QMDIS: QCRI-MIT Advanced Dialect Identification System

Sameer Khurana[1], *Maryam Najafian*[2], *Ahmed Ali*[1], *Tuka Al Hanai*[2], *Yonatan Belinkov*[2], *James Glass*[2]

[1]Qatar Computing Research Institute, HBKU, Doha, Qatar
[2]MIT Computer Science and Artificial Intelligence Laboratory (CSAIL), Cambridge, MA, USA
{skhurana, amali}@qf.org.qa, {najafian, tuka, belinkov, glass}@mit.edu

Abstract

As a continuation of our efforts towards tackling the problem of spoken Dialect Identification (DID) for Arabic languages, we present the QCRI-MIT Advanced Dialect Identification System (QMDIS). QMDIS is an automatic spoken DID system for Dialectal Arabic (DA). In this paper, we report a comprehensive study of the three main components used in the spoken DID task: phonotactic, lexical and acoustic. We use Support Vector Machines (SVMs), Logistic Regression (LR) and Convolutional Neural Networks (CNNs) as backend classifiers throughout the study. We perform all our experiments on a publicly available dataset and present new state-of-the-art results. QMDIS discriminates between the five most widely used dialects of Arabic: namely Egyptian, Gulf, Levantine, North African, and Modern Standard Arabic (MSA). We report $\approx 73\%$ accuracy for system combination. All the data and the code used in our experiments are publicly available for research.

Index Terms: Spoken Dialect Identification, Arabic, Phonotactic, Acoustic, Lexical, Logistic Regression, Support Vector Machine, Convolutional Neural Network

1. Introduction

The task of Dialect identification (DID) consists of classifying a given spoken utterance into one of the many dialects spoken in a particular language. DID is similar to the more general problem of language identification (LID). DID is more challenging than LID because of the small and subtle differences between the various dialects of the same language. A good DID system can be used to extract dialectal data from the speech database to train dialect specific acoustic models for speech-to-text transcription. A DID system built using Deep Learning models such as Convolutional Neural Networks (CNNs) can be used to generate dialect codes (embeddings) that can be used for dialect adaptation of Neural Network acoustic models for Automatic Speech Recognition (ASR). The aforementioned approach is similar to using speaker codes (i-vectors) for speaker adaptation of acoustic models while building an ASR system [1].

Although the problem of LID is far from solved, the field of spoken LID has flourished in the past decade and great advancements have been made in this direction. The three methods proposed in literature for LID are Lexical, Phonotactic and Acoustic. Lexical and Phonotactic approaches capture the n-gram word (or character) and phone statistics respectively from the word and phone transcripts. The transcripts are generated using a speech-to-text transcription system and one or more phone recognizers. The n-gram statistics are then used to construct a Vector Space Model of spoken utterances, where each utterance is represented as a vector encoding its n-gram word (or character) or phone statistics. A classifier such as SVM is then used to find a decision boundary in the n-gram Vector Space [2]. Acoustic methods work with low-level acoustic features such as Mel Frequency Cepstral Coefficients (MFCCs), Shifted Delta Cepstral Coefficients (SDCs) [3] and prosody [4]. The acoustic features are modeled using Gaussian Mixture Models (GMMs) and the popular i-vector modeling framework [5]. Other approaches such as using frame-by-frame phone posteriors (PLLRs) [6] as features for LID and non-negative factor analysis (NFA) for GMM weight decomposition and adaptation have also been explored [7] .

