Arabic Dialect Identification
with Deep Learning and Hybrid Frequency Based Features

Youssef Fares Zeyad El-Zanaty Kareem Abdel-Salam Muhammed Ezzeldin
Aliaa Mohamed Karim El-Awaad Marwan Torki

Faculty of Engineering, Alexandria University
{youssefe.fares, zeyadzanaty, karimamd95}@gmail.com,
{not.muhammedezz, aliaamohamedali284, kelawaad}@gmail.com
mtorki@alexu.edu.eg

Abstract

Studies on Dialectical Arabic are growing more important by the day as it becomes the primary written and spoken form of Arabic online in informal settings. Among the important problems that should be explored is that of dialect identification. This paper reports different techniques that can be applied towards such goal and reports their performance on the Multi Arabic Dialect Applications and Resources (MADAR) Arabic Dialect Corpora. Our results show that improving on traditional systems using frequency based features and non deep learning classifiers is a challenging task. We propose different models based on different word and document representations. Our top model is able to achieve an F1 macro averaged score of 65.66 on MADAR's small-scale parallel corpus of 25 dialects and Modern Standard Arabic (MSA).

1 Introduction

Dialect identification is the task of identifying the dialect of a particular segment of speech or text of any size (i.e., word, sentence, or document) automatically. The task of Arabic Dialect identification has attracted more attention recently. However, most efforts focus on a smaller and more distinct number of dialects, dialects by country rather than by city for example. Fine grained or city-based Arabic dialect identification is the more challenging task of not only classifying dialect by country but also by city. As such, the similarity between classes grows higher and the task grows more challenging.

Other efforts that did tackle such fine grained dialects and a larger number of classes have not explored the use of state of the art embedding models, language models and the use of deep learning in general.

The task remains challenging primarily because of the similarity between documents labeled with cities that are within the same country. The number of samples available for each class is 1,600 for each of the 26 cities given in Table 1 from (Salameh and Bouamor, 2018).

We report different data augmentation techniques used to expand the training set used. We also report the data analysis done on class similarity and model confusion from which we draw conclusions for suggested future work.

2 Data

The data used in all of the proposed system is one of the two parallel corpora made available by the Multi Arabic Dialect Applications and Resources (Bouamor et al., 2019) (MADAR) project: a 2,000-sentence parallel corpus with 25 parallel translations plus Modern Standard Arabic (MSA) which we will refer to as CORPUS-26 and the second corpus which has another 10,000 additional sentences translated to five selected dialects, which we will refer to as CORPUS-6.

The metrics reported for each model on CORPUS-6 or CORPUS-26 are trained on the same corpus for which the accuracy is reported. No more additional data is used except augmentations of the corpus used.

Data Preprocessing We apply a generic transformation that removes punctuation, diacritization and vowel elongation.

Data Augmentation Although there is no class imbalance, the number of samples per class and the fine grained classes were motivation to experiment with different data augmentation techniques. We used the following methods inspired by (Ibrahim et al., 2018)'s work to augment already existing documents:

- Unique Words Augmentation: for each document that contains a word repeated more than once, we remove duplicate words from it

Proceedings of the Fourth Arabic Natural Language Processing Workshop, pages 224–228
Florence, Italy, August 1, 2019. ©2019 Association for Computational Linguistics

Region	**Maghreb**				**Nile Basin**	**Levant**		**Gulf**		**Yemen**
Sub-region	Moroco	Algeria	Tunisia	Libya	Egypt/Sudan	South Levant	North Levant	Iraq	Gulf	Yemen
Cities	Rabat (RAB) Fes (FES)	Algeris (ALG)	Tunis (TUN) Sfax (SFX)	Tripoli (TRI) Benghazi (BEN)	Cairo (CAI) Alexandria (ALX) Aswan (ASW) Khartoum (KHA)	Jerusalem (JER) Amman (AMM) Salt (SAL)	Beirut (BEI) Damascus (DAM) Aleppo (ALE)	Mosul (MOS) Baghdad (BAG) Basra (BAS)	Doha (DOH) Muscat (MUS) Riyadh (RIY) Jeddah (JED)	Sanaa (SAN)

Table 1: Different region, sub-region, and city dialects in the MADAR dataset.

and create a new comment with only unique words.

