Discrimination between Similar Languages, Varieties and Dialects using CNN- and LSTM-based Deep Neural Networks

Chinnappa Guggilla
chinna.guggilla@gmail.com

Abstract

In this paper, we describe a system (CGLI) for discriminating similar languages, varieties and dialects using convolutional neural networks (CNNs) and long short-term memory (LSTM) neural networks. We have participated in the Arabic dialect identification sub-task of DSL 2016 shared task for distinguishing different Arabic language texts under closed submission track. Our proposed approach is language independent and works for discriminating any given set of languages, varieties and dialects. We have obtained 43.29% weighted-F1 accuracy in this sub-task using CNN approach using default network parameters.

1 Introduction

Discriminating between similar languages, language varieties is a well-known research problem in natural language processing (NLP). In this paper we describe about Arabic dialect identification. Arabic dialect classification is a challenging problem for Arabic language processing, and useful in several NLP applications such as machine translation, natural language generation and information retrieval and speaker identification (Zaidan and Callison-Burch, 2011).

Modern Standard Arabic (MSA) language is the standardized and literary variety of Arabic that is standardized, regulated, and taught in schools, used in written communication and formal speeches. The regional dialects, used primarily for day-to-day activities present mostly in spoken communication when compared to the MSA. The Arabic has more dialectal varieties, in which Egyptian, Gulf, Iraqi, Levantine, and Maghrebi are spoken in different regions of the Arabic population (Zaidan and Callison-Burch, 2011). Most of the linguistic resources developed and widely used in Arabic NLP are based on MSA.

Though the language identification task is relatively considered to be solved problem in official texts, there will be further level of problems with the noisy text which can be introduced when compiling languages texts from the heterogeneous sources. The identification of varieties from the same language differs from the language identification task in terms of difficulty due to the lexical, syntactic and semantic variations of the words in the language. In addition, since all Arabic varieties use the same character set, and much of the vocabulary is shared among different varieties, it is not straightforward to discriminate dialects from each other (Zaidan and Callison-Burch, 2011). Several other researchers attempted the language varsities and dialects identification problems. Zampieri and Gebre (2012) investigated varieties of Portuguese using different word and character n-gram features. Zaidan and Callison-Burch (2011) proposed multi-dialect Arabic classification using various word and character level features.

In order to improve the language, variety and dialect identification further, Zampieri et al. (2014), Zampieri et al. (2015b) and Zampieri et al. (2015a) have been organizing the Discriminating between Similar Languages (DSL) shared task. The aim of the task is to encourage researchers to propose and submit systems using state of the art approaches to discriminate several groups of similar languages and varieties. Goutte et al. (2014) achieved 95.7% accuracy which is best among all the submissions in 2014 shared task. In their system, authors employed two-step classification approach to predict first

Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects,
pages 185–194, Osaka, Japan, December 12 2016.

the language group of the text and subsequently the language using SVM classifier with word and character level n-gram features. Goutte and Leger (2015) and Malmasi and Dras (2015) achieved 95.65% and 95.54% state of the art accuracies under open and closed tracks respectively in 2015 DSL shared task. Goutte et al. (2016) presents a comprehensive evaluation of state of-the-art language identification systems trained to recognize similar languages and language varieties using the results of the first two DSL shared tasks. Their experimental results suggest that humans also find it difficult discriminating between similar languages and language varieties. This year, DSL 2016 shared task proposed two subtasks: first sub-task is about discriminating between similar languages and national language varieties. Second sub-task is about Arabic dialect identification which is introduced first time in DSL 2016 shared task. We have participated in the sub-task2 of dialect identification on Egyptian, Gulf, Levantine, and North-African, and Modern Standard Arabic (MSA) Arabic dialects. We describe about dataset used for dialect classification in section 4.

