UNSUPERVISED REPRESENTATION LEARNING OF SPEECH FOR DIALECT IDENTIFICATION

Suwon Shon, Wei-Ning Hsu, James Glass

Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139 USA
{swshon,wnhsu,glass}@mit.edu

ABSTRACT

In this paper, we explore the use of a factorized hierarchical variational autoencoder (FHVAE) model to learn an unsupervised latent representation for dialect identification (DID). An FHVAE can learn a latent space that separates the more static attributes within an utterance from the more dynamic attributes by encoding them into two different sets of latent variables. Useful factors for dialect identification, such as phonetic or linguistic content, are encoded by a segmental latent variable, while irrelevant factors that are relatively constant within a sequence, such as a channel or a speaker information, are encoded by a sequential latent variable. The disentanglement property makes the segmental latent variable less susceptible to channel and speaker variation, and thus reduces degradation from channel domain mismatch. We demonstrate that on fully-supervised DID tasks, an end-to-end model trained on the features extracted from the FHVAE model achieves the best performance, compared to the same model trained on conventional acoustic features and an i-vector based system. Moreover, we also show that the proposed approach can leverage a large amount of unlabeled data for FHVAE training to learn domain-invariant features for DID, and significantly improve the performance in a low-resource condition, where the labels for the in-domain data are not available.

Index Terms: language recognition, dialect identification, variational autoencoder, unsupervised learning

1. INTRODUCTION

Over the last few years, combinations of i-vectors and Deep Neural Networks (DNNs) [1, 2, 3] have achieved state-of-the-art results for speaker recognition and language identification (LID). DNN-based end-to-end systems [4, 5, 6, 7] have recently obtained comparable or slightly better performance on these tasks. In comparison to LID, Dialect Identification (DID) is a relatively unexplored task because DID is often regarded as a pecial case of LID. However, DID is, in fact, a much more challenging task compared to LID, due to the high similarity between dialects [8, 9, 10, 11].

In [12], the authors investigated an end-to-end approach to a DID task using an Arabic dialect dataset. The authors applied a Convolutional Neural Network (CNN) and conducted extensive experiments on comparing the use of different acoustic features as well as data augmentation methods, which demonstrates the importance of feature selection and the effectiveness of increasing the amount of data. However, such methods suffer from severe performance degradation under domain mismatch condition, where training and testing data are drawn from different domains. Unfortunately, this is not a rare condition due to the scarcity of such datasets. Thus, learning a better speech representation that is domain invariant becomes essential for DID. In previous studies, domain mismatch was mainly addressed in the context of speaker recognition[13, 14, 15, 16, 17]. Among them, Inter Dataset Variability Compensation (IDVC) is a simple but powerful approach when in-domain data are presented but unlabeled. We will also investigate this approach on dialect identification. Another approach that can be applied to this condition was studied by Zhang and Hansen [18, 19]. They presented general Autoencoder approach and unsupervised bottleneck feature (uBNF) extraction approach to language/dialect identification task. Because both approaches do not need ground truth label, they use a big amount of unlabeled data to compensate domain mismatches. They concluded uBNF achieved the best performance on the tasks. We conducted uBNF extraction on our experimental condition for performance comparison.

In this paper, we present unsupervised representation learning of dialectal speech using a Factorized Hierarchical Variational Autoencoder (FHVAE) [20]. An FHVAE is a variant of variational autoencoders (VAEs) [21], which models a generative process of sequential data with a hierarchical graphical model, and defines a corresponding inference model for variational inference. Similar to VAEs, an FHVAE is trained to maximize a lower bound of the marginal likelihood, and hence does not require any supervision, which enables utilization of an unlabeled in-domain dataset for representation learning. In particular, such a model can represent static and dynamic generating factors within an utterance with a latent sequence variable and latent segment variable respectively. As shown in [20], when an FHVAE model is trained on speech data, channel and speaker-related information will be encoded in the latent sequence variable, while language related information like accent, tone, rhythm are encoded in the latent segment variable. Therefore, if we use the latent segment variable as a new feature for dialect identification, the system could be more robust to a domain and speaker variability. We incorporated this latent segment variable into our end-to-end dialect identification system.

