Neural Network Architectures for Arabic Dialect Identification

Elise Michon, Minh Quang Pham, Josep Crego, Jean Senellart

SYSTRAN / 5 rue Feydeau, 75002 Paris, France
firstname.lastname@systrangroup.com

Abstract

SYSTRAN competes this year for the first time to the DSL shared task, in the Arabic Dialect Identification subtask. We participate by training several Neural Network models showing that we can obtain competitive results despite the limited amount of training data available for learning. We report our experiments and detail the network architecture and parameters of our 3 runs: our best performing system consists in a Multi-Input CNN that learns separate embeddings for lexical, phonetic and acoustic input features (F1: 0.5289); we also built a CNN-biLSTM network aimed at capturing both spatial and sequential features directly from speech spectrograms (F1: 0.3894 at submission time, F1: 0.4235 with later found parameters); and finally a system relying on binary CNN-biLSTMs (F1: 0.4339).

1 Introduction

Dialect identification (DID) consists in automatically identifying the corresponding dialect of an utterance, either written or spoken. This task is a particularly challenging case of language identification since dialects are closely related languages. It is not only useful but often a requirement for various Natural Language Processing (NLP) tasks such as Machine Translation (MT) or Automatic Speech Recognition (ASR). In the context of the shared task *Discriminating between Similar Languages, Varieties and Dialects (DSL)*, dialect identification can be seen as a multi-class sentence classification problem, in which participants must predict a label for each sentence, given several features describing the sentence.

We present our results for the Arabic Dialect Identification (ADI) subtask, where the similar languages to discriminate are Modern Standard Arabic and four dialects of Arabic: Egyptian, Gulf, Levantine and North African. Given their high success in many other NLP tasks and lower cost in feature engineering compared to more traditional machine learning methods, in this paper we mostly focus on the design of suitable Neural Networks, knowing that the limited size of the training dataset is a well known handicap for such models as already pointed out in previous editions of the DSL workshop.

2 Related Work

Arabic speakers are used to write in Modern Standard Arabic and express orally with Arabic dialects. Although closely related, dialects differ lexically, morphologically, phonetically and prosodically. Recently, the increasing use of social media has seen the rise of spoken and written materials with Arabic dialects, which motivates related NLP tasks such as Arabic Dialect Identification (Zaidan and Callison-Burch, 2014). To tackle this challenge, from 2016 the DSL shared task has proposed an ADI subtask with multi-dialectal Arabic data based on audio files accompanied with dialect labels. Best performances have so far been reached by Support Vector Machine (SVM), Kernel Ridge Regression (KRR), and sophistications of these traditional classifiers like ensemble methods. However, in this section we focus on describing the performance of neural networks in previous editions.

In 2016 (Malmasi et al., 2016), the data represented each utterance using a transcription in words obtained using the ASR engine described in (Ali et al., 2014). The best performing systems obtained F1

Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects, pages 128–136
Santa Fe, New Mexico, USA, August 20, 2018.

scores ranging from 0.495 to 0.513. Three teams reported experiments with neural network architectures that were finally not submitted as their models following other machine learning methods obtained higher accuracy scores, QCRI (Eldesouki et al., 2016), GW_ LT3 (Zirikly et al., 2016) and tufbasfs (Çöltekin and Rama, 2016). Two teams submitted systems training neural networks: the team cgli competed using a character-level Convolutional Neural Network (CNN) with the combination of filters 3*128, 4*128, 5*128 (convention adopted throughout the whole article: filter size*number of filters) inspired from (Kim, 2014) which gave a macro F1 score of 0.433. They also reported in their paper a Long-Short Term Memory network (LSTM) with pre-compiled word embeddings giving as F1 score 0.423 after bug correction (Guggilla, 2016). The team mitsls used another character-level CNN architecture based on (Kim et al., 2016) using filters of increasing size 1*50, 2*50, 3*100, 4*100, 5*100, 6*100, 7*100 which gave a better F1 score 0.483 and ranked 2^{nd} in the competition (Belinkov and Glass, 2016).

