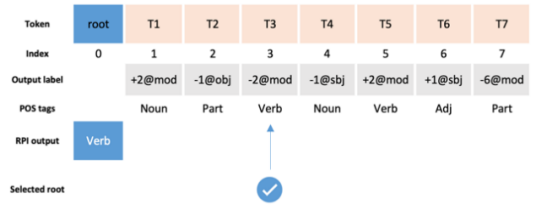


Figure 2: An example of assigning root using RPI.

To perform our proposed postprocessing step, we designed a simple but effective root POS identifier (RPI) based on BERT. It can recognize the POS tag of the syntactic root of the input sentence. We describe the details of BERT-based RPI in the next section.

2.2 The proposed RPI

We formulate the task of root POS identification as a text classification problem, which is a natural language processing task that aims to assign a label to a given sequence of words. For example, given a sentence, one can classify it as a positive or negative sentiment, or as a question or a statement.

We implemented an RPI as a simple text classification task using the pretrained BERT model (Devlin et al., 2018). We fine-tuned the pretrained language model to predict the root part-of-speech (POS) tag for an input sequence. For example, given an input sentence S with a sequence of tokens [T_1, …, T_n], the model predicts the POS tags that corresponds to the syntactic root of S.

3. Experiments

To test our proposed method, we used nine datasets as our input. Seven of them came from the UD2.12 universal dependency treebanks (Zeman et al., 2023). The other two were Arabic datasets: one was the converted version of the Penn Arabic Treebank (ATB) part2 v3.1 (Diab et al., 2013), which was part of the Columbia Arabic Treebank (CATiB), and the other was the Classical Arabic poetry dependency treebank (ArPoT) (Al-Ghamdi et al., 2021). Table 1 shows the number of different root part-of-speech (POS) tags (classes).

We measured the accuracy of our BERT-based RPI in predicting the correct POS tags of the root nodes. Then, we evaluated the effectiveness of our method with the BERT-based sequence labeling dependency parser (P-w-RPI) by comparing the UAS/LAS scores with the baseline parsers (BL) without RPI.

The baseline parser we built is based on (Vilares et al. 2020) and (Al-Ghamdi et al., 2023). The list of the fine-tuned BERT models used for each language is presented in Table 1. We also show the number of different POS tags for roots in each dataset.

We used Colab T4 GPUs to train all models on 10 epochs. All experiment's implementation codes and settings will be released on Github: https://github.com/Sharefah-Alghamdi

Table 1: Number of Root POS tags and BERT models for nine treebanks under study.

Dataset	Roots' POS	BERT Model
AR $_{ArPoT}$	5	bert-base-arabertv2
AR $_{CATiB}$	6	
AR $_{PADT}$	14	
EL $_{GDT}$	15	bert-base-greek-uncased-v1
EN $_{EWT}$	8	bert-base-uncased
FR $_{ParTUT}$	8	bert-base-french-europeana-cased
TA $_{TTB}$	6	tamil-bert
TR $_{IMST}$	13	bert-base-turkish-uncased
ZH $_{GSD}$	10	bert-base-chinese

4. Results and Analysis

The results of our RPI on the nine datasets are shown in Table 2. The highest accuracy was obtained for Tamil (96.67%), followed by Chinese (91.6%), Arabic (PADT) (91.32%), and Arabic (CATiB) (90.1%). The lowest accuracy was obtained for Turkish (76.92%), followed by Arabic (ArPoT) (79.34%), and Greek (84.21%). These results indicate that our model can effectively predict the root POS tag for most datasets, but there is still room for improvement for some languages.

Table 2: RPI accuracy for nine datasets.

Dataset	RPI
AR ArPoT	79.34
AR CATiB	91.23
AR PADT	91.32
EL GDT	84.21
EN EWT	88.88
FR ParTUT	88.18
TA TTB	96.67
TR IMST	76.92
ZH GSD	91.6
Average	**87.6**

We hypothesize that there is a relation between the difficulty of syntactic root POS identification and the difficulty of grammatical nature understanding for the language model. For example, some languages may have more complex or irregular word forms, or more syntactic variations than others. These factors may make it harder to predict the root POS tag for some languages than others. However, the average accuracy (87.6%) across all datasets shows that BERT-based text classification models can sufficiently perform our RPI.

We evaluated the impact of using RPI in the postprocessing stage on the parsing accuracy of the BERT-based dependency parser. The UAS/LAS scores of BL and BL using RPI on the nine treebanks are shown in Table 3. We calculated the average results over three runs to ensure the reliability and consistency of our models.

Table 3: UAS/LAS of the baseline (BL) parser with and without RPI.

Dataset	BL		P-w-RPI	
AR ArPoT	80.01	74.21	▲ 80.02	▲ 74.22
AR CATiB	87.54	86.47	■ 87.54	■ 86.47
AR PADT	84.49	80.55	■ 84.49	■ 80.55
EL GDT	62.24	54.88	▲ 62.37	▲ 55.31
EN EWT	83.94	80.62	▲ 84.05	▲ 80.71
FR ParTUT	89.05	87.67	▲ 89.11	▲ 87.73
TA TTB	58.50	47.16	▲ 58.64	▲ 47.24
TR IMST	63.75	51.19	■ 63.75	▼ 51.13
ZH GSD	77.10	74.02	▲ 77.21	▲ 74.12

We found that adding RPI in the postprocessing step improved UAS scores for six out of the nine treebanks, whereas the Arabic (CATiB and PADT) and Turkish had no change. The LAS scores also increased for six of treebanks. Arabic (CATiB and PADT) also had no change, and Turkish had a slight decrease (-0.06%). The highest UAS improvement was achieved by Tamil (+0.14%), followed by Greek (+0.13%). The lowest improvement was achieved by Arabic (ArPoT), which had negligible changes (0.01% and 0.02%) in UAS and LAS respectively. These results indicate that the postprocessing step using RPI can enhance the quality of parsing results by selecting more appropriate roots.

We analyzed the results of our experiments in different scenarios and found their strengths and limitations. Table 4 presents the analysis metrics of the proposed method on various datasets. The metrics are:

- **No root:** The percentage of trees without roots generated by the baseline parser.
- **Possible roots (PR):** The percentage of trees that are treated by using possible roots in the output labels.
- **Processed with RPI (w-RPI)**: The percentage of trees that processed with the use of our RPI. It is equal to the No root minus the PR columns.
- **Correct POS (c-POS)**: The percentage of correct root POS tags predicted by our RPI model.
- **First Token (FT)**: The percentage of cases where the first token with the predicted POS tag in the sentence is the correct root, as in the gold dataset. For example, the first verb is the correct root, not the second or third verb. (we counted only the roots that their POS predicted correctly).

Table 4: Average percentages for nine treebanks on metrics related to parse tree roots: No root, possible root (PR), processed by RPI (w-RPI), correct POS (c-POS), First Token roots with predicted POS (FT).

Dataset	No root	PR	w-RPI	c-POS	FT
AR ArPoT	8.36	7.63	0.74	83.33	66.67
AR CATiB	0.29	0	0.29	0	0
AR PADT	0.88	0	0.73	20.95	33.33
EL GDT	7.16	0	7.01	83.06	72.22
EN EWT	1.70	0	1.64	82.36	82.57
FR ParTUT	0.91	0	0.91	0	0
TA TTB	0.56	0	0.56	100	100
TR IMST	1.78	0	1.78	53.57	38.26
ZH GSD	3.20	0	3.13	84.32	33.98

Table 4 shows that the number of trees without roots varies across languages and treebanks, from 0.29% for Arabic (CATiB) to 8.36% for Arabic (ArPoT). Except for Arabic (ArPoT), none of the trees without roots have any possible root tokens (PR). That means our proposed postprocessing step is needed for most of the trees without root labels. The table also shows that our RPI was applied on a considerable proportion of the trees, especially for Greek (7.01%) and Chinese (3.13%).

The accuracy of RPI on predicting the root POS tag (c-POS) is also reported in Table 2. This metric explains why some datasets did not show any improvement after applying the postprocessing step. The datasets with a small number of trees that were generated without a root had lower accuracy of c-POS prediction. For instance, Arabic (CATiB and PADT) had one and six root-less trees in each respectively, and they reported low accuracy of root POS prediction by RPI. However, the postprocessing step of choosing the root instead of the first token still improved the results for these datasets, even when the POS tag was incorrectly identified for some languages such as French (improved by +0.06 in both UAS and LAS in Table 3). On the contrary, Turkish had a low accuracy of root POS prediction, which was consistent with the low performance of RPI for Turkish in Table 2.

Our method achieved better results on Greek treebank than on other treebanks, because this treebank had several factors that suited our method. First, the RPI treated a relatively large proportion of trees without roots. Second, the RPI was very accurate in predicting the POS tag of the root word in the sentence. Third, for most sentences, if there were more than one token with the predicted POS tag (three verbs, for example), usually the first one in the sequence was the correct root.

The results illustrated how our postprocessing step improves the quality and completeness of the parse trees by finding a valid root node. It also highlighted the languages' differences and challenges in predicting the correct root based on the POS tag.

5. Conclusion

The paper shows that our root POS identification as postprocessing can improve the results of the dependency parser as a sequence labeler by selecting a more proper syntactic root. The results show that our method can enhance the completeness of the dependency structure in the parse tree. We evaluated our method on nine treebanks, and demonstrated that it can enhance UAS/LAS scores over most of them.

The work also revealed that the postprocessing of the syntactic structures of sentences had different effects on different treebanks, depending on the nature of the syntactic relations in those treebanks. Therefore, we identified a limitation of our method, as the high accuracy of the RPI might not be enough to determine the correct root. For instance, if most of the sentences in a treebank have a root that is the second or third word in the sequence that has the predicted POS, our method might not be beneficial. Moreover, there might be some treebanks that we have not examined, and that might not produce sequences without roots, and in this case, there is no need for any postprocessing at all.

In the future, we might explore more treebanks and look for solutions that make our method universally applicable to all languages and treebanks.

6. References

David Vilares, Michalina Strzyz, Anders Søgaard, and Carlos Gómez-Rodríguez. 2020. *Parsing as pre-training*. In *Proceedings of the AAAI Conference on Artificial Intelligence*, (vol. 34, No. 05, pp. 9114-9121). https://doi.org/10.1609/aaai.v34i05.6446

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. *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) (pp. 4171-4186). http://dx.doi.org/10.18653/v1/N19-1423

Michalina Strzyz, David Vilares, and Carlos Gómez-Rodríguez. "*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), pp. 717-723. 2019. http://dx.doi.org/10.18653/v1/N19-1077

Ophélie Lacroix. 2019. *Dependency parsing as sequence labeling with head-based encoding and multi-task learning.* In *Proceedings of the Fifth International Conference on Dependency Linguistics* (Depling, SyntaxFest 2019) (pp. 136-143). http://dx.doi.org/10.18653/v1/W19-7716

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), Article # 4225. https://doi.org/10.3390/app13074225

7. Language Resource References

Daniel Zeman, Joakim Nivre; et al. 2023. *Universal dependencies 2.12,* LINDAT/CLARIAH-CZ *digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.*

Mona Diab, Nizar Habash, Owen Rambow and Ryan Roth. 2013. *LDC Arabic treebanks and associated corpora: Data divisions manual*.

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), Sofia, Bulgaria, 21–25 March 2021; pp. 1–9

AraMed: Arabic Medical Question Answering using Pretrained Transformer Language Models

Ashwag Alasmari[1], Sara Alhumoud[2], Waad Alshammari[3]
[1]Department of Computer Science, King Khalid University
[2]Imam Mohammad Ibn Saud Islamic University
[3]King Salman Global Academy for Arabic Language
Saudi Arabia
aasmry@kku.edu.sa, sohumoud@imamu.edu.sa, walshammari@ksaa.gov.sa

Abstract

Medical Question Answering systems have gained significant attention in recent years due to their potential to enhance medical decision-making and improve patient care. However, most of the research in this field has focused on English-language datasets, limiting the generalizability of MQA systems to non-English speaking regions. This study introduces AraMed, a large-scale Arabic Medical Question Answering dataset addressing the limited resources available for Arabic medical question answering. AraMed comprises of 270k question-answer pairs based on health consumer questions submitted to online medical forum. Experiments using various deep learning models showcase the dataset's effectiveness, particularly with AraBERT models achieving highest results, specifically AraBERTv2 obtained an F1 score of 96.73% in the answer selection task. The comparative analysis of different deep learning models provides insights into their strengths and limitations. These findings highlight the potential of AraMed to advance the creation and development of resources specific to Arabic medical question answering research and development.

Keywords: natural Language processing, medical question answering, answer selection, language models

1. Introduction

With the ever-increasing volume of health information available online, finding accurate answers to specific medical requests is becoming more difficult for health consumers (Alasmari & Zhou, 2019, 2021). Medical question and answer (MQA) platforms provide an online space where users can ask direct questions and medical experts can provide answers to these questions. This is the most intuitive ways for people to seek information online, especially when search engines fail to provide relevant and accurate results (Liu et al., 2012). In addition, MQA enables consumers to avoid long wait times when seeking health information, particularly if they require information quickly or need information about managing a health condition at home. Wicks et al. (Wicks et al., 2010) demonstrated that an online medical platform can assist users in managing their own symptoms, dealing with side effects, connecting with other patients, and seeking medication advice.

The explosive demand and growth of users and questions are creating a bottleneck for the limited number of doctors. Therefore, developing ways of automatically answer questions using the information from previously answered questions is important. Particularly for delivering responses that direct consumers to the potentially relevant information about their concern. Ideally, implementing effective solutions can result in the workload of doctors being greatly reduced and the consumer's experience of online medical systems being enhanced. That is, lower overhead cost on the medical institutions, the speed into which the answer is fetched, and harnessing the intelligence of already available MQA forums and datasets. MQA systems are designed to solve the automatic QA issue by the development of several tracks and tasks including question ranking, question similarity, and answers selection. Question ranking is the task of ranking a set of questions according to their relevance to a given question whereas question similarity is concerned with detecting the semantic similarity between two questions (Mishra & Jain, 2016). We focus on the task of answer selection in this work, which is the task of choosing the best answer to a given question from a set of possible answers.

Question Title	حمى ابنتي 38.4 من دون اعراض أخرى. "My daughter has a fever 38.4 and no other symptoms."
Question Description	تعاني ابنتي من حمى 38.4 منذ يومين ولا تعاني من اي اعراض اخرى ماذا افعل هل اخذها للطبيب ام انتظر. "My daughter has a fever 38.4 for two days and she does not have any other symptoms. What should I do? Should I take her to the doctor or wait?"
Question Category	امراض الأطفال "Pediatric"
Relevant Answer (RA)	أنصحك بعرض ابنتك علي طبيب الاطفال للفحص الطبي ومعرفه سبب الحرارة. مادامت الحرارة مستمرة منذ يومين. لإجراء الفحص والتشخيص وكتابه العلاج اللازم. تمنياتي لطفلك بالصحة والسلامة. "I advise you to visit a pediatrician for a medical examination and to find out the cause of the fever. as it continued for two days. This is to conduct the examination, diagnosis and write the necessary treatment. I wish your child health and peace."
RA Doctor specialty	طب أطفال "Pediatrics"
Irrelevant Answer (IA)	اشتباه عرق النساء وهذا عرض لمرض محتاج تعمل اشعه بالرنين المغناطيسي. "It might be sciatica, and this is a symptom of a disease that needs an MRI scan."
IA Doctor specialty	الروماتيزم والمفاصل "Rheumatology and joints"

Figure 1: An example of MQA sample in AraMed. The English translation is not a part of the corpus.

OSACT 2024 Workshop, pages 50–56

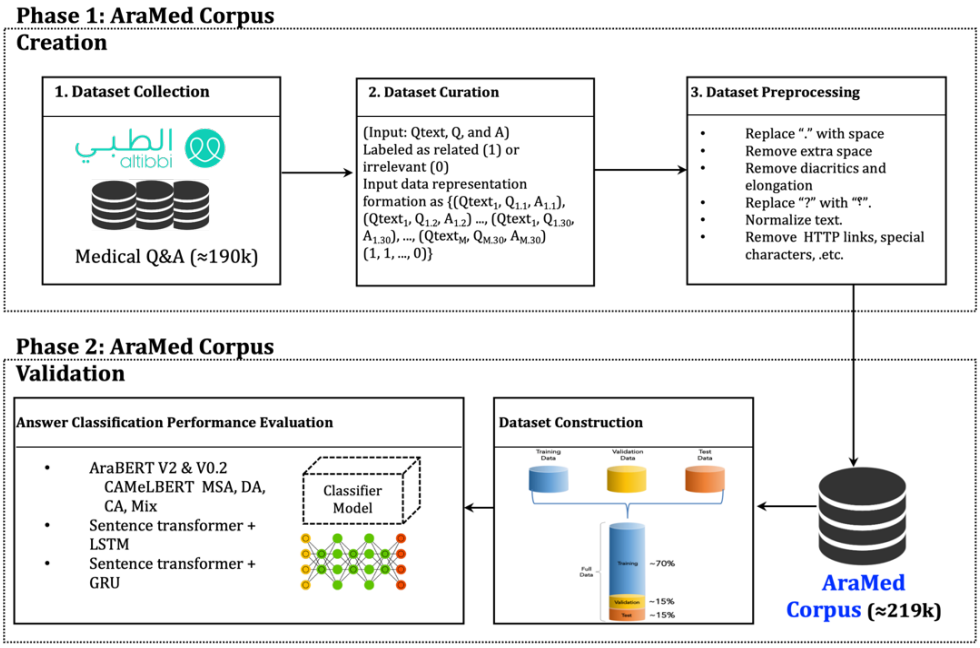


Figure 2: Flow diagram of AraMed dataset creation and validation.

Answer selection has received an increasing attention recently. There are several benefits of answer selection in MQA. Answer selection can improve the accuracy of the answers, reduce the time it takes to answer questions, and increase user satisfaction by providing users with the most relevant answers to their questions. The aim is to identify which of the candidate answers contain the correct answer to a question given a question and a set of candidate answers (Lai et al., 2018). Traditional approaches to answer selection typically rely on recurrent neural networks, including long short-term memory networks and Seq2seq models (Roy et al., 2023). However, the recent development of attention mechanisms, like those found in Transformers, BERT models and large language models has shifted the focus of answer selection research in the question answering research (Zhang et al., 2021; Guo et al., 2022).

The quality and accuracy of MQA systems are dependent mainly on two factors: the size and quality of the corpus and the efficacy of the machine learning model. In English multiple MQA corpora are available that can be used to build more efficient MQA systems. However, the Arabic language currently lacks such comprehensive resources.

To date, the contribution in the Arabic MQA corpora is limited to two, ARmed (Fehri et al., 2022) and CQA-MD (Adlouni et al., 2019; Balla et al., 2022). This could be due to the lack of resources, the morphological complexity of Arabic, and the multiple forms and dialects that are used for online expression and communication (Boudjellal et al., 2020). In particular, the first contribution (Fehri et al., 2022) created a corpus, ARmed, which included 350 manually annotated questions along with the corresponding responses which were collected from several resources for the purpose of automatically answering medical questions in Arabic. The second contribution is a widely used corpus, CQA-MD, which includes 57,764 question-answer pairs. This dataset was used for the SemEval 2016 and 2017 workshops (Agirre et al., 2016; Nakov et al., 2017), which were collected from three Arabic medical websites including WebTeb, Al-Tibbi, and Islamweb. Several studies built MQA systems utilizing this corpus (Adlouni et al., 2019; Balla et al., 2022). However, CQA-MD is small and unbalanced and contains only 1,943 directly relevant, 24,265 relevant, and 31,556 irrelevant answers. Direct relevant represents the correct answers and is only 4% of the dataset, and it is a low number not sufficient for deep learning purposes. In addition, the average sequence length of combining the new query with the question answer pairs is 1,101 which is considered large as an entry for transformer-based models. Another contribution related to the MQA (Faris et al., 2022) introduced an annotated corpus of 75,000 medical questions only, each of which is assigned one of 15 medical topics; the goal for this was to build an automated question classification. Clearly, the need for an Arabic MQA corpus of a decent size to enable creation of advanced MQA systems is evident.

To address this gap, this paper presents AraMed, a large-scale and a balanced MQA corpus containing 270k pairs of question and answer, along with the question categories, age and gender of the questioner, the specialty of the physician who answered the question, and the date. AraMed is constructed from Altibbi platform data, a popular medical website in the MENA region. This work has two major contributions:

- The collection and curation of an annotated dataset of 270k Arabic question-answer pairs based on health consumer questions

submitted to the Altibbi platform. The code and datasets will be made available upon request.

- Performing preliminary experimentation on the dataset using state-of-the-art pretrained models that are trained specifically for answer selection tasks to benchmark the dataset for future work.

	Train	Test
Questions	109,834	27,459
Answers	219,668	54,918
Categories	85	78
Relevant Answers	109,834	27,459
Irrelevant Answers	109,834	27,459

Table 1: Description of the AraMed corpus.

2. Dataset

In this section, we describe the four main aspects related to the creation of AraMed corpus: data collection, data curation, and data preprocessing. Those aspects are to be described below. Figure 2 illustrates the architecture of the AraMed corpus creation and validation process.

2.1 Data Collection

All data was obtained from the medical platform, Altibbi.com. This platform aims to provide users with reliable, up to date, simplified medical information in Arabic. The website includes thousands of medical articles, a medical glossary, a section of questions and answers (Q&A), the most recent medical news, and telehealth services and consultations. For this work, we collected the most recent Q&A, ranging from 2020 to 2021, and the most useful questions by the vote of users on the website. The resulting dataset includes 219,668 unique Q&A pairs.

2.2 Data Curation

Our corpus involved several preprocessing steps, beginning with the removal of questions with no answers and the removal of duplicate answers for the same question. Additionally, for the learning process to be effective, the data needed to include both questions with correct answers and the same questions with irrelevant answers. As the dataset does not contain irrelevant answers, a rule-based automated annotation approach was applied (Thuwaini & Alhumoud, 2022). This approach is summarized as follows:

For each question that have *n* correct answers, *n* candidate irrelevant answer is appended. To nominate the candidate irrelevant answer, a sliding window is moved over the answers by *m*, where *m=i+10* and *i* denominates the current row. The candidate m^{th} answer is checked against 2 conditions. The first is the category; if both the candidate m^{th} answer category and the current i^{th} answer category are the same, then the irrelevant answer is skipped, and *m* is incremented by 5. This is to confirm that the two answers, the current and the candidate, are not the same or similar. The second condition is that the m^{th} answer has more than 7 words. This is to ensure that the answer has a substantial content that would aid the learning process, as answers with a smaller number of words are not indicative or informative. If the candidate answer satisfies both conditions then it is used as the current question's irrelevant answer, its related data is appended to the question and *i* is incremented by 1 to proceed to the next question.

A unique ID is added for every question, relevant answer, and irrelevant answer. After curation and annotation process, the columns added to the dataset are question ID, relevant answer ID, irrelevant answer, and irrelevant answer ID, and answer date of irrelevant answer.

2.3 Data Preprocessing

The following preprocessing steps are applied to the data:

- Replace "." with space, since it has been used as delimited between tokens, such as, نعم...طبيعي.
- Remove extra spaces.
- Remove diacritics and elongation "ـ" using Pyarabic, an Arabic plugin tool for Python.
- Remove HTTP links, special characters, English alphabet, English numbers, Arabic numbers, and extra spaces using regular expressions, a built-in Python package.
- Normalize text, that is replace the letters ٱ, ٳ, أ, آ with ا
- Replace English question marks "?" with Arabic question marks "؟" to unify the letters.

3. Experimental design

We evaluated the performance of the answer selection classification model using several variants of transformer-based models on various data sets compiled from our developed corpus. In particular, we experimented with multiple variants of transformer-based models and compared their performances on the test set. All models were trained on the same training set and hyperparameters were optimized using the same validation set.

3.1 Transformer-based Models

Transformers achieved state-of-the-art performance in multiple NLP tasks. A transformer is a deep learning language model that is trained on a huge corpus to solve sequence to sequence tasks while easily handling long-range dependencies and predict the probability of the next token given the previous one (Wolf et al., 2020).

In our experiments, we used several Bidirectional Encoder Representations from Transformers (BERT) variants (Devlin et al., 2019). BERT is a transformer-based language model that represents the embedding based on the context of the sentence. There are various Arabic pretrained language models available, including AraBERT (Othman et al., 2020) and CAMeLBERT (Inoue et al., 2021), all of them are

based on transformers produced recently by the Arabic NLP community. The pretrained AraBERT and CAMeLBERT language models are fine-tuned with the target dataset using TensorFlow Estimators[1] and transformers for sequence classification[2], respectively. The maximum sequence length selected was 256 tokens in each QA pair, since the average length of questions and answers is 23.9 and 34.2 respectively. From previous experience using a larger sequence (Thuwaini & Alhumoud, 2022), length (512) has no effect on the accuracy and doubles the execution time.The models are as follows:

- AraBERT version 2 *(*AraBERT v2) is a version of the AraBERT (Othman et al., 2020), which uses pre-segmentation based on Farasa segmentation (Abdelali et al., 2016) that segment words into stems, prefixes, and suffixes.
- AraBERT version 0.2 (AraBERTv0.2) is a version of the AraBERT(Othman et al., 2020), which uses BERT-compatible tokenization, and the text is preprocessed without the use of Farasa segmentation.
- CAMeLBERT-MSA (Inoue et al., 2021) which pretrained with modern standard Arabic
- CAMeLBERT-DA (Inoue et al., 2021) which pretrained with dialectal Arabic
- CAMeLBERT-CA (Inoue et al., 2021) which pretrained with classical Arabic
- CAMeLBERT-Mix (Inoue et al., 2021) which is pretrained with a mix of Modern Standard Arabic (MSA), dialectal, and classical Arabic.

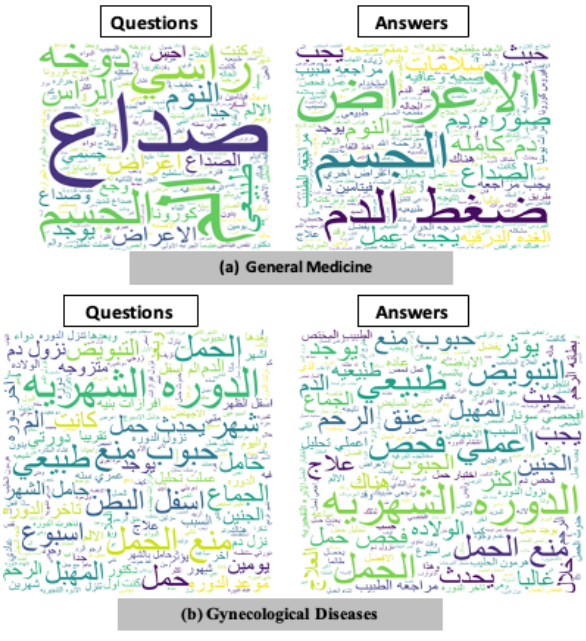


Figure 3: Frequently discussed medical topics in the top two categories in AraMed

3.2 Hybrid Sentence Embedding Models

To tackle the challenge of contextually representing the data, we developed a hybrid model. The first step involves using AraBERTv0.2, in combination with Sentence-BERT (Reimers & Gurevych, 2019), to encode the text data. Unlike BERT, which can capture the contextual relationships between words and phrases, Sentence-BERT is designed to represent entire sentences. By using these two models together, the input text can be encoded into a fixed-dimensional vector representation with 768 dimensions that capture the semantic similarity. Next, the sentence's contextual embedding is fed into a Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM) model. A dense layer with a sigmoid activation function is added to the model to predict the similarity.

4. Results

In this section, we present the characteristics of AraMed where we analyze various aspects of the annotated corpus, including the category distribution, and the top medical topics discussed in the corpus. Additionally, we present the performance of various deep learning models on the AraMed dataset for the answer selection task.

4.1 Descriptive Results

The AraMed corpus contains more than 270k questions and answers pairs about numerous medical topics, with an average of 23.9 tokens per medical question and an average of 34.2 tokens per answer. The dataset statistics are in Table 1 and an example from the AraMed corpus is shown in Figure 1.

We listed the categories where the questions in our corpus posted, and total number of questions for each category. For a detailed breakdown of descriptive statistics by category, please refer to Appendix Table 3. The three topics with the highest number of questions are gynecological diseases, sexual health, and general medicine, with more than 10,000 questions per topic.

In this collection, we used word clouds to visualize the highest frequency words in the top two categories to gain insights into the most frequent unigrams and bigrams used. Figure 3 shows word clouds of the most frequent words associated with the questions and answers for top two categories, which are general medicine and gynecological diseases. In general, the diagram shows that the most frequently occurring terms in questions from the general medicine category are صداع "*headache*", راس *"head"*, "*drowsy*", الأعراض *"symptoms"*, ألم الجسم *"body aches"*, and النوم *"sleep"*. Word clouds representing most frequent answers for questions in general medicine are as follows: الأعراض *"symptoms"*, ضغط الدم *"blood pressure"*, سلامات *"feel better"*, مراجعة طبيب *"doctor's visit"*, صورة دم

[1] https://github.com/aub-mind/arabert/

[2] https://huggingface.co/docs/transformers/model_doc/bert#transformers.BertForSequenceClassification

"blood sample". For questions related to gynecological diseases, the most frequent words are دورة شهرية *"menstruation"*, الحمل *"pregnancy"*, حبوب منع الحمل *"birth control pill"*, منع الحمل *"contraception"*, and اسفل البطن *"lower abdomen"*. Meanwhile, common words in the answers are تبويض *"ovulation"*, طبيعي *"normal"*, اعملي فحص *"do a check"*.

4.2 Experimental Results

The results presented in Table 2 indicate that the AraBERT models emerged as the top performers. The AraBERTv2 achieved the highest results, with F1 score 96.73%, demonstrating its effectiveness in capturing the contextual semantic of the Arabic language. AraBERTv0.2 closely followed, showing that both versions of AraBERT are highly capable in this task. The CAMeLBERT models, trained on different Arabic text forms, showed good performance, with CAMeLBERT-MSA leading among them.

On the other hand, hybrid models integrating Sentence Transformers with GRU and LSTM showed limitations, indicating the complexity of mixing sentence-level embeddings with sequence models. This finding suggests there is potential for advancing sentence embedding techniques to achieve more precise contextual and semantic representations.

Model	**Accuracy (%)**	**F1 (%)**
AraBERTv2	96.71	96.73
AraBERTv0.2	96.54	96.56
CAMeLBERT-MSA	95.78	95.74
CAMeLBERT-DA	93.72	93.62
CAMeLBERT-CA	94.38	94.31
CAMeLBERT-Mix	95.19	95.15
Sentence transformer AraBERT + GRU	86.29	89.49
Sentence transformer AraBERT + LSTM	86.81	86.57

Table 2: Finetuning and Hybrid models performance

5. Conclusion and Future Work

This paper presents large-scale Arabic Medical Question Answering Corpus (AraMed). AraMed addresses the need for a large-scale resource to study Arabic medical question answering system, answer selection and related tasks. Our evaluation demonstrates the value of AraMed. First, it serves as a benchmark for further research on answer selection and related tasks, achieving solid baseline performance with different variants of pretrained transformer language models. Moving forward, to explore future directions, tracks including advanced models, multimodal incorporation, and dialect-specific model adaptation, holds immense potential for building more efficient, informative, and user-centric medical question answering systems. Also, the QA model could be enhanced by sampling irrelevant answers that are semantically similar. That is done by not skipping similar categories when selecting irrelevant answers. By addressing these future directions, AraMed can become a crucial resource for advancing research in Arabic medical question answering and foster further exploration in related NLP tasks.

6. Ethical Consideration

We ensured the dataset protects user privacy. We anonymized question URLs and user IDs. We went a step further by removing any additional details that could potentially identify individuals, such as email addresses, physical locations, names, or website links. This comprehensive anonymization process allows researchers to safely use the data without the risk of uncovering personal information. AraMed is intended only for research purposes. Following a review process to understand the requester's intent and ensure responsible use, we plan to share the corpus upon research requests to facilitate further advancement in medical question answering research in Arabic.

7. References

Abdelali, A., Darwish, K., Durrani, N., & Mubarak, H. (2016). Farasa: A Fast and Furious Segmenter for Arabic. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, 11–16. https://doi.org/10.18653/v1/N16-3003

Adlouni, Y. El, Rodríguez, H., Meknassi, M., El Alaoui, S. O., & En-nahnahi, N. (2019). A multi-approach to community question answering. *Expert Systems with Applications*, *137*, 432–442. https://doi.org/10.1016/j.eswa.2019.07.024

Agirre, E., Banea, C., Cer, D., Diab, M., Gonzalez-Agirre, A., Mihalcea, R., Rigau, G., & Wiebe, J. (2016). SemEval-2016 Task 1: Semantic Textual Similarity, Monolingual and Cross-Lingual Evaluation. *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, 497–511. https://doi.org/10.18653/v1/S16-1081

Alasmari, A., & Zhou, L. (2019). How multimorbid health information consumers interact in an online community Q&A platform. *International Journal of Medical Informatics*, *131*, 103958. https://doi.org/10.1016/J.IJMEDINF.2019.103958

Alasmari, A., & Zhou, L. (2021). Share to Seek: The Effects of Disease Complexity on Health Information—Seeking Behavior. *J Med Internet Res*, *23*(3), e21642. https://doi.org/10.2196/21642

Balla, H. A. M. N., Llorens Salvador, M., & Delany, S. J. (2022). Arabic Medical Community Question Answering Using ON-LSTM and CNN. *2022 14th International Conference on Machine Learning and Computing (ICMLC)*, 298–307. https://doi.org/10.1145/3529836.3529913

Boudjellal, N., Zhang, H., Khan, A., Ahmad, A.,

Naseem, R., & Dai, L. (2020). A Silver Standard Biomedical Corpus for Arabic Language. *Complexity*, *2020*, 8896659. https://doi.org/10.1155/2020/8896659

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *ArXiv*, *abs/1810.0*.

Faris, H., Habib, M., Faris, M., Alomari, A., Castillo, P. A., & Alomari, M. (2022). Classification of Arabic healthcare questions based on word embeddings learned from massive consultations: A deep learning approach. *Journal of Ambient Intelligence and Humanized Computing*, *13*(4), 1811–1827. https://doi.org/10.1007/s12652-021-02948-w

Fehri, H., Dardour, S., & Haddar, K. (2022). ARmed question answering system. *Concurrency and Computation: Practice and Experience*, *34*(21), e7054. https://doi.org/10.1002/cpe.7054

Guo, Q., Cao, S., & Yi, Z. (2022). A medical question answering system using large language models and knowledge graphs. *International Journal of Intelligent Systems*, *37*(11), 8548–8564. https://doi.org/10.1002/int.22955

Inoue, G., Alhafni, B., Baimukan, N., Bouamor, H., & Habash, N. (2021). The Interplay of Variant, Size, and Task Type in Arabic Pre-trained Language Models. *WANLP*.

Lai, T. M., Bui, T., & Li, S. (2018). A Review on Deep Learning Techniques Applied to Answer Selection. *Proceedings of the 27th International Conference on Computational Linguistics*, 2132–2144.

Liu, Q., Agichtein, E., Dror, G., Maarek, Y., & Szpektor, I. (2012). When web search fails, searchers become askers. *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '12*, 801. https://doi.org/10.1145/2348283.2348390

Mishra, A., & Jain, S. K. (2016). A survey on question answering systems with classification. *Journal of King Saud University - Computer and Information Sciences*, *28*(3), 345–361. https://doi.org/10.1016/j.jksuci.2014.10.007

Nakov, P., Hoogeveen, D., Màrquez, L., Moschitti, A., Mubarak, H., Baldwin, T., & Verspoor, K. (2017). SemEval-2017 Task 3: Community Question Answering. *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, 27–48. https://doi.org/10.18653/v1/S17-2003

Othman, N., Faiz, R., & Smaïli, K. (2020). Improving the Community Question Retrieval Performance Using Attention-Based Siamese LSTM. In E. Métais, F. Meziane, H. Horacek, & P. Cimiano (Eds.), *Natural Language Processing and Information Systems* (pp. 252–263). Springer International Publishing.

Reimers, N., & Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv Preprint arXiv:1908.10084*.

Roy, P. K., Saumya, S., Singh, J. P., Banerjee, S., & Gutub, A. (2023). Analysis of community question-answering issues via machine learning and deep learning: State-of-the-art review. *CAAI Transactions on Intelligence Technology*, *8*(1), 95–117. https://doi.org/10.1049/cit2.12081

Thuwaini, W. A., & Alhumoud, S. (2022). TAQS: An Arabic Question Similarity System Using Transfer Learning of BERT With BiLSTM. *IEEE Access*, *10*, 91509–91523. https://doi.org/10.1109/ACCESS.2022.3198955

Wicks, P., Massagli, M., Frost, J., Brownstein, C., Okun, S., Vaughan, T., Bradley, R., & Heywood, J. (2010). Sharing Health Data for Better Outcomes on PatientsLikeMe. *Journal of Medical Internet Research*, *12*(2), e19. https://doi.org/10.2196/jmir.1549

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., & others. (2020). Transformers: State-of-the-art natural language processing. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 38–45.

Zhang, W., Chen, Z., Dong, C., Wang, W., Zha, H., & Wang, J. (2021). Graph-Based Tri-Attention Network for Answer Ranking in CQA. *ArXiv*, *abs/2103.03583*. https://api.semanticscholar.org/CorpusID:232135063

8. Appendices

8.1 Dataset Description

Each question “Q” has question ID, question category, gender, age, question time, question title, question description, number of answers, relevant answer ID, date of relevant answer, relevant answer, the answer specialty of the physician, irrelevant answer ID, doctor specialty of irrelevant answer, date of irrelevant answer, irrelevant answer.

To offer a deeper understanding of the AraMed corpus, let's delve into the nature of the questions and answers. The AraMed corpus is primarily composed of questions and answers formulated in Modern Standard Arabic (MSA). This reflects the platform's focus on providing reliable and standardized health information. However, the corpus also encompasses a variety of question types. Users engage in both knowledge-seeking queries, such as "What are the symptoms of the common cold?", and those seeking specific consultations, like "I have a persistent cough. What could it be?". This diversity enriches the data by reflecting real-world information needs within Arabic-speaking communities.

Categories	Number of Questions
Gynecological diseases	24399
Sexual health	18569
General Medicine	10656
Musculoskeletal and joint diseases	7898
Skin disease	5574
Gastrointestinal diseases	5567
Urinary and venereal diseases	5063
Sexually transmitted diseases	4980
Pharmacology	4266
Cardiovascular disease	4113
General Surgery	3513
Dental disease	2410
Otolaryngology	3364
Pregnancy and birth	3140
Psychiatric illness	2753
Children's diseases	2713
Internal medicine	2684
Ophthalmology	2277

Table 3: Total number of questions for each category.

