A Character Level Convolutional BiLSTM for Arabic Dialect Identification

Mohamed Elaraby
Raisa Energy
msalem@raisaenergy.com

Ahmed Ismail Zahran
Cairo University
zahran@ieee.org

Abstract

In this paper, we describe the contribution of CU-RAISA team to the 2019 Madar shared task 2 [1], which focused on Twitter User fine-grained dialect identification. Among participating teams, our system ranked the *4th (with 61.54%) F1-Macro measure.* Our system is trained using a character level convolutional bidirectional long-short-term memory (BiLSTM) network trained on approximately 2k users' data. We show that training on concatenated user tweets as input is further superior to training on user tweets separately and assign user's label on the mode of user's tweets' predictions.

1 Introduction

Dialect identification is a sub-domain of language identification, a task that aims to differentiate between different languages given a sample of spoken or written text. Language and dialect identification are active research areas due to their usefulness as preliminary steps for other applications, such as automatic speech recognition and machine translation. The task of dialect identification poses harder challenges due to the higher inter-class similarity, which becomes harder to learn with hidden text solely due to the absence of pronunciation information that exists in audio data. (Sibun and Reynar, 1996) made the first effort to distinguish between languages with high similarity. Their dataset contained some languages with similar content, such as Serbian and Croatian, among others.

Arabic dialect identification (ADI) aims to differentiate between dialects of the Arab world, spoken by citizens of the Middle East and North Africa. Multiple forms of categorization can exist when it comes to Arabic dialect identification. The first form is based on the geographic location, where the text is categorized with respect to the home origin of the individual. The second form is concerned with major dialects, grouping the variations from different countries into larger classes. The most common categorization of the second form for Arabic dialects is the one described by (Habash et al., 2012), which details five major dialects (Egyptian, Gulf, Iraqi, Levantine, and Maghrebi). In this paper, we will be exploring the first form of categorization. This form poses more challenges due to the increased granularity it adds to the classification task.

2 Related Work

Deep learning models have gained attention in the tasks of text-based ADI, spoken language-based ADI and hybrid (text+spoken language) ADI with the introduction of context-dependent architectures such as Long short-term memory (LSTM) and Convolutional neural networks (CNN's). Research in the past few years has explored both character-level and word-level models, along with combining these models with acoustic features from the audio recordings. (Sayadi et al., 2017) achieved a classification accuracy of 92.2% on a two-way classification task between Modern Standard Arabic (MSA) and Tunisian using a character-level LSTM model. The experiments were performed on the Tunisian Election Twitter dataset (Sayadi et al., 2016). For a fine-grained six-class classification task (MSA, Egyptian, Syrian, Jordanian, Palestinian and Tunisian) on the Multidialectal Parallel Corpus of Arabic dataset (Bouamor et al., 2014), the authors reached a classification accuracy of 63.4%. Elaraby and Abdul-Mageed (2018) experimented with attention-based bidirectional LSTM (BiLSTM) models on a two-way classification task (MSA vs. other dialects), a

[1] https://competitions.codalab.org/competitions/22475

Proceedings of the Fourth Arabic Natural Language Processing Workshop, pages 274–278
Florence, Italy, August 1, 2019.

three-way classification task (Egyptian, Gulf, and Levantine), and a four-way classification task that adds the MSA dialect to the previous three-way task. The dataset used in this study is the Arabic Online Commentary (AOC) dataset. (Zaidan and Callison-Burch, 2011). The system achieved an accuracy of 87.65%, 87.4% and 82.45% on the three aforementioned tasks, respectively using pretrained word embeddings trained on a large dialectly rich corpus described in (Abdul-Mageed et al., 2018). (Ali, 2018) used a character-level convolution neural network with a GRU layer for a five-way classification task (MSA, Egyptian, Gulf, Levantine, and North African). This architecture achieved 92.64% cross-validation accuracy on the training set, and a 57.59% F1 (macro) score on the test set. (Lulu and Elnagar, 2018) isolated the three most frequent dialects in AOC (Gulf, Egyptian, and Levantine). Using a word-based LSTM to differentiate between the three dialects, the authors obtained an accuracy of 71.4%, exceeding the performance of CNN, BLSTM and CLSTM models.

