Mawdoo3 AI at MADAR Shared Task: Arabic Fine-Grained Dialect Identification with Ensemble Learning

Ahmed Ragab* **Haitham Seelawi*** **Mostafa Samir*** **Abdelrahman Mattar**
Hesham Al-Bataineh **Mohammad Zaghloul** **Ahmad Mustafa**
Bashar Talafha **Abed Alhakim Freihat** **Hussein T. Al-Natsheh**
Mawdoo3 Ltd, Amman, Jordan
ai@mawdoo3.com

Abstract

In this paper we discuss several models we used to classify 25 city-level Arabic dialects in addition to Modern Standard Arabic (MSA) as part of MADAR shared task (sub-task 1). We propose an ensemble model of a group of experimentally designed best performing classifiers on a various set of features. Our system achieves an accuracy of 69.3% macro F1-score with an improvement of 1.4% accuracy from the baseline model on the DEV dataset. Our best run submitted model ranked as third out of 19 participating teams on the TEST dataset with only 0.12% macro F1-score behind the top ranked system.

1 Introduction

The term Arabic language is better thought of as an umbrella term for a gamut of the language varieties, spanning the far and apart geographies constituting the Arab world, some of which are not even mutually intelligible (Palmer, 2007). Until recently, the standard variety referred to as Modern Standard Arabic (MSA), was the only socially acceptable form of written communication. However, with the advent and ever-increasing adoption of web 2.0 technologies in the day to day life of Arab societies, dialectical variants of Arabic came to dominate written Arabic online, even though they usually don't have a formalized orthography or grammar (Zaidan and Callison-Burch, 2014). As a consequence, the detection of such dialects is having an increasingly larger number of use-cases of service and communication personalization for services providers targeting Arabic speaking customers over the internet.

The paper describes our submitted system to the MADAR shared task (sub-task 1) (Bouamor et al., 2019). The task problem is to predict the Arabic dialect out of 26 class which include 25 city-level dialect in addition to MSA. The number of the participating team who submitted the prediction of their proposed system on the TEST dataset were 19 teams. Our proposed system was ranked 3rd in the shared task leader board with F1-macro score of 67.20%, and a difference of 0.12% from the winning system.

Our approach to the problem involves using TF–IDF features, both at the level of tokens and characters, augmented with class probabilities of a number of linear classifiers, and language model probabilities; all together as our set of potential features. For the classification system we developed for the sub-task, we used a standalone logistic regression model, and an ensemble of different types of classifiers, taking into a hard vote the prediction of each (i.e. we use the most probable class of each model instead of the full classes probabilities, to decide on the final prediction of the total ensemble). The choice of an ensemble system stems from the empirical evidence that on the whole, they perform significantly better than a single model (Dietterich, 2000).

In Section 2, we briefly present a previous work that was proposed to solve the same task and the same DEV dataset which is described in Section 3. The description of our proposed models is then discussed in detail in Section 4. Finally, the results of our models on the share task DEV and TEST datasets are discussed in Section 5 in comparison with both the baseline and the best performing model of the task.

2 Related Work

The closest work to our approach is presented in Salameh et al. (2018). The authors of that work proposed several classification methods and ex-

* These authors contributed equally to the work and ordered alphabetically on the first-name.

Proceedings of the Fourth Arabic Natural Language Processing Workshop, pages 244–248
Florence, Italy, August 1, 2019.

plore a large space of features to identify the exact city of a speaker. The task covers 25 cities from across the Arab World (from Rabat to Muscat), in addition to Modern Standard Arabic. The authors extract word n-grams ranging from unigrams to 5-grams and use them as features, in addition to character n-grams ranging from 1-grams to 5-grams. They computed TF–IDF scores. To boost up the accuracy they used language model to measure how close each sentence is to the dialect. For classification, they trained Multinomial Naive Bayes. The authors reported accuracy score of 67.9%.