In classifying Arabic dialects, Elfardy and Diab (2013), Malmasi and Dras (2014), Zaidan and Callison-Burch (2014), Darwish et al. (2014) and Malmasi et al. (2015) employed supervised and semsupervised learning methods with and without ensembles and meta classifiers with various levels of word, character and morphological features. Most of these approaches are sensitive to the topic bias in the language and use expensive set of features and limited to short texts. Moreover, generating these features can be a tedious and complex process. In this paper, we propose deep learning based supervised techniques for Arabic dialect identification without the need for expensive feature engineering. Inspired by the advances in sentence classification (Kim, 2014) and sequence classification (Hochreiter and Schmidhuber, 1997) using distributional word representations, we use convolutional neural networks (CNN) and long short-term memory (LSTM)-based deep neural network approaches for Arabic dialect identification.

The rest of the paper is organized as follows: in section 2, we describe related work on Arabic dialect classification. In section 3, we introduce two deep learning based supervised classification techniques and describe about the proposed methodology. We give a brief overview about the dataset used in the shared task in section 4, and also we present experimental results on dialect classification. In section 5, we discuss about results and analyse various types of errors in dialect classification and conclude the paper. Additional analysis and comparison with the other submitted systems are available in the 2016 shared task overview (Malmasi et al., 2016)

2 Related Work

In recent years, a very few researchers have attempted the task of automatic Arabic dialect identification. Zaidan and Callison-Burch (2011) developed an informal monolingual Arabic Online Commentary (AOC) annotated dataset with high dialectal content. Authors in this work applied language modelling approach and performed dialect classification tasks on 4 dialects (MSA and three dialects) and two dialects (Egyptian Arabic and MSA) and reported 69.4% and 80.9% accuracies respectively. Several other researchers (Elfardy and Diab, 2013; Malmasi and Dras, 2014; Zaidan and Callison-Burch, 2014; Darwish et al., 2014) also used the same AOC and Egyptian-MSA datasets and employed different categories of supervised classifiers such as Naive Bayes, SVM, and ensembles with various rich lexical features such as word and character level n-grams, morphological features and reported the improved results.

Malmasi et al. (2015) presented a number of Arabic dialect classification experiments namely multi-dialect classification, pairwise binary dialect classification and meta multi-dialect classification using the Multidialectal Parallel Corpus of Arabic (MPCA) dataset. Authors achieved 74% accuracy on a 6-dialect classification and 94% accuracy using pairwise binary dialect classification within the corpus but reported poorer results (76%) between Palestinian and Jordanian closely related dialects. Authors also reported that a meta-classifier can yield better accuracies for multi-class dialect identification and shown that models trained with the MPCA corpus generalize well to other corpus such as AOC dataset. They demonstrated that character n-gram features uniquely contributed for significant improvement in accuracy in intra-corpus and cross-corpus settings. In contrast, Zaidan and Callison-Burch (2011; Elfardy and Diab (2013; Zaidan and Callison-Burch (2014) shown that word unigram features are the best features

for Arabic dialect classification. Our proposed approach do not leverage rich lexical, syntactic features, instead learns abstract representation of features through deep neural networks and distributional representations of words from the training data. Proposed approach handles n-gram features with varying context window-sizes sliding over input words at sentence level.

Habash et al. (2008) composed annotation guidelines for identifying Arabic dialect content in the Arabic text content, by focusing on code switching. Authors also reported annotation results on a small data set (1,600 Arabic sentences) with sentence and word-level dialect annotations.

Biadsy et al. (2009; Lei and Hansen (2011) performed Arabic dialect identification task in the speech domain at the speaker level and not at the sentence level. Biadsy et al. (2009) applied phone recognition and language modeling approach on larger (170 hours of speech) data and performed four-way classification task and reported 78.5% accuracy rate. Lei and Hansen (2011) performed three-way dialect classification using Gaussian mixture models and achieved an accuracy rate of 71.7% using about 10 hours of speech data for training. In our proposed approach, we use ASR textual transcripts and employ deep-neural networks based supervised sentence and sequence classification approaches for performing multi-dialect identification task.

