MIT-QCRI ARABIC DIALECT IDENTIFICATION SYSTEM FOR THE 2017 MULTI-GENRE BROADCAST CHALLENGE

Suwon Shon[1], *Ahmed Ali*[2], *James Glass*[1]

MIT Computer Science and Artificial Intelligence Laboratory (CSAIL), Cambridge, MA, USA[1]
Qatar Computing Research Institute, HBKU, Doha, Qatar[2]

ABSTRACT

In order to successfully annotate the Arabic speech content found in open-domain media broadcasts, it is essential to be able to process a diverse set of Arabic dialects. For the 2017 Multi-Genre Broadcast challenge (MGB-3) there were two possible tasks: Arabic speech recognition, and Arabic Dialect Identification (ADI). In this paper, we describe our efforts to create an ADI system for the MGB-3 challenge, with the goal of distinguishing amongst four major Arabic dialects, as well as Modern Standard Arabic. Our research focused on dialect variability and domain mismatches between the training and test domain. In order to achieve a robust ADI system, we explored both Siamese neural network models to learn similarity and dissimilarities among Arabic dialects, as well as i-vector post-processing to adapt domain mismatches. Both Acoustic and linguistic features were used for the final MGB-3 submissions, with the best primary system achieving 75% accuracy on the official 10hr test set.

Index Terms— Dialect Recognition, Arabic, MGB challenge, Siamese Network, Domain Adaptation

1. INTRODUCTION

One of the challenges of processing real-world spoken content, such as media broadcasts, is the potential presence of different dialects of a language in the material. Dialect identification can be a useful capability to identify which dialect is being spoken during a recording. Dialect identification can be regarded as a special case of language recognition, requiring an ability to discriminate between different members within the same language family, as opposed to across language families (i.e., for language recognition). The dominant approach, based on i-vector extraction, has proven to be very effective for both language and speaker recognition [1]. Recently, phonetically aware deep neural models have also been found to be effective in combination with i-vectors [2, 3, 4]. Phonetically aware models could be beneficial for dialect identification, since they provide a mechanism to focus attention on small phonetic differences between dialects with predominantly common phonetic inventories.

Since 2015, the Arabic Multi-Genre Broadcast (MGB) Challenge tasks have provided a valuable resource for researchers interested in processing multi-dialectal Arabic speech. For the ASRU 2017 MGB-3 Challenge, there were two possible tasks. The first task was aimed at developing an automatic speech recognition system for Arabic dialectal speech based on a multi-genre broadcast audio dataset. The second task was aimed at developing an Arabic Dialect Identification (ADI) capability for five major Arabic dialects. This paper reports our experimentation efforts for the ADI task.

While the MGB-3 Arabic ASR task included seven different genres from the broadcast domain, the ADI task focused solely on broadcast news. Participants were provided high-quality Aljazeera news broadcasts as well as transcriptions generated by a multi-dialect ASR system created from the MGB-2 dataset [5]. The biggest difference from previous MGB challenges is that only a relatively small development set of in-domain data is provided for adaptation to the test set (i.e., the training data is mismatched with the test data). For the ADI baseline, participants were also provided with i-vector features from the audio dataset, and lexical features from the transcripts. Evaluation software was shared with all participants using baseline features available via Github[1].

The evaluation scenario for the MGB-3 ADI task can be viewed as channel and domain mismatch because the recording environment of the training data is different from the development and test data. In general, channel or domain mismatch between training and test data can be a significant factor affecting system performance. Differences in channel, genre, language, topic etc. produce shifts in low-dimensional projections of the corresponding speech and ultimately cause performance degradations on evaluation data.

In order to address performance degradation of speaker and language recognition systems due to domain mismatches, researchers have proposed various approaches to compensate for, and to adapt to the mismatch [6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18]. For the MGB-3 ADI task, we utilized the development data to adapt to the test data recording domain, and investigated approaches to improve ADI performance both on the domain mismatched scenario, and the matching scenario,

[1] https://github.com/qcri/dialectID

978-1-5090-4788-8/17/$31.00 ©2017 IEEE

by using a recursive whitening transformation, a weighted dialect i-vector model, and a Siamese Neural Network.

