FFSVC2020 Challenge TASK2 : x-vector based solution

Ruijie Xu

University of Chinese Academy of Science

xuruijie19@mails.ucas.ac.cn

Abstract

In this report, we present the x-vector based systems for the Interspeech 2020 Far-Field Speaker Verification Challenge (FFSVC) Task2: far-field text-independent speaker verification using a single microphone array. The system consists with three part: data augmentation, network description and score fusion. Besides the provided FFSVC2020 training set, more than 10000 speakers external open-source datasets are used to increase the speaker diversity for a robust systems. Traditional time delay neural network system (TDNN) and extended-TDNN system are adapted in this report. Score-normalization and score fusion are further adopted in the tasks to improve the performance. Finally, minDCF of 0.5389 and EER 4.88% on development set are obtained.

Index Terms: speaker verification, x-vector, data augmentation

1. Introduction

Speaker verification is to verify the identity of target speakers. Near-end speaker verification system have achieved large progress. Due to the complexity and varied acoustic environment, far-field speaker verification face more challenges[1]. The TASK2 of the interspeech 2020 far-field speaker verification challenge (FFSVC) is focus on the problem of far-field text-independent speaker verification from single microphone array. The recording devices include one close-talking microphone, one iPhone at 25cm distance and 6 circular microphone arrays. The training data, the development data and the evaluation data have 120 speakers, 35 speakers, and 80 speakers, respectively.

Due to the few speakers of the training data, the open-access database shared on openslr.org before Feb 1st can be used as external datasets adding to the training sets. The channel mismatch exist in the in-domain FFSVC2020 sets and the out-domain open-access datasets. In order to reduce such mismatch, data augmentation are used in this report.

The network used in this task are x-vector based [2], including TDNN system and Extend-TDNN system. Finally, score normalization [3] and score fusion are used to reduce the imbalance of the different datasets and different the systems.

The report is organized as follows. Technical solution and experimental result are present in Section 2. Section 3 present the conclusions of this report.

2. Technical solution

This section describes our data augmentation, network description and experimental results.

2.1. Data augmentation

The training set consists of the FFSVC2020 sets and the open-access datasets. We devide the training set into two part : near-filed dataset and the far-field dataset. For the near-field dataset, we take two data augmentation policies. The first data augmentation method follow the KALDI recipe, which contains adding the additive noise and the convolution noise. The second data augmentation method is using the Pyroomacoustics toolkit [4] to generate simulated room impluse response (RIR). Then the near-field signal Convolve with the simulated. For the far-field dataset, weighted prediction error and beamforming method are used to reduce the reverberation and the environment noise. The specAugment proposed in [5] were used to do the frequency and time mask on the training set.

The preprocessed data will resample to 16000 Hz, and log Mel filter-banks with 40-dimensions features are adopeted. All the features are extracted every 10ms with a 25ms window. Then the cepstral mean-normalization (CMV) with a sliding window of 3s are performed on these features.

Table 1: *Extended TDNN Framework*

Num	Layer name	Layer Context	Size
1	TDNN-ReLU-BN	t-2:t+2	1024
2	Dense-ReLU-BN	t	1024
3	TDNN-ReLU-BN	t-2,t,t+2	1024
4	Dense-ReLU-BN	t	1024
5	TDNN-ReLU-BN	t-3,t,t+3	1024
6	Dense-ReLU-BN	t	1024
7	TDNN-ReLU-BN	t-4,t,t+4	1024
8	Dense-ReLU-BNN	t	1024
9	Dense-ReLU-BNN	t	1500
10	Pooling	Full Seq.	3000
11	Dense-ReLU-BNN	[0, T]	512
12	Dense-ReLU-BNN	[0, T]	512
13	Softmax	[0, T]	Num.Spks

Table 2: *TDNN Framework*

Num	Layer name	Layer Context	Size
1	TDNN-ReLU-BN	t-2,t-1,t,t+1,t+2	1024
2	TDNN-ReLU-BN	t-2,t,t+2	1024
3	TDNN-ReLU-BN	t-3,t,t+3	1024
4	LSTMP	t	1024
5	TDNN-ReLU-BN	t	1024
6	LSTMP	t	1024
7	TDNN-ReLU-BN	t	1500
8	Pooling	Full Seq.	3000
9	Dense-ReLU-BNN	[0, T]	512
10	Dense-ReLU-BNN	[0, T]	512
11	Softmax	[0, T]	Num.Spks

2.2. Network architectures

Table 1 present the detail of the ETDNN system. It consist three part: the frame-level, the pooling layer, and the segment-level.

Table 3: *Evaluation results on the development set*

ID	System	Cosine		Cosine(AS-norm)(dev)		Cosine(AS-norm)(eval)	
		minDCF	EER	minDCF	EER	minDCF	EER
1	Baseline System	0.5800	5.83	-	-	0.66	6.55%
2	TDNN	0.5914	5.81%	0.5652	5.26%	-	-
3	ETDNN	0.5733	5.67%	0.5511	5.17%	-	-
4	Fusion 2 &3	-	-	0.5389	4.88%	0.6087	5.64%

Then a softmax layer followed by the segment-level mapping the hidden nodes to the number of speakers. The output node in frame-level is 1024, the mean and Standard deviation with 1500 dimension used mapping the frame-level features to the segment-level feature.

The TDNN framework are shown in Table 2. We have a little changes compared to the traditional TDNN-based system. In the last 4 layer of the frame-level of the TDNN framework, we use a LSTMP layer followed by a TDNN-ReLU-BN layer to replace the single TDNN-ReLU-BN layer. The output of the frame-level is 1024.

2.3. Adaptive score normalization and Fusion

In this report, We adopt the Cosine similarity as the back-end scoring method. The Adaptive score normalization with cohort up to 200 files are used in this report to normalize the Cosine scores. The normalized scores are then equally averaged.

2.4. Experimental result

We evaluate our systems on the development sets provided by FFSVC2020. The results shown in Table 3 present the scores with and without the AS-norm process. It is clearly that the performance improved using the AS-norm post-process. The final result on the development achieves minDCF of 0.5389 and EER of 4.88 using the average method.

3. Conclusions

This paper describes the x-vector based system, including TDNN and ETDNN architecture, on the far-field datasets. Both the TDNN system and the ETDNN system get performance improvement compared to the baseline system.

4. References

[1] X. Qin, D. Cai, and M. Li, "Far-field end-to-end text-dependent speaker verification based on mixed training data with transfer learning and enrollment data augmentation," *Proc. Interspeech 2019*, pp. 4045–4049, 2019.

[2] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification." in *Interspeech*, 2017, pp. 999–1003.

[3] P. Matejka, O. Novotnỳ, O. Plchot, L. Burget, M. D. Sánchez, and J. Cernockỳ, "Analysis of score normalization in multilingual speaker recognition." in *Interspeech*, 2017, pp. 1567–1571.

[4] R. Scheibler, E. Bezzam, and I. Dokmanić, "Pyroomacoustics: A python package for audio room simulation and array processing algorithms," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 351–355.

[5] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.