- Random Mask Augmentation: for each document, we create a different new document by randomly removing up to 20% of the original document words.
- Random Swap Augmentation: for each document, we create a different new document by randomly swapping up to 20% of the original document words.
- Random Concatenation Augmentation: we choose two documents with few number of words at random and append them forming a new one with longer length.

We report that using data augmentation prevented over-fitting when using deep learning as we chose between applying different techniques or using the original document at random for each sample in each epoch.

For non deep learning models, we used such augmentation to increase the size of the data used to around quadruple the original number of documents, which resulted in a slight increase (close to 1%) in the baseline model accuracy.

3 Methodology

For such a complicated task we tried multiple approaches using different techniques to achieve the best results. We started by tuning the baseline given in (Salameh and Bouamor, 2018) which is a Multinomial Naive Bayes (MNB) using TF-IDF character + word features (without the KenLM language model). Experiments concluded with n-gram ranges of one to five for character features and one-gram for word features. A grid-search using (Pedregosa et al., 2011) was applied to the MNB which delivered an F1-score of **64.94%** on the dev-set.

We then took to deep neural networks, the models submitted are given in Table 2 and experiments that lead to those submission are given in Table 3. We did not observe much improvement over the baseline (MNB) until our first submission model.

3.1 LSTM + CharCNN, FastText embeddings + LSTM and Baseline Ensemble

It is an ensemble of three models, the first being an adaptation of the character-level model proposed in (Ali, 2018), which takes one-hot-encoded character features to multiple (five in our case) convolution layers with filter size of 256 -which is the same as the max length set for a sentence- preceded by a Gated Recurrent Unit (GRU) for context capturing of these features then a softmax layer for calculating log probabilities.

After multiple experiments and tuning we replaced the GRU with an LSTM. The second model we ensemble is another shallow network consisting of an embedding layer of fastText word embeddings(Mikolov et al., 2018) through a spatial dropout layer to avoid over-fitting, then through an LSTM, again for context capturing, but in this case for word features, then finally a softmax layer. The outputs of both softmax layers are averaged to give the final probabilities. We chose this approach to combine both character features and word features, this gave us the best result we could achieve on the dev-set with **63%** F1-score. After ensembling it with our MNB baseline (the third and final model) with weighted averaging, we surpassed the baseline achieving **66.1%** F1-score on the dev-set and ranked second among all of our submissions with an F1-score of **65.35%** on the test-set. All neural network models were built using Keras (Chollet et al., 2015). The full architecture can be seen in Figure 1.

Model	F1-Macro	Precision-Macro	Recall-Macro	Accuracy
ArbDialectID (Winning Team)	**67.32%**	**67.60%**	**67.29%**	**67.29%**
LSTM +CharCNN, fastText embeddings + LSTM, Baseline (1st submission)	65.36%	**66.07%**	65.38%	65.38%
Char TFIDF + WordTFIDF + NN, Baseline (2nd submission)	**65.66%**	65.79%	**65.75%**	**65.75%**
Bert + Document Pooling (3rd submission)	35.14%	42.61%	36.25%	36.25%

Table 2: Models submitted and their corresponding scores on the test-set.