In this work, we continue our investigation on Arabic DID. The Arabic language can be broadly divided into five major dialects; namely Egyptian (EGY), Gulf (GLF) or Arabian Peninsula, Levantine (LAV), Modern Standard Arabic (MSA) and North African (NOR) or Maghrebi. As argued in [2], there are sufficient differences between the various Arabic dialects such that they can be treated as different languages, and the problem is similar to that of LID. We make the same assumption in this paper. We present a comprehensive study of the three methods for spoken DID: lexical & phonotactic (§ 3) and acoustic (§ 2), focusing on Acoustic and Lexical methods. Inspired by the successful use of Convolutional Neural Networks (CNNs) for text classification [8] and acoustic modeling [9, 10], we extend our previous work [2] by investigating the applicability of CNNs to directly map the raw acoustic features to the corresponding dialect, unlike the traditional *latent variable modeling* approach, popularly known as the *i-vector* framework. We present the results of our investigation in § 4.2

2. Acoustic Methods for Spoken DID

2.1. Input Features

We parametrize the speech signal by extracting *Mel-frequency cepstral coefficients* (MFCCs) per 25 ms sliding window over the speech signal, having a 10 ms overlap. The MFCC feature vector is enriched using *shifted delta cepstral coefficients* (SDCs) [3]. We use the configuration 7-1-3-7 for extracting the MFCC-SDC features, similar to the one used in [11]. The aforementioned approach gives us a sequence of feature vectors for each spoken utterance, which is fed as input to the CNN. We use Kaldi [12], a publicly available Automatic Speech Recognition toolkit for feature extraction.

2.2. Convolutional Neural Network

Recently [11], encouraging results have been reported when using Deep Learning Methods directly on raw acoustic features for Spoken LID. In this work, we use a CNN as a mapping function from raw acoustic features (MFCC-SDC) to corresponding dialects.

We experiment with two CNN architectures: 1) A simple CNN (CNN_A) with one convolution layer followed by global max pooling [13] along the time axis and a fully connected hid-

den layer that gives the final speech representation. The representation is then fed to a softmax layer which outputs the final prediction.

2) A more complex CNN (CNN_B), with three convolution layers. The first two layers are followed by *Max Pooling* operation, while the third layer is followed by *Global Max Pooling*. The third convolution layer is followed by a fully connected hidden layer with dropout to get the final representation. The final representation is fed to a *softmax* layer that outputs the final prediction. The architecture is same as used in [10] for the purpose of encoding the spoken utterance into a fixed vector representation.

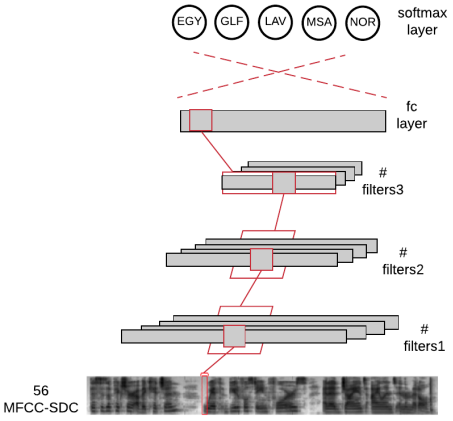

Figure 1: *CNN_B: CNN acting on 56 dimensional MFCC-SDC features. Architecture inspired by [10]*

Learning: During the learning phase, we tune the hyper-parameters of CNN_A and CNN_B using 5-fold cross-validation. Cross-validation is performed on the training dataset and the final evaluation is conducted on the test set.

Hyper-parameters for CNN_A are: 1) Number of filters in the convolution layer (*nb_filters*), 2) filter size (*filter_length)*, 3) number of hidden units (h) in the fully connected layer (f_c) and 4) f_c dropout (f_c*_drop*). We perform a grid search over the possible parameter values and the optimal values in our case are: *nb_filters:* 1024, *filter_length*: 4, h: 1024 and f_c*_drop*: 0.2

Hyper-parameters for CNN_B are: 1) the convolution layers setting given by $\{f_1, f_2, f_3 : nb_1, nb_2, nb_3\}$, where f_i is the filter size in the convolution layer i and nb_i is the number of corresponding filters. 2) The number of neurons on the fully connected layer, h. We fix the pool length and stride to be 4 and 2 respectively for the max pooling operation. The optimal parameter values for CNN_B is are: $\{2, 4, 4 : 32, 64, 128\}$ and h: 256, found using 5-fold cross-validation.

