STATISTICS POOLING TIME DELAY NEURAL NETWORK BASED ON X-VECTOR FOR SPEAKER VERIFICATION

Qian-Bei Hong[1], *Chung-Hsien Wu*[1,2], *Hsin-Min Wang*[1], *and Chien-Lin Huang*[3]

[1]Graduate Program of Multimedia Systems and Intelligent Computing,
National Cheng Kung University and Academia Sinica, Tainan, Taiwan
[2]Department of Computer Science and Information Engineering,
National Cheng Kung University, Tainan, Taiwan
[3]PingAn AI Lab, Palo Alto, CA 94306, USA

ABSTRACT

This paper aims to improve speaker embedding representation based on x-vector for extracting more detailed information for speaker verification. We propose a statistics pooling time delay neural network (TDNN), in which the TDNN structure integrates statistics pooling for each layer, to consider the variation of temporal context in frame-level transformation. The proposed feature vector, named as stats-vector, are compared with the baseline x-vector features on the VoxCeleb dataset and the Speakers in the Wild (SITW) dataset for speaker verification. The experimental results showed that the proposed stats-vector with score fusion achieved the best performance on VoxCeleb1 dataset. Furthermore, considering the interference from other speakers in the recordings, we found that the proposed stats-vector efficiently reduced the interference and improved the speaker verification performance on the SITW dataset.

Index Terms**—** Speaker verification, time delay neural network, statistics pooling

1. INTRODUCTION

Recently, deep neural networks (DNN) have been widely applied to capture speaker characteristics and produce speaker embedding as speaker representation in speaker verification (SV) tasks [1-5]. In previous studies, most SV systems were based on x-vector features [6-7], and the architecture consists of frame-level and segment-level feature transformations. The frame-level feature transformation is based on time delay neural network (TDNN) structure [8]. It has been proven that using TDNNs for extracting speech characteristics through multi-frame signals with shift-invariance is more efficient than single-frame signals [9]. The segment-level feature transformation applies statistics pooling to aggregate variable-length features to obtain a fixed-dimensional vector. In addition, most of the research used probabilistic linear discriminant analysis (PLDA) to compare embedding pairs for speaker verification [10].

Nowadays, many studies were focused on improving performance for speaker verification. Zhu et al. [11] proposed a self-attention mechanism for DNN embedding and computed the embedding as a weighted average of speaker's frame-level features. Tang et al. [12] integrated TDNN and long short-term memory (LSTM) to capture speaker information at different levels. Rahman et al. [13] explored the possibilities of improving speaker recognition performance by employing phonetic information for embedding network training.

On the other hand, speaker verification applied in real-world environments not only accepts speech data from one speaker, but also multi-speakers talking at the same time, especially conversations. Therefore, embeddings extracted from multi-speaker recordings will cause the confusion of speaker characteristics and decrease the recognition performance [14]. This paper proposes a statistics pooling TDNN structure to effectively improve the ability of x-vector learning by capturing more robust speaker characteristics.

2. RELATED WORK

Fig. 1 depicted the baseline x-vector features for speaker verification; the architecture is similar to that in [7]. In this system, a speaker discriminative DNN model is trained by the speech data from a large amount of speakers. We assume that in DNN model learning to capture the characteristics of different speakers from training speakers' recordings, the high-level embedding can be treated as classifiable features in this model and produce speaker embedding called x-vectors. After that, a PLDA backend is used to compare embedding pairs to determine whether the two embeddings are from the same speaker.

In this figure, the first six layers are frame-level transformations that are constructed by a TDNN structure. In this example, t is the current time step; the first layer is the spliced output of a context of frames from $t-2$ to $t+2$, and the second and third layers are the spliced output of the previous layer at frames $\{t-2, t, t+2\}$ and $\{t-3, t, t+3\}$, respectively. In this study, the fourth layer is added as the spliced output of the third layer at frames $\{t-4, t, t+4\}$. Thus, the fourth layer covers a total temporal context of 23 frames. The fifth and sixth layers are the transformations without considering temporal context.

