INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Target speaker recovery and recognition network with average x-vector and global training

Wenjie Li[1,3], *Pengyuan Zhang*[1,3], *Yonghong Yan*[1,2,3]
[1] Key Laboratory of Speech Acoustics and Content Understanding, Institute of Acoustics, Chinese Academy of Sciences, China
[2]Xinjiang Laboratory of Minority Speech and Language Information Processing, Xinjiang Technical Institute of Physics and Chemistry, Chinese Academy of Sciences, China
[3]University of Chinese Academy of Sciences
liwenjie@hccl.ioa.ac.cn

Abstract

It is very challenging to do multi-talker automatic speech recognition (ASR). Some speaker-aware selective methods have been proposed to recover the speech of the target speaker, relying on the auxiliary speaker information provided by an anchor (a clean audio sample of the target speaker). But the performance is unstable depending on the quality of the provided anchors. To address this limitation, we propose to take advantage of the average speaker embeddings to build the target speaker recovery network (TRnet). The TRnet takes the mixed speech and the stable average speaker embeddings to produce the T-F masks for the target speech. During training of the TRnet, we summarize the speaker embeddings on the whole training dataset for each speaker, instead of extracting on a randomly picked anchor. On the testing stage, one or very few anchors are enough to get decent recovery results. The results of the TRnet trained with average speaker embeddings show 13% and 12.5% relative improvements on WER and SDR, compared with the short-anchor trained model. Moreover, to mitigate the mismatch between the TRnet and the acoustic model (AM), we adopted two strategies: fine-tuning the AM and training an global TRnet. Both of them bring considerable reductions on WER. The results show that the global trained framework gets superior performance.

Index Terms: target speaker recovery, x-vector speaker embedding, speech recognition, global training

1. Introduction

Owing to the development of deep learning [1] [2] [3] [4], single talker speech recognition system have achieved good accuracy and met the requirements of many practical scenarios. However, the performance of ASR system severely degrades, when the target speakers talk in crowded surroundings. The original speech is distorted by the noises from other sound sources and other speakers. The human auditory system is capable of separating a single speaker from others easily. But it is a big challenge for machine, which is called "cocktail party problem". Moreover, these scenarios are quite common in many applications, including mobile voice assistant, smart home devices, audio content monitoring. In these scenarios, we want to filter out the speech of the target speakers and get rid of the interference from noises and speech of other speakers.

Recently, many deep learning based approaches have been explored to solve this problem, which fall into two major classes: blind source separation and speaker aware extraction methods.

In the blind source separation methods, the DNN is often used to estimate the time-frequency (T-F) masks for different sources from a mixed corrupted signal. If the speech mixture contains n speakers, the DNN takes the mixed speech as input and genrates n output masks for all the speakers respectivly. The training process minimizes the mean square error (MSE) between the produced clean speech and the reference [5] [6]. There are several successful related works, including Deep clustering (DPCL) [7] [8] [9], Deep attractor network (DAnet) [10] and TASnet [11]. DPCL treats the separation as a clustering problem. It produces an embedding for each T-F bin in the spectrum, then clusters them by pulling close the embeddings from the same source and distancing those from different sources. Similar with DPCL, for the DAnet, each T-F bin will also be mapped into a high dimensional embedding. Then a network is trained to pull together the T-F bins for the corresponding sources by creating the attractor points. The attractor point represents the centroid of each source in the embedding space. Another approach called permutation invariant training (PIT) [12] [13] [14] [15] has been drawing much attention recently. For PIT, the different sources are treated as a whole set and the network is trained to minimize the assignment with minimum error, which can solve the arbitrary source permutation problem.

The increasing applications of speech assistant devices boost the studies on target speech enhancement[17] and recovery [16]. This work is quite relevant to the speaker-aware target speech recovery. In this approach, to recover the target speech from mixed speech, the anchor is often provided, which is a clean audio sample for the target speaker. This anchor is used to capture the speaker characteristic, and then the auxiliary speaker characteristic and mixed speech are fed into a DNN to produce the speech of the target speaker. Related researches have been done for target speech recovery such as DEnet [18], Voicefilter [19] and Speakerbeam[20]. They often take advantage of a randomly picked anchor to recover the speech of the target speaker from speech mixture. These methods can avoid the unknown number of speakers and arbitrary permutation problems, which are two major issues for traditional multi-talker speech recognition.

