SPEAKER CHARACTERIZATION USING TDNN-LSTM BASED SPEAKER EMBEDDING

Chia-Ping Chen, Su-Yu Zhang, Chih-Ting Yeh, Jia-Ching Wang, Tenghui Wang, Chien-Lin Huang

National Sun Yat-Sen University, Taiwan
National Central University, Taiwan
cpchen@mail.cse.nsysu.edu.tw

ABSTRACT

In this paper we propose speaker characterization using time delay neural networks and long short-term memory neural networks (TDNN-LSTM) speaker embedding. Three types of front-end feature extraction are investigated to find good features for speaker embedding. Three kinds of data augmentation are used to increase the amount and diversity of the training data. The proposed methods are evaluated with the National Institute of Standards and Technology (NIST) speaker recognition evaluation (SRE) tasks. Experimental results show that the proposed methods achieve a decision cost of 0.400 with the pooled SRE 2018 development set with a single system. In addition, by applying simple average score combination on the outputs of 12 systems, the proposed methods achieve an equal error rate (EER) of 5.56% and a minimum decision cost function of 0.423 with the SRE 2016 evaluation set.

Index Terms— speaker embedding, TDNN-LSTM, NIST SRE2018

1. INTRODUCTION

The embedding-based speaker recognition systems recently demonstrate sound performance and they become the mainstream methods. The idea of speaker embedding is to find representation for speaker idiosyncrasy, which can be extracted for both enrollment data and test utterance, and then used to make decision regarding true speaker or imposter [1]-[4].

A speaker embedding method based on deep neural network (DNN) has been proposed in 2016 [5]. The proposed architecture was a feed-forward DNN and it outperformed the conventional i-vector. Snyder et al. investigated replacing i-vectors for text-independent speaker verification with embedding extracted from a feed-forward deep neural network [6]. The long-term speaker characteristic can be captured in the network by a temporal pooling layer that aggregates over the input speech. In addition to feed-forward neural networks, the convolutional neural networks (CNN) based speaker recognition was proposed in 2017, and they were experimented with a large-scale speaker recognition dataset called VoxCeleb [7, 8]. The VoxCeleb dataset is collected from Youtube with 16 kHz and in 16-bit format wideband speech. In 2018, data augmentation was used to improve the performance of DNN embedding for speaker recognition, and variable-length utterances are converted to fixed-dimensional embedding vectors, called X-vectors [9], which were trained to discriminate between speakers. X-vectors are based on the time-delayed neural network (TDNN) structure [10]. A self-attention pooling layer was proposed to replace the temporal average pooling layer in X-vectors for text-independent speaker verification [11]. The idea is to compute the speaker embedding as a weighted average of a speaker's frame-level hidden vectors, and their weights are automatically determined by an attention mechanism.

In this paper, we propose a speaker-embedding model called L-vectors based on TDNN and long short-term memory (LSTM) recurrent neural networks (RNN). The motivation of using both TDNN and LSTM in L-vectors is to better capture the temporal information in speech than using TDNN alone as in X-vectors. We investigate three types of front-end feature extraction to analyze speech from different signal aspects. In addition, three kinds of data augmentation are used to increase the amount and diversity of the available training data.

The rest of this paper is organized as follows. In Section 2, we introduce the proposed L-vectors methods based on TDNN and LSTM. In Section 3, we present the experimental results on NIST-SRE tasks. In Section 4, we summarize this work and draw conclusion.

2. THE PROPOSED SYSTEM

2.1. Speaker-Embedding L-vectors

We modify the original TDNN-based X-vector by replacing two TDNN layers (the second and third hidden layers in X-vectors [6]) with an LSTM layer, so the proposed embedding based speaker recognition system is based on TDNN-LSTM as shown in Fig. 1. We refer to this representation as L-vector. The first 4 hidden layers operate at frame-level, while the last 2 layers operate at segment-level. There is a statistics pooling layer between the frame-level and the segment-level layers that aggregates all frame-level outputs from the 4th layer and computes the mean and

