Character Level Convolutional Neural Network for Arabic Dialect Identification

Mohamed Ali
Cairo University, Egypt
mohamedali@aucegypt.edu

Abstract

This paper presents the systems submitted by the safina team to the Arabic Dialect Identification (ADI) shared task at the VarDial Evaluation Campaign 2018. The ADI shared task included five Arabic dialects: Modern Standard Arabic (MSA), Egyptian, Gulf, Levantine, and North-African. The proposed approach is to use character-level convolution neural network in addition to dialect embedding vectors, a low dimensional representation extracted from linguistic features, to distinguish the 5 dialects. We submitted three models with the same architecture except for the first layer. The first system uses one-hot character representation as input to the convolution layer. The second system uses an embedding layer before the convolution layer. The third system uses a recurrent layer before the convolution layer. The best results were obtained using the third model achieving 57.6% F1-score, ranked the second among six teams.[1]

1 Introduction

In the Arab world, several varieties of the Arabic language are co-existing together. Those varieties include Modern Standard Arabic (MSA), and many regional dialects as Egyptian, Gulf, Levantine and North-African dialects . Arabic Dialect Identification task is concerned with identifying the specific Arabic dialect in spoken and written forms which is a crucial task in many Natural Language Processing (NLP) applications.

In this paper we present the safina team submission for the 2018 ADI shared task which was organized as a part of Vardial Evaluation Campaign 2018 (Zampieri et al., 2018). We have used character-level Convolutional Neural Network approach to identify Arabic dialects using both lexical and dialect embedding features. Our team ranked the second with F1-weighted score 57.59%.

2 Related Work

Research in Arabic Dialect Identification took two tracks: spoken dialect identification and written dialect identification. For spoken dialect identification, Biadsy et al. (2009) described a system that can identify the Arabic dialect from a spoken text using acoustic features. In a later research, authors examined the role of prosodic features in identifying the speaker dialect and reported that using prosodic features showed a significant improvement over using phonotatcic-approach alone.

On the other hand, more research took place for written text dialect identification. Elfardy and Diab (2013) used word level labels to derive sentence-level features then used those features to label the sentence with the appropriate dialect. Zaidan and Callison-Burch (2014) used an annotated dialectal data set called Arabic On-line Commentary (AOC) for training a system to identify the dialect of an Arabic sentence. Later and using the same data set, Tillmann et al. (2014) used word-based binary features to

[1] The code for our submissions is available at: https://github.com/bigoooh/adi

Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects, pages 122–127
Santa Fe, New Mexico, USA, August 20, 2018.

train a linear support vector machine classifier to distinguish between the Egyptian dialect and MSA. Darwish et al. (2014) showed that the data in AOC corpus has some homogeneity as it is drawn from singular sources. This homogeneity would prevent models trained on this data from generalizing to unseen topics. Darwish et al. (2014) found that character-based n-grams outperformed word-based n-grams to distinguish between Egyptian dialect and MSA.

Ali et al. (2016) combined acoustic features with lexical features obtained from a speech recognition system which yield to a classifier stronger than classifiers that use acoustic-only or lexical-only features. In 2016, ADI appeared as a subtask of the Discriminating between Similar Languages (DSL) shared task. The data set in this subtask was a transcribed speech in MSA and in four dialects (Ali et al., 2016): Egyptian (EGY), Gulf (GLF), Levantine (LAV), and North African (NOR). Most of the teams in this subtask used character n-grams and the best result achieved using the support vector machine classifier over character n-grams (1-7) (Çöltekin and Rama, 2016). Belinkov and Glass (2016) used character-level convolution neural network, the same approach we are using, but they only used the ASR transcripts of Arabic speech as the acoustic features were not available at that time. Also, they did not experiment using a recurrent layer as an embedding layer which showed an improvement in our system. In 2017 ADI Shared Task, data set contained the original audio files and some low-level audio features, called i-vectors, along with the ASR transcripts of Arabic speech collected from the Broadcast News domain. Best result in this subtask achieved using a Kernel Discriminant Analysis (KDA) classifier trained on multiple kernel functions over character n-grams and the i-vectors features (Ionescu and Butnaru, 2017).

