IUNADI at NADI 2023 shared task: Country-level Arabic Dialect Classification in Tweets for the Shared Task NADI 2023

Yash A. Hatekar
Indiana University Bloomington
yhatekar@iu.edu

Muhammad S. Abdo
Indiana University Bloomington
mabdo@iu.edu

Abstract

In this paper, we describe our participation in the NADI2023 shared task for the classification of Arabic dialects in tweets. For training, evaluation, and testing purposes, a primary dataset comprising tweets from 18 Arab countries is provided, along with three older datasets. The main objective is to develop a model capable of classifying tweets from these 18 countries. We outline our approach, which leverages various machine learning models. Our experiments demonstrate that large language models, particularly Arabertv2-Large, Arabertv2-Base, and CAMeLBERT-Mix DID MADAR, consistently outperform traditional methods such as SVM, XGBOOST, Multinomial Naive Bayes, AdaBoost, and Random Forests.

1 Introduction

Officially Spoken in more than 20 countries, and in a myriad of regional variations, the Arabic language has consistently piqued the curiosity of researchers across various disciplines. This is because of Arabic's historical significance and pivotal role in shaping the cultural, religious, social, and political fabric of the Arab world. Historically, Arabic has often been typologically classified into three distinct categories: Classical Arabic, Modern Standard Arabic (MSA), and Dialectal Arabic (DA). Classical Arabic refers to the language used in the Holy Qur'an and pre-Islamic poetry, while MSA pertains to the language of newspapers, literature, education, official documents, and formal media and news broadcasts. DA, which is the primary focus of this paper, is more concerned with the language used in daily communication by speakers of Arabic. These dialects are often classified into: Egyptian, Levantine, Gulf, and Maghrebi Arabic. Within each of these distinct communities, an array of subdialects can be found in different geographical regions (Diab et al., 2010; Zaidan and Callison-Burch, 2014; Jarrar et al., 2017).

For the most part, MSA was the predominant variation used in written Arabic. But the advent of online forums and social media platforms, such as Twitter, gave the variations of DA a space to grow their written content presence. These dialects differ phonologically, morphologically, syntactically, and semantically. Yet, it is noteworthy to mention that there can still be some degree of overlap between DA and MSA. This is due to the fact that Arabic is a root-based language, which means that many words share common roots consisting of three or four letters. This unprecedented massive increase in digital content in DA has propelled the development of NLP tools that can read, manipulate, and potentially generate this content. While developing such tools to handle text in MSA has posed many challenges, this task has been even more arduous to do for text written in DA, e.g., tweets. Arab users of Twitter mainly use no standardized orthographic variation (e.g., الأهلي, الاهلي, الأهلى), emphasize their thoughts or sentiments through elongation by excessively repeating certain letters (e.g., مستحيييل, لييييييش), miss or add extra spaces between words (e.g., ما يصير, الحمدلله), vary their word choice to the same referent (e.g., أريد, أبغي, عاوز, عايز), to name but a few observations. All these issues present many challenges for developing a single system that can accurately classify all Arabic dialects (Darwish et al., 2014; Jarrar et al., 2014; Lulu and Elnagar, 2018).

In this article, we outline our system, which we entered in Task 1 of the NADI2023 shared task focusing on Arabic dialects classification (Abdul-Mageed et al., 2023). As with the three preceding NADI shared tasks (Abdul-Mageed et al., 2020), (Abdul-Mageed et al., 2021), and (Abdul-Mageed et al., 2022), the primary objective of this task is to develop models capable of categorizing tweets originating from 18 distinct Arab countries.

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 665–669

2 Methodology

2.1 Data

For the purposes of this task, a Twitter dataset of 23.4K tweets, covering 18 different dialects from 18 countries, is provided. This dataset is divided into 3 smaller sets: 18K tweets for training, 1.8K tweets for development, and 3.6K tweets for testing. Additionally, datasets from the previous two NADI tasks (Abdul-Mageed et al., 2020, 2021), and MADAR (Bouamor et al., 2018) were provided. Participants in this task were not allowed to use any other datasets.

2.2 Data Pre-Processing

In the pre-processing phase of our research, we implemented a series of essential steps to prepare the datasets for model training and evaluation. These steps aimed to enhance the quality and consistency of the data, ensuring optimal model performance. To accomplish this, we followed the data pre-processing methods outlined in previous studies (Badaro et al., 2018; Muaad et al., 2022). These pre-processing procedures collectively served to optimize the datasets for subsequent training and evaluation of our models. The pre-processing techniques are as follows:

Diacritics Removal: The small marks used to indicate pronunciation in Arabic were systematically eliminated from the datasets (e.g., مُجَتَمع > مجتمع).

Hamza Normalization: A glottal stop represented in multiple ways in Arabic, underwent a normalization process (آ، إ، أ > ا). This in turn included normalizing Lam Alif.

Kashida Removal: Excessive elongation of Arabic letters was adjusted (e.g., فلسطيـــــــــن > فلسطين).