The statistics pooling aggregates all sixth layer outputs to form a fixed-length vector, which computes the mean and standard deviation of all sixth layer outputs and concatenates them together as the segment-level features. After that, the features are forwarded to the seventh dense layer, the eighth dense layer and finally the softmax output layer. The x-vector is extracted from the seventh dense output, and the PLDA backend is used for scoring.

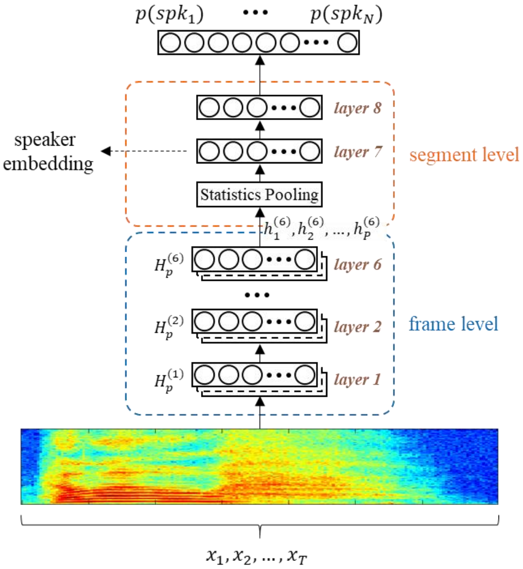

Fig. 1. Structure of the baseline system using x-vector

3. FRAME-LEVEL STATISTICS POOLING TDNN

This paper proposes a new structure to improve the x-vector representation, which is regarded as the state of the art feature representation for speaker verification. As the TDNN layer focuses on local feature extraction, high-level feature extraction through non-linear transformations with low weights in preceding layers may lose some important information using low-level features. Therefore, this study integrates TDNN with the statistics pooling to exploit the potential of the network by considering the variation of temporal context. Fig. 2 shows the proposed new architecture in this study. In order to further improve the information representation in TDNN, we propose a feature combination method for each time delay layer. Given a subsequence of F output vectors $H_p^{l-1} = \{h_{p,1}^{l-1}, h_{p,2}^{l-1}, \dots, h_{p,F}^{l-1}\}$ from the previous $(l-1)$th layer at time step p, the time delay layer output vector h_p^l at the l-th layer is obtained as follows.

$$h_p^l = \alpha\left(W^l H_p^{l-1} + b^l\right) \tag{1}$$

where $W^l \in \mathbb{R}^{D^l \times Q^l}$ is the weight matrix of size $D^l \times Q^l$, D^l is the number of output nodes and Q^l is the number of input nodes; b^l is the bias vector in layer l and $\alpha(\cdot)$ is the activation function.

To further consider the variation in the input features, we directly combine H_p^{l-1} and statistics pooling result of H_p^{l-1} to form a new input feature vector, which is then fed into the next layer:

$$\hat{h}_p^l = \alpha\left(W^l\left[H_p^{l-1} \oplus stat\left(H_p^{l-1}\right)\right] + b^l\right) \tag{2}$$

where $\oplus$ denotes a concatenation operation, $stat(\cdot)$ is the statistics pooling function that computes the mean and standard deviation. Considering that TDNN input is the spliced output of previous layer at different frames, and the continuity of speech means that different frames features are similar to each other, the statistics pooling can represent the variation of local features.