Since the voice of the target speaker can be varying with different contents and situations, the speaker characteristics of the same speaker in the randomly chosen anchor and mixed speech might be different. These will lead to an unstable recovery result for the anchor-based model. Moreover, the anchor is very short sometimes and we find that the duration and robustness of the anchor have big impact on the performance of the TRnet. To address this limitation, we propose the TRnet, which takes advantage of the average x-vector to recover the speech for the

http://dx.doi.org/10.21437/Interspeech.2019-1692

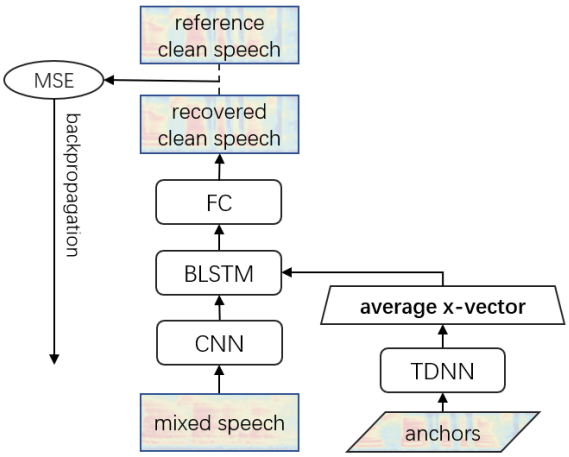

Figure 1: *The architecture of the proposed TRnet*

target speaker from the speech mixture. During training, we exploit the stable speaker characteristics by averaging the speaker embeddings over all the utterance of the same target speakers. On the testing stage, a single or very few anchors are enough to get decent results. Moreover, to mitigate the mismatch between the front-end TRnet and the back-end AM, we apply the AM fine-tuning and global training methods, which further reduces the recognition WER for the target speech.

The rest of this paper is organized as follows. In section 2 the x-vector speaker embedding model and the proposed TRnet are described in details. Section 3 presents experiment dataset and model configurations. In section 4, we report the experiment results. Finally, the conclusions are drawn in section 5.

2. Proposed methods

2.1. Speaker embedding network

Extracting speaker characteristics using DNN has been studied by many researcher [21] [22] [23] [24]. To do this, a speaker recognition model is often trained with speaker embedding extracted from the middle of the DNN. Here we use x-vector to encode the speaker discriminative information. X-vector was proposed in paper [25], and then improved with data augmentation [26] and self-attentive[27].

The x-vector network is applied to produce speaker embeddings from the anchors for the target speakers. To get the distinguishing embeddings for different speakers, a speaker recognition network is trained to have speaker discrimination. In the network, a pooling layer, which accumulates statistics on the whole input chunk, produces the long-term speaker characteristics for the input speech. The network is able to average the statistics on the entire speech signal and map the variable-length utterance to the fixed-dimensional embeddings.

The architecture of the speaker embedding TDNN model is shown in table 1. The input of the model is 23-dimensional MFCC feature, which is fed into 5 frame-level layers (frame1-5). Then the statistic pooling layer receives the output of the frame5 layer and aggregates over the input segment (T frames), computing the mean and standard deviation (1500-dimensional) over T frames. Then the mean and standard deviation are concatenated and passed to the following 2 segment-level layers (segment6-7), which can be used to compute the fixed-dimensional speaker embeddings. Finally, a softmax output layer is added for speaker classification. The output dimension N is equal to the number of speakers in the training set.

Table 1: *The architecture of speaker embedding TDNN model to extract x-vector*

Layer	Layer context	Input x output
frame1	$[t-2, t+2]$	115x512
frame2	$t-2, t, t+2$	1536x512
frame3	$t-2, t, t+2$	1536x512
frame4	t	512x512
frame5	t	512x1500
statis pooling	[0,T)	1500Tx3000
segment6	0	3000x512
segment7	0	512x512
softmax	0	512xN

After the speaker discriminative training, the TDNN model is able to produce the speaker embeddings that plays an important role in promoting the extraction performance for the TRnet.

2.2. The target speaker recovery and recognition network (TRnet)

2.2.1. The target speaker recovery

Different from the speech separation task, our object is recovering the speech for the target speaker from the mixed speech. Studies show that attention is important for the selective enhancement in the cocktail party problem[28]. In this work, the speaker characteristic extracted form the anchor serves as the attention trigger, which makes the recovery much easier. Since the speaker characteristic plays an important role in the TRnet, the length and quality of the anchor have big impact on the recovery performance.

