INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Improved Guided Source Separation Integrated with a Strong Back-end for the CHiME-6 Dinner Party Scenario

Hangting Chen[1,2], *Pengyuan Zhang*[1,2*], *Qian Shi*[1,2], *Zuozhen Liu*[1,2]

[1]Key Laboratory of Speech Acoustics & Content Understanding, Institute of Acoustics, CAS, China
[2]University of Chinese Academy of Sciences, Beijing, China

{chenhangting,zhangpengyuan,shiqian,liuzuozhen}@hccl.ioa.ac.cn

Abstract

The CHiME-6 dataset presents a difficult task with extreme speech overlap, severe noise and a natural speaking style. The gap of the word error rate (WER) is distinct between the audios recorded by the distant microphone arrays and the individual headset microphones. The official baseline exhibits a WER gap of approximately 10% even though the guided source separation (GSS) has achieved considerable WER reduction. In the paper, we make an effort to integrate an improved GSS with a strong automatic speech recognition (ASR) back-end, which bridges the WER gap and achieves substantial ASR performance improvement. Specifically, the proposed GSS is initialized by masks from data-driven deep-learning models, utilizes the spectral information and conducts a selection of the input channels. Meanwhile, we propose a data augmentation technique via random channel selection and deep convolutional neural network-based multi-channel acoustic models for back-end modeling. In the experiments, our framework largely reduced the WER to $34.78\%/36.85\%$ on the CHiME-6 development/evaluation set. Moreover, a narrower gap of $0.89\%/4.67\%$ was observed between the distant and headset audios. This framework is also the foundation of the IOA's submission to the CHiME-6 competition, which is ranked among the top systems.

Index Terms: Robust speech recognition, CHiME-6, multi-channel speech separation, acoustic modeling

1. Introduction

In recent years, automatic speech recognition (ASR) has achieved high accuracy on clean close-talk data [1][2][3]. Yet the ASR robustness remains a serious challenge under a relatively uncontrolled condition, where speaker overlap, noise and reverberation occur simultaneously [4]. The latest CHiME-6 challenge uses accurately synchronized audios from CHiME-5, which were recorded in real 4-people parties, and presents extreme speech overlap and unconstrained speaking style [5].

The guided source separation (GSS) [6] serves as the front-end of the official CHiME-6 baseline, which performs source separation based on the complex angular central Gaussian mixture model (CACGMM) [7] with given annotations to avoid the frequency permutation problem. This approach exhibits considerable reduction of the word error rate (WER) when using all microphone arrays [8] and when decoded by the factorized time-delayed neural network (TDNNF) [9]. However, the officially provided baseline of CHiME-6 exhibits a gap between the enhanced far-field and headset audios, which is approximately 10% on the development (Dev.) and evaluation (Eval.) set in our test [5]. To alleviate the performance degradation under the distant multi-talker condition and to improve the ASR accuracy on the low-resource task, we explore an improved GSS combined with deep learning-based methods, augmentation of the enhanced training data, and multi-channel acoustic models (AMs) based on the convolutional neural network (CNN), the residual connection (Res) and bidirectional long short-term memory (BLSTM).

In the front-end processing, 2-stage single-channel deep-learning models with an additional concentration loss are trained to simultaneously predict masks for GSS initialization and embeddings for spectral modeling. The concentration loss is designed for an improved clustering performance under the von Mises-Fisher (vMF) mixture model. Besides, the baseline GSS is improved with 3 modifications, resulting in an approximate 2% WER reduction under the CNN-TDNNF-BLSTM. In the back-end modeling, we use the random channel selection to generate more enhanced training data. A CNNRes-BLSTM is proposed integrated with the multi-channel branch. Compared with the official baseline, our framework reduces the WERs from $51.8\%/51.3\%$ to $34.78\%/36.85\%$ on the Dev./Eval. set, which exhibits a lower performance degradation of $0.89\%/4.67\%$ between the enhanced and headset audios, and was ranked among the top systems according to the competition results [10].