3 Dataset

The dataset used for this shared task is the one provided by the Multi-Arabic Dialect Applications and Resources (MADAR). The task name is *MADAR travel domain dialect identification task*. This task is one of two sub-tasks presented and run in the Fourth Arabic Natural Language Processing Workshop (WANLP 2019)[1].The dataset is divided into two separate corpora; the first one is referred to as CORPUS-26 which consists of 25 city-level Arabic dialect in addition to MSA forming 26 dialect classes, with each of the 26 classes consists of 1,600 examples as training data and 200 examples per class as the DEV set. The second corpus, referred to as CORPUS-6, consists of 9,000 examples in 6 classes (5 cities plus MSA) as the training data and 1,000 for each of the 6 classes as the DEV set (Bouamor et al., 2018). Both corpora are annotated with the a code for the respective city dialect it represents.
Tokenizing on spaces, CORPUS-26, has a total of 294,718 words with 85,249 of them are unique, while CORPUS-6, has a total of 388,041 words with 63,860 of them are unique.

In Figure 1, we show the percentage of unique words, i.e. words that exclusively appear in the respective dialect class in the CORPUS-26 dataset. The figure also shows that most of the words in each class, appear in more than 4 of the other dialect classes, which in turn, help us choosing the set of features to build our model.

4 Models

The three models corresponding with the three submissions we made were mainly built upon:

[1]https://sites.google.com/view/wanlp-2019

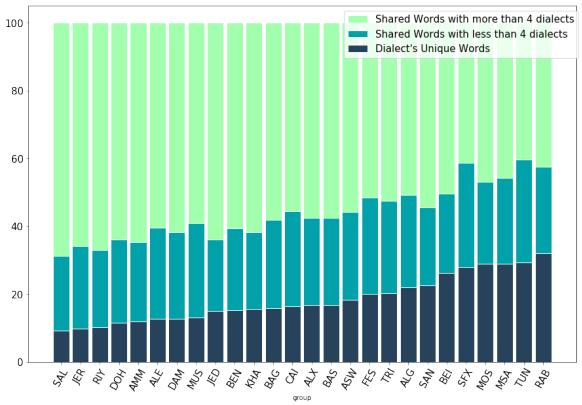

Figure 1: Words distribution among the 25 dialects and MSA sorted by the percentage of exclusive words.

i. TF–IDF vectorization of sentences

ii. Multinomial Naive Bayes classifier (MNB) similar to what is used in Salameh et al. (2018)

iii. The voting ensemble of multiple classifiers.

4.1 TF–IDF Features

We first preprocessed the data from CORPUS-26 by removing emojis and special characters. Then we extracted two sets of TF–IDF vectroized features: one on the words level, and the other on the character level.

Word n-grams: Word n-grams is one of the basic features used in dialect detection tasks and text classification tasks in general. We extracted word n-grams and vectorized the extractions in a feature vector using TF–IDF scores. Our experiments show that the feature vectors consisting of both word uni-grams and bi-grams result in more superior models than using any of them alone.

Character n-grams: While word n-grams are powerful features, they can suffer from a high out-of-vocabulary words (OOVs) rate when the testing set has a lot of varieties. This usually happens with Arabic text due to its morphological variance. Character n-grams on the other hand are able to mitigate this problem by capturing different parts of the word and hence reduces the effect of morphological segmentation on word similarities. We follow (Salameh et al., 2018) and use a TF–IDF vectorized feature set of character n-grams that range from 1-grams to 5-grams.

Moreover, we make sure that the extracted character n-grams respect the word boundaries; this has shown to perform better in our experiments in contrast to character n-grams that cross over the word boundaries.

We concatenate this feature vector into a bigger one that amounts to 236K features. This big vector is the main feature vector for our models.

4.2 Multinomial Naive Bayes (MNB)

In our base approach, we trained a multinomial naive Bayes classifier with additive smoothing to reduce the penalty of missing features in testing examples. While the smoothing parameter α is usually set to 1, our experiments showed that the best value for α was 0.1, which consists a Lidstone smoothing.

This setting achieved 68.3% accuracy on CORPUS-26 DEV set, which is 0.6% less than the best model in Salameh et al. (2018) although their model uses more features from dialectical language models. This MNB model was only used as a base for the other models that were submitted and it was not submitted itself.

