Dialect & Sentiment Identification in Nuanced Arabic Tweets Using an Ensemble of Prompt-based, Fine-tuned and Multitask BERT-Based Models

Reem Abdel-Salam
Cairo University, Faculty of Engineering, Computer Engineering / Giza, Egypt
reem.abdelsalam13@gmail.com

Abstract

Dialect Identification is important to improve the performance of various application as translation, speech recognition, etc. In this paper, we present our findings and results in the Nuanced Arabic Dialect Identification Shared Task (NADI 2022) for country-level dialect identification and sentiment identification for dialectical Arabic. The proposed model is an ensemble between fine-tuned BERT-based models and various approaches of prompt-tuning. Our model secured first place on the leaderboard for subtask 1 with an 27.06 F1-macro score, and subtask 2 secured first place with 75.15 F1-PN score. Our findings show that prompt-tuning-based models achieved better performance when compared to fine-tuning and Multi-task based methods. Moreover, using an ensemble of different loss functions might improve model performance.

1 Introduction

Arabic, spoken by over 500 million people worldwide, is the most populous member of the semitic language family. In general, Arabic can be divided into three categories: (1) Classical Arabic, the language of early literature; (2) Modern Standard Arabic (MSA), which is commonly used in school and formal settings; and (3) Dialectal Arabic (DA), a collection of geopolitically defined varieties. The existence of several dialects and complicated morphology are two distinguishing features of the Arabic language. Furthermore, the casual nature of social media chats, as well as the variations between MSA and DA, add to the complexity. Arabic dialects are not standardized. There are no formal grammar rules or formalism to guide the speakers. This makes various tasks such as machine translation and speech recognition challenging. Several works have been proposed to improve dialect identification as the recent shared-task NADI series (2020 and 2021) (Abdul-Mageed et al., 2021b, 2020). Several teams have used traditional methods as SVM with TF-IDF (Touileb, 2020; Nayel et al., 2021), others customized Bert-based models. AlKhamissi et al. (2021) added an adapter layer on top of MARBERT model. The authors of (El Mekki et al., 2021) used multi-task learning to predict dialect on provenance and country level. This paper presents our work in the Nuanced Arabic Dialect Identification (NADI) shared task (Abdul-Mageed et al., 2022). The NADI shared task (2022) consists of two subtasks. The first subtask is a country-level dialect identification, while the second subtask is sentiment analysis based on different Arabic dialects. Given that a key challenge in this task is the unbalanced distribution and the hard nature of the problem. We follow best practices from recent work on enhancing model generalization and robustness. The rest of the papers goes as follow: section 3 discusses the proposed methods, section 4 shows experimental results, and section 5 concludes the paper. The code has been made open-source and available on GitHub [1].

2 Data

Subtask	Train-set	Dev-set	Test-set	
1	20,398	4,871	4,758 test A	1,474 test B
2	1,500	500	3,000	

Table 1: Train-validation distribution for subtask 1 and 2.

The NADI dataset provided by the organizers consists of 2 datasets for each subtask. Table 1 shows the train-set, dev-set, and test-set distribution for both subtasks. Subtask 1 covers 18 country levels dialects: Algeria, Bahrain, Egypt, Iraq, Jordan,

[1] https://github.com/rematchka/Dialect-and-Sentiment-Identification-in-Nuanced-Arabic-Tweets

Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP), pages 452 - 457

Kuwait, Lebanon, Libya, Morocco, Oman, Palestine, Qatar, Saudi Arabia, Sudan, Syria, Tunisia, United Arab Emirates, and Yemen. However, the data is extremely unbalanced. The train-set consists of 20,398 tweets, while the dev-set consists of 4,871 tweets. Subtask 2 covers 3 labels positive, negative, and neutral for dialectal sentiment analysis. The tweets span different ten Arab dialects. The dataset does not suffer from class imbalance.

3 System Description

This section presents the various approaches used while developing the final models: a voting classifier, a weighted ensemble of BERT-based models, and a prompt-BERT-based model.