The Multilingual Corpus of World's Constitutions (MCWC)

Mo El-Haj and Saad Ezzini
UCREL NLP Group
School of Computing and Communications
Lancaster University, Lancaster, UK
{m.el-haj, s.ezzini}@lancaster.ac.uk

Abstract

The "Multilingual Corpus of World's Constitutions" (MCWC) is a rich resource available in English, Arabic, and Spanish, encompassing constitutions from various nations. This corpus serves as a vital asset for the NLP community, facilitating advanced research in constitutional analysis, machine translation, and cross-lingual legal studies. To ensure comprehensive coverage, for constitutions not originally available in Arabic and Spanish, we employed a fine-tuned state-of-the-art machine translation model. MCWC prepares its data to ensure high quality and minimal noise, while also providing valuable mappings of constitutions to their respective countries and continents, facilitating comparative analysis. Notably, the corpus offers pairwise sentence alignments across languages, supporting machine translation experiments. We utilise a leading Machine Translation model, fine-tuned on the MCWC to achieve accurate and context-aware translations. Additionally, we introduce an independent Machine Translation model as a comparative baseline. Fine-tuning the model on MCWC improves accuracy, highlighting the significance of such a legal corpus for NLP and Machine Translation. MCWC's diverse multilingual content and commitment to data quality contribute to advancements in legal text analysis within the NLP community, facilitating exploration of constitutional texts and multilingual data analysis.

Keywords: Constitutions, Corpus, Legal Documents, Fine-tuning, Machine Translation.

1. Introduction and Rationale

The "Multilingual Corpus of World's Constitutions" (MCWC) represents a contribution to the field of legal and multilingual natural language processing. This corpus spans the legal spectrum, with constitutions from across the globe, with a particular emphasis on those available in multiple languages, including English, Spanish, and Arabic. The acronym 'MCWC', is pronounced as 'Makkuk', a word that carries significance in the Arabic language, where it refers to a Space Shuttle مكوك. Constitutional documents, serving as the bedrock of legal systems across the globe, embody the principles and values upon which nations are built. They define the rights, responsibilities, and governance structures that shape societies (Hutson, 1981). These foundational texts, however, often transcend linguistic boundaries, existing in a multitude of languages, each with its unique nuances. The study and analysis of constitutional texts, particularly in a multilingual context, present both an intellectual challenge and an avenue for groundbreaking advancements in the realms of Natural Language Processing (NLP) and legal scholarship (Zhong et al., 2020). This paper introduces a contribution to the intersection of language technology and legal studies – The MCWC Corpus. Our corpus, comprising 223 constitutions from 191 countries, encompassing both current and previous versions of constitutions and offering translations into English, Arabic, and Spanish. Serving as a multilingual bridge, it connects legal documents across diverse linguistic backgrounds. Each country is accessible in English, with 95 constitutions available in all three languages, facilitating comprehensive multilingual research. Through an automatic translation pipeline, we expanded coverage to include all three languages for all 223 constitutions. Our experiments highlight the corpus's potential for the NLP community and researchers in constitutional analysis, machine translation, and cross-lingual legal studies.

MCWC holds importance beyond individual disciplines. Within its digital pages lies the constitutional heritage of nations, united by themes of justice, governance, and the rule of law. This corpus enables insights into the development of legal thought across cultures and languages, revealing shared values underlying global legal systems (Blaustein, 1991). In addition, it serves as a catalyst for research, driving progress in fields such as machine translation, information retrieval, cross-cultural legal studies, and beyond.

1.1. Motivation

MCWC Corpus emerges from a profound motivation rooted in the convergence of legal scholarship and NLP. Constitutional documents, as the embodiment of a nation's values and legal principles, hold paramount importance in the legal domain (Chitere et al., 2006). However, their analysis and cross-lingual study pose substantial challenges,

OSACT 2024 Workshop, pages 57–66
25 May, 2024.

and this corpus addresses these challenges with precision and foresight (Driskill et al., 2010).

In the field of Natural Language Processing (NLP), the domain of multilingualism represents a continually expanding frontier. The capacity to effectively process, analyse, and translate legal documents across different languages stands as a crucial milestone in language technology development (Wiesmann, 2019). MCWC plays a role in propelling forward the capabilities of NLP in the legal domain. By offering access to constitutional texts in multiple languages, it opens up fresh avenues for research and advancement in machine translation, sentiment analysis, summarisation, and various other NLP tasks within the legal context (Katz et al., 2023). MCWC has the potential to enhance state-of-the-art machine translation models through fine-tuning on constitutional texts, benefiting multilingual societies and legal practitioners, which enables comparative legal studies, shedding light on how legal concepts vary across languages and jurisdictions (Katz et al., 2023). This cross-cultural analysis contributes to an understanding of the global legal landscape. By making this resource publicly available, we encourage interdisciplinary collaboration and innovation across NLP, law, political science, linguistics, and more.

2. Related Work

The intersection of natural language processing and legal scholarship has sparked significant interest in recent years (Katz et al., 2023; Sanchez, 2019; Zhong et al., 2020; Moreno-Schneider et al., 2020). Researchers have explored various facets of legal text analysis, including case law, statutes, and regulations. However, the specific domain of constitutional texts, especially in a multilingual context, presents a unique set of challenges and opportunities (Shaheen et al., 2020; Lenci et al., 2007; Tsarapatsanis and Aletras, 2021).

The Comparative Constitutions Project (CCP)[1] is a research initiative dedicated to the comprehensive study of constitutions from across the globe. It has compiled a vast repository of constitutional texts, aiming to facilitate in-depth analysis of constitutional design, governance dynamics, and the intricate factors shaping the evolution of national constitutions (Elkins et al., 2009). It is worth noting that the original dataset was not optimised for advanced NLP and Machine Learning research. Lacking suitable formatting and organisation, our efforts were focused on formatting and extracting relevant text from each constitution. We have undertaken extensive cleaning, alignment, and refinement processes. Missing constitutions were collected from various sources, including each nation's government websites and Wikipedia[2]. In addition, our work on fine-tuning machine translation (MT) models on the MCWC has enabled us to compile a comprehensive list of the world's constitutions in all three languages (English, Arabic and Spanish), surpassing the offerings available on the CCP, government websites, or Wikipedia. This means that our collection includes translations for constitutions that were previously unavailable in multiple languages through conventional sources.

Legal NLP has evolved rapidly with advancements in machine learning and deep learning techniques. Early work focused on legal document classification and information retrieval, laying the groundwork for subsequent research (Wang et al., 2023). Recent efforts have turned to machine translation, with initiatives like the European Union's eTranslation project aiming to provide automated translation services for legal texts within the EU[3]. However, these initiatives often focus on specific languages and legal domains, leaving a gap in comprehensive multilingual constitutional analysis.

The development of multilingual corpora has played a pivotal role in the training and assessment of NLP models. Projects such as Universal Dependencies (UD) and Parallel Universal Dependencies (PUD) have assembled parallel datasets across multiple languages, facilitating research in areas like cross-lingual dependency parsing and sentiment analysis (De Marneffe et al., 2021). However, it is important to note that these corpora primarily consist of general text data and do not focus on specialised legal content. In a similar vein, the UN MultiUN Corpus is worth mentioning as it offers a multilingual corpus derived from United Nations documents, which, while not specific to legal content, represents another valuable resource for multilingual NLP research (Eisele and Chen, 2010).

Constitutional analysis has long been a cornerstone of legal scholarship (Bhagwat, 1997). Scholars have explored various dimensions of constitutional texts, including textual structure, legal reasoning, and historical context (Gammelgaard and Holmøyvik, 2014). However, much of this work has been conducted within specific linguistic and jurisdictional boundaries. Comparative constitutional analysis, which seeks to identify commonalities and differences across constitutions, has traditionally relied on manual examination and translation, presenting significant challenges in cross-lingual research (Bruteig, 1814).

[1] https://comparativeconstitutionsproject.org

[2] www.wikipedia.org

[3] https://commission.europa.eu/resources-partners/etranslation_en

Table 1: Statistics by Continent

Continent	Countries	Constitutions	Words	Tokens	Avg_Word	TTR
Africa	51	65	1,877,335	1,520,717	28,444.5	0.810
Asia	47	54	1,445,844	1,135,877	26,774.9	0.786
Europe	44	49	1,452,992	1,158,030	29,652.9	0.797
North America	23	26	1,121,426	875,036	43,131.8	0.780
Oceania	14	14	514,347	404,127	36,739.1	0.786
South America	12	15	1,137,263	934,696	75,817.5	0.822

Table 2: Statistics by Country (sample out of 191 countries)

Country	Continent	#Const	Avg_Words	Lang	TTR	Const_Years
Egypt	Africa	2	42,190.5	en, es, ar	0.832	2012, 2019
France	Europe	1	37,755	en, es, ar	0.826	2008
Argentina	South America	1	35,108	en, es, ar	0.806	1994
Australia	Oceania	1	41,735	en, es, ar	0.773	1985
Japan	Asia	2	8,066	en, es, ar	0.849	1889, 1946
USA	North America	1	22,275	en, es, ar	0.737	1992

3. Dataset and Preparation

We assemble a diverse corpus of constitutional texts from various countries, spanning continents and languages[4]. These constitution texts are sourced from publicly available data provided by the Comparative Constitutions Project[5] and Constitute Project[6] as well as Wikipedia and Government official websites. Initially, the data consists of the text of constitutions from 191 countries, primarily in XML format. In cases where XML files were unavailable, we resorted to extracting the constitution text directly from the respective country's governmental website. However, these XML files do not consistently adhere to the same tagging format, leading to challenges when extracting content, particularly in cases where a constitution is available in multiple languages. The Constitute Project site's service methods and detailed API documentation to enable developers to retrieve constitution and topic data[7]. To enhance accessibility, we not only created a corpus from this data but also augmented it to include additional constitutions in Arabic and Spanish, while ensuring alignment, refinement, and cleanliness, making the corpus ready for optimal use in NLP and ML applications in a standardised format, such as CSV.

Notably, aligning sentences across languages was achieved through an automated parser developed explicitly for this purpose. The parser relies on structural information present in the text itself, such as section numbers, article identifiers (e.g., Section 1, Artículo 1, and الفصل 1).

We employ straightforward gazetteer matching techniques to categorise constitutions according to their respective continents, facilitating a coarse-grained level of comparative analysis. This prepared dataset serves as the cornerstone for training and evaluating our machine learning model, enabling comprehensive research in constitutional analysis, machine translation, and cross-lingual legal studies.

Table 1 provides statistics organised by continent for MCWC. It presents a breakdown of various metrics, including the number of countries represented in each continent, the total number of constitutions available, the total word count, token count, average words per constitution, and the Type-Token Ratio (TTR). These statistics offer insights into the composition and characteristics of the corpus across different continents. For example, it is evident that South America has the highest average words per constitution and the highest TTR among the continents listed, indicating linguistic diversity and potentially complex legal language[8]. Conversely, North America has the lowest TTR, suggesting a lower degree of linguistic variation in its constitutional texts.

Table 2 summarises key statistics for countries within the MCWC. It includes information about each country's continent, the number of constitutions available from that country, the average word count across those constitutions, the number of languages in which the constitutions are available, TTR for the country's constitutions, and the span

[4] In the case of the UK, the Magna Carta is included in the MCWC Corpus as it serves as a foundational document, given the absence of a single written constitution in the country.

[5] comparativeconstitutionsproject.org

[6] https://constituteproject.org/

[7] https://constituteproject.org/content/data

[8] This takes into consideration constitutions available in languages other than English; i.e. Spanish and Arabic

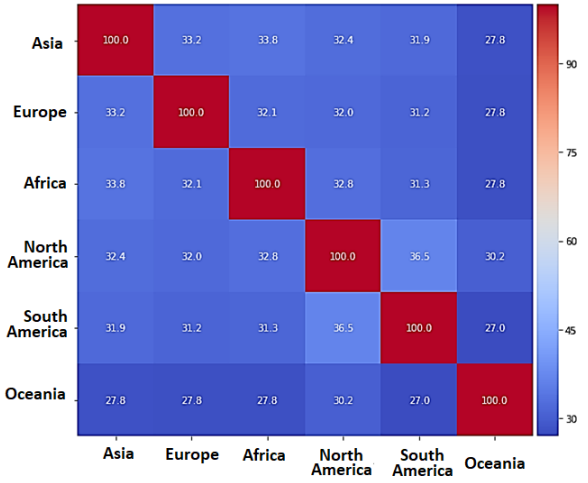


Figure 1: By-continent Vocabulary Overlap Heatmap for Constitutions written in English

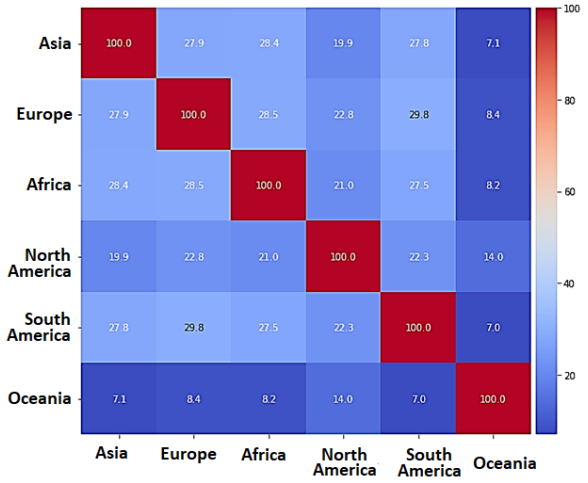


Figure 2: By-continent Vocabulary Overlap Heatmap for Constitutions written in Arabic

of years during which these distinct constitutions were enacted, revised or to account for constitutional reforms, historical changes, or different iterations over time.

Figures 1-3 show heatmaps displaying vocabulary overlap among continents' constitutions in English, Arabic and Spanish, respectively. These heatmaps provide a view of the linguistic commonalities and shared legal terminology across continents, facilitating cross-lingual legal studies and machine translation research.

4. MCWC Cosine Similarity Analysis

In the pursuit of assessing the similarities between the constitutions of diverse countries, our analysis commenced with the extraction of pertinent texts from a formatted CSV dataset. In order to facilitate a comparative analysis across continents, we

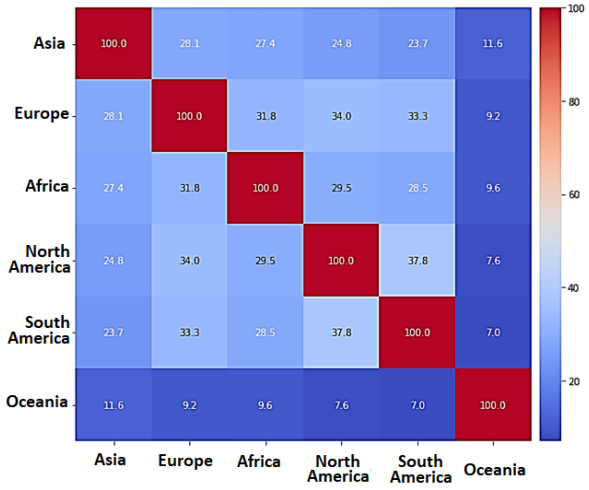


Figure 3: By-continent Vocabulary Overlap Heatmap for Constitutions written in Spanish

judiciously employed another CSV file to establish the mappings of countries to their respective continents. Our primary focus during this investigation remained directed towards the English language text, as the constitutions of each country in our corpus is available in the English language.

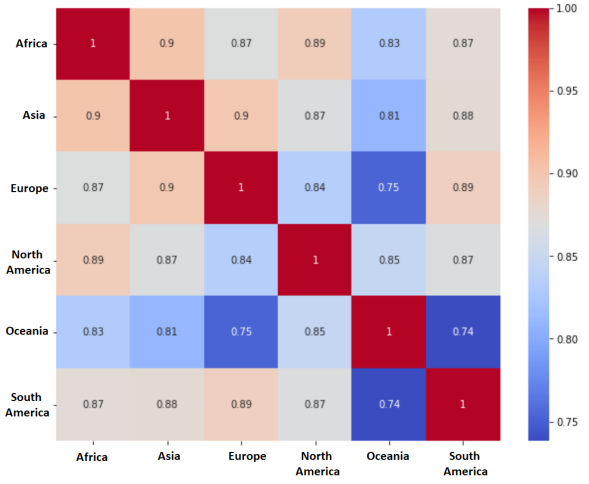


Figure 4: Cosine Similarity Between Continents (English)

Prior to embarking on the intricacies of similarity calculations, we conducted a series of text-cleansing procedures. In addition to removing common English stop-words, this initial stage involved the elimination of frequently occurring yet extraneous terms, such as "Article" and "Preamble", which were deemed irrelevant to the core analysis for being very repetitive. Furthermore, we removed numeric values and any special characters, thereby ensuring that our dataset was composed of unadulterated textual content. This preparation enabled us to explore the similarities

Table 3: Cosine Similarity Between Continents

Continents	Sim
Africa - Asia	0.90
Asia - Europe	0.90
Africa - North America	0.89
Europe - South America	0.89
Asia - South America	0.88
Africa - South America	0.87
Africa - Europe	0.87
Asia - North America	0.87
North America - South America	0.87
North America - Oceania	0.85
Europe - North America	0.84
Africa - Oceania	0.83
Asia - Oceania	0.81
Europe - Oceania	0.75
Oceania - South America	0.74

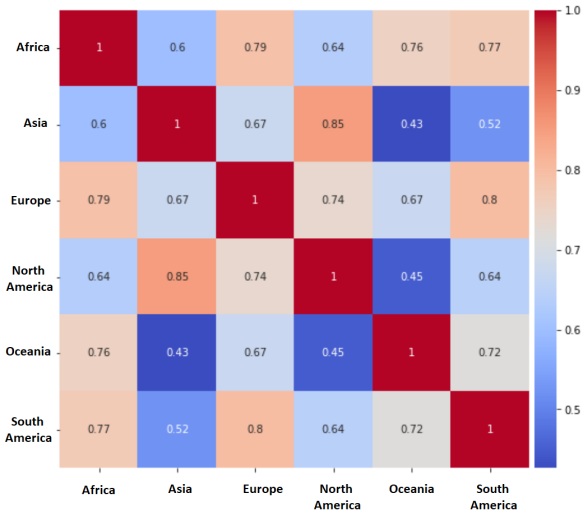


Figure 5: Cosine Similarity Between Continents (English - normalised)

between constitutions in greater depth.

Table 3 and Figure 4 present the cosine similarity values between continents without normalisation. These values range between 0.74 and 0.90, indicating the degree of resemblance between the constitutions of different continents. Notably, the highest similarity of 0.90 is observed between Africa and Asia, suggesting a substantial overlap in the content and structure of their constitutions. Similarly, the similarities between Asia and Europe (0.90) and Africa and North America (0.89) are noteworthy, indicating significant commonalities, shedding light on nuanced patterns in our corpus and aligning with the insights gleaned from Figures 1-3.

In our pursuit of objectivity, we proactively addressed the potential for bias towards continents endowed with a greater number of constitutions. To mitigate this, we undertook the essential step of vector normalisation. This involved the computation of average TF-IDF vectors for all countries within each continent and ensured that the representation of each continent's constitutional subcorpus remained equitable and unaffected by the quantity of constitutions it contributed (Table 4 and Figure 5). The normalisation process resulted in a drop in similarity scores, providing a clearer understanding of the true relationships between constitutional texts across different regions.

5. Constitutions Machine Translation

In organising our corpus, we adhered to a hierarchical structure aligned with the organisation of constitutional data on the Constitute Project website. The data is made publicly available there

Table 4: Cosine Similarity Between Continents (normalised)

Continents	Sim
Europe - South America	0.85
Africa - North America	0.80
Asia - Africa	0.79
Asia - North America	0.77
Asia - Oceania	0.76
Africa - South America	0.74
Oceania - North America	0.72
Europe - Africa	0.68
Africa - Oceania	0.67
South America - North America	0.64
Asia - South America	0.64
Asia - Europe	0.60
Europe - North America	0.52
South America - Oceania	0.45
Europe - Oceania	0.43

in XML format[9]. Our process involved extracting content segments tagged with language attributes, specifically English, Arabic, or Spanish, from these publicly accessible XML files. To facilitate NLP tasks like translations, we subsequently converted the constitutional text data into CSV format. This conversion also included the assignment of a consistent unique identifier (Align#) to each sentence across various languages. This identifier plays a pivotal role in simplifying the alignment of sentences during our machine translation experiments.

In preparation for our multilingual machine translation tasks, we curated the dataset to include constitutions available in at least two of the three lan-

[9]Example: Constitution of Argentina as XML: constituteproject.org/countries/Americas/Argentina

guages: English, Spanish, and Arabic. However, due to the shortage of constitutions available in Arabic and Spanish, we employed machine translation techniques through fine-tuning and training to augment the missing constitutions in these languages. Specifically, we translated the English versions of constitutions into Arabic and Spanish using the state-of-the-art Neural Machine Translation model Facebook's Seamless-m4t-v2-large[10].

To evaluate the effectiveness of this approach, we conducted a thorough assessment. First, we randomly sampled 500 constitution pairs in English-Arabic and English-Spanish to ensure the quality of our translation model. The assessment revealed a BLEU score of 0.68, which, within the context of this specific dataset and language pairs, suggests a high level of translation accuracy and is indicative of the model's effectiveness (Chouigui et al., 2021; El-Haj et al., 2014).

Additionally, we performed a human evaluation specifically for the augmented versions in Arabic. Two expert annotators, well-versed in the Arabic language and Arabic NLP, and both proficient in English, manually pair-annotated 50 paragraphs randomly selected from the Arabic translations of constitutions[11]. Initially, the inter-annotator agreement was measured using Cohen's Kappa, yielding a score of 0.30 with an agreement rate of 91% on positive translation quality.

To account for the substantial imbalance in the distribution of agreement categories and to provide a more robust measure of inter-annotator reliability, we further analysed the data using Krippendorff's Alpha. This metric, which is less sensitive to such imbalances and suitable for a variety of data levels, yielded a more accurate reflection of agreement at an impressive score of approximately 0.90. This high value indicates a good level of agreement between the annotators, reinforcing the reliability of the manual annotations despite the predominance of one category. The primary source of disagreement is explained in the Error Analysis (Section 5.1).

In total, our dataset encompasses pairwise sentence alignments across selected languages, resulting in 52,177 sentence pairs for English-Arabic (En-Ar), 48,892 for English-Spanish (En-Es), and 27,352 for Arabic-Spanish (Ar-Es). Additionally, we augmented our dataset to include a total of 236,156 parallel sentences in English, Arabic, and Spanish using the above-mentioned Facebook's Seamless-m4t-v2-large translation model. These language pairs and parallel sentences, along with our machine translation approach and evaluation results, are made available for reproduction and research purposes[12].

5.1. Error Analysis

The primary source of disagreement between annotators was rooted in the completeness of Arabic translations, which tended to be more concise than their English counterparts. This conciseness in translation, rather than a reduction in translation quality, contributed to discrepancies in the inter-annotator agreement metrics. Notably, one annotator would still deem a translation accurate and correct even if it did not translate the original text word for word, focusing instead on the preservation of overall meaning and intent.

Cohen's Kappa, yielding a score of 0.30 with a 91% agreement rate, may have been influenced by these variations. The kappa score, while indicative of a fair level of agreement, does not fully capture the essence of the translations' quality due to its sensitivity to the imbalance in the distribution of agreement categories.

Conversely, Krippendorff's Alpha, with a score of approximately 0.90, provided a more nuanced understanding of the inter-annotator agreement. By accommodating the data's imbalance and focusing on the ratio of observed to expected disagreement, Krippendorff's Alpha highlighted the consistency of the annotations in evaluating the translation quality, underscoring the annotators' alignment on the translations' overall fidelity to meaning despite variances in completeness.

The following examples illustrate instances where annotators disagreed, yet the translations remained faithful to the source material's essence:

1. "When both the Pyithu Hluttaw and the Amyotha Hluttaw have certain matters to study, apart from matters to be performed by the Committees as prescribed in Sub-Sections (a) and (b) of Section 115, the Speakers of these Hluttaws may co-ordinate among themselves and form a Joint Committee comprising an equal number of representatives from the Pyithu Hluttaw and the Amyotha Hluttaw. The Pyithu Hluttaw may elect and assign the Pyithu Hluttaw representatives included in that Committee." was translated as:

   عندما يكون لكل من Pythus Hluttaw و Amyutha Hluttaw بعض المسائل لدراستها ، باستثناء المسائل التي يجب أن تؤديها اللجان كما هو محدد في الفقرتين الفرعيتين (a) و (b) من المادة 115 ، يمكن لرؤساء هذه اللجان التنسيق بينهم وتشكيل

[10] https://huggingface.co/facebook/seamless-m4t-v2-large

[11] This smaller sample size, while offering valuable insights, might not capture the full dataset's diversity. Further research with a broader corpus is recommended to enhance the robustness of these results.

[12] https://huggingface.co/collections/ezzini

لجنة مشتركة تضم عدداً متساوياً من ممثلي Hluttaw Pythus و Hluttaw Amyutha ، يمكن لـ Hluttaw Pythus أن ينتخب ويعين Hluttaw Pythus المضمنين في تلك اللجنة.

2. “In order to provide for decentralised administration of the administrative divisions of the Maldives, elections to island councils, atoll councils and city councils as provided for in this Constitution shall be held before 1 July 2009.” was rendered as:

   من أجل توفير إدارة لامركزية للتقسيمات الإدارية في جزر المالديف ، سيتم إجراء الانتخابات لمجلس الجزر ومجالس الجزر المرجانية ومجالس المدن كما هو المنصوص عليه في هذا الدستور قبل 1 يوليو 2009.

3. “Whose father or mother, on the sixth day of August, 1962, became or would but for his or her death have become a citizen of Jamaica in accordance with subsection (1) of section 3,” translated to:

   الذي أصبح والده أو والدته في السادس من أغسطس 1962 مواطناً جامايكا وفقاً للفقرة الفرعية (1) من القسم 3.

4. The numeral “Four” was translated as “رابعاً”.

These examples underscore the annotators’ ability to navigate the complexities of linguistic and cultural nuances, ensuring the translations’ integrity while accommodating the inherent brevity of the Arabic language.

5.2. Machine Translation Setup

In the course of this research, we have established an experimental framework that leverages state-of-the-art models to empower Machine Translation exploration. Recognising the multilingual parallel nature of our dataset, we opted to conduct a machine translation experiment, demonstrating the significance of fine-tuning machine learning models on constitutional data. Our setup encompasses the evaluation of six machine translation models on our data, covering the six possible pairs: En-Ar, Ar-En, En-Es, Es-En, Ar-Es, and Es-Ar.

Machine Translation Models: We utilise the state-of-the-art Machine Translation models based on Marian NMT, known for its proficiency for bilingual neural machine translation (NMT)[13].

Fine-Tuning Process: We fine-tuned each Machine Translation model on the corresponding language pair subset of our constitutional corpus using three epochs, and a batch size of 32. The resulting six fine-tuned models are made public in our HuggingFace repository[14].

Evaluation Metric: We use the SacreBLEU implementation of the BLEU score to compare the translation models output with ground-truth[15].

Hardware: The experiments are conducted on a high-performance machine equipped with an NVIDIA GeForce RTX 2080 Ti GPU, accelerating both training and evaluation processes.

6. Results and Evaluation

Table 5: Cumulative BLEU Scores for Machine Translation Models: Original vs. fine-tuned

Pair	Original model	fine-tuned model
Es-En	0.261	0.557
En-Es	0.335	0.475
Ar-En	0.255	0.433
En-Ar	0.177	0.274
Ar-Es	0.216	0.271
Es-Ar	0.093	0.191

The evaluation results, presented in Table 5, demonstrate a significant improvement in the performance of our Machine Translation models following the fine-tuning process. Initially, the original models exhibited commendable BLEU scores across various language pairs, ranging from 0.093 (Es-Ar) to 0.335 (En-Es). However, the true significance of this experiment becomes evident when comparing these scores to those achieved by the fine-tuned models. Across all language pairs, the fine-tuned models consistently outperformed their original counterparts, as illustrated in Fig. 6. For instance, in the En-Ar translation task, the BLEU score increased from 0.177 to 0.274, representing a substantial enhancement in translation quality. Similarly, in the Es-En translation, the BLEU score surged from 0.261 to 0.557. These results underscore the effectiveness of fine-tuning in enhancing the accuracy and fluency of our Machine Translation models, highlighting the tangible quality of our parallel data.

This advancement in machine translation accuracy for constitutional text holds the potential to facilitate the automatic translation of constitutions across the globe into various languages. This capability is especially valuable for languages that may be digitally under-resourced, as it enables broader access to legal and constitutional documents, fostering cross-border collaboration and promoting legal discourse across linguistic boundaries.

[13] https://github.com/Helsinki-NLP/Opus-MT
[14] https://huggingface.co/collections/ezzini
[15] https://github.com/mjpost/sacreBLEU

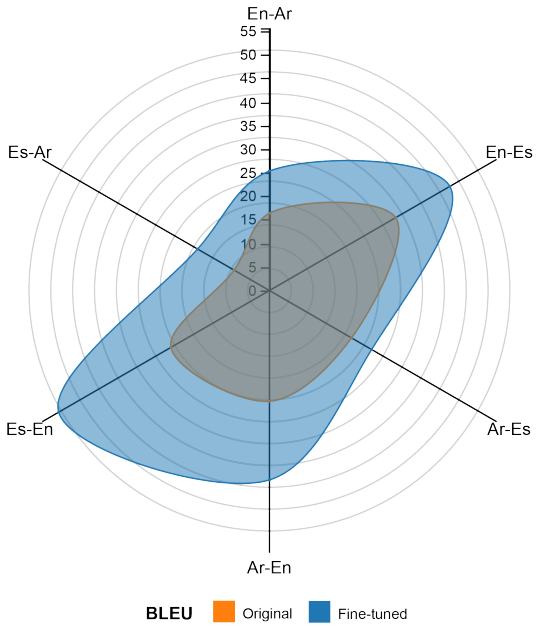


Figure 6: Translation Results

7. Conclusion

In this paper, we have introduced the Multilingual Corpus of World's Constitutions (MCWC), a resource comprising 223 constitutions from 191 countries. What sets MCWC apart is its inclusivity, encompassing not only the current versions of these constitutions but also previous iterations where applicable. The corpus goes beyond mere documentation, offering good quality translations into three prominent languages: English, Arabic, and Spanish. In essence, it provides a multilingual bridge, connecting legal documents from diverse linguistic backgrounds.

Within MCWC, every country is represented in English, underscoring its global accessibility. Furthermore, 95 constitutions are available in all three languages: English, Arabic, and Spanish, facilitating comprehensive multilingual research. Additionally, 58 constitutions are accessible in English and Spanish, while 50 are accessible in English and Arabic. Using our automatic translation pipeline, we augmented the available 223 constitutions to cover all three languages - English, Arabic, and Spanish. Our experiments, as showcased in this paper, leave no room for doubt about the corpus's potential and the exceptional quality of its multilingual aspect. It has the potential to become a valuable tool for the NLP community and researchers across various disciplines, including constitutional analysis, machine translation, and cross-lingual legal studies.

Looking ahead, our plans for MCWC involve ongoing refinement and expansion. We are dedicated to completing pending translations to enhance its comprehensiveness. Additionally, we aim to broaden the linguistic scope of MCWC by incorporating more languages and countries. This expansion seeks to create a more inclusive repository, promoting cross-cultural understanding, facilitating legal discourse, and supporting research in an increasingly diverse and interconnected world.

8. Ethical Considerations

We would like to acknowledge that the data used in the Multilingual Corpus of World's Constitutions (MCWC) has been sourced from the Comparative Constitutions Project[16] and Constitute Project[17], as made available on the Constitute website. The data is provided in open-linked data format, following the standards of the Semantic Web. The Constitute Project site's service methods and detailed API documentation to enable developers to retrieve constitution and topic data[18]. It is important to note that we are not republishing the original data from these projects. Instead, we are providing a processed, cleaned, and aligned version in CSV format for each language pair, as well as a machine-translated version of all English constitutions into Arabic and Spanish. Users who require the original data format can download it directly from the Constitute Project website, which offers service methods and detailed API documentation enabling developers to retrieve constitution and topic data

Limitations

This work has the following potential limitations:

Limited Translation Sources: While the paper utilises English translations from reputable sources like HeinOnline[19] and the Oxford Constitutions of the World[20] , it is important to acknowledge that the quality and comprehensiveness of translations can vary depending on the source. This introduces a potential limitation as the accuracy and nuances of the original texts may not be fully captured in these translations.

Variable Translation Quality: The use of translations provided by different entities, such as International IDEA for Arabic texts[21] and the Human Rights Lab of the University of Los Andes[22] for some Spanish texts, may result in variations in translation quality and consistency. These differ-

[16] comparativeconstitutionsproject.org
[17] https://constituteproject.org/
[18] https://constituteproject.org/content/data
[19] http://home.heinonline.org/
[20] http://oxcon.ouplaw.com/
[21] https://www.idea.int/
[22] https://uniandes.edu.co/en

ences could impact the overall quality of the multilingual corpus and subsequent analyses.

Potential Bias or Omissions: The reliance on translations from specific organisations may introduce bias or omissions in the corpus, as certain constitutional texts or specific nuances may not be included or may be subject to interpretation by the translation providers. This could affect the comprehensiveness and accuracy of the MCWC, potentially limiting its applicability in certain research contexts.

Lack of Control Over Translation Process: There are unavailable details from CCP on the translation process, such as the criteria used for selecting specific translations or the extent to which the translations were reviewed or edited. This lack of transparency regarding the translation process may limit the ability to assess the reliability of the translated texts.

9. Bibliographical References

References

Ashutosh Bhagwat. 1997. Purpose scrutiny in constitutional analysis. *Calif. L. Rev.*, 85:297.

Albert P Blaustein. 1991. Constitution drafting: The good, the bad, and the beautiful. *Scibes J. Leg. Writing*, 2:49.

Yordanka Madzharova Bruteig. 1814. Norwegian parliamentary discourse 2004–2014 on the norwegian constitution's language form. *i: Writing Democracy. The Norwegian Constitution*, 2014:151–163.

Preston Chitere, Ludeki Chweya, Japhet Masya, Arne Tostensen, and Kamotho Waiganjo. 2006. *Kenya Constitutional Documents: a comparative analysis*. Chr. Michelsen Institute.

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.

Marie-Catherine De Marneffe, Christopher D Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal dependencies. *Computational linguistics*, 47(2):255–308.

Samantha Tisdale Driskill, Paige LeForce DeFalco, Jill Holbert Lang, and Janette Habashi. 2010. Constitutional analysis: A proclamation of children's right to protection, provision, and participation. *The International Journal of Children's Rights*, 18(2):267–290.

Andreas Eisele and Yu Chen. 2010. Multiun: A multilingual corpus from united nation documents. In *LREC*.

Mahmoud El-Haj, Paul Rayson, and David Hall. 2014. Language independent evaluation of translation style and consistency: Comparing human and machine translations of camus' novel "the stranger". In *International Conference on Text, Speech, and Dialogue*, pages 116–124. Springer.

Zachary Elkins, Tom Ginsburg, and James Melton. 2009. The comparative constitutions project: A cross-national historical dataset of written constitutions. Technical report, Mimeo Chicago.

Karen Gammelgaard and Eirik Holmøyvik. 2014. *Writing Democracy: The Norwegian Constitution 1814-2014*, volume 2. Berghahn Books.

James H Hutson. 1981. Country, court, and constitution: antifederalism and the historians. *The William and Mary Quarterly: A Magazine of Early American History*, pages 338–368.

Daniel Martin Katz, Dirk Hartung, Lauritz Gerlach, Abhik Jana, and Michael J Bommarito II. 2023. Natural language processing in the legal domain. *arXiv preprint arXiv:2302.12039*.

Alessandro Lenci, Simonetta Montemagni, Vito Pirrelli, and Giulia Venturi. 2007. Nlp-based ontology learning from legal texts. a case study. *LOAIT*, 321:113–129.

Julián Moreno-Schneider, Georg Rehm, Elena Montiel-Ponsoda, Víctor Rodriguez-Doncel, Artem Revenko, Sotirios Karampatakis, Maria Khvalchik, Christian Sageder, Jorge Gracia, and Filippo Maganza. 2020. Orchestrating nlp services for the legal domain. *arXiv preprint arXiv:2003.12900*.

George Sanchez. 2019. Sentence boundary detection in legal text. In *Proceedings of the natural legal language processing workshop 2019*, pages 31–38.

Zein Shaheen, Gerhard Wohlgenannt, and Erwin Filtz. 2020. Large scale legal text classification using transformer models. *arXiv preprint arXiv:2010.12871*.

Dimitrios Tsarapatsanis and Nikolaos Aletras. 2021. On the ethical limits of natural language processing on legal text. *arXiv preprint arXiv:2105.02751*.

Steven H Wang, Antoine Scardigli, Leonard Tang, Wei Chen, Dimitry Levkin, Anya Chen, Spencer Ball, Thomas Woodside, Oliver Zhang, and Dan

Hendrycks. 2023. Maud: An expert-annotated legal nlp dataset for merger agreement understanding. *arXiv preprint arXiv:2301.00876*.

Eva Wiesmann. 2019. Machine translation in the field of law: A study of the translation of italian legal texts into german. *Comparative Legilinguistics*, 37(1):117–153.

Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 2020. How does nlp benefit legal system: A summary of legal artificial intelligence. *arXiv preprint arXiv:2004.12158*.

Tafsir Extractor: Text Preprocessing Pipeline preparing Classical Arabic Literature for Machine Learning Applications

Carl Kruse[1], **Sajawel Ahmed**[1,2]
[1]Goethe University Frankfurt, Germany
[2]University of California, Davis, United States
{carl.kruse,sahmed}@em.uni-frankfurt.de {sajahmed}@ucdavis.edu

Abstract

In this paper, we present a comprehensive tool of preprocessing Classical Arabic (CA) literature in the field of historical exegetical studies for machine learning (ML) applications. Most recent ML models require the training data to be in a specific format (e.g. *XML*, *TEI*, *CoNLL*) to use it afterwards for Natural Language Processing (NLP) tasks such as Named Entity Recognition (NER) or Topic Modeling (TM). We report on how our method works and can be applied by other researchers with similar endeavors. Thereby, the importance of this comprehensive tool of preprocessing is demonstrated, as this novel approach has no predecessors for CA yet. We achieve results that enable the training of current ML models leading to state-of-the-art performance for NER and TM on CA literature. We make our tool along its source code and data freely available for the NLP research community.