4.3 Logistic Regression (LR)

Our second approach consisted of appending the class probabilities from the MNB model to the big features vector we constructed from word/character n-grams. This new feature vector is then fed into a logistic regression model with L2-regularization.

This 2-layered model improved about 0.04% over the MNB's accuracy. This suggests that more classifiers trained on the same feature vector can yield a bigger improvement by accumulating their smaller improvements, and this was the motivation behind our highest accuracy model.

4.4 Ensemble Model

Instead of training just an MNB model on TF–IDF features vector, we also trained a logistic regression model and weak dummy classifier used on prior probabilities of each dialect. The class probabilities from these three models were concatenated with the TF–IDF feature vector and the concatenation is then used for the second layer of the model.

In the second layer, instead of training just a logistic regression model, we included other classifiers to be trained on the TF–IDF plus probability features. In addition to the logistic regression, we trained:

i. Another MNB with one-vs-rest approach

ii. Support vector machine

iii. Bernoulli Naive Bayes classifier

iv. k-nearest-neighbours classifier with one-vs-rest approach and with samples weighted by distance

v. A weak dummy classifier based on prior probabilities of each dialect.

These classifiers were ensembled together by hard voting where we pick the dialect that was detected most by all the classifiers to be the final predicted dialect. This ensemble managed to score 69.3% in accuracy on CORPUS-26 DEV set.

4.5 Enemble with Language Model Scores as Features

We trained several language models (LMs) on character and word level using KenLM (Heafield, 2011) from Moses using default parameters. Twenty six character level language models were trained on CORPUS-26. We preprocessed the data to replace the spaces between words with special character and inserted spaces between characters so that each character is considered as a single token. Character based language models capture fine specifics of each dialect such as using the letter Meem (م) as a prefix of a verb and the letter Sheyn (ش) as a suffix negates the verb in the Egyptian dialect. Moreover, Character level LMs complement word based LMs by reducing the number of out-of-vocabulary words (OOVs). In addition to the 64 language models suggested by (Salameh et al., 2018) (i.e., twenty six 5-gram character-level LMs trained on CORPUS-26, twenty six 5-gram word-level LMs trained on CORPUS-26, six 5-gram char level LMs and six 5-gram word-level LMs trained on CORPUS-6), we added 26 bi-gram word level LMs trained on CORPUS-26 and 6 bi-gram word level LMs trained on CORPUS-6. Each sentence in training, DEV, and TEST data was scored by these 96 language models and we scaled the scores to 0-1 scale to lie within the same range of the other features, mainly TF-IDF. We used the scaled scores as input features to the classifiers.

Model	DEV		TEST	
	F1	Acc	F1	Acc
Baseline MNB	68.28	68.23	-	-
Run1: Ensemble	**69.33**	**69.28**	67.17	67.06
Run2: LR	68.32	68.27	66.37	66.37
Run3: Ensemble+LMs	69.16	69.11	67.20	67.08
MNB	-	68.90	**69.00**	**67.90**
ArbDialectID	-	-	67.32	67.29

Table 1: Results in terms of macro F1-score (F1) and accuracy (Acc) of our experimental baseline, our three models (i.e., runs) which are *Ensemble*, *LR* and *Ensemble + LMs* respectively, the best model of (Salameh et al., 2018) (MNB), and the top ranked system in MADAR shared task (ArbDialectID).

