DOMAIN ATTENTIVE FUSION FOR END-TO-END DIALECT IDENTIFICATION WITH UNKNOWN TARGET DOMAIN

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

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

ABSTRACT

End-to-end deep learning language or dialect identification systems operate on the spectrogram or other acoustic feature and directly generate identification scores for each class. An important issue for end-to-end systems is to have some knowledge of the application domain, because the system can be vulnerable to use cases that were not seen in the training phase; such a scenario is often referred to as a domain mismatched condition. In general, we assume that there is enough variation in the training dataset to expose the system to multiple domains. In this work, we study how to best make use a training dataset in order to have maximum effectiveness on unknown target domains. Our goal is to process the input without any knowledge of the target domain while preserving robust performance on other domains as well. To accomplish this objective, we propose a domain attentive fusion approach for end-to-end dialect/language identification systems. To help with experimentation, we collect a dataset from three different domains, and create experimental protocols for a domain mismatched condition. The results of our proposed approach, which were tested on a variety of broadcast and YouTube data, shows significant performance gain compared to traditional approaches, even without any prior target domain information.

Index Terms— Dialect identification, language identification, self-attention, fusion

1. INTRODUCTION

Channel or domain mismatch between training and test data can be a significant factor affecting performance for language and dialect identification (DID) systems, but mismatch has not been addressed as seriously for these tasks as it has been in the speaker recognition arena. In 2013, a domain adaptation challenge (DAC13) was held on domain mismatch for speaker recognition [1]. From the success of DAC13, many researchers explored the domain mismatch problem on the speaker recognition task [2, 3, 4, 5]. However, the same mismatch issue for language/dialect recognition was not actively studied until the NIST 2017 Language Recognition Evaluation (LRE) [6] provided speech datasets from multiple domains. At both challenges, many studies tried to adapt the Gaussian Back-end or PLDA back-end on top of the i-vector or x-vector speaker embeddings [7, 8, 2, 9, 3, 4]. Although these approaches cannot be directly applied to end-to-end deep learning systems for these same tasks, they achieved reasonable performance when the target speech domain was known *a priori*.

The Multi-Genre Broadcast 3 (MGB-3) challenge also provided domain mismatched data for dialect identification. For MGB-3, unsupervised learning of dialectal speech was investigated by Zhang [10] and Shon [11, 12] to extract domain invariant features. By exploiting speech data from several domains without explicit language and domain labels, the networks could extract domain invariant representations from input speech. The approaches still needed some amount of labeled data to train subsequent identification systems. They achieved large performance gains when there were no language labels on the target domain training dataset compared to traditional acoustic features like MFCCs. Although the performance gap closed when enough labeled target domain data were available, they have an advantage for scenarios where large amounts of unannotated speech is available [11].

In this research, we do not assume any resource limitation or challenging situations like unlabeled target domain data. Instead we assume that we have enough data from multiple domains with labels for dialect identification. However, we also assume that we don't have any domain information about the target speech. In this case, a training model with labeled multiple domain data would easily provide superior performance over the previous efforts which adapt the back-end scoring to a target domain. Another possible approach is that score-level fusion of subsystems which are trained on single domain data. In the periodic series of NIST evaluations, it was observed that linear fusion of multiple subsystems consistently outperforms the single best system [13]. However, the performance of the fusion system depends strongly on the logistic regression fusion, whose parameters need to be calibrated to specific trials which reflect the test conditions. Thus, the system fusion was optimized to the specific domain of the test trials, so that if the test speech came from a random domain, the fusion system cannot guarantee the best performance.

To address the unknown domain speech input, we propose to use a self-attention layer in our end-to-end model and have fusion parameters which are calculated from the input speech. Once the domain attentive layer is trained using the training data, it automatically generates the best fusion weight of domain-specific systems by taking the output of each subsystem. Thus, ideally, the optimal fusion weight would be generated for every single input.

In the following sections, we examine baseline systems for unknown domain inputs and propose domain attentive layers. We also describe our data collection from YouTube, called Varieties and Dialects (VarDial) 2018, to provide a dataset for our experiments.

