THE NSYSU-PINGAN SYSTEM DESCRIPTION TO THE NIST SRE 2019

Po-Chin Wang, Chia-Ping Chen, and Chien-Lin Huang

National Sun Yat-Sen University, Taiwan and PAII Inc., USA
M073040069@student.nsysu.edu.tw, cpchen@mail.cse.nsysu.edu.tw, chiccocl@gmail.com

ABSTRACT

This paper briefly describes the NSYSU-PingAn system submitted to the NIST SRE 2019 CTS Challenge (National Institute of Standards and Technology Speaker Recognition Evaluation Conversational Telephone Speech Challenge).

Index Terms**—** NIST SRE 2019, speaker embedding, X-vector, L-vector, A-vector, E-TDNN X-vector

1. INTRODUCTION

NIST SRE 2019 is an ongoing speaker recognition evaluation conducted by the US National Institute of Standards and Technology since 1996. In every NIST SRE evaluations, NIST designs different challenges such as multilingual, multichannel, noisy speech. The evaluations encourage the research community to explore promising new idea in speaker recognition.

2. THE PROPOSED SYSTEMS

2.1. Neural Network based Speaker-Embedding

We develope four types of neural-network based speaker embedding methods, namely X-vector, L-vector, A-vector, and E-TDNN X-vector. The X-vector embedding method is based on time-delayed neural network (TDNN) [1]. The L-vector embedding method combines the TDNN and long short-term memory (LSTM) recurrent neural network (RNN) structure, which is called TDNN-LSTM [2, 3]. The A-vector embedding method incorporates the TDNN-LSTM and attention mechanism in the neural network model. The motivation of using both TDNN and LSTM in L-vector is to better capture the temporal information in speech than using TDNN alone as in the X-vector. In A-vector, an attention layer, instead of a statistical pooling layer, is used to capture information from the outputs of all frames over each speech segment. We also increase the depth of TDNN in the X-vector embedding method to an E-TDNN embedding method. Because E-TDNN has more layers, it can get a wider temporal context, which will help us extract better quality embedding. The E-TDNN architecture we use has a total of 10 frame level layers, which is twice as many as the original TDNN. On the classification final layer, we also replace the traditional Softmax function with Additive Margin Softmax function. Gaussian Error Linear Unit (GELU) is also used as an activation function of the embedding layer to improve performance. GELU implements a probabilistic view of a neuron's output, and outperforms traditional RELUs and ELUs in some speech tasks.

Most systems are trained on Fisher, Mixer6, NIST SRE, Switchboard (SWBD), VoxCeleb1 and VoxCeleb2 [4, 5], except that the E-TDNN X-vector systems using the E-TDNN architecture only trained a subset of above datasets as a result of the consideration of computing resources. Because the CTS is 8 kHz telephone speech, all the audio samples are down-sampled to 8 kHz and in 16-bit format using sox for feature extraction.

2.2. Front-end 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 [6, 7]. 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 [2, 8, 9, 10]. Because the neural network based speaker embedding is a data greedy approach, the different data augmentation methods are used to create 4~10 copies of the available training data, including adding room impulse responses, adding speed perturbation, adding volume perturbation, adding additive noises, babble, music, and so on. After data augmentation, 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.

2.4. Back-end Scoring

A classifier based on probabilistic linear discriminative analysis (PLDA) is used for our speaker embedding systems. All systems are centered, and then projected to the specific dimensionality using LDA [11]. In most systems, we use LDA projection to 150 dimensions, where E-TDNN X-vector projects to 250 or 450 dimensions based on our experience. In addition, the length normalization and PLDA are applied to X-vector, L-vector, A-vector, and E-TDNN X-vector. The LDA and PLDA are trained using the SRE , SWBD data or VoxCelebCat which represents the VoxCeleb data after concatenated the segment of the same original video into a longer segment. These data are all with data augmentation.

The test data of CTS is Arabic in SRE 2019. 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.

2.5. Score Fusion and Calibration

In the NSYSU-PingAn submission, there are total 33 subsystems by using different speaker embedding (X-vector, L-vector, A-vector, and E-TDNN X-vector), front-end feature analysis (MFCC, PLP, and FBP), and back-end scoring (PLDA and adapted PLDA). The calibration and fusion of 33 subsystems were done using BOSARIS toolkit [12]. The SRE 2018 evaluation dataset is used for system calibration.

2.6. Computational Resources

The experiments are run on machines of NVIDIA DGX station equipped with Intel Xeon E5-2698 CPU 2.2 GHz, 256 GB RDIMM DDR4 and Tesla V100 GPU.

3. EXPERIMENTS

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 2019 evaluation plan [13, 14]. Table 1 showed the score fusion results on NIST SRE 2018 and SRE 2019 evaluation datasets.

Table 1: Results on NIST SRE 2018 and SRE 2019 evaluation datasets

dataset	EER(%)	min_C	act_C
SRE 2018 Evaluation	5.78	0.349	-
SRE 2019 Evaluation	5.54	0.356	0.359

Due the data source of SRE 2018 evaluation and SRE 2019 are similar, the performance gain between them is small. We achieved an EER of 5.78%, a minimum DCF of 0.349 in the NIST SRE 2018 evaluation dataset with 2,063,007 trials. We achieved an EER of 5.54%, a minimum DCF of 0.356, an actual DCF of 0.359 in the NIST SRE 2019 evaluation dataset with 2,688,376 trials.

4. CONCLUSION

In this study, we explore different neural network speaker-embedding methods of X-vector, L-vector, and A-vector. For the front-end feature analysis, we use the acoustic features of MFCC, PLP, and FBP. For the back-end score fusion and calibration, the results of PLDA and adapted PLDA are both considered. We also investigate the difference between more layers of TDNN compared to traditional TDNN, and the impact of Additive Margin Softmax (AM-Softmax) and Gaussian Error Linear Unit (GELU) for the neural network functions used in the neural-network based embedding models.

REFERENCES

[1] D. Snyder, D. Garcia-Romero, G. Sell, A. McCree, and D. Povey, Sanjeev Khudanpur, “Speaker Recognition for Multi-Speaker Conversations Using X-vectors,” in *Proc. ICASSP*, 2019.

[2] C.-L. Huang, “Exploring Effective Data Augmentation with TDNN-LSTM Neural Network Embedding for Speaker Recognition,” in *Proc. ASRU*, 2019.

[3] C.-P. Chen, S.-Y. Zhang, C.-T. Yeh, J.-C. Wang, T. Wang, and C.-L. Huang, “Speaker Characterization Using TDNN-LSTM based Speaker Embedding,” in *Proc. ICASSP*, 2019.

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

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

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

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

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

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

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

[11] 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.

[12] Niko Bru¨mmer and Edward De Villiers, “The Bosaris Toolkit: Theory, Algorithms and Code for Surviving the New DCF,” arXiv preprint arXiv:1304.2865, 2013.

[13] NIST, “NIST 2018 Speaker Recognition Evaluation Plan,” 2018.

[14] NIST, “NIST 2019 Speaker Recognition Evaluation: CTS Challenge,” 2019.