Rather than randomly picking an audio sample as the anchor like other works, we propose to apply the average x-vector to exploit the stable speaker characteristics for the target speakers. It is done by averaging the x-vector over all the utterance of the same speakers. During training, the TRnet takes the mixed speech and the average x-vector to recover the speech of the target speakers. On the testing stage, a single or very few anchors are enough to get decent results, which helps with fast generalization to unseen speakers.

The architecture of the TRnet is in figure 1. To train this model, an audio pair and the average x-vector should be accessed in one pass. The audio pair includes the multi-talker mixed audio (mixed speech) and the individual clean audio for the target speaker in the mixture (reference clean speech). The x-vector is extracted using a previously trained speaker embedding model (TDNN). The TRnet takes the audio pair and the average x-vector to produce the T-F mask of the target speech. Here we use the phase-sensitive mask (PSM) [5], which is defined on Short-Time Fourier Transform (STFT) between each mixed speech and clean speech as follows:

$$M_{PSM}(t,f) = \frac{S(t,f)}{Y(t,f)}\cos\theta \tag{1}$$

the $|S(t,f)|$ and $|Y(t,f)|$ represent the spectrum of the clean and mixed speech respectively. The θ denotes the difference of the phase between the clean and mixed speech. The TRnet is trained to estimate the PSM for the speech of the target speaker.

As shown in figure 1, there is an independent speaker embedding network (TDNN) to generate the x-vector from the anchors. The major part of the TRnet contains two convolution layers (CNN), three bi-directional long short time memory (BLSTM) layers and one full connected layer (FC). The CNN layers take the spectrum of speech mixture as input. The following BLSTM layers accept the average x-vector and the output of the CNN layers, where the x-vector is repeatedly concatenated to the output of CNN in every frame. Finally, the full connection layer is added to map the output dimension to 257, which is equal to the dimension of the spectrum.

The TRnet produces the speaker aware mask for the target speech. The mask is then element-wise multiplied with the input mixed spectrum to generate the clean target spectrum. The loss function of training this TRnet is the MSE between the masked spectrum and the spectrum of the clean reference speech. It minimizes the following objective function:

$$\mathcal{L}_{MSE} = \sum_{t,f} ||S_{t,f} - Y_{t,f} \times M_{t,f}||_2^2 \quad (2)$$

where S is the clean spectrum of the target speaker, Y is the speech mixture, and M represents the estimated mask by the TRnet.

In the testing stage, only the mixed speech and the anchors for the target speakers are needed. If more than one anchor is provided, simply averaging the speaker embeddings of several anchors (for example 3) will benefit the extraction results a lot.

2.2.2. Acoustic model fine-tuning and global training for ASR

We train the baseline acoustic model (AM) with the clean speech that is used for producing the mixed speech. To mitigate the mismatch between the front-end TRnet and the clean AM, two approaches are adopted. On the one hand, we apply the TRnet to do extraction on the mixed training dataset. The resulted extracted audios, whose WER are lower than 30% (decoding by the clean AM), are selected. Then we train a fine-tuning AM using the selected better audios. The parameters of the fine-tuning AM are initialized with the clean AM.

On the other hand, the TRnet is global trained to optimize the recognition loss directly. This overall framework takes the MFCC feature as input and produces the probabilities for the GMM states, which performs the target speech recovery and speech recognition in a single computation step. In other words, we minimize the cross entropy between the reference alignments and the outputs. This strategy allows for the direct speech recognition of the target speech. It is a simpler and more compact framework for the multi-talker ASR.

3. Dataset and model configurations

3.1. X-vector speaker embedding network

To extracte the x-vector, we followed the setup in paper [26]. The training data contained more than 6000 speakers, including part of Switchboard dataset and the NIST speaker recognition evaluation (SRE) data. Moreover, the data augmentation technique[26] was applied to increase the diversity of the training data by adding additive noise and reverb [29]. The model was trained with both clean data and the augmented part. The feature was 23 dimensional MFCC. An energy based speech activity detection was applied to select the speech frames. The speaker embedding network was built with TDNN [3] model using Kaldi [30] toolkit, following the recipe at "https://github.com/kaldi-asr/kaldi/tree/master/egs/sre16/v2". The configurations of speaker embedding model was described in section 2. Overall, it was a 7-layer TDNN model with a statistic pooling layer, whose parameters are shown in table 1. After the speaker discriminative training, the 512-dimensional embeddings were extracted from the output of segment6 layer. Through this previously trained model, we can extract the compact embeddings for the target speaker.

