The JD AI Speaker Verification System for the FFSVC 2020 Challenge

Ying Tong, Wei Xue, Shanluo Huang, Lu Fan, Chao Zhang, Guohong Ding, Xiaodong He

JD AI Research

{tongying,xuewei27,huangshanluo,fanlu,chao.zhang,dingguohong,xiaodong.he}@jd.com

Abstract

In this report, we present the development of our systems for the Interspeech 2020 Far-Field Speaker Verification Challenge (FFSVC): far-field text-independent speaker verification using a single microphone array. We propose an entire technical solution, which contains the data augmentation, network structure, score normalization and system fusion. Both far-to-near (dereverberation) and near-to-far (reverberation) transformations are used as augmentation methods to expand the diversity of the dataset. Then different model structures with different pooling layer are investigated in this report. Probabilistic linear discriminant analysis (PLDA) and cosine similarity are simultaneously used as back-end scoring method for each systems. At last, a two-stage fusion method are applied to the adaptively normalized scores, which achieves a minimum of the detection cost function (minDCF) of 0.3407 and 0.4464 on the development set and the evaluation set of the challenge, respectively.

Index Terms: speaker verification, deep neural network, data augmentation, score normalization

1. Introduction

The goal of the Interspeech 2020 Far-Field Speaker Verification Challenge (FFSVC) is launched to facilitate the study on both far-field text-dependent and text-independent speaker verification [1–4] problems. In this paper, we describe our systems and the experimental results on FFSVC task 2, the text-independent speaker verification with a single microphone array. It is an open-track task since external open-access datasets are allowed to be used along with the officially-released 1,100-hour far-field training data (denoted as FFSVC20). Since the external datasets are recorded with different acoustic environments, in this report we apply various data augmentation method to improve the far-field speaker recognition performance.

We comprehensively describe the use of near-field to far-field transformation (near-to-far) and its reverse based on signal processing, as well as different data augmentation methods related to additive/convolutional noises and room impulse responses (RIRs). Different DNN structures and fusion methods are also investigated [5].

Both far-to-near (dereverberation) and near-to-far (reverberation) transformations are used for FFSVC and the external datasets to capture better channel-invariant for speaker characteristics. For FFSVC datasets, Weighted prediction error (WPE), beamforming, and Voice channel switching methods are used to add the channel invariant to the far-field signals. For the external datasets, data augmentation by transforming the near-field data to far-field is used to increase channel and acoustic environment information for speaker characteristics to fit the characteristics of FFSVC. The near-to-far data augmentation is implemented using a linear convolution using the RIRs estimated for each pair of near-field and far-field microphones in the FFSVC dataset.

All of our systems are deep-learning-based. Three different structures are used as encoder, namely ResNet [6], extend time-delay neural network (ETDNN) [7, 8] , and factorized TDNN (FTDNN) encoder. For each type of model, the output values are pooled across time using the multi-head self-attentive structure [9, 10] or the statistic pooling structure [11]. The angular softmax function is introduced in this work to increase the discrimination between the speakers and decrease the distance of the intra-speakers. Back-end scoring is performed using probabilistic linear discriminant analysis (PLDA) [12] and cosine similarity [13]. Adaptive score normalization [14] and BOSARIS toolkit [15] are used for the post-process in the end.

The remainder of the paper is organized as follows. Section 2 describes the data preparation pipeline. Section 3 presents the structures of the systems. Experimental results are presented in Section 4, followed by conclusions.

2. Data Augmentation

This section describes our data preparation pipelines for FFSVC20 dataset and other external open-access datasets.

2.1. Pipeline for FFSVC20

The FFSVC20 dataset was collected in multiple scenarios. The training set includes 120 speakers with a total number of 1,100 hour speech, and the development set consists of data from 35 speakers. The FFSVC20 provides a close-talking microphone, an iPhone at 25 cm distance, and three randomly selected 4-channel microphone arrays in the training and develepment sets. The following four data augmentation methods are used, which can help the DNN speaker classifier to capture better channel- and acoustic-environment-invariant features for speaker discrimination.