Keywords: preprocessing, named entity recognition, topic modeling, machine learning, historical NLP, classical Arabic, low-resource languages, theological studies

1. Introduction

While working within the field of Classical Arabic (CA) literature and its genre of exegetical studies (*tafsir*), it becomes clear that it is a very broad field in which different textual components can be examined along their topics, e.g. the textual component of oral traditions (*hadith*) along the topics of juridical rulings (*fiqh*), linguistics (*lugha*), and judeo-christian sources (*israiliyat*). One exegetical work that is particularly emphasized to research such topics is the monumental book of the theological scholar Al-Tabari (d. 923). His work *Tafsir Al-Tabari* is regarded among the most important exegeses in the Islamic theology and contains a large part of all relevant oral traditions that were in circulation at the beginning of the 10th century (Ahmed et al., 2022a). Through his work it is possible to gain insights into the mentioned topics (e.g. juridical rulings) to understand a given verse and its circumstances for scholars of historical literature.

Given the substantial volume of Al-Tabari's work, extracting and compiling oral traditions on specific topics from classical works to enhance Quranic explanations is a complex task. This complexity is exacerbated by the vast array of topics available for analysis in exegetical works, totaling 15 according to the classical categorization by Al-Suyuti (1505). Therefore, it becomes imperative to undertake digital preparations for such texts. This digital transformation allows efficient access to a wealth of information within the realm of exegesis, facilitating a more effective exploration of diverse topics.

To this end, we develop in this work the tool *Tafsir Extractor*, a comprehensive text preprocessing pipeline for preparing gold data that can be used to train machine learning (ML) models. Several steps are necessary to digitally process the Al-Tabari corpus. First, the corpus is extracted from the resource platform *Gawami' al-Kalim*[1] (GK), which is prepared via an optical character recognition (OCR) process from the original manuscripts. Second, it is annotated manually in the XML/TEI formats according to the annotation guidelines (Ahmed et al., 2022b). Third, the data is converted using our TEI2CoNLL module, a method that has been developed with various different options to convert the data automatically into the CoNLL format (Tjong Kim Sang and De Meulder, 2003) which is necessary for current ML models (Lample et al., 2016; Devlin et al., 2019; Brown et al., 2020) for the task of Named Entity Recognition (NER). Our extended CoNLL

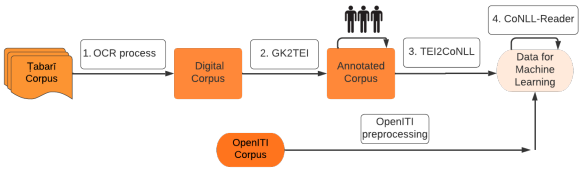


Figure 1: Processing steps for preparing the raw data from CA literature

format also contains a matrix with extracted information on Topic Modeling (TM) data. Additional columns can be added for syntactical and analytical information of a word or a sentence. Finally, this whole dataset can be used for ML evaluations (see Figure 1), either by training the heavyweight large language models with *MaChAmP* (van der

[1]https://gk.islamweb.net

OSACT 2024 Workshop, pages 67–73

Goot et al., 2021) (even in multi-task learning scenarios), or by lightweight embedding-based models (Mikolov et al., 2013; Ling et al., 2015) trained on the unlabeled OpenITI corpus (Miller et al., 2018) from scratch.

Thus, our work allows us to accelerate the process of digitizing information from the historical literature of theological studies. We lay the technical foundations for our ongoing research work on Natural Language Processing (NLP) for CA literature allowing the training and fine-tuning of current ML models for higher-level NLP tasks such as NER or TM (Ahmed et al., 2022b). Until now there was no software tool available that solved these respective tasks for CA. The tool developed by us is freely available along its data[2], so that the reproducible and further development of both the methodology and the results is enabled for the open source community.

2. Related Work

Various approaches have been utilized to create software tools that address preprocessing challenges in different languages. These tools enable the faster analysis of substantial amounts of text data, even for historical low-resource languages and their literature.

Recent research in Arabic NLP has produced new tools that provide different functions for text, sentence, word, pre- and suffix analysis, e.g. *CAMeL* tools (Obeid et al., 2020), *Stanza* (Qi et al., 2020), *MADAMIRA* (Pasha et al., 2014) and *Farasa* (Abdelali et al., 2016). CAMeL tools is a comprehensive NLP package specialized for Arabic language. Stanza offers various preprocessing methods for many languages including Arabic. MADAMIRA and Farasa are tools for Morphological Analysis and Disambiguation of Arabic. These tools mainly focusing on Modern Standard Arabic, rather than CA. Besides, these prior works cannot convert out-of-the-box a given piece of text in XML/TEI format into a specific format (e.g. CoNLL). In our research work, this target format is needed for ML downstream-task evaluations. Even with a combination of existing tools just mentioned above, the target format cannot be reached. Therefore, we solve this challenge by developing a comprehensive modular pipeline which, once started, automatically solves the required tasks.

Preprocessing Arabic, especially Classical Arabic, is challenging due to its complex morphology. Tasks include word evaluation, categorizing sentence components, and segmenting sentences. Analyzing words and parsing sentence components for specific topics or subtopics is challenging due to contextual dependencies. Context-aware strategies are needed to prevent misinterpretations, as words can have multiple meanings depending on context, such as "madina" meaning both a city and a personal name, and "mansur" meaning assistance or a personal name.

This field is relatively new and specialized. Our literature review revealed that to the best of our knowledge, there are no prior studies available. Hence, we are among the first to deal with this genre of literature from the perspective of computational linguistics. Despite that, we discovered that existing tools are helpful for certain aspects in our pipeline, while working on preparing ML training data.

To accomplish the tasks just mentioned and to meet the specific requirements of our research, we developed in this paper our own functions and methods, which are not available in any previous work, and introduced our very specific approach for textual preprocessing of CA literature for ML applications.

3. Preprocessing Approach

As previously illustrated, our preprocessing pipeline comprises four distinct stages (see Figure 1). We provide details for each module and put our emphasis on the most complex part, namely the *TEI2CoNLL* module.

OCR process The initial step is defining and extracting CA literature data through an *OCR process* which involves in our case the digitization of the Al-Tabari corpus. This is a foundational step of transformation in our preprocessing pipeline, which allows us to take any raw OCR text from the vast collection of CA literature and digitize it so that it can be used for further processing.

GK2TEI Afterwards our *GK2TEI module* diligently transforms the digitized data of the *OCR process* available from its very specific markup language into XML files applying the TEI format, leveraging a myriad of functions we embedded within this module. This format enables the structured coding and annotation of the data. Consequently, the data is ready to be used for manual annotation and further processing by our tool *TEI2CoNLL*.

TEI2CoNLL The digitization process of the Tabari text into XML format includes the annotation of NER, topics, and subtopics by experts in the field of theological studies. The annotation of the text data is carried out manually whereby the rest of the transformations are automated by our pipeline. Afterwards, the data is exported into XML files, serving as the base for *TEI2CoNLL*, which is the core

[2] https://github.com/sa-j/ArabicNLP

of our processing pipeline. The program provides versatile filtering options for generating specific outputs, including choices for NEs, topics, subtopics, either with or without *Isnad* (chain of transmitters). Users can customize data extraction using flags, and the order of functions can be adjusted, offering flexibility in data processing. The resulting output is presented in a specific matrix format (see Figure 2).

TEI2CoNLL reads preprocessed XML files, merges them into one, and extracts NEs, topics, subtopics, and Isnads. It then converts the merged file into CoNLL format, crucial for data analysis and training ML models. In these XML files, annotated data is identified using specific keys such as ‹*persName*› for *Isnad*, ‹*name ..*› for NEs, ‹*p..ana=..*› for topics, ‹*seg ana=..*› for subtopics, and ‹*said*› for the *Matn* (text of a transmission). The XML files are processed, creating a three-column matrix shown in Figure 2. Very long sentences are split based on heuristics considering factors like length and specific factors, as described in detail in Schweter and Ahmed (2019). Users can customize the inclusion NEs and Isnad into the sentences. NE tags are converted to the BIO scheme, marking start with B-[NE] and subsequent ones with I-[NE].

```
# adyan: 1
# asbab: 0
# fiqh: 0
# kalam: 1
# lugha: 1
# mushkilat: 0
# mutashabih: 0
# naskh: 0
# qiraat: 0
# science: 1
# sirah: 0
# sufism: 1
# takhsis: 0
# tikrar: 0
# israeliyat: 1
الْجِنْ    O    B-kalam
وَهُوَ    O    O
إِبْلِيسُ    B-OTH    B-kalam
،    O    O
وَالْأَخَرُ    O    O
فِرْعَوْنُ    B-PER    O
،    O    O
قَالَ    O    O
:    O    O
أَنَا    O    B-kalam
رَبُّكُمُ    O    I-kalam
الْأَعْلَى    O    I-kalam
.    O    O
```

Figure 2: Matrix representation of NEs, topics, and subtopics

CoNLL-Reader In the final phase of the pipeline, our *CoNLL-Reader* module empowers the modification of pre-existing ML training datasets, already strictly structured in CoNLL format. This flexibility enables the application of advanced grammatical, morphological, and script-dependent preprocessing techniques, thereby enhancing the depth of analysis for historical languages and their ancient writing systems. Our *CoNLL-Reader* module has specific filtering options to modify punctuation, diacritic marks (*tashkeel*) and even letters in the CoNLL files. This allows us to customize the CoNLL files to generate different preprocessed versions of ML training data, allowing us to develop our novel method of *script-compression* as part of our ongoing work on NLP for the CA language.

Preparation of data for language model training: OpenITI extraction Beside *TEI2CoNLL*, we analogously apply a specific preprocessing technique on the OpenITI corpus in order to extract data for training the language model from scratch. To get CA text data, we crawl the platform of *OpenITI*, which contains one of the largest collections of online available historical books for CA. The final data is stored in one large text data file in which per line one sentence is saved. To actually generate this format, we apply our sentence splitting heuristics along tokenization from CAMeL tools. This additional data helps us to train a lightweight model with state-of-the-art performance for NER or related tasks without relying on pre-trained language models.

4. Results

The preprocessing pipeline *Tafsir Extractor* produces text data for different stages of our ML analysis. In the following sections, we present the major results after the *Tafsir Extractor* has been applied on the input data set consisting of the entire Tabari corpus.

GK2TEI: data for human annotation The *GK2TEI module* standardizes the raw CA text from its very specific markup language by automatically generating the TEI files. This allows the usage of various tools which are based on the popular TEI format, such as the *Oxygen XML Editor*[3]. Hence,

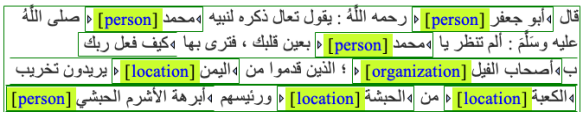


Figure 3: Screenshot of the annotation working environment in *Oxygen XML Editor* (figure taken from Ahmed et al. (2022b)).

this crucial step of conversion generates the data which enables the manual annotation of raw CA texts with NEs and Topics by experts and its further analysis by ML models. Figure 3 provides a view of the annotation environment.

[3] https://www.oxygenxml.com/

TEI2CoNLL: data for task-specific ML training (NER and TM) The output in Figure 2 presents a matrix displaying sentences with listed topics. Each sentence begins with topics marked as 1 or 0. Untagged sentences are denoted with 0, and undefined topics as 'nil.' Subtopics follow a BIO format akin to NE tags. Data extraction includes NEs, sentence-based topics, and span-based subtopics. Extracting NE tags involves boundary recognition and categorization into semantic types: persons (PER), organizations (ORG), locations (LOC), times (TME), and others (OTH), leveraging annotation data for concept analysis in theology. Sentence-based topics, in total 15, encompass a range of categories including topics like topics of juridical rulings (*fiqh*), theological topics (*kalam*), and linguistics (*lugha*). Span-based subtopics further refine these topics and include themes like specific historical topics (e.g. *tareekh*). Finally, the processed data is saved into three files (dev.conll, test.conll, train.conll).

Table 1 provides the results for NEs. We can see that there are twice as many NE tokens in the data with Isnad compared to the data without Isnad. Especially for the NE category PER, the amount is increased significantly, but not for the other four NE categories. This is not surprising, since by definition an Isnad consists predominantly of transmitters (i.e. PER). Therefore, the inclusion of Isnad has a greater impact on the number of PER tokens, rather than any other NE category and their tokens. Furthermore, according to our results presented in Table 1, the total count of tokens is 1,793,315. However, when Isnad is excluded from the calculation, the count drops to 913,749 tokens. This suggests that approximately half of the text consists of Isnad data.

NE w. Isnad	NE w.o. Isnad
1,409,334 O	775,010 O
176,105 B-PER	47,746 B-PER
149,292 I-PER	31,991 I-PER
22,026 B-ORG	21,459 B-ORG
12,453 B-OTH	12,142 B-OTH
8,456 I-ORG	8,122 I-ORG
4,160 B-TME	6,610 B-TME
5,583 B-LOC	4,990 B-LOC
4,087 I-TME	3,912 I-TME
1,032 I-OTH	1,008 I-OTH
787 I-LOC	759 I-LOC

Table 1: Results for NE tokens with/without Isnad

For topics, the picture is more dynamic while looking from the perspective of Isnad inclusion (see Table 2). For some topics (fiqh, sufism, adyan) the number depends highly on the Isnad inclusion, whereas for some other topics (qiraat, tikrar, takhsis) the number seems to be not strongly influenced by this inclusion. Further investigation is required to determine the reason for this pattern.

Topic	w. Isnad	w.o. Isnad
adyan (non-Islamic relig.)	31,931	20,536
asbab (occas. of revelation)	9,143	6,268
fiqh (jurisprudence)	21,381	9,753
israiliyat (Judeo-Christian)	7,795	4,533
kalam (Islamic theology)	36,133	19,384
lugha (linguistics)	29,573	15,776
mushkilat (problem)	59	28
mutashabih (allegorical)	360	175
naskh (abrogation)	1,257	727
qiraat (recitation style)	4,957	4,238
sirah (prophetic biography)	3,960	2,729
sufism (mysticism)	15,570	7,553
takhsis (specification)	400	317
tikrar (repetition)	405	381
ulum (science)	5,028	2,262

Table 2: Results for Topic tokens with/without Isnad

OpenITI: data for language model training (task-independent) Our results for the OpenITI corpus data are 134.17 Mio. sentences, extracted from 17 GB of raw text data, which is the largest amount yet to be used for CA. Thus, this allows the training of lightweight ML models for CA-NER and CA-TM without relying on pre-trained language models which are not made with regard to the domain of historical theology. We plan to upload this corpus data along its text generation module for the research community. This will give rise to the possibility of using the strengths of current heavyweight ML models (such as BERT, XLNet, GPT-3 (Devlin et al., 2019; Yang et al., 2019; Brown et al., 2020)) and training domain-specific versions of them as well, even when new historical text collections are added to the growing platform of OpenITI.

5. Conclusion

In this paper, we introduced the *Tafsir Extractor*, a comprehensive preprocessing tool designed for extracting raw text data from CA literature and converting it into a specific format (e.g. CoNLL and its extensions), to facilitate downstream-task evaluations for fundamental NLP tasks, such as NER and TM. The absence of a similar tools for CA literature prior to our work prompted the development of *Tafsir Extractor*. Consequently, our work paves the way for a large-scaled generation and analysis of historical CA literature with modern ML methods.

Our work highlights the challenge of sentence segmentation and word recognition in CA texts due to the absence of punctuation and the context-dependant changes in the semantics of words. To overcome these challenges, we have employed a specialized heuristics in our program, which considers word counts, customizable through a filter in our program, and takes into account sub/topic annotatations for segmentation. Determining contextual meanings of words still poses a formidable challenge for NLP methods in prospective projects. Despite minor deviations, the cleanliness of the data

enables its utilization for subsequent downstream-task evaluations without hindrance.

In future work, we propose improving sentence segmentation by developing a domain-specific neural network model which identifies sentence boundaries based on semantics rather than syntax of text. This approach holds promise for addressing the major limitations encountered in our current work.

Acknowledgments

This interdisciplinary work has been conducted as part of the research on NLP for Classical Arabic literature[4]. This work was supported by a fellowship of the German Academic Exchange Service (DAAD). Special thanks go to Prof. M. Syed (University of California, Davis) and Prof. G. Roig (Goethe University Frankfurt) for the resources made available for conducting the research presented in this paper.

6. Bibliographical 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.

Sajawel Ahmed, Misbahur Rehman, Joshua Tischlik, Carl Kruse, Edin Mahmutovic, and Ömer Özsoy. 2022a. Linked Open Tafsir—Rekonstruktion der Entstehungsdynamik (en) des Korans mithilfe der Netzwerkmodellierung früher islamischer Überlieferungen. In *8. Jahrestagung des Verbandes Digital Humanities im deutschsprachigen Raum (DHd)*.

Sajawel Ahmed, Rob van der Goot, Misbahur Rehman, Carl Kruse, Ömer Özsoy, Alexander Mehler, and Gemma Roig. 2022b. Tafsir dataset: A novel multi-task benchmark for named entity recognition and topic modeling in classical Arabic literature. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3753–3768, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Jalal Al-Din Al-Suyuti. 1505. *Al-itqan Fi 'ulum Al-Qur'an (The Perfect Guide to the Sciences of the Qu'ran)*. Garnet Publishing; Bilingual edition (May 1, 2012).

[4] https://tafsirtabari.com/about

Emily M Bender and Batya Friedman. 2018. Data statements for natural language processing: Toward mitigating system bias and enabling better science. *Transactions of the Association for Computational Linguistics*, 6:587–604.

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, NA. 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, Minnesota. 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. Association for Computational Linguistics.

Wang Ling, Chris Dyer, Alan Black, and Isabel Trancoso. 2015. Two/Too Simple Adaptations of word2vec for Syntax Problems. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Matthew Thomas Miller, Maxim G. Romanov, and Sarah Bowen Savant. 2018. Digitizing the textual heritage of the premodern islamicate world: Principles and plans. *International Journal of Middle East Studies*, 50(1):103–109.

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.

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).

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Stefan Schweter and Sajawel Ahmed. 2019. Deep-EOS: General-Purpose Neural Networks for Sentence Boundary Detection. In *Proceedings of the 15th Conference on Natural Language Processing (KONVENS)*.

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.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Rob van der Goot, Ahmet Üstün, Alan Ramponi, Ibrahim Sharaf, and Barbara Plank. 2021. Massive choice, ample tasks (MaChAmp): A toolkit for multi-task learning in NLP. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 176–197, Online. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32, NA. Curran Associates, Inc.

A. Data and Code Availability Statement

The project aims to promote cooperation and progress in the field of NLP. To ensure transparency and reproducibility, all datasets used in our experiments, along with the corresponding codebase, will be made readily available to the public through GitHub repositories (https://github.com/sa-j/ArabicNLP). The datasets will be provided in commonly used formats, accompanied by comprehensive documentation detailing their sources, preprocessing procedures, and any relevant licensing information. The codebase will be structured in a modular and well-documented manner. The aim is to offer researchers precise instructions for accessing and using the data, which will facilitate their understanding, extension, and adaptation of our algorithms and methodologies. The NLP community is encouraged to explore, critique, and build upon the contributions, promoting a culture of open collaboration and accelerating progress in the field.

Advancing the Arabic WordNet: Elevating Content Quality

Abed Alhakim Freihat [1], **Hadi Khalilia** [1,2,*], **Gábor Bella** [3], **Fausto Giunchiglia** [1]
[1]Department of Information Engineering and Computer Science, University of Trento, Italy
[2]Palestine Technical University – Kadoorie, Palestine
[3]Lab-STICC CNRS UMR 628, IMT Atlantique, Brest, France
[1] {abed.freihat, hadi.khalilia, fausto.giunchiglia}@unitn.it,
[2]h.khalilia@ptuk.edu.ps
[3]gabor.bella@imt-atlantique.fr

Abstract

High-quality WordNets are crucial for achieving high-quality results in NLP applications that rely on such resources. However, the wordnets of most languages suffer from serious issues of correctness and completeness with respect to the words and word meanings they define, such as incorrect lemmas, missing glosses and example sentences, or an inadequate, Western-centric representation of the morphology and the semantics of the language. Previous efforts have largely focused on increasing lexical coverage while ignoring other qualitative aspects. In this paper, we focus on the Arabic language and introduce a major revision of the Arabic WordNet that addresses multiple dimensions of lexico-semantic resource quality. As a result, we updated more than 58% of the synsets of the existing Arabic WordNet by adding missing information and correcting errors. In order to address issues of language diversity and untranslatability, we also extended the wordnet structure by new elements: *phrasets* and *lexical gaps*.

Keywords: Arabic, wordnet, quality, completeness, correctness, phraset, lexical semantics

1. Introduction

WordNets (Beckwith et al., 2021) are lexical databases that represent lemmas (lexemes, words) of a language, together with their meanings organised into a lexico-semantic network. Wordnets define meanings as sets of synonymous words called *synsets*. Synsets are described by a gloss (e.g., a definition in a natural language that represents the synset meaning) as well as example sentences that clarify the usage of words in context. WordNets are used in many NLP applications, such as machine translation (Poibeau, 2017), information retrieval (Nie, 2022), or word sense disambiguation (Navigli, 2009).

The English Princeton WordNet (PWN) (Miller, 1995), as the first wordnet, has been adapted and employed as a foundation for constructing wordnets in other languages.

In general, WordNets are constructed using either the *merge* or the *expand model* (Vossen, 1998). In the merge model, synsets are initially created from pre-existing resources (e.g., dictionaries) in a language. Then, for translability into other languages, the synsets have to be aligned with equivalent English synsets in PWN. For example, the IndoWordNet (Bhattacharyya, 2010) was built following this model. In the expand model, PWN synsets are 'localized' or 'translated' into target languages. For example, the Polish WordNet (Piasecki et al., 2009) was constructed using this model. In either case, when mapping across languages, the PWN synsets (and thus the English language) are usually used as a pivot when translating words across languages.

Wordnets often suffer from quality issues, in a large part due to the use of automated and semi-automated methods for building them (Khalilia et al., 2021a,b). In addition, mistakes can be hard to detect as most wordnets do not contain glosses or example sentences. The above are true of the existing Arabic wordnets. The first Arabic wordnet (AWN V1) was built following the expand model (Elkateb et al., 2006) and includes 9,618 synsets translated from PWN to modern standard Arabic. Its second version (AWN V2) (Regragui et al., 2016) extended AWN V1 to 11,269 synsets and was developed using a semi-automatic method and the expand model. As we show in our paper, both wordnets suffer from correctness and completeness issues, and lack glosses and examples. By correctness we refer to the accuracy of lemmas in representing the meaning of a synset, while completeness refers to the extent to which a synset includes all words that are synonymous based on the synset meaning. For example, without an Arabic gloss and example sentences, it is hard to judge the correctness and completeness of the AWN V1 synset {تحريك، تسيير، دفع، دسر} that corresponds to the English WordNet synset {*actuation, propulsion: the act of propelling; actuation of this app needs a password*}.

In this paper, we introduce AWN V3, a significantly extended and quality-enhanced version of AWN V1. The novel contents of this new Ara-

OSACT 2024 Workshop, pages 74–83

bic wordnet are: (a) the addition of glosses and examples to all synsets; (b) the improvement of the correctness and the completeness of the wordnet by adding missing lemmas and removing erroneous ones; (c) a reduced level of polysemy with respect to other wordnets through the elimination of redundant word meanings, based on our prior research; and (d) addressing phenomena of language diversity by introducing new linguistic information, namely *lexical gaps* that explicitly indicate untranslatability (Giunchiglia et al., 2018; Bella et al., 2022) and *phrasets*, i.e., free combinations of words that express the meaning of a synset in case of nonexistent equivalent lemmas (Bentivogli and Pianta, 2000). Such explicit representations of untranslatability distinguishes them from resource incompleteness (i.e., words merely missing from the resource) and give indications to both human and machine translators about particularly difficult cases of translation. Also, we tackle the polysemy problem of the source synsets by not inheriting specialization polysemy (Freihat et al., 2013) and compound noun polysemy (Freihat et al., 2015) problem in the target synsets.

Accordingly, the paper presents the following contributions: (1) the extension of the existing Arabic wordnet model by devices for tackling untranslatability: lexical gaps and phrasets; (2) a development methodology for lexical databases, inscribed within the expand model, that ensures a high-quality and diversity-aware output; (3) AWN V3, the new and freely available Arabic wordnet resource as described above.

The rest of the paper is organized as follows. In Section 2, we introduce the state of the art of Arabic wordnets. Sections 3 and 4 present our contributions in addressing language diversity and excessive polysemy, respectively. In Section 5, we describe our synset localization method. Section 6 presents AWN V3, the high-quality Arabic lexical resource resulting from our work. Finally, we provide conclusions and discuss future work in Section 7.

2. State of the Art

The first effort of building an Arabic wordnet was undertaken by Diab (2004). She introduced an automated approach known as SALAAM (Sense Assignment Leveraging Annotations And Multilinguality) to translate synsets from PWN into standard Arabic. This translation process relied on PWN 1.7 and an English-Arabic corpus as knowledge sources. Notably, her primary focus was on translating lemmas without glosses and example sentences. This approach was evaluated using a dataset comprising 447 synsets.

AWN V1 represents the inaugural Arabic WordNet developed by Elkateb et al. (2006). The development approach closely mirrors the methodology employed in creating EuroWordnet (Vossen, 1999), which consists of two phases. The first phase involves constructing a foundational core wordnet centered around *base concepts* (Vossen, 1998), while the second phase focuses on expanding the core wordnet's coverage by incorporating additional criteria. This version of AWN is aligned with PWN in terms of structure and content covering WordNet domains defined by Magnini and Cavaglia (2000). This wordnet also integrates the Suggested Upper Merged Ontology (SUMO) to provide a formal semantic framework (Elkateb et al., 2006).

In the case of the core Arabic WordNet, the process involves encoding the Common Base Concepts (CBCs) found in the EuroWordnet and BalkaNet (Tufis et al., 2004) as synsets. This is achieved through a manual translation effort, wherein all English synsets having an equivalence relation in the SUMO ontology are translated into their corresponding Arabic synsets. Figure 1 illustrates this process, showing an example of how Arabic Synsets are linked to overarching SUMO terms that directly correspond to the associated English synsets. Each translated synset is validated by evaluating the coverage of synset lemmas and the domain distribution of these synsets. These efforts produced 9,228 synsets in the core wordnet of AWN V1. The distribution of these synsets, categorized by Part-Of-Speech (POS), is detailed in Table 1.

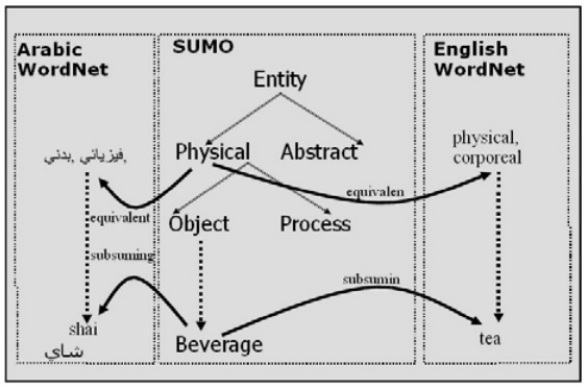


Figure 1: SUMO mapping to WordNets (Elkateb et al., 2006)

To expand the core of AWN, Elkateb et al. (2006) introduced the Suggested Translation semi-automatic method, using available bilingual (Arabic-English) resources to extract *<English word, Arabic word, POS>* tuples. This method served a similar purpose in the development of Spanish WordNet (Farreres et al., 2002) and BalkaNet (Tufis et al., 2004). Building on eight

POS/WN	AWN V1 (Core WN)	AWN V1 (Ext. WN)	AWN V2
Noun	6,252	6,558	7,960
Verb	2,260	2,507	2,538
Adjective	606	446	271
Adverb	106	107	500
Total	9,228	9,618	11,269

Table 1: The count of Arabic synsets in each AWN version based on POS

heuristic procedures, associations between Arabic words and PWN synsets were assigned scores, relying on Arabic-English bilingual resources. Lexicographers utilized these scores to create new synsets or supplement existing ones with additional lemmas. The total number of synsets in this version is 9,618.

After the first release of AWN, there were many attempts to enrich its content concerning the number of synsets, lemma, and the relations between them. Alkhalifa and Rodríguez (2009) introduced an automated method to enhance the coverage of named entities (NE) within AWN V1. This method used Wikipedia and established connections to PWN 2.0. In this study, 1,147 synsets were generated, covering 1,659 named entities across 31 general categories. In these studies, Boudabous et al. (2013); Batita and Zrigui (2018) proposed a hybrid linguistic approach grounded in morphological patterns. They used Wikipedia and PWN to enrich AWN with new semantic relations. The former augmented AWN by establishing relations between nominal synsets, while the latter incorporated antonym relations.

As part of the ongoing efforts to enrich AWN, Abouenour et al. (2013) introduced a semi-automatic method to increase the coverage of AWN V1. Their objective was to enhance named entities (NEs), verbs, and noun synsets. For the enrichment of NE synsets, the authors present a three-step methodology, which translates YAGO (Yet Another Great Ontology) entities (Suchanek et al., 2008) into Arabic instances and extracts Arabic synsets. Regarding verb synsets, the authors adopted a two-step approach inspired by Rodríguez et al. (2008). The first step involved suggesting new verbs by translating a set of verbs from VerbNet (Schuler, 2005) into standard Arabic. In the second step, Arabic verbs were interconnected with AWN synsets by establishing a graph connecting each Arabic verb with its corresponding English verbs in PWN. The authors employ a two-step method that detects hyponym/hypernym pairs from the web to enrich noun synsets. The overall result of this work is introducing a new version of AWN, known as AWN V2, including 11,269 synsets (for more details, see Table 1).

Despite the previous efforts, which primarily focused on expanding the coverage of synset lemmas, AWN still falls short compared to other WordNets in terms of content quality. This assessment was highlighted by Batita and Zrigui (2018), who emphasized in their research that "AWN has very poor content in both quantity and quality levels." Our work focuses on the synset quality level, mainly on the synset correctness and completeness dimensions. AWN V1 marks a significant milestone for several reasons. Firstly, it encompasses the most common concepts and word senses found in PWN 2.0, ensuring a comprehensive representation in AWN V1. Secondly, its design and integration with PWN synsets facilitate cross-language usability. Finally, like other wordnets, AWN establishes a connection with SUMO, further enhancing its utility. Conversely, several issues related to synset quality have been identified in the majority of the synsets in this resource. These issues are also observed in AWN V2, as outlined below:

1. All synsets lack gloss and/or illustrative examples.

2. Many synsets contain incorrect senses, lemmas (including incorrect word forms or repeated words), and incorrect relations between synsets.

3. Many synsets lack essential senses, lemmas, and necessary relations.

For instance, consider the following synset {ضجيج، ضوضاء} presented in AWN V1, corresponding to the English synset {*noise: sound of any kind, especially unintelligible or dissonant sound; he enjoyed the street noises*}. In AWN V2, this synset was enriched to include {ضجج، ضوض}, resulting in {ضجيج، ضوضاء، ضجج، ضوض}. In this case, the synset incorporates two erroneous lemmas {ضجج، ضوض }, which are not found in Arabic dictionaries such as المعاني Almaany dictionary[1]. Additionally, it lacks the lemma ضجة, which means *noise*.

In this paper, we enhance the accuracy of synset elements in AWN V1 by addressing incorrect lemmas and expanding the coverage of synsets through the addition of missing lemmas.

3. Addressing Language Diversity

Cultural and linguistic differences abound across the more than seven thousand languages in the

[1] http://www.almaany.com/thesaurus.php

world, to which we simply refer as language diversity. To give a few examples from lexical semantics, the English word cousin, meaning *the child of your aunt or uncle*, does not have any equivalent term in Arabic. In contrast, the Arabic word عم, which means *the brother of your father*, does not exist in English (Khalilia et al., 2023). Another example is from colors: the Italian word *marrone*, which means *chestnut color*, does not have an equivalent word in Persian and Welsh (McCarthy et al., 2019), while the Breton *glaz*, spanning a range of hues between blue and green, has no equivalent in English or in the majority of Indo-European languages.

Linguists refer to such cases of lexical untranslatability as *lexical gaps*. A lexical gap happens when a word in one language is not lexicalized in another language (Lehrer, 1970). In such cases, speakers can express a similar meaning through a free combination of words called *phrasets* (Bentivogli and Pianta, 2000).

As in most wordnets, instances of language diversity are not explicitly indicated in the existing versions of AWN, instead mapping Arabic synsets to PWN synsets in an approximate manner. Such inaccuracies lead to the corruption of resource quality and an Anglo-Saxon meaning bias, also reducing the performance of applications relying on the resource, such as translation tools.

This paper introduces a new version of AWN that explicitly represents gaps and phrasets. For example, *{adjectively: as an adjective; nouns are frequently used adjectively}* is identified a gap in the resulting resource; at the same time, this phraset على شكل صفة is used to describe this synset. In addition, in the case of lexicalizations (translated synsets), to increase the clarity of synset meaning and understandability, phrasets are used. For example, *{unwittingly, unknowingly, inadvertently: without knowledge or intention; he unwittingly deleted the references }* is translated { سهواً: على نحو غير مقصود ودون إدراك أو معرفة ، حذفت الملف بدون قصد }, and the phraset بدون قصد is used.

Lexical gaps are implemented in our resource at the synset level, while phrasets are implemented on the word level.

4. Addressing Polysemy

Polysemy is a well-known problem in PWN. It has been addressed in many studies, such as (Gonzalo, 2004; Mihalcea and Moldovan, 2001; Buitelaar, 1998; Freihat, 2014). In our previous research (Freihat et al., 2016), polysemy was classified into several types. These types are homonomy, metaphoric, metonymy, specialization polysemy, and compound noun polysemy. While the first three polysemy types are essential in lexical resources, the latter two are considered the main reasons behind the highly polysemous nature of WordNet that makes WordNet too fine-grained for NLP. As an example of compound noun polysemy, the word *head* has more than 30 synsets (meanings) in PWN. Another example of compound noun polysemy is the word *center*, which has 18 synsets. The problem becomes more clear in the Arabic ontology (Jarrar, 2021), which has more than 500 synsets مركز meaning *center*. For example, the word *turtledove* is polysemous because it belongs to the following two synsets: *{australian turtledove, turtledove: small Australian dove.}*, *{turtledove: any of several Old World wild doves.}* Of course, it is possible to use the word *turtledove* to refer to any kind of turtledoves when it is clear from the context which kind of turtledoves we are speaking about. At the same time, adding the word *turtledove* as a synonym to all kinds of turtledoves in the lexical resource is useless and just makes the resource hard to use.

According to our research (Freihat et al., 2015), the word sense disambiguation for these two types is similar to anaphora resolution and does not require including all these possible meanings in a lexical resource because they lead to the problem of sense enumeration which makes such resources very hard to use in NLP.

5. Addressing Synset Quality

In the following, we list the goals of our approach:

1. Synset glosses: Each synset should have a gloss that clearly identifies its meaning. Without such gloss, we will not be able to understand the synset, moreover, we will not be able to differentiate between the meanings of the same lemma in different synsets, for example, the word *'love'* has more than one meaning e.g, belongs to different synsets

2. Synset examples: Each lemma in a synset should have at least one example to clarify its usage. Such examples also allow us to verify the synonymity between the synset lemmas. This is crucial for the synset correctness.

3. Language diversity and phrasets: Ideas are expressed in cultures in different ways, which leads to untranslatability in some languages (e.g., a lexical gap). Another phenomenon in Arabic (and maybe in other languages) is the usage of prepositional phrasets to express a synset meaning. For example, the meaning of this synset { *someday: some unspecified time in the future; someday you will understand my actions*} is identified as a lexical gap in Arabic, and the phraset يوماً ما is used to express this

meaning. We add these phrasets to the Arabic WordNet to increase the understandability of synsets. Also, such phrasets can be used in NLP applications to identify the intended synset.

4. Errors in the source WordNet (PWN): PWN suffers from the polysemy problem. According to our previous approaches, the source of the polysemy problem is due to the specialization polysemy and sense enumeration. In our work, we avoid such polysemy types in the resulting resource to enhance AWN usability in NLP applications.

5. Named entities: A lexical resource should include concepts only. It is not the correct place to include named entities, which may be another source of noise in lexical semantic resources.

Our approach consists of three steps:

1. Task generation: We have collected the data from AWN V1 and prepared the spreadsheet to be provided for translation.

2. Task enhancement: The translators translated the corresponding PWN synset glosses, then performed the following: adding missing lemmas, and examples for the synset elmmas, removing wrong lemmas from the original Arabic synsets, identifying gaps in the case of untranlatability, and adding phrasets for increasing the understandability of synsets.

3. Task validation: Validation is carried out in two phases: 1) Each contribution provided by one of the translators was validated by the other. In the second phase, a linguistic expert validates and approves the contribution.

5.1. Task generation

This section describes the essential materials required for the next step of the methodology. The preparation process involves constructing a dataset containing AWN V1 synsets as well as the corresponding PWN synsets. In this context, AWN V1 and PWN browsers are utilized for data retrieval. This dataset is customized in a spreadsheet for usability and simplicity in providing contributions, in which the linguistic expert (the first author) organizes synsets into four categories (each in one sheet) based on the part of speech (POS). Each row within the spreadsheet represents a synset and includes information such as the synset ID, lemmas, gloss, and example sentences in Standard Arabic and English. Additionally, empty slots are provided for inserting missing lemmas, a gloss, examples, and comments by the data provider (translator) in Arabic. One additional slot is designated for validation purposes, along with comments from the validator. In this step, the linguistic expert excludes all (42 synsets) named entities from the spreadsheet.

5.2. Task enhancement

Contributions for synset enhancement, which involve the addition of missing information or correction of synset elements, are made by two translators and validated by a language expert. An overview of our contribution collection workflow is illustrated in Figure 2. As depicted, the workflow is structured into two cycles, with the aim of ensuring the quality of results. The first cycle operates between the two translators, where each translator's contributions are subject to verification by the other. The second cycle involves the validation of accepted contributions by a linguistic expert.

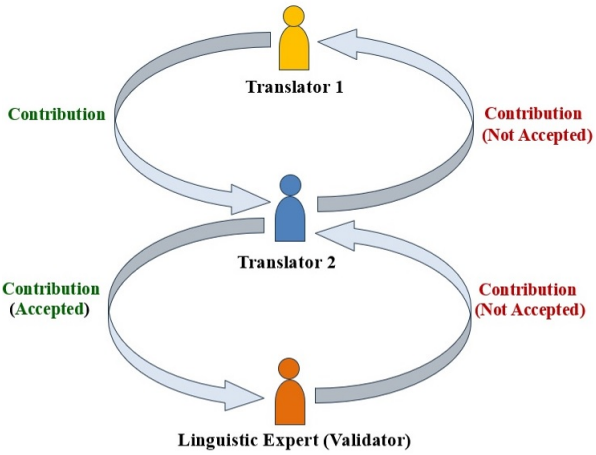


Figure 2: The workflow of the contribution collection

The process of synset enhancement in the first cycle was carried out by two native speakers. Regarding their socio-linguistic background, both translators possess at least a bachelor's degree in the field of translation (English-Arabic). Before the translation, translators have been trained as described in the following subsections.