We conducted experiments on two different conditions where labels of in-domain data are only available in one setting and the other is not. I-vector and end-to-end systems trained on traditional acoustic features also served as a baseline result. The results indicate that using the latent segment representation from the FHVAE model to train an end-to-end DID system outperforms all other baselines and suffers less degradation when a labeled in-domain dataset is not available.

2. DIALECT/LANGUAGE IDENTIFICATION SYSTEM

2.1. I-vectors and Bottleneck Features

The i-vector has been regarded as the state-of-the-art representation for speaker and language identification, especially when combined with a DNN for i-vector extraction. The most successful approach uses a DNN to generate bottleneck (BN) features from an acoustic model [1, 22, 2, 9, 8]. The acoustic model can be trained using DNNs with state-level alignment information, where the DNN has one relatively constricted layer with a small number of hidden units (i.e., a "bottleneck"). The bottleneck layer activation can be used as a new acoustic feature for training the i-vector extractor, or it can be fed into a second stage DNN acoustic model to produce a Stacked Bottleneck (SBN) feature [22, 23]. Although the acoustic model is often trained on monolingual data (e.g., English), bottleneck features perform reasonably well on LID tasks [22]. For DID, since the dialects are from the same language family, using an SBN that is extracted from a subset of the dialects can achieve excellent performance [24]. Training an i-vector extractor is exactly the same for a speaker recognition or an LID system. The only difference is learning the subsequent projections.

2.2. End-to-End CNN/DNN System

Recently, end-to-end approaches have achieved impressive performance compare to conventional i-vector approach for both LID [4, 5, 12, 25] and speaker recognition [7, 6, 26]. In [12], the authors conducted detailed experiments on an end-to-end system using a dataset augmentation approach with acoustic features ranging from Mel-Frequency Cepstral Coefficients (MFCCs) to spectrograms. The end-to-end system performed significantly better than i-vectors if the training dataset had a large amount of diversity.

In this work, we adopt the end-to-end system proposed in [12]. This system has a stack of CNN layers, followed by a global pooling layer that aggregates frame level representations to produce to utterance level representations. The output of global pooling layer is followed by two fully connected layers. Specifically, the network consists of four 1-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 fully connected layers (1500-600). The size of the final softmax layer is determined by the task-specific speaker or language labels and the softmax output can be used directly as a score for each dialect class for the DID task. We considered MFCCs and Mel-Filterbank energies (FBANK) as inputs to the end-to-end system.

3. UNSUPERVISED LEARNING USING FHVAE

In this section, we introduce an FHVAE model for unsupervised representation learning from dialectal speech and explain how we extract features from such models for dialect identification.

An FHVAE [20] is a variant of variational autoencoders [21, 27] that models a probabilistic hierarchical generative process of sequential data, and learns disentangled and interpretable representations. Generation of a sequence of N segments, $\boldsymbol{X} = \{\boldsymbol{x}^{(n)}\}_{n=1}^{N}$ involves one sequence-level latent variable: $\boldsymbol{\mu}_2$, and N pairs of segment-level latent variable: $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$, as follows:

1. a *s-vector* $\boldsymbol{\mu}_2$ is drawn from $p(\boldsymbol{\mu}_2) = \mathcal{N}(\boldsymbol{\mu}_2|\boldsymbol{0}, \sigma^2_{\boldsymbol{\mu}_2}\boldsymbol{I})$.
2. N i.i.d. *latent segment variables* $\boldsymbol{Z}_1 = \{\boldsymbol{z}_1^{(n)}\}_{n=1}^{N}$ are drawn from a global prior $p(\boldsymbol{z}_1) = \mathcal{N}(\boldsymbol{z}_1|\boldsymbol{0}, \sigma^2_{\boldsymbol{z}_1}\boldsymbol{I})$.
3. N i.i.d. *latent sequence variables* $\boldsymbol{Z}_2 = \{\boldsymbol{z}_2^{(n)}\}_{n=1}^{N}$ are drawn from a sequence-dependent prior $p(\boldsymbol{z}_2|\boldsymbol{\mu}_2) = \mathcal{N}(\boldsymbol{z}_2|\boldsymbol{\mu}_2, \sigma^2_{\boldsymbol{z}_2}\boldsymbol{I})$.
4. N i.i.d. sub-sequences $\boldsymbol{X} = \{\boldsymbol{x}^{(n)}\}_{n=1}^{N}$ are drawn from $p(\boldsymbol{x}|\boldsymbol{z}_1, \boldsymbol{z}_2) = \mathcal{N}(\boldsymbol{x}|f_{\mu_x}(\boldsymbol{z}_1, \boldsymbol{z}_2), diag(f_{\sigma^2_x}(\boldsymbol{z}_1, \boldsymbol{z}_2)))$, where $f_{\mu_x}(\cdot, \cdot)$ and $f_{\sigma^2_x}(\cdot, \cdot)$ are parameterized by a decoder neural network.

We illustrate this process in Figure 1. By imposing a sequence-dependent prior to $\boldsymbol{z}_2$, the model is encouraged to represent with $\boldsymbol{z}_2$ the generating factors that are relatively consistent within a sequence. For example, such factors can include microphone frequency response, room impulse response, and general vocal tract characteristics particular to a speaker. On the other hand, $\boldsymbol{z}_1$ tends to encode information about the residual generating factors that change from segment to segment, such as phonetic/linguistic content.

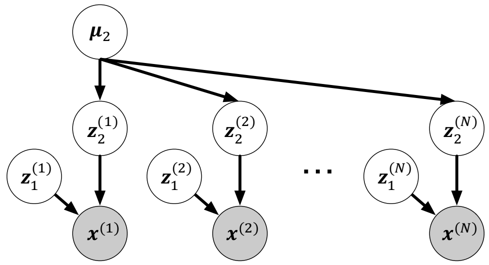

Fig. 1. Graphical illustration of the FHVAE generative model. Grey nodes denote the observed variables, and white nodes are the latent variables.

Since the exact posterior is intractable, an FHAVE approximates the true posterior with an amortized inference model $q(\boldsymbol{\mu}_2|\boldsymbol{X})\prod_{n=1}^{N} q(\boldsymbol{z}_1^{(n)}|\boldsymbol{x}^{(n)}, \boldsymbol{z}_2^{(n)})q(\boldsymbol{z}_2^{(n)}|\boldsymbol{x}^{(n)})$. Specifically, both $q(\boldsymbol{z}_1|\boldsymbol{x}, \boldsymbol{z}_2)$ and $q(\boldsymbol{z}_2|\boldsymbol{x})$ are Gaussian distributions, whose mean and variance are parameterized by two encoder neural networks with the variables they are conditioned on as inputs. Specifically, both $q(\boldsymbol{z}_1|\boldsymbol{x}, \boldsymbol{z}_2)$ and $q(\boldsymbol{z}_2|\boldsymbol{x})$ are Gaussian distributions. Each of them is parameterized by one encoder neural network that predicts the posterior mean and variance, with the variables it is conditioned on as inputs. On the other hand, $q(\boldsymbol{\mu}_2|\boldsymbol{X})$ is a fixed-variance Gaussian distribution, whose mean is parameterized with a maximum a posterior (MAP) estimation $\sum_{n=1}^{N} \hat{z}_2^{(n)}/(N + \sigma^2_{\boldsymbol{z}_2}/\sigma^2_{\boldsymbol{\mu}_2})$ for testing, where $\hat{z}_2^{(n)}$ denotes the posterior mean of $\boldsymbol{z}_2$ for $\boldsymbol{x}^{(n)}$. For training, a trainable lookup table of $\boldsymbol{\mu}_2$ posterior mean for each training sequence is used instead. which enables optimization at the segment level, and utilization of the discriminative loss proposed in [20] to encourage disentanglement.