Experimental setup for the fine-tuned models the learning rate was set to 4e-5 or 4e-6, cosine-annealing learning rate scheduler was used, the model's weight decay was set to 1e-8 and the length of the sentence for tokenization was set to 128 or 256. During training, batch size was set to 32, and at the end of each epoch, the model was evaluated on dev-set. The best-performing model in terms of F1-macro is saved.

3.1 Subtask 1 models

In subtask 1, the goal was to identify 18 different Arabic dialects, in an unbalanced dataset. In order to tackle this problem, we have experimented with several approaches. Most of the models used were BERT-based models such as MARBERT (Abdul-Mageed et al., 2021a), AraBERT (Antoun et al.), QARiB (Abdelali et al., 2021), AraELECTRA discriminator (Antoun et al., 2021a). Two methods Were used: 1) Fine-tuning, 2) Prompting-tuning. Table 2 shows a summary for models and techniques used. For MARBERT with prompt-tuning, openprompt library was used (Ding et al., 2021), which used P-tuning. In P-tuning (Lester et al., 2021) prompts are only inserted into the input embedding sequence, and this embedding is fed to the language model head and output is output to the linear classification head. One of the challenges in promoting is the design of the prompt and the output of the model. For the prompt we have used [MASK]اللغة هي ("**language is [MASK]**"), and for the output, we have used countries' names translated into Arabic.

Submitted systems for this subtask 3 systems were submitted, the first system was the prediction of MARBERT with prompting. The second is a weighted ensemble between all models listed in table 2. The weights were determined by using optimization, where the goal is to find weights that improve the prediction score in dev-set. As a result, some of the weights assigned to models were chosen to be zero. These models were Araecltra discriminator, AraBERT v2 twitter and AraGPT2 (Antoun et al., 2021b). The third system was a hard voting between MARBERT fine-tuned version and the prompt version.

3.2 Subtask 2 models

In subtask 2 the goal was to analyze sentiments in dialectal tweets. Several model experiments has been done as shown in table 3. In this subtask three approaches have been explored: 1) Multi-task learning (MTL), 2) Fine-tuning 3) Prompt-tuning.

3.2.1 MTL

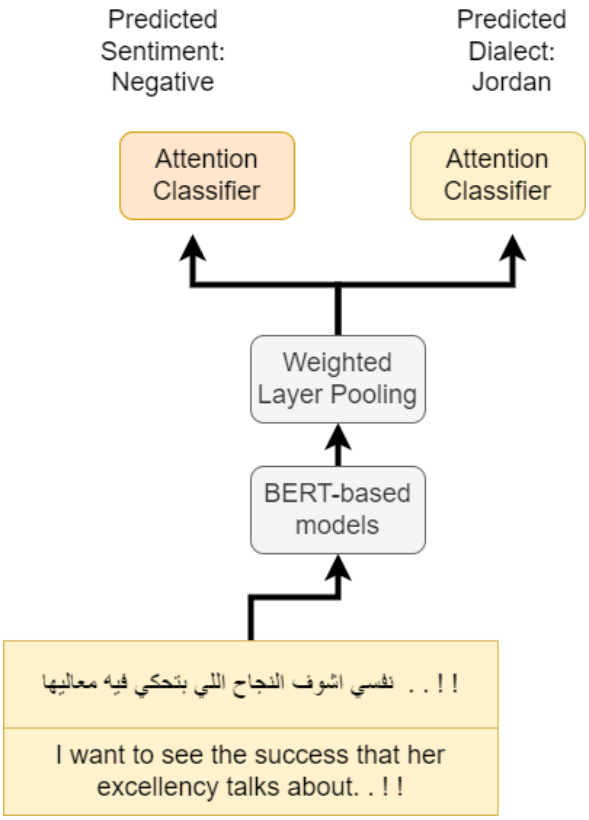

Figure 1: MTL architecture.