In contrast to the language recognition scenario, where there are different linguistic units across languages, language dialects typically share a common phonetic inventory and written language. Thus, we can potentially use ASR outputs such as phones, characters, and lexicons as features. N-gram histograms of phonemes, characters and lexicons can be used as feature vectors directly, and indeed, a lexicon-based n-gram feature vector was provided for the MGB-3 ADI baseline. The linguistic feature space is, naturally, completely different to the audio feature space, so a fusion of the results from both feature representations has been previously shown to be beneficial [19, 20, 21, 22, 23]. Moreover, the linguistic feature has an advantage in channel domain mismatch situations because the transcription itself does not reflect the recording environment, and only contains linguistic information.[2]

In this paper, we describe our work for the MGB-3 ADI Challenge. The final MIT-QCRI submitted system is a combination of audio and linguistic feature-based systems, and includes multiple approaches to address the challenging mismatched conditions. From the official results, this system achieved the best performance among all participants. The following sections describe our research in greater detail.

2. MGB-3 ARABIC DIALECT IDENTIFICATION

For the MGB-3 ADI task, the challenge organizers provided 13,825 utterances (53.6 hours) for the training (TRN) set, 1,524 utterances (10 hours) for a development (DEV) set, and 1,492 utterances (10.1 hours) for a test (TST) set. Each dataset consisted of five Arabic dialects: Egyptian (EGY), Levantine (LEV), Gulf (GLF), North African (NOR), and Modern Standard Arabic (MSA). Detailed statistics of the ADI dataset can be found in [24]. Table 1 shows some facts about the evaluation conditions and data properties. Note that the development set is relatively small compared to the training set. However, it is matched with the test set channel domain. Thus, the development set provides valuable information to adapt or compensate the channel (recording) domain mismatch between the train and test sets.

Dataset category	Training (TRN)	Development (DEV)	Test (TST)
Size	53.6 hrs	10 hrs	10.1 hrs
Genre	News Broadcasts		
Channel (recording)	Carried out at 16kHz	Downloaded directly from a high-quality video server	
Availability for system development	O	O	X

Table 1. MGB-3 ADI Dataset Properties.

3. DIALECT IDENTIFICATION TASK & SYSTEM

The MGB-3 ADI task asks participants to classify speech as one of five dialects, by specifying one dialect for each audio file for their submission. Performance is evaluated via three indices: overall accuracy, average precision, and average recall for the five dialects.

3.1. Baseline ADI System

The challenge organizers provided features and code for a baseline ADI system. The features consisted of 400 dimensional i-vector features for each audio file (based on bottleneck feature inputs for their frame-level acoustic representation), as well as lexical features using bigrams generated from transcriptions [24]. For baseline dialect identification, a multi-class Support Vector Machine (SVM) was used. The baseline i-vector performance was 57.3%, 60.8%, and 58.0% for accuracy, precision and recall respectively. Lexical features achieved 48.4%, 51.0%, and 49.3%, respectively. While the audio-based features achieved better performance than the lexical features, both systems only obtained approximately 50% accuracy, indicating that this ADI task is difficult, considering that there are only five classes to choose from.

3.2. Siamese Neural Network-based ADI

To further distinguish speech from different Arabic dialects, while making speech from the same dialect more similar, we adopted a Siamese neural network architecture [25] based on an i-vector feature space. The Siamese neural network has two parallel convolutional networks, G_W, that share the same set of weights, W, as shown in Figure 1(a). Let $\omega_1$and ω_2 be a

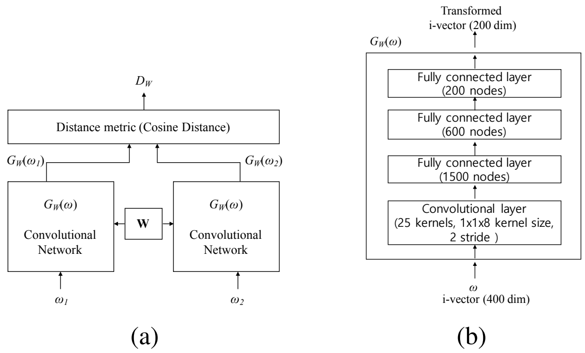
D_W
Distance metric (Cosine Distance)
$G_W(\omega_1)$
$G_W(\omega_2)$
$G_W(\omega)$
Convolutional Network
W
$G_W(\omega)$
Convolutional Network
ω_1
ω_2
(a)
Transformed i-vector (200 dim)
$G_W(\omega)$
Fully connected layer (200 nodes)
Fully connected layer (600 nodes)
Fully connected layer (1500 nodes)
Convolutional layer (25 kernels, 1x1x8 kernel size, 2 stride)
ω
i-vector (400 dim)
(b)

Fig. 1. (a) Siamese network for i-vector (b) Architecture of convolutional neural network G_W

[2] Of course, the word error rate might be higher due to the acoustic mismatch, which could indirectly affect the performance for the linguistic features.