- **WPE**: The NARA-WPE toolbox [16] is performed to dereverberation operation of the far-field recordings.
- **Beamforming**: The BeamformIt toolbox [17] is used to perform beamforming, which takes an arbitrary number of input channels without any prior information and computes an output by filter-and-sum beamforming.
- **Voice channel switching**: The voice channel switching method [18] is used to combine signals form different distance with different channel to increase spatial information.
- **PyRIR**: The pyroomacoustics toolkit [19] is used to generate multi simulated multi-channel room impluse response.

The authors would like to thank Shuai Wang and Yi Liu for useful discussions and suggestions.

Table 1: *Detailed specifications of the different ResNet architectures.*

Layer name	**ResNet-18**	**ResNet-34**	**ResNet-50**
Input	-	-	-
Conv2D-1	3×3, Stride 1	3×3, Stride 1	3×3, Stride 1
ResNetBlock-1	$\begin{bmatrix} 3\times3 & 64 \\ 3\times3 & 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3 & 64 \\ 3\times3 & 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1 & 64 \\ 3\times3 & 64 \\ 1\times1 & 256 \end{bmatrix} \times 3$
ResNetBlock-2	$\begin{bmatrix} 3\times3 & 128 \\ 3\times3 & 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3 & 128 \\ 3\times3 & 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1 & 128 \\ 3\times3 & 128 \\ 1\times1 & 512 \end{bmatrix} \times 4$
ResNetBlock-3	$\begin{bmatrix} 3\times3 & 256 \\ 3\times3 & 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3 & 256 \\ 3\times3 & 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1 & 256 \\ 3\times3 & 256 \\ 1\times1 & 1024 \end{bmatrix} \times 6$
ResNetBlock-4	$\begin{bmatrix} 3\times3 & 512 \\ 3\times3 & 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3 & 512 \\ 3\times3 & 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1 & 512 \\ 3\times3 & 512 \\ 1\times1 & 2048 \end{bmatrix} \times 3$
Conv2D	1× K, Stride 1		
Fully Connected Layer	512× 512 (Input × output)		
Fully Connected Layer	512× 1500 (Input × output)		
Self-attentive Pooling Layer	1500× 3000 (Input × output)		
Fully Connected Layer	3000× 512 (Input × output)		
Fully Connected Layer	512× 512 (Input × output)		
ArcSoftmax	512× N (Input × output)		

Table 2: *Detailed specifications of the FTDNN system.*

Num	Layer	Context Factor 1	Context Factor 2	Skip Conn. from Layer
1	TDNN	t-2:t+2		
2	FTDNN	t-2,t	t,t+2	
3	FTDNN	t	t	
4	FTDNN	t-3,t	t,t+3	
5	FTDNN	t	t	3
6	FTDNN	t-3,t	t,t+3	
7	FTDNN	t-3,t	t,t+3	2,4
8	FTDNN	t-3,t	t,t+3	4,6,8
9	FTDNN	t	t	
10	None/Lstm	t	t	
11	Dense	t	t	
12	Pooling	Full Seq.		
13	Dense	[0, T]		
14	Dense	[0, T]		
15	Softmax	[0, T]		

2.2. Pipeline for external open-access datasets

In this paper, two open-access speaker recognition datasets, CHData [1] and VoxCeleb2 [20], are used as the external open-access datasets to construct the systems. For CHData, the subsets SLR$\{18, 33, 47, 62, 68, 85\}$ are selected to use, which consists of a total number of 2897 hours of speech 5126 speakers. For VoxCeleb2, a total number of 5994 speakers are used. We have noticed that the SLR85 subset of CHData is recorded similarly to FFSVC20, and the other CHData subsets are the near-field data. Together with the PyRIR described in section 2.1, the SIAug and SREAug are applied to the open-access datasets.