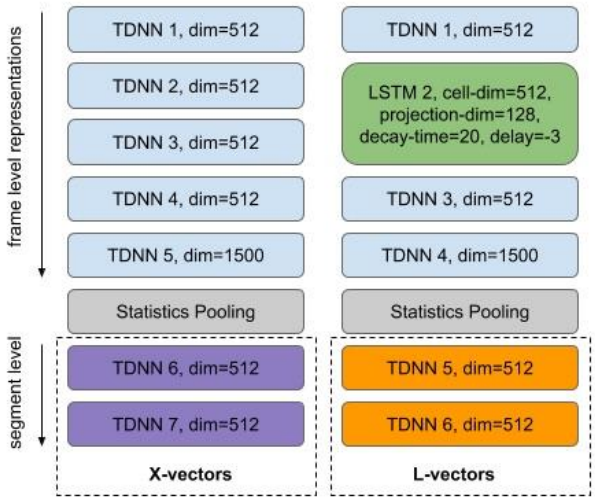

Fig. 1. Diagrams of TDNN for X-vectors and TDNN-LSTM for L-vectors. Speaker-embedding X-vectors and L-vectors are the first and second segment-level layers after the statistics-pooling layer. The statistics-pooling layer is used to estimate the mean and standard deviation from the variable-length inputs.

standard deviation over all frames for an input segment. After training, speaker-embedding L-vectors are extracted from the 512 dimensional affine components of the 5th and 6th layers, i.e. the first and second segment-level layers. The same data is used to train the speaker-embedding X-vectors with the default X-vector neural network setting [9]. After training, speaker-embedding X-vectors are extracted from the 512 dimensional affine components of the 6th and 7th layers, i.e. the first and second segment-level layers.

The proposed speaker-embedding L-vectors are trained on Fisher, Mixer6, NIST-SRE, Switchboard (SWBD) and VoxCeleb. Mixer6 and VoxCeleb are microphone speech. Fisher, NIST-SRE and SWBD are telephone speech. Fisher dataset, with parts 1 and 2, contains 23,392 utterances from 12,399 speakers, so the average number of utterances per speaker is 1.89. Mixer6 dataset contains telephone speech of 8,809 utterances from 591 speakers, and microphone speech of 3,423 utterances from 547 speakers, respectively. The average number of utterances per speaker is 14.91 for telephone speech and 6.26 for microphone speech, respectively. The NIST-SRE dataset, consisting of SRE 2004, 2005, 2006, 2008, and 2010, contains 50,850 utterances from 4,263 speakers. The average number of utterances per speaker is 11.93. There are 28,181 utterances from 2,594 speakers in the SWBD dataset (with phase 1, phase 2, phase 3, cellular part 1, and cellular part 2). The average number of utterances per speaker is 10.86. There are 1,245,525 utterances from 7,245 speakers in the VoxCeleb dataset (containing VoxCeleb1 and VoxCeleb2). The average number of utterances per speaker is 171.92. The training datasets for the proposed speaker-embedding L-vectors are summarized in Table 1. In total, there are approximately 1,360,000 utterances from 26,600 speakers in the training data. We keep the original format of all the audio samples. All the audio samples are down-sampled to 8 kHz and in 16-bit format for feature extraction.

Table 1. Summarization of the training data for the proposed speaker-embedding L-vectors.

Dataset	Utterance	Speaker
Fisher	23,392	12,399
Mixer6	12,232	591
SRE	50,850	4,263
SWBD	28,181	2,594
VoxCeleb	1,245,525	7,245

2.2. Feature Analysis

Three acoustic feature sets are extracted from audio files, including the Mel-frequency cepstral coefficients (MFCCs), perceptual linear predictive (PLP) analysis of speech, and the linear mel-scale filter-bank energies with pitch (FBP). MFCCs are computed using 24 Mel filter banks. The PLP analysis computes 18-order PLP-cepstra. FBP is estimated using 36 mel-scale filter-bank energies. The audio samples are coded with a 25-ms frame window, a 10-ms frame shift, and bandwidth is limited to the range of 100 Hz - 3,700 Hz [12, 13]. We apply three different front-end feature extraction of MFCC, PLP, and FBP to train embedding models. After doing feature extraction, energy-based voice activity detection (VAD) is used to estimate frame-by-frame speech activity, and the frames with silence or low signal-to-noise ratio in the audio samples are removed.