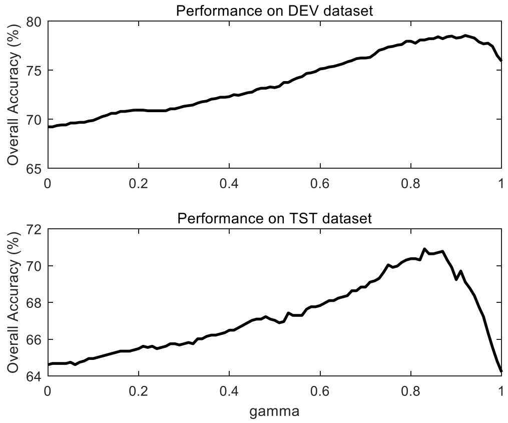

Fig. 2. Overall accuracy on DEV and TST sets by gamma: The DEV set shows the best performance at gamma = 0.91, while the TST set shows the best result at gamma=0.83. For our experiments, we used gamma= 0.91.

pair of i-vectors for which we wish to compute a distance. Let Y be the label for the pair, where Y = 1 if the i-vectors $\omega_1$and ω_2 belong to same dialect, and $Y = -1$ otherwise. To optimize the network, we use a Euclidean distance loss function between the label and the cosine distance, D_W, where

$$L(\omega_i, \omega_j, Y_{ij} = ||Y_{ij} - D_W(\omega_i, \omega_j)||_2^2$$

For training, i-vector pairs and their corresponding labels can be processed by combinations of i-vectors from the training dataset. The trained convolutional network G_W transforms an i-vector ω to a low-dimensional subspace that is more robust for distinguishing dialects. A detailed illustration of the convolutional network G_W is shown in Figure 1(b). The final transformed i-vector, $G_W(\omega)$, is a 200-dimensional vector. No nonlinear activation function was used on the fully connected layer. A cosine distance is used for scoring.

3.3. i-vector Post-Processing

In this section we describe the domain adaptation techniques we investigated using the development set to help adapt our models to the test set.

3.3.1. Interpolated i-vector Dialect Model

Although the baseline system used an SVM classifier, Cosine Distance Scoring (CDS) is a fast, simple, and effective method to measure the similarity between an enrolled i-vector dialect model, and a test utterance i-vector. Under CDS, ZT-norm or S-norm can be also applied for score normalization [26]. Dialect enrollment can be obtained by means of i-vectors for each dialect, and is called the i-vector dialect model: $\overline{\omega_d} = (1/n_d)\sum_{i=1}^{n_d}\omega_i^d$, where n_d is the number of utterances for each dialect d. Since we have two datasets for dialect enrollment, $\overline{\omega_d^{\mathrm{TRN}}}$ for the training set, and $\overline{\omega_d^{\mathrm{DEV}}}$ for the development set, we use an interpolation approach with parameter γ, where

$$\overline{\omega_d^{\mathrm{Inter}}} = (1-\gamma)\overline{\omega_d^{\mathrm{TRN}}} + \gamma\overline{\omega_d^{\mathrm{DEV}}}$$

We observed that the mismatched training set is useful when combined with matched development set. Figure 2 shows the performance evaluation by parameter γ on the same experimental conditions of System 2 in Section 4.3. This approach can be thought of as exactly the same as score fusion for different system. However, score fusion is usually performed at the system score level, while this approach uses a combination of knowledge of in-domain and out-of-domain i-vectors with a gamma weight on a single system.

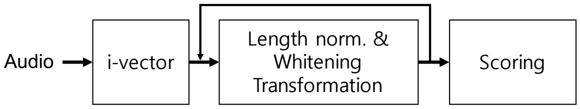

Fig. 3. Flowchart of recursive whitening transformation.