Along with exploring the performance of deep learning models on ADI, research has also continued to explore more classical models, such as kernel-based models and linear models, in addition to classical representations such as tf-idf. In a geographic location-based ADI task, Salameh et al. (2018) researched the effectiveness of combining multiple features with a Multinomial Naive Bias (MNB) classifier. The system combined multiple word-based and character-based n-grams with language models scoring probabilities as features. The authors used a translated version of the Basic Traveling Expression Corpus (BTEC) (Takezawa et al., 2007). For sentences with an average length of seven words, the system obtained a classification accuracy of 67.9%. As the average length of the sentence increases to 16 words, the performance of the system increased to more than 90%. This finding gives an intuition about the positive effect of sentence length on the performance of the classifier. In addition to the classification task, the authors analyzed the amount of pairwise dialect similarity between the dialects. To perform the analysis, the authors used hierarchical agglomerative clustering on the similarity matrix obtained from the percentage of shared tokens between dialects. The resulting analysis shows the amount of similarity between dialects in a certain area, as well as the proximity of some dialects to others (e.g.: Egyptian and Levantine). MSA falls closest to Muscat and Khartoum. (Butnaru and Ionescu, 2018) used multiple kernel learning on character n-grams from text and phonetic transcriptions, along with dialectal embeddings from the audio recordings. Their model obtained an accuracy of 58.65%. (El Haj et al., 2018) researched the subjects of code-switching and bivalent words (words that occur in multiple languages or dialects with similar semantic content) in dialect identification. They developed a method called Subtractive Bivalency Profiling to build a system that can handle both of these issues. Using support vector machines (SVM) for a task to distinguish between four dialects (MSA, Egyptian, Levant, and Gulf), they achieved 76% accuracy. (Lichouri et al., 2018) researched word-based and sentence-based methods on tf-idf vectors, in addition to applying majority and minority voting techniques. The authors experimented with Bernoulli Naive Bayes (BNB) and MNB, along with Linear SVM's (LSVM). Two datasets were used for this research. The first dataset, PADIC (Meftouh et al., 2015; Harrat et al., 2014), consists of multiple dialects (MSA, Tunisian, Moroccan, Algerian, Palestenian and Syrian). For this dataset, a sentence-level BNB achieved the highest accuracy (73.15%). The second dataset consisted of eight Algerian dialects (Tenes, Constantine, Djelfa, Ain-Defla, Tizi-Ouzou, Batna, Annaba, and Algiers), for which an LSVM model achieved the highest accuracy (41.05%).

3 Data

3.1 Dataset Description

We used the Arabic twitter dataset released by the organizers of the "User Dialect Identification task". The dataset is portioned into 217,593 tweets representing 2180 users for training, 29,870 for development representing 300 users, and 49,962 for testing representing 500 users. Full detailed description of the data can be found in task description paper Bouamor et al. (2019).

3.2 Accessibility of tweets

One challenging part of this task was the accessibility of tweets as some users' tweets weren't accessible at the time we crawled their timelines from twitter. Training data portion were reduced from 2180 users to 2032 users. The total number

of training tweets were reduced to 192,389. Development data were reduced from 300 to 281 users, while the number of development tweets was reduced to 26,528. The number of test users was reduced from 500 to 463.

4 Methods

4.1 Pre-processing

We adopt basic preprocessing techniques to our training, development, and test sets. This involves filtering out URLs and user mentions. For the vocabulary V, we train using character-based vocabulary. We filter out least frequent characters occurring < 20 times, which leaves $|V| = 2377$ of unique characters.

4.2 Data Preparation:

We conduct two sets of experiments; (1): train on tweet level annotated by the country of the user. In that case, the maximum input sequence length is 140. (2) : train on user's concatenated tweets together. Maximum sequence length grown to 12000 characters. In the results section, we show that training on concatenated user tweets improves performance compared to training on individual tweets. On the hidden units layer to prevent the network from over-fitting on training set.

4.3 Models

4.3.1 Traditional Models

Traditional models refer to models based on feature engineering methods with linear and probabilistic classifiers. In our experiments, we use (1) logistic regression, and (2) multinomial Naive Bayes as baselines. We use character ngrams, word ngrams, and a combination of both as feature set.

4.3.2 Deep Learning Models

We develop models based on deep neural networks based on variations of (1) convolution neural networks (CNNs) and (2) recurrent neural networks (RNNs) which have proved useful for several NLP tasks. Both RNNs, and CNNs s are able to capture sequential dependencies especially in time series data, of which language can be seen as an example.

Our Model: We use a combination of convolution neural network and bidirectional long short term memory (BiLSTM). The following part describes how we apply CNN to extract higher-level sequences of word features and BiLSTM to capture long-term dependencies over window feature sequences respectively.