Character-level CNNs present the advantage to be fast and able to learn local representations, which can be likened to char n-grams features successfully used by SVMs. One reason advanced by (Sadat et al., 2014) for the efficiency of character-level representations for Arabic Dialect Identification in speech transcription is that a great part of the variation between Arabic dialects is based on their affixes. However, as noticed by the organisers, Arabic speakers distinguish Arabic dialects not only according to words but also on the basis of speech cues absent from written transcripts. So for the DSL shared task 2017 (Zampieri et al., 2017), the ADI dataset contained twice more utterances than in 2016 and not only speech transcriptions but also acoustic features corresponding to sentences, namely i-vectors modelled on bottleneck features extracted from an ASR-Deep Neural Network as described in (Ali et al., 2016). This time the only team submitting neural networks, deepCybErNetRun, competed with a bidirectional Long Short Term Memory network (biLSTM) on words (F1: 0.208) and a biLSTM on i-vectors (F1: 0.574) but did not publish any description paper. Their lower performance compared to the best system of the competition, achieving a F1 score of 0.763, in this subtask and other subtasks of the challenge led to the conclusion that the size of the DSL data was insufficient for tuning the numerous parameters of a neural network.

3 Methodology and Data

3.1 Data description

This 2018 edition saw the apparition of phonetic features in addition to lexical and acoustic features, to represent sentences with finer-grained information. Furthermore, end-to-end deep learning approaches based on Mel-Frequency Cepstrum Coefficients (MFCCs) or spectrograms recently proved to provide better acoustic representation for dialect identification than previously used i-vectors. Thus this year ADI's dataset contained: word transcripts in Buckwalter by an ASR system; phone transcripts according to 4 phoneme recognisers (Czech, English, Hungarian, Russian) from Brno University of Technology including phonemes and non-phonetic units (*int* for intermittent noise, *pau* for short pause and *spk* for non-speech speaker noise); and 600-size acoustic embeddings extracted as the last fully-connected layer before the softmax layer in an end-to-end CNN system trained for audio dialect identification, all detailed in (Shon et al., 2018).

Acoustic embeddings released for the challenge were trained only on the train set to enable participants to use the dev set for parameter tuning, while acoustic embeddings from the system reported in (Shon et al., 2018) were trained on the train and dev set, leading to an increase in performance, and were made available after submission date. Two test datasets were released as a single one during the testing phase, and were later made available separately: one is the test set from the MGB-3 challenge made of extracts of multi-domain Youtube videos (Ali et al., 2017), the other is a Youtube surprise test dataset. The distribution of the 5 dialects Egyptian (EGY), Gulf (GLF), Levantine (LAV), Modern Standard Arabic (MSA) and North African (NOR) in the datasets is shown in Table 1.

We note a few facts about word transcripts: first, transcription is empty for a significant proportion of the sentences as reported in Table 1. For some audio files, the sentence is truly barely intelligible whereas for others, the sentence is clear but the sound file seems to have a lower volume. Second, word transcripts tend to contain less unknown words compared to previous years but many words are missing from the

Dialect		EGY	GLF	LAV	MSA	NOR	Total
Number of utterances	train	3177	2873	3117	2219	3205	14591
	dev	315	265	348	238	355	1566
	MGB-3 test	302	250	334	262	344	1492
	Youtube test	1143	1147	1131	944	980	5345
Empty word transcripts (%)	train	2.90	5.78	5.68	0.50	10.58	5.38
	dev	5.71	2.26	6.03	1.06	2.54	3.64
	MGB-3 test	1.66	1.20	1.50	0.00	1.45	1.21
	Youtube test	4.37	2.09	12.82	1.06	8.47	5.84

Table 1: Distribution of dialects and percentage of empty word transcripts in 2018 ADI datasets.

transcription. Finally, transcriptions seem dominated by MSA, with some dialectal phrases transcribed as MSA sequences.