3.3.2. Recursive Whitening Transformation

For i-vector-based speaker and language recognition approaches, a whitening transformation and length normalization is considered essential [27]. Since length normalization is inherently a nonlinear, non-whitening operation, recently, a recursive whitening transformation has been proposed to reduce residual un-whitened components in the i-vector space, as illustrated in Figure 3 [15]. In this approach, the data subset that best matches the test data is used at each iteration to calculate the whitening transformation. In our ADI experiments, we applied 1 to 3 levels of recursive whitening transformation using the training and development data.

3.4. Phoneme Features

Phoneme feature extraction consists of extracting the phone sequence, and phone duration statistics using four different speech recognizers: Czech, Hungarian, and Russian using narrowband model, and English using a broadband model [28]. We evaluated the four systems using a Support Vector Machine (SVM). The hyper-parameters for the SVM are distance from the hyperplane (*C* is 0.01), and penalty *l2*. We used the training data for training the SVM and the development data for testing. Table 2 shows the results for the four phoneme recognizers. The Hungarian phoneme recognition obtained the best results, so we used it for the final system combination.

System	Accuracy(%)	Precision(%)	Recall(%)
Czech	45	45.2	45.8
Hungarian	**47**	**47.3**	**48.1**
Russian	46	47	46.8
English	33.3	33	34

Table 2. Evaluating four phoneme recognition systems.

System (scoring method)	Accuracy (%)	Precision (%)	Recall (%)
i-vector (SVM) - baseline	57.20	60.80	58.00
i-vector (CDS)	56.36	59.86	57.19
LDA i-vector (SVM)	60.17	60.65	60.91
LDA i-vector (CDS)	58.46	61.31	59.14
Siam i-vector (SVM)	61.15	62.91	61.56
Siam i-vector (CDS)	**63.65**	**64.00**	**63.88**

Table 3. i-vector evaluation on DEV set: only TRN set is used for training. Note that scores in this table were not calibrated.

3.5. Character Features

Word sequences are extracted using a state-of-the-art Arabic speech-to-text transcription system built as part of the MGB-2 [29]. 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 rescoring. Our system uses a grapheme lexicon during both training and decoding. The acoustic models are trained on 1,200 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 [5]. We kept the <UNK> from the ASR system, which indicates out-of-vocabulary (OOV) words, we replaced it with special symbol. Space was inserted between all characters including the word boundaries. An SVM classifier was trained similarly to the one used for the phoneme ASR systems, and we achieved 52% accuracy, 51.2% precision and 51.8% recall. The confusion matrix is different between the phoneme classifier and the character classifier systems, which motivates us to use both of them in the final system combination.

3.6. Score Calibration

All scores are calibrated to be between 0 and 1. A linear calibration is done by the Bosaris toolkit [30]. Fusion is also done in a linear manner.

4. ADI EXPERIMENTS

For experiments and evaluation, we use i-vectors and transcriptions that are provided by the challenge organizers. Please refer to [24] for descriptions of i-vector extraction and Arabic speech-to-text configuration.

System (scoring method)	Accuracy (%)	Precision (%)	Recall (%)
Baseline word(SVM)	48.43	50.99	49.25
Character	**57.28**	**60.83**	**58.03**
Phoneme	47.18	47.66	48.23

Table 4. Linguistic feature evaluation on DEV set: TRN and DEV sets were used for training.

4.1. Using Training Data for Training

The first experiment we conducted used only the training data for developing the ADI system. Thus, the interpolated i-vector dialect model cannot be used for this experimental condition. Table 3 shows the performance on dimension reduced i-vectors using the Siamese network (Siam i-vector), and Linear Discriminant Analysis (LDA i-vector), as compared to the baseline i-vector system. LDA reduces the 400-dimension i-vector to 4, while the Siamese network reduces it from 400 to 200. Since the Siamese network used a cosine distance for the loss function, the Siam i-vector showed better performance with the CDS scoring method, while others achieved better performance with an SVM. The best system using Siam i-vector showed overall 10% better performance accuracy, as compared to the baseline.

4.2. Using Training and Development Data for Training

For our second experiment, both the training and development data were used for training. For phoneme and character features, we show development set experimental results in Table 4. For i-vector experiments, we show results in Table 5. In the table we see that the interpolated dialect model gave significant improvements in all three metrics. The recursive whitening transformation gave slight improvements on the original i-vector, but not after LDA and the Siamese network. The best system is the original i-vector with recursive whitening, and an interpolated i-vector dialect model, which achieves over 20% accuracy improvement over the baseline.