2.3. Data Augmentation

Data augmentation is often used to increase the amount and diversity of the available training data [14]. Three kinds of data augmentation methods are applied in this work to create a 6 copies of original data (feature vectors).

2.3.1 Adding babble, noise, and music

We use the MUSAN dataset [15] to corrupt the original audio files with additive noises, including babble noise, general noise, and music noise, resulting in 3-fold data augmentation.

Table 2. EER and DCF results of the SRE 2018 development set. Pooled means the average of CMN2 and VAST.

		L-vector TDNN 5			L-vector TDNN 6			*X-vector TDNN 6*			X-vector TDNN 7		
		EER	min_DCF	act_DCF	EER	min_DCF	act_DCF	EER	min_DCF	act_DCF	EER	min_DCF	act_DCF
MFCC	CMN2	6.91	0.441	0.446	8.13	0.433	0.442	7.58	0.434	0.453	7.54	0.404	0.413
	VAST	3.70	0.416	0.490	5.35	0.486	0.523	5.35	0.333	0.519	4.12	0.296	0.444
	Pooled			0.468			0.482			0.486			0.429
PLP	CMN2	6.98	0.424	0.435	8.55	0.444	0.448	7.31	0.430	0.437	7.93	0.415	0.427
	VAST	3.70	0.267	0.407	7.41	0.337	0.407	7.41	0.412	0.481	7.41	0.412	0.486
	Pooled			0.421			0.428			0.459			0.456
FBP	CMN2	6.77	0.412	0.429	8.86	0.422	0.431	7.06	0.402	0.409	8.06	0.377	0.384
	VAST	3.70	0.296	0.370	7.41	0.300	0.444	7.41	0.379	0.416	7.41	0.300	0.481
	Pooled			**0.400**			0.438			0.412			0.433

Table 3. EER and DCF results of the SRE 2016 evaluation set.

	L-vector TDNN 5		L-vector TDNN 6		*X-vector TDNN 6*		X-vector TDNN 7	
	EER	min_DCF	EER	min_DCF	EER	min_DCF	EER	min_DCF
MFCC	7.03	0.519	7.93	0.519	7.46	0.537	7.71	0.541
PLP	7.42	0.532	8.19	0.532	7.45	0.544	8.13	0.534
FBP	**6.99**	0.520	7.95	0.511	7.14	0.519	7.65	**0.505**

2.3.2 Adding simulated room impulse responses

We apply the simulated room impulse responses (RIRs) [9] to corrupt the original audio by convolving with simulated RIRs. The simulated room impulse responses include small and medium room size. The ranges from which the width and length of a room are uniformly sampled are 1m-10m and 10m-30m, respectively.

2.3.3 Speed perturbation

Speed perturbation method is applied to create two copies of the original signal with speed factors of 0.9 and 1.1 [16]. The speed function of the SoX toolkit [17] is used to modify the speed to 90% and 110% of the original rate.

Augmenting the original audio with the corrupted copies produces 441,357 utterances for SRE and Mixer6 datasets, 197,267 utterances for the SWBD dataset, 163,744 utterances for the Fisher dataset, and 8,718,675 utterances for the VoxCeleb dataset. After data augmentation, there are approximately 9,521,000 utterances available for training. To process such a large amount of data, we throw away the speakers with fewer than 8 utterances and remove features that are too short after removing silence frames. We require at least 400 frames per utterance for training. For speaker-embedding L-vector extraction, a neural network of 6 hidden layers with rectified linear unit (ReLU) non-linearity is trained to discriminate over 27,000 speakers in the training set with over 9,000,000 segments.

3. EXPERIMENTS

The proposed L-vectors are evaluated with NIST SRE 2016 and 2018 speaker detection tasks. In speaker detection, it is reasonable to assume the target ratio to be small. In verification, the target ratio is often high. In detection, the target ratio is often much smaller. The performance metrics are the equal error rate (EER) and the minimum of the detection cost function (DCF) at the target ratio of 0.01 and 0.005, per the standard in the NIST-SRE 2016 and 2018 evaluation plan.

