Arabic dialect identification: An Arabic-BERT model with data augmentation and ensembling strategy

Kamel Gaanoun[1] **Imade Benelallam**[1,2]

[1]SI2M Lab, National Institute of Statistics and Applied Economics, Morocco
[2]AIOX Labs, Morocco
kamel.gaanoun@gmail.com, i.benelallam@insea.ac.ma

Abstract

This paper presents the ArabicProcessors team's deep learning system designed for the NADI 2020 Subtask 1 (country-level dialect identification) and Subtask 2 (province-level dialect identification). We used Arabic-Bert in combination with data augmentation and ensembling methods. Unlabeled data provided by task organizers (10 Million tweets) was split into multiple subparts, to which we applied semi-supervised learning method, and finally ran a specific ensembling process on the resulting models. This system ranked 3rd in Subtask 1 with 23.26% F1-score and 2nd in Subtask 2 with 5.75% F1-score.

1 Introduction

With the increasing internet access to Arab populations, their contributions to internet content are growing in a remarkable way. Indeed, the internet penetration rate increased from 30.3% to 51.6% between 2012 and 2019 in Arab countries (ITU 2019) . Additionally, there are multiple social networking platforms facilitating the sharing of content for all users. This has led to the appearance of Arabic dialect on internet platforms, which for a long time remained limited to oral conversations of everyday life, unlike Modern Standard Arabic (MSA), which is the only structured Arabic language that serves as the official language of writing and communication in all Arab countries. As a result, there is a growing interest in the treatment and exploitation of these dialects, which differ substantially from MSA and also differ between different countries (Zaidan & Callison-Burch 2013).

We can differentiate between two types of works related to Arabic dialects identification (DID): coarse-grained and fine-grained, where the former focuses on binary classifications (Aridhi et al. 2017, Elfardy & Diab 2013), or large groups of dialects such as Egyptian, Gulf, Iraqi, Maghrebi and Levantine (Habash 2010, Lulu & Elnagar 2018, Zampieri et al. 2017). Works for fine-grained identification has widened the field by incorporating a multitude of dialects reaching, among others, 17 Arab countries (Shon et al. 2020) and 25 different cities (Bouamor et al. 2019, Salameh et al. 2018).

The Nuanced Arabic Dialect Identification (NADI) shared task (Abdul-Mageed et al. 2020) reinforces this type of DID by offering two subtasks, namely Subtask 1 (country-level dialect identification) with 21 different countries dialects, and Subtask 2 (province-level dialect identification) with 100 different province dialects. In this paper, we present our contribution for both subtasks. Indeed, we confirm in our work the performance of the BERT models, which, contrary to traditional statistical methods or classical machine learning techniques, have not been often used for DID problems. In this paper, we expand upon the work of (Zhang & Abdul-Mageed 2019) with an Arabic specific BERT model and a new data augmentation approach concluded with an ensemble model. This approach allowed us to go from an F1-score=15.41% for the baseline model to an F1-score=25.01%; an improvement of 9.1 percentage points. The semi-supervised stage contributed with an improvement of 1.56 percentage points (from a score of 23.45%) scores obtained on the DEV dataset, see Results section.

In the next sections, we describe used data in Section 2, describe our system in Section 3, present our results in Section 4, discuss the data in Sections 5, and finally summarize our work in Section 6.

2 Data

2.1 Distribution

Datasets have been provided for the two subtasks: TRAIN for model training, DEV for evaluation and Unlabeled-10M, intended for the improvement of the systems. While the first two were made available to the participants, the third was to be crawled directly using the Twitter API. Indeed, the organizers provided us with the tweets IDs that we crawled up afterwards. This process allowed us to retrieve 9,999,978 tweets in total. By analyzing the content of these tweets, 3,184,508 tweets were unavailable, resulting in 6,815,470 tweets for this dataset.

As for the content of the TRAIN and DEV datasets, they contained the tweet IDs, their texts, the two variables to be predicted, namely country labels (Subtask 1) and province labels (Subtask 2), for respectively 21 countries and 100 provinces. Table 1 shows the statistics for the different datasets.

Finally, the organizers also provided us with the unlabeled TEST dataset, intended for the final evaluation of the system. The score obtained on this dataset was used for system ranking.