While the Siamese i-vector network helped in the training data only experiments, it does not show any advantage over the baseline i-vector for this condition. We suspect this result is due to the composition of the data used for training the Siamese network. To train the network, i-vector pairs are chosen from from training dataset. We selected the pairs using both the training and development datasets. However, if we could put more emphasis on the development data, we suspect the Siamese i-vector network would be more robust on the test data. We plan to further examine the performances due to different compositions of data in the future.

4.3. Performance Evaluation of Submission

Tables 6 and 7 show detailed performance evaluations of our three submitted systems. System 1 was trained using only the training data as shown in Table 6. Systems 2 and 3 were

System (scoring method)	Whitening	i-vector Dialect Model	Accuracy (%)	Precision (%)	Recall (%)
i-vector (SVM)	-	-	64.79	65.40	65.25
i-vector (CDS)	Single	Averaged	68.11	68.56	68.30
	Single	Interpolated	75.52	75.87	75.66
	Recursive	Averaged	69.23	69.61	69.38
	Recursive	Interpolated	**78.54**	**78.75**	**78.70**
LDA i-vector (CDS)	Single	Averaged	69.16	69.50	69.35
	Single	Interpolated	**69.62**	**69.74**	**69.82**
	Recursive	Averaged	68.11	68.06	68.44
	Recursive	Interpolated	68.64	68.28	68.96
Siam i-vector (CDS)	Single	Averaged	67.78	37.97	68.33
	Single	Interpolated	76.05	76.15	76.35
	Recursive	Averaged	67.65	68.18	67.86
	Recursive	Interpolated	**76.31**	**76.39**	**76.60**

Table 5. i-vector evaluation on DEV set: both TRN and DEV sets were used for training.

System - Only TRN is used (scoring method)		DEV			TST		
		Accuracy (%)	Precision (%)	Recall (%)	Accuracy (%)	Precision (%)	Recall(%)
i-vector (SVM) - baseline		57.28	60.83	58.03	55.29	59.27	56.44
System 1	Siamese i-vector (CDS)	63.65	64.00	63.88	60.99	60.88	61.72
	+ score calibration w. DEV dataset	64.44	64.76	64.70	60.92	60.80	61.62
	+ fusion w. char/phone feature	**66.60**	**66.49**	**66.86**	**67.76**	**68.00**	**67.88**

Table 6. Detailed performance evaluation of submitted system: only TRN dataset was used for training.

System - TRN+DEV are used (scoring method)		DEV			TST		
		Accuracy (%)	Precision (%)	Recall (%)	Accuracy (%)	Precision (%)	Recall(%)
i-vector (SVM) - baseline		64.79	65.40	65.25	65.82	65.80	66.35
System 2	i-vector (CDS)	62.07	62.51	62.63	60.86	61.87	61.49
	+ 1st recursive whitening	68.11	68.56	68.30	63.61	64.00	64.27
	+ interpolated i-vector dialect model	75.52	75.98	75.66	68.23	68.95	68.56
	+ 2nd recursive whitening	77.89	78.15	77.98	69.91	70.28	70.24
	+ 3rd recursive whitening	78.54	78.75	78.70	69.97	70.37	70.37
	+ fusion w. char/phone feature	**76.38**	**76.33**	**76.70**	**75.00**	**75.46**	**75.03**
System 3	Siamese i-vector (CDS)	65.81	66.22	66.19	62.47	62.28	63.32
	+ 1st recursive whitening	67.78	68.33	67.97	63.54	63.53	64.22
	+ interpolated i-vector dialect model	76.05	76.15	76.35	68.23	68.75	68.63
	+ 2nd recursive whitening	76.18	76.26	76.49	68.30	68.81	68.69
	+ 3rd recursive whitening	76.31	76.39	76.60	68.30	68.81	68.69
	+ fusion w. char/phone feature	**73.43**	**73.18**	**73.76**	**72.72**	**73.02**	**72.99**

Table 7. Performance of MGB-3 ADI systems: TRN and DEV sets used for training. All scores calibrated with DEV set.

trained using both the training and development sets as shown in Table 7. We found the best linear fusion weight based on System 1 to prevent over-fitting was 0.7, 0.2 and 0.1 for i-vector, character, and phonetic based scores respectively. We applied the same weights to Systems 2 and 3 for fusion.