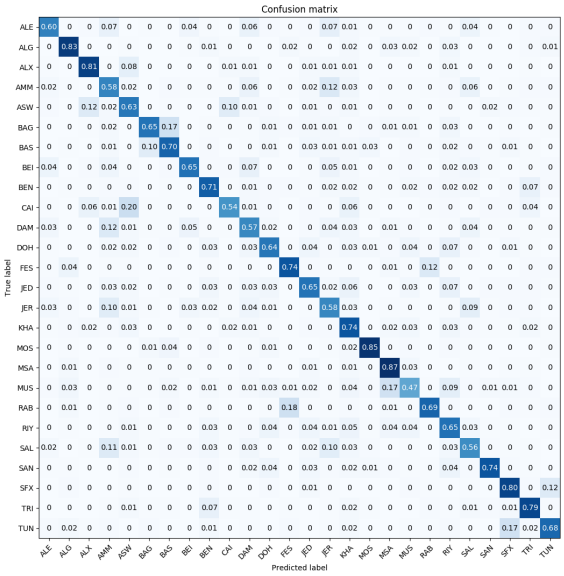

Figure 2: Normalized confusion matrix of our baseline MNB model on the DEV dataset.

5 Results and Discussion

In Table 1, we report the results of our models and Salameh et al. (2018) best model on the DEV and TEST sets using the macro F1-score and accuracy metrics. First, it is shown that our baseline MNB model have outperformed Salameh et al. (2018) exact counterpart model with the same set of features on the DEV set. We deem this as a result of the Lidstone smoothing of an α equal to 0.1 instead of 1, which we hypothesize that it reduced the noise to signal ratio in the 236k element feature vector by reducing the pseudo-count for the missing features which constitute the majority of the feature vector in comparison to the actual features present in the input text. It is also shown that the Ensemble model described in section 4.4 is the best scorer on the DEV set, although it was out performed by Salameh et al. (2018) MNB on the TEST set. Also on the contrary of Salameh et al. (2018) findings that the word uni-gram and the character n-grams ranging from 1-grams to 3-grams resulted in the best performing model on the DEV set, we have found that the word unigrams and bi-grams combined, alongside character n-grams ranging from 1-gram to 5-grams are the best performing features for our models.

It can be deduced from Figure 2 that the bulk of the error originates from the confusion between dialects within the same country or those that are very close geographically (e.g Cairo, Alexandria and Aswan dialects), the only exception to this would be the confusion between Mosul's dialect and MSA. This is demonstrated further by the best scoring Ensemble model on DEV which we hypothesize that its second layer managed to learn from the non-orthographic probability features of the first layer by detecting its biases and error distribution, thus enhancing upon it. We believe that a human benchmark might be useful for this fine-grained dialect detection problem, for which it would set a reasonable upper-bound that shows the significance of the orthographic features in determining the writer's dialect through the analysis of the human error.

6 Conclusion

We proposed a system for classifying 26 dialects of Arabic. Our system uses ensembles at the level of features and classifiers. At the feature level, we augment textual features extracted directly from text with class probabilities of a few linear classifiers. For the model level, we use an ensemble of a number of different discriminators. Our system achieved a macro F-1 score of 69.33% and 66.7% on the development and test sets of the MADAR Arabic Dialect Corpus, respectively. In the future, we plan on using word embedding as an extra set features to experiment with. This will focus on context aware word embedding such as ELMo (Peters et al., 2018), and BERT (Devlin et al., 2018).

References

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 Language Resources and Evaluation Conference (LREC)*, Miyazaki, Japan.

Houda Bouamor, Sabit Hassan, and Nizar Habash. 2019. The MADAR Shared Task on Arabic Fine-Grained Dialect Identification. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop (WANLP19)*, Florence, Italy.

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.

Thomas G. Dietterich. 2000. Ensemble methods in machine learning. In *Multiple Classifier Systems, First International Workshop, MCS 2000, Cagliari, Italy, June 21-23, 2000, Proceedings*, volume 1857 of *Lecture Notes in Computer Science*, pages 1–15. Springer.

Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation, WMT@EMNLP 2011, Edinburgh, Scotland, UK, July 30-31, 2011*, pages 187–197. Association for Computational Linguistics.

Jeremy Palmer. 2007. Arabic diglossia: Teaching only the standard variety is a disservice to students. *The Arizona Working Papers in Second Language Acquisition and Teaching*, 14:111–122.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics (ACL).

Mohammad Salameh, Houda Bouamor, and Nizar Habash. 2018. Fine-grained Arabic dialect identification. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 1332–1344, Santa Fe, New Mexico, USA.

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