This encouraged us towards methods directly leveraging acoustic information. We considered log-amplitude mel-spectrograms as an alternative acoustic representation computed on audio files thanks to the python library Librosa (McFee et al., 2015), resulting in an input shape of sequence length x 40. Nevertheless, inspection of the audio data reveals some inconsistencies in the attribution of the labels, sometimes based on the speaker, sometimes on the spoken dialect: for instance, some extracts where an Egyptian speaker is speaking Modern Standard Arabic are categorised as EGY, other similar ones are categorised as MSA (e.g. interviewed speaker in the news). Code-switching between dialectal Arabic and Modern Standard Arabic is also found to be common, especially in the EGY, GLF and LAV dialects. We finally noticed that some of the audio files were duplicates (e.g. up to 6% of the train set for North African dialect).

3.2 System description

As already outlined, previous editions of the ADI subtask showed how more traditional classifiers, like SVM or KRR, outperformed neural network approaches. In this edition, SYSTRAN participates by training several neural network models to show that we can also obtain competitive results compared to such classifiers. We are mostly interested in how neural networks perform for this task, due to their ability to learn adequate representation for the data. Not only do we investigate multi-input systems making the most of the various features on the sentences given in the challenge, but also end-to-end systems directly working on acoustic representation of speech data.

3.2.1 SVM

In order to compare traditional machine learning and neural network approaches, we trained a multi-class Support Vector Machine (SVM) classifier using a radial basis function. We used the freely available LIBSVM[1] software (Chang and Lin, 2011).

3.2.2 Multi-Input CNN (run 1)

Our search for a simple and fast architecture to independently learn input embeddings of different type and combine them oriented us towards the Multi Group Convolutional Neural Network, also called Multi-Input CNN (Zhang et al., 2016). Initially designed to join different word embeddings, these models allow the input embeddings to come from various sources and not to share the same dimensionality.

Therefore, our first run is a Multi-Input CNN that we tailored to take as input the lexical, phonetic and acoustic data proposed for the challenge: it independently learns char embeddings and 4 phone embeddings by running convolutions with various filter sizes, respectively char-level convolutions on word transcripts and phone-level convolutions on phone transcripts. Then it concatenates the 5 resulting embeddings and the given acoustic embeddings for the sentences, adds fully-connected layers and finally predicts the dialect. We implemented the model in Keras (Chollet and others, 2015) with Tensorflow back-end and made the code freely available[2].

[1] https://www.csie.ntu.edu.tw/~cjlin/libsvm/

[2] https://github.com/elisemicho/multi_input_classification

3.2.3 CNN-biLSTM (run 2)

In an attempt to improve the classification by better controlling the extracted acoustic features, we also decided to design and test several configurations of end-to-end neural networks taking as input acoustic representations of audio files and as output the dialect label. Motivated by their good performance in Music Classification (Choi et al., 2017) and Language Identification (Ganapathy et al., 2014), we chose CNN to slide over the sequence and learn representations of the signal. CNN are efficient in terms of locally representing information, but speech data are also sequences where earlier or later information can provide context for the current window. Therefore the combination of a Recurrent Neural Network (RNN) such as LSTM or Gated Recurrent Units network (GRU) to summarise temporal patterns after a CNN enables to capture both spatial and sequential features: (Choi et al., 2017) outperformed several CNN-only architectures by using a 4-layer CNN with 2D filters of size [3,3] followed by a RNN with GRU. Similarly in (Bartz et al., 2017), a 2D CNN with filters of decreasing size in increasing number [7,7]*16, [5,5]*32, [3,3]*64, [3,3]*128, [3,3]*256 followed by a biLSTM led to promising improvement for smaller dataset compared to similar CNN.