Dataset	**Number of tweets**	**Country labels**	**Provinces Labels**
TRAIN	21,000	21	100
DEV	4,957	21	100
Unlabeled-10M	9,999,978[1]	-	-
TEST	5,000	-	-

Table 1. Distribution of NADI 2020 datasets

For both country and province labels, the distributions were unbalanced, with a dominance of the following countries: Egypt, Iraq, Saudi Arabia and Algeria with a combined proportion of 52% in TRAIN and 53% in DEV. This finding is reflected at the level of the provinces, since the latter countries are respectively represented by 21,12,10 and 7 provinces, while the rest contain 1 to 6 provinces. See Figure 1 for distribution:

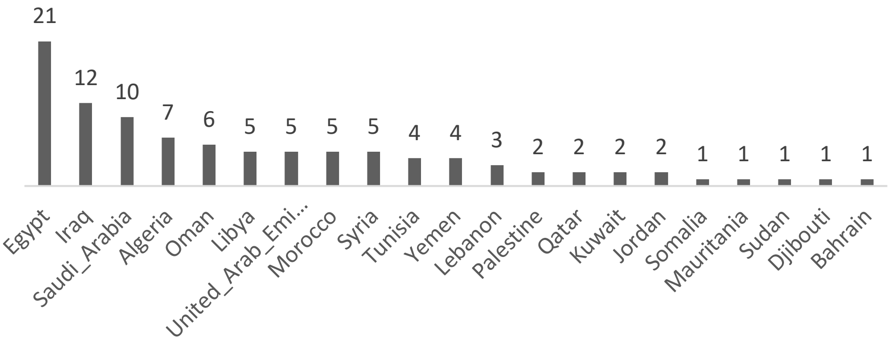

Figure 1. Provinces distribution per country

2.2 Preprocessing

We have defined different kinds of duplicate tweets, as follows:

- Islamic Duas (common Islamic invocations) duplicates, which are common to all countries, tweeted either directly by the user or via other platforms, like du3a.org, Gharedly.com, athkarapp.online, etc.

- MSA or classical Arabic text common for all Arabic countries like تصبحون على خير، بسم الله الرحمن الرحيم

- Common expressions in some dialects like, الله يبارك فيك(Morocco, Saudi_Arabia, Libya...)

[1] 3,184,508 tweets were unavailable, thus only 6,815,470 unlabeled tweets were used for this dataset

- Real duplicates specific to some dialect for the same country like محدش استحمل الجزء السيء مني، مرغوب فيا وانا لطيف بس.

We have therefore decided to delete (i) all tweets coming from other platforms, as they are not dialect specific, (ii) duplicate MSA tweets, (iii) real duplicates with same country. Hence, we removed tweets that could mislead the model's learning, since the same text is attributed to different entities (country or province). This step resulted in the deletion of 674 tweets from Train dataset.

The remaining preprocessing techniques consisted in removing usernames and hashtags, but keeping Arabic hashtags. We also removed words with the "Retweet" pattern, links, and numbers. Further, the Arabic text was standardized by removing diactrics, punctuation, and repeated characters. We also lowercased Latin characters and removed all other symbols.

3 System

The adopted system consists of three main steps, namely, (i) BERT-type model training, (ii) data augmentation with semi-supervised learning, and (iii) ensembling methods. These three steps will be described in this section.

3.1 BERT model

BERT (Bidirectional Encoder Representations from Transformers) is a language model that has proven its superiority in the field (Devlin et al. 2019). One of the strong points of this model is the possibility to fine-tune a pre-trained model on a new problem by simply adding a new output layer. This advantage was exploited in our system by adopting a pre-trained BERT model augmented by a multiple classification output layer.

The original BERT models named BERT-Base and BERT-Large were trained on a large English corpus, and since then several other language specific BERT models have been developed.

We chose to build our system using an Arabic-specific BERT model, i.e. a model that mimics the original BERT's architecture, but pre-trained on Arabic text. To this end, we compared two available models, namely AraBERT (Antoun et al. 2020) and Arabic-BERT (Safaya et al. 2020), and chose to use Arabic-BERT which obtained the best score on our problem. A brief comparison of the two models is given in Table 2

	AraBERT	**Arabic-BERT**
Corpus size	~23GB of text with ~3B words	~95GB of text with ~8.2B words
Training corpora	Arabic Wikidumps, The 1.5B words Arabic Corpus (El-Khair 2016) , The OSIAN Corpus, Assafir news articles, 4 other manually crawled news websites	Arabic version of OSCAR - filtered from Common Crawl, Recent dump of Arabic Wikipedia

Table 2. Brief comparative descriptive of Arabic-BERT and AraBERT

Table 3 summarizes the details of the infrastructure and used hyperparameters.