From Table 6, we see that the Siamese network demonstrates its effectiveness on both the development and test sets without using any information of the test domain. The interpolated i-vector dialect model also demonstrates that it reflects test domain information well as shown by Systems 2 and 3 in Table 7. As we expected that the linguistic features would not affected by the domain mismatch, character and phoneme features show useful contributions for all systems. We believe the reason for the performance degradation of Systems 2 and 3 after fusion on the development data can be seen in the fusion rule. We applied the fusion rule derived from System 1 which was not optimal for Systems 2 and 3, considering the development set evaluation. By including the development data as part of their training, Systems 2 and 3 are subsequently overfit on the development data, which was why we used the fusion rule of System 1. From the excellent fusion performance on the test data for Systems 2 and 3, we believe that the fusion rule from System 1 prevented an over-fitted result.

5. CONCLUSION

In this paper, we describe the MIT-QCRI ADI system using both audio and linguistic features for the MGB-3 challenge. We studied several approaches to address dialect variability and domain mismatches between the training and test sets. Without knowledge of the test domain where the system will be applied, i-vector dimensionality reduction using a Siamese network was found to be useful, while an interpolated i-vector dialect model showed effectiveness with relatively small amounts of test domain information from the development data. On both conditions, fusion of audio and linguistic feature guarantees substantial improvements on dialect identification. As these approaches are not limited to dialect identification, we plan to explore their utility on other speaker and language recognition problems in the future.

6. REFERENCES

[1] Najim Dehak, Patrick J Kenny, Reda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-End Factor Analysis for Speaker Verification," *IEEE Trans. on Audio, Speech, and Lang. Process.*, vol. 19, no. 4, pp. 788–798, may 2011.

[2] Fred Richardson, Douglas Reynolds, and Najim Dehak, "A Unified Deep Neural Network for Speaker and Language Recognition," in *Interspeech*, 2015, pp. 1146–1150.

[3] Yun Lei, Nicolas Scheffer, Luciana Ferrer, and Mitchell McLaren, "A Novel Scheme for Speaker Recognition using a Phonetically-aware Deep Neural Network," in *IEEE ICASSP*, 2014, pp. 1714–1718.

[4] David Snyder, Daniel Garcia-Romero, and Daniel Povey, "Time delay deep neural network-based universal background models for speaker recognition," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2016, pp. 92–97.

[5] Sameer Khurana and Ahmed Ali, "QCRI advanced transcription system (QATS) for the Arabic Multi-Dialect Broadcast media recognition: MGB-2 challenge," in *IEEE Workshop on Spoken Language Technology(SLT)*, 2016, pp. 292–298.

[6] Stephen Shum, Douglas a. Reynolds, Daniel Garcia-Romero, and Alan McCree, "Unsupervised Clustering Approaches for Domain Adaptation in Speaker Recognition Systems," in *Proceedings of Odyssey - The Speaker and Language Recognition Workshop*, 2014, pp. 265–272.

[7] Md Hafizur Rahman, Ahilan Kanagasundaram, David Dean, and Sridha Sridharan, "Dataset-invariant covariance normalization for out-domain PLDA speaker verification," in *Interspeech*, 2015, pp. 1017–1021.

[8] Elliot Singer and Douglas A. Reynolds, "Domain Mismatch Compensation for Speaker Recognition Using a Library of Whiteners," *IEEE Signal Processing Letters*, vol. 22, no. 11, pp. 2000–2003, 2015.

[9] Daniel Garcia-Romero, Xiaohui Zhang, Alan McCree, and Daniel Povey, "Improving Speaker Recognition Performance in the Domain Adaptation Challenge Using Deep Neural Networks," in *IEEE Spoken Language Technology Workshop (SLT)*, 2014, pp. 378–383.

[10] Daniel Garcia-Romero and Alan McCree, "Supervised domain adaptation for I-vector based speaker recognition," in *IEEE ICASSP*, 2014, pp. 4047–4051.

[11] Daniel Garcia-Romero, Alan McCree, Stephen Shum, Niko Brummer, and Carlos Vaquero, "Unsupervised Domain Adaptation for I-Vector Speaker Recognition," in *Proceedings of Odyssey - The Speaker and Language Recognition Workshop*, 2014, pp. 260–264.