Hence, our second run is a CNN-biLSTM neural network that we designed to take as input the log-amplitude mel-spectrograms obtained on the original audio files: layers of one-dimension convolutions with decreasing filter sizes but increasing number of filters compose the CNN part of the system. A bidirectional LSTM layer then takes its result and the initial sequence lengths of the signal to link the current window with previous and next windows in the sequence, and outputs a final prediction of the dialect. Our implementation used TensorFlow (Abadi et al., 2015) and its code is freely available[3].

3.2.4 Binary classification with CNN-biLSTM (run 3)

Our third run is a system using 5 CNN-biLSTM neural networks of the type previously described (run 2) that we adapted to perform binary classification (one dialect against the others). For each utterance, the network predicting that it belongs to its positive class with the highest probability wins the final decision:

$$class(utterance) = argmax_{i \in \{1,..,5\}} P(1|net_i, utterance)$$

4 Results

We present the results of our three runs in Table 2. Comparable to our SVM results (0.5270), our Multi-Input CNN achieves the highest F1 score of 0.5289, in line with other participants this year and ranking 3^{rd} in the competition. Contrary to our expectations, our CNN-biLSTM directly operating on audio data failed to learn better acoustic representations for dialect identification with a F1 score of 0.3894. However, using CNN-biLSTMs for binary classification and taking the maximum probability improved the performance to 0.4339. Both test sets come from Youtube videos, but all our systems performed notably better on MGB-3 test set, which is of much smaller size. As for the classification results of our best run summarised in Figure 1, the biggest error rate is for Levantine utterances often predicted to belong to North African dialect, whereas these dialects are neither geographically nor typologically close.

System	F1 (macro) on test sets		
	MGB-3	Youtube	Total
Random Baseline			0.1995
SVM	**0.5632**	0.5143	0.5270
Multi-input CNN (run 1)	0.5552	**0.5186**	**0.5289**
CNN-biLSTM (run 2)	0.4380	0.3711	0.3894
Binary CNN-biLSTMs (run3)	0.4600	0.4241	0.4339

Table 2: Results for the ADI task (macro-averaged F1 scores).

In the following subsections, we report our tests to tune the different models, placing a greater emphasis on Neural Networks. We also examine how the provided features, the various refinements or the model elements perform separately and in combination.

[3] `https://github.com/elisemicho/escolta`

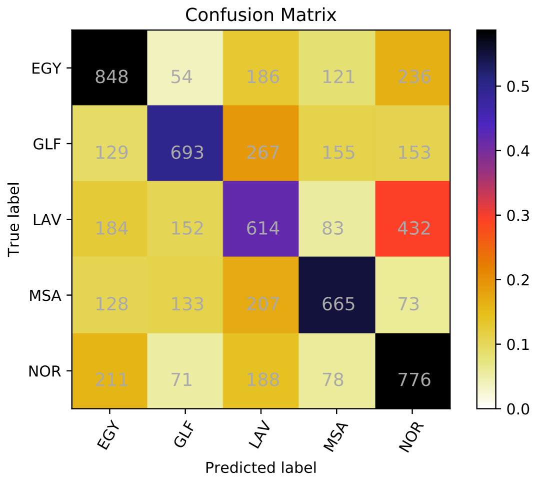

Figure 1: Confusion Matrix for Multi-Input CNN (run 1).

4.1 SVM

In Table 3 we present F1 scores we obtained with SVMs using only one or a combination of the features. Classification with lexical features gives better results (F1: 0.4502) than with phonetic features (F1: 0.2891), but classification with acoustic features only (F1: 0.5239) outperforms them both. Acoustic features therefore seem the most useful representation of the data for this dialect identification task. The combinations of features only achieve slightly higher results (F1: 0.5270), and not for the MGB-3 test set. The new acoustic embeddings trained on train and dev sets (released after submission date) further enhance performance up to 0.5905.

