Arabic Dialect Identification Using BERT Fine-Tuning

Moataz Mansour **Moustafa Tohamy** **Zeyad Ezzat** **Marwan Torki**

Faculty of Engineering, Alexandria University
{moataz.mansour.07, moustafatohamy96, zeyad.3ezzat}@gmail.com,
mtorki@alexu.edu.eg

Abstract

In the last few years, deep learning has proved to be a very effective paradigm to discover patterns in large data sets. Unfortunately, deep learning training on small data sets is not the best option because most of the time traditional machine learning algorithms could get better scores. Now, we can train the neural network on a large data set then fine-tune on a smaller data set using the transfer learning technique. In this paper, we present our system for NADI shared Task: Country-level Dialect Identification, Our system is based on fine-tuning of BERT and it achieves 22.85 F1-score on Test Set and our rank is 5th out of 18 teams.

1 Introduction

Arabic is a very complex language, a fast-growing one with more than 400 million speakers around the world. Online social media platforms (Twitter in our case) are a very good place where Arabic speakers can communicate, express, and share their thoughts. It will be very useful if we can study people's behavior so we can help them in various ways, and dialect identification is the first step to identify a given country to study the behavior of its people.

Text classification is considered an easy task, not because it is simple in general, but it is considered one of the oldest research areas in Natural Language Processing (NLP). If we have two classes separated with obvious features, it is a fairly simple process, but in our task, we have 21 classes with a lot of common features so it gets a lot more challenging. We will discuss that in the data section.

We have found that the MADAR Arabic Dialect Corpus and Lexicon (Bouamor et al., 2018) is a data set quite similar to the one provided in the task but instead of 21 labels for countries, there are 25 city labels. Top-ranked systems proposed by Abu Kwaik and Saad (2019) on the MADAR data set were built on traditional machine learning algorithms. In our first trial we have used n-gram TF-IDF characters and words features for train linear and nonlinear Support Vector Machine, Bernoulli Naive Bayes followed by ensembling all of the three aforementioned classifiers, we reached a score of 18.1 on the development set of NADI Shared Task (Abdul-Mageed et al., 2020). Our final model was built by fine-tuning a pre-trained Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2018). The original trained model provided from Safaya et al. (2020) and trained on around 95 gigabytes of Arabic text from different resources, all the parameters of training will be described in the system section.

2 Data

The data set was created for this shared task from organizers. The provided training data set consists of 21,000 tweets with 21 different labels representing 21 (Egypt, Bahrain, Iraq, Saudi_Arabia, Algeria, Oman, Syria, United_Arab_Emirates, Libya, Morocco, Yemen, Tunisia, Lebanon, Jordan, Kuwait, Palestine, Qatar, Mauritania, Djibouti, Somalia, Sudan).

For reporting results, there are 4,957 labeled tweets for development, 5,000 for testing, and an extra 10 million tweets for other purposes. We split training data into 19,950 tweets for training and 1,050 for validation so we can make use of most of the given data, for testing we have used development test.

Proceedings of the Fifth Arabic Natural Language Processing Workshop, pages 308–312
Barcelona, Spain (Online), December 12, 2020

A simple pre-processing function was used to remove links, punctuation, and repeated characters. We also tried removing directs but the performance was slightly better without removing them.

Looking into the data set distribution in Figure 1, we can notice that the data is totally imbalanced. For example, Egypt's tweets represent 21.3% of the data but Bahrain represents 0.01% of the data set. What makes the task even more challenging is the similarity between classes as the Gulf countries (Bahrain, Kuwait, Iraq, Oman, Qatar, Saudi Arabia, and the United Arab Emirates) have a very similar dialect. It is already hard for humans to differentiate between them and this data represents 37% of the data set.