In this work, we pool the in-domain and out-of-domain dialectal speech data to train an FHVAE, in order to learn a disentangled and interpretable representation. Note that since training of FHVAE models is unsupervised, we can actually apply the FHVAE model even though the dataset has no dialect label. In addition, because at the acoustic level, variation in dialects correlates with phonetic and lexical variability, but not channel response or vocal tract characteristics, we argue that useful information regarding dialect identification is actually encoded in $\boldsymbol{z}_1$ instead of $\boldsymbol{z}_2$. Hence, similar to [28], we extract $\boldsymbol{Z}_1$ for each sequence as the new feature representation and use it for training DNN-based end-to-end systems. After an FHVAE model is trained, we extract $\boldsymbol{Z}_1$ for each sequence and use it Among the disentangled generating factors, we hypothesize that useful information regarding dialect identification is encoded in $\boldsymbol{z}_1$

4. DOMAIN MISMATCHED DIALECTAL SPEECH

The MGB-3 dataset partitions are shown in Table 1. Each partition consists of five Arabic dialects: EGY, LEV, GLF, NOR, and MSA. Detailed corpus details can be found in [29]. Although the 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. In the following experiment, we divided the train and development set into a small subset and limit the use of labels on the development set to simulate a low-resource condition.

5. EXPERIMENT

5.1. Experiment setup

For i-vector extraction, MFCCs were used to generate 60-dimensional acoustic features which consist of 20 MFCCs and their delta and delta-delta's. Cepstral Mean Normalization was used for feature normalization. A GMM-UBM was trained using MFCCs with 2,048 mixture components, then a Total Variability (TV) matrix was trained to extract 600-dimensional i-vectors. Since the GMM-UBM and TV can be trained without labels, we used training and development dataset for all experimental conditions in next sections. A Support Vector Machine (SVM) was used to measure the similarity between a test utterance and 5 dialects [8]. We did not consider a bottleneck-feature (BNF) based i-vector system for comparison. A BNF extractor must be trained with supervision using additional data which have phoneme alignments. Using this extra information was not comparable to any system using only the MGB-3 data.

For the end-to-end DID system, we used MFCC and FBANK 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 coefficient were extracted for both features and then normalized to have zero mean and unit variance. The DNN structure is the same as [12], with four CNN and 2 FC layers as described in Section 2. 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. We used a different dataset for training the network considering various experimental condition since the end-to-end DNNs need dialect label.

Table 1. MGB-3 Dialectal Arabic Speech Dataset Properties.

Dataset	Training	Development	Test
Utterances	13,825	1,524	1,492
Size	53.6 hrs	10 hrs	10.1 hrs
Channel (recording)	Carried out at 16kHz	Downloaded directly from a high-quality video server	

Table 2. Baseline accuracy performance when using train and development dataset for training.

System	Accuracy on Test set	
	If Dev. set is labeled	If Dev. set is unlabeled
I-vector	57.44	46.11
End-to-End (MFCC)	**65.55**	**48.86**
End-to-End (FBANK)	64.81	47.11

To train FHVAE models, we let each segment $\boldsymbol{x}$ be 20 frames of FBANK features. Following the setting in [20], we set $\sigma^2_{\boldsymbol{z}_1} = \sigma^2_{\boldsymbol{\mu}_2} = 1$, $\sigma^2_{\boldsymbol{z}_2} = 0.25$, and dimensions of $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ to be both 32. We configured the two encoders and the decoder to be two-layer LSTM[30] networks with 256 memory cells, followed by affine transform layers predicting mean and log variance of corresponding variables, similar to the architecture used in [28]. FHVAE models are trained to maximize the discriminative segment variational lower bound proposed in [20] with a discriminative weight $\alpha = 10$. Adam [31] with $\beta_1 = 0.95$ and $\beta_2 = 0.999$ is used to optimize all models. Tensorflow [32] is used for implementation.