5.2.1. Synset understanding

Central to this process is ensuring that the translator possesses a clear understanding of the synset they are tasked with translating. Misunderstandings can arise when the translator does not grasp a thorough understanding of both the synset lemmas and the gloss in English. The translators are asked to understand each PWN synset in the

spreadsheet using the following notable instructions:

- Use external resources such as dictionaries and Wikipedia to understand the meaning of the synset.
- They are given the authority to skip the synset or leave a comment when they do not understand the meaning of the synset.

5.2.2. Lexical gap identification and synset lexicalisation

A lexical gap happens when either the meaning of the concept in a source language is not known in the culture of the target language or the concept can be lexicalized only through word-free combination (Giunchiglia et al., 2018). This means that there is no lexical unit (single word or restricted collocation) that corresponds to any of the source language lemmas. In this step, for each English synset in the spreadsheet, the translator decides whether it has an equivalent meaning in Arabic (lexicalisation exists) or is a lexical gap based on the understanding of the English synset and using a bilingual dictionary. If an English synset$_i$ is a gap, the translator performs **step A**; otherwise, she/he performs **steps B** and **C**.

Step A: Lexical gap processing, in this step, the translator is asked to mark the English synset$_i$ as a lexical gap in the spreadsheet and provide a phraset in Arabic. For example, the synset {*expressively: with expression, in an expressive manner; she gave the order to the waiter, using her hands very expressively*} is identified as a lexical gap in Arabic, and بشكل معبر meaning (an expressive way) is provided as a phraset to this synset.

Step B: Synset translation, after the translator confirms the existence of the meaning of the English synset$_i$ in Arabic, she/he translates this synset to Arabic. This translation includes the following steps.

1. **Translating synset gloss:** The translation is across language and cross-cultural communication. A translation should give a complete transcript of the synset; meanwhile, the style and manner of writing should be at least the same quality as the gloss of English. Above all, faithfulness, expressiveness, and closeness are the important three elements of translation. The gloss should explicitly express the semantics and the common attributes of a synset.
2. **Translating synset lemmas:** Translators should keep two key considerations in mind while translating synset lemmas. Firstly, this translation process does not entail a direct one-to-one correspondence between English and Arabic terms. Secondly, it is important to note that the set of lemmas within the English synset may not be exhaustive, meaning it might not contain all the synonyms associated with the synset. To translate the synset lemmas, we go through the following phases:
   - **English lemmas translation:** Translate the English synset lemmas into Arabic. The result of this step is a set of lemmas of the length n, where n is the number of lemmas in the English synset.
   - **Arabic synonyms collection:** For each translated lemma, the translators collect the lemma synonyms in Arabic. The result of this phase is m synonym sets in Arabic, $m \leq n$ (since some Arabic lemmas may have empty synonym sets).
   - **Arabic synonyms validation:** Based on the synset gloss, for each of the m synonym sets in Arabic, the translators exclude all synonyms that do not belong to the synset. Use the provided examples in the English synset and other examples to include/exclude the synonyms in this phase.
   - **Arabic lemmas collection:** The translators collect the Arabic lemmas, resulting in the translation process in phase (1) and the synonyms produced from phase (2) and put them as the Arabic synset lemmas. In the case of polysemy, we solve the specialization polysemy and compound noun polysemy. For example, جِسْم is excluded from this synset {جِسْم فِيزْيائِي، جِسْم} which corresponds {*object, physical object: a tangible and visible entity, an entity that can cast a shadow; pens, books and bags are school objects*}
   - **Arabic lemmas ordering:** The translators order the Arabic collected lemmas in phase (4), wherein the first lemma is the Arabic synset preferred term and so on (in descending order of importance). Based on the examples provided in phase (3) (and other examples if needed), the translator gives preferences for the lemmas based on these examples.
3. **Translating synset examples:** Examples within a synset contribute to a clearer comprehension of how to utilize the synset lem-

mas, consequently enhancing the overall understanding of the fully lexicalized synset. We employ the same examples crafted during the lemma translation phase as synset examples. This approach signifies that we do not solely translate the examples found in the English synset. Ideally, we provide an example sentence in Arabic for each synonym within the synset, even if the English synset does not contain examples at all. The provided examples are incorporated into the Aarbic synsets, aligning them with the order of their respective synonyms

Step C: Comparing the produced (translated) synset in Step (B) with the corresponding synset from AWN V1, At this stage, a translator compares the translated synset generated in Step B and its corresponding Arabic synset, as imported from AWN V1. This Arabic synset is designated to correspond with the English synset$_i$ in the spreadsheet. Based on the gloss and examples provided in the generated synset, the translators undertake the following actions: (1) Copy lemmas from the translated synset to the AWN V1 synset if they are missing from the AWN V1 synset. (2) Exclude the lemmas from the AWN V1 synset, which are not covered by the synset gloss and examples. (3) Copy the gloss and the examples from the translated synset to the AWN v1 synset if they are missing in the latter.

5.3. Task validation

The validation process consists of two phases. In the first phase, the two translators validate the resulting synsets (stored in a spreadsheet containing English and produced Arabic synsets) in an alternating manner, checking each synset (and gap) one by one. During the validation, each of them considers the following:

1. Gap validation: A translator validates synsets marked as lexical gaps in Arabic, either as confirmed gaps or as non-gaps due to an existing lexicalization in Arabic, which he/she should provide a gloss and lemmas of that synset.

2. Gloss validation: The Arabic gloss expresses the intended meaning of the English synset. Also, the Arabic gloss is easy to understand and does not contain typos or grammatical errors.

3. Lemmas validation: Synset lemmas should be correct (e.g., not include wrong lemmas) and complete (e.g., there are no missing lemmas). In addition, the validator can use the examples to check synonymity between lemmas.

4. Examples validation: Each lemma has at least one example. The examples are natural and express the intended usage.

In case of disagreement, the affected synsets are sent back to the translators with the validator's comment. The accepted synsets are sent to the expert validation.

In the second phase, An Arabic linguistic expert performs this validation on a spreadsheet containing the resulting synsets (and gaps) only, without including the English synsets, which both translators accepted in the previous step. His task is to approve the final resulting synsets. The same criteria used in the previous validation phase for validating gaps, glosses, lemmas, and examples are adopted in this step.

6. Evaluation and the Resulting Resource

This section demonstrates the use of the methodology described in Section 5 on evaluating and improving the content quality of AWN V1 depending on PWN as a reference to our work. As mentioned above, AWN V1 includes 9,618 synsets written in Modern Standard Arabic (MSA), which refers to the standard form of the language used in academic writing, formal communication, classical poetry, and religious sermons.

In this study, contributions are provided by two translators (each is an Arabic native speaker). They were born and educated within the Arabic-speaking community, having completed at least their high school education within this community.

Four experiments (one for each POS) are performed to evaluate the extended version of AWN V1 synsets and tackle synset quality issues using our method. In each experiment, a spreadsheet includes Arabic synsets imported from the AWN V1 and their corresponding English synsets. Each spreadsheet contains data for a specific POS and serves as an input dataset to the contribution (synset quality enhancement) collection step. The experiments are conducted on 6,516 nouns, 2,507 verbs, 446 adjectives, and 107 adverbs (see Table 2 for more details).

In the contribution collection, for each Arabic synset in a row in the spreadsheet, a translator is tasked to translate the corresponding PWN synset to Arabic or identify it as a lexical gap using a bilingual (English-Arabic) linguistic resource, such as the Al-Mawrid Al-Qareeb المورد القريب dictionary (Baalbaki, 2005). After that, if a lexicalization exists in Arabic, the translator tackles the latter by comparing a generated translated Arabic synset

	Noun	Verb	Adjective	Adverb	Total
Synsets	6,516	2,507	446	107	**9,576**
Words	13,659	5,878	761	262	**20,560**

Table 2: The count of synsets and words (imported from the extended AWN V1- without named entities) in the input dataset based on POS

with the AWN V1 synset in the same row, which follows by adding missing synset lemmas, gloss, and example sentences; and/or rectifying incorrect elements. Also, if the English synset is a gap in Arabic, he/she marks it as a lexical gap and provides a phraset to express the synset (Note that phraset is also used for some translated synsets to increase the understandability). To our knowledge, our resulting resource (AWN V3) is the first Arabic Wordnet that identifies gaps and provides phrasets.

The overall effort to collect contributions resulted in updating 5,554 synsets from AWN V1. We added 2,726 new lemmas, 9,322 new glosses, and 12,204 new example sentences. We also identified 236 lexical gaps and inserted 701 phrasets. Furthermore, we deleted 8751 incorrect lemmas. More details regarding the counts of these contributions are presented in Table 3. See the dataset uploaded to GitHub[2]. For each POS, two spreadsheets were uploaded to GitHub; the first file includes the final resulting Arabic synsets, and the second contains the added and deleted synset components.

	Noun	Verb	Adj	Adv	Total
Updated synsets	3,938	1,364	181	71	**5,554**
New lemmas	2,581	64	72	9	**2,726**
Deleted lemmas	6,050	2,387	223	91	**8,751**
New glosses	6,511	2,258	446	107	**9,322**
New examples	7,597	3,620	782	205	**12,204**
Gaps	28	187	0	21	**236**
Phrasets	364	275	0	62	**701**

Table 3: Statistics of the data addition and deletion into/from AWN

Validation was carried out by an Arabic linguistic expert who has a Ph.D. in the Arabic language and is a university instructor at the linguistics department. As introduced above, the expert follows the criteria described in Section 5.3 to verify produced synsets. Results can be seen in Table 4, where by correctness we understand the number of contributions validated as correct divided by the total number of contributions. These contributions can be newly added or deleted lemmas, collected glosses and example sentences, identified lexical gaps, and inserted phrasets. For example, in the case of an added lemma, the validator either confirms the addition or rejects it by leaving a comment. For instance, {مِقْيَاس} meaning *a measuring tool* is deemed an incorrect added word to the synset {مِقْدَار، قَدْر، كَمّ، كَمِّيَّة } which corresponds to *{measure, quantity, amount: how much there is of something that you can quantify; he has a big amount of money}*. In the case of identified gaps, the validator either as confirmed gaps or as non-gaps due to an existing lexicalization in Arabic, which the validator needs to indicate. For instance, the following English synset *{try, try on: put on a garment in order to see whether it fits and looks nice; Try on this sweater to see how it looks}* is considered a gap. The validator rejected it and provided this word قَاسَ with the same meaning.

Contribution	Correctness
New lemmas	97.34%
Deleted lemmas	98.89%
New glosses	98.76%
New examples	99.13%
Gaps	96.82%
Phrasets	97.54%
Total	98.08%

Table 4: Validator evaluation of translator contributions

Upon discussion between the validator (linguistic expert) and the translators, the mistakes made by the latter can be explained by misunderstandings of the meanings of certain concepts provided in English. The validator made sure to exclude or fix the mistakes, bringing the correctness of the final dataset closer to 100%.

7. Conclusion and Future Work

In this paper, we evaluate and address the quality—correctness and completeness—of synsets from AWN V1 across four parts of speech (nouns, verbs, adjectives, and adverbs). The resulting total of 9,576 synsets are introduced as AWN V3—an enhanced version of AWN with corrected and extended lemmas, as well as added glosses and example sentences. In order to represent English words not directly translatable to Arabic, we introduce *phrasets* to provide approximate phrase-level translations and *lexical gaps* to indicate untranslatability. As part of our future work, we will apply the methodology

[2] https://github.com/HadiPTUK/AWN3.0

described in order to increase the coverage of Arabic synsets, based on AWN V2 as well as the remaining synsets in PWN.

8. Bibliographical References

Lahsen Abouenour, Karim Bouzoubaa, and Paolo Rosso. 2013. On the evaluation and improvement of Arabic Wordnet coverage and usability. *Language resources and evaluation*, 47:891–917.

Musa Alkhalifa and Horacio Rodríguez. 2009. Automatically extending NE coverage of Arabic Wordnet using Wikipedia. In *Proc. Of the 3rd International Conference on Arabic Language Processing CITALA2009, Rabat, Morocco*, pages 23–30.

Rohi Baalbaki. 2005. *Al-mawrid Al-qareeb Arabic-English Dictionary*. Dar El Ilm Lilmalayin, lebanon.

Mohamed Ali Batita and Mounir Zrigui. 2018. The enrichment of Arabic Wordnet antonym relations. In *Computational Linguistics and Intelligent Text Processing: 18th International Conference, CICLing 2017, Budapest, Hungary, April 17–23, 2017, Revised Selected Papers, Part I 18*, pages 342–353. Springer.

Richard Beckwith, Christiane Fellbaum, Derek Gross, and George A Miller. 2021. Wordnet: A lexical database organized on psycholinguistic principles. In *Lexical Acquisition*, pages 211–232. Psychology Press.

Gábor Bella, Khuyagbaatar Batsuren, Temuulen Khishigsuren, and Fausto Giunchiglia. 2022. Linguistic diversity and bias in online dictionaries. *University of Bayreuth African Studies Online*, page 173.

Laura Benítez, Sergi Cervell, Gerard Escudero, Mònica López, German Rigau, and Mariona Taulé. 1998. Methods and tools for building the Catalan Wordnet. *arXiv preprint cmp-lg/9806009*.

Luisa Bentivogli and Emanuele Pianta. 2000. Looking for lexical gaps. In *Proceedings of the ninth EURALEX International Congress*, pages 8–12. Stuttgart: Universität Stuttgart.

Mohamed Mahdi Boudabous, Nouha Chaâben Kammoun, Nacef Khedher, Lamia Hadrich Belguith, and Fatiha Sadat. 2013. Arabic Wordnet semantic relations enrichment through morpho-lexical patterns. In *2013 1st International Conference on Communications, Signal Processing, and their Applications (ICCSPA)*, pages 1–6. IEEE.

Peter Paul Buitelaar. 1998. *CoreLex: systematic polysemy and underspecification*. Brandeis University.

Mona Diab. 2004. The feasibility of bootstrapping an Arabic Wordnet leveraging parallel corpora and an English Wordnet. In *Proceedings of the Arabic Language Technologies and Resources, NEMLAR, Cairo*.

Abed Alhakim Freihat. 2014. *An organizational approach to the polysemy problem in Wordnet*. Ph.D. thesis, University of Trento.

Abed Alhakim Freihat, Fausto Giunchiglia, and Biswanath Dutta. 2013. Solving specialization polysemy in wordnet. *Int. J. Comput. Linguistics Appl.*, 4(1):29–52.

Abed Alhakim Freihat, Fausto Giunchiglia, and Biswanath Dutta. 2016. A taxonomic classification of Wordnet polysemy types. In *Proceedings of the 8th Global WordNet Conference (GWC)*, pages 106–114.

Abed Alhkaim Freihat, Biswanath Dutta, and Fausto Giunchiglia. 2015. Compound noun polysemy and sense enumeration in Wordnet. In *Proceedings of the 7th International Conference on Information, Process, and Knowledge Management (eKNOW)*, pages 166–171.

Fausto Giunchiglia, Khuyagbaatar Batsuren, and Abed Alhakim Freihat. 2018. One world-seven thousand languages (best paper award, third place). In *International Conference on Computational Linguistics and Intelligent Text Processing*, pages 220–235. Springer.

Julio Gonzalo. 2004. Sense proximity versus sense relations. *GWC 2004*, page 5.

Mustafa Jarrar. 2021. The Arabic ontology–an Arabic Wordnet with ontologically clean content. *Applied ontology*, 16(1):1–26.

Hadi Khalilia, Gábor Bella, Abed Alhakim Freihat, Shandy Darma, and Fausto Giunchiglia. 2023. Lexical diversity in kinship across languages and dialects. *Frontiers in Psychology*, 14.

Hadi Khalilia, Abed Alhakim Freihat, and Fausto Giunchiglia. 2021a. The quality of lexical semantic resources: A survey. In *Proceedings of the 4th International Conference on Natural Language and Speech Processing (ICNLSP 2021)*, pages 117–129.

Hadi Khalilia, Abed Alhakim Freihat, Fausto Giunchiglia, et al. 2021b. The dimensions of lexical semantic resource quality. In *Proceedings of the Second International Workshop on NLP Solutions for Under Resourced Languages (NSURL 2021) co-located with ICNLSP 2021*, pages 15–21. ACL Anthology.

Adrienne Lehrer. 1970. Notes on lexical gaps. *Journal of linguistics*, 6(2):257–261.

Bernardo Magnini and Gabriela Cavaglia. 2000. Integrating subject field codes into Wordnet. In *LREC*, volume 1413.

Arya D McCarthy, Winston Wu, Aaron Mueller, Bill Watson, and David Yarowsky. 2019. Modeling color terminology across thousands of languages. *arXiv preprint arXiv:1910.01531*.

Rada Mihalcea and Dan I Moldovan. 2001. Ez. Wordnet: Principles for automatic generation of a coarse grained wordnet. In *FLAIRS conference*, pages 454–458.

Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM computing surveys (CSUR)*, 41(2):1–69.

Jian-Yun Nie. 2022. *Cross-language information retrieval*. Springer Nature.

Thierry Poibeau. 2017. *Machine translation*. MIT Press.

Horacio Rodríguez, David Farwell, Javi Ferreres, Manuel Bertran, Musa Alkhalifa, and Maria Antònia Martí. 2008. Arabic wordnet: Semi-automatic extensions using bayesian inference. In *LREC*.

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2008. Yago: A large ontology from Wikipedia and Wordnet. *Journal of Web Semantics*, 6(3):203–217.

Piek Vossen. 1998. A multilingual database with lexical semantic networks. *Dordrecht: Kluwer Academic Publishers. doi*, 10:978–94.

9. Language Resource References

Pushpak Bhattacharyya. 2010. IndoWordNet. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Sabri Elkateb, William Black, Horacio Rodríguez, Musa Alkhalifa, Piek Vossen, Adam Pease, and Christiane Fellbaum. 2006. Building a WordNet for Arabic. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, pages 29–34. European Language Resources Association.

Javier Farreres, Horacio Rodríguez, and Karina Gibert. 2002. Semiautomatic creation of taxonomies. In *COLING-02: SEMANET: Building and Using Semantic Networks*.

George A Miller. 1995. Wordnet: a lexical database for English. *Communications of the ACM*, 38(11):39–41.

Maciej Piasecki, Bernd Broda, and Stanislaw Szpakowicz. 2009. *A Wordnet from the ground up*. Oficyna Wydawnicza Politechniki Wrocławskiej Wrocław.

Yasser Regragui, Lahsen Abouenour, Fettoum Krieche, Karim Bouzoubaa, and Paolo Rosso. 2016. Arabic Wordnet: New content and new applications. In *Proceedings of the 8th Global WordNet Conference (GWC)*, pages 333–341.

Karin Kipper Schuler. 2005. *VerbNet: A broad-coverage, comprehensive verb lexicon*. University of Pennsylvania.

Dan Tufis, Dan Cristea, and Sofia Stamou. 2004. Balkanet: Aims, methods, results and perspectives. a general overview. *Romanian Journal of Information science and technology*, 7(1-2):9–43.

PJTM Vossen. 1999. Eurowordnet.

Arabic Speech Recognition of zero-resourced Languages: A Case of Shehri (Jibbali) Language

Norah Alrashoudi[1], Omar Said Alshahri[2], Hend Al-Khalifa[1]

[1] Department of Information Technology, College of Computer and Information Science, King Saud University, Riyadh, Saudi Arabia.

[2] Islamic Sciences Institute, Diwan of the Royal Court, Salalah, Sultanate of Oman.

omar9297@gmail.com, author3@hhh.com

Abstract

Many under-resourced languages face data scarcity issues due to a lack of standardized writing systems, making ASR training more challenging and costly. However, there's a growing need to adapt ASR for indigenous languages to support language documentation, preservation, and the development of learning materials for these communities. Shehri or Jibbali, a spoken language in Oman, lacks extensive annotated speech data. This paper aims to investigate transfer learning techniques to develop an ASR model for this under-resourced language. We collected a Shehri (Jibbali) speech corpus and utilized transfer learning by fine-tuning pre-trained ASR models on this dataset, including Wav2Vec2.0, HuBERT and Whisper. Evaluation using word error rate (WER) and character error rate (CER) showed that the Whisper model, fine-tuned on the Shehri (Jibbali) dataset, significantly outperformed other models, with the best results from Whisper-medium achieving 3.5% WER. This demonstrates the effectiveness of transfer learning for resource-constrained tasks, showing high zero-shot performance of pre-trained models.

Keywords: Automatic Speech Recognition (ASR), Speech Processing, Transfer Learning, Zero-Resource Languages, Indigenous Languages

1. Introduction

Languages with rich linguistic resources often have extensive corpora and annotated speech data which facilitate the development of accurate and robust Automatic Speech Recognition (ASR) systems. In contrast, languages with limited or zero resources face data scarcity issues, making it a challenge to train ASR models effectively. These languages often lack a standardized writing system, or their written form may be limited to a small number of experts (SIL International, 2022), (R. Coto-Solano et al, 2022). This complicates the transcription process, making it more challenging and costly compared to widely spoken languages. Despite these challenges, there is an increasing need to adapt ASR to work effectively on indigenous languages. One of the main reasons is to support indigenous communities in documenting their languages and preserving their linguistic heritage. Moreover, such adaptations enable these communities to develop learning materials for their languages and facilitate their continuous use (R. Coto-Solano et al, 2022).

One such under-resourced language is Shehri, also known as Jibbali, spoken in Oman. Shehri lacks extensive annotated speech corpora, making conventional supervised training approaches difficult to apply for building an ASR system. However, with a dropping number of fluent speakers, particularly among younger generations, there is a need to develop technological tools that can help document the language.

This study aims to investigate the application of transfer learning techniques to develop an initial ASR capability for Shehri (Jibbali) language without requiring a large, annotated dataset. Therefore, the main contributions of this work are as follows:

1) Collection of a Shehri (Jibbali) speech dataset,

2) Fine-tuning pre-trained ASR models like Wav2Vec2.0, HuBERT, and Whisper on the Shehri dataset, and

3) Evaluation of the adapted models on Shehri (Jibbali) using word error rate and character error rate metrics.

The rest of the paper is structured as follows: Section 2 provides an overview of the Shehri (Jibbali) language. Section 3 discusses related work in under-resourced languages ASR. Section 4 describes the methodology adopted including dataset collection and model fine-tuning. Section 5 presents the results and analysis. Finally, Section 6 concludes the paper and outlines directions for future work.

2. Shehri (Jibbali) Language

The Dhofar Governorate (محافظة ظفار) is situated in the southernmost region of the Sultanate of Oman, bounded to the east by the Al Wusta Governorate, and to the north and northwest by the Rub' al Khali desert, while sharing its southwestern border with Yemen (Oman Encyclopedia, 2013, p. 2321). Additionally, it shares a frontier with Saudi Arabia to the northwest. It encompasses ten administrative divisions, including Salalah, Taqah, Mirbat, Sadah, Shalim, the Halaniyat Islands, Thumrait, Muqshin, Al Mazunah, Dhalkut, and Rakhyout (Ministry of Information, 2020, p. 67).

The population stands at 416,458 individuals (as per the 2020 census). Covering an area of approximately 99,300 km² (Oman Encyclopedia, 2013, p. 2321), the Dhofar Governorate presents a rich linguistic tapestry despite its relatively modest size. It hosts a diverse array of contemporary South Arabian languages, including Shehri (Jibbali), Mahri, Bathari, and Hobyot, alongside Arabic dialects with close affinities to North

OSACT 2024 Workshop, pages 84–92

Arabian, encompassing both urban and Bedouin variants (Al-Kathiri et al., in press). The Shehri (jibbali) language is referred to by different names among its speakers, either exclusively among Shehri speakers or exclusively among Jibbali speakers. However, in research studies, it also sometimes appears with a combination of both, in order to address ambiguities (Al-Hafeezh, 1987; Johnstone, 1981; Rubin, 2014). This language shares 25 letters with Standard Arabic, which are: (/ʔ/ أ, /b/ب, /t/ت, /θ/ ث, /dʒ/ ج, /ħ/ ح, /x/ خ, /d/ د, /ð/ ذ, /r/ ر, /z/ ز, /s/ س, /ʃ/ ش, / tˤ/ ط, /ðˤ/ ظ, /ʕ/ ع, /ɣ/ غ, /f/ ف, /k/ ك, /l/ ل, /m/ م, /n/ ن, /h/ ه, /w/ و, and /j/ ي). It has retained old methods of pronouncing some letters (Al-Mashani, S, 2017). Researchers have proposed various alphabets for these letters, and after consulting with experts Watson and Al-Kathiri, the appropriate alphabet was settled upon Al-Kathiri et al., (2024), which is as follows: (/ɬ/پس , /ʒ/ ڸ, /tʃʾ/ ض, /g/ ج, /ʒʷ/ ج, /ʃ/ش , /ʃʷ/ش , /sʾ/ ص, /ɬʾ/ ض, and /kʾ/ ق). The common letters between Shihri (Jibbali) and Classical Arabic are similar in pronunciation, with some differences in letter characteristics. Some of them correspond to Arabic in both articulation and characteristics, while the other common letters correspond to Arabic in articulation but differ in some characteristics. For further details on the characteristics of these alphabets, refer to Al-Mashani, S,(2017); Watson & Al-Kathiri, (2022).

3. Literature Review

Previous studies have explored the challenges and opportunities in implementing ASR systems for languages with limited resources. R. Coto-Solano et al, (2022) and Gupata et al., (2020) analyzed the challenges involved in the transcription of spoken audio recordings in indigenous languages. In addition, Stan et al., (2022) conducted a comprehensive analysis of the challenges involved in developing ASR systems. Their work highlighted issues such as a lack of annotated data, phonetic variations, and the importance of cultural context in acoustic modeling. Some recent advancements in ASR have leveraged self-supervised learning (SSL) techniques to address resource constraints. A study conducted by Chen et al., (2023) demonstrated the efficiency of the SSL in adapting pre-trained models to indigenous languages, mitigating the need for extensive language-specific training data. In the context of multi-lingual ASR, Arisaputra et al., (2024) evaluated the performance of the XLS-R model on various low-resource languages. They incorporated a 5-gram KenLM into the optimized model and it has resulted in a significant decrease in the Word Error Rate (WER). In addition, Zellou et al., (2024) investigated a cross-language ASR transfer approach for the low-resource Tashlhiyt language, which shares similar phonological inventories with Arabic. Their experiment utilized a commercially available Arabic ASR system without any modifications for the target language, resulting in approximately 45% accurate word transcription. Furthermore, Woldemariam et al., (2020) investigated the efficiency of transfer learning to improve the performance of ASR for the under resourced Semitic language (Amharic). They utilized Deep Neural Network (DNN) acoustic models pre-trained on English and Mandarin as source languages, adapting them to Amharic. Experimental results demonstrate significant improvements through transfer learning compared to the baseline Amharic model. The best enhancements were observed with models transferred from English, achieving WER reductions of 5.75% and 8.06%. In contrast, the Mandarin model achieved a WER reduction of 14.65%, while the baseline only improved by 38.72%.

4. Methodology

The methodology of this study aims to improve an ASR model for a zero-resource language, Shehri (Jibbali) language. We collected speech data from Shehri (Jibbali) speakers and constructed a dataset for the training of ASR model. The study leveraged the efficiency of transfer learning to adapt a pre-trained ASR model to our specific task.

Transfer learning involves leveraging knowledge from pre-trained models on large-scale datasets and adapting them to perform specific tasks or domains with smaller, task-specific datasets (N. Das et al., 2021). This approach allows ASR systems to benefit from the generalization and feature extraction capabilities learned from the pre-training phase, improving performance and reducing the need for extensive labeled data in the target domain (Neyshabur et al., 2020). Transfer learning in ASR typically involves fine-tuning pre-trained models on task-specific data.

In the following subsections, we provide details about our Shehri (Jibbali) speech dataset, give an overview of the fine-tuned models, present our Shehri (Jibbali) ASR model, and explain the model evaluation criteria.

4.1 Dataset

The dataset for Shehri (Jibbali) language speech was collected from 30 speakers, including 23 males and 7 females. Speakers represent the eastern and central parts of Oman, and the western part adjacent to the central part (As'aib region), due to the similarity of dialects in these regions. Informed consent was obtained from all subjects involved in the study.

Each speaker uttered 15 sentences and repeated each sentence 5 times, with a total of 75 utterances for each speaker. In the selection of sentences, we focused on Shehri (Jibbali) phonemes that are not represented in the Arabic language, to ensure the model can effectively distinguish these unique sounds. Table 1 represents the selected sentences with their corresponding Arabic and English translations. The total duration of the dataset is 1 hour and 54 minutes, with an average duration of 3

seconds for each file. The dataset is publicly available through a GitHub repository[1].

English Translation	Arabic Translation	Shehri (Jibbali) Sentences
Light the fire and fetch the firewood	أشعل النار، وأحضر الأثافي	اعلق بسوط، بغد هير اوقودر
When the pot is full, turn off the water.	إذا امتلأ القدر اغلق الماء	هير اصفريت ميلوت قفل اميه
Watch out for the children so they don't fall into the water	انتبه للأطفال كي لا يسقطوا في الماء	اقول لقيلون او يهبي عق اميه
Let them just play on the beach	دعهم فقط يلعبوا على الشاطئ	قلع هوم بس ينحوج ظير حظ
If you exit 'Madinat al-Haqq' you will see my car on the road	إذا خرجت من "مدينة الحق" سترى سيارتي على الطريق	هير تيروفك بخيظول أتيسناً سيارهي ظير اورم
This man often opposes me	هذا الرجل كثيرا ما يعارضني	اغيج ذان يكين ار ديجحود تو
Did the funeral arrive or is it still there?	هل الجنازة وصلت أم لا زالت هناك؟	اچينوزت بروت زحوت من دعوت لحك
I will go and see if the ewe has given birth or is still in labor	سأذهب وأنظر هل ربّت الغنمة أم لا زالت	الغد ليسناً اووز بيروت غيچوت من دعوت
Do you think they are still asleep until now?	أتظنهم لا زالوا نائمين إلى الآن؟	تعمورهم دعود دشيف اد ناصنو؟
If you want it strong, put a lot of tea	إذا أردته جيدا ضع الكثير من الشاي	هير عك تش إصلح ازد شاهي حور
Your whole head is gray	رأسك كله شيب	ارشك بير كلش پسوب
He went in the morning to fetch provisions and has not returned yet	ذهب صباحا ليحضر الزاد ولم يعد بعد	بير اغد كحصف هير خصور بعود اوزحم لو
There appeared among them a wise man	ظهر فيهم رجل حكيم	ضهر عمقوهم غيج بيصير
If your foot falls asleep, don't walk on it	إذا تنملت رجلك فلا تمشي عليها	هير حيضوت فعمك اوتركت ليس لو
They spent their day searching, but they found nothing	ظلوا يومهم يبحثون ولكن لم يجدوا شيئا	قهب يوهم ديغولق ار هيس او كسئ پسي لو

Table 1 : Selected sentences for Shehri (Jibbali) speech dataset

For the training of Shehri (Jibbali) ASR model, we split the data into 80% for the training and 20% for the testing. Before splitting the dataset, we shuffled it to ensure a random distribution of the data. Table 2 represents details of the data division.

Subset	Utterances	Duration
Training	1800	1 hr. and 31 min
Testing	450	22 minutes
Total	2250	1 hr. and 54 min

Table 2: Dataset Summary of the Training and Testing Subset Statistics

4.2 Fine-Tuned Models

In this study, we selected three large-scale pre-trained models, including Wav2Vec2.0, HuBERT, and Whisper. We provide an overview description for each model in the following subsections.

4.2.1 Wav2Vec2.0

Wav2Vecc2.0 is a speech representation framework based on self-supervised learning, enabling the extraction of rich features from raw audio data without the need for annotations or labeled data. The framework was pre-trained on a large quantity of unlabeled data and leveraged Transformer architecture to achieve remarkable performance in speech-related tasks (Baevski et al., 2020). Wav2Vec2 employs a multi-stage architecture consisting of several key components, as shown in Figure 2. The speech features are extracted from raw audio using a CNN, followed by a Transformer layer for contextualized representation aggregation. Self-supervised training involves discretizing the output of the feature encoder into a finite set of speech representations using product quantization. Wav2Vec2.0 offers multiple models with varying parameters and training datasets. The base model, called 'Wav2vec2-base-960h', was trained with over 94 million parameters on 960 hours of the Librispeech corpus, which is designed for native English speakers. Additionally, a large-scale multi-lingual pre-trained model known as 'Wav2Vec2-XLS-R-300M' was pre-trained with up to 300 million parameters on 436,000 hours of unannotated speech data collected from diverse corpora spanning 128 languages.

[1] https://github.com/iwan-rg/Shehri-Jibbali-Speech-Dataset

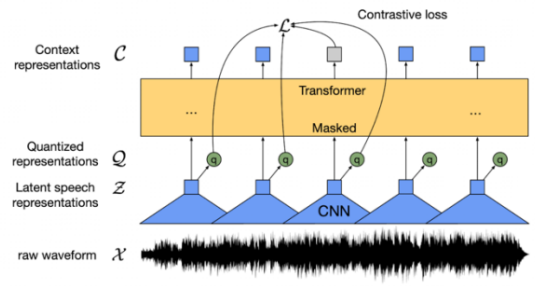


Figure 2: The architecture of Wav2Vec2.0 (Baevski et al., 2020)

4.2.2 HuBERT

Hidden unit BERT (HuBERT) is a self-supervised speech representation framework learned by masked prediction of hidden units (Hsu et al., 2021). Figure 3 shows the architecture of HuBERT framework. HuBERT integrates an offline clustering step for BERT-like pre-training with noisy labels. It utilizes a BERT model on masked continuous speech features to predict predetermined cluster assignments, focusing the predictive loss on masked regions to learn robust high-level representations. This setup enables simultaneous learning of acoustic and language models from continuous inputs, addressing acoustic modeling challenges and capturing long-range temporal relations in learned representations. HuBERT model was pre-trained on either standard Librispeech 960h or the Libri-Light 60k hours on three model sizes, including Base (90M parameters), Large (300M parameters), and X-Large (1B parameters).

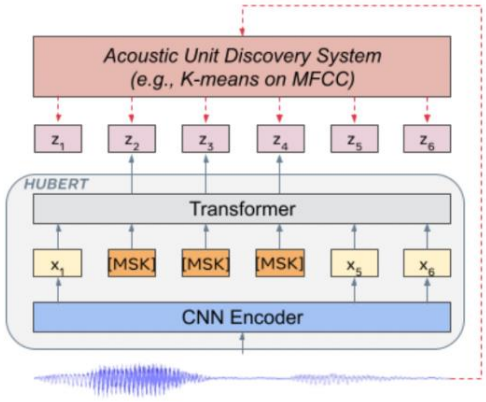


Figure 3: The architecture of HuBERT (Hsu et al., 2021)

4.2.3 Whisper

Whisper is a large-scale speech representation framework based on a weakly-supervised approach, that was pre-trained on 680,000 hours of labeled audio data, including English speech and multilingual data covering 96 languages to perform two different tasks: speech recognition and speech translation (Radford et al., 2022). The English-only models were trained on speech recognition tasks, whereas the multi-lingual models were trained on speech recognition and speech translation tasks to predict the transcription of different languages. Figure 4 illustrates the architecture of the Whisper, featuring an encoder-decoder Transformer chosen for reliability and scalability. Audio data is resampled to 16,000 Hz and transformed into an 80-channel log-magnitude Mel spectrogram representation. The encoder consists of a stem with two convolutional layers followed by sinusoidal position embeddings and pre-activation residual blocks. The decoder utilizes learned position embeddings and tied input-output token representations. Both the encoder and decoder have the same width and number of transformer blocks for consistency in processing the input and generating the output. Whisper was pre-trained on several models with different numbers of parameters, ranging from 39M to 1.5B parameters.

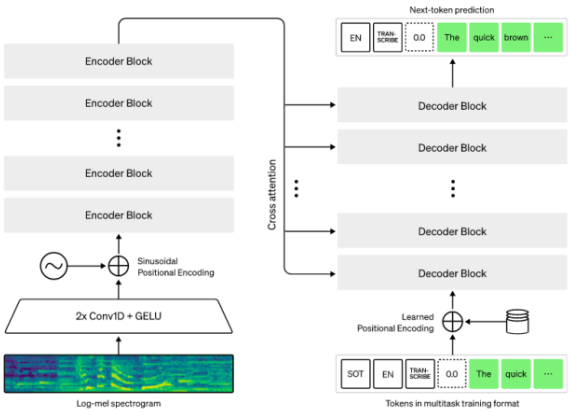


Figure 4: The architecture of Whisper (Radford et al., 2022)

4.3 Shehri (Jibbali) ASR Models

To implement an ASR model for the Shehri (Jibbali) language, we utilized a transfer learning approach by fine-tuning several pre-trained models, including Whisper (Radford et al., 2022), HuBERT (Hsu et al., 2021), and Wav2Vec2.0 (Babu et al., 2021) on our constructed speech dataset for Shehri (Jibbali) language. For Wav2Vec2.0, we selected the XLS-R model as a large-scale model for cross-lingual speech data that was trained on 436K hours of unannotated speech data including 128 different languages. For HuBERT, we selected the large model that was trained on both Libri-Light 60k and LibriSpeech 960 hours of speech data. For Whisper, we trained the base, tiny, small, medium, and large-v3 models with varying numbers of parameters, ranging from 39M to 1.5B of parameters.

Training Details. Models were trained on NVIDIA Tesla T4 GPU with 54GB of memory and CUDA

version 12.2. We utilized Huggingface trainer[2] to train each model, and PyTorch (version 2.1.0+cu121) to perform GPU-accelerated training. The pre-processing step was applied to both audio and textual data. The transcription texts contain some punctuation marks, such as '?' and ',', then we normalized the text by removing these marks before training. Additionally, the audio data was re-sampled to 16kHz and converted the raw waveform of the speech signals into a floating array. During the fine-tuning process, we selected similar configurations and hyperparameter settings for both the XLS-R and HuBERT models, because they were implemented on closely related architectures. According to (Babu et al., 2021), we trained these models with a learning rate of 3e-4, 500 warmup steps, 20 epochs, 16 for the batch size, and no weight decay. Table 3 shows the hyperparameters of all fine-tuned models.

For training Whisper models, we encountered issues related to the GPU and computational resources due to its huge number of parameters. To address these issues, we applied some parameter-efficient fine-tuning (PEFT) techniques for model optimization and improving the training process. PEFT is a technique employed in Natural Language Processing (NLP) and ASR to enhance the effectiveness of pre-trained language models on specific downstream tasks. It aims to decrease the hyperparameter numbers for the large-scale language models, which minimizes the computational resources and time compared to the training of the entire model (Z. Fu et al., 2023). We trained Whisper models using two PEFT methods, named Int8 matrix multiplication for Transformers at scale (LLM.int8) (Dettmers et al., 2022) and low-rank adaption of large language models (LORA) (E. J. Hu et al., 2023). LLM.int8 was utilized to lower the precision of floating-point data types, thereby reducing the memory required to store model weights. The LORA approach involves freezing the weights of the pre-trained model and incorporating trainable rank decomposition matrices into each layer of the Transformer architecture, which reduces the number of trainable parameters. After performing these methods, the number of parameters in Whisper models has reduced to utilize only 1% to 1.5% of all trainable parameters. For example, the number of parameters of the medium model has been reduced from 9.4 M to 773K parameters, which improves the performance of the training process using less memory and other computational resources.

