ZCU-NLP at MADAR 2019: Recognizing Arabic Dialects

Pavel Přibáň pribanp@kiv.zcu.cz
Stephen Taylor stepheneugenetaylor@gmail.com
Department of Computer Science and Engineering,
Faculty of Applied Sciences, University of West Bohemia,
Pilsen, Czech Republic
http://nlp.kiv.zcu.cz

Abstract

In this paper, we present our systems for the *MADAR Shared Task: Arabic Fine-Grained Dialect Identification*. The shared task consists of two subtasks. The goal of Subtask–1 (S-1) is to detect an Arabic city dialect in a given text and the goal of Subtask–2 (S-2) is to predict the country of origin of a Twitter user by using tweets posted by the user.

In S-1, our proposed systems are based on language modelling. We use language models to extract features that are later used as an input for other machine learning algorithms. We also experiment with recurrent neural networks (RNN), but these experiments showed that simpler machine learning algorithms are more successful. Our system achieves 0.658 macro F_1-score and our rank is 6^{th} out of 19 teams in S-1 and 7^{th} in S-2 with 0.475 macro F_1-score.

1 Introduction

The Madar shared tasks (Bouamor et al., 2019) are a follow-up to Salameh's (Salameh et al., 2018) work with the synthetic corpus of Bouamor (Bouamor et al., 2014) and Salameh's work with tweets based on the corpus. Two corpora are provided, a six-city corpus of travel sentences rendered into the dialects of five cities and MSA[1], and a 25-city + MSA corpus using a smaller number of sentences. In the first task, test data is classified as one of the 25 cities or MSA. For the second task, the organizers chose training, development and test tweet-sets for download from Twitter. The tweets are from 21 Arabic countries, and the goal is to determine, for each tweet author, the country of origin.

For S-1 we did not use any external data, only data provided by the shared task organizers. The organizers provided training and development data[2] consisting of sentences in different dialects with a label denoting the corresponding dialect. The training data contain 41K sentences and development data contain 5.2K sentences. Organizers also provided additional data with Arabic sentences in seven dialects.

S-2 uses a corpus of tweets. Twitter does not permit the organizers to distribute tweets, only the user ids and tweet ids. Every participant must arrange with Twitter to download the tweets themselves, and because tweets are subject to deletion over time, it is possible that each participant's version of the corpus and test is unique.

2 Related Work

The Arabic dialects have a common written form and unified literary tradition, so it seems most logical to distinguish dialects on the basis of acoustics, and there is a fair amount of work there, including Hanani et al. (2013, 2015); Ali et al. (2016).

—-

Biadsy et al. (2009) distinguish four Arabic dialects and MSA based on (audio) phone sequences; the phones were obtained by phone recognizers for English, German, Japanese, Hindi, Mandarin, Spanish, and three different MSA phone-recognizer implementations. The dialects were distinguished by phoneme sequences, and the results of classifications based on each phone-recognizer were combined using a logistic regression classifier. They train on 150 hours per dialect of telephone recordings. They report 61% accuracy on 5-second segments, and 84% accuracy on 120 second segments.

Zaidan and Callison-Burch (2011) describe building a text corpus, based on reader commen-

[1] Modern Standard Arabic

[2] The participants were not allowed to use these data for any training purposes.

tary on newspaper websites, with significant dialect content; the goal is to provide a corpus to improve machine translation for Arabic dialects. They used Amazon Mechanical Turk to provide annotation for a portion of the corpus. Zaidan and Callison-Burch (2014) describe the same work in greater detail, including dialect classifiers they built using the Mechanical Turk data for classes and origin metadata as additional features. They say these classifiers are 'approaching human quality.'

ElFardy and Diab (2013) classify EGY[3] and MSA sentences from the Zaidan and Callison-Burch (2011) corpus, that is, from text. Not only is this a binary task, but orthographic hints, including repeated long vowels, emojis and multiple punctuations, give strong clues of the register, and hence whether MSA is being employed. They do a number of experiments comparing various preprocessing schemes and different training sizes, ranging from 2-28 million tokens. They achieve 80% – 86% accuracy for all of their attempts.

Malmasi et al. (2015) do Arabic dialect identification from text corpora, including the Multi-Dialect Parallel Corpus of Arabic (Bouamor et al., 2014) and the Arabic Online Commentary database (Zaidan and Callison-Burch, 2011).

Hanani et al. (2015) perform recognition of several Palestinian regional accents, evaluating four different acoustic models, achieving 81.5% accuracy for their best system, an I-vector framework with 64 Gaussian components.

Ali et al. (2016) developed the corpus on which the DSL Arabic shared task is based. Their own dialect detection efforts depended largely on acoustical cues.

Arabic dialect recognition appeared in the 2016 edition of the VarDial workshop's shared task (Malmasi et al., 2016). The shared task data was text-only.

