CNN WITH PHONETIC ATTENTION FOR TEXT-INDEPENDENT SPEAKER VERIFICATION

Tianyan Zhou, Yong Zhao, Jinyu Li, Yifan Gong, Jian Wu

Microsoft Corporation, USA

ABSTRACT

Text-independent speaker verification imposes no constraints on the spoken content and usually needs long observations to make reliable prediction. In this paper, we propose two speaker embedding approaches by integrating the phonetic information into the attention-based residual convolutional neural network (CNN). Phonetic features are extracted from the bottleneck layer of a pretrained acoustic model. In implicit phonetic attention (IPA), the phonetic features are projected by a transformation network into multi-channel feature maps, and then concatenated with the raw acoustic features as the input of the CNN network. In explicit phonetic attention (EPA), the phonetic features are directly connected to the attentive pooling layer through a separate 1-dim CNN to generate the attention weights. With the incorporation of spoken content and attention mechanism, the system can not only distill the speaker-discriminant frames but also actively normalize the phonetic variations. Multi-head attention and discriminative objectives are further studied to improve the system. Experiments on the VoxCeleb corpus show our proposed system could outperform the state-of-the-art by around 43% relative.

Index Terms— speaker verification, attentive pooling, phonetic information

1. INTRODUCTION

Speaker verification is the task of determining whether a pair of speech recordings is spoken by the same identity. As one of the most natural and convenient ways for biometric authentication, it has become an increasing demand for a wide range of applications, including security-controlled access to confidential information, criminal investigation and mobile payment [1]. According to the application scenarios, the speaker verification task can be classified into two categories, text-dependent speaker verification (TDSV) and text-independent speaker verification (TISV). As TISV imposes no constraints on the spoken content, it is more challenging than TDSV.

In the past, traditional TISV systems are based on i-vectors [2]. The system consists of a universal background model (UBM-GMM), an unsupervised projection using factor analysis (i-vectors) and a supervised probabilistic linear discriminant analysis (PLDA) in the backend to compute a similarity score between i-vectors [3, 4, 5, 6]. These individual components are loosely connected and optimized using the different criteria.

In the last few years, more studies have presented the results of end-to-end training using deep neural networks. In [7], a supervised DNN was trained for TDSv to learn the frame-level speaker representation, called d-vectors, and achieved up to 25% improvement through combination with classical i-vector system. [8] further reduced the equal error rate by 3% on the "Ok Google" dataset. In TISV, [6] proposed a DNN to produce segment-level embedding, showing competitive results on long duration test conditions. [9] trained a robust DNN embedding called x-vectors and obtained superior performance on NIST SRE16 evaluation set.

More recently, deep convolutional neural network (CNN) based system has become an effective solution for speaker verification due to the ability to capture local temporal and frequency patterns. [10, 11, 12, 13, 14] all use CNN augmented by residual blocks [15] as the speaker embedding extractor, significantly outperforming i-vectors for short-term utterances under unconstrained conditions. We also choose CNN as our system architecture.

On the other hand, phonetic information provides better alignment for content and could assist speaker recognition system, especially for TISV. Researchers have proposed several strategies to utilize phonetic features, including replacing the tokens (i.e., GMM components) in i-vector framework with tied triphone states [16, 17, 18, 19] and directly introducing this information into DNN training [20, 21, 22].

In this paper, we propose two speaker embedding approaches by integrating the phonetic information via the attention mechanism into the CNN network for the TISV system. Given the local connectivity and spatial contiguity in convolutional operations, we cannot directly feed the phonetic features into the CNN network. In implicit phonetic attention (IPA), the phonetic features are projected by a transformation network into multi-channel feature maps, and then concatenated with the raw acoustic features as the input of the CNN network. An attentive pooling layer is employed to extract speaker-discriminative info from the augmented input features. In explicit phonetic attention (EPA), the phonetic features are directly connected to the attentive pooling layer to generate the attention weights. In order to match the

feature map length of CNN output, we apply several layers of 1-dim convolutions to phonetic features along the time axis. Experimental results on the VoxCeleb corpus show the effectiveness of the proposed phonetic attention systems.