Table 4 presents the number of model parameters, trainable parameters, and training time for each model. The '*All parameters*' represents the number of parameters for each pre-trained model, while the number of 'trainable parameters' refers to the number of parameters that are trainable during the training process. As shown in Whisper models, the number of trainable parameters was reduced after optimization and parameter reduction, while Wav2Vec2 and HuBERT remained unchanged. Additionally, the time consumed to train Wav2Vec2 and HuBERT models are closely similar, because they were trained on the same settings and have the same number of parameters. The training time of Whisper models is higher than Wav2Vec2 and HuBERT, despite their smaller sizes. However, the training time increased exponentially with the growth of model parameters.

Hyper-parameters	**XLS-R-Wav2Vec2**	**HuBERT**	**Whisper**
learning-rate	3e-4	3e-4	1e-3
warmup_steps	500	500	50
num_train_epochs	20	20	10
batch_size	16	16	6
gradient_accumulation_steps	2	2	1

Table 3: Hyperparameter settings for fine-tuning Wav2Vec2.0, HuBERT, and Whisper

Model	**All Parameters**	**Trainable Parameters**	**Training Time**
XLS-R-Wav2Vec2	315 M	315 M	49 min
HuBERT-large	315 M	315 M	43 min
Whisper-tiny	39 M	589 K	57 min
Whisper-base	75 M	1.1 M	1 hr.
Whisper-small	244 M	3.5 M	1 hr. and 56 min
Whisper-medium	769 M	9.4 M	3 hrs. and 24 min
Whisper-large-v2	1.5 B	15 M	4 hrs. and 28 min

Table 4: Number of model parameters, training parameters, and the training time consumed for each mode

4.4 Model Evaluation

The evaluation measures of each model are word error rate (WER) and character error rate (CER) which are commonly used to evaluate the performance of ASR systems. Both are used to measure the rate of errors in transcribing the recognized speech compared to the reference (ground truth) transcription. WER measures the rate of errors in recognized speech at the word level, while CER measures errors at the character level. The

[2] https://huggingface.co/docs/transformers/main/trainer

WER and CER are calculated as the following equations (S. Young et al., 1995):

$$WER = \frac{S + I + D}{N} \#(1)$$

$$CER = \frac{S + I + D}{N} \#(2)$$

N refers to the number of labels, whereas the S, I, and D, are referring to the number of substitutions, insertions, and deletions of the recognized words or characters. A lower rate of WER or CER indicates better accuracy in the ASR system's transcription output.

5. Results and Analysis

Table 5 presents the achieved results among all models. XLS-R-Wav2Vec2 and HuBERT-large achieved similar performance since they followed the same architecture and model size, both gained a WER of 19%, while XLS-R-Wav2Vec2 achieved a lower CER at 6.5%. In contrast, Whisper models demonstrated superior performance with WER ranging from 5.5% to 3.5% and CER ranging from 4% to 2.6%. Among the Whisper models, Whisper-medium has the lowest WER and CER, while Whisper-tiny has the highest. There is a noticeable improvement in performance as the model size increases within the Whisper models, with Whisper-medium and Whisper-large-v2 achieving the lowest WER and CER among all models.

Overall, Whisper models consistently outperform XLS-R-Wav2Vec2 and HuBERT-large in terms of both WER and CER, with Whisper-medium demonstrating the best performance among all models. These results highlight Whisper's efficiency in recognizing the Jibbali language, even with a limited amount of training data.

Figure 5 shows the confusion matrix of the recognized and misrecognized characters obtained from the Whisper-medium model. The confusion matrix demonstrates the effectiveness of the mode in character recognition with a higher number of correctly recognized characters compared to the misrecognized characters.

Model	WER (%)	CER (%)
XLS-R-Wav2Vec2	19.2%	7.5%
HuBERT-large	19.4%	11.8%
Whisper-tiny	5.5%	4.0%
Whisper-base	4.3%	3.2%
Whisper-small	3.8%	3.0%
Whisper-medium	**3.5%**	**2.62%**
Whisper-large-v2	**3.5%**	**2.65%**

Table 5: ASR Model evaluation results based on error rate (WER %) and character error rate (CER %)

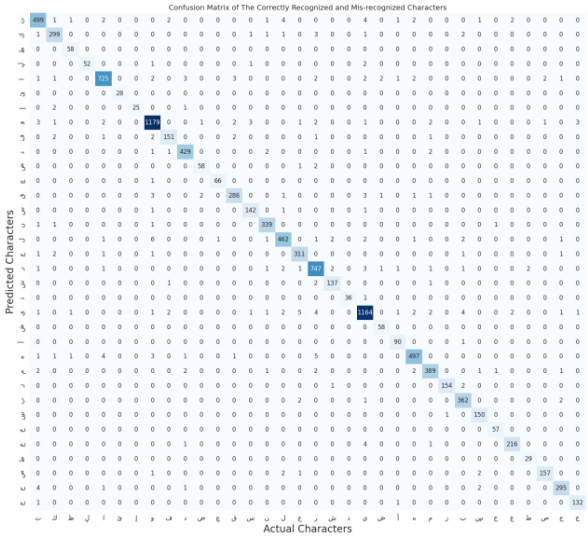


Figure 5 Confusion matrix of the recognized and misrecognized characters obtained from Whisper-medium mode

Table 6 represents examples of the transcribed text predicted from XLS-R-Wav2Vec2, HuBERT-large, and Whisper-medium models with the ground truth. These examples show how these models can identify Shehri (Jibbali) sounds that are not presented in the Arabic language.

The XLS-R-Wav2Vec and Whisper models were trained on a large amount of cross-lingual data, including Arabic. However, Arabic and Shehri (Jibbali) languages contain several similar sounds, as discussed in Section 2, which enabled these models to achieve high performance results. In contrast, the Shehri (Jibbali) language includes some unique sounds not presented in the Arabic language. Despite these unique sounds and the limited size of our dataset, the results obtained were high and accurate. This demonstrates the efficiency of the transfer learning approach for such resource-constrained

tasks and the high performance of the pre-trained models applied in this study.

Ground Truth	XLS-R-Wav2Vec2	HuBERT-large	Whisper-medium
ضهر عمقوهم غيج بيصير	ضهر عمقوهم غيج بيص بيصير	ضهر عمقوهم غيج بيص بيصير	ضهر عمقوهم غيج بيصير
هير تيروفك بخيضول أتپسناً سيارهي ظير اورم	هير تيروفك بخيضولأتپسناً سيارهي ظير اورم	هير تيروفك بخيضول أتپسناً سيارهي ظير اورم	هير تيروفك بخيضول أتپسناً سيارهي ظير اورم
بير اغد كحصف هير خصور بعود اوزحم لو	بير اغد كحصفهير خصور بعود اوزحم لو	بير اغد كحصف هير خصور بعود اوزحم لو	بير اغد كحصف هير خصور بعود اوزحم لو

Table 6 : Examples of predicted transcriptions obtained from XLS-R-Wav2Vec2, HuBERT-large, and Whisper-medium models with the ground truth.

To analyze speech recognition errors resulting from various models from a linguistic perspective, we selected several examples of misrecognized transcription from different models to discover the reasons behind these failures. Table 7 represents different reference examples with their predicted transcription. In the first example, the model misrecognized and deleted the long vowel (/w/ و) in the word "خيضول" because the speaker pronounced it rapidly which reduces the pronunciation rate of the vowel sound to the extent of hiding it, leading to the appearance of the word "خيضول" without the vowel sound, as follows: "خيضل". In the second example, the sounds (/ħ/ ح) and (/f/ ف) are both voiceless consonants, which are produced without the vibration of the vocal cords. This characteristic increases in the pronunciation of the sound (/ħ/ ح), so that its pronunciation is close to the sound (/h/ ه). In this case, the model predicted the sound (/ħ/ ح) as (/f/ ف), making the word "قحل" instead of "قفل," where the speaker in the example pronounced more like "قهل". In the last three examples, the speakers were very fast in their pronunciations which made the models mis-recognized some sounds. This leads us to one of the challenges in constructing a speech dataset, which is ensuring that speakers pronounce sentences at a balanced pace, as the speaking rate affects training results, especially if the language is new to the trained model.

Ground Truth	Predicted Transcription
هير تيروفك بخيضول أتپسناً سيارهي ظير اورم	هير تيروفك بخيضل أتپسناً سيارهي ظير اورم
هير اصفريت ميلوت قحل اميه	هير اصفريت ميلوت قفل اميه
اغيج ذان يكين ار ديجحود تو	ار عكذا تيجحود ار
اغيج ذان يكين ار ديجحود تو	عكوز بير كلش او كحصير
هير حيضوت فعمك اوتركت ليس لو	ضهر عمقوهم غيج بيصير

Table 7: *Examples of misrecognized transcription resulted from different trained models*

6. Conclusion

This study presented a promising approach to developing an ASR system for the under-resourced Shehri (Jibbali) language using transfer learning techniques. By fine-tuning various speech pre-trained models like Wav2Vec2.0, HuBERT, and Whisper on the collected Shehri (Jibbali) speech dataset, the research demonstrated the capability of transfer learning methods to address the limitations in data availability that are typically faced for under-resourced languages. The evaluation results showed that the Whisper models significantly outperformed the other models that were evaluated, achieving word error rates as low as 3.5%. This highlights the efficiency of Whisper models in adapting to low-resource tasks even with limited training data.

While the results obtained were encouraging, there is still room for improvement. The performance of the models could be enhanced further by collecting a larger and more diverse Shehri (Jibbali) speech dataset containing a greater variety of speakers, accents, acoustic environments, and content. This would allow the models to learn from more varied data and generalize better. Additionally, future work could explore utilizing multilingual models that have been trained on languages that are closely related to Shehri (Jibbali) both linguistically and geographically. Such models may learn representations that transfer even better.

Overall, this research achieved promising outcomes and demonstrated that transfer learning is an effective solution for overcoming the computational challenges presented by under-resourced languages due to a lack of annotated data resources. With continued efforts to develop larger datasets and optimize model architectures, even more advanced ASR capabilities can be developed to support the documentation, preservation and technological empowerment of under-represented languages like Shehri (Jibbali). The approach presented in this study paves the way for applying similar techniques to other low-resource languages.

7. References

— (2003). Lisān Ẓufār al-ḥimyarī al-muʿāṣir. Dirāsah muʿjamiyyah muqāranah, Jāmiʿat al-Sulṭān Qābūs, Markaz al-dirāsāt al-ʿumāniyyah.

"Ethnologue: Languages of Africa and Europe, Twenty-Fifth Edition," SIL International, 2022. Accessed: Feb. 18, 2024. [Online]. Available: https://www.sil.org/resources/publications/entry/93719

A. Babu *et al.*, "XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale,", 2021 arXiv.org. Accessed: Aug. 08, 2023. [Online]. Available: https://arxiv.org/abs/2111.09296v3

A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations," in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2020, pp. 12449–12460. Accessed: Apr. 06, 2022. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/92d1e1eb1cd6f9fba3227870bb6d7f07-Abstract.html

A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust Speech Recognition via Large-Scale Weak Supervision." arXiv, Dec. 06, 2022. Accessed: Nov. 05, 2023. [Online]. Available: http://arxiv.org/abs/2212.04356

Āl Ḥāfiẓ, ʿAlī Muḥsin. (1987). Min lahajāt "mahrah" wa-'ādābihā. Majallat an-nahḍah al-ʿumāniyyah. Muscat.

Al-Kathiri, A, Al-Mashani, A & Alshahri, O. (2024). Al-Luban Wal-Turath Al-Thaqafi. Ministry of Culture, Sports, and Youth, Literary Forum. Muscat, Sultanate of Oman.

Al-Kathiri, Amer; Al-Maashani, Abdulaziz; Al-Kathiri, Salem. (In press). Al-Wadu' al-lughawi fi Dhofar: Dirasat lughawiyat ijtimaiyah. Arab Journal of Humanities Sciences.

Al-Maashani, Saeed. (2017). Al-Ishtiqagh fi al-Lughah al-Jabaliiyah: Mawazanah bil Lughah al-Arabiyyah. Research and Cognitive Communication Center. Riyadh.

B. Neyshabur, H. Sedghi, and C. Zhang, "What is being transferred in transfer learning?," in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2020, pp. 512–523. Accessed: Feb. 18, 2024. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/0607f4c705595b911a4f3e7a127b44e0-Abstract.html

C.-C. Chen, W. Chen, R. Zevallos, and J. E. Ortega, "Evaluating Self-Supervised Speech Representations for Indigenous American Languages." arXiv, Oct. 08, 2023. Accessed: Jan. 22, 2024. [Online]. Available: http://arxiv.org/abs/2310.03639

E. J. Hu *et al.*, "LoRA: Low-Rank Adaptation of Large Language Models." arXiv, Oct. 16, 2021. Accessed: Nov. 06, 2023. [Online]. Available: http://arxiv.org/abs/2106.09685

G. V. Stan, A. Baart, F. Dittoh, H. Akkermans, and A. Bon, "A Lightweight Downscaled Approach to Automatic Speech Recognition for Small Indigenous Languages," in *14th ACM Web Science Conference 2022*, Barcelona Spain: ACM, Jun. 2022, pp. 451–458. doi: 10.1145/3501247.3539017.

G. Zellou and M. Lahrouchi, "Linguistic disparities in cross-language automatic speech recognition transfer from Arabic to Tashlhiyt," *Sci. Rep.*, vol. 14, no. 1, Art. no. 1, Jan. 2024, doi: 10.1038/s41598-023-50516-3.

Johnstone. (1981). Jibbali Lexicon. London: Oxford University Press.

Ministry of Heritage and Culture. (2013). The Omani Encyclopedia (Vol. 6). Muscat: Ministry of Heritage and Culture.

N. Das, S. Bodapati, M. Sunkara, S. Srinivasan, and D. H. Chau, "Best of Both Worlds: Robust Accented Speech Recognition with Adversarial Transfer Learning." arXiv, Mar. 09, 2021. Accessed: Feb. 18, 2024. [Online]. Available: http://arxiv.org/abs/2103.05834

National Center for Statistics and Information. (2020). Census 2020. Muscat: National Center for Statistics and Information.

P. Arisaputra, A. T. Handoyo, and A. Zahra, "XLS-R Deep Learning Model for Multilingual ASR on Low- Resource Languages: Indonesian, Javanese, and Sundanese." arXiv, Jan. 12, 2024. Accessed: Jan. 31, 2024. [Online]. Available: http://arxiv.org/abs/2401.06832

R. Coto-Solano *et al.*, "Development of Automatic Speech Recognition for the Documentation of Cook Islands Māori", 2022.

Rubin, A. D. (2014). The Jibbali (Shahri) language of Oman: grammar and texts. In The Jibbali (Shaḥri) Language of Oman. Brill.

S. Young *et al.*, *The HTK Book*, vol. 3.4. 1995. Accessed: Nov. 05, 2023. [Online]. Available: https://www.inf.u-szeged.hu/~tothl/speech/htkbook.pdf

T. Dettmers, M. Lewis, Y. Belkada, and L. Zettlemoyer, "LLM.int8(): 8-bit Matrix Multiplication for Transformers at Scale." arXiv, Nov. 10, 2022. doi: 10.48550/arXiv.2208.07339.

V. Gupta and G. Boulianne, "Speech Transcription Challenges for Resource Constrained Indigenous Language Cree," in *Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL)*, D. Beermann, L. Besacier, S. Sakti, and C. Soria, Eds., Marseille, France: European Language Resources association, May 2020, pp. 362–367. Accessed: Jan. 22, 2024. [Online]. Available: https://aclanthology.org/2020.sltu-1.51

W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, “HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units,” *IEEEACM Trans. Audio Speech Lang. Process.*, vol. 29, pp. 3451–3460, 2021, doi: 10.1109/TASLP.2021.3122291.

Watson, J. C., & Al-Kathiri, A. A. A. (2022). A phonetically “unnatural” class in Central and Eastern Shehret (Jibbali). Kervan: International Journal of Afro-Asiatic Studies, 26(1), 129-159.

Y. Woldemariam, “Transfer Learning for Less-Resourced Semitic Languages Speech Recognition: the Case of Amharic,” in Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL), D. Beermann, L. Besacier, S. Sakti, and C. Soria, Eds., Marseille, France: European Language Resources association, May 2020, pp. 61–69. Accessed: Mar. 30, 2024. [Online]. Available: https://aclanthology.org/2020.sltu-1.9

Z. Fu, H. Yang, A. M.-C. So, W. Lam, L. Bing, and N. Collier, “On the Effectiveness of Parameter-Efficient Fine-Tuning,” *Proc. AAAI Conf. Artif. Intell.*, vol. 37, no. 11, Art. no. 11, Jun. 2023, doi: 10.1609/aaai.v37i11.26505.

OSACT6 Dialect to MSA Translation Shared Task Overview

Ashraf Elneima, AhmedElmogtaba Abdelaziz, Kareem Darwish
aiXplain Inc.,
San Jose, CA, USA
{ashraf.hatim,ahmed.abdelaziz,kareem.darwish}@aixplain.com

Abstract

This paper presents the Dialectal Arabic (DA) to Modern Standard Arabic (MSA) Machine Translation (MT) shared task in the sixth Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT6). The paper describes the creation of the validation and test data and the metrics used and provides a brief overview of the submissions to the shared task. In all, 29 teams signed up, and 6 teams made submissions to the competition's leaderboard, with five of them submitting papers to the OSACT6 conference. The teams used a variety of datasets and approaches to build their MT systems. The most successful submission involved using zero-shot and n-shot prompting of ChatGPT.

Keywords: Machine translation, Dialectal translation

1. Introduction

While **M**odern **S**tandard **A**rabic (MSA) serves as the standardized formal language across the Arab world, **D**ialectal **A**rabic (DA) encompasses various regional dialects with unique vocabulary and morphology. However, resources for processing DA are scarce, posing challenges for tasks like machine translation. To overcome this, researchers have explored methods such as using MSA as a bridge language for translation. By pivoting on MSA, the translation accuracy of highly dialectal Arabic text into other languages could be enhanced.

The dialect to MSA machine translation shared task offers an opportunity for researchers and practitioners to tackle the intricate challenge of translating various Arabic dialects into Modern Standard Arabic. With the rich linguistic diversity across Arabic-speaking regions, this task aims to advance machine translation capabilities and bridge the gap between colloquial spoken Arabic and the formal written language. Participants worked on developing and refining translation models that can accurately and fluently convert dialectal Arabic text into MSA, making it a crucial initiative for improving communication and comprehension in the Arabic-speaking world.

The shared task covers multiple dialects, namely: Gulf, Egyptian, Levantine, Iraqi, and Maghrebi. For each dialect, there is a set of 200 sentences written in both MSA and dialect will be provided for fine-tuning (validation set), and the testing was done on a blind set of 1,888 test sentences that cover all 5 dialects (test set). The participants were free to use whatever resources at their disposal to train and fine-tune their systems. In this paper we:

- Describe the dataset and metrics that were used
- Introduce the common approaches that the participants used in their submissions

The shared task was run on CodaLab, and the details of submissions, data formats, and leaderboard reside there[1].

2. Related Work

Several works focused on machine translation from dialectal Arabic to MSA. For instance, Guellil et al. (2017) proposed a neural system translating Algerian Arabic (Arabizi and Arabic script) to MSA, while Baniata et al. (2018) introduced a system for translating Levantine and Maghrebi dialects to MSA. The **N**uanced **A**rabic **D**ialect **I**dentification (NADI) (Abdul-Mageed et al., 2020, 2021, 2022, 2023) task series is dedicated to addressing challenges in general Arabic dialect processing. While the first two versions focused on dialect identification and sentiment, the 2023 edition emphasized machine translation from Arabic dialects to MSA, a critical yet relatively nascent NLP task. Subtasks 2 and 3 of NADI2023 focused on machine translation from four Arabic dialects (Egyptian, Emirati, Jordanian, and Palestinian) to MSA at the sentence level. The datasets for these subtasks, named MT-2023-DEV and MT-2023-TEST, were manually assembled. MT-2023-DEV consists of 400 sentences, with 100 representing each dialect, while MT-2023-TEST comprises a total of 2,000 sentences, with 500 from each dialect. For subtask 3 training, participants were given the freedom to use additional datasets, whereas subtask 2 was restricted to utilizing MADAR-4-MT only. The MADAR corpus contains parallel sentences representing the dialects of

[1] https://codalab.lisn.upsaclay.fr/competitions/17118

OSACT 2024 Workshop, pages 93–97

25 cities across the Arab world, with translations in English, French, and MSA (Bouamor et al., 2019a). Addressing the original dataset's lack of country-level labels, a mapping was executed to link the 25 cities to their respective countries, resulting in the creation of MADAR-18. Furthermore, MADAR-4-MT integrates dialectal-to-MSA data from four specific dialects (Egyptian, Emirati, Jordanian, and Palestinian) extracted from MADAR-18, tailored for training MT systems in subtask 2.

3. Data and Metrics

3.1. Data

To create the validation and test set, we extracted 2,000 random segments per dialect from the **S**audi **A**udio **D**ataset for **A**rabic (SADA), which is an Arabic audio dataset composed of roughly 650 hours that are transcribed and annotated with gender and dialect (Alharbi et al., 2024). For the Gulf dialect, SADA used finer-grained labels, namely Najdi, Hijazi, Gulf, Shamali, and Gulf. Thus, we combined all of them when picking the random samples. Similarly, we combined Algerian and Moroccan segments for the Maghrebi dialect. Given the randomly extracted samples, we followed a two-step process to translate them into MSA. First, we prompted chatGPT to translate the dialectal sentences to MSA using the following prompt:

ترجم النصوص التالية للغة العربية الفصحى ،
اكتب كلا من النص الاصلي وترجمته بالعربية الفصحى
وافصل بينهما باستخدام هذا الرمز #

Translation: *Translate the following texts to standard Arabic. Write the original text followed by the standard Arabic and separate between with them with # symbol.*

In the second step, we enlisted the help of native speakers of the different dialects to review the translations to ascertain their correctness and to correct the translations as needed. The reviewers had the option of accepting the translation as is, editing and accepting, or skipping if: the source dialect was different, the source was MSA, or the source was not comprehensible or translatable. The reviewing was done using a version of Label Studio[2] on the aiXplain platform[3] with the interface shown in Figure 1. We asked the reviewers to review at least 500 segments. Table 1 shows the breakdown of the reviewed segments.

As can be seen, we surpassed 500 segments for all dialects except Iraqi. For all, we randomly picked 200 for validation and used the rest for testing. The validation set was provided with the ground truth translation, while the test set was provided without translation. Table 2 shows reviewed samples for the different dialects.

Dialect	Total	Valid	Test
Gulf	786	200	586
Levantine	768	200	568
Maghrebi	543	200	343
Egyptian	514	200	314
Iraqi	277	200	77

Table 1: The breakdown of the reviewed segments.

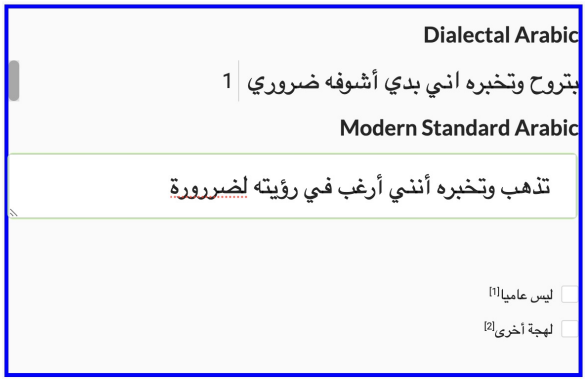


Figure 1: Reviewer interface

3.2. Metrics

For evaluation, we elected to use 2 different metrics that require ground-truth references, namely BLEU (Papineni et al., 2002) and Comet DA (Rei et al., 2022), which reportedly better correlates with human judgments compared to BLEU. While BLEU ranges between 0 and 1, with 1 being the highest possible score, Comet DA ranges between -1 and 1, with 1 being the highest score. BLEU was computed using the NLTK toolkit[4]. Since the computation of Comet DA is relatively computationally expensive, the computation was done on the aiXplain platform[5].

4. Submissions

Out of the 29 teams that signed up for the shared task, 6 teams made submissions. The teams used a variety of datasets and approaches to train their MT systems. Table 3 showcases the outcomes achieved by the participating teams.

MBZUAI (Atwany et al., 2024): The MBZUAI team used the MADAR dataset (Bouamor et al., 2019b) for training, which includes 95,600 dialectal

[2] https://labelstud.io/
[3] https://label.aixplain.com
[4] https://www.nltk.org/
[5] https://platform.aixplain.com

dialects	source	target
Gulf	عبد الله من جد يعني خاش	عبد الله دخل حقا
Egyptian	هتكون مين يعني العروسة؟	من ستكون إذا العروسة؟
Levantine	إي حركة لا تخليه لوحده	أي حركة لا تتركه وحده
Iraqi	هلا هلا والله بوخي وعليكم السلام عوافي عوافي يا وخي	مرحباً بك يا صديقي وعليكم السلام، أصابتك العافية
Maghrebi	ربي يهدينا ويرزقنا حسن الخاتمة ياااارب	اللهم اهدنا وارزقنا خاتمة حسنة يا رب

Table 2: Random samples from the validation set

Group	BLEU	Comet DA
MBZUAI	29.6	0.028
aiXplain	25.2	-0.005
ASOS	22.3	0.004
MSAizer	21.8	0.002
nourrabih	10.1	-0.098
Sirius_Translators	9.6	-0.064

Table 3: Results for teams who submitted results and papers.

sentences with their corresponding MSA equivalents. The team experimented with a variety of models including the No Language Left Behind (NLLB) MT model from Meta, with and without finetuing, AraT5 with fine-tuning (Nagoudi et al., 2022), and chatGPT in zero-shot and 3-shot settings. Their team achieved the best results in the shared task using chatGPT prompting with 29.6 and 0.028 BLEU and Comet DA scores respectively. The $nourrabih$ team seems to have merged with the MBZUAI team.

aiXplain (Abdelaziz et al., 2024): The aiXplain team used two training datasets, namely the NADI dataset (124,000 sentences) (Derouich et al., 2023) and segments that were extracted from the SADA dataset and automatically translated to MSA using chatGPT 3.5 (1,027,153). For the MT model, they used two different neural MT toolkits, namely MarianMT (Junczys-Dowmunt et al., 2018) and Joey NMT (Kreutzer et al., 2019). Their best results were 25.2 and -0.005 for BLEU and Comet DA respectively on the test set.

ASOS (Nacar et al., 2024): The ASOS team employed data augmentation techniques utilizing GPT-3.5 and GPT-4 to increase the validation set size from 200 to 600 examples per dialect. They leveraged a dataset comprising 3000 samples (600 for each of the 5 dialects) for fine-tuning AraT5 v2. Their top-performing results on the test set were 22.3 for BLEU and 0.004 for Comet DA.

MSAizer (Fares, 2024): The MSAizer team fine-tuned the AraT5 model using four different datasets. Three of these datasets consisted of dialect to MSA pairs, namely: MADAR (95,600 sentences) (Bouamor et al., 2019b), NLC (120,600) (Krubiński et al., 2023), and PADIC (41, 680) (Meftouh et al., 2015). The fourth dataset was created by back-translating sentences from MSA, using a subset of OPUS data (965, 020) (Tiedemann, 2012). The final training dataset comprised 700,386 dialect-MSA sentence pairs. Their best results on the test set were 21.79 BLEU and 0.002 for Comet DA, respectively.

Sirius_Translators (Alahmari, 2024): This teams used 5 different datasets to train an MT model, namely MADAR (95,600 sentences) (Bouamor et al., 2019b), PADIC (32,060) (Meftouh et al., 2018), Dial2MSA (60,277) (Mubarak, 2018), Arabic STS (5,516) (Al Sulaiman et al., 2022), SADID (5,994) (Abid, 2020). For translation, the team fine-tuned multiple AraT5 models, namely AraT5 base, AraT5v2-base-1024, AraT5-MSA-Base, and AraT5-MSA-Small, with AraT5v2-base-1024 (Nagoudi et al., 2022) achieving the best results with 9.6 and -0.064 for BLEU and Comet DA respectively on the test set.

5. Conclusion

In this paper, we presented the dialectal Arabic to MSA translation shared task for OSACT6. The validation and test data for the shared task were prepared using a combination of LLM-based automatic translation and human verification and correction. In all, 29 teams signed up for the shared task, with 6 of them making submissions to the competition's leaderboard and 5 of them submitting system papers. Two main themes appeared in the submission, namely: using LLMs for data augmentation and creation, and finetuing either NMT models or LLMs (most notably AraT5) for translation. The best results were attained using LLMs, specifically chatGPT, using zero-shot and n-shot prompting.

6. References

AhmedElmogtaba Abdelaziz, Ashraf Elneima, and Kareem Darwish. 2024. Llm-based mt data creation: Dialectal to msa translation shared task. In *Proceedings of the 6th Workshop on Open-Source Arabic Corpora and Processing Tools, LREC'2024*.

Muhammad Abdul-Mageed, AbdelRahim Elmadany, Chiyu Zhang, El Moatez Billah Nagoudi, Houda Bouamor, and Nizar Habash. 2023. Nadi 2023: The fourth nuanced arabic dialect identification shared task. *arXiv preprint arXiv:2310.16117*.

Muhammad Abdul-Mageed, 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, Abdel-Rahim 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, Abdel-Rahim Elmadany, Houda Bouamor, and Nizar Habash. 2022. Nadi 2022: The third nuanced arabic dialect identification shared task. *arXiv preprint arXiv:2210.09582*.

Wael Abid. 2020. The sadid evaluation datasets for low-resource spoken language machine translation of arabic dialects. In *International Conference on Computational Linguistics*.

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):e0272991.

Salwa Alahmari. 2024. Sirius_translators at osact6 2024 shared task: Fin-tuning arat5 models for translating arabic dialectal text to modern standard arabic. In *Proceedings of the 6th Workshop on Open-Source Arabic Corpora and Processing Tools, LREC'2024*.

Sadeen Alharbi, Areeb Alowisheq, Zoltan Tuske, Kareem Darwish, Abdullah Alrajeh, Abdulmajeed Alrowithi, Aljawharah Bin Tamran, Asma Ibrahim, Alnajim Raneem Aloraini, Raghad, Ranya Alkahtani, Renad Almuasaad, Sara Alrasheed, Shaykhah Alsubaie, and Yaser Alonaizan. 2024. Sada: Saudi audio dataset for arabic. *ICCASP 2024*.

Hanin Atwany, Nour Rabih, Ibrahim Mohammed, Abdul Waheed, and Bhiksha Raj. 2024. Osact 2024 task 2: Arabic dialect to msa translation. In *Proceedings of the 6th Workshop on Open-Source Arabic Corpora and Processing Tools, LREC'2024*.

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

Houda Bouamor, Sabit Hassan, and Nizar Habash. 2019a. The madar shared task on arabic fine-grained dialect identification. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 199–207.

Houda Bouamor, Sabit Hassan, and Nizar Habash. 2019b. 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.

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 ArabicNLP 2023*, pages 683–689, Singapore (Hybrid). Association for Computational Linguistics.

Murhaf Fares. 2024. Arat5-msaizer: Translating dialectal arabic to msa. In *Proceedings of the 6th Workshop on Open-Source Arabic Corpora and Processing Tools, LREC'2024*.

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.

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.

Julia Kreutzer, Jasmijn Bastings, and Stefan Riezler. 2019. Joey NMT: A minimalist NMT toolkit for novices. 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): System Demonstrations*, pages 109–114, Hong Kong, China. Association for Computational Linguistics.

Mateusz Krubiński, Hashem Sellat, Shadi Saleh, Adam Pospíšil, Petr Zemánek, and Pavel Pecina. 2023. Multi-parallel corpus of north levantine arabic. In *Proceedings of ArabicNLP 2023*, pages 411–417.

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.

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. *OSACT*, 3:49.

Omer Nacar, Abdullah Alharbi, Serry Sibaee, Samar Ahmed, Lahouari Ghouti, and Anis Koubaa. 2024. Asos at osact6 shared task: Investigation of data augmentation in arabic dialect-msa translation. In *Proceedings of the 6th Workshop on Open-Source Arabic Corpora and Processing Tools, LREC'2024*.

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 Association for Computational Linguistics*, pages 311–318.

Ricardo Rei, José G. C. de Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André F. T. Martins. 2022. COMET-22: Unbabel-IST 2022 submission for the metrics shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 578–585, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Lrec*, volume 2012, pages 2214–2218. Citeseer.

OSACT 2024 Task 2: Arabic Dialect to MSA Translation

Hanin Atwany[1], **Nour Rabih**[1], **Ibrahim Mohammed**[1], **Abdul Waheed**[1], **Bhiksha Raj**[1,2]
[1]Mohamed bin Zayed University of Artificial Intelligence,
[2]Carnegie Mellon University
{hanin.atwany, nour.rabih, ibrahim.mohammed,abdul.waheed,bhiksha.raj}@mbzuai.ac.ae

Abstract

We present the results of Shared Task "Dialect to MSA Translation", which tackles challenges posed by the diverse Arabic dialects in machine translation. Covering Gulf, Egyptian, Levantine, Iraqi and Maghrebi dialects, the task offers 1001 sentences in both MSA and dialects for fine-tuning, alongside 1888 blind test sentences. Leveraging GPT3.5, a state-of-the-art language model, our method achieved a BLEU score of 29.61. This endeavor holds significant implications for Neural Machine Translation (NMT) systems targeting low-resource languages with linguistic variation. Additionally, negative experiments involving fine-tuning AraT5 and No Language Left Behind (NLLB) using the MADAR Dataset resulted in BLEU scores of 10.41 and 11.96, respectively. Future directions include expanding the dataset to incorporate more Arabic dialects and exploring alternative NMT architectures to further enhance translation capabilities.

1. Introduction

Arabic, a language spoken by over 420 million people globally, boasts a rich tapestry of dialectal variations. This linguistic landscape comprises both Modern Standard Arabic (MSA), the formal variant employed in official domains such as government communications, national media, and education, and a myriad of regional dialects used predominantly in everyday interactions (Harrat et al., 2017). The differences between these dialects, which range from subtly distinct to completely unintelligible (Abdul-Mageed et al., 2022), pose a formidable challenge for machine translation systems.

Historically, the focus of machine translation systems has been predominantly on MSA. This causes those systems to struggle to capture the intricate differences inherent in dialects. Consequently, achieving accurate translation between these linguistic variants remains paramount. Addressing this challenge is crucial to enhance communication and comprehension within the Arabic-speaking world.

In the field of Natural Language Processing (NLP), dialect identification and translation are two critical areas of research. This paper concentrates on the latter, specifically examining the performance of various models in translating sentences from diverse Arabic dialects into MSA. This investigation is set in the context of the second shared task at The 6th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT6), which aims to address the complexities of dialect translation.

In particular, we examined the performance of three distinct methods. Firstly, we fine-tuned the AraT5 transformer model (Nagoudi et al., 2022) using diverse corpora sourced from MADAR (Bouamor et al., 2018a). Secondly, we explored the inference capabilities of the NLLB model (Costa-jussà et al., 2022). Lastly, we employed a prompting technique with GPT3.5 to facilitate dialect-to-MSA translation. By comparing these three methods, we aim to evaluate the strengths and limitations of each approach and identify the most effective solution for dialect-to-MSA translation. Our investigation provides valuable insights into the challenges of dialect translation and highlights the potential of state-of-the-art language models in addressing these challenges.

Task 2: Dialect to MSA Machine Translation
The objective of this task is to develop a model that converts Arabic from five (Gulf, Egyptian, Levantine, Iraqi, and Maghrebi) dialects to MSA. Participants can use any resources available to develop their systems.

2. Related Work

Over the past decade, advancements in the field of dialect to Modern Standard Arabic translation have been notable, driven by the imperative to foster communication and comprehension across varied Arabic dialects and the standardized form of the language (Mohamed et al., 2024). Despite these strides, challenges persist in achieving high-quality translations (Abdelali et al., 2024).

A study by (Al-Sabbagh, 2024) scrutinized the performance of Google Translate in translating Egyptian Arabic adjuncts, revealing low BLEU scores and various issues, including literal translations of idiomatic adjuncts and misinterpretation of dialectal adjuncts.

In addressing the translation challenges within NMT systems for Arabic dialects, (Moukafih et al.,

OSACT 2024 Workshop, pages 98–103

2021) investigated multitasking learning strategies, yielding noteworthy enhancements in BLEU scores for Algerian Modern Standard Arabic and Moroccan Palestinian dialects.

Recent efforts have focused on developing models that deal with translating single dialects to MSA. For instance, (Sghaier and Zrigui, 2020) proposed a rule-based machine translation system for translating Tunisian dialect to MSA, achieving a BLEU score of 55.22. Furthermore, (Sallam and Mousa, 2024) assessed the performance of AI chatbot ChatGPT in responding to health queries in Tunisian and Jordanian Arabic dialects. Their study revealed that GPT-4 exhibits slightly better performance than ChatGPT[1], with above-average scores in Jordanian Arabic but average scores in Tunisian Arabic. However, responses in both dialects fell significantly short compared to English, emphasizing the importance of linguistic and cultural diversity in AI model development, particularly in healthcare.

A comprehensive evaluation conducted by (Kadaoui et al., 2023) assessed Bard and ChatGPT for machine translation across ten Arabic varieties, encompassing Classical Arabic (CA), MSA, and country-level dialectal variants. Their findings indicated that Large Language Models (LLMs) may face challenges with dialects possessing minimal public datasets but generally outperform existing commercial systems in dialect translation. However, instruction-tuned LLMs still trail behind commercial systems like Google Translate in CA and MSA translation. Their human-centric study also underscored Bard's limited ability to adhere to human instructions in translation contexts.

In conclusion, these studies underscore the necessity for continued research and development aimed at enhancing the linguistic inclusivity of LLMs and addressing the distinctive hurdles associated with translating diverse dialects to MSA.

3. Data

3.1. Shared Task Data

We conducted thorough evaluations on both the validation and test sets provided for this shared task.

3.1.1. Validation Dataset

The validation dataset provided in this shared task, comprised a total of 1001 source-to-target examples, evenly distributed among dialects as follows: 200 Egyptian, 200 Maghrebi, 200 Levantine, 201 Gulf, and 200 Iraqi examples.