3 Methodology and Data

3.1 Character-Level Convolutional Neural Network

Convolutional Neural Networks (CNN) were invented to deal with images and it have achieved excellent results in computer vision (Krizhevsky et al., 2012; Sermanet et al., 2013; Ji et al., 2013). Later, it have been applied in Natural Language Processing (NLP) tasks and outperformed traditional models such as bag of words, n-grams and their TFIDF variants (Collobert and Weston, 2008; Zhang et al., 2015). The architecture, shown in Figure 1, describes the character-level CNN model we have used in identifying the Arabic dialects. We formulate the task as a multi-class classification problem. Given the ASR transcript $t^{(i)}$, 600-dimensional dialect embedding feature vectors $v^{(i)}$ and the corresponding label $l^{(i)}$, we need to predict l using v and t. We designed a neural network classifier that takes as input both the transcript as one-hot encoded array of characters (padded or truncated to match a predefined maximum length) and the corresponding dialect embedding feature vector. As shown in Figure 1, the transcript text goes through the convolution layer then a softmax layer while the embedding vector goes directly to another softmax layer. The network final output is the average of the two softmax layers and it represents the probability distribution over the 5 Arabic dialects. The network layers are as follows:

- **Input Layer:** mapping each character to one-hot vector.

- Optional **Embedding or Recurrent Layer :** using embedding or GRU recurrent layer to capture the context of the character (Chung et al., 2014).

- **Convolutional Layer:** contains multiple filter widths and feature maps which is applied to a window of characters to produce new features. Each convolution is followed by a Rectified Linear Unit (ReLU) nonlinearity and batch-normalization layers (Glorot et al., 2011; Ioffe and Szegedy, 2015).

- **Max-Pooling Layer:** apply max-over-time pooling operation over the feature map of each filter and take the maximum value as a feature for this filter (Collobert et al., 2011). The max-pooling operation is followed by a dropout layer to prevent over-fitting (Srivastava et al., 2014).

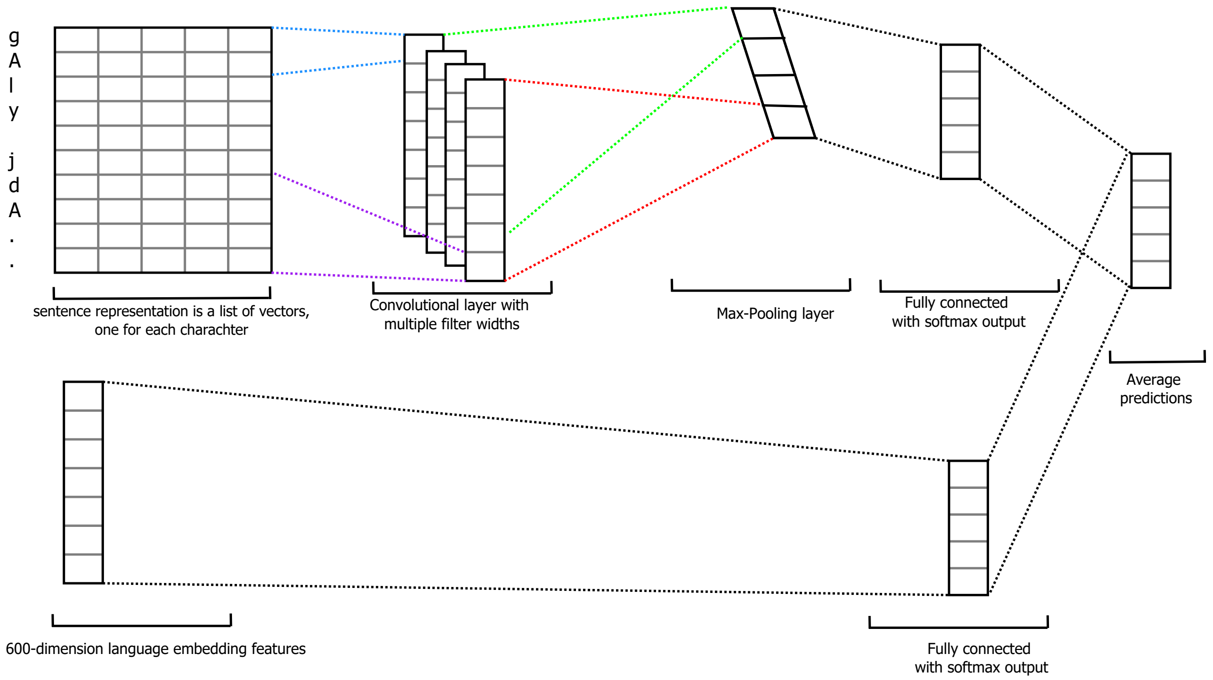
g
A
l
y
j
d
A
.
.
sentence representation is a list of vectors, one for each charachter
Convolutional layer with multiple filter widths
Max-Pooling layer
Fully connected with softmax output
Average predictions
600-dimension language embedding features
Fully connected with softmax output