The rest of this paper is organized as follows: Section 2 describes the fundamentals of the CNN-based text-independent speaker verication system. Section 3 provides the detailed implementation of the proposed phonetic attention system. Experimental results will be presented and discussed in Section 4. In Section 5, we conclude our contribution and state future work.

2. CNN-BASED SPEAKER EMBEDDING

This section describes the structure of the CNN-based text-independent speaker verication system developed for this study. A typical end-to-end speaker verification system consists of three main components. First of all, a frame-level speaker embedding extractor. Then, an aggregation layer is applied to summarize the frame-level representations and yield a fixed-dimensional utterance-level speaker embedding. Lastly, a speaker-discriminative criterion is designed to minimize the training objective. We describe the details of these components in the remaining of this section.

2.1. CNN architecture

CNN has been proved to be extremely successful in many applications. However, with network depth increasing, we always suffer from saturation and performance degradation. The residual convolution neural network (ResNet) introduces shortcut connections between blocks of convolutional layers, which allows training deeper networks without incurring gradient vanishing problem [15]. Its superior performance has been demonstrated in vision, speech and many other areas.

Table 1 shows the architecture of our ResNet-based network. It consists of 5 convolutional layers without residual connection and on top of each of the first three layers, two residual blocks are inserted. These residual blocks do not reduce the size of the feature maps. All convolutional layers, 17 in total, are followed by a batch normalization layer and rectified linear units (ReLU) activation function. The acoustic features we used are 80-dimensional log filter banks (LFB). Our frame-level speaker embedding has a dimension of 128.

In order to apply batch processing, we randomly crop input utterances to 5 seconds. If an utterance is shorter than 5 seconds, we extend the signal by duplicating the utterance.

2.2. Aggregation layer

In order to obtain a fixed length speaker embedding for utterances of variable length, we could simply apply a temporal average pooling (TAP) to the frame-level representations. However, not all frames provide equal evidence to infer speaker identities. Instead of averaging, the attention mechanism [23] provides a better alternate to actively select the hidden representations and emphasize speaker-discriminative information.

Table 1: *CNN architecture for speaker embedding. Notation for convolutional layer: (channel, kernel size, stride). TAP: temporal average pooling, AP: attentive pooling*

Module	Output Size
input layer	$80 \times T \times 1$
conv: $(64, 3 \times 3, 2)$ $2 \times \begin{bmatrix} 64, 3 \times 3, 1 \\ 64, 3 \times 3, 1 \end{bmatrix}$	$39 \times T/2 \times 64$
conv: $(64, 3 \times 3, 2)$ $2 \times \begin{bmatrix} 64, 3 \times 3, 1 \\ 64, 3 \times 3, 1 \end{bmatrix}$	$19 \times T/4 \times 64$
conv: $(128, 3 \times 3, 2)$ $2 \times \begin{bmatrix} 128, 3 \times 3, 1 \\ 128, 3 \times 3, 1 \end{bmatrix}$	$9 \times T/8 \times 128$
conv: $(256, 3 \times 3, 1)$	$4 \times T/8 \times 256$
conv: $(128, 3 \times 3, 1)$	$1 \times T/8 \times 128$
aggregation layer	TAP / AP
fc: $(128, 5994)$	classification layer

We implement the shared-parameter non-linear multi-head attentive pooling (AP), similar to [24, 25].

$$U = softmax(V^T tanh(W^T H + b)) \tag{1}$$

where H has a size of $d_h \times T'$, it could be the frame-level speaker representations (equals R) or phonetic features. d_h corresponds to hidden feature dimension, and T' is proportional to input utterance length T. W, b, V are learnable parameters with sizes of $d_h \times d_a, d_a \times 1, d_a \times N$ respectively, d_a corresponds to the hidden units of attention layer and N is the number of attention heads. U represents the normalized attention weights with a size of $N \times T'$. Then the final utterance-level speaker embedding could be calculated as a weighted sum.

$$z = UR^T \tag{2}$$

where R represents our frame-level speaker embedding and has a size of $d_h \times T'$. z is the weighted sum of size $N \times d$. If $N = 1$, z would be the final utterance-level speaker embedding, otherwise we flatten z and perform a linear projection to still obtain the d-dimensional utterance-level embedding.