2. DIALECTAL LANGUAGE DATASET

For this work, we used the two dialect datasets called MGB-3 and VarDial 2018, to generate domain mismatched conditions for our experiments. As shown in Table 1, the MGB-3 data consists of recorded and high-quality broadcasts, while the VarDial data consists of YouTube videos. Each dataset contains data that has

Data name	MGB-3						VarDial 2018			
Type	Training		Development		Testing		Training		Testing	
Domain	Recorded Broadcast		High-quality Broadcast				YouTube			
Dialect	Ex.	Dur.	Ex.	Dur.	Ex.	Dur.	Ex.	Dur.	Ex.	Dur.
EGY	3,093	12.4	298	2.0	302	2.0	93,408	206.3	1,143	5.5
GLF	2,744	10.0	264	2.0	250	2.1	92,603	204.5	1,147	5.6
LEV	2,851	10.3	330	2.0	334	2.0	232,585	513.6	1,131	5.5
MSA	2,183	10.4	281	2.0	262	1.9	9,518	21.0	944	4.6
NOR	2,954	10.5	351	2.0	344	2.1	24,841	54.9	980	4.8
Total	13,825	53.6	1,524	10.0	1,492	10.1	452,955	1000.3	5,345	26.0

Table 1: Arabic dialect data breakdown for the MGB-3 and VarDial 2018 datasets.

been labeled from five Arabic dialects: Egyptian (EGY), Levantine (LEV), Gulf (GLF), North African (NOR), and Modern Standard Arabic (MSA). The MGB-3 data collection was distributed equally across all five dialects for both training/dev/test partitions, while the VarDial data has significantly more data, though more unevenly spread across dialects, due to the manner in which it was collected. Although the MGB-3 development set is relatively small compared to the training set, it matches the test set channel conditions, and thus provides valuable information about the test domain. More details about MGB-3 are available in [14].

The VarDial 2018 dataset was collected from YouTube in a semi-supervised technique. Initially, we identified more than 30 YouTube channels. The dialect for each channel is known. However, we are unable to guarantee that there is no cross-dialectal speech in the channels. For every channel, we crawled more than 100 video clips. For each video, we ran voice activation detection [15] and the data was sliced into small audio clips between 5 and 30 seconds. Furthermore, the test set was manually verified and was uniformly distributed. The accuracy of verifying the test set and the non-speech clips were about 92%. More details about VarDial 2018 are available in [16].

3. BASELINE SYSTEM DID EXPERIMENTS

3.1. End-to-end dialect identification system

In this work, we adopt the end-to-end dialect identification system proposed in [17]. This system has a stack of convolutional neural network (CNN) layers, followed by a global pooling layer that aggregates frame level representations to produce utterance level representations. The output of the global pooling layer is followed by two feed forward (FF) layers. Specifically, the network consists of four one-dimensional CNN layers (40×5 - 500×7 - 500×1 - 500×1 filter sizes; with 1-2-1-1 strides; the number of filters is 500-500-500-3000) and two FF layers (1500-600). The size of the final softmax layer is determined by the task-specific language or dialect labels and the softmax output can be used directly as a score for each dialect class for the DID task. We used MFCCs as inputs to the end-to-end system since they obtained the best performance without any dataset augmentation. Note that no dataset augmentation was performed for these experiments.

3.2. Training on multiple domains

Consider datasets S_1 and S_2 with two unknown data distributions $\mathcal{D}_1$ and $\mathcal{D}_2$. If the target domain is the same as S_1, we can discard S_2 and use only S_1 to train a network. To cope with multiple domains, multiple networks could be learned using each domain dataset. In this case, we need N networks for N domain target domains.

Score-level fusion of N single-domain networks can boost performance. Linear logistic regression based fusion is a common method for learning an optimal linear combination of the multiple systems. However, this approach relies on target domain sample trials to estimate the regression parameters, and is vulnerable if the trial domain is mismatched with the target domain.

For efficient multi-domain learning, we can also use multiple domain datasets to learn a single network. Parameter sharing during training can be full or partial [18, 19]. Since the task of each domain is the same,i.e. Arabic dialect identification, we will share all parameters for multi-domain learning with a single network. One of the advantages of multi-domain learning is that the input domain information is not needed whereas the single domain trained network needs domain information for maximum performance.