Model	**Test sets**		
	MGB-3	**Youtube**	**Total**
Only lexical features	0.4493	0.4494	0.4502
Only phonetic features	0.3035	0.2839	0.2891
Only acoustic features	**0.5656**	0.5097	0.5239
Lexical + acoustic features	0.5630	0.5138	0.5267
Lexical + phonetic + acoustic features	0.5632	**0.5143**	**0.5270**
Only new acoustic features	0.7334	0.5379	0.5823
Lexical + new acoustic features	0.7350	0.5439	0.5873
Lexical + phonetic + new acoustic features	**0.7376**	**0.5470**	**0.5905**

Table 3: Results of the SVM models (macro-averaged F1 scores).

4.2 Multi-Input CNN (run 1)

Model	**Test sets**		
	MGB-3	**Youtube**	**Total**
Only lexical features	0.2988	0.3635	0.3505
Only phonetic features	0.3347	0.3251	0.3307
Only acoustic features	0.5495	0.5100	0.5209
Lexical + acoustic features	0.5483	0.5075	0.5184
Lexical + phonetic + acoustic features (run 1)	**0.5552**	**0.5186**	**0.5289**
Only new acoustic features	**0.7260**	0.5363	**0.5791**
Lexical + new acoustic features	0.7105	**0.5374**	0.5767
Lexical + phonetic + new acoustic features	0.7212	0.5258	0.5697

Table 4: Results of the Multi-Input CNN models (macro-averaged F1 scores).

We selected as our best configuration for Multi-Input CNN char embeddings and phone embeddings of size 32, learnt separately through 1D convolutions with filters [5*8, 3*8] (filter size*number), $tanh$ activation function, dropout 0.5 and global max-pooling; then once concatenated, one fully-connected layer of size 32, ReLu activation function, dropout 0.5. We achieve the best accuracy on dev set after 7 epochs, but we note that on the train set the loss is already very low and the accuracy very high from the beginning of the 2nd epoch, signalling probable overfitting. Our tests of higher (64) or lower (16, 8) size of embeddings, higher dropout (0.7), higher filter sizes [3, 5, 7], and two fully-connected layers of size 16 all led to comparable, slightly lower results. In Table 4 we present F1 scores we obtained with systems relying on this configuration, selectively using one, some or all of the features. Comparably with SVM, we notice that acoustic features alone achieve similar performance to the combination of features, suggesting that classification in our system mostly relies on acoustic information. We observe a clear jump in performance when using the new acoustic embeddings trained on train and dev sets, especially on MGB-3 test set.

4.3 CNN-biLSTM (run 2)

We built a CNN-biLSTM firstly with two layers of 1D convolutions using filters of decreasing size, each followed by ReLU activation function, dropout 0.5, and max-pooling, then a bi-LSTM of comparable hidden unit size with the number of filters in the last convolutional layer and finally a softmax layer. Both SGD with learning rate 0.001 and Adam with learning rate 0.0001 performed well. Dynamic padding was applied to batches to save computational resources. At the time of submission, we achieved our best score on the dev set with filters 8*200, 4*400, which yielded a F1 score of 0.3894 on the test set. After submission, we found that an earlier of our models, simply decreasing the number of filters to 8*64, 4*64, outperforms the previous configuration with an F1 score of 0.4235 on test set as shown in Table 5.

Conv	**Filters and Options**	**Test sets**		
		MGB-3	**Youtube**	**Total**
1D	8*200, 4*400 (run 2)	0.4380	0.3711	0.3894
1D	8*200, 4*400 with masking	0.3587	0.2843	0.3013
1D	8*200, 4*400 with masking + batch normalization	0.2506	0.1932	0.2062
1D	**8*64, 4*64**	**0.4614**	**0.4098**	**0.4235**
1D	8*64, 4*64 with masking + balanced batch	0.3542	0.3046	0.3174
1D	8*64, 4*64 with batch normalization	0.1421	0.1390	0.1398
1D	3*3, 3*3, 3*3, 3*3	0.0749	0.0620	0.0649
2D	[3x3]*3, [3x3]*3, [3x3]*3, [3x3]*3	0.0652	0.0620	0.0763
2D	[7x7]*16, [5x5]*32, [3x3]*64, [3x3]*128, [3x3]*256	0.2507	0.2771	0.2721

Table 5: Results for the CNN-biLSTM models (macro-averaged F1 scores). Layers are described with the convention filter size*number of filters.