2.3. Training criterion

The typical training criterion for end-to-end speaker verification is to have a classification layer and use softmax loss. At training stage, networks aim at reducing classification error over a set of known speaker identities. At testing stage, this classification layer is removed and the intermediate bottleneck features are extracted as speaker embedding which are

expected to generalize to any number of speakers beyond the training identities. Therefore there exists a gap between the objective of training and testing, i.e., trained network emphasizes the separation over a set of speakers but do not directly optimize the discrimination of speaker embedding.

Many approaches have been proposed to ease this problem. [26, 27] directly compare and optimize the distances between positive and negative pairs/triplets. However, innumerable sample combinations and unstable convergence make them hard to train. L_2-constrained softmax [28] and A-softmax [29] stick with the softmax loss but either have constraints on learned embedding or classification weights, forcing the original softmax loss to increase inter-class distances and decrease intra-class distances. We adopt these two methods due to their effectiveness in face verification systems. The objective function for L_2-constrained softmax is as follows,

$$L = -\frac{1}{M}\sum_{i=1}^{M}\log\frac{e^{W_{y_i}^T z_i + b_{y_i}}}{\sum_{j=1}^{C} e^{W_j^T z_i + b_j}} \tag{3}$$

subject to $\|z_i\|_2 = \alpha,\ i = 1, 2, ...M.$ where z_i is the utterance-level speaker embedding in a mini-batch of size M with label y_i. C is the number of training identities and α is magnitude of z_i, which could either be static or learnable.

Equation 4 defines the objective of A-softmax,

$$L = -\frac{1}{M}\sum_{i=1}^{M}\log\frac{e^{\|z_i\|\cos(m\theta_{y_i,i})}}{e^{\|z_i\|\cos(m\theta_{y_i,i})} + \sum_{j\neq y_i} e^{\|z_i\|\cos(m\theta_{j,i})}} \tag{4}$$

where $\theta_{j,i}$ is the angle between vector W_j and z_i and m is a hyper parameter adjusting the angular margin.

3. PHONETIC ATTENTION

Spoken content is the predominant component in the speech signal and has a profound impact on the perception of speaker identities. Previous studies have demonstrated that incorporating the phonetic information of input utterances yields signicant gains for speaker recognition [16, 21, 20].

It is expected that in the attentive pooling step, high-energy and slow-changing phones, such as vowels, should receive higher attention weights. Silence and voiceless consonants usually contain less identity clues and the network should learn to deemphasize them. Many works [24, 25] have shown that the attentive pooling produced significant improvement for speaker verification task. However, explicitly exploiting the phonetic information in the attentive pooling has not been well investigated.

In this section, we present the two approaches by incorporating phonetic information into the the attention mechanism.

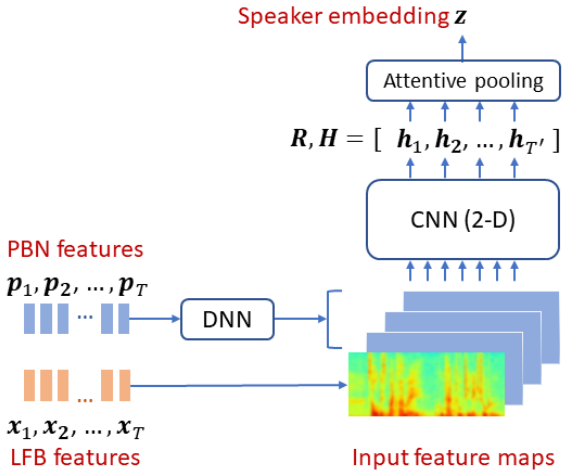

Fig. 1: *Implicit phonetic attention by combining LFB and PBN features at the input layer (LFB: log filter bank; PBN: phonetic bottleneck).*

3.1. Implicit phonetic attention

One popular approach to exploiting the phonetic information is to extract the phonetic bottleneck (PBN) features from the last hidden layer of a pre-trained ASR network. The PBN features are concatenated with the raw acoustic features and fed into the network. Thus, the attentive layer is able to learns the optimal attention weights with the aid of phonetic features. Another benefit is that the network would actively normalize phonetic variations in the meanwhile.