GPU	Tesla P100-PCIE-16GB
Language	Python 3.6.9
Main librairies	Hugging Face Transformers 2.5.1, Torch 1.5.1, Sklearn 0.22.2, Pandas 1.0.5, Numpy 1.18.5, bayesian-optimization-1.2.0
Bert Hyperparameters	Epochs: 3 , Batch size: 80 , Learning rate: 2e-5, Embedding maximum length: 128
Training average time	Initial TRAIN data: 10 minutes unlabeled-10M subparts: 180 minutes

Table 3. Used infrastructure and hyperparameters

3.2 Data Augmentation

Before the use of Unlabeled-10M, a first data augmentation was carried out starting with TRAIN. This operation consisted of taking all the tweets from the TRAIN, splitting them into three parts, then mixing them for each given country; this created new text while keeping the vocabulary of the dialect.

We called the model based on these data "Mix"; this model has proven to be efficient since it will appear in all the retained ensemble models, and more precisely, the model with the best score, as will be discussed in the results section.

We then used Unlabeled-10M to improve the predictive performance of our system using a semi-supervised method, defined by the following steps:

a. Unlabeled-10M was subdivided into 5 subparts

b. Label prediction for subpart 1 with our initial Arabic-BERT model based on TRAIN data.

c. Extraction of the tweets with the best softmax probabilities:

- For the majority countries (Egypt, Iraq, Saudi Arabia) we use the probabilities above the 99th percentile.

- For the other countries we use probabilities above the 80th percentile.

Indeed, we noticed that the model already scored well on the majority countries, but still needed improvement for the other countries, which is why it was decided to further augment the minority countries. This approach improved the minority countries' scores and thus the overall score of the model. That said, for subtask2, probabilities above the 90th percentile were taken for all provinces.

We point out that a total of 1,056,984 tweets were retained from Unlabeled-10M.

d. Concatenation of the predictions of the current subpart with the initial TRAIN data

e. Training a new model on this new data

For this step we train two models each time, namely a model based on the initial Arabic-BERT, and a model based on our initial model trained on the TRAIN data. The better of these two models will be compared to the model obtained for the previous sub-part (see step f, below).

f. Compare the obtained score with the last model's score and select the one that provided the best score for the prediction on the next subpart.

g. Repeat steps b,c,d,e,f for the following subparts with the selected model.

Our described model is a modified version of that done by (Zhang & Abdul-Mageed 2019), who used self-training to augment their model. By using self-training we risk falling into the problem of "catastrophic forgetting", i.e. our initial model may forget some of its original learning by forcing itself to learn the new data. Added to this, training and initial prediction errors may also become more pronounced. We also opted to subdivide the unlabeled-10M into several parts before applying our process to avoid a considerable amount of processing and resource use time in case of repetitive use of all the data, and also to be able to ensemble the different models obtained on each sub-part.

3.3 Ensemble methods

Once the data augmentation step is completed, we proceed to the ensembling of the different obtained models. In fact, we have a list of models containing data augmentation models in addition to our initial model Arabic-BERT and the MIX model. These different models will be ensembled according to the following methods:

- Hard majority voting
    a Original approach

The first method used is hard majority voting, which consists of taking for each tweet the label that was most often predicted by all the models. That's the statistical mode of predicted labels distribution.

b Modified approach

The original mode-based approach poses a problem if there is an equal number of predictions for some labels. Indeed, if we take the example where labels 1,3 and 5 have all been predicted 3 times, the mode calculated in this case (with stats scipy's library on python) will be equal to 1, i.e. the smallest label.

We have modified the mode calculation to change this behavior by creating a modified version of the mode computation in scipy. This version consists of forcing the mode to be equal to the label corresponding to the majority country (Egypt, Iraq or Saudi Arabia) when it is predicted at the same frequency as other minority countries. For example:

On all the predictions, the most predicted countries were as follows: Djibouti (label 1), Egypt (label 3), and Jordan (label 5), the modified mode will be equal to the label 3 corresponding to the majority country (Egypt) instead of 1.