- **SIAug**: For each pair of the near-field/far-field signals in the FFSVC20 dataset, a large collection of RIRs are estimated by performing system identification (SI) [21] from the near-field source signal to the signal captured by the far-field microphone array. All near-field training data are augmented by convolving the near-field signals with a randomly selected RIR.
- **SREAug**: The SREAug pipeline from the x-vector based speaker recognition system in the Kaldi SRE16 recipe [22] is used to increase the diversity of noise interferences and RIRs in the dataset. The SREAug contains the following steps: (a) Mixing with babel, music and noise signals from the MUSAN corpus [23]; (b) Convolving with the RIRs from the AIR dataset [23].

[1] https://openslr.org/resources.php

2.3. Data preprocess

The full training sets are compose with the official FFSVC20 dataset, the external datasets and its augmented versions. Three kinds of acoustic features including 60-dimension log-Mel filter-banks (FBK) +Pitch (FBKP), 80-dimension FBK+Pitch and 30-dimension FBK+Pitch are employed in this task. Audios are resampled to 16 kHz, and all the features are extracted from the raw signals with 25 ms frame length and 10 ms overlap. The energy-based voice activity detection (VAD) from Kaldi SRE16 recipe is used to select the speech period. Then the features are processed through local Cepstral Mean Normalization (CMN) over a 3-second sliding window before fed into the deep speaker network.

3. Model Architectures

In this section, we introduce three types of DNN architectures and the score normalization used by our system.

3.1. DNN-based systems

All of our systems are deep speaker embeddings-based, which accept variable-length segments and produce an utterance-level score. The ETDNN, ResNet and FTDNN based systems are

Table 3: *Performance comparison using different training model*

ID	System	PLDA (Dev)		Cosine (Dev)		Fusion (Dev)		Fusion (Eval)	
		minDCF	EER(%)	minDCF	EER(%)	minDCF	EER(%)	minDCF	EER(%)
1	FFSVC20 baseline system[24]	-	-	0.5800	5.83	-	-	0.66	6.55
2	Res18-att-FBKP60	0.5732	4.97	0.4948	3.95	0.4575	3.56	-	-
3	Res34-att-FBKP60	0.5338	4.44	0.4664	3.44	0.429	3.1	-	-
4	Res34-stat-FBKP60	0.5273	4.49	0.4487	3.42	0.4131	3.22	-	-
5	Res50-att-FBKP60	0.5851	5.03	0.558	4.5	0.5027	4.03	-	-
5	ThinRes34-GVLAD-FBKP60	0.6906	6.19	0.6068	5.09	0.557	4.71	-	-
6	Etdnn-stat-FBKP60	0.6049	5.73	0.555	4.77	0.4975	4.44	-	-
7	Etdnnf-att-FBKP60	0.6591	6.05	0.5061	4.2	0.4849	3.97	-	-
8	FTDNN-LSTM1-sta-FBKP60	0.5927	5.46	0.5442	4.56	0.491	4.26	-	-
9	FTDNN-LSTM2-sta-FBKP60	0.5513	5.05	0.4895	3.75	0.4482	3.63	-	-
10	Res34-att-FBKP80	0.5357	4.5	0.4601	3.6	0.4293	3.18	-	-
11	Etdnn-stat-FBKP80	0.6102	5.01	0.551	4.57	0.4988	4.49	-	-
12	Etdnn-stat-PLPP30	0.6449	6.11	0.5706	4.78	0.5394	4.56	-	-
13	Res34-att-FBKP60-WPEBF	0.5241	4.27	0.4419	3.32	0.4142	2.99	-	-
14	Res34-stat-FBKP60-WPEBF	0.5315	4.42	0.4349	3.32	0.4080	3.08		
15	fusion	-	-	-	-	**0.3407**	**2.67**	**0.4464**	**3.61**

developed, and the main differences of these systems are in the encoder part. The details will be described as follows.