If the speaker embedding network is DNN or long short-term memory (LSTM), it is straightforward to append the phonetic features to the raw acoustic features. However, for the 2-D convolutional networks such as the ResNet in the previous section, this method is inappropriate because we need to preserve spectral contiguous patterns in the input. Figure 1 illustrate the proposed implicit phonetic attention (IPA) to combine two features into feature maps for CNN training. We first transform the bottleneck features using a small fully connected transformation network and reshape the outputs into N feature maps in the same size as raw acoustic features. Then, these features are combined into $N + 1$ multi-channel features as the input of the CNN. The transformation network is jointly trained with the following CNN, so that the phonetic features are projected into the feature maps in the same spatial patterns as the original input features.

In our system, the acoustic feature we used is 80-dim LBF. The phonetic feature has a dimension of 256. The transformation DNN consists of 3 fully connected layers, each of which has 256 hidden units except for the last one with $80 \times N$ hidden units. We set $N = 3$, so there are 4 channels in total at the CNN input layer. Note that our phonetic features are ex-

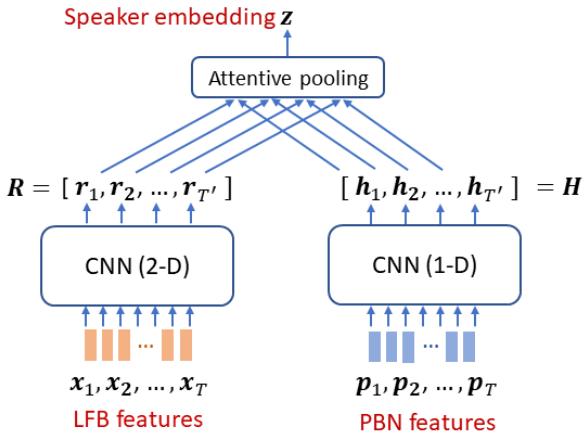

Fig. 2: *Explicit phonetic attention by routing LFB and PBN features through separate networks (LFB: log filter bank; PBN: phonetic bottleneck).*

tracted from a LSTM model. Thus, the whole system can be interpreted as a cascade of LSTM and CNN.

3.2. Explicit phonetic attention

The second approach, called explicit phonetic attention (EPA), is to directly connect phonetic features to the attentive pooling layer and generate the attention weights. This is analogous to cross-layer attention [25] where one layer is used to compute the frame-level outputs, the other layer controls the attention weights.

One issue is that the phonetic outputs need to be synchronized with the frame-level embedding outputs in the striding and pooling operations of convolutions. We applied the 1-dim CNN network to the phonetic features, where the striding operations are synchronized with the main ResNet along the time axis. Thus, the frame-level phonetic and acoustic outputs are generated in parallel and the attention weights can be calculated by substituting the phonetic outputs for H in Eq. (1). In our system, the 1-dim CNN consists of 5 convolutional layers without residual blocks are inserted in between. These layers are followed by a batch normalization and ReLU activation functions.

4. EXPERIMENTS

4.1. Data description

We evaluate the proposed approach for speaker verification on the VoxCeleb corpus [11, 12]. VoxCeleb is a large-scale text-independent dataset with real-life conversational speech collected in unconstrained conditions. Short-term utterances (7.8s in average) and diverse acoustic environments make it more challenging compared with telephone recordings or clean speech. As shown in Table 2, we train the embedding models using VoxCeleb2 'dev' part and mainly evaluate them on the VoxCeleb1 test set composed of 40 speaker identities. For completeness, we also evaluate the other two test sets listed from VoxCeleb2 dataset: the extended VoxCeleb1-E list which uses the entire VoxCeleb1 set (1251 identities), and hard VoxCeleb1-H list which contains speakers with same gender and nationality. All the following experiments share the same training and evaluation sets. Stochastic gradient descent (SGD) is used to train our network with mini-batches of size 128. Momentum and weight decay are set to 0.9 and 0.0001 respectively.