The rest of the paper is organized as follows. Section 2 describes our front-end processing. Section 3 presents the training data augmentation and acoustic models. Section 4 and 5 show the experimental configuration and results, respectively. Finally, Section 6 concludes this work.

2. Front-end Processing

2.1. Single-channel speech separation with concentrated embeddings

Single-channel speech separation is performed with data-driven deep learning-based models, which provide the masks and embeddings for the GSS. A 2-stage procedure with speaker-dependent models is conducted here, similar with [11]. The 1st-stage speech separation model (SS1) utilizes the non-overlapped far-field segments, while the 2nd-stage (SS2) additionally uses the enhanced audios from the SS1 and the baseline GSS. The architectures of the SS1 and SS2 models are plotted in Figure 1. The SS1 model is trained for each speaker in each session with the indirect mapping loss of ideal ratio masks (IRM) [12]. The SS2 model is trained for each session with one-hot vectors indicating speakers. Its middle and final layers output embeddings and phase-sensitive masks (PSMs) [13], respectively. The SS2 model optimizes both the mean squared

This work is partially supported by the Strategic Priority Research Program of Chinese Academy of Sciences (No.XDC08010300).

*corresponding author

http://dx.doi.org/10.21437/Interspeech.2020-1606

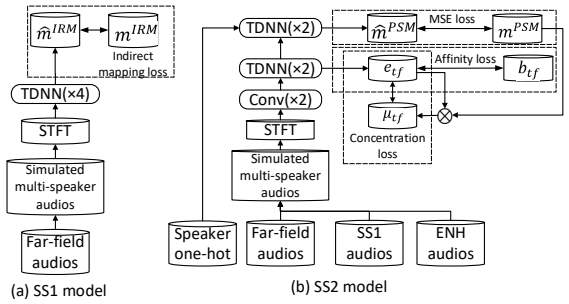

Figure 1: *The schematic diagram of* 2*-stage separation models.*

error (MSE) of the PSM and the affinity matrix [14],

$$L_m = \frac{1}{T \times F} \sum_{t,f} (\hat{m}_{t,f,s} - m_{t,f,s})^2, \tag{1}$$

$$L_e = \frac{1}{(T \times F)^2} \sum_{t,f,t',f'} (\mathbf{e}_{t,f}^T \mathbf{e}_{t',f'} - \mathbf{b}_{t,f}^T \mathbf{b}_{t',f'})^2, \tag{2}$$

where s is the target speaker index, $\hat{m}_{t,f,s}$, $m_{t,f,s}$, $\mathbf{e}_{t,f} \in \mathbb{R}^{D\times 1}$ and $\mathbf{b}_{t,f} \in \{0,1\}^{S\times 1}$ represent the estimated PSM, the oracle PSM, the estimated embedding and the speaker membership indicator for each time-frequency (T-F) bin, T and F are the total number of frames and frequency bins in the sample, D and S are the embedding dimension and the number of speakers.

A concentration loss is proposed here to derive a more compact embedding,

$$\mathbf{u}_s = \frac{\bar{\mathbf{u}}_s}{\|\bar{\mathbf{u}}_s\|_2} \text{ with } \bar{\mathbf{u}}_s = \frac{1}{\sum_{t,f} m_{t,f,s}} \sum_{t,f} m_{t,f,s} \mathbf{e}_{t,f}, \tag{3}$$

$$L_c = -\frac{1}{\sum_{t,f} m_{t,f,s}} \sum_{t,f} m_{t,f,s} \mathbf{u}_s^T \mathbf{e}_{t,f}, \tag{4}$$

where $\mathbf{u}_s$ is the mean direction of speaker s. Optimizing the concentration loss leads to more concentrated embeddings, that is, it forces the similarity between the mean vector and the T-F embeddings with large $m_{t,f,s}$ close to 1. Moreover, the concentration loss can be viewed as the M step in the expectation-maximization (E-M) algorithm. The M step of the vMF mixture model is to optimize