Table 2 shows DID accuracy on the MGB-3 and VarDial 2018 Test sets when using different domain training datasets. While systems trained using a single domain dataset such as $\mathcal{A}$ and $\mathcal{B}$ show robust performance only on the matched domain test set, system $\mathcal{Z}$ performs efficiently on both test set. Note that we doubled the number of filters in the neural network structure for system $\mathcal{Z}$ to match the network capacity.

Training data	System ID	DID Accuracy (%)	
		MGB-3 Test	VarDial 2018 Test
MGB-3 Train + MGB-3 Dev	$\mathcal{A}$	**65.82**	48.87
YouTube Train	$\mathcal{B}$	51.27	**86.40**
MGB-3 Train + MGB-3 Dev + YouTube Train	$\mathcal{A}+\mathcal{B}$	61.86	81.53
Fusion of $\mathcal{A}$ and $\mathcal{B}$ (optimized for $\mathcal{A}$)	-	**68.63**	77.57
Fusion of $\mathcal{A}$ and $\mathcal{B}$ (optimized for $\mathcal{B}$)	-	57.84	**86.94**

Table 2: Baseline dialect identification performance evaluation.

Score level fusion can be applied by logistic regression for maximum efficiency on multiple domains. The trials for optimizing parameters of logistic regression were generated by randomly combining utterances using the target domain training set. The fusion approach achieves the best performance when the fusion rule was optimized for the target domain and achieves the worst performance on the test from another domain. Thus, the fusion approach is not practical if the system has no information about the domain. Although there is some performance degradation, the multi-domain trained system $\mathcal{Z}$ generally works on both domains.

4. DOMAIN ATTENTIVE FUSION

4.1. Fusion layer for system combination

Traditional logistic regression for score-level fusion can be replaced by a neural network by adding a fusion layer on top of the single

domain trained networks as shown in figure 1. We used a fully connected layer with 600 hidden nodes and added a softmax layer which generates a score for each dialect. The fusion network was trained after the other networks were trained and fixed. By learning from the training dataset, the fusion network dynamically selects the most useful scores which are invariant to domain mismatch.

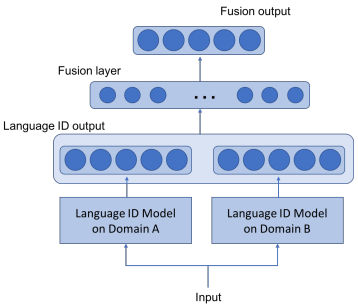
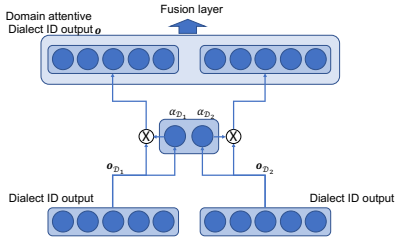


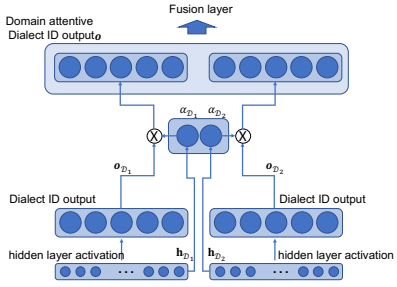
Fig. 1: Fully-connected layer for score-level fusion.

4.2. Self-attention based weighting

The neural network attention mechanism is a powerful technique to focus on the significant or critical part of an input signal. An attention layer enables to focus on the important information of the input sequence by providing more weight on it. Thus, in speech processing, it usually applied to the frame-level neural network layer to represent long sequence more effectively. In this research, we used an attention layer to adapt the domain by using multiple networks which were trained on different domains, as shown in Figure 2(a).



(a) Self-attention layer



(b) Self-attention layer using embeddings

Fig. 2: Domain attentive fusion

Suppose the L-dimensional vector output $\mathbf{o}_d$ of an end-to-end system trained using dataset with distribution $d \in \{\mathcal{D}_1, \mathcal{D}_2\}$ where L is total number of dialects to be identified. We could learn a scalar score $e_d \in \mathbb{R}$ for output $\mathbf{o}_d$ as

$$e_d = f(\mathbf{o}_d). \tag{1}$$

The scoring function $f(\cdot)$ can be calculated as

$$f(\mathbf{o}_d) = \mathbf{v}_d^T \tanh(\mathbf{W}_d \mathbf{o}_d + \mathbf{b}_d) \tag{2}$$