The best classifiers (Malmasi et al., 2016; Ionescu and Popescu, 2016) for the shared task performed far below the best results reported by some of the preceding researchers, in particular Ali et al. (2016) which used some of the same data.

Part of the reason must be that the amount of training data for the workshop is much smaller than that used by some of the other researchers; the workshop data also did not include the audio recordings on which the transcripts are based.

[3]Egyptian dialect

The absence of audio was remedied for the 2017 and 2018 VarDial workshops, (Zampieri et al., 2017, 2018)

However, the five dialects plus MSA targeted by the VarDial shared task comprise a small fraction of Arabic's dialectical variation. Salameh et al. (Salameh et al., 2018) use a corpus (Bouamor et al., 2018) which differentiates between twenty-five different cities and MSA. This still doesn't address urban rural divides, but it begins to reflect more realistic diversity.

3 Overview

3.1 Language Modelling

In `S-1`, both of our systems used for the official submission take as an input language model features. In our case the objective of a language model in its simplest form is to predict probability $p(S)$ of sentence S which is composed from strings (words or character n-grams) $s_1, s_2 \dots s_N$, where N is a number of strings in the sentence. The probability estimation of $p(S)$ can be computed as a product of conditional probabilities $p(s_i|h_i)$ of its strings $s_1, s_2 \dots s_N$, where h_i is a history of a string s_i. The probability of string s_i is conditioned by history h_i i.e. $n-1$ preceding strings $s_{i-n+1}, s_{i-n+2}, \dots s_{i-1}$ which can be rewritten as s_{i-n+1}^{i-1}. The resulting formula for the $p(S)$ estimation looks as follows:

$$p(S) = \prod_{i=1}^{N} p(s_i|h_i) = \prod_{i=1}^{N} p(s_i|s_{i-n+1}^{i-1}) \quad (1)$$

The conditioned probability $p(s_i|h_i)$ can be estimated with *Maximum Likelihood Estimate* (*MLE*) which is defined as:

$$p^{MLE}(s_i|h_i) = \frac{c(s_{i-n+1}, s_{i-n+2} \dots s_i)}{c(s_{i-n+1}, s_{i-n+2} \dots s_{i-1})} \quad (2)$$

where $c(s_{i-n+1}, s_{i-n+2} \dots s_i)$ is a number of occurrences of string s_i with history h_i and $c(s_{i-n+1}, s_{i-n+2} \dots s_{i-1})$ is a number of occurrences of history h_i. These counts are taken from a training corpus.

We followed Salameh (Salameh et al., 2018) in using the `kenlm` language modelling tool (Heafield et al., 2013). `kenlm` doesn't have an option to use character n-grams instead of words,

so in order to get character-based language models, we prepared input files with characters separated by spaces. Instead of encoding space as a special word, we surrounded words with a <w></w>pair. This enables noticing strings which occur at the beginning or end of a word (as would a special sequence for space) but reduces the possible amount of inter-word information which the language model can keep for a given order, the parameter which indicates to `kenlm` the largest n-gram to index. We used order 5 for all our `kenlm` language models. We prebuilt models for each dialect. We prepared six directories, each containing word or character models for each dialect in one of the three corpora.

We wrote a `LangModel` class which quacks like a `sklearn` classifier, that is, it supports `fit()`, `predict()`, and `predict_proba()`, but its choices are based on a directory of language models. `predict()` returns the dialect name whose model gives the highest score. `predict_proba()` provides a list of language-model-score features, adjusted to probabilities.

4 Subtask–1 System Description

In this section we describe our models[4]. We submitted results for the `S-1` from two systems – *Tortuous Classifier* and *Neural Network Classifier*.

4.1 Tortuous Classifier

This submission uses a jumble of features and classifiers, most from the sklearn module (Buitinck et al., 2013). The final classifier is a hard voting classifier with three input streams:

1. Soft voting classifier on:
   (a) Multinomial naive Bayes classifier on word 1-2grams,
   (b) Multinomial naive Bayes classifier on char 3-5grams,
   (c) Language model scores adjusted to probabilities, for word-based language models of the corpus 26 dialects
   (d) Language model scores adjusted to probabilities, for char-based language models of the corpus 26 dialects
   (e) Multinomial naive Bayes classifier on language-model-scores for character and language models on the corpus-6 language models and character language models for the corpus-26 language models.
2. Support vector machine, `svm.SVC(gamma='scale', kernel = 'poly', degree = 2)` with the same features as item 1e.
3. Multinomial naive Bayes classifier using word and char language model features for corpus-6 and corpus-26 features, tfid vectorized word 1-2grams, and tfid vectorized char 3-5grams.

The classifier did better on the development data, suggesting that it is over-fitted, but the language model features, which are the most predictive, also did better on the development data.