3.2. The TRnet

To train the TRnet, part of data from Wall Street Journal (WSJ) corpus[31] was applied to produce the two-talker mixture (WSJ0-2mix) dataset. This dataset has already been used in many related works[7] [8] [12] [13], which can be found at "http://www.merl.com/demos/deep-clustering". As mentioned in paper [7], there were about 30 hours training and 10 hours validation mixed data generated from WSJ0 set si_tr_s, containing 101 different speakers. The 5 hours mixed testing data was obtained from the WSJ0 si_dt_05 and si_et_05 including 16 unseen speakers. To generate the mixed speech, we randomly chose an interfering speech from another speaker and then mixed it with the target clean speech. It was done by summing the clean target speech and interfering speech with different signal-to-noise ratios between 0dB and 5dB.

For the front-end TRnet, the log spectral magnitude was provided as input feature, which was 257-dimensional STFT spectrum. The network was trained with TensorFlow and was composed of 2 CNN layers, 3 BLSTM layers and 1 FC layer. The kernel size was 9×9 and the number of channels were 64 for both CNN layers. The BLSTM layers contained 640 hidden units. Then the fully connected layer accepted the 1280-dimensional output of the last BLSTM layer and generated 257-dimesional mask for the target speech. This mask was multiplied with the input mixed spectrum to obtain the clean spectrum, which was the final output of the TRnet.

To build the back-end clean AM, we followed the standard setup in kaldi WSJ example. The 5-layer 650-unit TDNN AM was trained with the 30-hour clean data generating the wsj0-2mix training set. The clean AM was trained with cross entropy criteria, whose WERs are indicated as "clean AM" in table 2. Then the AM fine-tuning strategy mentioned in section 2.2.2 was applied. The WERs are shown in table 2 as "fine-tuning AM".

As for the global training model, since the model did both target speaker recovery and recognition, we extended the 3 layers 650-unit BLSTM to 3 layers 1024-unit BLSTM. The input feature was 43-dimensional pitch MFCC, with cepstral mean and variance normalization (CMVN). The output was 3352 senones, representing the GMM states. TensorFlow [32] was used to build the TRnet.

To evaluate the performance of the TRnet, we adopt two criterias: word error rate (WER) and the source to distortion ratio (SDR) from speech recognition and signal processing perspectives. We present the WER for "clean AM" and "fine-tuning AM". The SDR (higher is better) shown in the table 2 at the last column is computed using the bss_eval toolbox [33].

4. Experiments and results

The experiment results are shown in table 2. The third and fourth columns present the different kinds of anchors used for training and testing respectively, which means:

Table 2: *WER (%) and SDR (dB) for TRnet with different anchors*

num	Model	anchor		WER		SDR
		train	test	clean AM	fine-tuning AM	
1	-	-	-	82.35	79.13	1.12
2	TRnet	short	short	44.64	20.12	9.78
3	TRnet	long	long	41.93	17.29	10.17
4	TRnet	per-spk	short	42.2	17.59	10.21
5	TRnet	per-spk	long	40.47	16.31	10.37
6	TRnet	per-spk	3-average	38.96	14.24	10.93
7	TRnet	per-spk	5-average	**38.81**	**13.79**	**11.0**
8	TRnet	per-spk	20-average	38.69	13.92	10.99
9	TRnet	per-spk	per-spk	38.8	13.94	10.98
10	Global TRnet	per-spk	3-average	14.29		-
11	Global TRnet	per-spk	5-average	**13.59**		-
12	Global TRnet	per-spk	20-average	**13.19**		-
13	Global TRnet	per-spk	per-spk	13.19		-

- "short": the anchor to generated speaker embedding is a random utterance shorter than 3 seconds;
- "long": the anchor is a random utterance around 10 seconds;
- "per-spk": we average speaker embeddings over all the utterance of the same speaker in the training dataset ;
- "n-average": we randomly choose n utterance for target speaker to compute the average speaker embeddings.

For all the experiments, the anchors for the target speaker are different from the clean target speech generating the mixed speech. We present the WER and SDR of the mixed speech in table 2 at the first line. The results of the TRnet trained with single short and long anchors are shown in line 2 and 3, which indicate that the duration of the anchor has big impact on the recovery performance for the single-anchor TRnet. The reason might be that the speaker characteristic extracted from longer anchor is more reliable, which also indicates the robustness of the speaker embedding is quite important for the TRnet. Then, we applied the average speaker embeddings to train the TRnet and tested with single short and long anchors. The results on line 4 and 5 show improvements on both WER and SDR, compared with the TRnet trained with the single anchor (2-3), especially for the short anchors. Training the TRnet with average x-vector, the gap between using short and long anchor for testing becomes much smaller. We find that with the stable average speaker embeddings, the TRnet learns more robust construct capacity for the target speech recovery.