where $\mathbf{W}_d$ is an m by L matrix and $\mathbf{b}_d$ and $\mathbf{v}_d$ are m-dimensional vectors. m is a hyper-parameter that can be tuned. The normalized weights α_d can be computed as

$$\alpha_d = \frac{exp(e_d)}{(exp(e_{\mathcal{D}_1}) + exp(e_{\mathcal{D}_2}))} \tag{3}$$

so, $\alpha_{\mathcal{D}_1} + \alpha_{\mathcal{D}_2}$ is equal to 1. Finally, we obtain the domain attentive output as

$$\mathbf{o} = [\alpha_{\mathcal{D}_1} * \mathbf{o}_{\mathcal{D}_1}, \alpha_{\mathcal{D}_2} * \mathbf{o}_{\mathcal{D}_2}] \tag{4}$$

Since domain related information is more likely remain in the intermediate layer than in the output layer, we can also incorporate hidden layer activations into the attention layer because they can be contain complimentary information when we calculate the end-to-end system output. In this case, we use the hidden layer activations $\mathbf{h}_d$ as the input for the scoring function, and the scalar score can be calculated as $e_d = f(\mathbf{h}_d)$. This approach is depicted in Figure 2 (b).

5. DIALECT IDENTIFICATION EXPERIMENTS

For the end-to-end DID systems, we used MFCC features. To extract the features, a spectrogram was computed using a 400 sample FFT window length with 160 sample advance which is equivalent to 25ms window and 10ms frame-rate for 16kHz audio. A total of 40 coefficients were extracted and then normalized to have zero mean and unit variance. The end-to-end structure is the same as in [17], with four CNN and two FF layers as described in Section 3. The stochastic gradient descent (SGD) learning rate was 0.001 with a decay every 50,000 mini-batches with a factor of 0.98. Rectified Linear Units (ReLUs) were used for activation nonlinearities. For the attention layer, we set m as 10.

Performance was measured in accuracy, Equal Error Rate (EER) and minimum decision cost function C_{avg}*100. Accuracy was measured by choosing the dialect with the maximum score for each test utterance. Minimum C_{avg} *100 was computed from hard decision errors and a fixed set of costs and priors from [20].

Depending on the experimental condition, we used different datasets for training the network. Since we have three domains for training and two domains for testing from the MGB-3 and VarDial 2018 datasets, we could partition our experimental conditions into two categories, "seen" and "unseen" test domains. For the seen test condition, we used a training dataset which is matched to the test domain, so that all test domains are already seen when the network is learning. For the unseen test condition, we excluded the training dataset which matched the test domain, so that the network could not learn about the test domain dataset distribution.

5.1. Seen domain test condition results

Table 3 shows the "seen" condition experimental results whereby both the MGB-3 and VarDial 2018 domains could be learned in the training process. From the results, we observe that training individual networks for each domain and fusing the results yields better performance than training multiple domains using a single network. It is interesting that the proposed domain attentive fusion perform

Training data	Test on								
	MGB-3 Test			VarDial 2018 Test			Averaged		
	Acc.	EER	Cavg	Acc.	EER	Cavg	Acc.	EER	Cavg
MGB-3 Train + MGB-3 Dev ($\mathcal{A}$)	65.82	20.43	19.60	48.87	28.39	28.50	58.35	24.41	24.05
YouTube Train ($\mathcal{B}$)	51.27	28.37	27.41	86.40	9.57	9.96	68.84	18.97	18.69
MGB-3 Train + MGB-3 Dev + YouTube Train ($\mathcal{A}$+$\mathcal{B}$)	61.86	22.92	21.41	81.53	11.13	11.76	71.70	17.03	**16.59**
Logistic regression fusion of $\mathcal{A}$ and $\mathcal{B}$ (optimized for $\mathcal{A}$)	**68.63**	**19.05**	**18.04**	77.57	13.78	14.16	**73.10**	**16.42**	16.10
Logistic regression fusion of $\mathcal{A}$ and $\mathcal{B}$ (optimized for $\mathcal{B}$)	57.84	24.36	23.35	**86.94**	**9.23**	**9.56**	72.39	16.80	16.46
Using fusion layer on $\mathcal{A}$ and $\mathcal{B}$ (Figure 1)	67.69	19.30	18.39	82.86	11.19	11.58	75.28	15.25	14.99
Domain Attentive fusion of $\mathcal{A}$ and $\mathcal{B}$ (Figure 2 (a))	67.49	18.52	18.01	83.93	10.03	10.22	75.71	14.28	14.12
Domain Attentive fusion of $\mathcal{A}$ and $\mathcal{B}$ (Figure 2 (b))	**68.23**	**18.30**	**17.69**	**85.01**	**9.13**	**9.40**	**76.62**	**13.72**	**13.55**

Table 3: Dialect identification performance for the "Seen" test domain condition.