4.2 Neural Network Classifier

We experimented with several neural networks. Our model for the `S-1` submission uses as input 26 features which correspond to one of our 26 pretrained dialect language models. Each feature represents the probability of a given sentence for one language model. The probability scores measure how close each sentence is to the dialect.

We train Multilayer Perceptron (MLP) with one hidden (dense) layer with 400 units. The output of the hidden layer is passed to a final fully-connected softmax layer. The output of the softmax layer is a probability distribution over all 26 classes. The class with the highest probability is predicted as a final output of our model. As an activation function in the hidden layer of the MLP a Rectified Linear Unit (ReLu) is employed.

We also tried to combine character n-gram features with the language model features. The input is a sequence of first 200 character n-grams of a given text. Each sequence of character n-grams is used as a separate input followed by a randomly initialized embedding layer and then two layers of Bidirectional LSTM (BiLSTM)(Graves and Schmidhuber, 2005) with 64 units are employed (see Figure 1).

The output vector of the BiLSTM layers is concatenated with the language model features and this concatenated vector is passed to the MLP layer with 400 units (the same as described above). All models were implemented by using Keras (Chollet et al., 2015) with TensorFlow backend (Abadi et al., 2015)

[4]The source code is available at https://github.com/StephenETaylor/Madar-2019

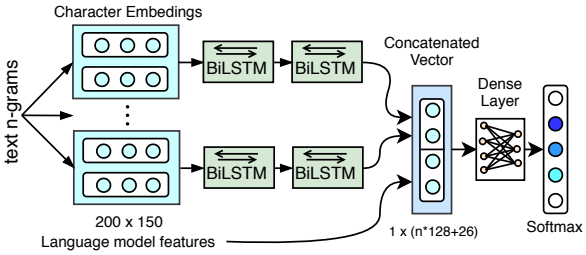

Figure 1: Neural network model architecture

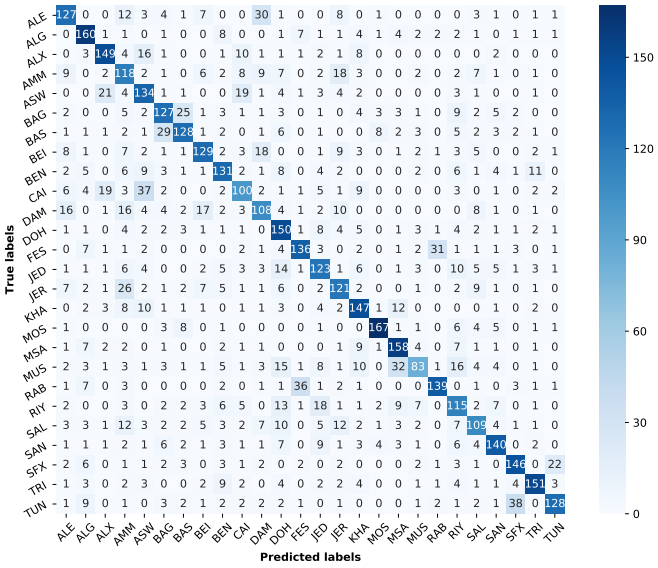

Figure 2: Tortuous Classifier confusion matrix

4.3 Neural Netwok Model Training

We tune all hyperparameters on the development data. We train our model with Adam (Kingma and Ba, 2014) optimizer with learning rate 0.01 and without any dropout. The number of epochs is 800 and we do not use mini-batches or dropout regularization technique. The model with these hyperparameters achieves the best result (0.661 macro F_1-score) on the development data and was used for the final submission.

We also experimented with the n-gram inputs. We tried a different number of character n-grams and we achieve the best result (0.555 macro F_1-score) on the development data using three inputs - character unigrams, bigrams and trigrams, with learning rate 0.005, mini-batches of size 256 for 11 epochs and with the Adam optimizer.

5 Subtask–2 System Description

Our tortuous classifier did less well on the tweet data, so we used a simpler classifier.

The features are the kenlm language model scores for the 21 countries, computed for each of the training tweets, then exponentiated and normalized to sum to 1. The tweets are classified using

```
y_test = KNeighborsClassifier
       (n_neighbors=31)
       .fit(X_train,y_train)
         .predict(X_test)
```

The users are predicted based on the plurality prediction for all of their tweets, that is, the country to which the largest number of their tweets were assigned.

There were a significant number of tweets unavailable, about 10% in the training and development sets, and 12% in the test set. After the submissions had closed we experimented with eliminating the unavailable and non-Arabic tweets from training and testing and choosing *Saudi_Arabia* (which is the origin for the plurality of tweets at 36%) for users with no remaining tweets. This improved tweet classification accuracy by about 5%, but actually decreased user classification accuracy on the development set.