In a more recent work, Franco-Salvador et al. (2015) employed word embeddings based continuous Skip-gram model approach (Mikolov et al., 2013a; Mikolov et al., 2013b) to generate distributed representations of words and sentences on HispaBlogs[1] dataset, a new collection of Spanish blogs from five different countries: Argentina, Chile, Mexico, Peru and Spain. For classifying intra-group languages, authors used averaged word embedding sentence vector representations and reported classification accuracies of 92.7% on original text and 90.8% accuracy after masking named entities in the text. In this approach, authors utilizes sentence vectors generated from averaged word embeddings and uses logistic regression or Support Vector Machines (SVMs) for detecting dialects where as in our proposed approach, we build the task of dialect identification using end to end deep neural representation by learning abstract features and feature combinations through multiple layers. Our results are not directly comparable with this work as we use different Arabic dialect dataset.

3 Methodology

Deep neural networks, with or without word embeddings, have recently shown significant improvements over traditional machine learning–based approaches when applied to various sentence- and document-level classification tasks.

Kim (2014) have shown that CNNs outperform traditional machine learning–based approaches on several tasks, such as sentiment classification, question type classification, and subjectivity classification, using simple static word embeddings and tuning of hyper-parameters. Zhang et al. (2015) proposed character level CNN for text classification. Lai et al. (2015; Visin et al. (2015) proposed recurrent CNN while Johnson and Zhang (2015) proposed semi-supervised CNN for solving text classification task. Palangi et al. (2016) proposed sentence embedding using LSTM network for information retrieval task. Zhou et al. (2016) proposed attention-based bidirectional lstm Networks for relation classification task. RNNs model text sequences effectively by capturing long-range dependencies among the words. LSTM-based approaches based on RNNs effectively capture the sequences in the sentences when compared to the CNN and SVM-based approaches. In subsequent sub sections, we describe our proposed CNN and LSTM based approaches for multi-class dialect classification.

3.1 CNN-based Dialect Classification

Collobert et al. (2011) adapted the original CNN proposed by LeCun and Bengio (1995) for modelling natural language sentences. Following Kim (2014), we present a variant of the CNN architecture with four layer types: an input layer, a convolution layer, a max pooling layer, and a fully connected softmax layer. Each dialect in the input layer is represented as a sentence (dialect) comprised of distributional word embeddings. Let $v_i \in \mathbb{R}^k$ be the k-dimensional word vector corresponding to the ith word in the