2.3. Baseline: I-vector

We use the i-vector system as a comparison with the CNN dialect classifier. I-vectors are extracted using the standard pipeline [2]. First, we extract the bottleneck feature (BNF) representations from the MFCC representation by using a Deep Neural Network (DNN) based ASR system. Details about the DNN and the BNF extraction are given in our previous work [2] and we do not repeat them here due to space constraints.

The BNF are fed into the i-vector modeling framework. It consists of building a Universal Background Model (UBM) GMM on a large amount of speech data represented using BNFs. GMM-UBM's mean and variance statistics give a general picture of the data spread in the high dimensional Vector Space. The UBM mean *supervector* is is adapted to each utterance. This update information is encoded in a low-dimensional latent vector known as an i-vector. In this work, the GMM-UBM model has 2048 Gaussian components, MFCC features are extracted using a 25 ms window and the i-vectors are 400 dimensional. For more details we refer to our previous work [2]. We also perform Linear Discriminant Analysis (LDA) and Within-Class Co-variance Normalization (WCCN), the two commonly used post-processing operations that are shown to improve the DID/LID performance [5].

The resulting i-vectors are input to a discriminative classifier. Here, we experiment with Support Vector Machine (SVM) and Logistic Regression (LR) as the two backend classifiers. Hyper-parameters for the SVM are distance from the hyperplane (C) and *penalty*. The optimal values are: C : 0.01 and *penalty:* $l2$. For LR, the learning rate (α) for the *Stochastic Gradient Descent* algorithm and *penalty* are the hyper-parameters. The optimal values are: α: 0.001 and *penalty*: $l1$. Hyper-parameters are tuned using 5-fold CV. The search method used to find the hyperparameters is *grid-search*.

3. Lexical & Phonotactic Methods

3.1. Feature Extraction

The feature extraction steps consist of extracting the phone sequence, and phone duration statistics from three recognizers trained on Arabic language. We conduct an experiment with a grapheme based lexicon (setup (1)), as well as pronunciation rules from the literature by El-Imam et al. [14] (setup (2)), and Biadsy et al. [15] under a setup presented by Hanai et al. [16] (setup (3)). Set up (1) uses a grapheme lexicon and its Gaussian Mixture Model (GMM) based acoustic models are trained on 1200 hours of Arabic broadcast speech. For more details see the system description in [17]. Set ups (2) and (3) were trained using 70 hours of GALE Broadcast Conversation. In setup (2), the lexicon was generated after diacratizations of the words in the transcript using the MADA+TOKAN Toolkit 3.2 with SAMA 3.1 [18], and the phone level transcriptions are extracted using a GMM based acoustic model. In setup (3), we generated pronunciations of these diacratized words according to linguistic based pronunciation rules developed by Biadsy et al. [15], and the phone level transcriptions are extracted using a GMM based acoustic model. We use a repetition mechanism such that phones with longer duration are represented by longer sequences. We estimated the mean, M, and standard deviation, S, of the phone durations, D, in the dev set. If $D < M - \alpha S$ the phone, W, will be repressed as a single unit. If $M - \alpha S < D < M$the phone, W, will be shown as two units, WW. If $M < D < M + \beta D$, the phone will be shown as three unites, WWW, else it will be shown as four consecutive phones $WWWW$. Here, α and β are weight values that are trained from the dev set. Here, α and β are weight values that are trained from the dev set (10% split of the training set (§ 4.1)).

Word sequences are extracted using a state-of-the-art Arabic speech-to-text transcription system built as part of the Multi-Genre Broadcast Challenge (MGB-2) [19]. The system is a combination of a Time Delayed Neural Network (TDNN), a Long Short-Term Memory Recurrent Neural Network (LSTM) and Bidirectional LSTM acoustic models, followed by 4-gram and Recurrent Neural Network (RNN) language model rescor-

ing. Our system uses a grapheme lexicon during both training and decoding. The acoustic models are trained on 1200 hours of Arabic broadcast speech. We also perform data augmentation (speed and volume perturbation) which gives us three times the original training data. For more details see the system description paper [17].