If more than one anchor is provided during testing, we can average the x-vector on several anchors to get stable speaker characteristics for the target speech recovery. To do this, we randomly choose different number of anchors for the target speakers, producing the average x-vector. The results are presented on line 6-8, which achieve considerable improvements. To balance the extraction performance and the number of the anchors needed for testing, we prefer 5 anchors to generate average speaker embeddings. It achieves 13% and 12.5% relative improvements on WER and SDR compared with the short-anchor trained TRnet. As a reference, the results of using per-speaker x-vector for testing is also shown at line 9. From this, we can see that 5 anchors are enough to achieve nearly optimal recovery performance for the TRnet trained with the average embeddings. It is easy to get 3-5 anchors for the target speakers, which makes it a practical and effective way to use TRnet with average speaker embeddings.

Then we fine-tune the clean AM using the extracted speech (training set) with better quality, whose results is shown in table 2 as "fine-tuning AM". The model fine-tuning method is quite effective to improve the recognition accuracy, which reduces the WER for 25.02% on 5-average test. To further improve the ASR performance and system simplicity, we trained a global TRnet to do both target speech recovery and ASR simultaneously. The WERs are presented on line 10-13. From this, we can see that the global trained TRnet gets 4.3% WER reduction over the front-end TRnet with fine-tuning AM respectively.

5. Conclusions

In this paper, the speaker aware TRnet is proposed to recover speech for the target speaker from speech mixture. The TRnet takes the mixed speech and speaker embedding called x-vector to generate the target speech. We improve the recent single-anchor based extraction with average speaker embeddings. During training of the TRnet, the x-vector is summarized on the whole training dataset to exploit stable long-term speaker characteristics for each target speaker. On the inference stage, very few anchors are enough to achieve decent extraction results. The experiment results show that the robustness of the speaker embedding has big impact on the recovery performance. Thus, the proposed TRnet outperforms the single short-anchor baseline for about 13% and 12.5% on SDR relatively. Moreover, using the extracted speech to fine-tune the AM, the recognition WER can be reduced a lot. If we train a global TRnet to optimize the recognition loss directly, it will achieve better ASR performance with system simplification. This work can be improved by using more training data with more speakers. More challenging tasks can be done to recover the target speech form the speech mixture in noise surroundings.

6. Acknowledgements

This work is partially supported by the National Key Research and Development Program (Nos. 2016YFB0801203, 2016YFB0801200), the National Natural Science Foundation of China (Nos. 11590774, 11590770),the Key Science and Technology Project of the Xinjiang Uygur Autonomous Region (No.2016A03007-1),the Pre-research Project for Equipment of General Information System (No.JZX2017-0994/Y306).

7. References

[1] D. Yu and L. Deng, Automatic speech recognition: A deep learning approach. Springer, 2014.

[2] H. Sak, A. Senior, and F. Beaufays, Long short-term memory recurrent neural network architectures for large scale acoustic modeling, in Fifteenth annual Conference of the International Speech Communication Association, 2014.

[3] V. Peddinti, D. Povey, and S. Khudanpur, A time delay neural network architecture for efficient modeling of long temporal contexts, in INTERSPEECH, 2015.

[4] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, Convolutional, long short-term memory, fully connected deep neural networks, in ICASSP, 2015, pp. 45804584.

[5] D. Wang and J. Chen, 'Supervised speech separation based deep learning: An overview,' arxiv, 2017.

[6] A. Narayanan and D.Wang, Improving robustness of deep neural network acoustic models via speech separation and joint adaptive training, IEEE/ACM Trans. ASLP, vol. 23, no. 1, pp. 92101, 2015.

[7] J. R. Hershey, Z. Chen, J. L. Roux, and S. Watanabe, 'Deep clustering: Discriminative embeddings for segmentation and separation,' ICASSP 2016, pp. 31-35, 2016.

[8] Y. Isik, J. L. Roux, Z. Chen, S. Watanabe, and J. R. Hershey, 'Single-channel multi-speaker separation using deep clustering,' INTERSPEECH2016, pp. 545-549.