[1] https://github.com/autoritas/RD-Lab/ tree/master/data/HispaBlogs

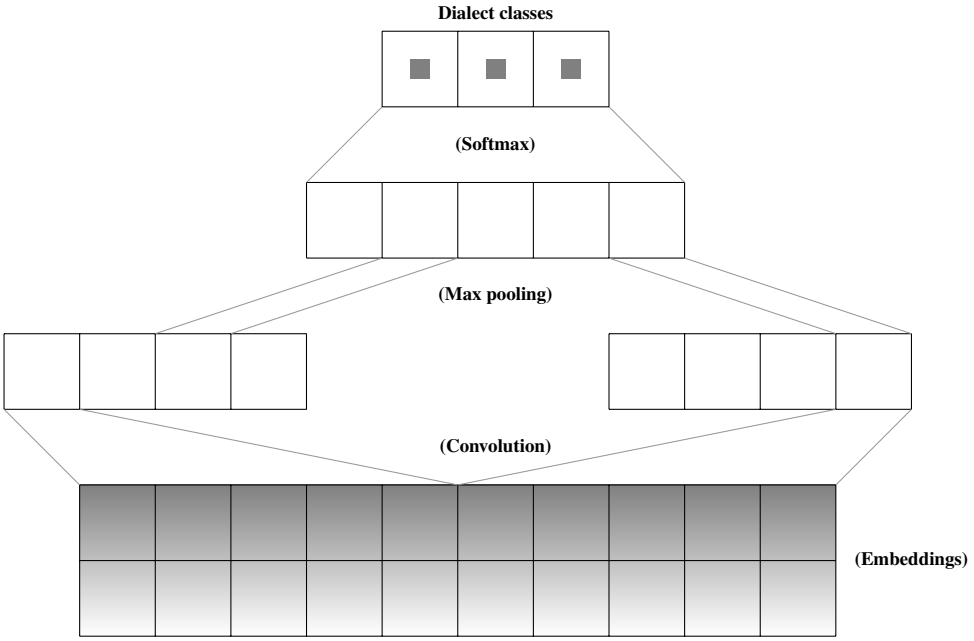

Figure 1: Illustration of convolutional neural networks with an example dialect

sentence. Then a dialect S of length ℓ is represented as the concatenation of its word vectors:

$$S = v_1 \oplus v_2 \oplus \cdots \oplus v_\ell. \tag{1}$$

In the convolution layer, for a given word sequence within a dialect, a convolutional word filter P is defined. Then, the filter P is applied to each word in the dialect to produce a new set of features. We use a non-linear activation function such as rectified linear unit (ReLU) for the convolution process and max-over-time pooling (Collobert et al., 2011; Kim, 2014) at pooling layer to deal with the variable dialect size. After a series of convolutions with different filters with different heights, the most important features are generated. Then, this feature representation, Z, is passed to a fully connected penultimate layer and outputs a distribution over different labels:

$$y = \text{softmax}(W \cdot Z + b), \tag{2}$$

where y denotes a distribution over different dialect labels, W is the weight vector learned from the input word embeddings from the training corpus, and b is the bias term.

3.2 LSTM-based Dialect Classification

In case of CNN, concatenating words with various window sizes, works as n-gram models but do not capture long-distance word dependencies with shorter window sizes. A larger window size can be used, but this may lead to data sparsity problem. In order to encode long-distance word dependencies, we use long short-term memory networks, which are a special kind of RNN capable of learning long-distance dependencies. LSTMs were introduced by Hochreiter and Schmidhuber (1997) in order to mitigate the vanishing gradient problem (Gers et al., 2000; Gers, 2001; Graves, 2013; Pascanu et al., 2013).

The model illustrated in Figure 2 is composed of a single LSTM layer followed by an average pooling and a softmax regression layer. Each dialect is represented as a sentence (S) in the input layer. Thus, from an input sequence, $S_{i,j}$, the memory cells in the LSTM layer produce a representation sequence $h_i, h_{i+1}, \ldots, h_j$. Finally, this representation is fed to a softmax layer to predict the dialect classes for unseen input dialects.