6 Results

For the Subtask–1 we achieved 0.658 macro F_1-score on the test data, sixth among nineteen submissions with the *Tortuous Classifier*. The *Neural Network Classifier* achieved a macro F_1-score of 0.648 on the test data. For the Subtask–2 we submitted a single entry. It ranked 7^{th} among 9 submissions with 0.475 macro F_1-score.

Figure 2 shows that many of the errors are geographically plausible. For example, ASWan ALXandria and CAIro are all in Egypt, and each has a sizeable chunk of mistaken identity for the others. Similarly, DAMascus, ALEppo, AMMan, BEIrut, JERusalem which are all 'Levantine' and only a few hundred miles apart.

7 Conclusion

This paper presents an automatic approach for Arabic dialect detection in the *MADAR Shared Task*. Our proposed systems for the Subtask-1 use language model features. Our experiments showed that simpler machine learning algorithms outperform RNN using language model features. Subtask–2 turned out to be more challenging because Tweets, which are real-world wild data, are

more difficult to process than systematically prepared texts.

Acknowledgments

This work has been partly supported by Grant No. SGS-2019-018 Processing of heterogeneous data and its specialized applications, and was partly supported from ERDF "Research and Development of Intelligent Components of Advanced Technologies for the Pilsen Metropolitan Area (InteCom)". Access to computing and storage facilities owned by parties and projects contributing to the National Grid Infrastructure MetaCentrum provided under the programme "Projects of Large Research, Development, and Innovations Infrastructures" (CESNET LM2015042), is greatly appreciated.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Ahmed Ali, Najim Dehak, Patrick Cardinal, Sameer Khurana, Sree Harsha Yella, James Glass, Peter Bell, and Steve Renals. 2016. Automatic dialect detection in Arabic broadcast speech. In *Proceedings of Interspeech 2016*, pages 2934–2938.

Fadi Biadsy, Julia Hirschberg, and Nizar Habash. 2009. Spoken arabic dialect identification using phonotactic modeling. In *Proceedings of the EACL Workshop on Computational Approaches to Semitic Languages*, pages 53–61.

Houda Bouamor, Nizar Habash, and Kemal Oflazer. 2014. A multidialectal parallel corpus of Arabic. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC14)*. European Language Resources Association (ELRA).

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 11th International Conference on Language Resources and Evaluation*.

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.

Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. 2013. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122.

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

Heba ElFardy and Mona Diab. 2013. Sentence level dialect identification in Arabic. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 456–461.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5-6):602–610.

Abualsoud Hanani, Hanna Basha, Yasmeen Sharaf, and Stephen Taylor. 2015. Palestinian Arabic regional accent recognition. In *The 8th International Conference on Speech Technology and Human-Computer Dialogue*.

Abualsoud Hanani, Martin J. Russell, and Michael J. Carey. 2013. Human and computer recognition of regional accents and ethnic groups from British english speech. *Computer Speech and Language*, 27(1):5974.

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, , and Philipp Koehn. 2013. Scalable modified kneser-ney language model estimation. In *ACL*.

Radu Tudor Ionescu and Marius Popescu. 2016. UnibucKernel: An Approach for Arabic Dialect Identification Based on Multiple String Kernels. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pages 135–144, Osaka, Japan.

Diederik P. Kingma and Jimmy Lei Ba. 2014. Adam: A method for stochastic optimization. *arXiv:1412.6980v9*.

Shervin Malmasi, Eshrag Refaee, and Mark Dras. 2015. Arabic dialect identification using a parallel multidialectal corpus. In *Proceedings of the 14th Conference of the Pacific Association for Computational Linguistics (PACLING 2015)*, pages 209–217, Bali, Indonesia.

Shervin Malmasi, Marcos Zampieri, Nikola Ljubei, Preslav Nakov, Ahmed Ali, and Jrg Tiedemann. 2016. Discriminating between similar languages and arabic dialect identification: A report on the third dsl shared task. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2016)*.

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 F. Zaidan and Chris Callison-Burch. 2011. The Arabic Online Commentary dataset: An annotated dataset of informal Arabic with high dialectal content. In *Proceedings of ACL*, pages 37–41.

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

Marcos Zampieri, Shervin Malmasi, Nikola Ljubei, Preslav Nakov, Ahmed Ali, Jrg Tiedemann, Yves Scherrer, and Nomi Aepli. 2017. Findings of the vardial evaluation campaign 2017. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2017*.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Ahmed Ali, Suwon Shon, James Glass, Yves Scherrer, Tanja Samardžić, Nikola Ljubešić, Jörg Tiedemann, Chris van der Lee, Stefan Grondelaers, Nelleke Oostdijk, Dirk Speelman, Antal van den Bosch, Ritesh Kumar, Bornini Lahiri, and Mayank Jain. 2018. Language identification and morphosyntactic tagging: The second VarDial evaluation campaign. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pages 1–17, Santa Fe, New Mexico, USA. Association for Computational Linguistics.