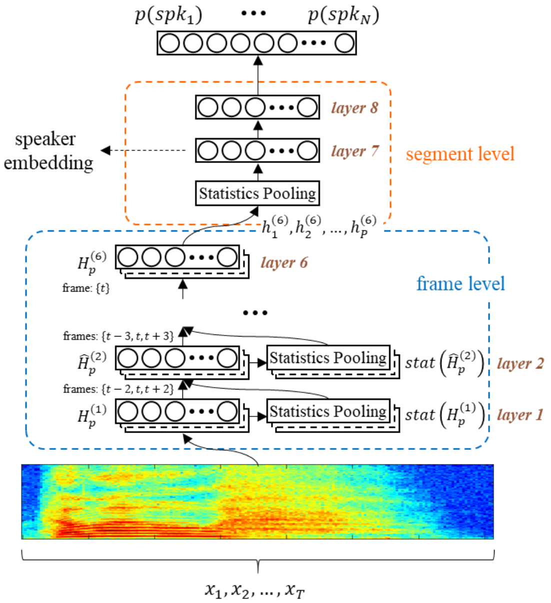

Fig. 2. Structure of the proposed system.

Assuming that the input is stationary speech, each output vector is similar to the other output vectors. The transformation can thus be simplified as follows.

$$\hat{H}_p^{l-1} = \{\hat{h}_{p,1}^{l-1}, \hat{h}_{p,2}^{l-1}, \dots, \hat{h}_{p,F}^{l-1}\} \tag{3}$$

$$\hat{H}^{l-1} = \{\hat{H}_1^{l-1}, \hat{H}_2^{l-1}, \dots, \hat{H}_P^{l-1}\} \tag{4}$$

$$\begin{aligned}\hat{h}_p^l &\approx \alpha\left(W^l\left[E\left[\hat{H}^{l-1}\right] \oplus mean\left(E\left[\hat{H}^{l-1}\right]\right) \oplus std\left(E\left[\hat{H}^{l-1}\right]\right)\right] + b^l\right) \\ &\approx \alpha\left(W^l\left[\hat{H}_p^{l-1} \oplus \hat{h}_{p,f}^{l-1} \oplus std\left(\hat{H}_p^{l-1}\right)\right] + b^l\right)\end{aligned} \tag{5}$$

where $\hat{H}^{l-1}$ is a set of subsequences corresponding to P time steps obtained from the previous $(l-1)$th layer, $mean(\cdot)$ is the mean function and $std(\cdot)$ is the standard deviation function. If the all vectors are equal in layer l, the expectation

$E\left[\widehat{H}^{l-1}\right]$ is a mean vector of $\widehat{H}^{l-1}$, each $std(\cdot)$ operation will produce a zero vector, and the equation can use any output vector $\hat{h}_{p,f}^{l-1}$ instead of $mean\left(E\left[\widehat{H}^{l-1}\right]\right)$ in the previous layer. Thus, $\hat{h}_p^l$ is an approximation of h_p^l as a result of the assumptions on stationarity. Furthermore, if there are different phonemes in the input speech, the features in the preceding layers will obtain different local means and standard deviations. This means that the transformation represents high resolution results, and the statistics pooling represents the low resolution results.

4. EXPERIMENTS AND RESULTS

4.1. Training data

The baseline system and the proposed system were trained on VoxCeleb2 [15] dataset. The VoxCeleb2 dataset provided two datasets for evaluation, in which the *DEV* dataset was used to train the speaker verification systems in this study. The *DEV* set of VoxCeleb2 contained 1,092,009 utterances from 5,994 celebrities, which were obtained from YouTube videos. Because this paper focuses on improving the x-vector for learning more robust speaker representation, data augmentation and noise addition will not be considered. The PLDA model was trained on the same *DEV* dataset of VoxCeleb2. Before PLDA training, the linear discriminant analysis (LDA) was used for dimensionality reduction and the representations were length-normalized.

4.2. Testing data

In this paper, two datasets were used for speaker verification evaluation, including VoxCeleb1 [16] dataset and the Speakers in the Wild (SITW) [17] dataset. The VoxCeleb1 dataset contained 153,516 utterances from 1,251 celebrities, which was also obtained from YouTube videos. VoxCeleb1 dataset was used to evaluate the speaker verification performance on the assumption that the training data and the testing data were collected under the same conditions. The SITW dataset provided samples of approximately 300 individuals across different scenarios and contained multi-speaker presentations in the same utterances. The *EVAL* dataset was used to evaluate the speaker verification performance, which contained 2,883 recordings from 180 speakers.