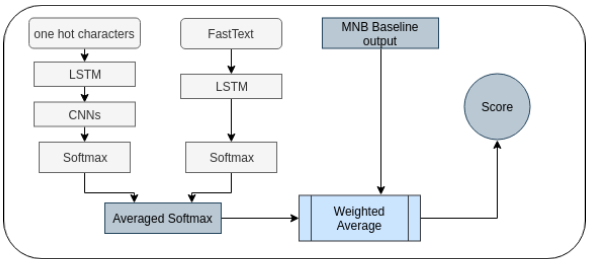

Figure 1: 1st Submission Architecture

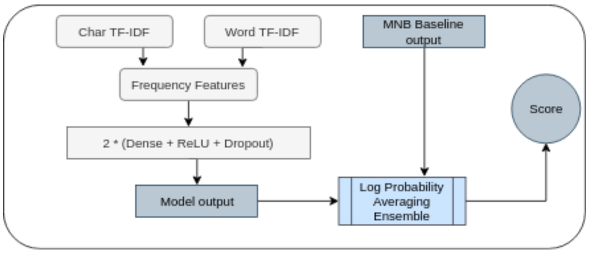

Figure 2: 2nd Submission Architecture

3.2 CharTFIDF + WordTFIDF + NN and Baseline Ensemble

It is an ensemble of the MNB baseline and a deep learning model applied to baseline features. The deep learning model takes as input the frequency based features for which the MNB achieved 64.94% dev-set F1-score and improves it to **65.57%**.

The model architecture in Figure 2 consists of two hidden fully connected layers followed by an output layer. The two hidden layers are followed by ReLU activations and dropout layers with 20% probability. The number of inputted features to the neural network is equal to the number of dimensions of the frequency based vectors (char-based and word-based). Adam optimizer (Kingma and Ba, 2014) is used for training with 3e-6 learning rate and the cross entropy loss as criterion.

The ensemble of the model produced with the baseline using log probability averaging produces **66.78%** dev-set F1-score and **65.66%** test-set F1-score which is less than 2% below the winning team results and was ranked the seventh out of 19 submissions in the shared task competition.

3.3 Language-Model Based Models

We propose a number of other systems that produced sub-optimal results on corpus-26 data, but are experiments worth mentioning towards other future ensembles and systems.

i. A character level forward and backward language model trained using multi-layer RNNs whose features are combined with fastText and bytepair (Heinzerling and Strube, 2017) subword embeddings produced 58% devset F1-score.

ii. A model using multi-lingual BERT (Devlin et al., 2018) and a multi-layer RNN for document representation also followed by a single layer linear classifier reaches 55% dev-set F1-score.

iii. A model using Aravec (Mohammad et al., 2017) word embeddings and a shallow LSTM for document representation (feeding word embedding sequence to LSTM and using hidden layers as features) produces 50% dev-set F1-score when using a one layer linear classifier.

4 Discussion

Multiple observations and experiments show that the fine-grained nature of classes is the most challenging aspect of the task. Differentiating between Cairo and Alex or Beirut and Damascus is a much harder problem than differentiating between Levant and Gulf for example. We report some results towards such conclusions when classifying by city within a single regions' data as shown in Figure 3 and Figure 4.

Bench-marking all of the fore-mentioned models on corpus-26 data with regions and MSA as classes instead of cities produces results comparable to that of corpus-6 data (80% at worst on the dev-set). So the higher scores reported on corpus-6 data are not only owing to the larger number of samples but also owing to the affinity between sub region classes in corpus-26.

Another conclusion we can draw from how

Model	**Dev-set F1-score**
Char TFIDF + WordTFIDF + NN, Baseline	66.6%
LSTM + CharCNN, fastText embeddings LSTM, Baseline	66.1%
Character-level bi-directional LM (RNNs) + fastText + BytePair, Linear Classifier	58%
Bert + RNN Document Representation + Linear Classifier	55%
AraVec Word Embeddings + Shallow LSTM with dropout	50%

Table 3: Top scoring models on the dev-set

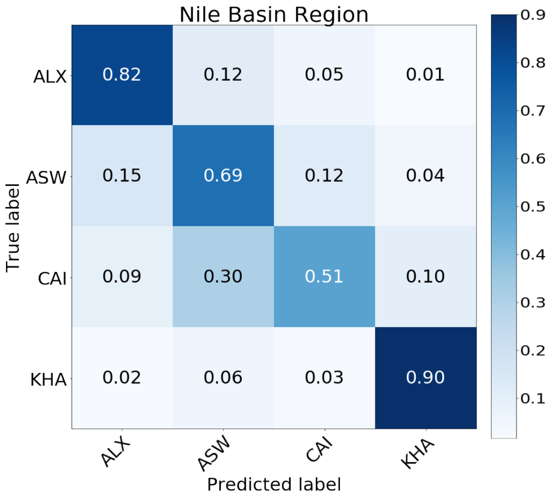

Figure 3: Confusion matrix for MNB classifier on Nile Basin region data and classes only