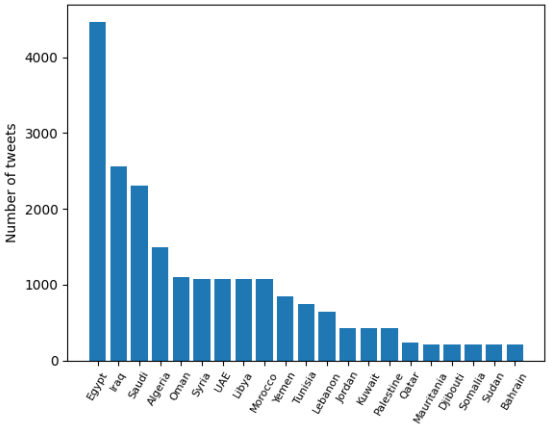

Figure 1: Training dataset labels distribution

3 Systems

In this section, we are going to introduce two systems with totally different approaches. The BERT system, which ranked as 5th in the leader board, and another system using feature extraction and ensemble of different Algorithms.

3.1 BERT System

BERT in its vanilla form is the English language model that can be used in different natural language processing tasks. It makes use of transformers which is an attention mechanism that tries to learn the contextual relations between words. We used a multilingual BERT (mBERT) base model with 12 hidden layers, 12 attention heads, 768 hidden layer size, and 110 million parameters. The Difference Between BERT and mBERT is that BERT is an English pre-trained language model but mBERT Provides pre-trained weights for Arabic and another 103 languages.

As mentioned earlier, we had used a pre-trained model and it was pre-trained on around 7.2 billion words from different resources such as the Arabic version of Oscar and recent Wikipedia Arabic articles. Our training process was divided into two phases: tuning the language model using 10 million tweets and then tuning using 19,950 tweets for the classification task.

Language Model Training : We have trained the model for one epoch on all unlabeled tweets, in the next section, we will provide more details about the training process. At first, we have truncated the long tweets and padded the shorter ones to get a 10-word length for all tweets. As described in Table 1, we had to add a drop out layer then linear layer mapping from the last hidden layer to vocabulary vectors After the BERT base model so we can do fine-tuning of it as a continuous bag of word model.

We have trained the model on 10 sessions and in each session, it was trained with 1 million tweets for one epoch. The training time for a session lasts around 12 hours as the optimizer used for training is Adam algorithm with weight decay (AdamW) with learning rate=2e-5, epsilon=1e-5, weight decay=0.001, and betas (0.9,0.999). All the training time was around 120 hours running on Azure virtual machine with a Tesla M60 GPU.

Classification Training: As described in table 1 We had to add a batch normalization, dropout, and linear layer mapping to classes. We had used the same optimizer, Adam, with decay with the same

parameters as before. We used the weights after the language fine-tuning. We trained the model for 4 epochs, It takes around 1-hour training on GeForce RTX 2060 GPU.

language model	classification
BERT Base model	BERT language model
Dropout layer with 0.1 drop_prop	Batch normalization with tanh activation function
linear layer	Dropout layer with 0.1 drop_prop
	Linear layer

Table 1: Modified BERT Model architecture

3.2 Traditional Machine Learning

Deep learning consumes a lot of time in training, however, it saves a lot of time spent on feature extraction. Here we present our experiment of feature extraction and traditional models to solve the provided task.

Feature Extraction: we have used a combination of n-gram characters and unigram words of TF-IDF features.

- TF-IDF for unigram with max_df=0.05,min_df=0.0001 after removing stop words contained in natural language processing toolkit library in python(NLTK)
- (2, 9) character n-grams with respect to boundaries, using sublinear transformation and maximum number of feature =40,000 (twice)

  These features have been chosen empirically after a lot of different trials.

Models: These three models have been chosen and ensemble in a voting classifier with equal weights.

- One vs one linear Support Vector Machine and balanced class weights
- One vs one non-linear Support Vector Machine with RBF kernel function and balanced class weights
- Bernoulli Naive Bayes with alpha 0.1

4 Results and discussion

For the task, we have built several systems to achieve the models described above. A comparison between the results described in Table 2 illustrates different ways used that helped us find the best solution that suits the task. Non-deep learning models with different features have a very similar performance. We tried different character n-grams with and without respect to boundaries. TF-IDF word features are not the best for this task and increasing the n-grams most of the time decreases the performance. BERT model without fine-tuning of the language model is still very powerful just by adding a classification layer and after one hour of training, it reaches a 23.5 F1 score. The only way we found to increase the performance is by using the extra unlabeled tweets because it is closer to the tweets from Wikipedia articles which our pre-trained model provided from Safaya et al. (2020) originally trained on.

