NLPeople at NADI 2023 Shared Task: Arabic Dialect Identification with Augmented Context and Multi-Stage Tuning

Mohab Elkaref[1] and **Movina Moses**[2] and **Shinnosuke Tanaka**[1] and
James Barry[1] and **Geeth De Mel**[1]
IBM Research Europe[1] and IBM Research[2]
{mohab.elkaref, movina.moses, shinnosuke.tanaka, james.barry}@ibm.com
geeth.demel@uk.ibm.com

Abstract

This paper presents the approach of the **NLPeople** team to the Nuanced Arabic Dialect Identification (NADI) 2023 shared task. Subtask 1 involves identifying the dialect of a source text at the country level. Our approach to Subtask 1 makes use of language-specific language models, a clustering and retrieval method to provide additional context to a target sentence, a fine-tuning strategy which makes use of the provided data from the 2020 and 2021 shared tasks, and finally, ensembling over the predictions of multiple models. Our submission achieves a macro-averaged F1 score of 87.27, ranking 1st among the other participants in the task.

1 Introduction

The task of dialect identification involves predicting the source variety of a given text or speech segment. Recently, there have been a number of shared tasks that have focused on predicting the nuanced dialects of Arabic (Abdul-Mageed et al., 2020, 2021, 2022). Arabic can be broadly categorised into the following three languages: Classical Arabic (CA), Modern Standard Arabic (MSA), and Dialectal Arabic (DA), where DA can be further sub-categorised based on the geographic region where it is spoken.

Arabic dialect identification represents a challenging task for a number of reasons. Firstly, Arabic languages exhibit rich morphology, where words are highly-inflected, which can lead to issues related to data sparsity. Another challenge present in the NADI shared tasks, is that the text to be classified consists of tweets, a form of user-generated content (UGC). As pointed out by Cassidy et al. (2022), UGC contains features not typically found in other forms of text data such as spoken language and standardised written language. For instance, UGC in the form of tweets tend to be short, exhibit non-standard use of grammar, and contain increased usage of emojis and abbreviated text.

This paper describes the **NLPeople** submission to the 2023 NADI shared task (Abdul-Mageed et al., 2023). In order to deal with the challenge of Arabic dialect identification, we develop a system which makes use of the following components:

- **Language-specific language models:** We utilise language models trained on Arabic and Arabic UGC.
- **Additional context retrieval:** We retrieve similar texts from a reference set for a given target text and append the retrieved text and corresponding labels as additional input.
- **Staged fine-tuning on additional data:** We first perform generic fine-tuning on the 2020 and 2021 data that was made available to participants, followed by a final round of fine-tuning on the 2023 data.
- **Model ensembling**: We combine the predictions of numerous models.

We empirically show that each of these components improves upon the metric of macro-averaged F1 score over the included dialects. Overall, our results rank 1st among 16 participants with a macro-averaged F1 score of 87.27.

2 Dataset

The label distribution of the used datasets are given in Figure 1. For the NADI-2023 data, a total of 18 country-level labels are present, and the training and development data have an equal distribution of 1000 and 100 labels, respectively. Additionally, we include the NADI-2020 and NADI-2021 datasets that were released by the shared task organisers as additional data for training our models. These datasets exhibit an imbalanced label distribution compared to the NADI-2023 data, with the UAE label being absent, and certain dialects such as Bahranian and Qatari being less represented than

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 642–646

dialects such as Egyptian and Saudi Arabian. The total number of unlabelled instances in the 2023 test set is 3600.

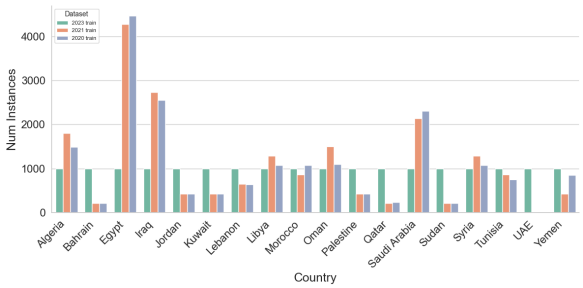

Figure 1: Number of instances per dialect across the 2023, 2021 and 2020 training data.