The performance was measured in accuracy, Equal Error Rate (EER) and minimum decision cost function $\mathrm{C_{avg}}$*100. Accuracy was measured by taking the dialect showing the maximum score between a test utterance and the 5 dialects. Minimum $\mathrm{C_{avg}}$ *100 was computed from hard decision errors and a fixed set of costs and priors from [33].

5.2. Resource Limitation Impact on Domain Mismatch

We compared the performance of the baseline approaches in Table 2. When using the training and development set including their label, the end-to-end system shows impressive performance, since the approach is powerful when the training and test conditions are the same. However, when the development set has no labels, all three baselines show similar accuracy and the end-to-end system no longer has an advantage compared to the traditional i-vector approach. It is interesting that i-vector does not have advantage on this case although UBM is trained using both train and development set since the model does not need supervision. This analysis implies that although the development dataset is only 10 hours long, averaging 2 hours for each dialect, it carries valuable information about the target domain and has a powerful impact on the performance. Also, it is observed that GMM-UBM is not efficient to learn domain mismatched information when the in-domain set is very small. Both i-vector and end-to-end systems are unable to use this valuable information properly due to lack of labels and accuracy subsequently degraded about 20% and 25% respectively compared to when there are labels. We conducted more detailed performance comparison experiments on the two condition in the next section.

Table 3. Performance on resource-rich condition (development set with labels).

	Accuracy	EER	$\mathrm{C_{avg}}$*100
i-vector	57.44	24.43	23.79
End-to-end (MFCC)	65.55	20.24	19.92
End-to-end (FBANK)	64.81	20.22	19.91
End-to-end (FHVAE_$\boldsymbol{z}_1$)	**67.98**	**18.62**	**18.32**
End-to-end (FHVAE_$\boldsymbol{z}_2$)	54.55	27.39	27.35

Table 4. Performance on resource-poor condition (development set without labels).

	Accuracy	EER	$\mathrm{C_{avg}}$*100
i-vector	46.11	32.77	32.08
End-to-end (MFCC)	48.86	29.31	28.61
End-to-end (FBANK)	47.86	30.19	29.67
End-to-end (FHVAE_$\boldsymbol{z}_1$)	**58.16**	**25.40**	**24.66**
End-to-end (FHVAE_$\boldsymbol{z}_2$)	36.36	39.00	38.32

5.3. Resource-Rich Condition

Consider the resource-rich condition in which there are labels on the in-domain data. In this condition, we can fully utilize the development set with dialect labels as part of training. All end-to-end approaches except FHVAE_$\boldsymbol{z}_2$ show significantly better performance than i-vector approaches on all measurements. Using $\boldsymbol{z}_1$ of FHVAE is the best system and $\boldsymbol{z}_2$ of FHVAE is the worst as shown in Table 3. As we expected, linguistic information to distinguish language is encoded at $\boldsymbol{z}_1$, the latent segment variable, and shows better performance than other features. On the other hand, because the latent sequence variable $\boldsymbol{z}_2$ carries information that is not directly related to the dialect identity, such as channel, it is not surprising that the model trained on $\boldsymbol{z}_2$ has the worst performance.

5.4. Resource-Poor Condition

As a fully-labeled in-domain dataset is not always available or can be difficult to collect, we consider a resource-poor condition, where an in-domain dataset is available but without any labels. For this condition, an i-vector extractor and GMM-UBM are trained without supervision, so the in-domain data could be used for training even though it is a small amount. However, the end-to-end system is unable to use the in-domain data because training must be done in a supervised way. As for the resource-poor condition, using $\boldsymbol{z}_1$ of FHVAE shows the best performance on all measurements as shown in Table 4. It is interesting that the performance is very similar to that achieved by the i-vector system on the resource-rich condition.