[9] Seki H, Hori T, Watanabe S, et al. 'A purely end-to-end system for multi-speaker speech recognition'[J]. arXiv preprint arXiv:1805.05826, 2018.

[10] Z. Chen, Y. Luo, and N. Mesgarani, 'Deep attractor network for single-microphone speaker separation,' ICASSP2017.

[11] Luo Y, Mesgarani N. 'TasNet: time-domain audio separation network for real-time, single-channel speech separation'[C]//2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018: 696-700.

[12] Y. Dong, K. Morten, T. Zheng-Hua, and J. Jesper, 'Permutation invariant training of deep models for speaker-independent multitalker speech separation,' ICASSP2017, pp. 31-35.

[13] Kolbaek M , Yu D , Tan Z H , et al. 'Multi-talker Speech Separation with Utterance-level Permutation Invariant Training of Deep Recurrent Neural Networks' [J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2017.

[14] M. Kolbaek, D. Yu, Z. H. Tan, and J. Jensen, 'Joint separation and denoising of noisy multi-talker speech using recurrent neural networks and permutation invariant training,' MLSP, 2017.

[15] Chang X , Qian Y , Yu K , et al. 'END-TO-END MONAURAL MULTI-SPEAKER ASR SYSTEM WITHOUT PRETRAINING,' arxiv, vol. abs/1707.06527, 2018.

[16] Brian King, I-Fan Chen, Yonatan Vaizman, et.al, 'Robust Speech Recognition via Anchor Word Representations', Interspeech 2017, pp.1570-1574

[17] K. Zmolikova, M. Delcroix, K. Kinoshita, T. Higuchi, A. Ogawa, and T. Nakatani, 'Speaker-aware neural network based beamformer for speaker extraction in speech mixtures,' Interspeech17, 2017.

[18] Wang J, Chen J, Su D, et al. 'Deep extractor network for target speaker recovery from single channel speech mixtures'[J]. arXiv preprint arXiv:1807.08974, 2018.

[19] Wang Q, Muckenhirn H, Wilson K, et al. Voicefilter: Targeted voice separation by speaker-conditioned spectrogram masking[J]. arXiv preprint arXiv:1810.04826, 2018.

[20] Delcroix M, Zmolikova K, Kinoshita K, et al. Single channel target speaker extraction and recognition with speaker beam[C]//2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018: 5554-5558.

[21] Ehsan Variani, Xin Lei, Erik McDermott, Ignacio Lopez Moreno, and Javier Gonzalez-Dominguez, 'Deep neural networks for small footprint text-dependent speaker verification,' in Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on. IEEE, 2014, pp. 4052-4056.

[22] Chao Li, Xiaokong Ma, Bing Jiang, Xiangang Li, Xuewei Zhang, Xiao Liu, Ying Cao, Ajay Kannan, and Zhenyao Zhu, 'Deep speaker: an end-to-end neural speaker embedding system,' CoRR, vol. abs/1705.02304, 2017.

[23] Shi-Xiong Zhang, Zhuo Chen, Yong Zhao, Jinyu Li, and Yifan Gong, 'End-to-end attention based text-dependent speaker verification,' CoRR, vol. abs/1701.00562, 2017.

[24] Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno, 'Generalized end-to-end loss for speaker verification,' in International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018, pp. 48794883.

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

[26] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, Sanjeev Khudanpur, 'X-vectors: Robust DNN Embeddings for Speaker Recognition', ICASSP 2018.

[27] Zhu, Yingke, et al. 'Self-attentive speaker embeddings for text-independent speaker verification.' Interspeech 2018, pp 3573-3577.

[28] A. W. Bronkhorst, The cocktail-party problem revisited: early processing and se- lection of multi-talker speech, Attention, Perception, and Psychophysics, Springer, 2015.

[29] 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 Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing. IEEE, 2017, pp. 52205224.

[30] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz et al., 'The Kaldi speech recognition toolkit,' in Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop, 2011.

[31] J. Garofolo et al., 'CSR-I (WSJ0) Complete LDC93s6a,' 1993, philadelphia: Linguistic Data Consortium.

[32] Girija S S. 'Tensorflow: Large-scale machine learning on heterogeneous distributed systems'[J]. 2016.

[33] Emmanuel Vincent, R'emi Gribonval, and C'edric F'evotte, 'Performance measurement in blind audio source separation,' IEEE transactions on audio, speech, and language processing, vol. 14, no. 4, pp. 1462-1469, 2006.