Optimisation tricks such as masking (to compute the convolutions only on the signal and not on the padding), balancing the batch (so that it necessarily contains items from the 5 classes) or applying batch normalisation after convolutions, surprisingly all negatively impacted the performance. We found that 2D convolutions that could have learnt even more localised features in the spectrograms (Choi et al., 2017; Bartz et al., 2017), were computationally expensive, even when we reduced batch size to 5 instead of 10, and less successful than 1D convolutions in our case. We note that CNN alone and biLSTM alone perform at the random baseline (F1: 0.1995), suggesting that learning in our CNN-biLSTM comes from the combination of the two architectures.

4.4 Binary classification with CNN-biLSTM (run 3)

	F1 in binary systems		**F1 in final system**
	This dialect	**Other dialects**	**This dialect**
EGY	0.50	0.78	0.49
GLF	0.53	0.84	0.54
LAV	0.39	0.82	0.31
MSA	0.38	0.92	0.37
NOR	0.47	0.84	0.47
F1 (macro)			**0.43**

Table 6: Results by dialect in the 5 binary systems and final system (F1 scores).

Run 2 and Run 3 give similar error profiles, the latter of which is displayed in Figure 2: Egyptian, Gulf and North African dialects are in majority correctly classified, but what is striking is the high precision but poor recall in classification of Modern Standard Arabic and the high uncertainty of the model for Levantine. This suggests that our CNN-biLSTM can usefully recognise that the 4 dialects are different from MSA but fails to recognise Levantine or MSA utterances, assigning a label at chance level. A tentative explanation of this confusion is the high presence of MSA in utterances of other dialects, namely in the Egyptian, Gulf and Levantine audio files. As shown in Table 6, the F1 score of each dialect in our final system of CNN-biLSTMs is virtually identical with the F1 score in the binary systems, which entails no real benefit was gained from combining the binary classifiers. Thus, Levantine and MSA for which the system exhibits the highest confusion and that are actually typologically the closest dialects, present the lowest F1 score in binary systems.

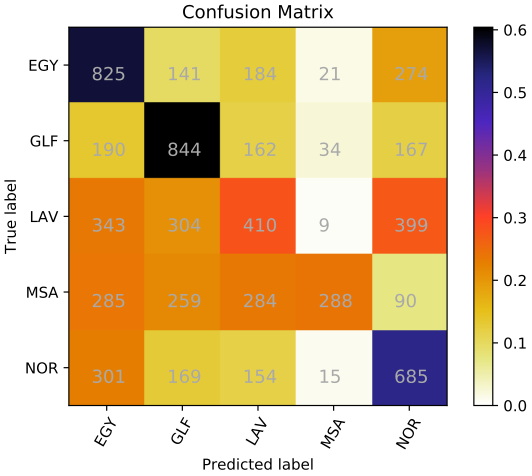

Figure 2: Confusion Matrix for binary CNN-biLSTMs (run 3).

5 Conclusion

In this paper we described our first contribution to the Arabic Dialect Identification subtask in the DSL shared task: our best run is a Multi-Input CNN, learning independent embeddings on lexical and phonetic features and concatenating them with the provided acoustic features to make a prediction. This architecture stays relatively simple yet highly parallel and comparable to more traditional machine learning methods as it ranked 3^{rd} in the competition. Since most of the classification power of this system came from the acoustic features, we investigated end-to-end models directly operating on speech data and generating acoustic features, by means of a CNN-biLSTM and its use for binary classification. Even if these runs proved more informative than competitive, we would like to encourage research with neural networks in future DSL shared tasks despite of the limited size of datasets as it still seems embryonic and evolving at that stage. Furthermore, this approach follows the trend to leave dialect-specific or task-specific features and shift to simpler but powerful architectures, learning useful representations for Dialect Identification while optimising another task such as Automatic Speech Recognition (Li et al., 2017). We were actually led to VarDial DSL shared task by our research in written Arabic Dialect Identification. However, the relative scarcity of word transcripts and their discrepancy with human evaluation, especially for dialectal utterances, do not allow to consider them as realistic data. We advocate that the existence of a separate task for written Arabic Dialect Identification, as MGB challenge exists for spoken Arabic Dialect Identification, would be fruitful for the research community interested in ADI since text (either formal or informal) is a frequently encountered raw data type.