3.2. Convolutional Neural Network

We borrow the recent ideas on sentence classification from the *Natural Language Processing* (NLP) community [8, 20], and investigate their applicability on word and phone sequence classification obtained using a speech transcriber.

We experiment with two CNN architectures for word sequence classification: 1) Word-CNN, which is same as the Simple CNN (CNN_A) used for acoustic modeling (§ 2). Instead of an acoustic feature vector, we have input from the word embedding layer and 2) Char-CNN (Fig 2) that has a number of parallel convolution layers [21]. Each convolution layer can be seen as projecting the input sequence into a character n-gram Vector Space, where n is the size of the filter in the convolution layer and the number of such vector spaces is given by the number of filters. Filter settings for Char-CNN is given by: $\{f_1, f_2, f_3, \ldots, f_k: nb_1, nb_2, nb_3, \ldots, nb_k\}$, which refers to k parallel convolutions with filter sizes $\{f_i\}_{i=1\ to\ k}$ with corresponding filters, $\{nb_i\}_{i=1\ to\ k}$.

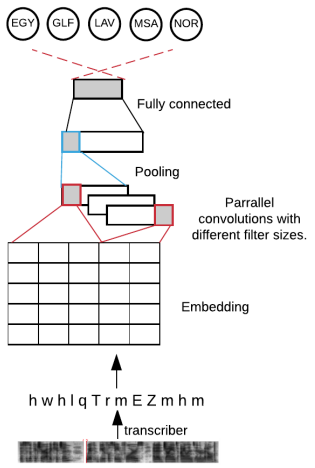

Figure 2: *Character CNN*

Learning: Hyper-parameters for the word-CNN are: 1) Word Embedding dimensions (*embedding_dims*), 2) size of the filter in the convolution layer (*filter_length*), 3) number of filters (*nb_filters*), 4) number of neurons (h) in the fully connected (f_c) layer and 5) f_c dropout. The optimal parameters found using 5-fold CV are: *embedding_dims*: 64, *filter_length*: 8, *nb_filters*: 256, h: 256 and f_c dropout: 0.2.

Hyper-parameters for the char-CNN are: 1) Char (or phone) embedding dimensions (*embedding_dims*), 2) *filter setting* ($\{f_1, f_2, f_3, \ldots, f_k: nb_1, nb_2, nb_3, \ldots, nb_k\}$) , 3) number of neurons (h) in the fully connected (f_c) layer and 4) f_c dropout. The optimal parameters for Char-CNN found using 5 fold CV are: *filter setting*: {1, 2, 3, 4, 5, 6, 7, 8, 9: 50, 50, 100, 100, 100, 100, 100, 100, 100}, *embedding_dims*: 128, h: 512 and f_c dropout: 0.2.

We fix the embedding layer dropout to 0.2, batch size to 64 and training epochs to 20 for both word-CNN and char-CNN.

3.3. Baselines

We train LR and SVM classifiers on n-gram bag-of-words feature representations to use as baselines. Linear classifiers with bag-of-word features have been shown to outperform more complex architectures for the task of Arabic DID [22].

Hyper-parameters for the SVM are distance from the hyperplane (C), *penalty* and *n-gram range*. The optimal values for word n-grams are: C : 0.01, *penalty:* $l2$ and *n-gram range*: 1-3 and for character n-grams are: C : 0.001, *penalty:* $l2$ and *n-gram range*: 2-11. For LR, the learning rate (α) for the *SGD* algorithm, *penalty* and *n-gram range* are the hyper-parameters. For word-based features, the optimal values are: α: 0.0001 *penalty*: $l1$ and *n-gram range*: 1-5 and for character n-gram features, the optimal values are: α: 0.001 *penalty*: $l2$ and *n-gram range*: 1-7 Hyper-parameters are tuned using 5-fold CV and *grid-search*

4. Experiments

4.1. Data Corpus

We use the publicly available Arabic Dialect Identification corpus used in vardial2017[23]. Table 1 gives the details about the corpus. Training is used for system development and the test set is used for final system evaluation. The training set is a mix of in-domain and out-of-domain data collected from the *youtube* Arabic channels and Aljazeera news channel's official dialectal speech database, while the test set is extracted from the Aljazeera news channel.