Notably, some examples featured Arabised text, where English words were transcribed using Arabic letters, as demonstrated below:

{ "id": 419221,
"dialect": "Iraqi",
"source": "يس آي لاف يو تو ماتش تعالي وين نص دينار",
"target": "نعم أنا أحبك كثيرًا تعالي أين نصف دينار"
"English translation": "Yes, I love you too much, come where is half a Dinar"

In this instance, the source sentence incorporates English phrases represented in Arabic script, while the corresponding target sentence reflects the translation into Modern Standard Arabic. Such instances posed unique challenges during evaluation and were included in the validation dataset to assess translation quality comprehensively.

Moreover, the validation dataset includes 22 sentences with a length greater than 128 characters, further enriching the evaluation process and highlighting the model's ability to handle complex linguistic structures

3.1.2. Test Datasets

The test dataset, comprised 1888 examples, each presenting its own unique linguistic challenge. These examples were distributed across different dialects as follows: 314 Egyptian, 343 Maghrebi, 568 Levantine, 77 Iraqi, and 586 Gulf.

The source sentences provided cover a broad spectrum of topics and linguistic structures, reflecting the rich diversity of Arabic dialects. They encompass both everyday conversational phrases and more formal expressions, offering a comprehensive representation of language usage in real-world scenarios.

Among these sentences, 45 exceed a length of 128 characters, presenting additional complexity to the translation task. Furthermore, the dataset includes instances of words with repeated characters, as exemplified by:

{ "مسااااء النور والسرووور وحنا نوحشنااااك بزاااااف عمري" }

Despite these challenges, the diversity in content and language enriches the dataset, enabling a thorough evaluation of the model's proficiency in handling various linguistic features and contexts.

3.2. Finetuning Dataset

The MADAR Arabic Dialect Corpus and Lexicon (Bouamor et al., 2018a), utilized in our study to fine-tune the models, represents a comprehensive resource designed to facilitate research in machine translation, particularly focusing on the translation challenges presented by Arabic dialects. The dataset consists of 25 parallel translations for 25 cities having 2,000 sentences each, in addition to their MSA equivalents and is divided into training, development, and test sets. This dataset is

[1]We refer GPT-3.5 as ChatGPT in our work.

Dialect Region	Cities Included
Egyptian	Cairo, Alexandria
Gulf	Doha, Jeddah, Muscat, Riyadh
Iraqi	Baghdad, Basra, Mosul
Levantine	Aleppo, Amman, Beirut, Damascus, Jerusalem, Salt
Maghreb	Algiers, Fes, Rabat, Sfax, Tunis

Table 1: MADAR Dataset Dialect Divisions

instrumental in understanding the linguistic diversity across the Arabic-speaking world, featuring a collection of text samples from a wide array of cities, each with its unique dialectical characteristics. For the purpose of our experiments, the dataset was meticulously organized into five distinct groups (as specified by the task), each representing a major geographical and dialectal region within the Arab world. This division was used in dialect specific finetuning.

4. Methodology

4.1. Supervised Models

NLLB. NLLB model is designed to bridge language gaps by extending translation support to a wide array of languages, with a particular focus on those with limited resources. It employs an innovative conditional compute model based on the Sparsely Gated Mixture of Experts framework, along with curated datasets and training techniques tailored for low-resource languages. In our assessment, we evaluated the NLLB 3.3B model in two scenarios: with fine-tuning on the development dataset and without fine-tuning on the test dataset.

Supervised NLLB. We finetuned NLLB 3.3B. Utilizing the MADAR Parallel Corpus Dataset, which contains data from various Arabic dialects translated into Modern Standard Arabic (MSA) (Bouamor et al., 2018a).

AraT5. AraT5 is a state-of-the-art language model specifically designed for understanding and generating Arabic text. Building upon the T5 (Text-to-Text Transfer Transformer) architecture (Raffel et al., 2023), which treats every text-based task as a "text-to-text" problem, AraT5 is fine-tuned to excel in processing and generating Arabic content across a wide range of tasks. These include text summarization, question answering, text classification, and translation. The model has been trained on a diverse corpus of Arabic text, enabling it to grasp the nuances of the language, including its dialects and classical forms.

Supervised AraT5. In the initial phase of our experiments, the model was deployed for translation tasks without any prior fine-tuning. This approach, however, did not yield successful outcomes in generating translations, primarily attributable to the constraints of the model's training. Specifically, the model was architected to facilitate machine translation from dialectal Arabic to English, with no inherent training to support translation from various dialects into Modern Standard Arabic (Nagoudi et al., 2022).

To address this, a subsequent stage of fine-tuning was implemented, utilizing the MADAR dataset as a foundational corpus. This dataset was anticipated to enhance the model's dialectal comprehension and translation efficacy. However, the results fell short of expectations, which revealed a lower than anticipated BLEU score.

AraT5-finetuned dialect-specific. Recognizing the need for a more tailored approach to capture the characteristics of each Arabic dialect, the models were fine-tuned separately for each specific dialect contained in this task. This refined strategy was predicated on the hypothesis that dialect-specific fine-tuning would enable the model to more accurately learn and replicate the unique linguistic features and idiomatic expressions inherent to each dialect. This method was designed to fix the early problems the model had when trying to translate in a general way. By doing this, we hoped to make the translations better overall and get higher scores on translation quality tests (BLEU scores).

4.2. Zero-Shot Models

We evaluate GPT3.5 (Ouyang et al., 2022) extensively to translate various dialects into modern standard Arabic. Especially, we evaluate GPT3.5 in zero and few-shot settings. We choose three examples in the few-shot setting as (Kadaoui et al., 2023) show it as the optimal setting across a wide range of Arabic to English translation tasks. We provide more details about our prompt in Table 2.

Zero-shot. We evaluate GPT3.5 in a zero-shot setting with a simple prompt asking the model to translate dialectal Arabic into MSA. We provide the zero-shot prompt template in Table 2.

Few-Shot. We also use GPT3.5 in the 3-shot setting by providing three examples from each dialect. We keep the example static throughout the dialect. Our 3-shot prompt can be found in Table 2.

Few-Shot with Self-Correction. We find that despite providing examples there seem to be issues with the translation. To address this issue, we experiment with a modified prompt that asks the model to find its mistakes and correct itself. We provide a step-by-step guide to do the task. Our refinement process improves our score by approximately

Shot	Prompt
Zero-shot	Translate the given input text from {dialect} Arabic dialect into Modern Standard Arabic (MSA). {dialect}:{input} MSA: []
Few-Shot	Translate the following input text from {dialect} Arabic dialect into the Modern Standard Arabic (MSA). The output should be in Arabic script only. Here are some examples: {examples} {dialect}:{input} MSA: []
Few-Shot with Self-Correction	Following is the Modern Standard Arabic (MSA) translation from {dialect} Arabic. {dialect}: {input} MSA: {msa} Please correct the MSA translation for the input in {dialect}. An accurate translation should consist solely of Modern Standard Arabic (MSA) words and accurately translate the given input. Here are some examples: {examples} Here is a step-by-step guide to do the task: 1. Identify any mistakes in the translation. 2. Correct the mistakes by replacing them with the correct MSA words or phrases. 3. Provide the final corrected MSA translation. Generate only the corrected MSA translation; no additional information is needed. If no changes are required, then produce the same translation. {dialect}: {input} Corrected MSA: []

Table 2: Zero-shot, few-shot, and self-correcting prompt templates. We format the prompt with appropriate input and examples before feeding it to ChatGPT.

2 points in terms of BLEU score. We report our self-correction prompt in Table 2.

4.3. Experimental Setup

We initially explored the efficacy of zero-shot prompting for Arabic dialect-to-Modern Standard Arabic translation tasks. While zero-shot prompting of GPT3.5 provided a solid baseline, we further investigated the impact of increasing the prompt complexity through a three-shot prompting approach. Remarkably, our experiments revealed a substantial improvement in BLEU scores when transitioning from zero-shot to three-shot prompting. By incorporating additional context and refining the prompts, the model gained a deeper understanding of the translation task, resulting in more accurate and fluent translations.

5. Results

BLEU score obtained using several models is recorded in Table 3. Our results show that GPT3.5 outperformed the other models in dialectal Arabic to MSA translation, with a BLEU score of 29.61.

The NLLB 3.3B Base model achieved a BLEU score of 11.96. However, the fine-tuned NLLB yielded a BLEU score lower than that of the base NLLB model without fine-tuning of 9.00.

There could be several reasons for this unexpected result:

- Heterogeneous dataset: Fine-tuning the NLLB model on the entire dataset while specifying the source language as "arb_Arab" is inaccurate, considering the dialectal variations within the MADARA dataset.

  The MADAR dataset is diverse, comprising data from multiple Arabic dialects, which may have contributed to a decline in performance

owing to the substantial differences among each dialect.

- Lack of dialect-specific fine-tuning: The fine-tuning process did not involve separate fine-tuning for each dialect. This could have led to the model being unable to learn the specific characteristics of each dialect, resulting in a lower BLEU score.

On the other hand, the AraT5 fine-tuned model achieved a BLEU score of 9.41 across all dialects. However, when fine-tuned specifically for each dialect, there was a notable improvement, with a BLEU score of 10.41.

These results suggest that GPT3.5 is more effective in capturing the features of dialectal Arabic and translating them into MSA compared to the other models.

The lower BLEU scores for the NLLB 3.3B Base and AraT5 finetuned models may be due to the complexity and variability of dialectal Arabic, which can make it challenging to generalize from the training data.

The highest BLEU scores were achieved through iterative improvements to the prompting strategy applied to GPT-3.5. Initially, the model's performance was enhanced by incorporating examples of dialect-to-Modern Standard Arabic translations into the prompt, resulting in a BLEU score of 28. Subsequently, further refinement was achieved by integrating step-by-step instructions for self-correction within the prompt framework. This iterative approach culminated in the attainment of the highest BLEU score on the test dataset, reaching 29.61.

Model	BLEU
NLLB-3.3B finetuned	9.00
AraT5 finetuned	9.41
AraT5-finetuned dialect-specific	10.41
NLLB-3.3B	11.96
ChatGPT (0-shot)	21.84
ChatGPT (3-shot)	28.00
ChatGPT (3-shot) with self-Correction	29.61

Table 3: BLEU score on the *Test* dataset.

6. Conclusion

Our experiments highlight the challenges in dialectal Arabic to MSA translation, particularly in dealing with heterogeneous datasets and the importance of dialect-specific fine-tuning. Our results also demonstrate the potential of using state-of-the-art language models like GPT to improve translation performance. Future work could involve exploring different fine-tuning strategies such as the mixture of experts to improve the BLEU score further.

7. Bibliographical References

Ahmed Abdelali, Hamdy Mubarak, Shammur Chowdhury, Maram Hasanain, Basel Mousi, Sabri Boughorbel, Samir Abdaljalil, Yassine El Kheir, Daniel Izham, Fahim Dalvi, et al. 2024. Larabench: Benchmarking arabic ai with large language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 487–520.

Muhammad Abdul-Mageed, Chiyu Zhang, Abdel-Rahim Elmadany, Houda Bouamor, and Nizar Habash. 2022. NADI 2022: The third nuanced Arabic dialect identification shared task. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 85–97, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Rania Al-Sabbagh. 2024. The negative transfer effect on the neural machine translation of egyptian arabic adjuncts into english: The case of google translate. *International Journal of Arabic-English Studies*, 24(1):95–118.

Houda Bouamor, Nizar Habash, Mohammad Salameh, Wajdi Zaghouani, Owen Rambow, Dana Abdulrahim, Ossama Obeid, Salam Khalifa, Fadhl Eryani, Alexander Erdmann, and Kemal Oflazer. 2018a. 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, Nizar Habash, Mohammad Salameh, Wajdi Zaghouani, Owen Rambow, Dana Abdulrahim, Ossama Obeid, Salam Khalifa, Fadhl Eryani, Alexander Erdmann, et al. 2018b. The madar arabic dialect corpus and lexicon. In *Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018)*.

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.

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*.

Salima Harrat, Karima Meftouh, and Kamel Smaïli. 2017. Machine translation for arabic dialects (survey). *Information Processing Management*, 56:262–273.

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*.

Yasir Abdelgadir Mohamed, Akbar Khanan, Mohamed Bashir, Abdul Hakim HM Mohamed, Mousab AE Adiel, and Muawia A Elsadig. 2024. The impact of artificial intelligence on language translation: A review. *IEEE Access*, 12:25553–25579.

Youness Moukafih, Nada Sbihi, Mounir Ghogho, and Kamel Smaïli. 2021. Improving machine translation of arabic dialects through multi-task learning. In *International Conference of the Italian Association for Artificial Intelligence*, pages 580–590. Springer.

El Moatez Billah Nagoudi, AbdelRahim Elmadany, and Muhammad Abdul-Mageed. 2022. Arat5: Text-to-text transformers for arabic language generation.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer.

Malik Sallam and Dhia Mousa. 2024. Evaluating chatgpt performance in arabic dialects: A comparative study showing defects in responding to jordanian and tunisian general health prompts. *Mesopotamian Journal of Artificial Intelligence in Healthcare*, 2024:1–7.

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.

ASOS at OSACT6 Shared Task: Investigation of Data Augmentation in Arabic Dialect-MSA Translation

Omer Nacar, Serry Sibaee, Samar Ahmed,
Abdullah I. Alharbi, Lahouri Ghouti, Anis Koubaa
Robotics and Internet-of-Things Lab, Prince Sultan University
Faculty of Computing and Information Technology Rabigh, King Abdulaziz University
Riyadh 12435 Saudi Arabia, Jeddah 22254 Saudi Arabia
{onajar, ssibaee, lghouti, akoubaa}@psu.edu.sa
aamalharbe@kau.edu.sa, Samar.sass6@gmail.com

Abstract

The translation between Modern Standard Arabic (MSA) and the various Arabic dialects presents unique challenges due to the significant linguistic, cultural, and contextual variations across the regions where Arabic is spoken. This paper presents a system description of our participation in the OSACT 2024 Dialect to MSA Translation Shared Task. We explain our comprehensive approach, which combines data augmentation techniques using generative pre-trained transformer models (GPT-3.5 and GPT-4) with the fine-tuning of AraT5 V2, a model specifically designed for Arabic translation tasks. Our methodology has significantly expanded the training dataset, thus improving the model's performance across five major Arabic dialects, namely Gulf, Egyptian, Levantine, Iraqi, and Maghrebi. We have rigorously evaluated our approach, using the BLEU score, to ensure translation accuracy, fluency, and the preservation of meaning. Our results demonstrate the effectiveness of our refined data and models, achieving a BLEU score of 85.5% on the validation set and 22.6% on the blind test set, indicating a successful bridging of the gap between different dialects. However, it's important to note that while utilizing a larger dataset resulted in significantly higher evaluation BLEU scores, the performance on the blind test set was relatively lower. This observation underscores the importance of dataset size in model training, revealing potential limitations in generalization to unseen data due to variations in data distribution and domain mismatches.

Keywords: Machine Translation, Data Augmentation, BLEU Score, Arabic Dialects

1. Introduction

The Arabic language, characterized by its rich diversity of dialects, is the primary mode of communication for over 420 million individuals across the Middle East and North Africa. This linguistic landscape is distinguished by a phenomenon known as diglossia, wherein Modern Standard Arabic (MSA) coexists with various regional dialects (Qudah et al., 2017). As the formal variant, MSA is ubiquitously employed in official discourse, educational frameworks, and literary works across the Arab domain. Conversely, dialectal Arabic (DA) encompasses the myriad vernacular languages intrinsically linked to specific regions, encapsulating the essence of local identities and cultural intricacies.

The coexistence of MSA and DA within this linguistic ecosystem poses substantial challenges for machine translation. The pronounced variations in dialectal expressions, coupled with the scarcity of extensive parallel corpora essential for practical training, often culminate in suboptimal translation outputs when conventional models, predominantly trained on MSA, are utilized for DA content. This predicament underscores the critical need for translation methodologies tailored to accommodate the unique attributes of DA, enhancing accuracy and contextual relevance in this linguistically complex environment (Darwish et al., 2021).

In seeking to address the aforementioned challenges, we participated in the OSACT 2024 Dialect to MSA Translation Shared Task, which aims to evaluate the performance of translation models across five major Arabic dialects: Gulf, Egyptian, Levantine, Iraqi, and Maghrebi. The primary objective of this work is to test the efficacy of sequence-to-sequence translation models, particularly those using the Text-to-Text Transfer Transformer (T5) framework, in translating DA into MSA (Raffel et al., 2020).

Our participation in this shared task entailed pre-training specific models and carefully enhancing the training data for the above dialects. We used the dataset provided by the shared-task organizers for the training phase, after which our models were evaluated using the development and test sets (consisting of 500 unseen sentences for each dialect during the test phase). We conducted several experiments to evaluate the performance of the models comprehensively (Nagoudi et al., 2022); we also implemented different training settings to improve the results and accuracy of the translation between DA and MSA.

The subsequent sections are structured as follows: Section 2 reviews prior studies, Section 3 describes our proposed method, Section 4 details

OSACT 2024 Workshop, pages 104–111

our experimental result, and, finally, we conclude with a summarization of our main findings.

2. Related Works

Given the increasing need for effective communication across diverse cultures and global borders, it has become essential to establish systems that tackle the challenges of multiple dialects. However, ensuring precise and efficient translations has become increasingly complex. Therefore, our goal is to explore a variety of approaches to improve the effectiveness of translation systems, specifically for MSA and Arabic dialects.

Sghaier and Zrigui (2020) propose a machine translation system designed to translate Tunisian Dialect (TD) text into MSA through a rule-based methodology. The translation process comprises three key stages: morphological analysis and disambiguation, lexical and structural transfer, and morphological generation with spelling corrections, resulting in the output text in MSA. Sajjad et al. (2020) present a benchmarking effort for dialectal Arabic-English machine translation aimed at tackling the challenges encountered in low-resource machine translation, particularly concerning Arabic dialects. It introduces an evaluation suite designed as a standard for measuring the effectiveness of Arabic-English machine translation systems specialized in dialectal Arabic. By combining existing Arabic-English dialectal resources and generating new test sets, it provides a comprehensive evaluation framework, covering various dialect categories, genres, and levels of dialectal diversity. The study employs a transformer-based seq2seq model for this purpose.

Al-Ibrahim and Duwairi (2020) delves into the application of Neural Machine Translation (NMT) for translating the Jordanian dialect into MSA using deep learning techniques, specifically the RNN encoder-decoder model. The RNN encoder-decoder model proves to be effective in translating the Jordanian dialect into MSA, achieving a high accuracy rate for word-to-word translation and a lower accuracy rate for sentence translation. Additionally, Convolutional Neural Networks (CNN) are utilized to enhance translation accuracy. Moreover, the study (Moukafih et al., 2021) addresses the challenges of machine translation for six Arabic dialects: Tunisian, Algerian, Moroccan, Syrian, and Palestinian. It introduces the PADIC dataset, a parallel corpus of Arabic dialects and MSA. It presents a neural multi-task learning framework leveraging inter-dialectal relationships to achieve superior translation results.

Furthermore, Alzamzami and Saddik address challenges in translating Arabic dialects on social media by introducing a multi-dialectal Arabic-English dataset. It details the dataset construction process, emphasizing meticulous translator selection and cultural considerations. Additionally, it highlights deep learning-based translation models for four Arabic dialects, utilizing transfer learning and Transformer architecture for improved accuracy. The proposed dataset and models aim to address the limitations in current translation systems for Arabic dialects, particularly in informal social media contexts, spotlighting deep learning-powered translation models tailored for four distinct Arabic dialects: Gulf, Levantine (Shami), Iraqi, and Yemeni.

3. Methodology

In this section, we present a comprehensive approach for tackling the shared issue of translating different Arabic dialects into Modern Standard Arabic (MSA). Considering the wide range of linguistic variations among Arabic-speaking areas, our approach aims to improve translation models for precision and fluency while also bridging the gap between formal written Arabic and informal spoken Arabic. In order to do this, we have used a blend of sophisticated data augmentation methods and processes for fine-tuning that are especially suited to the distinctive qualities of the Arabic dialects—Gulf, Egyptian, Levantine, Iraqi, and Maghrebi. Our method improves the accuracy and consistency of dialect-to-MSA translation by utilizing the most recent developments in machine translation technology, such as the use of generative pre-trained transformer models.

3.1. Data Augmentation

A key component of our approach is data augmentation, which aims to significantly expand the variety and amount of training data available for optimizing our translation models (Shorten et al., 2021). The model's capacity to generalize across many dialects and linguistic subtleties, as well as the lack of sufficient training data, are major obstacles that must be overcome in order to successfully complete machine translation tasks.

3.1.1. Implementation of Data Augmentation

To implement our data augmentation strategy, we utilized a novel approach by incorporating the capabilities of generative pre-trained transformer models, specifically GPT-3.5 and GPT-4 models. These models were tasked with generating additional training examples from the original set of 200 sentences provided for each dialect. The augmentation process involved the following steps:

Source Sentence Preparation: For each source sentence in the provided dialectal Arabic datasets, we prepared a prompt designed to guide

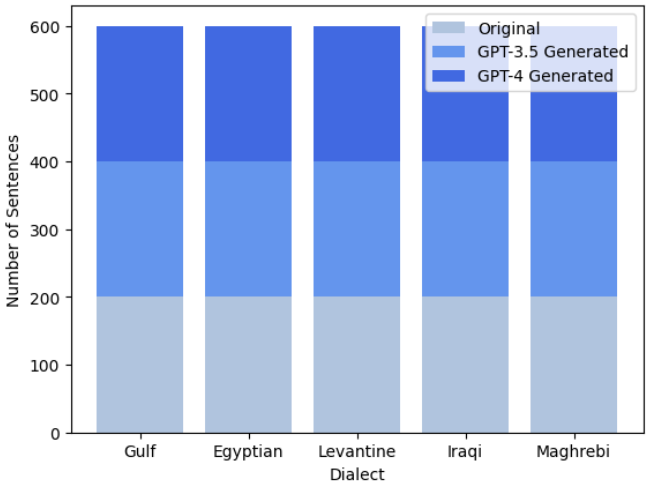


Figure 1: Dataset Size Before and After Augmentation by Dialect

the generative model towards producing a synonymous translation in MSA. The prompt explicitly instructed the model to ensure that the translation maintains the original sentence's meaning, adheres to Modern Standard Arabic grammar, and matches the original sentence in word count as closely as possible.

Model Interaction: We interacted with the GPT-3.5 and GPT-4 models through the OpenAI API [1], submitting each prepared prompt as input. The models were prefaced with a system message that outlined their role as language models trained for translating dialectal Arabic to MSA, emphasizing the need for accuracy, grammatical adherence, and word count maintenance.

Translation Generation: Upon receiving each prompt, the models generated translations that were then evaluated for quality and adherence to the specified criteria. This process allowed us to significantly expand our dataset with high-quality, model-generated translations, thereby enriching the training material available for fine-tuning our translation system. Figure 1. illustrates the dataset size before and after augmentation for each Arabic dialect.

3.1.2. Evaluation of Augmented Data

In assessing the quality of sentences generated by GPT models, traditional and advanced metrics provide insights into the linguistic and semantic fidelity of the output compared to target sentences. This evaluation highlights the challenges and solutions in quantifying the effectiveness of generative models in language tasks.

BLEU's Limitations in Sentence Evaluation, the Bilingual Evaluation Understudy (BLEU) metric, widely utilized in machine translation to measure the similarity of generated text to reference translations, showed significant limitations in our context of the evaluation step where BLEU evaluates the correspondence of n-grams between the generated and target texts, offering a score from 0 to 1. However, this method's reliance on exact matches often fails to capture the essence of semantic similarity and sentence structure, particularly in languages with rich morphology or when dealing with nuanced textual differences. A notable example from our dataset noted during evaluation illustrates this limitation in Figure 2,

As shown figure 2, Despite the generated sentence being semantically identical to the original target, except for the addition of a question mark, BLEU assigned a score of 0, demonstrating its inefficacy in capturing semantic equivalence and punctuation nuances.

Advantages of METEOR in Overcoming BLEU's Shortcomings, on the other hand and due to BLEU score sensitivity, the metric for evaluation of GPT models predictions are underscored with Explicit Ordering (METEOR) which offers a more nuanced evaluation by accounting for synonymy and stemming, in addition to exact matches. METEOR's alignment-based approach, which allows for a flexible matching of words and phrases, provides a more comprehensive assessment of similarity between the generated text and the target. Employing METEOR in our evaluation of GPT generated sentences yielded scores that more accurately reflected the semantic and syntactic correspondence between the target and GPT4 generated sentences as shown in Figure 3.

The average METEOR score across GPT4 and GPT3.5 augmented dataset are 73.22% and 67.48% respectively, indicating a strong alignment with the original ground truth MSA target sentences

[1] https://openai.com/blog/openai-api

Original Target	Generated Target	BLEU Score
كيف تتعلم How do you learn	كيف تتعلم؟ How do you learn?	0.00

Figure 2: BLEU Score Evaluation Demonstrating Sensitivity to Punctuation.

Original Target	Generated Target	METEOR Score
كيف تتعلم How do you learn	كيف تتعلم؟ How do you learn?	63.92
هل تعتقدين أننا سنصبح مثلهم في يوم من الأيام؟ Do you think we will be like them one day?	تعتقدين أننا سنكون مثلهم في يوم من الأيام؟ Do you think we will become like them one day?	81.17
نعم، لا يصدق Yes, unbelievable	أه لا يصدق Oh, unbelievable	72.31
أمي قالت لي لا بأس My mother told me it was okay	أمي قالت لي عفواً My mother said excuse me	45.36

Figure 3: METEOR Scores for Evalation of GPT-Enhanced Data

of similarity and the ability of METEOR to capture nuanced linguistic features.

Qualitative Evaluation with GPT-4 , In addition to quantitative metrics, we employed GPT-4 for a qualitative evaluation of sentence similarity. Using a custom prompt, sentences were assessed on a scale from 1 to 5, with 5 indicating identical semantic content. This approach allowed us to incorporate contextual understanding and nuanced judgment beyond the capability of automated metrics. Selected examples from our evaluation of GPT4 generated sentences are shown in Figure 4.

The average similarity score across evaluated pairs for GPT4 and Gpt3.5 are 4.59 and 4.43 respectively, demonstrating the efficacy of GPT-4 in understanding and evaluating semantic nuances.

Through evaluating GPT-3.5 and GPT-4 generated sentences, we harnessed their high-quality outputs for data augmentation, significantly boosting the AraT5 V2 machine translation performance from dialect to MSA. This approach effectively enriched our training dataset, showcasing the value of leveraging advanced language models in enhancing machine translation tasks.

3.2. Fine-Tuning AraT5-V2 for Enhanced Performance

Following the strategic data augmentation outlined in the previous section, we transition to the fine-tuning of AraT5 V2, a process central to our methodology aimed at enhancing Arabic dialect to MSA translation. AraT5 V2, the successor to the foundational AraT5 model, embodies a series of substantial upgrades that elevate its capabilities in Arabic language translation tasks significantly.

AraT5 (Nagoudi et al., 2022) is based on the same architectural foundation as the original T5 model, but trained solely on Arabic data comprising both MSA and dialectal Arabic (tweets) resulting in 29 Billion token with more than 248 GigaBytes of dataset. The most recent version of AraT5, AraT5 V2 was utilized in this work. A key improvement in AraT5 V2 lies in its training across a broader and more diverse Arabic data corpus. AraT5 V2 enhances the model's sequence length capability from 512 to 1024 tokens, doubling its capacity for handling longer text passages, ensuring context preservation and resulting in more accurate and coherent translations.

In order to evaluate the effectiveness of this model in our paper, we compare AraT5 V2 against different sequence to sequence machine translation models, including the ARaT5-base (Nagoudi et al., 2022), mT5 (Xue et al., 2020) models, to showcase the efficacy of AraT5 V2 in translating dialectal Arabic to Modern Standard Arabic (MSA). This benchmarking underscores why AraT5 V2 was the optimal choice for our study, highlighting its superior performance over the augmented dataset and specific advantages in addressing the complexities of dialect-to-MSA translation tasks. Table 1 illustrates the comparative analysis showing the validation loss and BLEU under the same training

Original Target	Generated Target	Similarity Score
كيف ذلك؟ How is that?	ماذا يعني ذلك؟ How is that?	3.5
كيف صحتك؟ How is your health?	كيف الصحة How is health	4.75
نعم، أعرفها Yes, I know her	أجل، أعرفها Yes, I know her	5

Figure 4: GPT4 Sentence Similarity Evaluation, Highlighting Semantic Alignment.

Model name	Validation loss	Validation BLEU
AraT5 V2	2.523	**0.255**
mt5	1.932	0.174
AraT5 Base	3.441	0.113

Table 1: Validation Loss and BLEU Scores for AraT5 V2, mt5, and AraT5 Base

configuration of all models.

As shown in Table 1, three models were evaluated based on their validation loss and BLEU scores: AraT5 V2, mT5, and AraT5 Base. AraT5 V2 demonstrated a compelling balance of performance metrics, recording a validation loss of 2.523 and a BLEU score of 0.255. Although mT5 presented a lower validation loss at 1.932, its BLEU score of 0.174 was notably inferior to that of AraT5 V2, indicating less effective translation quality. AraT5 Base, Although a key model, AraT5 Base had the highest validation loss of 3.441 and the lowest BLEU score of 0.113, putting it behind the others. These results clearly support chosen AraT5 V2 for our experiment, not only due to its superior BLEU score, which maintains a satisfactory balance between loss and translation quality, proving its possibility in handling the translation of dialect-to-MSA.

3.3. Training Configuration

The fine-tuning of AraT5 V2 is done by using two NVIDIA A100 GPUs for efficient large-scale machine learning tasks. The model was based on the $UBC - NLP/AraT5v2 - base - 1024$ model from hugging face, which is specifically designed for Arabic language tasks. The training used 128 tokens for source and target texts, a per-device batch size of 16, and 22 epochs to adapt the model without overfitting. The learning rate was 5e-5, using the AdamW optimizer, reflecting best practices in transformer-based models for NLP tasks.

Training was conducted on a dataset comprising 2,666 examples, with a validation set of 297 examples, ensuring the model's performance was evaluated. The dataset was split from a larger corpus, incorporating a diverse range of Arabic dialects and ensuring a comprehensive representation of linguistic nuances.

The model's performance was primarily evaluated using the $BLEU$ score, a widely recognized metric in machine translation that assesses the correspondence between the model's output and the target translations. This metric, coupled with our dataset, provided a robust framework for assessing translation quality and model effectiveness.

The AraT5 V2 model have been tested a thorough evaluation on a test set of 500 blind sentences after its training and fine-tuning phases, as part of the OSACT 2024 shared task. These sentences, representing a broad spectrum of Arabic dialects, provided a robust benchmark for testing the model's translation abilities. The evaluation, conducted blindly by the shared task organizers, primarily utilized the BLEU score to assess translation quality, focusing on accuracy, fluency, and meaning preservation.

The AraT5 V2 model's performance was comprehensively assessed through supplementary experiments, including augmenting the training dataset with dialectical variations like MADAR and evaluating its performance on synthetically generated datasets generated by GPT4 without fine-tuning, contributing to a comprehensive assessment of its efficacy across various real-world translation scenarios.

4. Evaluation and Results

Our experiments spanned a range of scenarios, each designed to evaluate different factors of model behavior and performance. We explore the impact of dataset size, data augmentation techniques, and fine-tuning strategies on model performance, lever-

Experiment ID	Experiment Type	Training Dataset	Dataset Size	Number of Steps	Val loss	Val BLEU
1	Dev Only FT	Dev Only	1k	5k	3.567	0.234
2	Dev Only FT	Dev Only	1k	10k	4.526	0.254
3	Madar + Dev FT	Madar and Dev	80k	85k	0.194	**0.855**
4	GPT4 Generated	Test Dataset	1k	-	-	-
5	Augmented Data with GPT4 + Dev FT	Dev + GPT generated	2k	4.5k	2.228	0.248
6	Augmented Data with GPT4 + Dev FT	Dev + GPT generated	2k	6k	2.523	0.255
7	Augmented Data with GPT3.5 + GPT4 + Dev FT	Dev + GPT generated	3k	2k	1.658	0.241
8	Augmented Data with GPT3.5 + GPT4 + Dev FT	Dev + GPT generated	3k	4k	1.732	0.237

Table 2: Summary of Experiments Results - Evaluation Metrics

Experiment ID	Experiment Type	Training Dataset	Dataset Size	Number of Steps	Test BLEU
1	Dev Only FT	Dev Only	1k	5k	0.215
2	Dev Only FT	Dev Only	1k	10k	0.215
3	Madar + Dev FT	Madar and Dev	80k	85k	0.172
4	GPT4 Generated	Test Dataset	1k	-	0.171
5	Augmented Data with GPT4 + Dev FT	Dev + GPT generated	2k	4.5k	0.222
6	Augmented Data with GPT4 + Dev FT	Dev + GPT generated	2k	6k	**0.226**
7	Augmented Data with GPT3.5 + GPT4 + Dev FT	Dev + GPT generated	3k	2k	0.205
8	Augmented Data with GPT3.5 + GPT4 + Dev FT	Dev + GPT generated	3k	4k	0.208

Table 3: Summary of Experiments Results - Test Metrics

aging both synthetic and real-world data sources. Additionally, we provide an error analysis framework to further understand the predictions and their limitations. All these experiment results are chosen based on the best epoch results of both validation loss and BLUE and they are fully summarized in Table 2 for validation set and Table 3 for blind test set.

Augmentation Method Effectiveness: Experiments 1 and 2 demonstrate pre-augmentation outcomes, achieving a 21.5% score post-training over 5k and 10k steps, respectively. With augmenting the training data with GPT4-generated samples (Experiments 5 and 6) demonstrated notable improvements in both evaluation and test BLEU scores achieving 22.6% as best score among others and compared to the baseline. This suggests that augmenting the dataset with diverse synthetic data can effectively enhance the model's performance, potentially by exposing it to a wider range of linguistic variations and nuances.

Impact of Dataset Size: The study, Experiment 3, used the larger Madar dataset Bouamor et al. (2018) and development dataset to achieve an impressive evaluation BLEU score of 85.5%. However, this performance did not extend to unseen test sets, where the score dropped to around 17.1%. The high score was observed when the 80K dataset was divided into training and validation sets, suggesting overfitting or overlap. The study highlights the importance of dataset composition and partitioning in model training, as larger datasets may not predict effectiveness on unseen data due to potential domain mismatches or differences in data distribution.

Untuned GPT-4 Translation Performance, experiment 4, which utilizes predictions directly generated by the GPT-4 model without any fine-tuning has acheived a BLEU score of 17.1%, surprisingly yields results comparable to those achieved with fine-tuned models. This observation suggests GPT-4's inherent capability to understand and translate Arabic dialects, underscoring its potential even in the absence of task-specific optimization.

Balancing Data Augmentation and Fine-tuning Experiments 7 and 8, which combined data from GPT3.5 and GPT4 for augmentation, yielded mixed results. While the evaluation BLEU improved compared to the baseline, the test BLEU scores did not show significant improvement. This suggests that a careful balance between data augmentation techniques and fine-tuning strategies is necessary to achieve optimal performance across various datasets and evaluation metrics.

GPT-4-Driven Error Analysis and Feedback, in our evaluation framework, we implemented a concise error analysis using four metrics—lexical, syntactic, semantic, and orthographic—to assess the translation quality from Arabic dialects to MSA. By utilizing GPT-4, we analyzed generated translations for adherence to the original sentences' meaning and structure, facilitating a targeted assessment of model performance across diverse dialects. This methodology enabled us to isolate areas of excellence and deficiency within each model, providing specific feedback on critical sentences representative of each dialect.

This strategic approach underscores the pivotal role of nuanced linguistic analysis in refining translation models, setting a foundation for subsequent enhancements. Figure 5 shows some samples of the performance of our translation models on selected sentences for Experiments IDs of 3, 6 and 8 which show better results among others.