In MTL single-input multi-output approach was used, where we have two task-specific attention classifier layers, which help with the classification of the dialect and correspondent sentiment for a tweet. These layers work on top of the weighted pooling that used the output of the last 4 layers of

Models	Methods	Classification head	Loss Function	Macro-F1
MARBERT	Fine-tuning	Attention classifier	F1-CrossEntropy	33
Arabelectra Discriminator	Fine-tuning	Weighted pooling with Attention Classifier	Ensemble of F1-CrossEntropy and Focal Loss	22
AraBERT V2 twitter	Fine-tuning	Weighted pooling with Attention Classifier	Ensemble of F1-CrossEntropy and Focal Loss	29
AraGPT2	Fine-tuning	Attention classifier	Ensemble of F1-CrossEntropy and Focal Loss	22
QARiB	Fine-tuning	Weighted pooling with Attention Classifier and multi-sample dropout	Ensemble of F1-CrossEntropy and Focal Loss	25
MARBERT	**Prompt-tuning**	-	**CrossEntropy**	**37**

Table 2: Models and techniques developed during the experimental phase for subtask 1 and the F1-macro on the dev-set.

BERT-based-model, as shown in figure 1. In order to get the dialect and sentiment of a corresponding tweet, we have used a fine-tuned model to provide pseudo-labels for both datasets (subtasks 1 and 2). The train-set of both subtasks was concatenated and used for training MTL model.

3.2.2 Prompt-tuning

For prompt-based tuning, several approaches have been explored as prefix prompting (Li and Liang, 2021), OpenPrompt library, and P-tuning V2 with and without LSTM encoder. For prefix prompting, language model generation versions of BERT-base models were used. For the prompt, we have used [MASK]تحليل المشاعر ("**sentiment analysis is [MASK]**"), and for the output, we limited the model to generate three labels corresponding to sentiments, which are سعيد، سلبي، محايد ("**neutral, negative, happy**"). Figure 2 shows the architecture. During experiments, we tried to make the model generate the synonyms for these three labels. However, it turns out that limiting model generation to generate only 3 labels text was the best option in this task in terms of dev-set score. For OpenPrompt library, P-tuning V2 with and without LSTM, several prompts were used as [MASK]؟ما هو شعور الكاتب (" **what is the Sentiment of the writer? [MASK]**") , [MASK]تحليل المشاعر ("**sentiment analysis is [MASK]**"), and [MASK]المشاعر ("**Sentiment is [MASK]**").

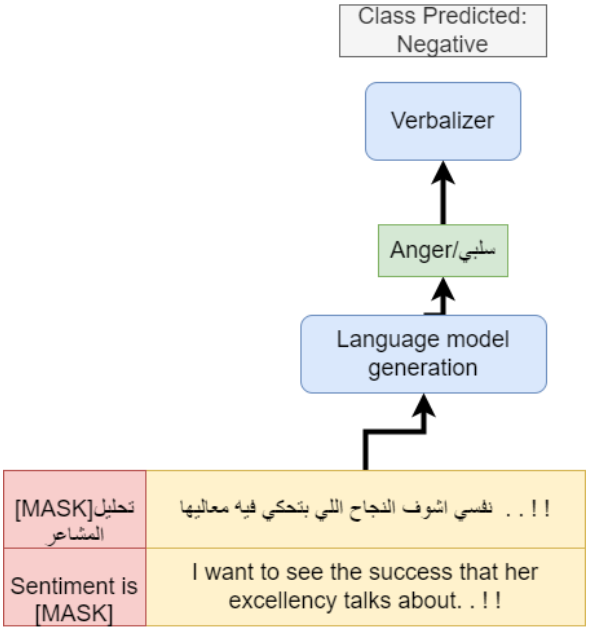

Figure 2: Prefix prompting architecture.

3.2.3 Submitted systems

for this subtask four different systems were submitted. The first system is an ensemble of the last 7 models in table 3. For determining weights we used the optimization method, where the goal is to find the best weight that improves overall pre-