	Training			Test		
	Sent.	Dur.	Words	Sent.	Dur.	Words
EGY	5k	23.4	87k	302	2	11.6k
GLF	4.7k	21.9	67.9k	250	2.1	12.3k
LAV	4.9k	20.6	63.3k	334	2	10.9k
MSA	4.2k	23.8	82.4k	262	1.9	13k
NOR	4.9k	20.4	47.1k	344	2.1	10.3k
Tot.	15524	110.1	347.5k	1492	10.1	58.1k

Table 1: *Data Corpus. # training & test sentences, # words, speech duration (Dur.) in hours*

4.2. Results

Acoustic Methods: Table 2 presents the DID performance of the acoustic methods. The performance is not competitive with the i-vector modeling framework, but the combination of a CNN system with the i-vector system gives us performance improvements.

		Accuracy	
Model	ID	Cross Val	Test set
CNN_A (§ 2.2)	$\mathcal{A}_1$	0.65	0.47
CNN_B (§ 2.2)	$\mathcal{A}_2$	0.70	0.51
LR (i-vec 400d)	$\mathcal{B}_1$	0.86	0.63
SVM (i-vec 400d)	$\mathcal{C}_1$	0.85	0.63
LR (i-vec + LDA + WCCN)	$\mathcal{D}_1$	0.87	0.66
SVM (i-vec + LDA + WCCN)	$\mathcal{F}_1$	0.88	0.67
$\mathcal{A}_2 + \mathcal{D}_1 + \mathcal{F}_1$	$\mathcal{E}_{acoustic}$	-	**0.69**

Table 2: *DID Results for Acoustic Methods.*

Lexical & Phonotactic Methods: Table 3 gives DID performance when using lexical methods. Systems built using word and character n-gram features are complimentary and their combination gives us performance improvements. Although the word-CNN and char-CNN are not able to beat the linear classifiers, the results are comparable. Combination of CNN systems with n-gram features based system gives further improvements.

		Accuracy	
Model	ID	Cross Val	Test set
LR (word 5-gram)	$\mathcal{A}_1$	0.65	0.59
SVM (word 3-gram)	$\mathcal{B}_1$	0.64	0.56
Word-CNN	$\mathcal{C}_1$	0.63	0.57
$\mathcal{A}_1$+ $\mathcal{B}_1$ + $\mathcal{C}_1$	$\mathcal{E}_1$	-	0.61
LR (char 9-gram)	$\mathcal{D}_1$	0.66	0.58
SVM (char 11-gram)	$\mathcal{F}_1$	0.67	0.59
Char-CNN	$\mathcal{G}_1$	0.65	0.55
$\mathcal{D}_1$ + $\mathcal{F}_1$ + $\mathcal{G}_1$	$\mathcal{E}_2$	-	0.62
$\mathcal{E}_1$ + $\mathcal{E}_2$	$\mathcal{E}_{lex}$	-	**0.64**

Table 3: *DID Results for Lexical Methods.*

Table 4 presents the DID performance using the Phonotactic methods. We experiment with three types of phone sequences generated by the transcription system using grapheme based lexical (setup (1)), MADA (setup (2)), and Biadsy pronunciation (setup (3)) rules, based on phone n-gram sequences, as well as phone sequences in which individual phones are repeated based on the phone duration (Rep. Phone Sequence).

System	Test Acc
Phone sequence + Char-CNN setup (1)	0.56
Rep. Phone Seq. + Char-CNN setup (1)	0.57
Phone sequence + Char-CNN setup (2)	0.58
Rep. Phone Seq. + Char-CNN setup (2)	0.57
Phone sequence + Char-CNN setup (3)	0.57
Rep. Phone Seq. + Char-CNN setup (3)	0.58

Table 4: *DID Results for Phonotactic Methods.*

Final Combination: Performing score averaging of the output scores of the best lexical, $\mathcal{E}_{lex}$, and acoustic system, $\mathcal{E}_{acoustic}$, for DID gives us an accuracy of **73%** on the test set.