$$\sum_s \sum_{t,f} \gamma_{t,f,s} ln(\text{vMF}(\mathbf{e}_{t,f}|\Theta_{vMF}[s]), \tag{5}$$

where $\gamma_{t,f,s}$ obtained from the E step describes the category probability and $\Theta_{vMF}[s]$ is the parameters of the sth cluster [15]. By setting $\gamma_{t,f,k} = m_{t,f,s}$, it is simplified as

$$\sum_s \sum_{t,f} c_1 m_{t,f,s} \mathbf{u}_s^T \mathbf{e}_{t,f} + c_2, \tag{6}$$

where c_1 and c_2 are related to the concentration parameter κ. The difference between Eq. (4) and (6) is that the M step only optimizes $\mathbf{u}_s$ with provided samples $\mathbf{e}_{t,f}$, but Eq. (4) optimizes $\mathbf{e}_{t,f}$ to achieve a more concentrated pattern. By rewriting Eq. (4) as $L_c = -\|\bar{\mathbf{u}}_s\|_2$, its gradient vector at a given embedding $\mathbf{e}_{t,f}$ is

$$\frac{\partial L_c}{\partial \mathbf{e}_{t,f}} = \frac{-m_{t,f,s}}{\sum_{t,f} m_{t,f,s}} \mathbf{u}_s, \tag{7}$$

implying that the embedding is pulled to the mean vector if the T-F bin is dominated by the target speaker.

In the training phase, the input speaker one-hot for speaker s is randomly chosen regarding to the mixed speakers. The total loss can be written as

$$L = L_m + \alpha_1 L_e + \alpha_2 L_c, \tag{8}$$

where α_1 and α_2 are the loss balance factors.

2.2. Multi-channel separation with an improved GSS

The GSS conducts mask estimation and beamforming with the provided time annotation or the ASR alignment. It first dereverberates the short time frequency transform (STFT) of the utterance with temporal context by using weighted prediction error (WPE) [16][17]. Then, the CACGMM clusters the T-F bins with soft labels for each source. The MVDR beamforming is employed to output enhanced utterances [18]. In the following, we list our 3 modifications to improve the baseline GSS.

First, the CACGMM is initialized by $p_{init}(t,f,s)$, which integrates single-channel masks and interpolated frame-level source presence probability (SPP) $a(t,s)$,

$$a(t,s) = (1-\beta) a_{annot}(t,s) + \beta a_{align}(t,s), \tag{9}$$

$$p_{init}(t,f,s) = \hat{m}(t,f,s) a(t,s), \tag{10}$$

where β is the confidence factor of the alignments, a_{annot} and a_{align} are SPP from the annotation and ASR alignment. The interpolation is aimed to alleviate the inaccuracy of the alignment caused by the ASR transcription. The initialization of each source $p_{init}(t,f,s)$ in CACGMM is computed with the SS2 models. The CACGMM is updated following the rules in [19] with the interpolated $a(t,s)$.

Second, the CACGMM uses few spectral cues since its input is the normalized multi-channel STFT. The vMF-CACGMM [20] is deployed to model both spectral and spatial information, whose log likelihood function is formulated as

$$\begin{aligned} &log\mathcal{L}(\mathbf{e}_{t,f}, \tilde{\mathbf{y}}_{t,f}|\Theta_{vMF}, \Theta_{CACG}) \\ &= \textstyle\sum_{t,f} log \sum_k \pi_{t,f}^{(k)} \text{vMF}(\mathbf{e}_{t,f}|\Theta_{vMF})^{\nu} \mathcal{N}_c(\tilde{\mathbf{y}}_{t,f}|\Theta_{CACG}), \end{aligned} \tag{11}$$

where $\tilde{\mathbf{y}}_{t,f}$ is the normalized vector of the M-channel STFT signal $\mathbf{y}_{t,f}$, ν is the spectral weight, Θ_{vMF} and Θ_{CACG} are the parameters of the vMF and the complex Gaussian $\mathcal{N}_c$, $\pi_{t,f}^{(k)}$ is the time-invariant mixture weight.