- *Input layer:* an input layer to map word sequence w into a sequence vector x where x_w is a real-valued vector ($X_w \epsilon \mathbb{R}^{d_{emb}}$ where $d_{emb} = 50$). Character embedding are randomly initialized and not learnt externally.

- *Convolution layer:* Multiple convolution operations are applied in parallel to the input layer to map input sequence **x** into a hidden sequence **h**

  A filter $k \epsilon \mathbb{R}^{w_{demb}}$ is applied to a window of concatenated word embedding of size w to produce a new feature c_i . Where $c_i \epsilon \mathbb{R}, c_i = k \cdot x_{i:i+w-1+b}$ b is the inductive bias term $b \epsilon \mathbb{R}$, and $x_{i:i+w-1}$ is a concatenation of $x_i, x_{i+1}, .., x_{i+w-1}$

  The filter sizes used are ranging from 1-13 and the number of filters used is ranging from 10-150. Finally, different convolution outputs are concatenated into a sequence $c \epsilon \mathbb{R}^{n-h+1}$ and passed into a time distributed layer to convert it into suitable output for the BiLSTM layer.

- *BiLSTM Layer:* We use a Bidirectional LSTM architecture consisting of 256 dimensions hidden units. The BiLSTM is designed to capture long-term dependencies via augmenting a standard RNN with two memory states, forward and backward. The forward direction state $\overrightarrow{C}_t$, with $\overrightarrow{C}_t \in \mathbb{R}$ at time step t. The forward LSTM takes in a previous state $\overrightarrow{h}_{t-1}$ and input x_t, to calculate the hidden state $\overrightarrow{h}_t$ as follows:

$$\overrightarrow{i}_t = \sigma(W_{\overrightarrow{i}}[\overrightarrow{h}_{t-1}, x_t] + b_{\overrightarrow{i}})$$
$$\overrightarrow{f}_t = \sigma(W_{\overrightarrow{f}}.[\overrightarrow{h}_{t-1}, x_t] + b_{\overrightarrow{f}})$$
$$\overrightarrow{C}_t = tanh(W_{\overrightarrow{C}}.[\overrightarrow{h}_{t-1}, x_t] + b_{\overrightarrow{C}})$$
$$\overrightarrow{C}_t = \overrightarrow{f}_t \odot \overrightarrow{C}_{t-1} + i_t \odot \overrightarrow{C}$$
$$\overrightarrow{o}_t = \sigma(W_o[\overrightarrow{h}_{t-1}, x_t] + b_{\overrightarrow{o}})$$
$$\overrightarrow{h}_t = o_t \odot tanh(\overrightarrow{C}_t)$$

  where σ is the sigmoid, $tanh$ is the hyperpolic tangent function, and $\odot$ is the dot product between two vectors. The $\overrightarrow{i}_t$, $\overrightarrow{f}_t$, $\overrightarrow{o}_t$

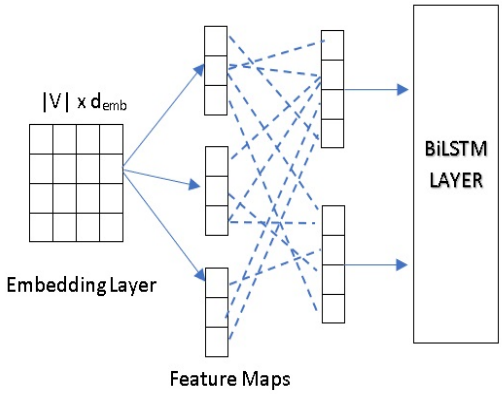

Figure 1: Our char level Convolution -BiLSTM

are the *input, forget*, and *output* gates, and the $\overrightarrow{C}_t$ is a new memory cell vector with candidates that could be added to the state in the forward direction. The same operation is done for the backward direction. We apply L2 regularization to avoid network overfitting.

- *Softmax Layer:* Finally, the combined hidden units (forward and backward) is converted into a probability distribution over l via softmax function, where l is the number of classes in our case (21 classes).

Figure 1 shows a block diagram of our network architecture.

Training and Optimization

We try a small set of hyper-parameters, identifying best settings on our validation set using grid search. We train the network for 40 epochs each. For optimization, we use Adam (Kingma and Ba, 2014), The models weights W are initialized from a normal distribution $W \sim N$ with a small standard deviation of $= 0.05$ We apply two sources of regularization: dropout: we apply a dropout rate of 0.2 on the input embeddings to prevent co-adaptation of hidden units activation, and $L2$ norm: we also apply an L2-norm regularization with a small value (0.002)

5 Results

We evaluated most of the experiments on the development set using an accuracy metric. Table 1 concluded our experimentation results on development set which consists of 281 users in total after excluding tweets of non-accessible users.