Models	Methods	Classification head	Loss Function	Macro F1-PN
AraELECTRA-base-discriminator	Multi-task learning	Weighted layer pooling with Attention Classifier	FocalLoss (Lin et al., 2017)	66.5
MARBERT	Fine-tuning	Weighted layer pooling with Attention Classifier	Ensemble of F1-CrossEntropy and Focal Loss	71.5
MARBERT	Feature Engineering and Fine-tuning	LSTM with Classifier	CrossEntropy	72
AraBERT	Fine-tuning	Weighted layer pooling with Attention Classifier	F1-CrossEntropy	63.5
AraELECTRA-base-discriminator	Fine-tuning	Attention Classifier	CrossEntropy	58
AraBERT	p-tuning v2	Classifier	CrossEntropy	67.5
AraELECTRA-base-discriminator	p-tuning v2	Classifier	CrossEntropy	61.5
MARBERT	**p-tuning V2**	**LSTM to encode prompt and Classifier**	**CrossEntropy**	**73.5**
MARBERT	Prefix-Prompt tuning	-	CrossEntropy	72.5
AraBERT V2 twitter	Prompt-tuning	-	CrossEntropy	72.5
MARBERT	Prompt-tuning	-	CrossEntropy	73
AraBERT Large V2 twitter	Prompt-tuning	-	CrossEntropy	71.5
GigaBERT-v3 (Lan et al., 2020)	Prompt-tuning	-	CrossEntropy	62.5
AraGPT2	Prompt-tuning	-	CrossEntropy	60
CAMeLBERT (Inoue et al., 2021)	Prompt-tuning	-	CrossEntropy	67.5

Table 3: Models and techniques developed during the experimental phase for subtask 2 and macro F1-PN on dev-set.

diction on the dev-set. It turns out, that the best weight chosen is uniform $1/7$. The second and third submissions were hard and soft voting based on the prediction of the last 7 models in table 3. The fourth submission was based on a weighted ensemble between the first four models in the table. Similarly, optimization has been carried out to choose the best weights. It turns out that the third model (MARBERT with feature engineering and LSTM) was not important, and its weight was set to zero.

4 Results

In this section, The performance of the model is reported based on the official metric during dev-phase and test-phase. Moreover, error analysis is conducted to identify weaknesses of the proposed models. For subtask 1 the official metric is the macro average F1-score, while for subtask 2 the official metric is the macro-F1-PN score (macro f1-score for the negative and positive classes only).

4.1 Dev-phase results

The table 2 illustrates our model's dev-phase scores for subtask 1 using the macro F1-score metrics. It is clear that the low results reflect the difficulty of the task. The key problem, we believe, is the dataset's unbalanced nature. To improve performance, we tried a variety of ways. We tried oversampling, undersampling, batch-sampler, and balanced sam-

System Submission	Macro-F1	
	Test A	Test B
System 1: MARBERT with Prompt	36.3556	17.5
System 2: Weighted Ensemble	36.4807	17.6
System 3: Hard Voting	36.3291	17.17
Over All Performance	**27.06**	

Table 4: Performance of the submitted models on the leaderboard in subtask 1.

pling, but none of these produced satisfactory results. Table 3 shows results on dev-set for subtask 2. It can be concluded that prompt-based model performance was better than fine-tuning methods.

4.2 Test-phase results

Table 4 and 5 show performance the submitted model in the test-phase. For subtask 1, in test A the best-performing model was the weighted ensemble voting. For the second place, the MARBERT with prompt comes in place. For test B, the best performing model was the weighted ensemble, while the best second model was MARBERT with a prompt which achieved a good results (0.1) error difference compared to the weighted ensemble. In Subtask 2 the best performing model was system 4 which was an ensemble of fine-tuned models, MTL, and different versions of prompting.

4.3 Error analysis

As seen in Figure 3, our model performs well when predicting Egyptian, Saudi Arabian, Algerian, Oman, Libyan, and Iraqi languages. According to the confusion matrix, most dialects were incorrectly classified as these five dialects. We assume this is due in part to a large number of tweets from each dialect in the training-set. Further examination of the output revealed that our model performs very poorly on the less common dialects. Our approach is unable to reliably forecast the dialects of Palestine, Qatar, Bahrain, and the United Arab Emirates. We believe this is due to the skewed nature of the data once again, but also to the difficulty in distinguishing various dialects in general.

System Submission	F1-PN
System 1: Weighted Ensemble	72.77
System 2: Hard Voting	72.224
System 3: Soft Voting	72.224
System 4: Weighted Ensemble	**75.155**

Table 5: Performance of the submitted models on the leaderboard in subtask 2.