3 System Description

3.1 Initial System

In this section we discuss the NLPeople system. At its core, our model relies on a Transformer encoder model (Vaswani et al., 2017) to encode a sequence of words into a sequence of hidden states, which are passed to a feedforward network to predict the label. More formally, given a sentence $X = x_1, \ldots, x_n$ containing n words, a pre-trained language model LM is used to extract features $[x^l_{CLS}, x^l_1, \ldots, x^l_n] = LM^l([CLS], x_1, \ldots, x_n)$, where l is the last layer of the encoder, and x^l_i is the layer-l vector corresponding to the first word-piece in the word x_i. We take the output vector corresponding to the special [CLS] token x^l_{CLS} and pass this vector into a two-layer feedforward network to produce scores for all possible tags.

Model hyperparameters are given in Table 1. The models were trained on an NVIDIA A100 GPU with 80GB of VRAM. Training took around 1.5 hours for the 2023 training data.

Hyperparameter	Value
Learning Rate	1e-5
Batch Size	8
Transformer embedding size (base)	768
Transformer embedding size (large)	1024
Feedforward Size	768
Num. Feedforward Layers	2
Feedforward activation (first-layer)	ReLU
Dropout Rate	0.3
Epochs	10

Table 1: Model Hyperparameters

3.2 System Enhancements

Language-specific Language Models In order to deal with the morphological complexity of the Arabic dialects, we utilise pre-trained language models trained on Arabic. In particular, we experiment with the `MARBERTv2`[1] and `bert-large-arabertv02-twitter`[2] models. In the case of the `bert-large-arabertv02-twitter` model, it is trained on Twitter data which should be similar to the domain of the shared task data.

Additional Context Retrieval Given that the shared task data consists of short texts in the form of tweets, we experiment with adding context to the input data. For a given target item, which in this case can be a text instance from the training, development or test set, we retrieve the top-k most similar texts from the training data. Specifically, the fine-tuned `MARBERTv2` model is employed to obtain dense vectors for all instances in the training, development and test data, and for a given target item, instances from the train set with the k-nearest Euclidean distances are appended after the target text. In the additional context, the corresponding labels of the retrieved items are also included as special tokens. The augmented instances are shown below where we refer to x_i as a target text, y_i as the target label, and x_{top_j} and y_{top_j} represent the top-jth retrieved item's text and label, respectively:

$$x_i, [y_{top_1}]x_{top_1}, \ldots, [y_{top_k}]x_{top_k} = y_i$$

Training and evaluation then proceeds as normal using the augmented train, development and test sets.

Staged Fine-tuning on Additional Data Along with the 2023 training and development data, the shared task organisers provided participants with training data from the 2020 and 2021 shared tasks. We conduct a number of experiments involving the mixture of data to use for model training, and also consider a staged fine-tuning approach where the model is first fine-tuned on the data from the previous years, and is then fine-tuned on the current 2023 data.

Model Ensembling We consider model ensembling via two approaches: 1) *score ensembling*

[1] https://huggingface.co/UBC-NLP/MARBERTv2

[2] https://huggingface.co/aubmindlab/bert-large-arabertv02-twitter

Model	Type	Macro F1
arabertv02	MLM	76.62
arabertv02-twitter	MLM	80.61
AraT5-base	Gen	75.67
AraT5-tweet-base	Gen	78.53
JABER	MLM	78.95
MARBERT	MLM	84.65
MARBERTv2	MLM	**86.05**
XLM-R	MLM	68.44

Table 2: Development scores using different pre-trained language models. MLM: masked language model, Gen: generative model.

where we stack the raw score predictions from multiple models and select the highest-scoring label, and 2) *label ensembling* where we perform majority voting on the predicted label for each test instance.

4 Results and Discussion

4.1 Development Experiments

Choice of Language Model The first set of experiments involve the choice of language model. The results are reported in Table 2. We considered two types of language models: masked language models (MLMs) and generative language models (Gen). In the former, the model is used to encode an input sentence which is then fed to a classifier component (Section 3.1). In the latter, the model is tasked with *generating* the output label in an auto-regressive manner given an input sentence.[3]

For the MLM models, when considering the arabert models, we note that the version trained on Twitter data performs better on the shared task data (80.61 vs. 76.62 F1). The MARBERT models perform the best among the Arabic language models, where the MARBERTv2 model has an F1 score of 86.05, the highest-scoring model overall. For the generative modelling approach, we tried various T5 variants, where the tweet content is fed as input and the model is tasked with generating the label. We also note that the variant of this model trained on Twitter data performs better (78.53 vs 75.67 F1). Finally, we consider a multilingual MLM baseline in XLM-R which performs worse than the Arabic language models with an F1 score of 68.44.