Third, we conduct 2 different selection methods to remove channels with poor information. The CHiME-6 data was recorded by multiple arrays, each with 4 channels. The signal-to-interference-plus-noise ratio (SINR)-based selection removes channels according to the rank of SINRs. To obtain the SINR information with MVDR [18][21], 24-channel GSS is first performed to calculate the power spectral density matrix of the target speaker and the interference. Another coherence-based method [22] ranks the channels in the order of the coherence value with the reference one. The reference channel r is selected as that with the highest SINR value among the 4 channels in the reference array. Note that the reference array is known in the CHiME-6 dataset. The coherence ρ_m of channel m is calculated as follows,

$$\rho_m = \frac{E[y_{t,f,m} y_{t,f,r}]}{E|y_{t,f,r}|^2}, \tag{12}$$

where $E[\cdot]$ is the expectation across the T-F bins.

The whole procedure is plotted in Figure 2. Compared with the baseline, our approach uses better initialization, integration of spatial and spectral information, and channel selection to achieve better multi-channel separation.

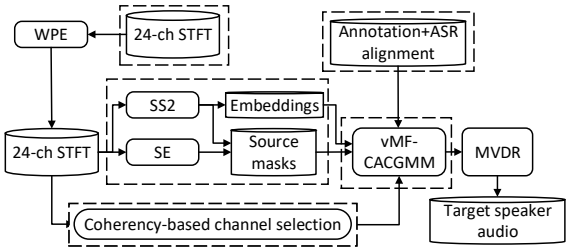

Figure 2: *The flow chart of the improved GSS, where the dashed blocks indicate the modifications. A universal speech enhancement model (SE) is trained to output noise masks.*

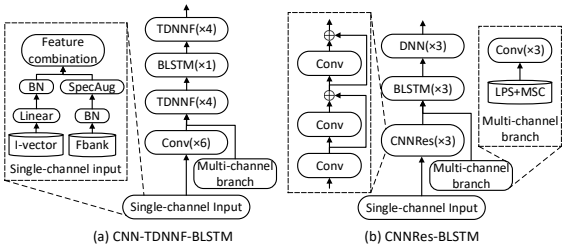

Figure 3: *The architectures of (a) CNN-TDNNF-BLSTM and (b) CNNRes-BLSTM, which use the same single-channel input and multi-channel branch. The input of multi-channel branch is* 4*-channel LPS and cross-channel MSC, forming a feature map of size* $T \times 5 \times F$*. Each "Conv" layer consists of convolutional, ReLU and batch normalization (BN) operators.*

3. Back-end Modeling

3.1. Data augmentation

Besides the SpecAug [23], we propose a data augmentation technique via random channel selection to partially solve the lack of enhanced training data and to catch the variability of the testing data. Following the improved GSS proposed in Section 2.2, the SINR/coherence-based channel selection is replaced by a random method to introduce diversity. The algorithm extends the training data from 1 fold to 7. The detailed description is listed in Table 1.

3.2. Deep CNNs with the multi-channel branch

Compared with TDNNF, the CNN-TDNNF architecture can already achieve a considerable performance boost [8]. In this paper, 2 AMs are proposed with an additional mountable multi-channel branch (Figure 3). The CNN-TDNNF-BLSTM is derived from the CNN-TDNNF with an inserted BLSTM layer. Another model is CNNRes-BLSTM, composed of deep CNN blocks as well as multiple BLSTM layers. In our practice, we have found that deep CNNs with residual connections are much more suitable for the low-resource task. Moreover, the BLSTM layers enable sequence modeling to be performed. Inspired by [6], we adopt a multi-channel branch with log power spectral (LPS) and magnitude squared coherence (MSC) [24]. The multi-channel branch uses CNNs instead of BLSTM since it is considered to be a low-level feature extraction module.