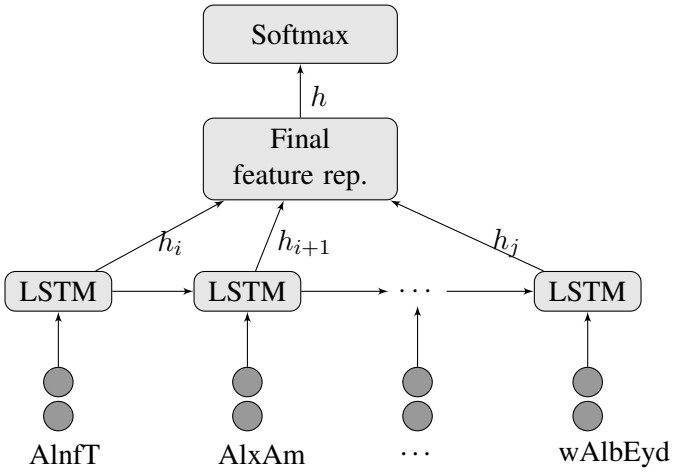

Figure 2: Illustration of LSTM networks with an example dialect

3.3 Experimental Setup

We modeled dialect classification as a sentence classification task. We tokenized the corpus with white space tokenizer. We performed multi-class 5-way classification on the given arabic data set containing 5 language dialects. We used Kim's (2014) Theano implementation of CNN [2] for training the CNN model and a variant of the standard Theano implementation[3] for training the LSTM network. We initialized and used the randomly generated embeddings in both the CNN and LSTM models in the range $[-0.25, 0.25]$. We used 80% of the training set for training and 20% of the data for validation set and performed 5-fold cross validation in CNN. In LSTM, we used 80% of the given training set for building the model and rest 20% of the data is used as development set. We updated input embedding vectors during the training. In the CNN approach, we used a stochastic gradient descent–based optimization method for minimizing the cross entropy loss during the training with the Rectified Linear Unit (ReLU) non-linear activation function. We used default window filter sizes set at $[3, 4, 5]$. In the case of LSTM, model was trained using an adaptive learning rate optimizer-adadelta (Zeiler, 2012) over shuffled mini-batches with the sigmoid activation function at input, output and forget gates and tanh non-linear activation function at cell state. Post competition we performed experiments without and with average pooling using LSTM networks and reported the results as shown in tables 5 and 6.

Hyper Parameters. We used hyper-parameters such as drop-out for avoiding over-fitting), and batch size and learning rates on 20% of the cross-validation/development set. We varied batch sizes, drop-out rate, embedding sizes, and learning rate on development set. We obtained the best CNN performance with learning rate decay 0.95, batch size 50, drop-out 0.5, and embedding size 300 and ran 20 epochs on cross validated dataset. For LSTM, we got the best results on development set with learning rate 0.001, drop-out 0.5, and embedding size 300, batch-size of 32 and at 12 epochs. We used same settings similar to the development set but varied drop-out rate over [0.5,0.6,0.7] and obtained best results on test set using drop-out 0.7. We obtained best results on test set with drop-out 0.5 using average pooling.

Pre-compiled Embeddings. We used the gensim (ehek and Sojka, 2010) word2vec program to compile embeddings from the given training corpus. We compiled 300-dimensional embedding vectors for the words that appear at least 3 times in the Arabic dialect corpus, and for rest of the vocabulary, embedding vectors are assigned uniform distribution in the range of $[-0.25, 0.25]$. We used these pre-compiled embeddings in LSTM and reported run2 results in the test set.

4 Datasets and Results

In this section we describe about DSL 2016 shared task data sets and the experimental results.

[2] https://github.com/yoonkim/CNN_sentence
[3] http://deeplearning.net/tutorial/lstm.html

	egy	glf	lav	msa	nor	Total
Train	1578	1671	1758	999	1612	7618
Test	315	256	344	274	351	1540
Total	1893	1927	2102	1273	1963	9158

Table 1: The distribution of training and test data sets

4.1 Datasets

In 2016, for the first time the DSL shared task included a sub-task on Arabic dialects for 5 dialects: Egyptian, Gulf, Levantine, North-African, and Modern Standard Arabic (MSA) As dialects are mostly used in conversational speech, DSL 2016 shared task supplied training and test datasets (Malmasi et al., 2016) containing ASR transcripts. Test set contains uniform distribution of dialects related to ASR texts. The distribution of training and test splits are shown in table 1. The samples in test set are slightly unbalanced.