[3]To fine-tune the T5 models, we use the resources released by Nagoudi et al. (2022).

Context size	Macro F1
none	86.05
1	86.58
5	86.71
10	**86.79**

Table 3: Development scores using different counts for the number of retrieved texts.

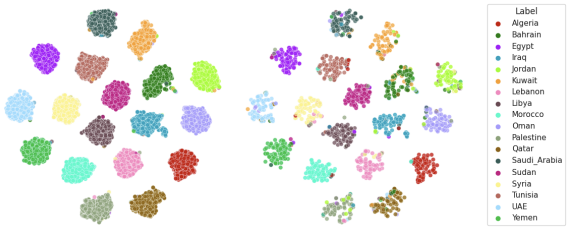

Figure 2: t-SNE visualisation of embeddings produced by fine-tuned MARBERTv2. The left plot corresponds to the training set, while the right plot corresponds to the development set.

Additional Context Retrieval The results concerning additional context retrieval are given in Table 3. We use the best-performing language model from the previous set of experiments, i.e. the MARBERTv2 model. Firstly, using the standard 2023 training data provides an F1 score of 86.05. By retrieving the top-1 most similar context, the score increases to 86.58. When retrieving the top-5 and top-10 most similar contexts to a target item, the score increases to 86.71 and 86.79 F1, respectively. To demonstrate the effectiveness of the retrieval, we present t-SNE plots depicting the embeddings of the training and development sets in Figure 2. Notably, distinct clusters form for each label, revealing that data points in proximity to target sentences often belong to the same cluster.

Staged Fine-tuning on Additional Data We experiment with using different variations of the provided data. The results are given in Table 4. We find that adding the 2020 data to the 2023 data harms performance when compared to training on the 2023 data alone, where the F1 score decreases from 86.05 to 83.51. The same is the case when adding the 2021 data to the 2023 data and adding both the 2020 and 2021 data to the 2023 data. In a final experiment, we first trained a model on the 2020 data, which was further fine-tuned on the 2021 data, and finally fine-tuned on the 2023 data. Interestingly, performing generic fine-tuning on the

Additional Data	Macro F1
2023	86.05
2023, 2020	83.51
2023, 2021	83.19
2023, 2021, 2020	83.01
Three-staged finetune	**87.02**

Table 4: Development scores using different sources of data.

Ensemble type	Count	Macro F1
none	1	86.05
score	5	86.78
score	10	**86.88**
label	5	86.07
label	10	86.74

Table 5: Development scores using different ensemble techniques.

noisier additional data followed by fine-tuning on the task-specific data results in the best-performing model with an F1 score of 87.02.

Model Ensembling To examine the effect of model ensembling, we utilised a selection of models that were trained as part of a hyperparameter sweep for the `MARBERTv2` model. The models were trained between 20-50 epochs, had a batch size of either 8 or 16, and used the CLS representation for classification. We consider two types of model ensembling: 1) score-ensembling where the scores of multiple models are stacked, and 2) label-ensembling where we perform majority voting on the predicted labels. Results are given in Table 5.

We find that combining model predictions is helpful in all cases. When considering score-based ensembling, the ensemble with 10 predictions performs best with a score of 86.88, which is the best score overall for this experiment. When considering label-based ensembling, the ensemble with 10 predictions performs best with a score of 86.74.

4.2 Official Results

Submitted System We trained up to 10 models for each setting using different random seeds through language model selection, additional context retrieval, staged fine-tuning, and combinations thereof. For the ensemble, from the pool of all trained models, we randomly selected between 2 and 20 models and recorded the development set score from the particular ensemble. We repeated this process until the highest-scoring ensemble was found. Details of the models used for the highest-performing system are presented in Table 6. This system employed `MARBERTv2` and `arabertv02-twitter` as language models, utilising both regular and staged fine-tuning techniques, resulting in remarkable performance through score ensembling. Unexpectedly, despite achieving high individual scores, additional context models were absent from this top ensemble. Individual model F1 scores ranged from 81.30 to 86.05 and extended to 89.56 through ensembling.