Table 2: *Statistics of dataset.*

Items	Count
Training set (Voxceleb 2 dev)	# utterances: 1092009 # speakers: 5994
Evaluation set (Voxceleb 1)	lists: vox1-test / vox1-all / vox1-hard # test pairs: 37720 / 581480 / 552536

4.2. ASR model for extracting phonetic features

The phonetic bottleneck features are extracted from a deep LSTM acoustic model for large vocabulary speech recognition. The input feature for every 10ms speech frame is 80-dimensional static log Mel filter-banks. The output layer has 9404 nodes, modeling senones. Two acoustic models are trained. The first is a standard 4-layer LSTM [30], trained with 3.4 k hours US English recordings. The second is a 6-layer contextual layer trajectory LSTM (cltLSTM) [31], more powerful than the the 4-layer LSTM, trained with 30 k hours of US English production data. All the LSTM units inside both models are with 1024 memory cells and the output dimension of each layer is reduced to 512 by linear projection. Both acoustic models are then compressed with singular value decomposition (SVD) [32] by keeping 60% of total singular values. After SVD, the linear matrix connecting the last hidden layer with the softmax layer is reduced to two low rank matrices with size 256x512 and 9404x256 respectively. Hence, the bottleneck dimension is 256, and we use the features extracted from this bottleneck layer of each acoustic model as our phonetic representations. On general ASR tasks, the cltLSTM model usually yields more than 20% relative improvement in WER over the 4-layer LSTM model. Unless otherwise stated, the cltLSTM model is used throughout the experiments.

4.3. Effect of attentive pooling

We test the performance of attention-based pooling for LFB features. In our experiments, d is set to 128 and d_a equals 64. The first two rows in Table 3 shows the equal error rate (EER)

using LFB features only with TAP and AP. Attention-based pooling indeed helps to extract useful information for speaker verification. With 16 attention heads, the relative EER reduction is 15.5%, 12.1%, 12.5%. This conclusion is consistent with results reported in the literature [24, 25], confirming the effectiveness of attentive pooling layer.

We believe phonetic attention can further boost the performance of attention layer and also generalize well to unseen speakers and conditions.

Table 3: *Evaluation results with temporal average pooling (TAP) and attentive pooling (AP, 16 attention heads). EERs are reported on VoxCeleb1-test, VoxCeleb1-E and VoxCeleb1-H respectively.*

Method	Aggregation	EER(%)
LFB	TAP	2.64 / 2.48 / 4.12
LFB	AP	**2.23 / 2.18 / 3.61**
IPA	TAP	2.04 / 1.94 / 3.61
IPA	AP	**1.85 / 1.70 / 3.13**
EPA	AP	2.16 / 2.03 / 3.79

4.4. Effect of phonetic attention

Evaluation results with implicit phonetic attention (IPA) and explicit phonetic attention (EPA) are shown in the last three rows of Table 3. Both methods have the ability to boost system performance. However, IPA achieves better numbers than EPA. Comparing second row and fourth row, the relative EER reduction is 17%, 22.1%, 13.3% respectively, which proves phonetic attention could further improve our system on top of normal attention layer. Comparing the first row and third row, we can observe phonetic features play an important role in identity inference. By properly incorporating the LFB and PBN features at a early stage, we could achieve decent results even with simple average pooling.

In order to provide some intuitive insights into the phonetic attention, we plot the spectrogram of an utterance and its corresponding learned 16-heads attention weights. As shown in Figure 3, this utterance is from VoxCeleb1 with file name 'id10270-5r0dWxy17C8-00004.wav'. It contains short pauses and breath in between. We can observe that non-speech areas are mostly assigned with smaller weights than voiced areas, which implies the phonetic attention can effectively emphasize or deemphasize the importance of frames depending on the speech content. For voiced area, the weights of different heads exhibits different patterns. The relevance between the weights and phonetic context demands further study.