- Soft voting

a Unweighted approach

While hard voting is based on predicted labels, soft voting takes into account the softmax probabilities assigned to each label., The probabilities of each label are summed across the different models, then the label with the highest average probability will be assigned by the ensemble model.

b Weighted approach

The unweighted method does not take into account the performance of the sub-models. We therefore proceeded to weight the models according to their score (larger weight for larger score). The choice of weights for this method was first done manually, followed by Bayesian optimization to maximize the F1-score.

We draw attention to the fact that these different methods were applied not only to all the models, but also to all their possible subsets. Indeed, we generate all possible combinations of the models before putting them together and retain the sub-set with the highest F1-score.

4 Results

We oriented the development of our system towards the use of the BERT model after performing a comparative study between the chosen BERT model and machine learning models; namely Naive-Bayes, Logistic Regression and XGBoost. These last three were all tested with a CountVectorizer (word counts) as an input, and TF-IDF (word and char levels with 2 and 3 ngrams). The best result was obtained with the XGBoost model with an F1-score=15.41%, but was still less performant than the chosen BERT. We consider XGBoost as the baseline model for our comparisons. Table 4 summarizes the results of the machine learning and BERT models on the DEV data for subtask 1.

Model	F1-Score
Naive Bayes	8.58%
Logistic Regression	14.30%
XGBoost	15.41%
Arabic-BERT	23.45%

Table 4. Arabic-BERT compared to other machine learning models (discarded TF-IDF based models results due to their low scores)

- **Subtask 1**

For subtask 1, four model variants were submitted for development data and three for test data.
Once the prediction step on the subparts of the Unlabeled-10M was completed, we proceeded to try different ensemble methods and compared the results obtained on the development and test data.

Ensembling proved to be an improving element in the predictive power of the system, but the score varied depending on the method used. A first comparison between soft voting and hard majority voting lends an advantage to the latter, based on the mode (the label most often predicted by the ensemble models) with a score of 24.15% Vs 24.01% compared to the soft voting method. Modifying the hard voting by changing the mode's calculation increased the score to 24.32%. It is this method that recorded the best submitted score with an F1-Score=23.26%, we call it BestTest. However, it is by adopting the weighted version of the soft voting that we obtain the best score on the development data with respectively 25.006% with manually defined weights, and 25.01% using Bayesian optimization, this model will be called BestDev. (See Table 5)

Model variant	DEV F1-Score	TEST F1-Score
Basic Hard Majority Voting	24.15%	22.52%
Modified Hard Majority Voting	24.32%	23.26%
Basic Soft Majority Voting	24.01%	23.21%
Manual Weighted Soft Majority Voting	25.006%	23.07%
Bayesian Weighted Majority Soft Voting	25.01%	23.03%

Table 5. Subtask 1 scores

- **Subtask 2**

For subtask 2, we applied a similar process as for subtask 1, except that the ensembling did not result in an improvement. We have retained the subpart of the Unlabeled-10M that obtained the best score, namely an F1-score=4.72% for the development data corresponding to an F1-score = 5.75% on the test data.

5 Discussion

	Majority countries	Minority countries
BestTest	0.63	0.93
BestDev	1.10	1.67

Table 6. F1-score enhancements

One of the challenges we encountered during the creation of our system was the low quality of predictions of minority labels. As illustrated for subtask 1 (similar finding for the provinces) in Table 6, it appears that our semi-supervised method combined with the ensemble method had a higher impact on minority labels, with an improvement reaching +1.67 percentage points for minority labels compared to +1.10 percentage points for majority labels (Egypt, Iraq, Saudi Arabia, Algeria). However, the problem still persists since we note that our models are unable to predict certain dialects such as Bahrain, Qatar, Kuwait. It would therefore be judicious in future work to improve our data augmentation method by focusing even more on these minority labels. Additionally, our MIX model did not take this into consideration; the mixing of tweets did not favour the augmentation of minority labels, and should be explored in the future.

Our models also encountered the problem of tweets written in MSA, which makes their predictions confusing since these tweets are not specific to any dialect. It would be a good idea to have our models rerun after removing all MSA tweets. This would require the creation of a pre-model for the detection of MSA text.

Finally, we can still explore other ways to improve BERT-based models by using the embedding obtained with BERT as input for CNN networks, a method that has already proven its efficiency as at (Kim 2014, Zheng & Yang 2019).