Language Model	Additional Data	Count	Macro F1 (range)
`arabertv02-twitter`	2023	5	81.30 - 81.90
`arabertv02-twitter`	Three-staged	3	81.47 - 81.49
`MARBERTv2`	2023	5	85.25 - 86.05
`MARBERTv2`	Three-staged	2	85.57 - 86.04

Table 6: 15 models used for the score ensemble which achieved the highest performance.

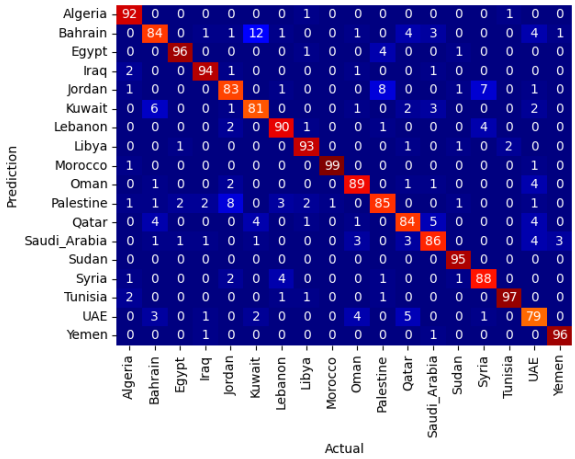

Figure 3: Confusion matrix for the submitted system on the development set.

The confusion matrix for the submitted system is shown in Fig 3. Among the 18 labels, it indicates that predictions are accurate for 90% or more for 9 of these labels. Particularly, Morocco achieves a remarkable accuracy by correctly predicting 99 out of 100 instances. On the other hand, UAE exhibits the highest error rate, with results falling below 80%. In the pair analysis, the most significant misprediction was observed, where 12% of Kuwait data was incorrectly labelled as Bahrain.

Results on the Test Set The official results on the final test set for the top five teams are presented in Table 7. Our system outperformed in not only

Team	Macro F1	Accuracy	Precision	Recall	Rank
rematchka	86.18	86.17	86.29	86.17	2
Arabitools	85.86	85.81	86.10	85.81	3
SANA	85.43	85.39	85.60	85.39	4
Frank	84.76	84.75	84.95	84.75	5
NLPeople (ours)	**87.27**	**87.22**	**87.37**	**87.22**	1

Table 7: Top five results on the test set from the official leaderboard.

F1 score but also across all other metrics.

5 Conclusion

In this work, we described the NLPeople submission to the 2023 NADI shared task (Abdul-Mageed et al., 2023). Our submission combines four different techniques: (1) language-specific language models (2), similar context retrieval (3), a staged fine-tuning approach over all available data, and (4) model ensembling. We demonstrated that each of the above components impacts our evaluation scores positively, and our final submission which uses the above techniques achieves a score of 87.27, which ranks 1st among 16 participants. Furthermore, our system is less impacted by the short input length due to our step of augmenting the input sentence with retrieved similar contexts.

Limitations

In the context of this study, it is essential to consider several limitations. Firstly, our retrieval methodology entails embedding the train, development, and test sets separately for the additional context retrieval method. This process imposes additional computational demands. Secondly, our adoption of staged fine-tuning introduces a similar computational overhead by training on more data. Furthermore, our findings have demonstrated that incorporating supplementary data adversely affects performance. Therefore, future works in this domain should carefully consider their data augmentation strategy, as indiscriminate inclusion of additional data may not yield improved results. Lastly, our ensemble approach, while effective, is computationally intensive. This technique may pose challenges in resource-constrained or time-sensitive scenarios where loading and maintaining multiple models concurrently may be impractical.

References

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

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

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

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

Lauren Cassidy, Teresa Lynn, James Barry, and Jennifer Foster. 2022. TwittIrish: A Universal Dependencies treebank of tweets in Modern Irish. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6869–6884, Dublin, Ireland. Association for Computational Linguistics.

El Moatez Billah Nagoudi, AbdelRahim Elmadany, and Muhammad Abdul-Mageed. 2022. AraT5: Text-to-text transformers for Arabic language generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 628–647, Dublin, Ireland. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.