4. Experimental Configuration

Our experiments were conducted on the CHiME-6 dataset, derived from the CHiME-5 audios with accurate synchronization among microphone arrays. Track 1 provides segmented utterances as well as the time annotations of each speaker. The official baseline with the GSS front-end and the TDNNF acoustic model has achieved WERs of $51.8\%/51.3\%$ on Dev./Eval., respectively [5]. Following [8], a more advanced baseline is adopted in this paper with the GSS using ASR alignments and CNN-TDNNF acoustic models, which achieved $44.52\%/47.20\%$ in our experiments.

Table 1: *The data augmentation for the enhanced training data.*

Description of channel selection	Fold number
Remove 4 channels via SINR rankings	1
Randomly remove 1 array	1
Randomly remove 2 arrays	1
Keep the outer channels in each array [5]	1
Randomly keep 2 arrays	1
Randomly choose channels	2

The STFT was conducted every $16ms$ over $64ms$ for the proposed front-end processing. The simulated audios for SS1 training have a size of 50 hours and a SINR level of $-5, 0, 5, 10dB$ for each speaker in each session. The simulated audios for SS2 have a size of 200 hours and a SINR level ranging from $-10dB$ to $10dB$ in each session. The dimension of embeddings D was 20. The loss balance factors α_1 and α_2 in Eq. (8) were set to 100.0 and 1.0, respectively.

In GSS, the temporal context was set to $15s$. The SS2 models provided source masks and embeddings by averaging the output over channels. The interpolation coefficient β was set to 0.6. The spectral weight ν for vMF-CACGMM was 0.5. The proposed GSS was conducted with the following steps. First, the baseline ASR model was used to align the audios from baseline GSS and transcriptions decoded by CNN-TDNNF. Then, the SINR information of all channels was obtained by the 24-channel vMF-CACGMM. In total, $4/5$ channels were removed from $20/24$, respectively. Finally, the vMF-CACGMM was rerun with the selected channels.

The standard CHiME-6 recipes were used to train GMM-HMM alignment models, clean up the training data, augment the data with 3-fold speed perturbation.[1] The 1-fold data was around 120 hours after the perturbation. The input feature of acoustic models was 40-dimensional FBank and 100-dimensional i-vectors. The input of the mountable 4-channel branch was 257-dimensional LPS and MSC with a total dimension of 1285. The single-channel CNN-TDNNF-BLSTM used the headset ($\times 2$) and augmented enhanced data ($\times 7$), approximately $1k$ hours. The CNNRes-BLSTM additionally utilized the original far-field ($\times 6$) and simulated data ($\times 6$), approximately $2.6k$ hours totally. The multi-channel branch was partially updated [25] with the 12-fold enhanced data (2 out of 7-fold enhanced data and each segment recorded by 6 microphone arrays). The convolutional layers in CNNRes blocks employed kernels with sizes varying from 3 to 9 and the output channels of $128, 256, 256$. The AMs were trained with lattice-free maximum mutual information (LF-MMI) criterion. The L2 and cross-entropy loss was applied with scales of 10^{-5} and 0.1, respectively. The 3-gram language model provided by the competition was used for decoding. The WER results listed in the paper were from the 2-stage decoding, which refines the i-vector extraction based on the first pass [26].