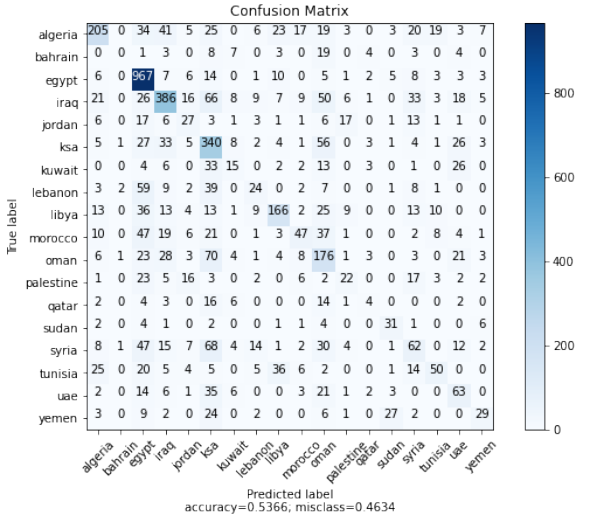

Figure 3: Confusion matrix of the predictions of the MARBERT Prompt model in subtask 1 on the dev-set.

5 Conclusion

In this paper, we have presented our work submitted to NADI shared task. Our proposed solution is an ensemble of different BERT-base models. These Models are developed differently, some are MTL models, fine-tuned models, or prompt-based models. The obtained results have shown that our proposed models achieve good results in both subtasks, by achieving first place in subtask 1 and first place in subtask 2. future work will focus more on building a robust model to improve recognition of some dialects. Furthermore to investigate and find features that best discriminate dialects.

References

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

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2021a. ARBERT & MARBERT: Deep bidirectional transformers for Arabic. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7088–7105, Online. Association for Computational Linguistics.

Muhammad Abdul-Mageed, Chiyu Zhang, Houda Bouamor, and Nizar Habash. 2020. NADI 2020: The first nuanced Arabic dialect identification shared task. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 97–110, Barcelona, Spain (Online). Association for Computational Linguistics.

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

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, Houda Bouamor, and Nizar Habash. 2022. NADI 2022: The Third Nuanced Arabic Dialect Identification Shared Task. In *Proceedings of the Seven Arabic Natural Language Processing Workshop (WANLP 2022)*.

Badr AlKhamissi, Mohamed Gabr, Muhammad El-Nokrashy, and Khaled Essam. 2021. Adapting marbert for improved arabic dialect identification: Submission to the nadi 2021 shared task. *arXiv preprint arXiv:2103.01065*.

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

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

Wissam Antoun, Fady Baly, and Hazem Hajj. 2021b. AraGPT2: Pre-trained transformer for Arabic language generation. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 196–207, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Ning Ding, Shengding Hu, Weilin Zhao, Yulin Chen, Zhiyuan Liu, Hai-Tao Zheng, and Maosong Sun. 2021. Openprompt: An open-source framework for prompt-learning. *arXiv preprint arXiv:2111.01998*.

Abdellah El Mekki, Abdelkader El Mahdaouy, Kabil Essefar, Nabil El Mamoun, Ismail Berrada, and Ahmed Khoumsi. 2021. Bert-based multi-task model for country and province level msa and dialectal arabic identification. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 271–275.

Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2021. The interplay of variant, size, and task type in Arabic pre-trained language models. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, Kyiv, Ukraine (Online). Association for Computational Linguistics.

Wuwei Lan, Yang Chen, Wei Xu, and Alan Ritter. 2020. Gigabert: Zero-shot transfer learning from english to arabic. In *Proceedings of The 2020 Conference on Empirical Methods on Natural Language Processing (EMNLP)*.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

Hamada Nayel, Ahmed Hassan, Mahmoud Sobhi, and Ahmed El-Sawy. 2021. Machine learning-based approach for arabic dialect identification. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 287–290.

Samia Touileb. 2020. Ltg-st at nadi shared task 1: Arabic dialect identification using a stacking classifier. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 313–319.