4.2 Results

We evaluated the given test set using both LSTM and CNN and presented the results as shown in table 2. DSL shared task results are evaluated using weighted-F1 measure for ranking of various participating systems. Due to the imbalance of classes in the test set, majority baseline is used in this Arabic dialect classification task. We have obtained run1 results (0.1779 F1-weighted) with LSTM-based dialect classification model using random embedding weights at the input layer. Run2 results (0.1944 F1) are obtained using LSTM-model with pre-compiled word embeddings. Though run2 results are better than run1 but LSTM-model poorly performed when compared to the base line results (0.2279 weighted-F1) on the test set. We have obtained fairly comparable results on experimental held-out development set without pre-compiled embeddings as shown in table 3. We identified that the poor results on test set are due to the bug in the code of LSTM-results compilation. Post competition, we fixed the bug and re-evaluated results on test set as shown in tables 5 and 6. We observe that LSTM without using pooling before the softmax layer, performed slightly better (0.4231 F1-weighted) than using average pooling (0.4172 F1-weighted). LSTM without pooling classified 'egy', 'msa' and 'nor' dialects more accurately than the LSTM with average pooling. LSTM with average pooling performed better than the LSTM without pooling in classifying 'glf' and 'lav' dialect classes. Run 3 results are obtained using CNN classification model without using pre-compiled embeddings. We observe that the CNN performance (0.4329 F1-weighted) is better than the LSTM performance (0.4231 F1-weighted). The performance of different dialect classes accuracy using CNN is visualized in the confusion matrix as shown in figure 3. We also present the 5-fold cross validation results as shown in the table 4. CNN in cross validation setting outperformed LSTM-results on development set in four dialect classes (egy,lav,msa,nor) where as LSTM performed better in case of 'glf' dialect classification. It took 24 hours to perform 5-fold cross validation using CNN on a single CPU, 8-GB RAM, Intel, i7-processor machine. We have also tried building model using CNN and LSTM on sub-task1 but took 10 days of time to train on entire training set and unable to test it on the test set and produce results in-time. The limitation of CNN and LSTM is that they need more time to train on on CPU machines and this can be avoided by using GPU machines.

Test Set	Track	Run	**Accuracy**	**F1 (micro)**	**F1 (macro)**	**F1 (weighted)**
run1	C	closed	0.1961	0.1961	0.1715	0.1779
run2	C	closed	0.2162	0.2162	0.1876	0.1944
run3	C	closed	0.4377	0.4377	0.4364	0.**4329**
baseline	-	-	-	-	-	0.2279

Table 2: Results for test set C for all runs (closed training).

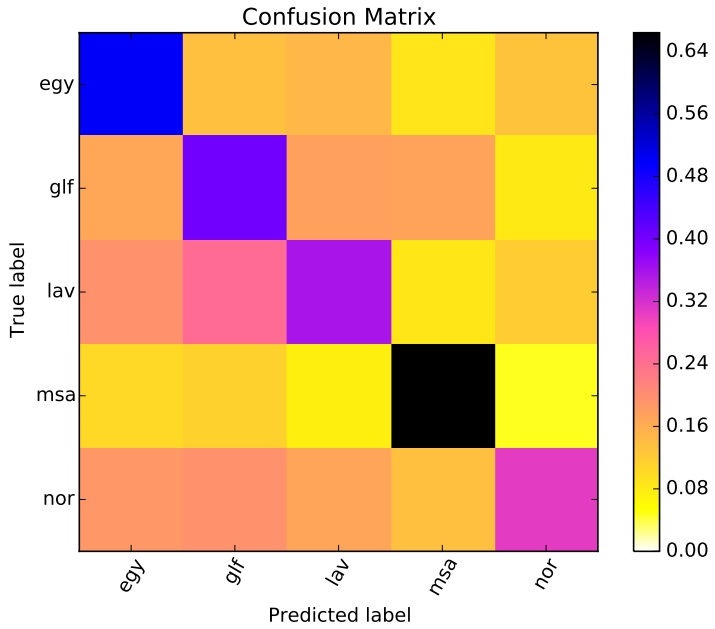

Figure 3: Run3 confusion matrix using CNN multi-class classification

	Precision	Recall	f1-score
egy	0.5694	0.5484	0.5587
glf	0.4444	0.5562	0.4940
lav	0.4704	0.4389	0.4541
msa	0.5922	0.6731	**0.6301**
nor	0.5444	0.4171	0.4723

Table 3: LSTM experimental results (run1) on development set without embeddings after 12 epochs of training.

