The Nan7U Speaker Verification System for the FFSVC 2022 Challenge

Ziyang Zhang

zyzhang182@mail.ustc.edu.cn

Abstract

This paper describes the systems developed by the Nan7U team for the 2022 Far-Field Speaker Verification Challenge (FFSVC) Task 1: fully supervised far-field speaker verification. We develop the system for the Fixed Condition on the VoxCeleb and FFSVC dataset. In addition to using the basic ECAPA-TDNN and ResNet34 model, we also used some modified models of ResNet34. The final score is the result of fusing all the above systems.

1. Introduction

The Nan7U system to FFSVC 2022 challenge is based on x-vector/PLDA framework. Six independent systems for the fully supervised far-field speaker verification challenge are developed using the Kaldi [1] and TensorFlow [2] platform. The frame-level encoder of each system use different structures, including ECAPA-TDNN and subsequential variants of ResNet34. The PLDA algorithm is adopted as the backend classifier for all the x-vectors. These sub-systems are fused at the score-level using the BOSARIS Toolkit as the final submitted system.

The remainder of this document is organized as follows. Section 2 presents the data preparation. In section 3, we describe various deep embedding systems. In section 4, we introduce the back-end processing approach. After it, section 5 lists results on development and evaluation set.

2. Data Preparation

2.1. Individual Datasets

Following by the fixed training conditions, we only use the VoxCeleb 1&2 datasets to pre-train the model, and use the FFSVC2020 train/dev/supplementary set to finetune the pretrained model. Note that we down-sample all the training data to 16kHz.

2.2. Data Augmentation

To increase the diversity of the training data, we applied a 5-fold augmentation strategy that combines clean data with 4 copies of the augmented data. Following by the Kaldi recipe, MUSAN [3] and RIRs [4] datatsets are used in our data augmentation process. The following strategies are used for data augmentation.

- Reverb: The speech utterances are artificially reverberated via convolution with simulated RIRs, and we didn’t add any additive noise here.
- Music: A single music file (without vocals) is randomly selected from MUSAN corpus, trimmed or repeated as necessary to match duration, and added to the original signal at 5-15dB SNR.
- Noise: MUSAN noises are added at one second intervals to the original signal at 0-15dB SNR.
- Babble: Three to seven speakers are randomly selected from MUSAN speech, summed together, then added to the original signal at 13-20dB SNR.

2.3. Acoustic Features and VAD

For the acoustic features, all of our systems make use of Fbank features. The 64-dimensional and the 80-dimensional Fbank features are extracted separately, and the frame length is 25ms with 10ms shift. Short-time cepstral mean subtraction is applied over a 3-second sliding window and then energy based VAD is used to remove the non-speech frames. The acoustic features are randomly truncated into short slices ranging from 200 to 400 frames for VoxCeleb data and 180 to 200 frames for FFSVC training data.

In addition, we try to de-reverberate the speech signal using the weighted prediction error (WPE) [5] algorithm, as a front-end signal processing method. However, this method didn’t bring obviously performance improvement.

3. Deep Speaker Embeddings

We will introduce four types of backbone networks that we used in this section. All the models are implemented using the Kaldi and TensorFlow toolkit. It is worth mentioning that all models use a two-step training strategy. The first step is to pre-train the model with VoxCeleb dataset. The second step is to fine-tune the pretrained model with FFSVC training data.

3.1. ECAPA-TDNN network

Considering the excellent performance of ECAPA-TDNN structure on the VoxCeleb dataset, we use it as one of our baseline systems. In standard ECAPA-TDNN structure, the encoder is composed of three SE-Res2Block modules. The channel size of the bottleneck in the SE-Block is set to 1024. Finally, a 192-dimensional speaker embedding is extracted from the fully connected layer. AAM-softmax loss [8] is used and we set margin and scale as 0.25 and 30, respectively. In the pre-training phase, the Adam [9] optimizer is used to optimize the model parameters. While, in the fine-tuning phase, we replace it with SGD optimizer.

3.2. ResNet34 network

We use the classical ResNet34 structure as the main baseline system’s frame-level feature extractor. This structure includes an input convolutional layer and four residual stages. Each residual stage contains a set of residual blocks, where each block is composed of two convolutional layers. The channels of four residual stages are set to 32, 64, 128, 256 respectively.

Then, the statistics pooling is used to convert variable-length frame-level representations into a fixed-length vectors. Two fully-connected layers are used as the utterance-level layers to extract the deep speaker embeddings. The Adam optimizer and AM-softmax [10] are used. In the pre-training phase, the learning rate gradually decreases from 1e-3 to 1e-4. In fine-tuning phase, we set the learning rate to one-tenth of the original and increase the margin from 0.15 to 0.25, keeping the scale still at 30. If not specified, the pre-training and fine-tuning strategies of the following variants of ResNet34 model are consistent with the ResNet34 baseline model.

3.3. ResNet34 network with bidirectional multiscale feature aggregation

In order to exploit the time-frequency context information between different layers in deep neural networks, multiscale feature aggregation (MFA) has been developed extensively. In our model, bidirectional multiscale feature aggregation (BMFA) [11] structure is used in the ResNet34 network. Features of different stages are extracted from the ResNet34 backbone. Then, the information is fused through bidirectional (top-down and bottom-up) pathways. The attention fusion module is also used to learn better fusion weights. It has been proved by relevant experiments that making use of the bidirectional feature aggregation and attentional fusion modules can get more discriminative speaker representation and improve system performance.