5. Results and Discussion

We evaluated the data augmentation and the CNN-TDNNF-BLSTM before the front-end experiments, which can be ac-

[1] https://github.com/kaldi-asr/kaldi/tree/master/egs/chime6/s5_track1

Table 2: *The Dev./Eval. WERs with different training data and architectures (Arch.). CT, CTB and CTB+MC represent the CNN-TDNNF, CNN-TDNNF-BLSTM and CNN-TDNNF-BLSTM with the multi-channel branch, respectively.*

Data(fold)	SpecAug	Arch.	WERs(%)
Headset(×2)+Enh(×1)	×	CT	44.52/47.20
Headset(×2)+Enh(×1)	✓	CT	43.04/45.12
Headset(×2)+Enh(×7)	✓	CT	41.35/43.12
Headset(×2)+Enh(×7)	✓	CTB	40.82/42.43
Enh(×12)	✓	CTB+MC	**40.07/40.39**

Table 3: *The Dev./Eval. WERs under the multi-channel CNN-TDNNF-BLSTM with different front-end settings, including channel selection (Chs.), initialization (Init.), frame-level SPP controlled by β and the distribution of the probabilistic model. We used reference (REF), all (24), outer (12) channels and channel selection based on the SINR and coherence (Coh) values. The embeddings of vMF*/vMF were generated from the SS2 model without/with the concentration loss.*

Chs.	Init.	SPP(β)	Distribution	WERs(%)
REF	SS2	-	vMF*	69.96/59.76
REF	SS2	-	vMF	67.46/57.20
12	-	1.0	CACG	40.07/40.39
12	SS2	1.0	CACG	39.41/39.78
12	SS2	0.6	CACG	39.02/39.31
12	SS2	0.6	vMF+CACG	38.72/38.77
24	SS2	0.6	vMF+CACG	37.94/38.81
SINR	SS2	0.6	vMF+CACG	37.55/38.51
Coh	SS2	0.0	vMF+CACG	**37.51/38.30**
Headset				36.19/34.86

counted for by 2 reasons. First, a strong AM ensures that the gains in the front-end are not to be eaten up if using a better back-end. Second, the CNN-TDNNF-BLSTM is decoded faster than CNNRes-BLSTM and is also enough to test the effectiveness of the proposed data augmentation and the multi-channel branch. As listed in Table 2, the data augmentation under CNN-TDNNF achieved a WER reduction of 3.17%/4.08%, including using more training data and the SpecAug technique [23]. In addition, the CNN-TDNNF-BLSTM integrated with the multi-channel branch improved 1.28%/2.73% compared with the single-channel CNN-TDNNF.

The experiments of the front-end processing were conducted under the multi-channel CNN-TDNNF-BLSTM (Table 3). The embeddings from the SS2 model were tested under the vMF mixture model, where the WER results were much higher than that of the CACGMM. With the proposed concentration loss, the embeddings were more suitable for the vMF clustering, which gave 2.50%/2.56% WER reduction on the Dev./Eval. set. As for the CACGMM, its performance was improved from 40.07%/40.39% to 39.02%/39.31% with the better initialization and the interpolation of the annotation and alignment. The vMF-CACGMM replaced the CACGMM to model both the spectral and spatial information, which reduced WERs by approximately 0.30%/0.54%. However, using all channels led to performance degradation on the Eval. set. We selected the channels via the SINR and coherence rankings, both of which utilized the SINR information generated from the MVDR beamforming of the 24-channel vMF-CACGMM. The coherence-based method achieved the best WERs of 37.51%/38.30% with a reduction of 2.56%/2.09% compared with our baseline GSS, a gap of 1.32%/3.44% with the headset audios. Spectral examples are depicted in Figure 4, where the audio enhanced by the

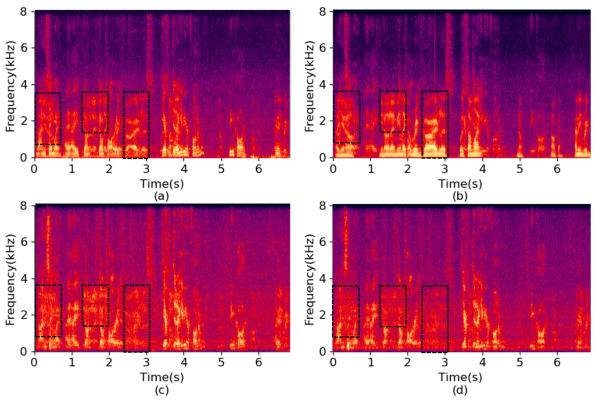

Figure 4: *The STFT spectrum recorded by (a) the target speaker's headset microphone, (b) the interference speaker's headset microphone, enhanced by (c) our baseline GSS, (d) the proposed improved GSS. The black boxes are the interference signals, which can be heard even on the target speaker's headset microphone. The proposed GSS achieved better separation and denoising than the baseline.*