3.1.1. ETDNN-based systems

We use a bigger network with more neurons in extended-TDNN layers, namely BETDNN. The detailed description of the network is summarized in [8]. The first 10 layers of the x-vector system operate on the frame level, with a small temporal context window centred at the current frame t, followed by a self-attentive pooling layer. Then the segment-level statistics are concatenated and passed through the segment-level layers.

3.1.2. ResNet-based systems

Table 1 summarizes the adopted ResNet-based network architecture. The differences among ResNet-18/ResNet-34/ResNet-50 are in the depth and structure of the residual layer. The ArcSoftmax loss [25] was utilized to further increase the distance between the speakers while retaining a small intra-speaker distance. Besides the ResNet-based network shown in Table 1, we also take experiment on Thin-ResNet-34 with GhostVLAD introduces in [26].

3.1.3. FTDNN-based systems

The detail description about the FTDNN xvector architecture are summarized in Table 2. Three kinds of FTDNN models are experiment in this task.

- **FTDNN-LSTM1**: This system is shown in Table 2 with statistic pooling. The output size of each layer is 512, and the inner size of the FTDNN layer is 128. Two LSTM layers with 512-dim cell, 256-dim recurrent and non-recurrent projection units, are added after the FTDNN layer.
- **FTDNN-LSTM2**: This system have similar stucture with the FTDNN-LSTM1 system, except that the output size of each layer in the frame layer is 1024, and the inner size of FTDNN layer is 256.
- **EFTDNN**: The extended FTDNN introduced in [27] is a combination of ETDNN and FTDNN structure. Angular softmax loss is used in this system.

3.2. Adaptive score normalization

In the adaptive score normalization, only top X closest files are selected as the cohort to compute mean and variance for normalization. In this paper, we use the top 200 files.

4. Experimental Results

We evaluate our technical solution on the FFSVC development sets Results are reported in terms of the primary evaluation metric used by FFSVC20, which are the minDCF with $P_{target} = 0.01$ and EER. During testing, the scores of different channels and their augmentations in the same microphone array are equally weighted. The results shown in Table 3 are processed by the Adaptive S-norm method. We generate larger training sets with 11240 speakers using the above mentioned data augmentation methods. Table 3 shows the results by using different network with different input features. As shown in Table 3, the listed systems outperforms the baseline system after using the first stage fusion. Further performance improvement are obtained using BOSARIS toolkit to fusion more systems.

5. Conclusions

This paper describes the development of the JD AI speaker verification system for task 2 of FFSVC20. We experiment various augmentation methods and various network in this report. The score normalization and two-stage score fusion method achieve promising performance of minDCF 0.3407 and EER 2.67% on development sets and minDCF 0.4464 and EER 3.61% on evaluation sets.

6. References

[1] X. Qin, M. Li, H. Bu, R. K. Das, W. Rao, S. Narayanan, and H. Li, "The FFSVC 2020 evaluation plan," *arXiv preprint arXiv:2002.00387*, 2020.

[2] 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*, pp. 4045–4049, 2019.

[3] S. Wang, J. Rohdin, L. Burget, O. Plchot, Y. Qian, and K. Yu, "On the usage of phonetic information for text-independent speaker

embedding extraction," in *Proc. Interspeech*, 2019, pp. 1148–1152.

[4] L. You, W. Guo, L. Dai, and J. Du, "Multi-task learning with high-order statistics for x-vector based text-independent speaker verification," in *Proc. Interspeech*, 2019, pp. 1158–1162.

[5] A. Kanagasundaram, S. Sridharan, S. Ganapathy, P. Singh, and C. B. Fookes, "A study of x-vector based speaker recognition on short utterances," in *Proc. Interspeech*, 2019, pp. 2943–2947.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016, pp. 770–778.