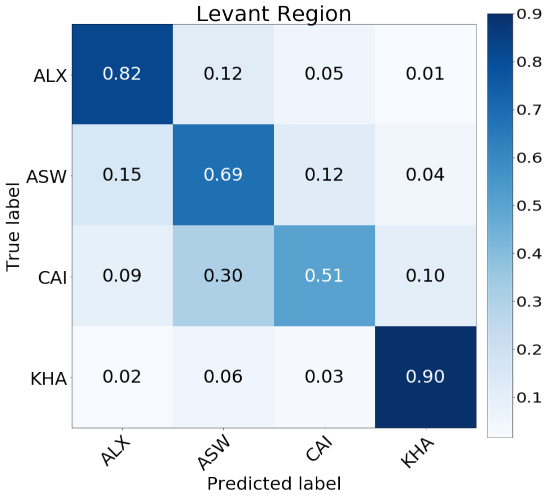

Figure 4: Confusion matrix for MNB classifier on Levant region data and classes only

closely all of the deep learning based models plateau, is that coming up with a better system for this task may require the use of other external labelled or unlabelled data. With the internet rich with blogs that are country specific or city specific. We can use unlabelled data from specific sources (e.g. tweets on Alexandria, Facebook posts from a public group based in Khartoum, and so on and so forth).

That can enable the training of embeddings from scratch on large data, and it can also be used on language model training improving the performance of models based on such techniques. The training of embeddings on such data specifically makes sense because of the percentage of out of vocab words and how they are handled in the embedding techniques we used. Because the embeddings were for the most part trained themselves on MSA data. The out of vocab (OOV) words which were usually 10-20% of the words in the corpus-26 data, were handled by averaging the rest of the embeddings of all words in the document or by being given a zero vector. Inconveniently, the OOV words are clearly the words we are most interested in because they are most likely to be the dialect specific words that differentiate between the classes. Therefore, if we are able to reduce the number of OOV words, the scores are expected to significantly improve. That can be achieved by the fore-mentioned training of embeddings on corpora that are not MSA only, or at least using smarter techniques to handle OOV words, such as character-based representation (Bojanowski et al., 2016).

5 Conclusion

We introduce multiple neural network based models built on word and document representations. We are able to produce results comparable to the MNB baseline on n-gram frequency based features despite of the small size of the dataset, which maybe an indication of even better results on larger data. We ensemble the neural network based models with the baseline to produce better results than the baseline.

Future work will explore further ensembles of the language model based classifiers and ensembling using other techniques than probability averaging (e.g. stacking). We will also explore the training of embeddings on data that is comprised of diverse dialectical data, not only MSA, and better handling of OOV words when using embeddings.

References

Mohamed Ali. 2018. Character level convolutional neural network for Arabic dialect identification. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pages 122–127, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching Word Vectors with Subword Information. *arXiv e-prints*, page arXiv:1607.04606.

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.

François Chollet et al. 2015. Keras. `https://keras.io`.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv e-prints*, page arXiv:1810.04805.

Benjamin Heinzerling and Michael Strube. 2017. BPEmb: Tokenization-free Pre-trained Subword Embeddings in 275 Languages. *arXiv e-prints*, page arXiv:1710.02187.

Mai Ibrahim, Marwan Torki, and Nagwa El-Makky. 2018. Imbalanced toxic comments classification using data augmentation and deep learning. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 875–878.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv e-prints*, page arXiv:1412.6980.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Abu Bakr Mohammad, Kareem Eissa, and Samhaa El-Beltagy. 2017. Aravec: A set of arabic word embedding models for use in arabic nlp. *Procedia Computer Science*, 117:256–265.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.*, 12:2825–2830.

Mohammad Salameh and Houda Bouamor. 2018. Fine-grained Arabic dialect identification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1332–1344, Santa Fe, New Mexico, USA. Association for Computational Linguistics.