Table 4: *A comparison of Dev./Eval. WERs between the enhanced audios (Enh.) processed by our best front-end and headset audios (Headset).*

Front-end	Back-end	WERs(%)	
		Enh.	Headset
GSS [5]	TDNNF	∼ 51.8/51.3	41.40/39.90
GSS [6]	CNN-TDNNF	44.52/47.20	37.22/36.97
Improved GSS	CNNRes-BLSTM	35.44/37.95	34.66/33.08
Improved GSS	CNNRes-BLSTM + RNNLM	**34.78/36.85**	**33.89/32.18**

improved GSS exhibits a cleaner spectrum than that from the baseline GSS.

The results of multi-channel CNNRes-BLSTM are listed in Table 4, exhibiting a WER reduction of 2.07%/0.45% compared with that of the multi-channel CNN-TDNNF-BLSTM. With the RNNLM rescore [27], the gap between the enhanced and the headset audios was 0.89%/4.67%, suggesting that the bottleneck of lowering the WER on the Dev. set may involve the acoustic and language models, whereas the Eval. set remains the potential of the front-end improvement.

6. Conclusions

In this paper, an ASR framework is proposed with the front-end processing of an improved GSS, the data augmentation via random channel selection, and a multi-channel CNNRes-BLSTM acoustic model. The experiments were conducted on the CHiME-6 dataset, which was recorded in a party scenario. The framework achieved a WER reduction of 17.02%/14.45% and 9.74%/10.35% on the Dev./Eval. set compared with the official baseline and our baseline, respectively. Meanwhile, the framework narrowed the gap between the headset audios and enhanced audios to 0.89%/4.67%. However, the WER results indicated the proposed framework performed worse on the Eval. set than the Dev. set. In addition, the improved GSS is rather complex to implement. Our future work is to explore the reason of the performance degradation on the Eval. set and to speed up the front-end processing.

7. References

[1] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[2] W. Xiong, J. Droppo, X. Huang, F. Seide, M. L. Seltzer, A. Stolcke, D. Yu, and G. Zweig, "Toward human parity in conversational speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 12, pp. 2410–2423, 2017.

[3] W. Xiong, L. Wu, F. Alleva, J. Droppo, X. Huang, and A. Stolcke, "The microsoft 2017 conversational speech recognition system," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5934–5938.

[4] J. Barker, S. Watanabe, E. Vincent, and J. Trmal, "The fifth 'chime' speech separation and recognition challenge: Dataset, task and baselines," in *INTERSPEECH*, 2018.

[5] S. Watanabe, M. Mandel, J. Barker, and E. Vincent, "Chime-6 challenge: Tackling multispeaker speech recognition for unsegmented recordings," *ArXiv*, vol. abs/2004.09249, 2020.

[6] N. Kanda, C. Böddeker, J. Heitkaemper, Y. Fujita, S. Horiguchi, K. Nagamatsu, and R. Haeb-Umbach, "Guided source separation meets a strong asr backend: Hitachi/paderborn university joint investigation for dinner party asr," *ArXiv*, vol. abs/1905.12230, 2019.

[7] N. Ito, S. Araki, and T. Nakatani, "Complex angular central gaussian mixture model for directional statistics in mask-based microphone array signal processing," *2016 24th European Signal Processing Conference (EUSIPCO)*, pp. 1153–1157, 2016.