[7] D. Snyder, D. Garcia-Romero, G. Sell, A. McCree, D. Povey, and S. Khudanpur, "Speaker recognition for multi-speaker conversations using x-vectors," in *Proc. ICASSP*, 2019, pp. 5796–5800.

[8] H. Zeinali, S. Wang, A. Silnova, P. Matějka, and O. Plchot, "BUT system description to VoxCeleb speaker recognition challenge 2019," in *The VoxSRC Workhsop 2019*, 2019.

[9] Y. Zhu, T. Ko, D. Snyder, B. Mak, and D. Povey, "Self-attentive speaker embeddings for text-independent speaker verification." in *Proc. Interspeech*, 2018, pp. 3573–3577.

[10] T. K. Koji Okabe and K. Shinoda, "Attentive statistics pooling for deep speaker embedding," in *Proc. Interspeech*, 2018, pp. 2252–2256.

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

[12] P. Kenny, "Bayesian speaker verification with heavy-tailed priors," in *Proc. Odyssey*, 2010, p. 14.

[13] N. Dehak, R. Dehak, P. Kenny, N. Brümmer, P. Ouellet, and P. Dumouchel, "Support vector machines versus fast scoring in the low-dimensional total variability space for speaker verification," in *Proc. Interspeech*, 2009, pp. 1559–1562.

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

[15] N. Brümmer and E. De Villiers, "The BOSARIS toolkit: Theory, algorithms and code for surviving the new DCF," *arXiv preprint arXiv:1304.2865*, 2013.

[16] L. Drude, J. Heymann, C. Boeddeker, and R. Haeb-Umbach, "NARA-WPE: A python package for weighted prediction error dereverberation in Numpy and Tensorflow for online and offline processing," in *13. ITG Fachtagung Sprachkommunikation (ITG 2018)*, Oct 2018, pp. 1–5.

[17] X. Anguera, C. Wooters, and J. Hernando, "Acoustic beamforming for speaker diarization of meetings," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 7, pp. 2011–2022, 2007.

[18] Q. Wang, H. Wu, Z. Jing, F. Ma, Y. Fang, Y. Wang, T. Chen, J. Pan, J. Du, and C.-H. Lee, "The USTC-iFlytek system for sound event localization and detection of DCASE2020 challenge," DCASE2020 Challenge, Tech. Rep., July 2020.

[19] R. Scheibler, E. Bezzam, and I. Dokmanić, "Pyroomacoustics: A python package for audio room simulation and array processing algorithms," in *Proc. ICASSP*, 2018, pp. 351–355.

[20] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep speaker recognition," in *Proc. Interspeech*, 2018, pp. 1086–1090.

[21] Y. Huang and J. Benesty, "A class of frequency-domain adaptive approaches to blind multichannel identification," *IEEE Transactions on signal processing*, vol. 51, no. 1, pp. 11–24, 2003.

[22] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlíček, Y. Qian, P. Schwarz, J. Silovský, G. Stemmer, and K. Veselý, "The Kaldi speech recognition toolkit," in *Proc. ASRU*, 2011.

[23] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *Proc. ICASSP*, 2017, pp. 5220–5224.

[24] X. Qin, M. Li, H. Bu, W. Rao, R. K. Das, S. Narayanan, and H. Li, "The Interspeech 2020 far-field speaker verification challenge," *http://2020.ffsvc.org/BaselinePaper*, 2020.

[25] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proc. CVPR*, 2019, pp. 4690–4699.

[26] W. Xie, A. Nagrani, J. S. Chung, and A. Zisserman, "Utterance-level aggregation for speaker recognition in the wild," in *Proc. ICASSP*, 2019, pp. 5791–5795.

[27] Y. Liu, T. Liang, C. Xu, X. Zhang, X. Chen, W.-Q. Zhang, L. He, D. Song, R. Li, Y. Wu, P. Ouyang, and S. Yin, "THUEE system description for NIST 2019 SRE CTS challenge," *arXiv preprint arXiv:1912.11585*, 2019.