5 Discussion and Conclusion

We can assess the degree of confusion between various dialect classes from the confusion matrix of CNN classification model as shown in figure 3. MSA and Egypt are the dialects that are more accurately identified when compared to the other dialects. North-african and Laventine have the highest degree of confusion, mostly with Egypt and gulf Arabic dialects. This might be due to the geographically in close contact with these languages. We also observe significant amount of confusion between gulf and the Egyptian and leventine dialects. In our experiments, we observed that CNN performed better than the LSTM for Arabic dialect classification. There are number of potential directions to improve the dialect classification accuracy. One possible future work might be to compile the common vocabulary among most confusing dialect classes and for these vocabulary compile the word embeddings from large, unlabeled dialect corpora using neural networks, and encode both syntactic and semantic properties of words. Studies have found the learned word vectors to capture linguistic regularities and to collapse similar words into groups (Mikolov et al., 2013b).

As our proposed CNN model is built using default network parameters, tuning of hyper-parameters can significantly improve the dialect classification accuracy and this will be considered as our future work. Learning word embeddings from the larger dialect corpus and using them in the input layer of CNN and LSTM networks can also improve the dialect classification accuracy. Since Arabic language dialects are morphologically rich and pose various syntactic and semantic challenges at word level, experimenting with character level CNNs and bi-directional LSTMs can be more useful for accurate classification of

	Precision	Recall	f1-score
egy	0.5582	0.6363	0.5947
glf	0.4716	0.4629	0.4672
lav	0.6153	0.4861	0.5432
msa	0.6597	0.7356	**0.6956**
nor	0.5750	0.6174	0.5954

Table 4: CNN Average 5-fold cross-validation results (run3) without embeddings after 20 epochs

	Precision	Recall	f1-score
egy	0.4444	0.4190	0.4314
glf	0.3172	0.2305	0.2670
lav	0.4179	0.4215	0.4197
msa	0.4605	0.6606	**0.5427**
nor	0.4637	0.4188	0.4401
F1 (macro)	-	-	**0.4202**
F1 (weighted)	-	-	**0.4232**

Table 5: LSTM experimental results on test set **without pooling**

	Precision	Recall	f1-score
egy	0.4353	0.3523	0.3895
glf	0.2678	0.3516	0.3040
lav	0.4059	0.4389	0.4218
msa	0.5301	0.5146	**0.5222**
nor	0.4662	0.4131	0.4381
F1 (macro)	-	-	**0.4151**
F1 (weighted)	-	-	**0.4172**

Table 6: LSTM experimental results on test set **with average pooling**

various Arbaic dialects. As our proposed approach do not rely much on language specific analysis on the corpus, it can be easily adapted to more similar languages, varieties and classification tasks.

References

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.

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

Kareem Darwish, Hassan Sajjad, and Hamdy Mubarak. 2014. Verifiably effective arabic dialect identification. In *EMNLP*, pages 1465–1468.

Heba Elfardy and Mona T Diab. 2013. Sentence level dialect identification in arabic. In *ACL (2)*, pages 456–461.

Marc Franco-Salvador, Paolo Rosso, and Francisco Rangel. 2015. Distributed representations of words and documents for discriminating similar languages. In *Proceedings of the Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects (LT4VarDial)*, pages 11–16, Hissar, Bulgaria.

Felix A. Gers, Jürgen Schmidhuber, and Fred Cummins. 2000. Learning to forget: Continual prediction with LSTM. *Neural Computation*, 12(10):2451–2471.

Felix Gers. 2001. *Long Short-term Memory in Recurrent Neural Networks*. Ph.D. thesis, Universität Hannover.

Cyril Goutte and Serge Leger. 2015. Experiments in discriminating similar languages. In *Proceedings of the Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects (LT4VarDial)*, pages 78–84, Hissar, Bulgaria.

Cyril Goutte, Serge Léger, and Marine Carpuat. 2014. The nrc system for discriminating similar languages. In *Proceedings of the First Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects (VarDial)*, pages 139–145, Dublin, Ireland.