3.4. ResNet34 network with global-local information-based dynamic convolution

Another variant of ResNet34 that we used is the global-local information-based dynamic convolution neural (GLIDCNN) network [12]. This structure can better adapt the model to both the inter- and intra-session variations. In this method, the input feature from different time-frequency positions is convolved with dynamically generated kernel coefficients. The generation of these coefficients is related to the global vector of the whole utterance and the local vector around the current input feature. The global features can capture long-range utterance-dependent information including channel and noise types, and the local features can capture the local channel dependencies and reflects the intra-session variation. The GLIDCNN structure can focus on more fine-grained speaker information and its effectiveness has been proven through evaluation on multiple datasets.

4. Back-end processing

4.1. Scoring

For all systems, we compute similarity scores using LDA-PLDA back-end. After extracting speaker embeddings, the vectors are centered using the mean vector computed from the training set and are projected into a low-dimensional space (150 in our setup) with LDA at first. After the length normalization, PLDA algorithm trained on FFSVC training data is adopted as backend classifier for speaker verification. We have tried to use the adaptive score normalization (AS-norm) [13] which computes an average of normalized scores from Z-norm and T-norm. However, we found that it brings effects in the development set, and degrades the performance in the evaluation set. Therefore, we didn't use this method.

4.2. System Fusion

Since we have four types of different model structures, we believe that score fusion can make good use of the complementarity between these different models. Instead of using a simple weighted average method, we choose the BOSARIS Toolkit [14], which was developed to provide solutions for automatic speaker recognition. By using it to train a group of weights for the results of each system in development set, we can use the weights to fuse the scores in evaluation set.

5. Experiment Results

5.1. Results on the FFSVC 2022 Development and Evaluation set

By combining different model structures and features, we developed 6 independent systems finally. The results of each system and the fusion system are shown in Table 1. “Fbank-dim” refers to the dimension of the acoustic features used. “BMFA” represents ResNet34 network with bidirectional multiscale feature aggregation. “GLIDCNN” represents ResNet34 network with global-local information-based dynamic convolution. “Fusion” represents the result of fusing all the listed single systems.

Table 1 The performance of the different systems on Development and Evaluation set.

Model	Fbank-dim	FFSVC 2022 Dev		FFSVC 2022 Eval	
		minDCF	EER(%)	minDCF	EER(%)
ECAPA-TDNN	80	0.6696	7.022	-	-
ResNet34	64	0.6345	6.031	-	-
	80	0.6231	6.322	-	-
BMFA	64	0.5743	5.642	-	-
GLIDCNN	64	0.5833	5.506	-	-
	80	0.5605	5.753	-	-
Fusion	-	0.4806	4.912	0.4820	4.930

The ResNet34 model based on global-local information dynamic convolution using 80-dimensional Fbank features achieves the best results on single system. Score fusion using the BOSARIS Toolkit can further utilize the complementarity between models to improve system performance.

6. References

[1] Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., ... & Silovsky, J. (2011). The Kaldi speech recognition toolkit (No. CONF). IEEE Signal Processing Society.

[2] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard et al., “Tensorflow: A system for large-scale machine learning,” in 12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16), 2016, pp. 265–283.

[3] David Snyder, Guoguo Chen, and Daniel Povey, “MUSAN: A Music, Speech, and Noise Corpus,” arXiv:1510.08484 [cs], Oct. 2015, arXiv: 1510.08484.

[4] Tom Ko, Vijayaditya Peddinti, Daniel Povey, Michael L. Seltzer, and Sanjeev Khudanpur, “A Study on Data Augmentation of Reverberant Speech for Robust Speech Recognition,” in 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Mar. 2017, pp. 5220–5224, IEEE.

[5] T. Nakatani, T. Yoshioka, K. Kinoshita, M.Miyoshi, and B. Juang, "Speech dereverberation based on variance-normalized delayed linear prediction," IEEE Transactions on Audio, Speech, and Language Processing, vol. 18, no. 7, pp. 1717–1731, 2010.

[6] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck, "ECAPA-TDNN: Emphasized Channel Attention, propagation and aggregation in TDNN based speaker verification," in Interspeech 2020, 2020, pp.3830–3834

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

[8] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 4690–4699a

[9] Diederik P . Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," ICLR, 2015.

[10] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 212–220.

[11] J. Qi, W. Guo, and B. Gu, "Bidirectional multiscale feature aggregation for speaker verification," 2021, arXiv:2104.00230.

[12] Gu, B. and W. Guo, Dynamic Convolution with GlobalLocal Information for Session-Invariant Speaker Representation Learning. IEEE Signal Processing Letters, 2021.

[13] P. Matˇejka, O. Novotn`y, O. Plchot, L. Burget, M. D. S´anchez, and J.ˇCernock`y, "Analysis of score normalization in multilingual speaker recognition," Proc. Interspeech 2017, pp. 1567–1571, 2017.

[14] Niko Br ̈ummer and Edward De Villiers, "The bosaris toolkit: Theory, algorithms and code for surviving the new dcf," arXiv preprint arXiv:1304.2865, 2013.