Punctuation Removal: All punctuation marks were removed from the datasets.

Spelling Error Correction: Common spelling errors in the text were systematically corrected.

In addition, as part of our pre-processing pipeline, we implemented another step involving the mapping of numerical labels to their corresponding country names. Linking numerical labels to countries helped us associate data with geographic regions during the stages of analysis and training. It was an important initial step in preparing the data for further processing. The labels 0 to 17 were respectively associated with the following countries: Iraq, Oman, Syria, Yemen, Morocco, Lebanon, Tunisia, Kuwait, Algeria, UAE, Sudan, Libya, Jordan, Egypt, Bahrain, Palestine, Saudi Arabia, and Qatar.

2.3 Classifiers

We deemed this as a classification task. We used Transformer-based models such as Arabertv2 base, and large (Antoun et al., 2020) and CAMeLBERT-Mix DID MADAR (Inoue et al., 2021). The choice of these BERT-based models was because they were trained on data we were allowed to use. We also used traditional models such as Naive Bayes, SVC, XGBoost, AdaBoost, and Random Forests. All the models were trained on a combined dataset of all the provided datasets. Our BERT-based model Arabertv2-large performed the best on the development dataset. To fine-tune AraBERT for sequence classification, we employ the same approach that (Antoun et al., 2020) used. This involves taking the final hidden state of the initial token, specifically associated with the word embedding of the special "[CLS]" token positioned at the beginning of each sentence. Subsequently, we integrate a basic feed-forward layer coupled with the standard Softmax function to yield a probability distribution across the predicted output classes. During the fine-tuning process, both the classifier and the pre-trained model's weights are collaboratively trained to maximize the log probability of correctly predicting the class.

In terms of the training setup, we utilize a set of configuration parameters encapsulated in the 'TrainingArguments' variable. The parameters we used are similar to that of (Antoun et al., 2020) provided in their examples notebook. We set 'adam_epsilon' to a value of 1e-8 for optimization, 'learning_rate' at 2e-5 for the learning rate, and 'fp16' can be enabled when using high-performance GPUs like V100 or T4. The 'per_device_train_batch_size' is set at 16, although it can go up to 64 when working with 16GB of GPU memory and sequences of a maximum length of 128. To manage memory effectively, 'gradient_accumulation_steps' is configured at 2, allowing for an increase in batch size.

The training process spans 'num_train_epochs' for 3 cycles. 'warmup_ratio' is set to 0, indicating no warm-up steps. Evaluation is incorporated ('do_eval = True'), and this evaluation strat-

Model Name	F1-Score	Accuracy	Precision	Recall
Arabertv2-Large	0.71	0.71	0.71	0.71
Arabertv2-base	0.71	0.71	0.70	0.71
CAMeLBERT-Mix DID MADAR	0.71	0.71	0.70	0.71
XGBoost	0.52	0.51	0.60	0.51
Random Forest	0.43	0.42	0.51	0.42
Naïve Bayes	0.41	0.45	0.73	0.45
SVC	0.39	0.40	0.58	0.40
AdaBoost	0.18	0.18	0.50	0.18

Table 1: Model Performance Comparison on Development Data

Model Name	F1	Accuracy	Precision	Recall
Arabertv2-large	70.22	70.78	71.32	70.78

Table 2: Official results of the IUNADI submission

egy is executed 'epoch' by 'epoch'. Further, the 'save_strategy' also operates 'epoch' by 'epoch', and it's designed to 'load_best_model_at_end' for automatic selection of the best model based on a specified metric, 'macro_f1', where 'greater_is_better' is set to true. 'Macro_f1' is used because F1 was the official metric. Lastly, a 'seed' value of 47 is employed for reproducibility.

Finally, during the training of Arabert-Large, we employed a training ensemble methodology within the framework of a 5-fold cross-validation setup. Our final predictions were derived by aggregating the scores of the individual models. This ensemble approach facilitated improved model performance and robustness in our research.

3 Evaluation

For subtask 1, the evaluation metrics will include precision, recall, f-score, and accuracy. Macro-averaged F-score will be the official metric; hence we report our results using this metric along with all the evaluation metrics. We decided which models to submit based on the model's performance on the development dataset provided by the organizers.

4 Results

As shown in table 2, we only submitted a single system for evaluation, namely, Arabertv2-large. We selected this model because it has over 2.5 times more parameters than Arabertv2-base and CAMeLBERT-Mix DID MADAR. Our system achieved an F-1 score of 70.22 on the test set.

In addition to the officially submitted systems, we performed a more extensive evaluation of the development set. We trained and evaluated 8 different classifiers. The results of these experiments are shown in table 1. The best model performance was achieved by the three models Arabertv2-large, Arabertv2-base, and CAMeLBERT-Mix DID MADAR. The non-neural classifiers generally showed lower performance than transformers.