For the test which set consists of 500 users, we were able to access 463 users which we predicted using our C-BiLSTM network. For the left 37 users we assign the most common class to it which is ***"Saudi_Arabia"*** . The final result reported by organizers on the test set was very close in terms of both accuracy and F1 macro measure achieving an accuracy of 72.6% and 61.5%.

Models	Accuracy	F1-Macro
Individual tweets		
Logistic Regression (1-11 ngrams)	36.5	-
Multinomial Naive Bayes (1-11 ngrams)	36.75	-
Char-Level CNN	50.12	-
Char-Level C-BiLSTM	**51.7**	42.3
Concatenated tweets		
Logistic Regression (1-11 ngrams)	45.5	-
Multinomial Naive Bayes (1-11 ngrams)	46.7	-
Char-Level CNN	68.8	-
Char-Level C-BiLSTM	**71.92**	62.21

Table 1: Experimental results on development set

6 Conclusion

In this paper, we described our system submitted to MADAR shared task, focused on country level dialect identification from Twitter data. We explored the utility of tuning different word- and character-level based models. A char based convolutional BiLSTM achieved the best performance in terms of both accuracy and F1-macro measure. Given our limited resources at that time we weren't able to experiment transfer learning techniques as pre-trained embeddings or language models which proved to be beneficial in various Natural Language Processing tasks. In future work, we plan to exploit a number of those techniques in the fine-grained dialect identification task.

References

Muhammad Abdul-Mageed, Hassan Alhuzali, and Mohamed Elaraby. 2018. You tweet what you speak: A city-level dataset of arabic dialects. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*.

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.

Houda Bouamor, Nizar Habash, and Kemal Oflazer. 2014. A multidialectal parallel corpus of arabic. In *LREC*, pages 1240–1245.

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.

Andrei M Butnaru and Radu Tudor Ionescu. 2018. Unibuckernel reloaded: First place in arabic dialect identification for the second year in a row. *arXiv preprint arXiv:1805.04876*.

Mahmoud El Haj, Paul Edward Rayson, and Mariam Aboelezz. 2018. Arabic dialect identification in the context of bivalency and code-switching.

Mohamed Elaraby and Muhammad Abdul-Mageed. 2018. Deep models for arabic dialect identification on benchmarked data. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pages 263–274.

Nizar Habash, Mona T Diab, and Owen Rambow. 2012. Conventional orthography for dialectal arabic. In *LREC*, pages 711–718.

Salima Harrat, Karima Meftouh, Mourad Abbas, and Kamel Smaili. 2014. Building resources for algerian arabic dialects. In *15th Annual Conference of the International Communication Association Interspeech*.

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

Mohamed Lichouri, Mourad Abbas, Abed Alhakim Freihat, and Dhiya El Hak Megtouf. 2018. Word-level vs sentence-level language identification: Application to algerian and arabic dialects. *Procedia Computer Science*, 142:246–253.

Leena Lulu and Ashraf Elnagar. 2018. Automatic arabic dialect classification using deep learning models. *Procedia computer science*, 142:262–269.

Karima Meftouh, Salima Harrat, Salma Jamoussi, Mourad Abbas, and Kamel Smaili. 2015. Machine translation experiments on padic: A parallel arabic dialect corpus. In *The 29th Pacific Asia conference on language, information and computation*.

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.

Karim Sayadi, Mansour Hamidi, Marc Bui, Marcus Liwicki, and Andreas Fischer. 2017. Character-level dialect identification in arabic using long short-term memory. In *International Conference on Computational Linguistics and Intelligent Text Processing*, pages 324–337. Springer.

Karim Sayadi, Marcus Liwicki, Rolf Ingold, and Marc Bui. 2016. Tunisian dialect and modern standard arabic dataset for sentiment analysis: Tunisian election context. In *Second International Conference on Arabic Computational Linguistics, ACLING*, pages 35–53.

Penelope Sibun and Jeffrey C Reynar. 1996. Language identification: Examining the issues.

Toshiyuki Takezawa, Genichiro Kikui, Masahide Mizushima, and Eiichiro Sumita. 2007. Multilingual spoken language corpus development for communication research. *International Journal of Computational Linguistics & Chinese Language Processing, Volume 12, Number 3, September 2007: Special Issue on Invited Papers from ISCSLP 2006*, 12(3):303–324.

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 the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 37–41. Association for Computational Linguistics.