5. Discussion

Looking at the confusion matrix it can be inferred that Gulf is the most confused dialect, most often with Levantine (LAV) and Modern Standard Arabic (MSA). The second most confused dialect is North African, most often with Levantine. It can be inferred that it is difficult to distinguish among the following three dialects; Gulf, Levantine and North African.

We hypothesize that the reason our DID system performs worst in case of Gulf and North African dialects is due to code switching[24], where the same speaker alternates between two dialects in the context of a single conversation. We perform a further investigation into the error patterns for utterances of different duration. Assuming that the speech is spoken in a single dialect, the DID accuracy should increase as the duration of the speech utterances increases. Gulf and North African do not follow the aforementioned pattern (See Fig 4), unlike other dialects. This leads us to believe that there is code switching

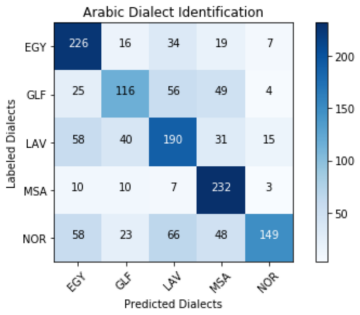

Figure 3: *DID Confusion Matrix for the final combined system.*

in the spoken utterances of these two dialects which make them the two most difficult dialects to recognize.

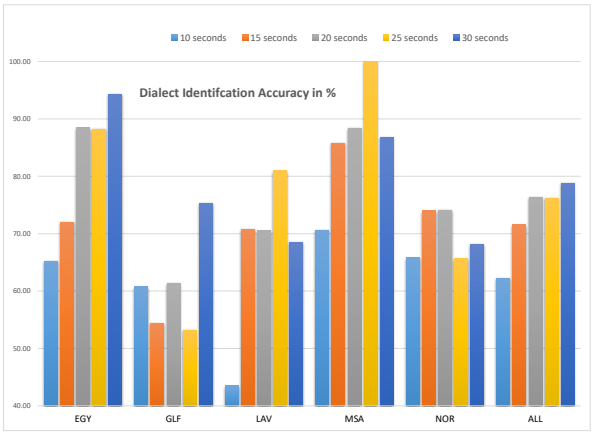

Figure 4: *DID accuracy over 10-30 seconds bins.*

6. Conclusions

In this paper, we present a comprehensive performance study of Spoken DID methods for the Arabic language. Along with investigating the traditional methods for DID such as i-vector and n-gram lexical features in a linear classifier, we also investigate the feasibility of using Convolutional Neural Networks for direct mapping of acoustic and lexical features to one of the five dialects. For our future work, we would continue our investigation into approaches that can directly map the raw acoustic waveform to the corresponding dialects. In particular, we would explore Long Short-Term Memory RNN to make dialect predictions per frame. The frame by frame prediction would also give us a picture of code switching between dialectal speech and MSA. Another line of research worth exploring is the effect of adding dialectal data collected from sources such as Youtube and radio podcasts during the classifier training.

7. References

[1] Y. Miao, H. Zhang, and F. Metze, "Speaker adaptive training of deep neural network acoustic models using i-vectors," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, pp. 1938–1949, 2015.

[2] A. Ali, N. Dehak, P. Cardinal, S. Khurana, S. H. Yella, J. Glass, P. Bell, and S. Renals, "Automatic Dialect Detection in Arabic Broadcast Speech," in *INTERSPEECH*, 2016, pp. 2934–2938.

[3] P. A. Torres-Carrasquillo, E. Singer, M. A. Kohler, R. J. Greene, D. A. Reynolds, and J. R. Deller Jr, "Approaches to language identification using Gaussian mixture models and shifted delta cepstral features," in *INTERSPEECH*, 2002.