Training data	Test on								
	MGB-3 Test (Unseen)			VarDial 2018 Test (Seen)			Averaged		
	Acc.	EER	Cavg	Acc.	EER	Cavg	Acc.	EER	Cavg
MGB-3 Train ($\mathcal{C}$)	48.79	31.80	30.74	41.14	34.70	34.27	44.97	33.25	32.51
YouTube Train ($\mathcal{B}$)	51.27	28.37	27.41	86.40	9.57	9.96	68.84	18.97	18.69
MGB-3 Train + YouTube Train ($\mathcal{B}$+$\mathcal{C}$)	**56.37**	**25.07**	**24.10**	83.85	9.87	10.30	70.11	**17.47**	**17.20**
Logistic regression fusion of $\mathcal{B}$ and $\mathcal{C}$ (optimized for $\mathcal{C}$)	55.29	25.67	24.84	83.26	11.09	11.15	69.28	18.38	18.00
Logistic regression fusion of $\mathcal{B}$ and $\mathcal{C}$ (optimized for $\mathcal{B}$)	54.22	26.69	25.67	**87.56**	**8.96**	**9.36**	**70.89**	17.83	17.52
Using fusion layer on $\mathcal{B}$ and $\mathcal{C}$ (Figure 1)	54.76	26.29	25.48	85.11	9.97	10.28	69.94	18.13	17.88
Domain Attentive fusion of $\mathcal{B}$ and $\mathcal{C}$ (Figure 2 (a))	**55.83**	25.67	24.92	85.63	9.84	9.97	70.73	17.76	17.45
Domain Attentive fusion of $\mathcal{B}$ and $\mathcal{C}$ (Figure 2 (b))	55.76	**25.03**	**24.05**	**86.90**	**8.36**	**8.71**	**71.33**	**16.70**	**16.38**

Table 4: Dialect identification performance for the "Unseen" and "Seen" test domain conditions.

remarkably better on both domains, and even better than logistic regression fusion which is optimized for each domain. Note that the domain attentive fusion approach doesn't need target domain information *a priori* and the attention layer automatically generates the weight of the network for fusion for an arbitrary input.

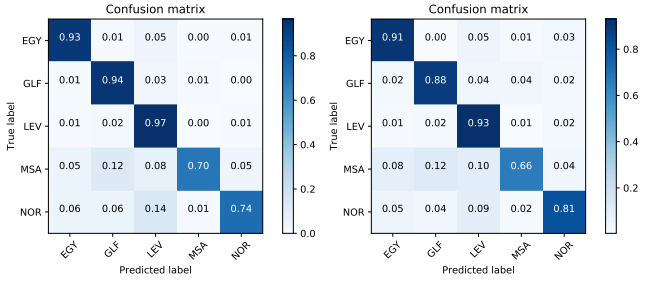
5.2. Unseen domain test condition results

Table 4 shows the "unseen" test condition experimental results whereby only the VarDial2018 domain was learned by the end-to-end network, so that the MGB-3 domain was unseen in the training process. On average, the domain attentive fusion approach shows the best performance among all approaches. Performance improvements on the unseen domain is not as impressive as for the "seen" domain condition. However, we verified that the attention layer still learns about domain information from the end-to-end system output and the hidden layer activations and generates a reasonable result.

6. DISCUSSION

For both "seen" and "unseen" conditions, the domain attentive fusion shows performance improvements compared to conventional approaches. Apart from the performance, the proposed approach has a significant advantage when the input domain is unknown for practical situations. We believe that this approach can be extended to multiple domains and will enable the automatic calculation of the contribution of each sub-system to achieve the best result.