Apart from the absolute numbers in the tables, the greater performance degradation due to lack of in-domain labels implies that the feature or system is more dependent on the domain. The FHVAE_$\boldsymbol{z}_1$ based end-to-end approach accuracy

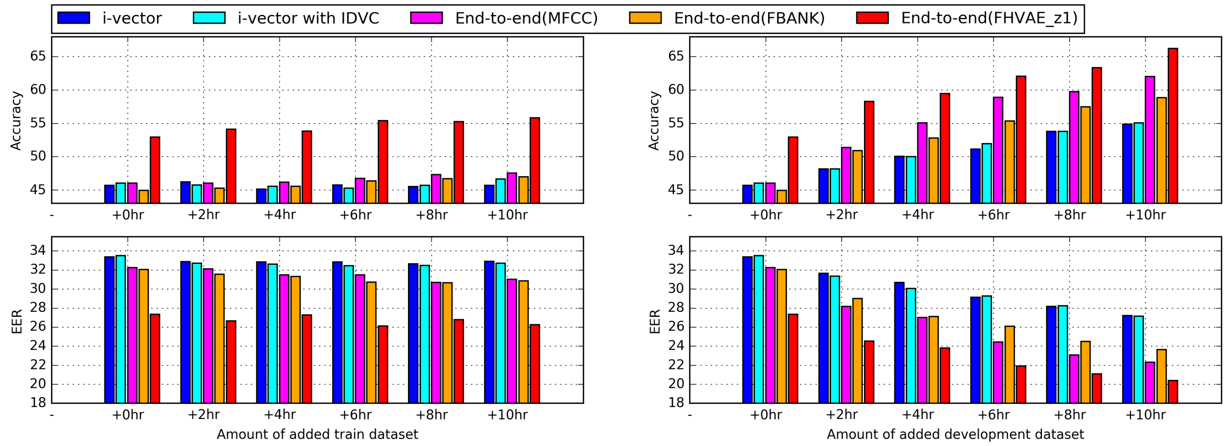

Fig. 2. Performance variation with incremental train (mismatched domain) and development data (matched domain)

Table 5. Division of Training and Development Dataset.

Dataset	Trn-30h	Trn-1	Trn-2	Trn3	Trn-4	Trn5
Utterances	7441	554	511	518	548	549
Size	30hrs	2hrs	2hrs	2hrs	2hrs	2hrs
Dataset	Dev-1	Dev-2	Dev-3	Dev-4	Dev-5	
Utterances	308	303	322	318	315	
Size	2hrs	2hrs	2hrs	2hrs	2hrs	

Table 6. Dataset usage by system. α is additional labeled set that can be added from $\{$Trn-1$\sim$5, Dev-1$\sim$5$\}$ from Table 5.

System	Unlabeled data (Train + Dev.)	Labeled data (Train-30h + α)
i-vector	UBM, TV	SVM
End-to-end (MFCC, FBANK)	-	End-to-end model
End-to-end (FHVAE)	FHVAE model	End-to-end model

degraded only 14% compared to the resource-rich condition, but the other methods degraded about 20%-26%. Particularly, FHVAE_$\boldsymbol{z}_2$ degraded by 33% which implies that the $\boldsymbol{z}_2$ variable is significantly domain-related as we argued previously.

To examine the efficiency of the proposed approach, we partition the train and development datasets into smaller subsets to have specific amounts of data, as shown in Table 5. While all speech from the train and development sets are available, we only used labels for 30 hours of speech from the train set as an "essential" system, and gradually added from 2 to 10 hours of additional labels from the train and development sets in 2 hour increments. Since each system has a different algorithm, we specified dataset usage in Table 6. For example, to add 6 hours of labels from the development set, we used Dev-1, Dev-2 and Dev-3 labels from Table 5. The result is shown in Figure 2. We also applied IDVC, which can be applied to the i-vector system for domain compensation.