By looking at the confusion matrix for BERT with fine-tuning of language model in Figure 2, we can notice the consequences of class imbalance and overlapping of the features. The most obvious one is in the gulf countries mentioned before, we can observe that the model predicted most of them as Saudi Arabia and did not predict any of Bahrain correctly.

Source code: `https://github.com/zeyad3ezzat/Nadi-Shared-Task`.

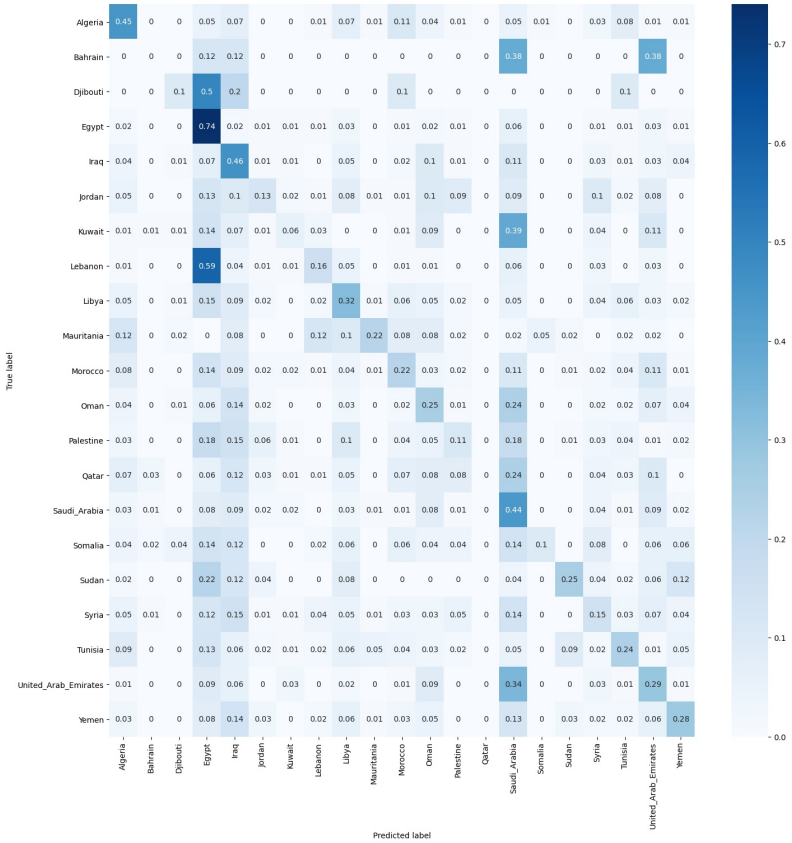

Figure 2: Confusion matrix of development set predictions

Models	Features		F1 score
	char_wb	word	
Non-linear SVM	(2,9)		17.1
Non-linear SVM		unigram	13.1
Non-linear SVM	(2,9)	unigram	17.2
Linear SVM	(2,9)	unigram	16.3
Bernolli NB	(2,9)	unigram	16.0
Voting classifier	(2,9)	unigram	**18.1**
	Deep Learning		
BERT	Without fine-tuning of language model		**23.5**
BERT	With fine-tuning of language model		**24.05**

Table 2: F1 Score of different models on development set

We could make use of the similarity between classes by combining similar dialects in a hierarchical classification approach. for example, a first classification level may be classified into(sham, gulf, Nile, others). The second level will be the normal 21 labels.

5 Conclusion

This paper described two methods applied to NADI shared sub-task 1 to predict an Arabic dialect from tweets. Our experiments show to show that the BERT model achieved the best F1 score betting other machine learning models. Using the unlabeled tweets to fine-tuning the language model is a great way to increase the performance but needs a lot of time and computation power. We had achieved a relatively high f1 score of 24.05 on the development set and 22.85 on the test set.

References

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 (WANLP 2020)*, Barcelona, Spain.

Kathrein Abu Kwaik and Motaz Saad. 2019. ArbDialectID at MADAR shared task 1: Language modelling and ensemble learning for fine grained Arabic dialect identification. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 254–258, Florence, Italy, August. Association for Computational Linguistics.

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, May. European Language Resources Association (ELRA).

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Ali Safaya, Moutasem Abdullatif, and Deniz Yuret. 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)*.