Our pre-processing pipeline had a positive effect on the Random Forests model, improving the F1 score to 0.43, compared to 0.39 without pre-processing. In contrast, it had a detrimental impact on the Naive Bayes model, reducing F1 to 0.41 from 0.43 without pre-processing. The pipeline had no impact on the results of XGBOOST, SVC, and AdaBoost. It is important to note that pre-trained models already incorporate their own internal pre-processing pipelines. Even though the pipeline did not achieve significant results, we still believe it was necessary to eliminate redundancy and reduce data size.

5 Discussion

The Arabic dialect identification task, as explored in this research, addresses a crucial challenge in natural language processing, particularly for applications involving Arabic text. We observed promising results during the evaluation phase, demonstrating the system's ability to correctly identify Arabic dialects with a high degree of accuracy. However, it is essential to recognize that the task itself presents inherent challenges due to the nuances and variations present within Arabic dialects. Arabic speakers often code-switch between dialects and Standard Arabic, which affects the performance of models. Given an additional three months to work on this task, several avenues for improvement and further development can be pursued:

Fine-Tuning Strategies: Experimenting with

advanced fine-tuning techniques, such as domain adaptation or multi-task learning, may help the model handle ambiguous phrases and code-switching more effectively.

Post-Processing Techniques: Implementing post-processing techniques, such as dialect consistency checks, to ensure that the identified dialect remains consistent within a given text could mitigate errors caused by code-switching.

6 Conclusion

In this paper, we have detailed our contributions to the NADI 2023 shared task on Arabic tweet classification across 18 Arab countries. Our experiments have revealed that employing Arabertv2-large yields the most promising results. Our system achieved a ranking of 13th out of 16 participating teams. Looking ahead, our future research will explore the potential benefits of employing ensemble-based approaches with transformer-based models. Additionally, we are keen to investigate the potential advantages of incorporating tokenization, stop word removal or splitting, and stemming into our pre-processing pipeline.

Ethics Statement

This work is primarily for the benefit of the Arabic language community, which despite having hundreds of millions of speakers, still lacks computational resources. While we believe that our project does not pose any potential harm, we urge users to take all ethical considerations into account when using it.

Acknowledgments

This research was supported in part by Lilly Endowment, Inc., through its support for the Indiana University Pervasive Technology Institute. We would also like to thank Indiana University for providing access to their computing resources and servers, which were instrumental in conducting the experiments for this research.

References

Muhammad Abdul-Mageed, AbdelRahim Elmadany, Chiyu Zhang, ElMoatez Billah Nagoudi, Houda Bouamor, and Nizar Habash. 2023. NADI 2023: The Fourth Nuanced Arabic Dialect Identification Shared Task. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*.

Muhammad Abdul-Mageed, Chiyu Zhang, Houda Bouamor, and Nizar Habash. 2020. Nadi 2020: The first nuanced arabic dialect identification shared task. *arXiv preprint arXiv:2010.11334*.

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, Houda Bouamor, and Nizar Habash. 2021. Nadi 2021: The second nuanced arabic dialect identification shared task. *arXiv preprint arXiv:2103.08466*.

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, Houda Bouamor, and Nizar Habash. 2022. NADI 2022: The third nuanced Arabic dialect identification shared task. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 85–97, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

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

Gilbert Badaro, Ouijdane El Jundi, Ali Khaddaj, Ahmad Maarouf, Reine Kain, Hazem Hajj, and Wassim El-Hajj. 2018. Ema at semeval-2018 task 1: Emotion mining for arabic. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 236–244.

Houda Bouamor, Nizar Habash, Mohammad Salameh, Wajdi Zaghouani, Owen Rambow, Dana Abdulrahim, Dima Obeid, Salam Khalifa, Fatima Eryani, Andreas Erdmann, and Kemal Oflazer. 2018. The madar arabic dialect corpus and lexicon. In *Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018)*.

Kareem Darwish, Hassan Sajjad, and Hamdy Mubarak. 2014. Verifiably effective arabic dialect identification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1465–1468.

Mona Diab, Nizar Habash, Owen Rambow, Mohamed Altantawy, and Youssef Benajiba. 2010. Colaba: Arabic dialect annotation and processing. In *Lrec workshop on semitic language processing*, pages 66–74.

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

Maha Jarrar, Nizar Habash, Dana Akra, and Nasser Zalmout. 2014. Building a corpus for palestinian arabic: a preliminary study. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 18–27.

Maha Jarrar, Nizar Habash, Fatima Alrimawi, Dana Akra, and Nasser Zalmout. 2017. Curras: an annotated corpus for the palestinian arabic dialect. *Language Resources and Evaluation*, 51:745–775.

Luma Lulu and Ahmed Elnagar. 2018. Automatic arabic dialect classification using deep learning models. *Procedia computer science*, 142:262–269.

Ali Y Muaad, Harisha J Davanagere, D S Guru, Jelili B Benifa, Chanda Chola, Hani AlSalman, Abdullah Gumaei, and Moulay A Al-antari. 2022. Arabic document classification: performance investigation of preprocessing and representation techniques. *Mathematical Problems in Engineering*, 2022:1–16.

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