[12] Ondrej Glembek, Jeff Ma, Pavel Matejka, Bing Zhang, Oldrich Plchot, Lukas Burget, and Spyros Matsoukas, "Domain adaptation via within-class covariance correction in i-vector based speaker recognition systems," in *IEEE ICASSP*, 2014, pp. 4060–4064.

[13] Hagai Aronowitz, "Inter dataset variability compensation for speaker recognition," in *IEEE ICASSP*, 2014, pp. 4002–4006.

[14] Suwon Shon, Seongkyu Mun, Wooil Kim, and Hanseok Ko, "Autoencoder based Domain Adaptation for Speaker Recognition under Insufficient Channel Information," in *Interspeech*, 2017, pp. 1014–1018.

[15] Suwon Shon, Seongkyu Mun, and Hanseok Ko, "Recursive Whitening Transformation for Speaker Recognition on Language Mismatched Condition," in *Interspeech*, 2017, pp. 2869–2873.

[16] Suwon Shon and Hanseok Ko, "KU-ISPL Speaker Recognition Systems under Language mismatch condition for NIST 2016 Speaker Recognition Evaluation," *ArXiv e-prints arXiv:1702.00956*, 2017.

[17] Ahilan Kanagasundaram, David Dean, and Sridha Sridharan, "Improving out-domain PLDA speaker verification using unsupervised inter-dataset variability compensation approach," in *IEEE ICASSP*, 2015, pp. 4654–4658.

[18] Hagai Aronowitz, "Compensating Inter-Dataset Variability in PLDA Hyper-Parameters for Robust Speaker Recognition," in *Proceedings of Odyssey - The Speaker*

and Language Recognition Workshop, 2014, pp. 280–286.

[19] Ahmed Ali, Najim Dehak, Patrick Cardinal, Sameer Khurana, Sree Harsha Yella, James Glass, Peter Bell, and Steve Renals, "Automatic dialect detection in Arabic broadcast speech," in *Interspeech*, 2016, vol. 08-12-Sept, pp. 2934–2938.

[20] Abualsoud Hanani, Aziz Qaroush, and West Bank, "Identifying dialects with textual and acoustic cues," in *VarDial*, 2017, pp. 93–101.

[21] Radu Tudor Ionescu and Andrei M Butnaru, "Learning to Identify Arabic and German Dialects using Multiple Kernels," in *VarDial*, 2017, pp. 200–209.

[22] Taraka Rama, "T ubingen system in VarDial 2017 shared task : experiments with language identification and cross-lingual parsing," in *VarDial*, 2017, pp. 146–155.

[23] Shervin Malmasi and Marcos Zampieri, "Arabic Dialect Identification Using iVectors and ASR Transcripts," in *VarDial*, 2017, number 2015, pp. 178–183.

[24] Ahmed Ali, Stephan Vogel, and Steve Renals, "Speech Recognition Challenge in the Wild: ARABIC MGB-3," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2017, p. to be appeared.

[25] Jane Bromley, James W. Bentz, Léon Bottou, Isabelle Guyon, Yann Lecun, Cliff Moore, Eduard Säckinger, and Roopak Shah, "Signature Verification Using a Siamese Time Delay Neural Network," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 07, no. 04, pp. 669–688, 1993.

[26] Stephen Shum, Najim Dehak, Reda Dehak, and James R Glass, "Unsupervised Speaker Adaptation based on the Cosine Similarity for Text-Independent Speaker Verification," in *Proceedings of Odyssey - The Speaker and Language Recognition Workshop*, 2010.

[27] Daniel Garcia-Romero and Carol Y Espy-Wilson, "Analysis of i-vector Length Normalization in Speaker Recognition Systems.," in *Interspeech*, 2011, pp. 249–252.

[28] Petr Schwarz, Pavel Matejka, and Jan Cernocky, "Hierarchical structures of neural networks for phoneme recognition," in *IEEE ICASSP*. IEEE, 2006.

[29] Ahmed Ali, Peter Bell, James Glass, Yacine Messaoui, Hamdy Mubarak, Steve Renals, and Yifan Zhang, "The mgb-2 challenge: Arabic multi-dialect broadcast media recognition," in *IEEE Spoken Language Technology Workshop (SLT)*, 2016.

[30] Niko Brümmer and Edward de Villiers, "The BOSARIS Toolkit: Theory, Algorithms and Code for Surviving the New DCF," in *NIST SRE'11 Analysis Workshop*, apr 2011.