4.3. Experimental setup

The input features were 40-dimentional Mel-frequency cepstral coefficients (MFCCs), and the spectrogram was extracted from a 25ms window with a stride of 10ms. The baseline system using x-vector and the proposed system were built on the same architecture. In frame-level transformation, there were 512 output nodes in the first five layers and 1,500 output nodes in the sixth layer, while the statistics pooling produced the output nodes that was twice the length of the input nodes. The seventh dense layer and the eighth dense layer also consisted of 512 output nodes. Additionally, if the time delay layer input is a temporal context (not single-frame vector), we concatenated the subsequence H_p^{l-1} of output vectors from previous layer with the statistics pooling results $stat\left(H_p^{l-1}\right)$ in the same context, to form a new input feature vector, e.g., second, third and fourth layers. Batch normalization (BN) and rectified linear unit (ReLU) activation function were applied to each transformation layer for non-linear mapping. All systems were implemented by Kaldi toolkit.

We reported speaker verification results in term of equal error rate (EER) and the minimum detection cost function (DCF) at $P_{target} = 0.01$ (DCF10^{-2}) and $P_{target} = 0.001$ (DCF10^{-3}).

4.4. Experimental results

In the following results, "*x-vector*" refers to baseline system using x-vector described in Section 2. "*stats-vector*" refers to the system using the proposed feature representation described in Section 3. Finally, the term "*fusion*" refers to the score fusion method as follows.

$$scoreF_i = \frac{1}{K}\sum_{k=1}^{K}\left(score_i(k) - \frac{1}{S}\sum_{s=1}^{S} score_s(k)\right) + \frac{1}{KS}\sum_{k=1}^{K}\sum_{s=1}^{S} score_s(k) \qquad (6)$$

where K is the number of speaker verification systems, S is the number of embedding pairs, and $scoreF_i$ is the i-th score that was determined by the average score of each system and total average score of all systems.

4.4.1. Evaluation on *VoxCeleb1*

Table 1 shows the performance of the systems on VoxCeleb1 dataset. The three evaluation conditions are formed by pairing an enrollment condition with a test condition. In general, the stats-vector system outperformed the baseline x-vector system by considering the variation of temporal context in frame-level TDNN. Compared to the baseline x-vector system, the stats-vector system performed better by 6.0%, 1.7% and 1.3% in EER, respectively. Furthermore, the system using score fusion significantly improved the performances on the list of trial pairs in VoxCeleb1 (cleaned), which was improved by 15.4% in EER and 11.6% in DCF10^{-2}. Fig.3 shows the detection error tradeoff (DET) curves for the systems.

4.4.2. Evaluation on *SITW*

The SITW dataset contained different conditions for speaker verification. The four evaluation conditions were formed by pairing an enrollment condition with a test condition. Table 2 shows the performance on SITW *EVAL* dataset. The first trial name refers to the enrollment conditions, and the second trial name refers to the test conditions, e.g., "core" denotes only

Table 1. Results on the VoxCeleb1.

System	VoxCeleb1 (cleaned)			VoxCeleb1-E (cleaned)			VoxCeleb1-H (cleaned)		
	EER	DCF10^{-2}	DCF10^{-3}	EER	DCF10^{-2}	DCF10^{-3}	EER	DCF10^{-2}	DCF10^{-3}
x-vector	3.50	0.4009	0.6012	3.45	0.3915	0.6248	6.02	0.5387	0.7740
stats-vector	3.29	0.3633	**0.4820**	3.39	0.3844	0.6276	5.94	0.5439	0.7849
fusion	**2.96**	**0.3542**	0.5238	**3.11**	**0.3629**	**0.6065**	**5.48**	**0.5184**	**0.7597**

Table 2. Results on the SITW EVAL set.