A classifier based on probabilistic linear discriminative analysis (PLDA) is used for L-vector and X-vector systems. The L-vectors and X-vectors are centered, and then projected to 150 dimensionality using LDA. The LDA and PLDA are trained using the SRE data with data augmentation. In addition, the length normalization and PLDA are applied to L-vectors and X-vectors. Both L-

vector and X-vector were implemented using the open-source Kaldi Speech Recognition Toolkit [18].

3.1. NIST SRE 2018

In NIST SRE 2018 [19], two types of training conditions, fixed and open, are defined with different restriction of data/resources usage. We focus on the fixed condition tasks in this work. Most of the training data are English. However, the main part of the 2018 speaker recognition evaluation data is the Call My Net 2 (CMN2) spoken in Tunisian Arabic. The other part of evaluation data, Video Annotation for Speech Technology (VAST), is extracted from YouTube videos and spoken in English. Results of the SRE 2018 are shown in Table 2. The best result was boldface. In total, we have 12 system results by using MFCC, PLP and FBP features combined with the first and second segment-level layers of speaker-embedding neural networks for L-vectors or X-vectors. We can see that the proposed L-vector with FBP feature achieves the best actual DCF of 0.400.

The CMN2 part of SRE 2018 is in Arabic. Because the training data is essentially all in English, the English (speaker) PLDA can be treated as out-of-domain PLDA. The SRE 2018 unlabeled data is used to adapt the out-of-domain PLDA. The adapted PLDA can be treated as Arabic (speaker) PLDA, because the SRE 2018 unlabeled data is Arabic. Since there is no VAST in the SRE 2018 unlabeled data, we do not adapt PLDA in the VAST part of SRE 2018.

The VAST audio data in SRE 2018 is in English, with 44.1 kHz and in 16-bit format. Most of the available training data on SRE 2018 fixed condition are 8 kHz narrow-band speech data. For speaker detection on VAST data, we train 16 kHz L-vector and X-vector systems in addition to 8 kHz systems. The VAST data is down-sampled to 16 kHz. The SRE data is up-sampled to 16 kHz and combined with 16 kHz VoxCeleb to build speaker-embedding systems based on 16 kHz data. Based on 16 kHz and 8 kHz systems, we apply the average score combination of PLDA classifiers to generate the VAST results.

We apply 10-fold cross-validation to the CMN2 part of SRE 2018 development data for fusion parameters. On the CMN2 part (2,063,007 trials) of SRE 2018 evaluation data, the system achieves minimum DCF of 0.392 and actual DCF of 0.393.

3.2. NIST SRE 2016

Results of the SRE 2016 evaluation set [20] are shown in Table 3. We can see that FBP is consistently the best feature compared with MFCC and PLP. The proposed L-vector with FBP feature achieves the lowest EER of 6.99%. The X-vector achieves the lowest minimum DCF of 0.505.

The default X-vectors [9] are extracted at TDNN layer 6 which is comparable to the proposed L-vectors of TDNN layer 5. In Table 2 and Table 3, we found the proposed L-vectors are better than the X-vectors in both EER and DCF.

In addition, Snyder et al. suggested [6] the combined embedding of the first and second segment-level layers is better than the first or second segment-level layers. To keep the variation and achieve the better result, we simply applied the average score combination on both 5th and 6th layers speaker embedding of L-vectors, and 6th and 7th layers speaker embedding of X-vectors to evaluate NIST SRE 2016. We obtain an EER of 5.56% and minimum DCF of 0.423 in NIST SRE 2016 evaluation set by using the average score fusion.

3.2. Computational Resources

The experiments in this work have been implemented with machines equipped with Intel i7-8700 CPU with 32GB DDR4-2666 RAM and GeForce GTX 1080 Ti. For representation learning, it takes approximately two weeks to finish training L-vector or X-vector speaker-embedding models. For inference, a randomly selected trial with 132-second enrollment data and 84-second test segment takes a single system 32.85 seconds to process, including audio feature generation, speaker-embedding L-vector or X-vector extraction, and LDA and PLDA scoring.