[4] D. Martínez, L. Burget, L. Ferrer, and N. Scheffer, "iVector-based prosodic system for language identification," in *ICASSP*, 2012, pp. 4861–4864.

[5] N. Dehak, P. A. Torres-Carrasquillo, D. A. Reynolds, and R. Dehak, "Language Recognition via i-vectors and Dimensionality Reduction," in *INTERSPEECH*, 2011.

[6] O. Plchot, M. Diez, M. Soufifar, and L. Burget, "PLLR Features in Language Recognition System for RATS," in *INTERSPEECH*, 2014.

[7] M. H. Bahari, N. Dehak, L. Burget, A. M. Ali, J. Glass *et al.*, "Non-negative Factor Analysis of Gaussian Mixture Model Weight Adaptation for Language and Dialect Recognition," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 22, no. 7, pp. 1117–1129, 2014.

[8] Y. Kim, "Convolutional Neural Networks for Sentence Classification," *arXiv preprint arXiv:1408.5882*, 2014.

[9] D. Amodei, R. Anubhai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos *et al.*, "Deep Speech 2: End-to-end Speech Recognition in English and Mandarin," *arXiv preprint arXiv:1512.02595*, 2015.

[10] D. Harwath, A. Torralba, and J. Glass, "Unsupervised Learning of Spoken Language with Visual Context," in *Advances in Neural Information Processing Systems*, 2016, pp. 1858–1866.

[11] R. Zazo, A. Lozano-Diez, J. Gonzalez-Dominguez, D. T. Toledano, and J. Gonzalez-Rodriguez, "Language Identification in Short Utterances Using Long Short-Term Memory (LSTM) Recurrent Neural Networks," *PLOS one*, vol. 11, no. 1, p. e0146917, 2016.

[12] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The Kaldi speech recognition toolkit," in *ASRU*, no. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *European Conference on Computer Vision*. Springer, 2014, pp. 346–361.

[14] Y. A. El-Imam, "Phonetization of arabic: rules and algorithms," *Computer Speech & Language*, vol. 18, no. 4, pp. 339–373, 2004.

[15] F. Biadsy, N. Habash, and J. Hirschberg, "Improving the arabic pronunciation dictionary for phone and word recognition with linguistically-based pronunciation rules," in *NAACL*, 2009, pp. 397–405.

[16] J. G. Tuka Al Hanai, "Lexical modeling for arabic asr: A systematic approach," in *INTERSPEECH*, 2014, pp. 2605–2609.

[17] S. Khurana and A. Ali, "QCRI Advanced Transcription System (QATS) for the Arabic Multi-Dialect Broadcast Media Recognition: MGB-2 Challenge," in *SLT*, 2016.

[18] N. Habash, O. Rambow, and R. Roth, "Mada+ tokan: A toolkit for arabic tokenization, diacritization, morphological disambiguation, pos tagging, stemming and lemmatization," in *Proceedings of the 2nd international conference on Arabic language resources and tools (MEDAR), Cairo, Egypt*, vol. 41, 2009, p. 62.

[19] A. Ali, P. Bell, J. Glass, Y. Messaoui, H. Mubarak, S. Renals, and Y. Zhang, "The MGB-2 Challenge: Arabic Multi-Dialect Broadcast Media Recognition," *arXiv preprint arXiv:1609.05625*, 2016.

[20] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in neural information processing systems*, 2015, pp. 649–657.

[21] Y. Belinkov and J. Glass, "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 (VarDial)*, 2016.

[22] M. Eldesouki, F. Dalvi, H. Sajjad, and K. Darwish, "QCRI@ DSL 2016: Spoken Arabic Dialect Identification Using Textual," *VarDial 3*, p. 221, 2016.

[23] M. Zampieri, S. Malmasi, N. Ljubešic, P. Nakov, A. Ali, J. Tiedemann, Y. Scherrer, and N. Aepli, "Findings of the vardial evaluation campaign 2017," *VarDial 2017*, 2017.

[24] P. Auer, *Code-Switching in Conversation: Language, Interaction and Identity*. Routledge, 2013.