As shown in Figure 5, the error analysis of Arabic dialect experiments reveals that GPT-4 models consistently maintain high fidelity to the original sentences' semantic content, syntactic structure,

Source			Predictions			Error Anaylsis Type				Feedback
ID	Dialect	Source	Experiment 6	Experiment 3	Experiment 8	Lexical Errors	Syntactic Errors	Semantic Errors	Orthographic Errors	-
418452	Egyptian	"وبعدين معناه ايه السؤال ده ده حتى اللي قتل لو جي حد يسأله" "What does this question mean, even if someone killed comes to ask who?"	"ثم ماذا يعني هذا السؤال، حتى لو كان القاتل إذا جاء أحد يسأله؟" "Then what does this question mean, even if the killer comes to ask?"	"إذن ، ماذا عن سؤال آخر لم يطرحه أحد من قبل ؟" "So, what about another question that no one has ever asked before?"	"ثم ما هو معنى هذا السؤال، هذا حتى لو جاء شخص يسأله؟" "Then what is the meaning of this question, even if someone comes to ask it?"	Exp 6 and 8 accurately preserve meaning; Madar's translation in Exp 3 significantly deviates, introducing an unrelated scenario.	Exp 6 and 8 match source syntax; Madar in Exp 3 correct but deviates with new meaning.	Exp 6 and 8 capture original semantics; Madar in Exp 3 introduces unrelated concept, missing the source's essence.	No spelling errors; All Exp use punctuation correctly, despite Madar's semantic shift.	Exp 6 and 8 align closely with source; Madar in Exp 3 significantly diverges, possibly misinterpreting context.
418437	Egyptian	"آه ناقصهم حتة إلتزام شوية" "Ah, they lack a bit of commitment"	"نعم، ينقصهم بعض الالتزام قليلا" "Yes, they lack some commitment a little"	"نعم ولكن ببعض التخفيف في المصاريف." "Yes, but with some reduction in expenses."	"نعم، ينقصهم شيء بسيط." "Yes, they lack something small."	Exp 5 shows accurate translation, Madar in Exp 3 shows incorrect context and Exp 8 shows somewhat curate, lacks specificity.	Structure matches source in Exp 3, note in Ex 3: Structure okay, but unrelated content and in Exp 8 ,simplified but correct structure	Exp 5 captures original meaning but in Exp 3 it misinterprets source, unrelated. and in Exp 8 shows some misses nuance such as "التزام".	Correct usage for all experiments	Exp 5 closely aligns with the source's intent and wording. Exp 3 introduces unrelated concepts, Exp 8 provides a somewhat accurate translation but lacks in specificity.
419298	Iraqi	"أكل أمي آمنة ذكرني بأمي" "My mother Amina's cooking reminds me of my mother"	"أكل أمي آمنة ذكرني بأمي" "My mother Amina's cooking reminds me of my mother"	"طعام أمي آمن ، ذكرني إلى أمي." "My mother Amina's food, remind me of my mother."	"أكل أمي آمنة ذكرني بأمي" "My mother Amina's cooking reminds me of my mother"	Exp 6 and 8 Exact match. Madar in Exp 3: Alters "آمنة" to "آمن," changing meaning.	All models maintain original structure effectively.	Exp 6 and 8 Preserve meaning. Madar in Exp 3: Shifts "آمنة" to "safe," losing specific reference.	No errors detected across models; punctuation and spelling are correct.	Exp 6 and 8 accurately replicate the source. Madar's in Exp3 semantic shift from "آمنة" to "safe" alters the intended personal reference, diverging from the original meaning.
420067	Levantine	"إحنا كلياتنا مضغوطين أرجوكم" "We're all under pressure, please"	"نحن طلابنا مضغوطين، أرجوكم" "Our students are under pressure, please"	"كل منا مشغول على حدة ، من فضلك." "Each of us is busy on their own, please"	"نحن طلابنا مضغوطون أرجوكم" "Our students are pressured, please"	Exp 6 and 8 : Accurate. Madar in Exp 3 Changes context to individual busyness.	Exp 6 and 8 : Structure matches source. Madar in Exp 3 Alters structure to emphasize individuals.	Exp 6 and 8 : Maintain collective stress context. Madar in Exp 3 Misinterprets collective stress as individual tasks.	No orthographic errors in any model; all maintain correct spelling and punctuation.	Exp 6 and 8 accurately translate the collective stress, matching the source's intent. Madar in Exp 3 shifts focus to individual busyness, diverging from the collective emphasis of the original sentence.
426600	Magharebi	"ايه عاقلة يا بعدي هههههههههه" "You're polite, my dear hahaha"	"أيه عاقلة يا بعدي هههههههههه" "You're polite, my dear haha"	"ماذا عن التي تليها ؟ هاها ، هاها." "What about the one that follows? Haha, haha"	"نعم، عاقلة يا بعدي هههههههههه" "Yes, polite, my dear hahaha"	Exp 6 and 8 Lexically faithful to source. Madar in Exp 3: Deviates significantly, altering context.	Exp 6 and 8 Preserve original structure. Madar in Exp 3 Changes sentence structure entirely.	Exp 6 and 8 Captures the playful tone. Madar in Exp 3 Loses original's affectionate tone, introduces unrelated query.	All Exp: Minor variation in laughter representation. Correct usage.	Exp 6 and 8 closely align with the source, maintaining its playful and affectionate tone. Madar in Exp 3 introduces an unrelated concept, diverging from the original sentiment.
50550	Gulf	"بس أشوفكم بعدين سلام" "Just see you later, goodbye"	"فقط أراكم ثم سلام." "Just see you then, goodbye."	"سأعود إليك عندما تكون أقل انشغالا." "I will come back to you when you are less busy."	"ولكن أراكم لاحقا سلام" "But I'll see you later, peace"	Exp 6 and 8 Minor adjustments, remain faithful. Madar in Exp 3: Adds context not present in the original.	All models present a simplified but correct structure.	Exp 6 and 8 Retain the casual farewell intent. Madar in Exp 3 Shifts focus, implying a return during less busy times.	Correct spelling and punctuation across all models.	Exp 6 and 8 accurately capture the casual, farewell nature of the original. Madar, in Exp 3 however, introduces a nuance of returning during a less busy time, diverging from the direct farewell intention.

Figure 5: Sample Error Analysis Across Models, Highlighting Linguistic Accuracy and Model Performance.

and lexical choice, demonstrating their ability to translate Arabic dialects to MSA with minimal errors. However, Experiment 3 (Madar) often diverges from the source, indicating a potential gap in capturing the original's intent. The study emphasizes the importance of model selection in achieving high-quality translations of Arabic dialects and suggests targeted improvements for models struggling with semantic fidelity.

5. Conclusion

In our study for the OSACT 2024 Shared Task on translating Arabic dialects to MSA, we leveraged AraT5 V2 and data augmentation techniques with GPT-3.5 and GPT-4, achieving our best BLEU score of 22.6% with AraT5 V2. This underscores AraT5 V2's effectiveness in capturing the linguistic intricacies of Arabic dialects. Our error analysis further illuminated the strengths of GPT-4 in enhancing translation accuracy across lexical, syntactic, semantic, and orthographic dimensions. These results not only demonstrate the power of AraT5 V2 in handling Arabic translation tasks but also the importance of nuanced error analysis in refining model performance. Moving forward, we aim to integrate emerging technologies to push the boundaries of machine translation for Arabic dialects and MSA.

6. Acknowledgements

The authors thank Prince Sultan University for their support

7. Bibliographical References

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.

Fatimah Alzamzami and Abdulmotaleb El Saddik. 2023. Osn-mdad: Machine translation dataset for arabic multi-dialectal conversations on online social media. *arXiv preprint arXiv:2309.12137*.

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.

Youness Moukafih, Nada Sbihi, Mounir Ghogho, and Kamel Smaïli. 2021. Improving machine translation of arabic dialects through multi-task learning. In *International Conference of the Italian Association for Artificial Intelligence*, pages 580–590. Springer.

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.

Mahmoud Ali Qudah et al. 2017. A sociolinguistic study: Diglossia in social media. In *Conference Proceedings. Innovation in Language Learning 2017*.

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.

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.

Mohamed Ali Sghaier and Mounir Zrigui. 2020. Rule-based machine translation from tunisian dialect to modern standard arabic. *Procedia Computer Science*, 176:310–319.

Connor Shorten, Taghi M Khoshgoftaar, and Borko Furht. 2021. Text data augmentation for deep learning. *Journal of big Data*, 8:1–34.

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*.

8. Language Resource References

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)*.

LLM-based MT Data Creation: Dialectal to MSA Translation Shared Task

AhmedElmogtaba Abdelaziz, Ashraf Elneima, Kareem Darwish
aiXplain Inc.,
San Jose, CA, USA
{ahmed.abdelaziz,ashraf.hatim,kareem.darwish}@aixplain.com

Abstract

This paper presents our approach to the Dialect to Modern Standard Arabic (MSA) Machine Translation (MT) shared task, conducted as part of the sixth Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT6). Our primary contribution is the development of a novel dataset derived from The Saudi Audio Dataset for Arabic (SADA), an Arabic audio corpus. By employing an automated method utilizing ChatGPT 3.5, we translated the dialectal Arabic texts to their MSA equivalents. This process not only yielded a unique and valuable dataset but also showcased an efficient method for leveraging large language models (LLMs) in dataset generation. Utilizing this dataset, alongside additional resources, we trained a machine translation model based on the Transformer architecture. Through systematic experimentation with model configurations, we achieved notable improvements in translation quality with BLEU scores advancing from a baseline of 25.5 to a peak of 31.5 in varied experimental setups. Our findings highlight the significance of LLM-assisted dataset creation methodologies and their impact on advancing machine translation systems, particularly for languages with considerable dialectal diversity like Arabic.

Keywords: Modern Standard Arabic, Dialectal Translation

1. Introduction

The field of neural machine translation (NMT) has seen remarkable progress in recent years. Yet, translating Arabic dialects to Modern Standard Arabic (MSA) presents unique challenges. These challenges stem from the vast linguistic diversity across Arabic dialects and the scarcity of dialect-specific corpora for training effective machine translation systems. Further, there is large lexical overlap Arabic dialects and MSA, and many dialects exhibit common syntactic properties. This paper details our approach to addressing these challenges, highlighting our participation in the Dialect to Modern Standard Arabic Machine Translation shared task at the sixth Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT6) (Elneima et al., 2024).

A pivotal aspect of our contribution lies in leveraging the Saudi Audio Dataset for Arabic (SADA) (Alharbi et al., 2024), an extensive Arabic audio corpus, as the foundation for generating a novel text-based dataset. We developed an automated method that employs ChatGPT 3.5 to translate dialectal Arabic text to corresponding MSA text. This process not only generates a substantial corpus of reasonable quality dialect-specific data but also demonstrates the potential of using Large Language Models (LLMs) for dataset creation in an automated and scalable manner.

Building on this foundation, we explored the efficacy of our newly created dataset, both independently and in conjunction with existing datasets, to train state-of-the-art transformer-based MT models (Vaswani et al., 2017). Our methodology encompasses a detailed examination of model configurations, focusing on optimizing attention heads and embedding dimensions to enhance translation accuracy and fluency. Our contributions are as follows:

- We show the efficacy of using LLMs (ChatGPT 3.5) for creating parallel dialectal-MSA data.
- We build a robust dialect to MSA MT system that combines both existing datasets and LLM generated data.
- We train a single transformer model that translates all dialects to MSA.

2. Related Work

In the evolving landscape of neural machine translation for Arabic dialects, research has predominantly been focused on bridging the linguistic divide between various regional dialects and Modern Standard Arabic. Despite the advancements, the challenge remains in developing comprehensive models that can accommodate the wide array of dialects spoken across the Arab world. In light of these challenges, our work draws inspiration from and seeks to build upon the foundation laid by previous studies, while introducing novel methodologies to enhance translation accuracy and efficiency.

OSACT 2024 Workshop, pages 112–116

The study by Al-Ibrahim and Duwairi (2020) on the Jordanian Arabic dialect and the investigation into Levantine dialects by Baniata et al. (2018) underscore the potential of deep learning techniques in dialect translation and highlight the limitations imposed by dataset size. Our approach similarly leverages deep learning while incorporating an innovative dataset expansion strategy using an existing dataset, namely SADA (Alharbi et al., 2024), combined with automated translation via LLMs, namely chatGPT 3.5, to overcome the corpus size limitation.

Moreover, the transductive transfer learning strategy employed by Yazar et al. (2023) for the Algerian Arabic dialect showcases the effectiveness of knowledge transfer between models. They utilize a pre-trained AraT5 transformer model as the backbone for their translation system. The introduction of TURJUMAN (Nagoudi et al., 2022) represents a significant leap forward, offering a versatile tool for translating multiple languages into MSA. Our work aligns with the spirit of TURJUMAN, emphasizing flexibility and the use of advanced deep learning models. However, we differentiate our approach by focusing on the automated generation of a good-quality dialect-specific dataset that can further refine the translation process.

Lastly, Kchaou et al. (2023) developed a hybrid model using JoeyNMT for the Tunisian dialect translation, achieving good results. We extend this concept by experimenting with various model configurations and training strategies leading to good results across multiple dialects.

3. Experimental Setup

3.1. Data

For training our dialectal to MSA MT systems, we used two different datasets, namely the NADI dataset (Abdul-Mageed et al., 2023) comprising a total of 124,000 segments across various Arabic dialects as detailed in Table 1, and a conversational dataset that we extracted from the SADA speech corpus (Alharbi et al., 2024) and automatically translated to MSA using chatGPT 3.5, which will henceforth refer to as SADA-DA. The dataset contains 1,027,153 segments of naturally occurring dialectal conversations with the breakdown per dialect shown in Table 2.

3.1.1. NADI Dataset

NADI dataset (Abdul-Mageed et al., 2023) is particularly notable for its diversity, encompassing a wide range of Arabic dialects from across the Arab world. The inclusion of the NADI dataset significantly enriched our training corpus, providing a broad spectrum of dialectal variations and linguistic nuances. It spans many Arabic dialects with their sub-dialects. Table 1 lists the dialects in the NADI dataset.

Dialect	Segments
Tunisian	14,000
Iraq	4,000
Libya	4,000
Morocco	14,000
Syria	4,000
Saudi Arabia	4,000
Egypt	4,000
Jordan	42,000
Palestinian	2,000
Qatar	12,000
Yemen	2,000
Algeria	2,000
Lebanon	12,000
Oman	2,000
Sudan	2,000

Table 1: Breakdown of NADI dataset

Dialect	Segments
Hijazi	690,784
Najdi	298,866
Egyptian	11,900
Levantine	7,542
Morrocan	5,540
Algerian	4,677
Janubi	3,603
Iraqi	2,683
Shamali	1,558

Table 2: Breakdown of SADA-DA dataset

3.1.2. SADA-DA

SADA is an Arabic audio dataset composed of roughly 650 hours that are transcribed, diarized, and annotated with gender, approximate age, and dialect (Alharbi et al., 2024). From the SADA dataset, we extracted the transcription of the audio segments that were marked as dialectal. One of the main advantages of the SADA dataset is that the segments are composed of naturally occurring dialectal conversations spanning many genres and topics. Table 2 shows the breakdown per dialect for the SADA-DA.

As can be seen from the dataset, Gulf dialects, namely Hijazi, Najdi, Janubi, and Shamali, are over represented. We prompted chatGPT 3.5 to produce their MSA equivalents. Here are some sample segments with their automatically generated MSA equivalents:

- Shamali:

  - أبي الغدا الغدا بسرعة واللي يرحم والدينك
  - أريد الغداء الآن بسرعة، وبمن يرحم والديك
- Najdi:
  - لا لا بس بس أغراضي خلها اسمع
  - لا لا، فقط أغراضي اتركها واستم
- Morrocan:
  - اه دبما كيقول هكذا
  - دائماً يقول هكذا.

An important note here is that since the validation and test sets for the shared task were also drawn from SADA, we made sure that none of our training sentences were in either set.

To guide the translation process and ensure consistency in the output, we crafted a specific prompt that directed ChatGPT 3.5 to translate texts into MSA, maintain the original text alongside its translation, and separate them using a designated symbol. The prompt used was as follows:

ترجم النصوص التالية للغة العربية الفصحى ،
اكتب كلا من النص الاصلي وترجمته بالعربية الفصحى
وافصل بينهما باستخدام هذا الرمز #

Translation: *Translate the following texts to MSA. Output the original text and its translation with a # as a separator between them.*

This approach allowed for the automated generation of good-quality parallel sentences, where the original dialectal Arabic text and its MSA translation were clearly delineated by the # symbol.

Our experiments illuminated the critical influence of prompt structure on the ChatGPT 3.5 output quality and the tendency to generate hallucinations or inaccurate content. It became evident that simplicity and clarity in prompt design were paramount. By formulating prompts that were succinct and to the point, we minimized the likelihood of hallucinations, thereby enhancing the reliability and accuracy of the translations produced by ChatGPT 3.5. This strategic approach to prompt crafting, focusing on brevity and directness, proved instrumental in facilitating more more accurate machine translations from dialectal Arabic to MSA.

We carried out a preprocessing step that looked into how the lengths of the original and translated texts varied. By spotting and excluding translations with major length discrepancies, we honed in on including only the most promising translations. Building upon this foundation, we proceeded with a manual review by assessing a randomly chosen sample of ChatGPT 3.5's translations, focusing on the translations' fluency. This step was important for identifying language subtleties that automated evaluations, such as BLEU scores, might miss.

3.2. Translation Model

We employed a transformer-based architecture (Vaswani et al., 2017) to address the challenge of translating dialectal Arabic to MSA. The Transformer model, renowned for its effectiveness in capturing complex dependencies in sequence-to-sequence tasks, consists of an encoder-decoder structure. Both the encoder and decoder comprise 6 layers, with each layer hosting 8 attention heads, facilitating the model's ability to focus on different parts of the input sequence simultaneously.

The embedding layers are post-processed with dropout, and the subsequent layers undergo dropout, addition, and normalization, enhancing the model's generalization capability. We employed a dropout rate of 0.1 and label smoothing of 0.1 to mitigate overfitting and improve the model's performance on unseen data.

We utilized tied embeddings, a technique that shares the weight matrix across the input and output embeddings and the decoder's pre-output layer, reducing the model's parameters and encouraging more semantic representations. We used an Adam optimizer with the hyperparameters: 0.9, 0.98, and 1e-09 and a gradient clipping norm of 5. The learning rate is set to 0.0003 with a warm-up of 16,000 steps followed by an inverse square root decay, facilitating a stable and effective convergence.

To facilitate the training of our translation model, we leveraged two state-of-the-art neural machine translation frameworks: Marian NMT (Junczys-Dowmunt et al., 2018) and JoeyNMT (Kreutzer et al., 2019). These frameworks are known for their efficiency, flexibility, and the high quality of the translation models they can produce.

4. Results

The effectiveness of our translation models was rigorously evaluated using the BLEU score (Papineni et al., 2002), a benchmark metric for assessing the quality of machine-translated text relative to a set of reference translations. Our evaluation strategy involved two sets of experiments to discern the impact of dataset composition on translation accuracy.

In the initial phase, we utilized SADA-DA exclusively. With the model configured with 4 attention heads and embedding dimensions of 256 for both the encoder and the decoder, we achieved a BLEU score of 25.5 on the validation set. This served

as a solid baseline, demonstrating the feasibility of our approach for the translation task.

Next, we increased the model's capacity, adjusting the number of attention heads to 8 and the embedding dimensions to 512 for both the encoder and the decoder. This resulted in a notable improvement in translation quality, with the BLEU score reaching 30.2 on the same validation set. This marked a significant performance boost, highlighting the advantages of expanding model capacity for this specific translation challenge.

In the second set of experiments, we combined SADA-DA and NADI datasets to train our models, aiming to reap the benefits of a richer, more diverse training corpus. Under the initial configuration (4 attention head – 256 embedding dimensions) with the combined datasets, the BLEU score improved to 27.3, while the enhanced configuration (8 attention head – 512 embedding dimensions) yielded a further improved BLEU score of 31.5. These results underscore the value of leveraging composite datasets to improve the model's understanding and translation of diverse Arabic dialects into Modern Standard Arabic.

The combination of the NADI and SADA-DA datasets to train our machine translation systems resulted in an approximate 1% enhancement in translation accuracy, as indicated by improved BLEU scores. This enhancement can be attributed to several factors related to the diversity and complementarity of the datasets:

- Increased Linguistic Diversity: The NADI dataset, with its text-based collection spanning various Arabic dialects, and the SADA-DA dataset, derived from conversational audio, collectively encompass a wide linguistic spectrum. This diversity introduces the model to a broader range of dialectal variations, idiomatic expressions, and syntactic structures, enabling it to learn more comprehensive translation patterns.

- Complementary Data Characteristics: The NADI dataset primarily focuses on textual data from digital platforms, which may include formal and semi-formal dialectal usage. In contrast, SADA-DA, being sourced from conversational speech, includes informal dialectal expressions and colloquialisms.

- Robustness to Variability: Training on a mix of text-based and speech-derived datasets exposes the MT system to variations in spelling, grammar, and usage across different contexts.

- Improved Generalization: The combination of datasets mitigates the risk of overfitting to the peculiarities of a single dataset.

- Data Augmentation Effect: The addition of the SADA-DA dataset effectively serves as a form of data augmentation, increasing the volume of training data. This augmentation is particularly beneficial for dialects that are underrepresented in text-based corpora.

Dataset	Heads	Embed	BLEU
SADA-DA*	4	256	25.5
SADA-DA*	8	512	30.2
SADA-DA+NADI*	4	256	27.3
SADA-DA+NADI†	8	512	31.5

Table 3: Experimental results on the validation set using SADA-DA alone and SADA-DA+NADI (*MarianMT, †JoeyNMT)

These experiments illustrate the positive impact of dataset diversity and model capacity on machine translation performance, particularly in the context of translating Arabic dialects to MSA. The advancements in BLEU scores from using the combined SADA-DA and NADI datasets reaffirm the importance of comprehensive and varied training data in developing effective translation models.

5. Conclusion

In this paper, we presented our participation in the OSACT6 shared task on the translation of Arabic dialects to MSA, leveraging state-of-the-art neural machine translation techniques. Our research introduced a novel approach to dataset creation and utilization, primarily focusing on the automated generation of a text corpus from the SADA dataset. This method highlights the efficacy of using LLMs for data generation and the potential of using audio corpora in enriching machine translation training sets.

Our experiments were methodically designed to assess the impact of dataset composition and model configuration on translation performance. The initial experiments, conducted using the SADA-DA dataset alone, set a solid baseline for our translation models. Subsequent experiments with enhanced model capacities further improved translation quality, as shown by the observed increases in BLEU scores. The integration of the SADA-DA dataset with the NADI dataset enabled our models to benefit from a richer and more linguistically diverse training sets. This combination led to notable improvements in BLEU scores, underscoring the value of diverse training corpora in the realm of machine translation. For future work, we plan to experiment with a greater variety of dialect-to-MSA parallel corpora and with n-shot prompting of LLMs.

6. References

Muhammad Abdul-Mageed, AbdelRahim Elmadany, Chiyu Zhang, El Moatez Billah Nagoudi, Houda Bouamor, and Nizar Habash. 2023. NADI 2023: The fourth nuanced Arabic dialect identification shared task. In *Proceedings of ArabicNLP 2023*, pages 600–613, Singapore (Hybrid). Association for Computational Linguistics.

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.

Sadeen Alharbi, Areeb Alowisheq, Zoltan Tuske, Kareem Darwish, Abdullah Alrajeh, Abdulmajeed Alrowithi, Aljawharah Bin Tamran, Asma Ibrahim, Alnajim Raneem Aloraini, Raghad, Ranya Alkahtani, Renad Almuasaad, Sara Alrasheed, Shaykhah Alsubaie, and Yaser Alonaizan. 2024. Sada: Saudi audio dataset for arabic. *ICCASP 2024*.

Laith 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.

Ashraf Elneima, AhmedElmogtaba Abdelaziz, and Kareem Darwish. 2024. Osact6 dialect to msa translation shared task overview. In *Proceedings of the 6th Workshop on Open-Source Arabic Corpora and Processing Tools, LREC'2024*.

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.

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).

Julia Kreutzer, Jasmijn Bastings, and Stefan Riezler. 2019. Joey NMT: A minimalist NMT toolkit for novices. 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): System Demonstrations*, pages 109–114, Hong Kong, China. Association for Computational Linguistics.

El Moatez Billah Nagoudi, AbdelRahim Elmadany, and Muhammad Abdul-Mageed. 2022. 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.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

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.

Bılge Kağan Yazar, Durmuş Özkan Şahın, and Erdal Kiliç. 2023. Low-resource neural machine translation: A systematic literature review. *IEEE Access*, 11:131775–131813.

Sirius_Translators at OSACT6 2024 Shared Task: Fin-tuning Ara-T5 Models for Translating Arabic Dialectal Text to Modern Standard Arabic

Salwa Alahmari[1,2], Eric Atwell[1] and Hadeel Saadany[3],

[1]University of Leeds, UK, [2]University of Hafr Al Batin, Saudi Arabia, [3]University of Surrey, UK
scssala@leeds.ac.uk, E.S.Atwell@leeds.ac.uk, hadeel.saadany@surrey.ac.uk

Abstract

This paper presents the findings from our participation in the 6th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT6) in 2024. Our specific focus was on the second task (Task 2), which involved translating text at the sentence level from five distinct Dialectal Arabic (DA) (Gulf, Egyptian, Levantine, Iraqi, and Maghrebi) into Modern Standard Arabic (MSA). Our team, Sirius_Translators, fine-tuned four AraT5 models namely; AraT5 base, AraT5v2-base-1024, AraT5-MSA-Small, and AraT5-MSA-Base for the Arabic machine translation (MT) task. These models were fine-tuned using a variety of parallel corpora containing Dialectal Arabic and Modern Standard Arabic. Based on the evaluation results of OSACT6 2024 Shared Task2, our fine-tuned AraT5v2-base-1024 model achieved an overall BLEUscore of 21.0 on the development (Dev) set and 9.57 on the test set, respectively.

1 Introduction

To emphasize the significance of addressing Arabic dialects, it's noteworthy that Ethnologue[1] ranks Arabic as the language with the 5th highest number of native speakers, totalling approximately 420 million individuals across 22 countries in the Middle East and North Africa region. Arabic is characterized by diglossia, a linguistic phenomenon where MSA is used in formal contexts, while DA is prevalent in informal settings (Al-Sobh et al., 2015; Abdul-Mageed et al., 2022). Dialects are broadly categorized by region, such as Egyptian or Gulf dialects, but they also exhibit nuanced variations even within individual countries. The linguistic variation presents substantial challenges for MT models trained on MSA. Employing these models, designed specifically for MSA, on DA can be problematic, resulting in subpar translation outcomes when applied to DA. One potential solution to overcome this challenge involves creating parallel corpora, including MSA translations of text written in DA. Recently, considerable efforts have been devoted to translating dialects into MSA. However, the prevalent approach across most studies involves treating each dialect independently. As a result, it is crucial to formulate models with the capability to collectively manage and process at least the most common Arabic dialects.

In this paper, we detail the experiments conducted to develop DA MT model. More precisely, we evaluate the results of fine-tuning different architectures (versions) of the AraT5 transformer model (Nagoudi et al., 2021), employing various datasets for the training phase. The structure of the paper is as follows: Section 2 provides background information about Arabic dialects. Section 3 outlines related works. Section 4 describes the dataset used. The research methodology, including the fine-tuning of AraT5 models and training configuration, is presented in Section 5. In Section 6, we discuss the obtained results. Finally, Section 7 offers a conclusive summary and discusses potential future work.

2 Arabic Dialects Overview

Provided here is contextual background on the variation found in Arabic dialects. MSA represents the formal variant of Arabic, taught in educational institutions and utilized for formal texts and news presentations. MSA has its roots in the Classical Arabic of the Qur'an, albeit experiencing changes in vocabulary and specific aspects of grammar over time. Nevertheless, the majority of Arabs speak their regional dialect as their natural language which is notably different from MSA form of Arabic. While the precise categorization of regional dialects may not be entirely consistent, here are a few main groups:

[1]https://www.ethnologue.com

OSACT 2024 Workshop, pages 117–123

1. **Gulf:** spoken in Gulf countries including Saudi Arbia, Kuwait, Bahrain, Oman and Qatar.
2. **Egyptian:** spoken in Egypt only.
3. **Levantine:** spoken in Levant countries including Lebanon, Jordan, Syria, and Palestine.
4. **Iraqi:** spoken in Iraq and regions of neighbouring countries, also referred to as Mesopotamian Arabic.
5. **Maghrebi:** Spoken in Morocco, Algeria, Tunisia, Libya, Western Sahara, and Mauritania, Maghrebi is influenced by French and Berber(Turki et al., 2016)

Elaborating on how dialectal variations may manifest in their written form, is detailed from a Natural Language Processing(NLP) perspective by Zaidan and Callison-Burch (2014). For example, concerning morphology, they observe that the absence of grammatical cases in dialects is primarily evident in the accusative when a suffix is introduced. This is attributed to the fact that grammatical cases in MSA are typically indicated by short vowels, which are commonly omitted from the text. The absence of duals and feminine plurals is also observable, and the inclusion of circumfix negation. In terms of syntax, the prevalence of the verb–subject–object word order is noted to be higher in MSA compared to dialects. Lastly, distinctions in vocabulary are also discernible in the written text.

3 Related Work

In the realm of neural machine translation (NMT) for DA, the predominant emphasis has revolved around translating these dialects into MSA. Nonetheless, a majority of these studies often centre on a singular dialect, as discussed earlier in this document, leading to a deficiency in models that cover a wide range of Arabic dialects.

As an example, Al-Ibrahim and Duwairi (2020), conducted research focusing on translating the Jordanian Arabic dialect into MSA through deep learning techniques, employing an RNN encoder-decoder model. The progress of their work was, however, constrained by the limited size of the corpus.

Likewise, Baniata et al. (2018) addressed the task of translating Levantine dialects, encompassing Jordanian, Syrian, and Palestinian, into MSA. They utilized a comparatively small dataset of parallel sentences sourced from MADAR PADIC corpora. In their approach, they adopted a multitask learning model, where the decoder was shared across various language pairs, while each source language had its dedicated encoder.

In a similar fashion, Kchaou et al. (2022) adopted a hybrid approach in constructing a translation model for the Tunisian dialect. They proposed various augmentation methods to generate a large corpus and subsequently tested different NMT models using this corpus.

In the domain of low-resource NMT for the Algerian Arabic dialect, Hamed et al. (2023), introduced a transductive transfer learning approach. In this approach, the knowledge is conveyed from parent to child models. The evaluation was conducted employing two datasets; MADAR and PADIC. The implementation of the transductive transfer learning approach done by using two types of NMT models namely: Seq2Seq and Attentional-Seq2Seq.

Furthermore, Nagoudi et al. (2022b) developed TURJUMAN [2], a comprehensive neural toolbox with the capability of translating 20 different languages into MSA. The TURJUMAN toolbox leverages the strengths of AraT5 model and explores its proficiency in Arabic decoding. TURJUMAN was developed to utilize semantic similarity for collecting parallel data samples that are openly accessible, ensuring the quality of the collected data. Most recently, researchers in the NMT field, have come to the fact that transfer learning through straightforward fine-tuning is an effective method, particularly when applied between closely related high-resource and low-resource languages (Zoph et al., 2016).

4 Datasets

In Task 2 of OSACT6 Workshop, organizers shared the Dev and Test set in CodaLab [3], for developing and testing purposes respectively. In addition, participants were free to use any of the available linguistics resources and corpora for training their models, called the Training set (Train). Table 1 gives the total number of dialectal sentences in each of the three datasets (train, Dev and Test) used in this research.

Our methodology starts with the training of the chosen AraT5 models, employing five distinct

[2]https://demos.dlnlp.ai/turjuman/

[3]https://codalab.lisn.upsaclay.fr/competitions/17118

datasets, and fine-tuning their hyper-parameters as a result. After the training phase, we evaluated the performance of our models on the Dev set provided by the organizers of the OSACT6 2024 shared task. Ultimately, predictions were generated using the optimal model configuration on the test set. The following sections in this paper will provide details about the selected datasets, AraT5 models, and the training configuration.

Data set	#Sentences
Train	180,211
Dev	1001
Test	1888

Table 1: Number of Sentences in Train, Dev and Test sets

4.1 Train Set:

Shared Task 2 of the OSACT6 Workshop allowed the participants to use any available resources and tools for training and fine-tuning their models. This section details the datasets employed in training our models. While exploring potentially valuable publicly available datasets, we considered those encompassing various Arabic dialects, specifically regional variations pertinent to the five dialects of interest in Shared Task 2 of OSACT6. We identified and made use of five datasets: 1)MADAR, 2)PADIC, 3)Dial2MSA, 4) Arabic semantic textual similarity (STS) and 5)SADID datasets. Table 2 provides statistics regarding the size of each dataset, measured by the number of pairs of DA sentences alongside their corresponding MSA translations.

MADAR (Bouamor et al., 2019) is a parallel corpus that encompasses different Arabic dialects spoken in 25 cities in Arabic world, along with MSA and English. MADAR stands out as the sole corpus in our training data that covers all five dialects of Shared Task 2, namely: Gulf, Egyptian, Levantine, Iraqi, and Maghrebi.

While **PADIC** (Meftouh et al., 2018) is a parallel corpus comprising texts that belong to two primary Arabic dialects alongside the MSA form. It comprises three sub-dialects from Maghrebi: Algerian, Anab, and Tunisian. Additionally, it incorporates two sub-dialects from Levantine: Syrian and Palestinian.

Dial2MSA The Dial2MSA dataset, as outlined by Mubarak (2018), encompasses tweets written in four distinct Arabic dialects: Egyptian, Gulf, Levantine, and Maghrebi, along with their respective MSA translations. It's important to note that the validation process for the translations was carried out manually only for the Egyptian and Maghrebi dialects. In this research, the entire PADIC dataset, which includes translations that have not undergone validation, was employed during the training phase of our models.

Arabic STS dataset collected by, Al Sulaiman et al. (2022), focuses on determining semantic similarity between two given Arabic sentences. Each English phrase was translated into three target languages namely: MSA, Egyptian and Saudi dialect

SADID (Abid, 2020) is a parallel corpus for English, Egyptian, Levantine and MSA. The dialectal texts were collected from three distinct sources:1) Wikipedia for its diverse domains and clear language, 2) Aesop's Fables for its narrative style, and 3) specific dialogues from movie subtitles. English was chosen as the source language for sentences rather than MSA to avoid introducing bias into the translations (Bouamor et al., 2014). Various translators offer translations with varying degrees of dialectal influence.

	Glf	Egy	Lev	Iraqi	Magh
MADAR	15400	13800	18600	18600	29200
PADIC	0	0	12824	0	19236
Dial2MSA	18010	16355	18000	0	7912
Arabic STS	2758	2758	0	0	0
SADID	0	2997	2997	0	0
Total	36168	35910	52421	18600	37112

Table 2: The number of dialect-to-MSA translation sentences in each of the datasets used in Task 2

4.2 Dev Set

The development set[4] is structured as a JSON file, containing 1001 sentences, with approximately 200 sentences allocated to each dialect. This dataset is essential for improving and evaluating translation systems, with a focus on achieving outstanding results. As you can see in the figure 2, each sentence in the development set has a unique identifier ("id"). The second key is the dialect name label ("dialect"), to which the sentence belongs. The third key in the dictionary is ("source"), representing the textual content of the sentence. Additionally, the key ("target") contains the translation of the sentence into MSA.

[4] https://osact-lrec.github.io

Figure 1: Capture of the JSON File Structure for the Dev Set

4.3 Test Set

The test set[5] is structured as a JSON file, containing a total of 1888 sentences, with approximately 377 sentences allocated to each dialect. These test sentences have been carefully crafted to evaluate the performance of translation systems to accurately convert DA text into MSA. As you can see in Figure 2, each sentence in the test set has a unique identifier ("id"). The second key is the dialect name label ("dialect"), to which the sentence belongs. The third key in the dictionary is ("source"), representing the textual content of the sentence.

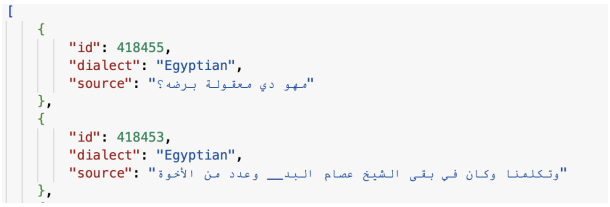


Figure 2: Capture of the JSON File Structure for the Test Set

5 Methodology

Within this section, we present the AraT5 models that serve as our foundation, illustrate the fine-tuning process, and delve into the optimization of hyper-parameters.

5.1 Training Configurations

From the train set, we have observed that the dialectal text and the corresponding MSA text share the same words between them. Based on this observation, we have applied the same method as (Khered et al., 2023), this involves generating an additional pair for every translation pair in our Train set, in which both the source and the target consist of text written in MSA. Table 3 shows an example of the additional pair generation in the Train set. Leveraging these additional pairs empowers our models to grasp the nuances of sentences containing words shared with MSA. In our training setup, we've incorporated all dialect-to-MSA translation pairs from the Train set, focusing on regions pertinent to the five targeted dialects used in training a single model. Consequently, translation pairs from datasets covering Gulf, Egyptian, Levantine, Iraqi, and Maghrebi dialects were employed in the model learning process.

Source	Target
Original Pair	
رجال ياكل مكرونة	رجل يأكل المعكرونة
Additional Pair	
رجل يأكل المعكرونة	رجل يأكل المعكرونة
English Translation	
A man is eating pasta	

Table 3: An example of adding MSA pair to the Train set in which, the source and target are both the MSA translation of the source text

5.2 Fine-Tuning AraT5 Models

The Text-To-Text Transfer Transformer (T5) model transforms various natural language processing (NLP) tasks into a consistent textual format. Among the NLP tasks on which T5 has been pre-trained is MT (Raffel et al., 2020). In our study, we conducted fine-tuning on four distinct AraT5 models: AraT5 base, AraT5v2-base-1024, AraT5-MSA-Small, and AraT5-MSA-Bases

- The **AraT5 base** This model by Nagoudi et al. (2022a), is a tailored version of T5, meticulously fine-tuned to handle and process Arabic text. Functioning as a fundamental model, It demonstrates versatility across a range of natural language processing tasks, including text classification, text generation, and machine translation (MT). AraT5-base effectively leverages the Transformer architecture and pre-trained embeddings to understand and generate Arabic text proficiently.
- The **AraT5v2-base-1024** model signifies an advanced version of AraT5-Base. In the latest iteration of AraT5, AraT5v2[6], the sequence length has been expanded from 512 to 1024, this represented as "1024" in its name. This

[5] https://osact-lrec.github.io

[6] https://huggingface.co/UBC-NLP/AraT5v2-base-1024

extended sequence length significantly enhances the model's adaptability across various NLP tasks. Notably, the fine-tuning process of AraT5v2-base-1024 demonstrates convergence approximately 10 times faster than its predecessor, AraT5-base. This accelerated convergence has the potential to considerably expedite both training and fine-tuning procedures, thereby improving overall efficiency. The selection of this model to be included in our experiments stems from its outstanding performance, as illustrated in Table 5, where its performance surpassed that of other models.

- The **AraT5-MSA-Base** (Nagoudi et al., 2022a), represents an enhanced iteration of AraT5, specifically designed to proficiently handle diverse standard Arabic natural language processing tasks. With an augmented architecture and an increased number of parameters, it excels in tackling intricate tasks that require a profound understanding of the language. AraT5-MSA-Base stands out as an ideal choice for research projects and applications demanding advanced linguistic modelling.

- In contrast **AraT5- MSA-Small** (Nagoudi et al., 2022a), is a refined iteration of the AraT5 model, known as AraT5-MSA-Small, is specifically designed for the streamlined processing of Modern Standard Arabic (MSA) data. It operates at an accelerated pace and requires fewer computational resources compared to its "Base" counterpart. AraT5-MSA-Small is commonly utilized in applications where operational efficiency is crucial, all without a substantial sacrifice in quality.

Our methodology encompassed fine-tuning the above mentioned models using the whole Train set together with all of the four selected AraT5 models. Moreover, the same hyper-meters being used for fin-tuning the models. This standardized methodology empowered us to conduct significant comparisons between the models' performance in our experiments. Table 4 provides information about the hyper-meters used during the training process. All the models employed in our experiments were obtained from the Hugging Face[7] repository. The PyTorch Transformers library[8] is used for designing and executing our Python codes. These hyper-parameters were meticulously chosen to attain optimal performance while reducing the duration of training

Parameters	Values
learning_rate	5e-5
max_target_length	128
max_source_length	128
per_device_train_batch_size	16
per_device_eval_batch_size	16
save_steps	1000
eval_steps	1000
num_train_epochs	2

Table 4: Hyper-parameters for fin-tuning the AraT5 models

6 Results and Discussion

All models utilized in our research underwent evaluation using the BiLingual Evaluation Understudy (BLEU) metric (Papineni et al., 2002), which measures the matching between text generated by the machine (model) and the reference translation based on overlapping words.Table 5 presents the evaluation of model performance, measured in terms of BLEUscore. Notably, the AraT5v2-base-1024 model stands out as the top-performing model, achieving overall BLEUscore of 21.0 when used on the Dev set.

The performance evaluation on the chosen AraT5 models and learning hyper-parameters underscores the intricacies of the translation task, particularly in translating from AD to MSA.The low BLEUscores in our experiments can be attributed to various factors. These encompass issues in the availability of corpora for some dialects in this study, notably the small size of the Iraqi dialect in the total Train set. Additionally, due to time and computational resource constraints,we could not investigate the the impact of the values of varying hyperparameter on the AraT5 models' performance. These combined factors pose challenges in obtaining higher performance reults in Arabic MT tasks. Enhancement of existing resources and creation of new comprehensive Arabic parallel datasets will lead to improvement in the translation outcomes in the future.