When adding the train set, the systems show slight or no improvement except the end-to-end system using the FHVAE $\boldsymbol{z}_1$ variable. For both conditions, the addition of IDVC does not substantially improve the baseline i-vector results. When incrementally providing an additional 2 hours of the train dataset, the improvement on accuracy and EER is under 1% for all systems. However, when incrementally adding 2 hours of the dev dataset, the accrual is between 4% and 7% for the i-vector based and end-to-end approaches respectively. In particular, the gap between MFCC based and $\boldsymbol{z}_1$ of the FHVAE based end-to-end approach is getting closer when adding in-domain (development) set while they keep the same separation when adding out-of-domain (train) set. This observation tells us that if we did not decide on a target domain or did not know about the target domain, we can extract domain invariant features using FHVAE by adding large quantities of unlabeled data from various domains. And it also indicates that if we know the target domain precisely, and it is possible to obtain a target domain dataset with rich labels, using MFCCs or raw feature on the end-to-end system can be optimized to the specific domain and gives a comparable performance to the $\boldsymbol{z}_1$ FHVAE system.

5.5. Performance comparison with uBNF

In this section, we compared uBNF feature [18, 19] which was extracted from DNN trained in an unsupervised manner. The approach was successfully adopted on DID task by modifying network structure considering the relatively small size of dataset [34]. We trained GMM-UBM model with same parameter as [34]. Using the posterior label estimation, 4-layer DNN was trained to extract 40-dimensional uBNF feature. The extracted uBNF feature was used to train the end-to-end DID system in both resource-rich and poor condition. In both conditions, the proposed FHVAE_z_1 shows better per-

Table 7. Performance comparison on MGB-3 test set.

Resource-poor	Accuracy	EER	C_{avg}*100
End-to-end (uBNF [18]	56.64	27.46	26.92
End-to-end (FHVAE_z_1)	**58.16**	**25.40**	**24.66**
Resource-rich	Accuracy	EER	C_{avg}*100
End-to-end (uBNF [18])	66.24	19.98	19.63
End-to-end (FHVAE_z_1)	**67.98**	**18.62**	**18.32**

formance in all indexes. A possible reason for this difference is that the uBNF obtain the ground truth label from GMM-UBM, the performance is dependent on the GMM-UBM. But the FHVAE does not depend on such as label obtained from another unsupervised learning, it has more advantage to learning the speech unsupervised manner.

6. CONCLUSION

In this paper, we describe domain invariant features from unsupervised learning of dialectal speech. The feature, a latent segmental variable, can be encoded by an FHVAE. We investigated the proposed approach along with several baselines such as i-vectors and end-to-end methods based on CNNs using conventional acoustic features. The experiments explored two scenarios, whether the in-domain dataset has a dialect label or not, in order to explore the effectiveness of unsupervised learning of dialectal speech in various domains. From the experiments, we observed that the proposed approach is able to separate segmental and sequential level information that generalize better to new domains. While the proposed approach shows significant improvement in all conditions, we verified that it has a greater advantage in the case where a large amount of unannotated audio is available.

7. REFERENCES

[1] Patrick Cardinal, Najim Dehak, Yu Zhang, and James Glass, "Speaker adaptation using the i-vector technique for bottleneck features," in *Interspeech*, 2015, pp. 2867–2871.

[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] Najim Dehak, Pedro a. Torres-Carrasquillo, Douglas Reynolds, and Reda Dehak, "Language recognition via Ivectors and dimensionality reduction," in *Interspeech*, 2011, pp. 857–860.

[4] Ma Jin, Yan Song, Ian McLoughlin, Wu Guo, and Li Rong Dai, "End-to-end language identification using high-order utterance representation with bilinear pooling," in *Interspeech*, 2017, pp. 2571–2575.