Figure 1: Character-level CNN architecture

- **Softmax Layers**: there are two softmax layers, one for lexical features and another for the embedding features.
- **Output Layer**:the final output is the average of the two softmax layers' output and it represents the probability distribution over the labels.

Depending on our cross-validation results we used the following parameters for the neural network architecture:

- **Sentence maximum length:** 256 characters
- **Embedding length:**128
- **GRU layer unites:**128 units
- **Convolution filters sizes:** from 2 to 8
- **Convolution filters feature maps:** 256 feature map for each filter
- **Dropout rate:** 0.2

In our implementation, we used Keras framework with TensorFlow as a backend (Chollet and others, 2015; Abadi et al., 2015).

3.2 Data

The ADI shared task data set contains four sets of features for 14591 utterances for training and 1566 utterances for validation (Ali et al., 2016):

- **Raw audio wave files:** contains the audio recordings at 16Khz segmented to remove speaker overlaps and non-speech parts as music or background noise
- **ASR transcripts:** generated by a multi-dialect Arabic Large Vocabulary Speech Recognition (LVCSR) system

- **Dialect Embedding Features:** 600-dimensional dialect embeddings for each utterance. These features were extracted from linguistic features using Siamese neural network approach (Shon et al., 2018).

- **Phonetic Features:** contains phoneme sequence as output of four different speech recognizers (Czech, Hungarian, Russian and English).

In our experiments, we have used only the ASR transcripts and the dialect embedding features.

4 Results

4.1 Cross-Validation Results

We combined the training data and the validation data provided by the shared task to apply 5-fold cross validation. We tested our three different configurations in addition to a TF-IDF features based classifier, Logistic Regression classifier implemented in scikit-learn toolkit (Pedregosa et al., 2011), as a baseline. Results are shown in Table 1.

System	**Accuracy**
Logistic Regression using TF-IDF features	0.5898
CNN with one-hot encoded input	0.9214
CNN with an embedding layer	0.9262
CNN with a GRU recurrent layer	**0.9264**

Table 1: Cross-validation results

4.2 Test Set Results

Our submission results are shown in Table 2. We have used the same configuration for three runs except for the input to the convolution layer. In the first run, we fed the one-hot encoded vectors for the sequence of characters directly to the convolution layer. In the second run, we fed the one-hot encoded vectors to an embedding layer before the convolution layer. In the third run, we fed the one-hot encoded vectors to a GRU recurrent layer before the convolution layer. As shown in the results, using a recurrent layer achieved a slightly better results than feeding the one-hot encoded representation directly to the convolution layer. However, the cost of this slight enhancement was huge in the training time as training the network with recurrent layers took about ten times the period of training the network without the recurrent layer. As shown in the confusion matrix in Figure 2 In the ADI shared task evaluation, the submitted systems were ranked according to it F1-weighted score. Our team ranked the second with F1-weighted score 57.59%. Figure 2 shows the confusion matrix for our best run. From the matrix, we can see that Gulf dialect is the most confusing one; it is highly recognized as Levantine dialect. Also, the Levantine dialect is highly recognized as North African dialect.