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, Yifan Zhang, Patrick Cardinal, Najim Dahak, Stephan Vogel, and James Glass. 2014. A complete kaldi recipe for building arabic speech recognition systems. In *Spoken Language Technology Workshop (SLT), 2014 IEEE*, pages 525–529. IEEE.

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.

Ahmed Ali, Stephan Vogel, and Steve Renals. 2017. Speech Recognition Challenge in the Wild: Arabic MGB-3. *arXiv preprint arXiv:1709.07276*.

Christian Bartz, Tom Herold, Haojin Yang, and Christoph Meinel. 2017. Language identification using deep convolutional recurrent neural networks. In *International Conference on Neural Information Processing*, pages 880–889. Springer.

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.

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.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27.

Keunwoo Choi, György Fazekas, Mark Sandler, and Kyunghyun Cho. 2017. Convolutional recurrent neural networks for music classification. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 2392–2396. IEEE.

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

Mohamed Eldesouki, Fahim Dalvi, Hassan Sajjad, and Kareem Darwish. 2016. QCRI @ DSL 2016: Spoken Arabic Dialect Identification Using Textual Features. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pages 221–226, Osaka, Japan.

Sriram Ganapathy, Kyu Han, Samuel Thomas, Mohamed Omar, Maarten Van Segbroeck, and Shrikanth S Narayanan. 2014. Robust language identification using convolutional neural network features. In *Fifteenth annual conference of the international speech communication association*.

Chinnappa Guggilla. 2016. Discrimination between Similar Languages, Varieties and Dialects using CNN- and LSTM-based Deep Neural Networks. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pages 185–194, Osaka, Japan.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2016. Character-aware neural language models. In *AAAI*, pages 2741–2749.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Bo Li, Tara N Sainath, Khe Chai Sim, Michiel Bacchiani, Eugene Weinstein, Patrick Nguyen, Zhifeng Chen, Yonghui Wu, and Kanishka Rao. 2017. Multi-dialect speech recognition with a single sequence-to-sequence model. *arXiv preprint arXiv:1712.01541*.

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.

Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. 2015. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, pages 18–25.

Fatiha Sadat, Farnazeh Kazemi, and Atefeh Farzindar. 2014. Automatic identification of arabic dialects in social media. In *Proceedings of the first international workshop on Social media retrieval and analysis*, pages 35–40. ACM.

Suwon Shon, Ahmed Ali, and James Glass. 2018. Convolutional Neural Networks and Language Embeddings for End-to-End Dialect Recognition. *ArXiv e-prints arXiv:1803.04567*.

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

Marcos Zampieri, Shervin Malmasi, Nikola Ljubešić, Preslav Nakov, Ahmed Ali, Jörg Tiedemann, Yves Scherrer, and Noëmi Aepli. 2017. Findings of the VarDial Evaluation Campaign 2017. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, Valencia, Spain.

Ye Zhang, Stephen Roller, and Byron Wallace. 2016. Mgnc-cnn: A simple approach to exploiting multiple word embeddings for sentence classification. *arXiv preprint arXiv:1603.00968*.

Ayah Zirikly, Bart Desmet, and Mona Diab. 2016. The GW/LT3 VarDial 2016 Shared Task System for Dialects and Similar Languages Detection. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pages 33–41, Osaka, Japan.