Cyril Goutte, Serge Léger, Shervin Malmasi, and Marcos Zampieri. 2016. Discriminating Similar Languages: Evaluations and Explorations.

Alex Graves. 2013. Generating sequences with recurrent neural networks. *CoRR*, abs/1308.0850.

Nizar Habash, Owen Rambow, Mona Diab, and Reem Kanjawi-Faraj. 2008. Guidelines for annotation of arabic dialectness. In *Proceedings of the LREC Workshop on HLT & NLP within the Arabic world*, pages 49–53.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Rie Johnson and Tong Zhang. 2015. Semi-supervised convolutional neural networks for text categorization via region embedding. In *Advances in neural information processing systems*, pages 919–927.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1746–1751.

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. In *AAAI*, pages 2267–2273.

Yann LeCun and Yoshua Bengio. 1995. Convolutional networks for images, speech, and time series. In Michael A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*, pages 255–258. MIT Press, Cambridge, MA.

Yun Lei and John HL Hansen. 2011. Dialect classification via text-independent training and testing for arabic, spanish, and chinese. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(1):85–96.

Shervin Malmasi and Mark Dras. 2014. Arabic native language identification. In *Proceedings of the Arabic Natural Language Processing Workshop (EMNLP 2014)*, pages 180–186. Citeseer.

Shervin Malmasi and Mark Dras. 2015. Language identification using classifier ensembles. In *Proceedings of the Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects (LT4VarDial)*, pages 35–43, Hissar, Bulgaria.

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, May.

Shervin Malmasi, Marcos Zampieri, Nikola Ljubešić, Preslav Nakov, Ahmed Ali, and Jörg Tiedemann. 2016. Discriminating between similar languages and arabic dialect identification: A report on the third dsl shared task. In *Proceedings of the 3rd Workshop on Language Technology for Closely Related Languages, Varieties and Dialects (VarDial)*, Osaka, Japan.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013a. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751.

Hamid Palangi, Li Deng, Yelong Shen, Jianfeng Gao, Xiaodong He, Jianshu Chen, Xinying Song, and Rabab Ward. 2016. Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(4):694–707.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning*, volume 3, pages 1310–1318.

Radim ehek and Petr Sojka. 2010. Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*.

Francesco Visin, Kyle Kastner, Kyunghyun Cho, Matteo Matteucci, Aaron Courville, and Yoshua Bengio. 2015. Renet: A recurrent neural network based alternative to convolutional networks. *arXiv preprint arXiv:1505.00393*.

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.

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

Marcos Zampieri and Binyam Gebrekidan Gebre. 2012. Automatic identification of language varieties: The case of portuguese. In *KONVENS2012-The 11th Conference on Natural Language Processing*, pages 233–237. Österreichischen Gesellschaft für Artificial Intelligende (ÖGAI).

Marcos Zampieri, Liling Tan, Nikola Ljubešić, and Jörg Tiedemann. 2014. A report on the dsl shared task 2014. In *Proceedings of the First Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects (VarDial)*, pages 58–67, Dublin, Ireland.

Marcos Zampieri, Binyam Gebrekidan Gebre, Hernani Costa, and Josef van Genabith. 2015a. Comparing approaches to the identification of similar languages. In *Proceedings of the Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects (LT4VarDial)*, pages 66–72, Hissar, Bulgaria.

Marcos Zampieri, Liling Tan, Nikola Ljubešić, Jörg Tiedemann, and Preslav Nakov. 2015b. Overview of the dsl shared task 2015. In *Proceedings of the Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects (LT4VarDial)*, pages 1–9, Hissar, Bulgaria.

Matthew D. Zeiler. 2012. ADADELTA: An adaptive learning rate method. *CoRR*, abs/1212.5701.

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.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *The 54th Annual Meeting of the Association for Computational Linguistics*, page 207.