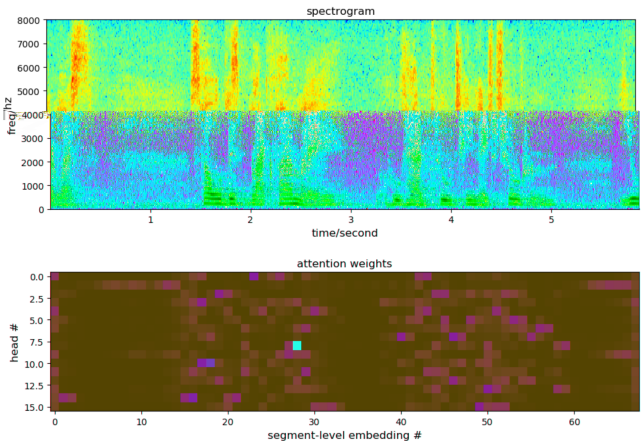

Fig. 3: *Visualization of 16-heads learned attention weights.*

4.5. Effect of attention heads

We examine the performance of attention-based pooling with respect to the various numbers of attention heads. Number of attention heads are increased from 1 to 16 exponentially. Table 4 shows the equal error rate (EER) on VoxCeleb1-test, VoxCeleb1-E and VoxCeleb1-H. As the number of attention heads increasing, system performance keeps improving. A small fluctuation between one head and two heads is because we append an additional linear projection layer when we have more than one head. Comparing first row and last row, with 16 attention heads, the relative EER reduction is 15.5%, 12.1%, 12.5% for LFB only, and 9.3%, 12.4% , 13.3% for adding extra phonetic information.

Table 4: *Evaluation results with different number of attention heads. EERs are reported on VoxCeleb1-test, VoxCeleb1-E and VoxCeleb1-H respectively.*

# heads	LFB	IPA
TAP	2.64 / 2.48 / 4.12	2.04 / 1.94 / 3.61
1	2.56 / 2.38 / 3.99	1.95 / 1.87 / 3.50
2	2.66 / 2.44 / 3.96	2.14 / 2.05 / 3.67
4	2.47 / 2.30 / 3.78	1.99 / 1.86 / 3.37
8	2.40 / 2.25 / 3.73	1.85 / 1.76 / 3.29
16	**2.23 / 2.18 / 3.61**	**1.85 / 1.70 / 3.13**

4.6. Effect of ASR models

Table 6 examines the effect of ASR models for the speaker verification task. We trained two models using implicit phonetic attention, where the PBN features are extracted from the LSTM and cltLSTM models, respectively. The system using the cltLSTM improves the EER slightly by 4.1%, 5.0%

Table 5: *Evaluation results on VoxCeleb1-test, VoxCeleb1-E, and VoxCeleb1-H. [11, 14, 33, 34] do not report results on VoxCeleb1-E, and VoxCeleb1-H.*

System	Training set	Inputs	Loss	Aggregation	EER(%)
Nagrani et al. [11]	VoxCeleb1	i-vectors	-	-	8.80 / - / -
Cai et al. [14]	VoxCeleb1	LFB	A-softmax	AP	4.40 / - / -
Okabe et al. [33]	VoxCeleb1	MFCC	softmax	AP	3.85 / - / -
Hajibabaei et al. [34]	VoxCeleb1	spectrogram	A-softmax	TAP	4.30 / - / -
Chung et al. [12]	VoxCeleb2	spectrogram	softmax+contrastive	TAP	4.19 / 4.42 / 7.33
Xie et al. [13]	VoxCeleb2	spectrogram	softmax	GhostVLAD	3.22 / 3.13 / 5.06
Ours	VoxCeleb2	LFB	softmax	AP	2.23 / 2.18 / 3.61
Ours	VoxCeleb2	LFB	A-softmax	AP	2.85 / 3.01 / 5.57
Ours	VoxCeleb2	LFB	L_2-softmax	AP	2.38 / 2.14 / 3.57
Ours	VoxCeleb2	IPA	softmax	AP	1.85 / 1.70 / 3.13
Ours	VoxCeleb2	IPA	A-softmax	AP	4.43 / 4.05 / 8.46
Ours	VoxCeleb2	IPA	L_2-softmax	AP	**1.81 / 1.68 / 3.12**

and 2.8%, respectively, over the one using the LSTM. The cltLSTM model is trained with ten times more data and has more complicated structure compared with LSTM model. It implies that our phonetic attention network is not very sensitive to the PBN feature quality. A moderate presentation of spoken content could provide enough assistance in speaker-discriminative learning, saving both computational and storage resources.