[8] C. Zorila, C. Böddeker, R. Doddipatla, and R. Haeb-Umbach, "An investigation into the effectiveness of enhancement in asr training and test for chime-5 dinner party transcription," *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 47–53, 2019.

[9] D. Povey, G. Cheng, Y. Wang, K. Li, H. Xu, M. Yarmohammadi, and S. Khudanpur, "Semi-orthogonal low-rank matrix factorization for deep neural networks," in *INTERSPEECH*, 2018.

[10] https://chimechallenge.github.io/chime6/results.html.

[11] L. Sun, J. Du, T. Gao, Y. Fang, F. Ma, and C.-H. Lee, "A speaker-dependent approach to separation of far-field multi-talker microphone array speech for front-end processing in the chime-5 challenge," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, pp. 827–840, 2019.

[12] L. Sun, J. Du, L.-R. Dai, and C.-H. Lee, "Multiple-target deep learning for lstm-rnn based speech enhancement," *2017 Hands-free Speech Communications and Microphone Arrays (HSCMA)*, pp. 136–140, 2017.

[13] H. Erdogan, J. R. Hershey, S. Watanabe, and J. L. Roux, "Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks," *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 708–712, 2015.

[14] J. R. Hershey, Z. Chen, J. L. Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 31–35, 2016.

[15] A. Banerjee, I. S. Dhillon, J. Ghosh, and S. Sra, "Clustering on the unit hypersphere using von mises-fisher distributions," *J. Mach. Learn. Res.*, vol. 6, pp. 1345–1382, 2005.

[16] T. Yoshioka and T. Nakatani, "Generalization of multi-channel linear prediction methods for blind mimo impulse response shortening," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, pp. 2707–2720, 2012.

[17] L. Drude, J. Heymann, C. Böddeker, and R. Haeb-Umbach, "Nara-wpe: A python package for weighted prediction error dereverberation in numpy and tensorflow for online and offline processing," in *ITG Symposium on Speech Communication*, 2018.

[18] M. Souden, J. Benesty, and S. Affes, "On optimal frequency-domain multichannel linear filtering for noise reduction," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, pp. 260–276, 2010.

[19] C. Böddeker, J. Heitkaemper, J. Schmalenstroeer, L. Drude, J. Heymann, and R. Haeb-Umbach, "Front-end processing for the chime-5 dinner party scenario," in *INTERSPEECH 2018*, 2018.

[20] L. Drude and R. Haeb-Umbach, "Integration of neural networks and probabilistic spatial models for acoustic blind source separation," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, pp. 815–826, 2019.

[21] H. Erdogan, J. R. Hershey, S. Watanabe, M. I. Mandel, and J. L. Roux, "Improved mvdr beamforming using single-channel mask prediction networks," in *INTERSPEECH*, 2016.

[22] V. M. Tavakoli, J. R. Jensen, M. G. Christensen, and J. Benesty, "A framework for speech enhancement with ad hoc microphone arrays," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, pp. 1038–1051, 2016.

[23] 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," in *INTERSPEECH*, 2019.

[24] Z. qiu Wang and D. Wang, "On spatial features for supervised speech separation and its application to beamforming and robust asr," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5709–5713, 2018.

[25] N. Kanda, Y. Fujita, S. Horiguchi, R. Ikeshita, K. Nagamatsu, and S. Watanabe, "Acoustic modeling for distant multi-talker speech recognition with single- and multi-channel branches," *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6630–6634, 2019.

[26] V. Peddinti, G. Chen, V. Manohar, T. Ko, D. Povey, and S. Khudanpur, "Jhu aspire system: Robust lvcsr with tdnns, ivector adaptation and rnn-lms," *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 539–546, 2015.

[27] T. Mikolov, S. Kombrink, A. Deoras, L. Burget, and J. Cernocky, "Rnnlm - recurrent neural network language modeling toolkit," 2011.