[5] Trung Ngo Trong, Ville Hautamaki, and Kong Aik Lee, "Deep Language : a comprehensive deep learning approach to end-to-end language recognition," in *Proceedings of Odyssey - The Speaker and Language Recognition Workshop*, 2016, pp. 109–116.

[6] Arsha Nagraniy, Joon Son Chung, and Andrew Zisserman, "VoxCeleb: A large-scale speaker identification dataset," in *Interspeech*, 2017, pp. 2616–2620.

[7] David Snyder, Pegah Ghahremani, Daniel Povey, Daniel Garcia-Romero, and Yishay Carmiel, "Deep Neural Network Embeddings for Text-Independent Speaker Verification," in *Interspeech*, 2017, pp. 165–170.

[8] 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.

[9] Sameer Khurana, Maryam Najafian, Ahmed Ali, Tuka Al Hanai, Yonatan Belinkov, and James Glass, "QMDIS : QCRI-MIT Advanced Dialect Identification System," in *Interspeech*, 2017, pp. 2591–2595.

[10] Maryam Najafian, Sameer Khurana, Suwon Shon, Ahmed Ali, and James Glass, "Exploiting Convolutional Neural Networks for Phonotactic Based Dialect Identification ," in *ICASSP*, 2018.

[11] 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.

[12] 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.

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

[14] 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.

[15] 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.

[16] 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.

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

[18] 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.

[19] Qian Zhang and John H.L. Hansen, "Dialect recognition based on unsupervised bottleneck features," in *Proc. Interspeech 2017*, 2017, pp. 2576–2580.

[20] Wei-Ning Hsu, Yu Zhang, and James Glass, "Unsupervised learning of disentangled and interpretable representations from sequential data," in *Advances in Neural Information Processing Systems*, 2017.

[21] Diederik P Kingma and Max Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[22] Pavel Matejka, Le Zhang, Tim Ng, Sri Harish Mallidi, Ondrej Glembek, Jeff Ma, and Bing Zhang, "Neural Network Bottleneck Features for Language Identification," in *Proceedings of Odyssey - The Speaker and Language Recognition Workshop*, 2014, pp. 299–304.

[23] Yu Zhang, Ekapol Chuangsuwanich, and James Glass, "Extracting Deep Neural Network Bottleneck Features Using Low-Rank Matrix Factorization," in *IEEE ICASSP*, 2014, pp. 185–189.

[24] 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.

[25] David Snyder, Daniel Garcia-Romero, Alan McCree, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur, "Spoken language recognition using x-vectors," in *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop*, 2018, pp. 105–111.

[26] Suwon Shon, Hao Tang, and James Glass, "Frame-level Speaker Embeddings for Text-independent Speaker Recognition and Analysis of End-to-end Model," in *IEEE Spoken Language Technology Workshop (SLT)*, 2018.

[27] Wei-Ning Hsu, Yu Zhang, and James Glass, "Learning latent representations for speech generation and transformation," in *Interspeech*, 2017, pp. 1273–1277.

[28] Wei-Ning Hsu and James Glass, "Extracting domain invariant features by unsupervised learning for robust automatic speech recognition," in *IEEE ICASSP*. IEEE, 2018.

[29] 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.

[30] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[31] Diederik Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[32] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al., "Tensorflow: A system for large-scale machine learning.," in *OSDI*, 2016, vol. 16, pp. 265–283.

[33] "The NIST 2015 Language Recognition Evaluation Plan", Available : https://www.nist.gov/document/lre15 evalplanv23pdf," .

[34] Ahmet E. Bulut, Qian Zhang, Chunlei Zhang, Fahimeh Bahmaninezhad, and John H. L. Hansen, "UTD-CRSS Submission for MGB-3 Arabic Dialect Identification: Front-end and Back-end Advancements on Broadcast Speech," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2017.