Table 6: *Evaluation results for implicit phonetic attention with different ASR models.*

ASR models	EER(%)
LSTM	1.93 / 1.79 / 3.22
cltLSTM	**1.85 / 1.70 / 3.13**

4.7. Effect of loss function

Except for softmax loss, we also experiment with two discriminative objective functions, i.e., A-softmax[29] and L_2-constrained softmax[28]. For L_2-softmax, we use a static α and fix it to 10 for all experiments. For A-softmax, we set $m = 4$ and the annealing strategy is used as suggested in the paper appendix. As we can observe from the last 6 lines in Table 5, L_2-softmax slightly improves performance while A-softmax appears to degrade system behavior, especially with phonetic inputs. Further experiments with different configurations should be done to explore the potential for these two loss functions.

4.8. Comparison

In Table 5, we also list other evaluation results reported in the literature on VoxCeleb corpus. The first 4 systems are trained on VoxCeleb1 'dev', so they do not have the evaluation results on VoxCeleb1-E and VoxCeleb-H. Our best system is trained with implicit phonetic attention using L_2-softmax loss, which achieves better performance than all the reported systems on VoxCeleb data and outperform the current state-of-the-art results [13] by 43.8%, 46.3% and 38.3% on three test conditions.

5. CONCLUSIONS

In this paper, we proposed an attention-based deep convolutional network using phonetic information for text-independent speaker verification. The phonetic bottleneck features are extracted from a trained acoustic model for speech recognition, projected by a transformation network into multi-channel feature maps, and then fed into the network together with raw acoustic features. We integrate the system with the multi-head attention and discriminative loss functions to further improve the system performance. The whole system is learned in an end-to-end fashion, so that the system can not only pay attention to the speaker-discriminant frames using phonetic information, but also actively normalize the phonetic variations. Moreover, the proposed architecture allows for the flexible incorporation of phonetic information, which can be simply disabled when it's not available. Experiments on VoxCeleb dataset shows that the use of phonetic features reduce EER by 22.7%, and multi-head attention further reducing EER by 9.3%. Our best system outperforms the current state-of-the-art result by around 43%.

6. REFERENCES

[1] Douglas A Reynolds, "An overview of automatic speaker recognition technology," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2011, pp. IV–4072–IV–4075.

[2] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2010.

[3] S Prince and J Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *IEEE 11th International Conference on Computer Vision*, 2007, pp. 1–8.

[4] Jess A Villalba and Niko Brmmer, "Towards fully bayesian speaker recognition: Integrating out the between-speaker covariance," in *INTERSPEECH 2011, Conference of the International Speech Communication Association, Florence, Italy, August*, 2011, pp. 505–508.

[5] Daniel Garcia-Romero and Carol Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *INTERSPEECH 2011, Conference of the International Speech Communication Association, Florence, Italy, August*, 2011, pp. 249–252.

[6] David Snyder, Daniel Garcia-Romero, Daniel Povey, and Sanjeev Khudanpur, "Deep neural network embeddings for text-independent speaker verification," in *INTERSPEECH*, 2017, pp. 999–1003.

[7] Ehsan Variani, Xin Lei, Erik Mcdermott, Ignacio Lopez Moreno, and Javier Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2014, pp. 4052–4056.

[8] Georg Heigold, Ignacio Moreno, Samy Bengio, and Noam Shazeer, "End-to-end text-dependent speaker verification," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 5115–5119.

[9] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018, pp. 5329–5333.

[10] Chao Li, Xiaokong Ma, Bing Jiang, Xiangang Li, Xuewei Zhang, Xiao Liu, Ying Cao, Ajay Kannan, and Zhenyao Zhu, "Deep speaker: an end-to-end neural speaker embedding system," *arXiv preprint arXiv:1705.02304*, 2017.