[7]https://huggingface.co

[8]https://pytorch.org/hub/huggingface_pytorch-transformers/

Model	BLEUScore
AraT5 base	19.26
AraT5v2-base-1024	21.0
AraT5-MSA-Base	16.88
AraT5- MSA-Small	15.97

Table 5: BLEUscores on the Dev set of the chosen models.

7 Conclusion

This paper outlines our contributions to the OSACT6 2024 Shared Task 2, which revolves around MT of AD into MSA using five Arabic parallel datasets: MADAR, PADIC, Dial2MSA, Arabic STS, and SADID. Throughout our research, we examined four variants of the AraT5 model: AraT5 base, AraT5v2-base-1024, AraT5-MSA-Small, and AraT5-MSA-Base. The experimental findings presented in this study suggest the potential application of these methods to automate the construction of Arabic parallel corpora. Moreover, our commitment extends to advancing research through additional exploration of fine-tuning techniques for transformer models.

Potential future directions include the development of a multilingual model tailored to DA and MSA. Another avenue involves the creation of additional Arabic parallel corpora covering under-resourced Arabic dialects, for example, a corpus of Saudi regional dialects. Additionally, the prevalence of Arabizi—where young Arabs on social media use the Latin script and numerals to represent Arabic sounds—represents an important phenomenon to consider for future research endeavours.

References

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.

Wael Abid. 2020. The sadid evaluation datasets for low-resource spoken language machine translation of arabic dialects. In *International Conference on Computational Linguistics*.

Roqayah M. Al-Ibrahim and Rehab Duwairi. 2020. Neural machine translation from jordanian dialect to modern standard arabic. *2020 11th International Conference on Information and Communication Systems (ICICS)*, pages 173–178.

Mahmoud A Al-Sobh, Abdel-Rahman H Abu-Melhim, and Nedal A 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(2):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.

Laith H. Baniata, Seyoung Park, and Seong-Bae Park. 2018. A multitask-based neural machine translation model with part-of-speech tags integration for arabic dialects. *Applied Sciences*, 8(12).

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, Florence, Italy. Association for Computational Linguistics.

Injy Hamed, Nizar Habash, Slim Abdennadher, and Ngoc Thang Vu. 2023. Investigating lexical replacements for Arabic-English code-switched data augmentation. In *Proceedings of the The Sixth Workshop on Technologies for Machine Translation of Low-Resource Languages (LoResMT 2023)*, pages 86–100, Dubrovnik, Croatia. 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.

Abdullah Khered, Ingy Abdelhalim, Nadine Abdelhalim, Ahmed Soliman, and Riza Batista-Navarro. 2023. UniManc at NADI 2023 shared task: A comparison of various t5-based models for translating Arabic dialectical text to Modern Standard Arabic. In *Proceedings of ArabicNLP 2023*, pages 658–664, Singapore (Hybrid). Association for Computational Linguistics.

K. Meftouh, S Harrat, and Kamel Smaïli. 2018. PADIC: extension and new experiments. In *7th International Conference on Advanced Technologies ICAT*, Antalya, Turkey.

Hamdy Mubarak. 2018. Dial2msa: A tweets corpus for converting dialectal arabic to modern standard arabic. *OSACT*, 3:49.

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. 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. 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.

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.

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.

Houcemeddine Turki, Emad Adel, Tariq Daouda, and Nassim Regragui. 2016. A conventional orthography for maghrebi arabic. In *International Conference on Language Resources and Evaluation*.

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

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575, Austin, Texas. Association for Computational Linguistics.

AraT5-MSAizer: Translating Dialectal Arabic to MSA

Murhaf Fares
Independent Researcher
murhaf@proton.me

Abstract

This paper outlines the process of training the `AraT5-MSAizer` model, a transformer-based neural machine translation model aimed at translating five regional Arabic dialects into Modern Standard Arabic (MSA). Developed for Task 2 of the 6th Workshop on Open-Source Arabic Corpora and Processing Tools, the model attained a BLEU score of 21.79% on the held-out test set associated with the task.

Keywords: Arabic, Neural Machine Translation, T5

1. Introduction

Arabic—a Semitic language spoken by over 400M people—encompasses a range of languages and dialects that have varying degrees of mutual intelligibility (Bergman and Diab, 2022). Perhaps what is even more defining of the Arabic language is the state of *diglossia* where all regional and local Arabic dialects co-exist with a "very divergent, highly codified (often grammatically more complex) superposed variety" (Ferguson, 1959, p. 336)—which is the Modern Standard Arabic (MSA). MSA is often used in formal and legal contexts across Arab countries, while dialectal Arabic (DA) comprises a rich array of regional and local dialects, differing in phonology, morphology, syntax and semantics (Habash, 2022). These variations between Arabic dialects and MSA pose challenges for Arabic Natural Language Processing (NLP) systems, particularly because many of the existing datasets and corpora have been focused on MSA rather than the myriad of Arabic dialects, and the very fact that MSA is shared across the Arab world (Bender, 2019; Bergman and Diab, 2022).[1]

This paper presents a fine-tuned encoder-decoder model to translate dialectal Arabic into MSA. The model is the result of participating in Task 2 under the 6th Workshop on Open-Source Arabic Corpora and Processing Tools; the shared task is presented in more detail in Section 2. The model itself, along with the data used to train it, are described in Section 3. In Section 4 we report the results on the development and test datasets provided by the task organizers. We briefly refer to related work in Section 5 and reflect on findings and the way forward in Section 6.

2. Task Description

The Dialect to MSA Machine Translation Shared Task revolves around translating various Arabic dialects into Modern Standard Arabic, with the intention to bridge the gap between colloquial Arabic and formal written language. Participants were asked to develop models to accurately translate (or convert) dialectal Arabic into MSA. The task covered five regional dialects, namely: the Gulf, Egyptian, Levantine, Iraqi, and Maghrebi dialects. The development and test datasets provided in the task are modestly sized. The development set comprises $1,001$ sentence pairs—200 pairs per dialect—whereas the test set includes $1,888$ sentence pairs that are unevenly distributed over the dialects, as illustrated in Table 1.[2] Participants were allowed to utilize whichever resources available to train and/or fine-tune their systems. All submissions to the shared task were evaluated using two metrics, viz. BLEU (Papineni et al., 2002) and Comet DA (Rei et al., 2022).[3]

3. Model Description

We dubbed our model `AraT5-MSAizer`, and it is the result of fine-tuning the AraT5$_{v2}$ model by Nagoudi et al. (2022)—a pre-trained encoder-

[1] We suspect that there are political as well as religious factors contributing to the marginalization of dialectal Arabic, or even looking down at dialectal varieties as 'ill-formed' Arabic. Though not discussed any further here, it is imperative to examine the status of Arabic NLP resources in light of this, while acknowledging efforts like the OSACT 2024 Shared Task, among others.

[2] According to the Shared Task's website there was supposed to be 500 MSA-dialect pairs for each dialect, both for development and testing. "For each dialect, a set of 500 sentences written in both MSA and dialect will be provided for finetuning, and the testing will be done on a set of 500 blind sentences" `https://osact-lrec.github.io`.

[3] More details on the shared task and the results can be found on: `https://codalab.lisn.upsaclay.fr/competitions/public_submissions/17118`

OSACT 2024 Workshop, pages 124–129

Dialect	No. sentence pairs
Gulf	586
Levantine	568
Magharebi	343
Egyptian	314
Iraqi	77

Table 1: Dialect-wise breakdown of sentence pairs in the test dataset from the shared task.

decoder transformer model (Raffel et al., 2020).[4] We chose to fine-tune this specific model because it was pre-trained on Twitter data, among other datasets, which encompass dialectal Arabic (Nagoudi et al., 2022). In addition, as we describe in Section 5, the AraT5$_{v2}$ model has been used in other related shared tasks for dialect-to-MSA translation.[5] We approached the task as translation from dialect to MSA without distinguishing between the different dialects (even though those were provided in the development and test datasets).

In the following sub-sections, we present the training data used to fine-tune the model and the training configuration.

3.1. Training Data

To fine-tune our model, we used a blend of four distinct datasets; three of which comprised 'gold' parallel MSA-dialect sentence pairs. The fourth dataset, considered 'silver', was generated through back-translation from MSA to dialect, as detailed in Section 3.1.2.

3.1.1. Gold Data

The Multi-Arabic Dialects Application and Resources (MADAR). MADAR includes a parallel corpus of 25 Arabic city-level dialects in addition to MSA (Bouamor et al., 2018). As mentioned before, we train one model to translate from all dialects to MSA, and therefore we 'collapsed' all dialects and sub-dialects in MADAR to just DA, leading to a total of $88,200$ sentence pairs. We reserve an additional $9,800$ pairs for early evaluation and experimentation.[6] MADAR was also used in former related shared tasks such as the Nuanced Arabic Dialect Identification Shared Task organized by Abdul-Mageed et al. (2023).

The North Levantine Corpus. Krubiński et al. (2023) recently introduced a multi-parallel corpus focusing on the North Levantine dialect (aka the 'Shami' or Syrian dialect). The corpus is basically a subset of the OpenSubtitles2018 parallel corpora (Lison et al., 2018) where the Arabic sentences have been manually translated to the North Levantine Arabic dialect.[7] The corpus includes about $120,600$ Shami-MSA pairs; we used 90% of which for training.

The Parallel Arabic Dialect Corpus (PADIC). PADIC is a multi-dialect parallel corpus covering six Arabic (sub-)dialects of the Levantine and Maghrebi regional dialects (Meftouh et al., 2015, 2018). Like with MADAR, we do not distinguish between the different dialects for the purpose of training our model and, hence, end up with a dataset of $41,680$ dialect-MSA pairs.

3.1.2. Synthetic Data

One way to augment our training data is to exploit monolingual data (i.e. MSA-only datasets or corpora). Back-translation is an effective approach to 'create' more training data (Sennrich et al., 2016), where an MT system or model is trained in reverse; that is, the model is trained to translate target (MSA) to source (Arabic dialect). The resulting model can then translate target-side monolingual data back into the source language, creating a synthetic (or silver) parallel corpus for training a source-to-target model.

To generate the synthetic data, we first fine-tuned `AraT5`$_{v2}$ to translate from MSA into dialectal Arabic on the combination of the three aforementioned gold datasets.[8] We then used the resulting MSA-to-dialect model to translate a subset of the Arabic sentences in OPUS (Tiedemann, 2012; Zhang et al., 2020).[9] We filtered the sentences in OPUS to only include Arabic sentences that are longer than 5 characters and shorter than 450 characters.

Given the nature of the data in OPUS, some of the MSA-dialect pairs in the synthesized data included parentheses around foreign names in

[4] `AraT5v2-base-1024` is available on https://huggingface.co/UBC-NLP/AraT5v2-base-1024

[5] It is important to highlight that the model selection and training as well as the data creation process were also constrained by the limited resources available to the author as an independent researcher.

[6] We did not follow the original train-dev-test split in MADAR for selecting those sentences.

[7] The corpus includes pairings with several Indo-European languages but these are not relevant to the work presented here.

[8] We acknowledge that there isn't a singular entity called "dialectal Arabic". However, we posit that if the reverse-translation model is capable of producing any variation of dialectal Arabic, the reuslting synthetic corpus could prove beneficial.

[9] See: https://huggingface.co/datasets/Helsinki-NLP/opus-100

MSA, but not in the dialect translation; we post-processed the dataset to replace the opening and closing parentheses with the empty string in such cases.[10] The resulting synthetic parallel corpus consists of $965,020$ MSA-dialect pairs. As we will see in the following sub-section, not all of those pairs will be used for fine-tuning the final model.

One significant caveat of the MSA-to-dialect translation model is the dominance of the Levantine dialect, which is present in the three gold datasets used to train the model. Indeed the North Levantine Corpus is almost as large as PADIC and MADAR combined, and the last two already include Levantine sentences (cf. Table 2).

3.1.3. Training Dataset

The dataset used to train the model is the combination of the three gold datasets in addition to a further filtered version of the synthetic dataset. After the first round of experiments, we decided to filter out more sentence pairs from the synthetic dataset.

We used the MSA text length, again, to filter out all sentences that are shorter than 25 characters and longer than 300 characters. We opted to keep shorter sentences, as we observed the translation quality degrading as the sentence length increased. Lastly, we augmented the dataset with about $17,000$ randomly-selected sentences from MADAR where MSA is used as both the source and the target.[11] We included those instances to present the model with cases where no changes are required to 'transform' the source text into MSA.

The final combined dataset consists of $700,386$ dialect-MSA sentence pairs in its train split and $77,800$ pairs in the development split. Table 2 summarizes the size of the different datasets.

Dataset	No. pairs
MADAR	$88,200$
PADIC	$41,680$
North Levantine Corpus	$120,600$
Synthetic dataset - OPUS	$965,020$
Gold+synthetic†	$700,386$

Table 2: Number of dialect-MSA sentence pairs in the gold and synthetic datasets. † Gold+synthetic is the final combined and filtered dataset used to train the model.

[10] Parentheses are often used to enclose foreign names in Arabic (open) subtitles.

[11] On second thought, we think those examples could have been sampled from some other monolingual MSA resource.

3.2. Model Fine-tuning

We trained our models by fully fine-tuning AraT5$_{v2}$ for one epoch only using the Transformers library (Wolf et al., 2020). The maximum input length is set to 1024 (same as in the original pre-trained model) whereas the maximum generation length is set to 512. The learning rate and batch size were set to 2e-5 and 32, respectively.[12][13]

4. Results

To gauge the effect of fine-tuning on datasets of varying sizes and qualities, we fine-tuned three AraT5$_{v2}$ models:[14]

(1) `AraT5`$_{\texttt{MADAR}}$ trained on MADAR only

(2) `AraT5`$_{\texttt{Gold}}$ trained on the concatenation of the three gold datasets

(3) `AraT5`$_{\texttt{gold+synthetic}}$ trained on the gold and synthetic datasets

Table 3 shows result of evaluating the three models on the OSACT 2024 development split. From the table we clearly see that the model trained on both the gold and synthetic data outperforms the model trained on gold data only. This observation is consistent with the findings reported by Scherrer et al. (2023) regarding the effectiveness of back-translated data in enhancing the performance of their neural models. To understand how good (or bad) those models are we need a baseline 'model'. We simply used a leave-as-is baseline (Scherrer et al., 2023), where the dialect text is used as translation for MSA (i.e. copy the source to target) and attain 0.1445 in BELU score. With only MADAR data for fine-tuning, we end up with a lower performance than such a basic baseline approach.

As mentioned in Section 3.2, our models are trained for one epoch only, but we did evaluate `AraT5`$_{\texttt{gold+synthetic}}$ on the OSACT 2024 development set every $2,000$ steps. The result of this evaluation can be seen in Figure 1. Note that only greedy search was used with generation when evaluating on the development split. As can be seen from the figure, the model reaches its top performance (with 0.2325 in BLEU) after almost $15,000$ steps, but we don't restore the weights of the best performing model at the end training.

Even though we trained one model for all dialects, we can still examine the results per dialect, which are shown in Table 4.

[12] The training configuration as well as the training script can be found on https://github.com/Murhaf/AraT5-MSAizer

[13] The models were trained on one NVIDIA RTX A6000.

[14] All models were trained using the same configuration and (hyper)parameters outlined in Section 3.2

Model	BLEU
AraT5$_{\text{MADAR}}$	0.1140
AraT5$_{\text{Gold}}$	0.2038
AraT5$_{\text{gold+synthetic}}$†	0.2302
Baseline	0.1445

Table 3: BLEU score on the development split of the AraT5$_{v2}$ model fine-tuned on the MADAR dataset only, three gold datasets and the gold and synthetic datasets combined. † aka AraT5-MSAizer

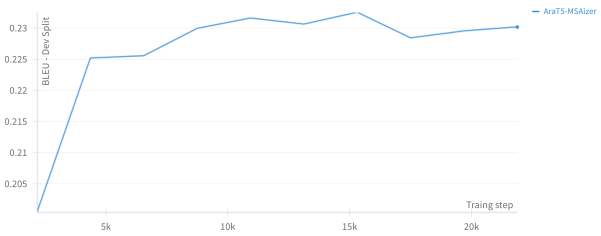


Figure 1: AraT5-MSAizer BLEU score on the OSACT 2024 development set every $2,000$ steps.

The results in Table 4 can be partly explained by the observation made by Bouamor et al. (2014) where they found that Egyptian had the highest lexical overlap with MSA while Tunisian had the least lexical overlap with MSA amongst all the dialects they studied.[15]

Lastly, Table 5 shows the official result of our fine-tuned model, AraT5-MSAizer, on the test split. We used beam search for the final translation submission (specifically, 6 beams) as beam search has proved to lead to better translation performance—at the cost of decoding speed though (Freitag and Al-Onaizan, 2017). Our BLEU score does seem reasonable compared to previously reported results on dialect-to-MSA translation (albeit on different evaluation datasets, cf. Section 5).

[15]We checked the lexical overlap between MSA and the five dialects in the OSACT 2024 development set and found that Magharebi has indeed the least overlap. Note that our lexical overlap method is rather simple, we tokenized the source and target sentences in the dataset, computed the lexical overlap between each pair, and then averaged the lexical overlap per dialect.

Dialect	BLEU
Egyptian	0.2708
Gulf	0.2373
Iraqi	0.2209
Levantine	0.2255
Magharebi	0.2087

Table 4: AraT5-MSAizer BLEU scores for the different dialects in the OSACT 2024 development set

Model	BLEU	Comet DA
AraT5-MSAizer	0.2179	0.0016

Table 5: Official evaluation results on the test split.

5. Related Work

There exists a substantial body of research on statistical and neural machine translation from DA to MSA, but in this section we only focus on Subtask 3 of the NADI-2023 Shared Task (Abdul-Mageed et al., 2023) as it is the most relevant to the OSACT 2024 Shared Task. Of the three participating teams, UniManc (Khered et al., 2023) and Helsinki-NLP (Scherrer et al., 2023) are the most similar to our approach. Both works—among other things—fine-tuned the AraT5$_{v2}$ model on existing parallel corpora for dialect-to-MSA translation. In addition, Scherrer et al. (2023) used a statistical machine translation model (SMT) to back-translate monolingual datasets into dialects which they then used as synthetic parallel corpora to train or fine-tune neural machine translation models.

UniManc—the winning team of task 3 in the NADI-2023 Shared Task—reached their best overall performance by fine-tuning the AraT5$_{v2}$ model on what they call "joint regional" configuration, where all dialect-to-MSA pairs were used to train the same model. We followed a similar approach in the work presented in this paper, but with the addition of synthetic data.

Helsinki-NLP achieved their best performance with SMT models. However, they also fine-tune the AraT5$_{v2}$ model on gold data (viz. MADAR) as well as synthetic back-translated data. Their findings are pretty much in line with ours in that fine-tuning on MADAR-only is barely enough and that back-translation can be effective in the context of fine-tuning pre-trained models.

6. Conclusion

In this paper we presented a machine translation model that builds on a pre-trained text-to-text language model to translate from five different Arabic dialects to MSA. We showed that we can utilize the already existing, though scare, parallel corpora to produce more training data from monolingual resources. We clearly demonstrated that such synthetic data (via back-translation) does indeed help boost the model's performance, in contrast to only relying on gold training data. Despite

the promising results showcased in this paper—which align with recent results in related tasks—we believe that back-translation is not exploited to its fullest yet. One pitfall we would like to avoid in future work is re-using the same 'genre' of text in the different datasets; this is especially the case for the North Levantive Corpus and the synthetic data we chose to back-translate. In addition, we believe one can try and test the idea of iterative back-translation (Hoang et al., 2018), but we suspect a better starting point for the reverse translation system is needed.

7. Bibliographical References

Muhammad Abdul-Mageed, AbdelRahim Elmadany, Chiyu Zhang, El Moatez Billah Nagoudi, Houda Bouamor, and Nizar Habash. 2023. NADI 2023: The fourth nuanced Arabic dialect identification shared task. In *Proceedings of ArabicNLP 2023*, pages 600–613, Singapore (Hybrid). Association for Computational Linguistics.

Emily Bender. 2019. The# benderrule: On naming the languages we study and why it matters. *The Gradient*, 14.

A. Bergman and Mona Diab. 2022. Towards responsible natural language annotation for the varieties of Arabic. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 364–371, Dublin, Ireland. 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).

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

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.

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

Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24, Melbourne, Australia. Association for Computational Linguistics.

Abdullah Khered, Ingy Abdelhalim, Nadine Abdelhalim, Ahmed Soliman, and Riza Batista-Navarro. 2023. UniManc at NADI 2023 shared task: A comparison of various t5-based models for translating Arabic dialectical text to Modern Standard Arabic. In *Proceedings of ArabicNLP 2023*, pages 658–664, Singapore (Hybrid). Association for Computational Linguistics.

Mateusz Krubiński, Hashem Sellat, Shadi Saleh, Adam Pospíšil, Petr Zemánek, and Pavel Pecina. 2023. Multi-parallel corpus of North Levantine Arabic. In *Proceedings of ArabicNLP 2023*, pages 411–417, Singapore (Hybrid). Association for Computational Linguistics.

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).

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.

Karima Meftouh, Salima Harrat, and Kamel Smaïli. 2018. PADIC: extension and new experiments. In *7th International Conference on Advanced Technologies ICAT*.

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 Association for Computational Linguistics*, 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. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Ricardo Rei, José G. C. de Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André F. T. Martins. 2022. COMET-22: Unbabel-IST 2022 submission for the metrics shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 578–585, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Yves Scherrer, Aleksandra Miletić, and Olli Kuparinen. 2023. The Helsinki-NLP submissions at NADI 2023 shared task: Walking the baseline. In *Proceedings of ArabicNLP 2023*, pages 670–677, Singapore (Hybrid). 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.

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).

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.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1628–1639, Online. Association for Computational Linguistics.

ASOS at Arabic LLMs Hallucinations 2024: Can LLMs detect their Hallucinations :)

Serry Sibaee, Samar Ahmed, Omer Nacar,
Abdullah I. Alharbi, Lahouri Ghouti, Anis Kouba

Robotics and Internet-of-Things Lab, Prince Sultan University
Faculty of Computing and Information Technology Rabigh, King Abdulaziz University
Riyadh 12435 Saudi Arabia, Jeddah 22254 Saudi Arabia
{ssibaee, onajar, lghouti, akoubaa}@psu.edu.sa
aamalharbe@kau.edu.sa, Samar.sass6@gmail.com

Abstract

This research investigates hallucination detection in Large Language Models (LLMs) using datasets in the Arabic language. As LLMs gain widespread application, they tend to produce hallucinations—grammatically coherent but factually inaccurate content—posing substantial challenges. We participated in the OSACT 2024 Shared-task, which focuses on the Detection of Hallucination in Arabic Factual Claims Generated by ChatGPT and GPT-4. Our approach evaluates several methods for detecting and mitigating hallucinations, employing models such as GPT-4, Mistral, and Gemini within an innovative experimental framework. Our findings demonstrate significant variability in the models' ability to categorize claims as Fact-Claim (FC), Fact-Improvement (FI), and Non-Fact (NF), highlighting the challenges of dealing with hallucinations in morphologically complex languages. The results underline the necessity for more sophisticated modelling and training strategies to improve the reliability and factual accuracy of the content generated by LLMs. This study lays the foundation for future work on reducing the risks of hallucinations. Notably, we achieved an F1 score of 0.54 in detecting hallucinations with the GPT-4 model.

Keywords: Large Language Models(LLMs), Hallucination Detection, and Arabic Text Classification

1. Introduction

LLMs have experienced a rapid increase in popularity and application since the introduction of GPT in 2021. Capable of producing diverse forms of content including text, code, images, and videos, these advanced models have revolutionized neural natural language generation (NLG) systems. Their enhanced realism in text generation has proven beneficial across a variety of real-world applications such as question-answering, summarization, translation, and paraphrasing. However, alongside these advancements, LLMs face a significant challenge: the phenomenon of hallucination.

Hallucination, as defined by (Ji et al., 2023), is the generation of text or responses that, while grammatically accurate and coherent, deviate from the source inputs in terms of faithfulness or factual accuracy. Essentially, it results in the production of misaligned or factually incorrect information, posing substantial risks to the deployment of LLMs in sensitive real-world applications. With the demand for integrating LLMs into various domains to streamline operations, addressing hallucinations has become a critical concern.

Research to combat this issue generally adopts two main strategies: hallucination detection and mitigation. Hallucination Detection, as explored in (Luo et al., 2024), entails identifying potential hallucinations within LLM-generated responses, at both token and sentence levels, to flag content that significantly diverges from the input. Hallucination Mitigation, on the other hand, aims to reduce the occurrence of hallucinations by enhancing the factual accuracy and reliability of generated content, with methods including the integration of knowledge graphs and retrieval systems.

This study seeks to build upon existing research on Hallucination Detection. The paper is organized as follows: Section 2 reviews related work, Section 3 presents our proposed methodology, Section 4 discusses our experimental results, and Section 5 concludes the paper with a summary of our key findings.

2. Related Work

Prior studies have focused on the detection of hallucinations in LLMs. The research conducted by Snyder et al. (2023) aimed to answer factual questions while examining outputs from three models: OpenLLaMA, OPT, and Falcon. A variety of techniques, including integrated gradient token attribution, SoftMax probabilities, self-attention scores, and fully connected activations, were utilized to distinguish between hallucinated and non-hallucinated generations. While input attribution sometimes per-

OSACT 2024 Workshop, pages 130–134

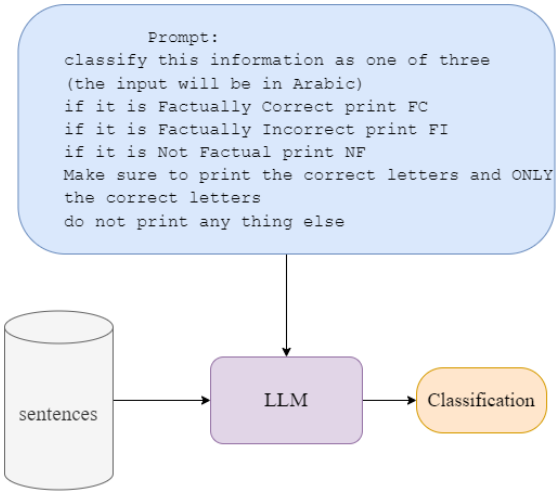


Figure 1: General Framework for our proposed system and the prompt used for the task

formed only marginally better than random chance across different datasets, other techniques demonstrated superior performance on certain datasets. Li et al. (2023) introduced HaluEval, a two-stage framework designed to generate hallucinated samples and conduct high-quality hallucination filtering to evaluate LLMs' performance in recognizing hallucinations. This framework incorporates strategies such as knowledge retrieval, Chain-of-Thought (CoT) reasoning, and sample contrast, enhancing LLMs' abilities to recognize hallucinations and analyze their informational blind spots.

Varshney et al. (2023) proposed an approach for detecting and mitigating hallucinations, focusing on the text generation process. Utilizing GPT-3.5, their study showcased the effectiveness of detection and mitigation techniques, achieving an 88% recall rate and successfully mitigating 57.6% of detected hallucinations without introducing new ones. Liang et al. (2024) emphasized the importance of self-awareness in LLMs for mitigating factual hallucinations. They proposed DreamCatcher, an automated tool designed to evaluate the extent of hallucinations in LLM outputs, classify them by factual accuracy, and provide data for refining LLMs to reduce factual hallucinations. Additionally, the Reinforcement Learning from Knowledge Feedback (RLKF) training framework aims to enhance the factuality and honesty of LLM outputs.

In a comprehensive survey, Tonmoy et al. (2024) discussed the issue of hallucination in LLMs and its impact on their real-world deployment. They highlighted the importance of mitigating hallucinations through prompt engineering and model development techniques. Furthermore, they provided a taxonomy of hallucinations in text generation tasks, analyzed the theoretical aspects of hallucinations in LLMs, and presented existing detection and improvement methods, proposing future research directions in this area. This study aims to contribute to the understanding and mitigation of hallucinations in LLMs.

3. Methodology

In this section, we provide a detailed description of the dataset released by the organizers of the shared task, followed by an explanation of the task itself. We then describe the methods we employed, including the models we experimented with in this study.

3.1. Data and Task Definition

The task involves working with datasets in the Arabic language for Subtask A and Subtask B. For our study, we participated exclusively in Subtask A. In this subtask, participants are required to utilize only the "claim" and "label" columns. The data is tab-separated and includes columns for "claim ID," "word position," "readability," "model," "claim text," and "label." The labels—FC (Factually Correct), FI (Factually Incorrect), and NF (Non-factual)—are used to classify claims into these categories based on their factual accuracy. While Subtask B permits the use of all columns in the dataset, our focus remained solely on Subtask A. Participants are provided with training, development, and testing datasets.

3.2. Models

In our initial experiments, we attempted to use Arabic pre-trained models, such as AraBERT, and fine-tuned them on the provided training data. Unfortunately, this approach did not yield promising results,

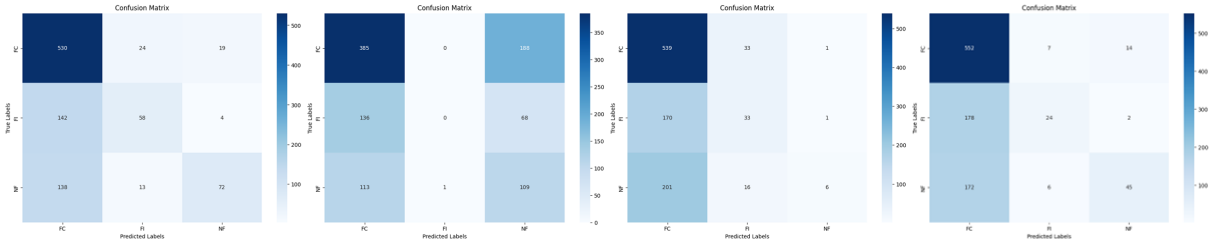


Figure 2: The confusion matrices for the three voter models: a) GPT-4, b) Mistral, c) Gemini and d) ensemble (Majority Voting) for all three models, respectively.

with the maximum F1 score achieved being 44%. Another approach was to represent the input as a one sentence to try to learn the distribution (was was assume to be one that generate correct and wrong sentences) of the data $\wp$ (this assumption was extremely hard to implement). The model used to represent the sentence (input) was "distiluse base multilingual cased" from sentence transformers library (Reimers and Gurevych, 2019) that has a 512 dimensions then was forwarded to a neural network. Despite trying multiple architectures, this approach did not produce encouraging results, prompting us to explore alternative methods.

The main idea of this experiment is to test LLMs ability to detect hallucinations or classify given information as either factually correct, factually incorrect, or non-factual (data that is not declarative). This was achieved by forwarding the text, wrapped in a comprehensive prompt, to control the output format. The LLMs used were GPT-4 and Gemini, both capable of handling Arabic text directly, and Mistral 7B, which was used with a pipeline approach due to its training on English. For Mistral 7B, inputs were translated to English using the Google Translate API before being fed into the model, which was accessed through the Hugging Face Transformers library.

- GPT-4 (OpenAI et al., 2024): GPT-4, the latest iteration in OpenAI's Generative Pre-trained Transformer series, marks a significant leap in natural language processing. With a larger model size and enhanced architecture, GPT-4 excels in tasks like text generation, comprehension, and translation. Its adaptability across various linguistic domains and improved fine-tuning capabilities make it versatile for applications such as conversational agents and sentiment analysis. Despite its technical prowess, GPT-4 prioritizes ethical AI development, focusing on bias mitigation and safety measures. Overall, GPT-4 represents a milestone in NLP, offering unprecedented sophistication and ethical considerations for human-computer interaction and communication.

- Gemini (Team and Rohan Anil, 2023): Gemini, a multimodal AI model by Google, comprehends text, code, and figures, allowing it to read vast scientific literature, reason across disciplines, and answer complex questions. This empowers researchers to conduct faster literature reviews, generate novel hypotheses, and gain insights from complex datasets, ultimately accelerating scientific discovery.

- Mistral : (Jiang et al., 2023) Mistral 7B is a high-performing language model with 7 billion parameters designed for superior efficiency. It surpasses even larger models like Llama 2 (13 billion parameters) across various benchmarks. Mistral 7B particularly outshines in reasoning, mathematics, and code generation compared to Llama 1 (34 billion parameters). The model employs grouped-query attention (GQA) for faster inference and sliding window attention (SWA) to handle sequences of any length efficiently. Additionally, a fine-tuned version, Mistral 7B – Instruct, excels in following instructions, outperforming Llama 2 13B – chat model in both human and automated benchmarks. Overall, Mistral 7B demonstrates outstanding performance and efficacy in natural language processing tasks.

4. Experiments and Results

In this section, we detail the procedure adopted to tackle the problem, beginning with the development of an effective and comprehensible prompt for the used LLMs.

4.1. Experimental Setup

The final prompt, arrived at after several iterations, is depicted in Figure 1. This prompt was utilized with GPT-4, Gemini, and Mistral. For the Mistral model, sentences were translated to English using the Google Translate API before being fed into the model, due to its English-centric training.

Model	Precision	Recall	F1-score
GPT-4	0.67	0.51	0.54
Gemini	0.58	0.38	0.34
Mistral	0.67	0.43	0.42

Table 1: Results for Subtask A on Dev set.

Model	Precision	Recall	F1-score
GPT-4	0.663	0.495	0.516

Table 2: Test Results for Subtask A on test set.

4.2. Results

Our study on the detection of hallucination in Arabic statements generated by LLMs revealed how different models, including GPT-4, Mistral, and Gemini, performed in classifying claims into FC, FI, and NF. The outcomes, presented in Table 1 and through confusion matrices in Figure 2, demonstrate varied model performances. GPT-4 showed overall strong performance but faltered with FI claims, highlighting a deficiency in grasping nuanced content. Mistral had limited success, especially with FI claims, which revealed its difficulty with complex classifications. Gemini, while accurate with NF claims, showed a low recall rate, indicating a potential overemphasis on specific claim types. The use of a Majority Voting technique improved the recall for FC claims but did not significantly improve the classification of FI and NF claims. This highlights the complex nature of nuanced text classification and the need for improved modelling and training approaches to handle the intricacies of languages such as Arabic effectively.

The variance in the performance of different models in various categories highlights the importance of carefully selecting models and utilizing ensemble methods in downstream tasks. The consistent challenge faced with FI claims across all models calls for further investigation into the models' ability to identify and categorize subtle factual changes. In addition, the partial success of the Majority Voting method suggests that combining model outputs does not entirely solve the nuanced classification challenges, which indicates a potential focus for future research in model architecture or training data refinement. Ultimately, we submitted our final results based on the findings obtained from GPT-4, as detailed in Table 2.

5. Conclusion

Identifying and categorizing sentences as factual, non-factual, or uncertain is a challenging task. This challenge arises from the need for models to interpret and extract factual meaning, which is not always a straightforward task. In our research, we introduced a structured prompt designed to utilize LLMs as a tool for factual verification. We tested several models, including GPT-4, Gemini, and Mistral, and found that GPT-4 was the most effective, achieving a Macro F1 Score of 0.54. In future work, we plan to investigate the optimization of Arabic LLMs, with a particular focus on models like Jais, AceGPT, AraGPT, and ArabianLLM, to enhance further their capabilities in verifying factual content.

6. Acknowledgements

The authors thank Prince Sultan University for their support.

7. Bibliographical References

BSI. 1973a. *Natural Fibre Twines*, 3rd edition. British Standards Institution, London. BS 2570.

BSI. 1973b. Natural fibre twines. BS 2570, British Standards Institution, London. 3rd. edn.

A. Castor and L. E. Pollux. 1992. The use of user modelling to guide inference and learning. *Applied Intelligence*, 2(1):37–53.

J.L. Chercheur. 1994. *Case-Based Reasoning*, 2nd edition. Morgan Kaufman Publishers, San Mateo, CA.

N. Chomsky. 1973. Conditions on transformations. In *A festschrift for Morris Halle*, New York. Holt, Rinehart & Winston.

Umberto Eco. 1990. *The Limits of Interpretation*. Indian University Press.

Paul Gerhard Hoel. 1971a. *Elementary Statistics*, 3rd edition. Wiley series in probability and mathematical statistics. Wiley, New York, Chichester. ISBN 0 471 40300.

Paul Gerhard Hoel. 1971b. *Elementary Statistics*, 3rd edition, Wiley series in probability and mathematical statistics, pages 19–33. Wiley, New York, Chichester. ISBN 0 471 40300.

Otto Jespersen. 1922. *Language: Its Nature, Development, and Origin*. Allen and Unwin.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand,

Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Junyi Li, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023. Halueval: A large-scale hallucination evaluation benchmark for large language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Yuxin Liang, Zhuoyang Song, Hao Wang, and Jiaxing Zhang. 2024. Learning to trust your feelings: Leveraging self-awareness in llms for hallucination mitigation. *arXiv preprint arXiv:2401.15449*.

Junliang Luo, Tianyu Li, Di Wu, Michael Jenkin, Steve Liu, and Gregory Dudek. 2024. Hallucination detection and hallucination mitigation: An investigation. *arXiv preprint arXiv:2401.08358*.

OpenAI, :, and Josh Achiam etal. 2024. Gpt-4 technical report.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks.

Charles Joseph Singer, E. J. Holmyard, and A. R. Hall, editors. 1954–58. *A history of technology*. Oxford University Press, London. 5 vol.

Ben Snyder, Marius Moisescu, and Muhammad Bilal Zafar. 2023. On early detection of hallucinations in factual question answering. *arXiv preprint arXiv:2312.14183*.

Jannik Strötgen and Michael Gertz. 2012. Temporal tagging on different domains: Challenges, strategies, and gold standards. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, pages 3746–3753, Istanbul, Turkey. European Language Resource Association (ELRA).

S. Superman, B. Batman, C. Catwoman, and S. Spiderman. 2000. *Superheroes experiences with books*, 20th edition. The Phantom Editors Associates, Gotham City.

Gemini Team and etal Rohan Anil. 2023. Gemini: A family of highly capable multimodal models.

SM Tonmoy, SM Zaman, Vinija Jain, Anku Rani, Vipula Rawte, Aman Chadha, and Amitava Das. 2024. A comprehensive survey of hallucination mitigation techniques in large language models. *arXiv preprint arXiv:2401.01313*.

Neeraj Varshney, Wenlin Yao, Hongming Zhang, Jianshu Chen, and Dong Yu. 2023. A stitch in time saves nine: Detecting and mitigating hallucinations of llms by validating low-confidence generation. *arXiv preprint arXiv:2307.03987*.

Hongbin Ye, Tong Liu, Aijia Zhang, Wei Hua, and Weiqiang Jia. 2023. Cognitive mirage: A review of hallucinations in large language models. *arXiv preprint arXiv:2309.06794*.

Author Index