System	EVAL core-core			EVAL core-multi			EVAL assist-core			EVAL assist-multi		
	EER	DCF10^{-2}	DCF10^{-3}	EER	DCF10^{-2}	DCF10^{-3}	EER	DCF10^{-2}	DCF10^{-3}	EER	DCF10^{-2}	DCF10^{-3}
x-vector	4.87	0.4691	0.7023	7.72	0.5635	0.7744	7.67	0.5134	0.7279	9.22	0.5705	0.7859
stats-vector	4.74	0.4506	**0.6635**	**7.37**	**0.5427**	**0.7524**	**7.31**	**0.4987**	**0.6835**	**8.78**	**0.5507**	**0.7493**
fusion	**4.69**	**0.4495**	0.6773	7.44	0.5450	0.7581	7.43	0.5005	0.7014	8.97	0.5545	0.7627

one speaker in the recordings, “assist” denotes one or more speakers in the enroll recordings and “multi” denotes one or more speakers in the test recordings. The *EVAL* core-core is a common list of trial pairs, in which there is no interference from other speakers. Compared to the baseline x-vector, the stats-vector performed better by 2.7% in EER and 3.9% in DCF10^{-2}. Score fusion further improved the performance by 3.7% in EER and 4.2% in DCF10^{-2}. In addition, considering the interference from other speakers in the recordings, the other three trial lists were used for evaluation. As shown in Table 2, compared to baseline x-vector, the stats-vector obtained the best performance on *EVAL* assist-multi trial list, outperforming by 4.8% in EER and 3.5% in DCF10^{-2}. This means that the stats-vector can efficiently reduce the interference when there are multiple speaker presentations in the recordings. Fig. 4 shows the DET curves comparison on the *EVAL* assist-multi trial pair list. Interestingly, the performance of the method using score fusion did not perform better than the stats-vector. This is because the score fusion is similar to arithmetic mean; different learning errors of interference from other speakers in each system will cause these errors to be accumulated. According to the law of large numbers, the average score obtained from a large amount of systems will tend to the expected value and reduce the effect of prediction errors.

5. CONCLUSIONS

This paper proposes a statistics pooling TDNN architecture for speaker verification. The new structure integrating TDNN with statistics pooling for each layer can effectively consider the variation of temporal context and improve the performance for speaker verification. The experiments were evaluated on VoxCeleb dataset and SITW dataset. We found that stats-vector with score fusion can significantly improve the speaker verification performance on VoxCeleb1 dataset. Furthermore, we evaluated the system performance on the SITW dataset which presented the multiple speakers conversation conditions, and found that the stats-vector can efficiently reduce the interference from other speakers in the recordings. In fact, this study only changed three layers in the frame-level transformation which could improve the performance of speaker verification. In future work, we will use the proposed structure to build a deeper network to further capture more important information with local characteristics to achieve a better performance. Moreover, statistics pooling replaced by attention mechanism has been proven that providing different speaker discriminative information of frames can achieve better performance; we will investigate the potential of different attention methods such as combining articulatory features to further consider the pronunciation manners and places in speech.

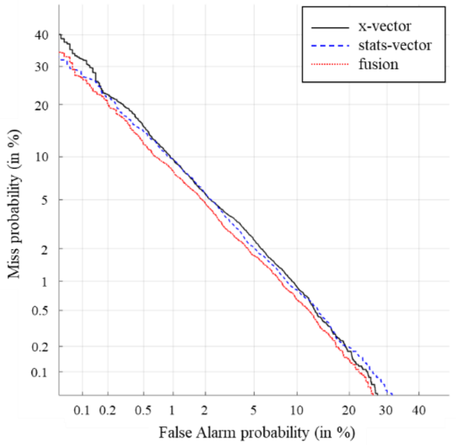

Fig. 3. DET curve for the trial pairs in VoxCeleb1 (cleaned)