4. CONCLUSION

In this study, we propose a speaker-embedding model called L-vector based on TDNN-LSTM neural networks. We investigate the MFCC, PLP, and FBP front-end features, with FBP showing the best performance. We use training datasets (Fisher + Mixer6 + SRE + SWBD + VoxCeleb) and apply three data augmentation methods to increase the amount and diversity of available training data, including noise addition, convolution, and speed perturbation. We evaluate the proposed L-vectors with NIST SRE 2016 and SRE 2018 speaker detection tasks. The best single system is the proposed L-vector with FBP features, achieving an actual DCF of 0.400 in the NIST SRE 2018 development set. We adopt the average score combination of 8 kHz and 16 kHz models for the 44.1 kHz YouTube speech of VAST. We use the average score combination of out-of-domain PLDA and in-domain adapted PLDA models to recognize the Arabic speech of CMN2. By using the average score combination of 12 systems (including L-vectors and X-vectors), we achieve an EER of 5.56% and a minimum DCF of 0.423 in NIST SRE 2016 evaluation set.

ACKNOWLEDGEMENT

We thank the Ministry of Science and Technology, Taiwan ROC, under the Project number of 107-2218-E-110-011 and 108-2634-F-008-004, for funding this work.

REFERENCES

[1] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *IEEE Signal Processing Letters*, vol. 13, pp. 308-311, 2006.

[2] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-End Factor Analysis For Speaker Verification," *IEEE Trans. Audio, Speech and Language Processing*, vol. 19, no. 99, pp. 788-798, 2010.

[3] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A Novel Scheme for Speaker Recognition Using a Phonetically-Aware Deep Neural Network", in *Proc. ICASSP* 2014.

[4] J.-C. Wang, Y.-H. Chin, C.-L. Huang, K.-Y. Wang, and C.-H. Wu, "Speaker Identification Using Discriminative Features and Sparse Representation," *IEEE Trans. on Information Forensics & Security*, vol. 12, no. 8, pp. 1979-1987, 2017.

[5] D. Snyder, P. Ghahremani, D. Povey, D. Garcia-Romero, Y. Carmiel, and S. Khudanpur, "Deep Neural Network-based Speaker Embeddings for End-to-end Speaker Verification," *IEEE Spoken Language Workshop (SLT),* 2016.

[6] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep Neural Network Embeddings for Text-independent Speaker Verification," in *Proc. Interspeech,* 2017.

[7] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: a large-scale speaker identification dataset," *in Proc. Interspeech*, 2017.

[8] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep Speaker Recognition," in *Proc. Interspeech*, 2018.

[9] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN Embeddings for Speaker Recognition," in *Proc. ICASSP*, 2018.

[10] V. Peddinti, D. Povey, and S. Khudanpur, "A Time Delay Neural Network Architecture for Efficient Modeling of long temporal contexts," in *Proc. Interspeech*, 2015.

[11] Y. Zhu, T. Ko, D. Snyder, B. Mak, and D. Povey, "Self-attentive Speaker Embeddings for Text-independent Speaker Verification," in *Proc. Interspeech*, 2018.

[12] C.-L. Huang, H. Su, B. Ma, and H. Li, "Speaker Characterization Using Long-Term and Temporal Information," in *Proc. Interspeech*, 2010.

[13] C.-L. Huang, C. Hori, H. Kashioka, and B. Ma, "Ensemble Classifiers Using Unsupervised Data Selection for Speaker Recognition," in *Proc. Interspeech*, 2012.

[14] D. Snyder, D. Garcia-Romero, A. McCree, G. Sell, D. Povey, and S. Khudanpur, "Spoken Language Recognition using X-vectors," in *Proc. Odyssey*, 2018.

[15] D. Snyder, G. Chen, and D. Povey, "Musan: A music, speech, and noise corpus," arXiv preprint, 2015.

[16] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio Augmentation for Speech Recognition," in *Proc. Interspeech*, 2015.

[17] SoX, Online: http://sox.sourceforge.net/

[18] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi Speech Recognition Toolkit," in *Proc. ASRU*, 2011.

[19] NIST, "NIST 2018 Speaker Recognition Evaluation Plan," 2018. https://www.nist.gov/document/sre18-evaluation-plan

[20] NIST, "NIST 2016 Speaker Recognition Evaluation Plan," 2016.