[11] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman, "Voxceleb: A large-scale speaker identification dataset," in *INTERSPEECH*, 2017, pp. 2616–2620.

[12] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman, "Voxceleb2: Deep speaker recognition," in *INTERSPEECH*, 2018, pp. 1086–1090.

[13] Weidi Xie, Arsha Nagrani, Joon Son Chung, and Andrew Zisserman, "Utterance-level aggregation for speaker recognition in the wild," *CoRR*, 2019.

[14] Weicheng Cai, Jinkun Chen, and Ming Li, "Exploring the encoding layer and loss function in end-to-end speaker and language recognition system," *CoRR*, vol. abs/1804.05160, 2018.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[16] Yun Lei, Nicolas Scheffer, Luciana Ferrer, and Mitchell Mclaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2014, pp. 1695–1699.

[17] Daniel Garcia-Romero, Xiaohui Zhang, Alan Mccree, and Daniel Povey, "Improving speaker recognition performance in the domain adaptation challenge using deep neural networks," in *Spoken Language Technology Workshop*, 2015.

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

[19] Fred Richardson, Douglas Reynolds, and Najim Dehak, "Deep neural network approaches to speaker and language recognition," *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1671–1675, 2015.

[20] Md. Hafizur Rahman, Ivan Himawan, Mitchell McLaren, Clinton Fookes, and Sridha Sridharan, "Employing phonetic information in dnn speaker embeddings to improve speaker recognition performance," in *INTRSPEECH*, 2018, pp. 3593–3597.

[21] Yi Liu, Liang He, Jia Liu, and Michael T. Johnson, "Speaker embedding extraction with phonetic information," in *INTERSPEECH*, 2018, pp. 2247–2251.

[22] Shi-Xiong Zhang, Zhuo Chen, Yong Zhao, Jinyu Li, and Yifan Gong, "End-to-end attention based text-dependent speaker verification," in *2016 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2016, pp. 171–178.

[23] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, "Neural machine translation by jointly learning to align and translate," *CoRR*, 2014.

[24] Yingke Zhu, Tom Ko, David Snyder, Brian Mak, and Daniel Povey, "Self-attentive speaker embeddings for text-independent speaker verification," in *INTERSPEECH*, 2018, pp. 3573–3577.

[25] F A Rezaur Rahman Chowdhury, Quan Wang, Ignacio Lopez Moreno, and Li Wan, "Attention-based models for text-dependent speaker verification," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018, pp. 5359–5363.

[26] Raia Hadsell, Sumit Chopra, and Yann LeCun, "Dimensionality reduction by learning an invariant mapping," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006, pp. 1735–1742.

[27] Florian Schroff, Dmitry Kalenichenko, and James Philbin, "Facenet: A unified embedding for face recognition and clustering," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 815–823.

[28] Rajeev Ranjan, Carlos D. Castillo, and Rama Chellappa, "L2-constrained softmax loss for discriminative face verification," *CoRR*, vol. abs/1703.09507, 2017.

[29] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song, "Sphereface: Deep hypersphere embedding for face recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6738–6746.

[30] Hasim Sak, Andrew W. Senior, and Françoise Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *INTERSPEECH*, 2014, pp. 338–342.

[31] Jinyu Li, Liang Lu, Changliang Liu, and Yifan Gong, "Improving layer trajectory LSTM with future context frames," in *Proc. ICASSP*, 2019.

[32] J. Xue, J. Li, and Y. Gong, "Restructuring of deep neural network acoustic models with singular value decomposition," in *Proc. Interspeech*, 2013, pp. 2365–2369.

[33] Koji Okabe, Takafumi Koshinaka, and Koichi Shinoda, "Attentive statistics pooling for deep speaker embedding," in *INTERSPEECH*, 2018, pp. 2252–2256.

[34] Mahdi Hajibabaei and Dengxin Dai, "Unified hypersphere embedding for speaker recognition," *CoRR*, vol. abs/1807.08312, 2018.