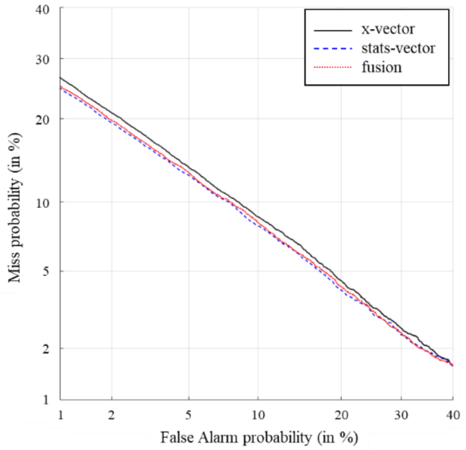

Fig. 4. DET curve for the trial pairs in EVAL assist-multi

6. REFERENCES

[1] H. Muckenhirn, M.M. Doss, and S. Marcell, “Towards directly modeling raw speech signal for speaker verification using CNNs,” *in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, pp. 4884-4888, 2018.

[2] N. Li, D. Tuo, D. Su, Z. Li, and D. Yu, “Deep discriminative embeddings for duration robust speaker verification,” *in INTERSPEECH*, pp. 2262-2266, 2018.

[3] E. Variani, X. Lei, E. McDermott, I.L. Moreno, and J. Gonzalez-Dominguez, “Deep neural networks for small footprint text-dependent speaker verification,” *in 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, pp. 4052-4056, 2014.

[4] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, “End-to-end text-dependent speaker verification,” *in 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, pp. 5115-5119, 2016.

[5] Q. Wang, C. Downey, L. Wan, P.A. Mansfield, and I.L. Moreno, “Speaker diarization with lstm,” *in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, pp. 5239-5243, 2018.

[6] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, “Deep neural network embeddings for text-independent speaker verification,” *in INTERSPEECH*, pp. 999-1003. 2017.

[7] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, “X-vectors: robust dnn embeddings for speaker recognition,” *in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, pp. 5329-5333, 2018.

[8] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K.J. Lang, “Phoneme recognition using time-delay neural networks,” *IEEE Transactions on Acoustics Speech and Signal Processing*, vol. 37, no. 3, pp. 328-339, 1989.

[9] D. Snyder, D. Garcia-Romero, and D. Povey, “Time delay deep neural network-based universal background models for speaker recognition,” *in 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 92-97, 2015.

[10] L. Burget, O. Plchot, S. Cumani, O. Glembek, P. Matějka, and N. Brümmer, “Discriminatively trained probabilistic linear discriminant analysis for speaker verification,” *in 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, pp. 4832-4835, 2011.

[11] Y. Zhu, T. Ko, D. Snyder, B. Mak, and D. Povey, “Self-attentive speaker embeddings for text-independent speaker verification,” *in INTERSPEECH*, pp. 3573-3577, 2018.

[12] Y. Tang, G. Ding, J. Huang, X. He, and B. Zhou, “Deep speaker embedding learning with multi-level pooling for text-independent speaker verification,” *in ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, pp. 6116-6120, 2019.

[13] M.H. Rahman, I. Himawan, M. Mclaren, C. Fookes, and S. Sridharan, “Employing phonetic information in DNN speaker embeddings to improve speaker recognition performance,” *in INTERSPEECH*, pp. 3593-3597, 2018.

[14] D. Snyder, D. Garcia-Romero, G. Sell, A. McCree, D. Povey, and S. Khudanpur, “Speaker recognition for multi-speaker conversations using x-vectors,” *in ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, pp. 5796-5800, 2019.

[15] J. S. Chung, A. Nagrani, and A. Zisserman, “Voxceleb2: deep speaker recognition,” *in INTERSPEECH*, 2018.

[16] A. Nagrani, J. S. Chung, and A. Zisserman, “Voxceleb: a largescale speaker identification dataset,” *in INTERSPEECH*, 2017.

[17] M. McLaren, L. Ferrer, D. Castan, and A. Lawson, “The speakers in the wild (SITW) speaker recognition database,” *in INTERSPEECH*, pp. 818-822, 2016.