Figure 3 shows a confusion matrix of system $\mathcal{B}$ and the domain attentive fusion system in the last row of Table 3 on the VarDial 2018 test. Since the amount of data for each dialect is not balanced (see table 1, the confusion matrix shows poor performance of MSA and NOR compared to others. We believe that this unbalanced situation always happens in low-resourced languages such as dialects, and we plan to address this issue in the future.



(a) System $\mathcal{B}$ (Accuracy=86.40%)

(b) After domain attentive fusion (Accuracy=85.01%)

Fig. 3: DID confusion matrix on VarDial 2018 Test.

7. CONCLUSION

A neural network-based end-to-end system has achieved the best performance on dialect/language identification tasks. But it remains vulnerable to domain mismatches, especially when the test domain is unknown. To recognize and adapt input domains automatically, we propose a domain attentive layer for fusion of multiple networks that are trained on a single domain. A domain attentive layer calculates the contribution of each network automatically by using the end-to-end language identification system outputs or hidden layer activations. The proposed approach was shown to be robust on test conditions without any *a priori* target domain knowledge.

For future work, we plan to expand the Arabic dialect identification task from 5 dialects to a larger number of country-specific dialects. We also plan to explore the scalability of the proposed approach to multiple domains, and balance performance using unbalanced datasets.

8. REFERENCES

[1] "JHU 2013 Speaker Recognition Workshop", Available : http://www.clsp.jhu.edu/wp-content/uploads/sites/75/2015/10/WS13-Speaker-DAC.pdf.

[2] 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.

[3] 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.

[4] Jesus Villalba and Eduardo Lleida, "Unsupervised Adaptation of PLDA by Using Variational Bayes Methods," in *IEEE ICASSP*, 2014, pp. 744–748.

[5] 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.

[6] Seyed Omid Sadjadi, Timothee Kheyrkhah, Audrey Tong, Craig Greenberg, Douglas Reynolds, Elliot Singer, Lisa Mason, and Jaime Hernandez-Cordero, "The 2017 nist language recognition evaluation," in *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop*, 2018, pp. 82–89.

[7] Mitchell Mclaren, Mahesh Kumar Nandwana, Diego Castán, and Luciana Ferrer, "Approaches to multi-domain language recognition," in *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop*, 2018, pp. 90–97.

[8] Jesus Antonio Villalba Lopez, Niko Brummer, and Najim Dehak, "End-to-end versus embedding neural networks for language recognition in mismatched conditions," in *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop*, 2018, pp. 112–119.

[9] 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.

[10] Qian Zhang and John HL Hansen, "Language/dialect recognition based on unsupervised deep learning," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 26, no. 5, pp. 873–882, 2018.

[11] Suwon Shon, Wei-Ning Hsu, and James Glass, "Unsupervised Representation Learning of Speech for Dialect Identification," in *IEEE Spoken Language Technology Workshop (SLT)*, 2018.

[12] Suwon Shon, Ahmed Ali, and James Glass, "MIT-QCRI Arabic Dialect Identification System for the 2017 Multi-Genre Broadcast Challenge," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2017, pp. 374–380.

[13] Kong Aik Lee and SRE'16 I4U Group, "The i4u mega fusion and collaboration for nist speaker recognition evaluation 2016," in *Proc. Interspeech 2017*, 2017, pp. 1328–1332.

[14] 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, pp. 316–322.

[15] Sylvain Meignier and Teva Merlin, "Lium spkdiarization: an open source toolkit for diarization," in *CMU SPUD Workshop*, 2010.

[16] Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Ahmed Ali, Suwon Shon, James Glass, Yves Scherrer, Tanja Samardžić, Nikola Ljubešić, and Jörg Tiedemann, "Language Identification and Morphosyntactic Tagging: The Second VarDial Evaluation Campaign," *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pp. 1–17, 2018.

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

[18] Hakan Bilen and Andrea Vedaldi, "Universal representations:The missing link between faces, text, planktons, and cat breeds," *arXiv preprint arXiv:1701.07275*, 2017.

[19] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi, "Efficient parametrization of multi-domain deep neural networks," in *Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 8119–8127.

[20] "The NIST 2015 Language Recognition Evaluation Plan", Available : https://www.nist.gov/sites/default/files/documents/2016/10/06/lre15_evalplan_v23.pdf.