6 Conclusion

In this paper, we have described our contribution to NADI shared task 2020 subtasks. Subtask 1 dealt with Arabic dialects identification (DID) at the country level (21 Arab countries) and subtask 2 focused on dialects by provinces (100 provinces). Our system is based on an Arabic-specific BERT model, a semi-supervised method for data augmentation and an ensembling of different models following the data augmentation. By recording an F1-score=23.26% at subtask 1 (ranked third) and 5.75% at subtask 2 (ranked second), this approach demonstrates the efficiency of our BERT models in the field of DID.

References

Abdul-Mageed, M., Zhang, C., Bouamor, H. & Habash, N. (2020), NADI 2020: The First Nuanced Arabic Dialect Identification Shared Task, *in* ‘Proceedings of the Fifth Arabic Natural Language Processing Workshop (WANLP2020)’, Barcelona, Spain.

Antoun, W., Baly, F. & Hajj, H. (2020), Arabert: Transformer-based model for arabic language understanding, *in* ‘LREC 2020 Workshop Language Resources and Evaluation Conference 11–16 May 2020’, p. 9.

Aridhi, C., Hadhemi, A., Souissi, E. & Younes, J. (2017), Word-level identification of romanized tunisian dialect, pp. 170–175.

Bouamor, H., Hassan, S. & Habash, N. (2019), The MADAR shared task on Arabic fine-grained dialect identification, *in* ‘Proceedings of the Fourth Arabic Natural Language Processing Workshop’, Association for Computational Linguistics, Florence, Italy, pp. 199–207. https://www.aclweb.org/anthology/W19-4622

Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. (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)’, Association for Computational Linguistics, Minneapolis, Minnesota, pp. 4171–4186. https://www.aclweb.org/anthology/N19-1423

El-Khair, I. A. (2016), ‘1.5 billion words arabic corpus’, *ArXiv* **abs/1611.04033**.

Elfardy, H. & Diab, M. (2013), Sentence level dialect identification in arabic, Vol. 2, pp. 456–461.

Habash, N. Y. (2010), *Introduction to Arabic Natural Language Processing*, Morgan & Claypool. https://ieeexplore.ieee.org/document/6813521

ITU (2019), Measuring digital development, facts and figures, Technical report, International Telecommunication Union.

Kim, Y. (2014), Convolutional neural networks for sentence classification, *in* ‘Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)’, Association for Computational Linguistics, Doha, Qatar, pp. 1746–1751. https://www.aclweb.org/anthology/D14-1181

Lulu, L. & Elnagar, A. (2018), ‘Automatic arabic dialect classification using deep learning models’, *Procedia Computer Science* **142**, 262–269.

Safaya, A., Abdullatif, M. & Yuret, D. (2020), Kuisail at semeval-2020 task 12: Bert-cnn for offensive speech identification in social media, *in* ‘Proceedings of the International Workshop on Semantic Evaluation (SemEval)’.

Salameh, M., Bouamor, H. & Habash, N. (2018), Fine-grained Arabic dialect identification, *in* ‘Proceedings of the 27th International Conference on Computational Linguistics’, Association for Computational Linguistics, Santa Fe, New Mexico, USA, pp. 1332–1344. https://www.aclweb.org/anthology/C18-1113

Shon, S., Ali, A., Samih, Y., Mubarak, H. & Glass, J. (2020), Adi17: A fine-grained arabic dialect identification dataset.

Zaidan, O. & Callison-Burch, C. (2013), ‘Arabic dialect identification’, *Computational Linguistics* .

Zampieri, M., Malmasi, S., Ljubešic, N., Nakov, P., Ali, A., Tiedemann, J., Scherrer, Y. & Aepli, N. (2017), Findings of the VárDial evaluation campaign 2017, *in* ‘Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)’, Association for Computational Linguistics, Valencia, Spain, pp. 1–15. https://www.aclweb.org/anthology/W17-1201

Zhang, C. & Abdul-Mageed, M. (2019), No army, no navy: BERT semi-supervised learning of Arabic dialects, *in* ‘Proceedings of the Fourth Arabic Natural Language Processing Workshop’, Florence, Italy.

Zheng, S. & Yang, M. (2019), *A New Method of Improving BERT for Text Classification*, pp. 442–452.