System	**F1 (macro)**
Random Baseline	0.1995
CNN with one-hot encoded input	0.5711
CNN with an embedding layer	0.5697
CNN with a GRU recurrent layer	**0.5759**

Table 2: Our three runs results, the best run in bold

5 Conclusion

In this work, we presented our team's three submissions for the ADI shared task. Our approach is to use Character level CNN as a feature extractor from text in addition to the dialect embedding features extracted from text. Our best submission achieved by using a GRU recurrent layer as an embedding

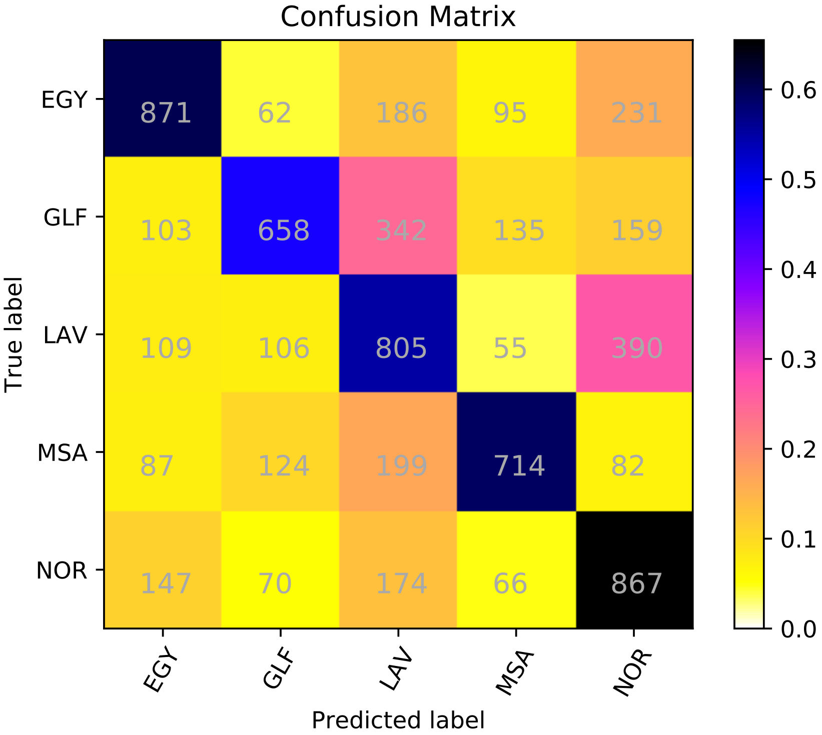

Figure 2: Confusion matrix for ”CNN with a GRU recurrent layer” run

layer before the convolutional layer. However, the gain of using the recurrent layer was minor compared to the cost of the long time used for training a network with a recurrent layer compared to that for a network with a regular embedding layer.

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*, pages 2934–2938.

Yonatan Belinkov and James Glass. 2016. A Character-level Convolutional Neural Network for Distinguishing Similar Languages and Dialects. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pages 145–152, Osaka, Japan.

Fadi Biadsy, Julia Hirschberg, and Nizar Habash. 2009. Spoken arabic dialect identification using phonotactic modeling. In *Proceedings of the eacl 2009 workshop on computational approaches to semitic languages*, pages 53–61. Association for Computational Linguistics.

Çağri Çöltekin and Taraka Rama. 2016. Discriminating Similar Languages with Linear SVMs and Neural Networks. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pages 15–24, Osaka, Japan.

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

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.

Kareem Darwish, Hassan Sajjad, and Hamdy Mubarak. 2014. Verifiably effective arabic dialect identification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1465–1468.

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 (Volume 2: Short Papers)*, volume 2, pages 456–461.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 315–323.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Radu Tudor Ionescu and Andrei Butnaru. 2017. Learning to identify arabic and german dialects using multiple kernels. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 200–209, Valencia, Spain, April.

Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 2013. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. 2013. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*.

Suwon Shon, Ahmed Ali, and James Glass. 2018. Convolutional neural network and language embeddings for end-to-end dialect recognition. In *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop*, pages 98–104.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Christoph Tillmann, Saab Mansour, and Yaser Al-Onaizan. 2014. Improved sentence-level arabic dialect classification. In *Proceedings of the First Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects*, pages 110–119.

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

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, 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)*, Santa Fe, USA.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.