8th International Workshop on Speech Processing in Everyday Environments (CHiME 2024)
6 September 2024, Kos, Greece

The NAIST System for the CHiME-8 NOTSOFAR-1 Task

Yuta Hirano[1], *Mau Nguyen*[2], *Kakeru Azuma*[1], *Jan Meyer Saragih*[1], *Sakriani Sakti*[1]

[1]Nara Institute of Science and Technology, Japan
[2]Japan Advanced Institute of Science and Technology, Japan

hirano.yuta.ia4@naist.ac.jp, maudn@jaist.ac.jp, azuma.kakeru.am9@naist.ac.jp,
jan.meyer_saragih.jn9@naist.ac.jp, ssakti@is.naist.jp

Abstract

This paper describes the NAIST system for the NOTSOFAR-1 (Natural Office Talkers in Settings Of Far-field Audio Recordings) task of the CHiME-8 challenge. Although there is a critical need for real-time processing in everyday applications, most evaluations in the CHiME challenge focus solely on reducing word error rate. Here, we aim to reduce inference speed while improving recognition accuracy. To tackle this issue, we propose enhancing the modular architecture of the baseline by modifying both the CSS and ASR modules. Specifically, our ASR module was built on WavLM-large feature extractor and Zipformer transducer. Additionally, we employed block-wise weighted prediction error (WPE) for dereverberation before the speech separation module. Our system achieved a relative tcpWER reduction of 11.6% over the baseline system in the single-channel track and 18.7% in the multi-channel track. Moreover, our system is two to six times faster than the baseline system while achieving better tcpWER results.

Index Terms: CHiME-8, NOTSOFAR-1, Multi-talker ASR, Zipformer, WavLM

1. Introduction

Thanks to recent advances in deep learning, the performance of automatic speech recognition (ASR) has reached human parity [1]. However, multi-talker ASR still remains a challenging task while the task is fundamental for applications in real world. The NOTSOFAR-1 [2] task of the CHiME-8 challenge focuses on developing multi-talker transcription systems, particularly for meeting transcription tasks, using information about the geometry of channels in microphones. The baseline system comprises three modules: continuous speech separation (CSS), automatic speech recognition (ASR), and speaker diarization. These modules are integrated into a pipeline system. The CSS module separates the input mixture audio into noise-free and overlap-free audio streams. The ASR module then receives these audio streams from the CSS module and transcribes them into word sequences with time boundaries for each word. Afterwards, the speaker diarization module assigns speaker labels to each word by applying spectral clustering to word-level speaker embeddings.

In this paper, we present the NAIST system for the NOTSOFAR-1 task of the CHiME-8 challenge. Our primary objective is to improve the transcription accuracy and inference speed of the ASR module. The system adopts a pipeline architecture similar to the baseline system but incorporates several significant enhancements. We replaced the Whisper large-v3 model [3] with a transducer model [4] featuring Zipformer [5], a powerful yet faster variant of Conformer [6] as shown in the paper [5]. Additionally, we employed WavLM-large [7] as a feature extractor to further boost recognition accuracy. For the multi-channel track, we applied dereverberation using block-wise weighted prediction error (WPE) [8] before CSS, as we observed that the audio streams from the Conformer CSS module still contained reverberation. The main contributions of this paper are summarized as follows.

- By adopting Zipformer transducer as a ASR model, we significantly reduced the inference time by up to 83.4%.
- We experimentally show that the representations from the last layer of WavLM-large can enhance ASR models enough outperform Whisper large-v3 with a limited amount of training data.
- We found that there is still room for improvement in terms of dereverberation in the baseline's Conformer CSS model, even though the model is trained on a large amount of data.

2. Proposed System

Figure 1 illustrates the architecture of our proposed system compared to the baseline system. While maintaining a modular architecture, our system significantly advances the CSS model and the ASR module to enhance recognition accuracy and inference speed. Table 1 shows the components of each system. Note that system 2 in the single-channel track and system 2 in the known-geometry multi-channel track have different components.

2.1. Continuous Speech Separation

In the speech separation module, we utilized the Conformer speech separation model provided by the baseline system. Upon reviewing the output from the Conformer CSS model, we discovered that the signals still contained reverberation, despite the model being trained to predict reverberation-free signals. To address this, we added a dereverberation module using block-wise weighted prediction error (WPE) [8] as a frontend module to the Conformer CSS model for the known-geometry multi-channel track (as shown in Table 1). We used nara-WPE [9] for this purpose.

Initially, we also applied WPE dereverberation to the single-channel track. However, we did not observe any improvement, possibly due to the single-channel audio recording devices already incorporating effective dereverberation functions. As a result, we opted to exclude WPE dereverberation from the single-channel track in our final system.

2.2. Automatic Speech Recognition

2.2.1. ASR Frontend

For the frontend of our ASR model, we utilized WavLM-large representations [7], as they have shown promising results in the

10.21437/CHiME.2024-13

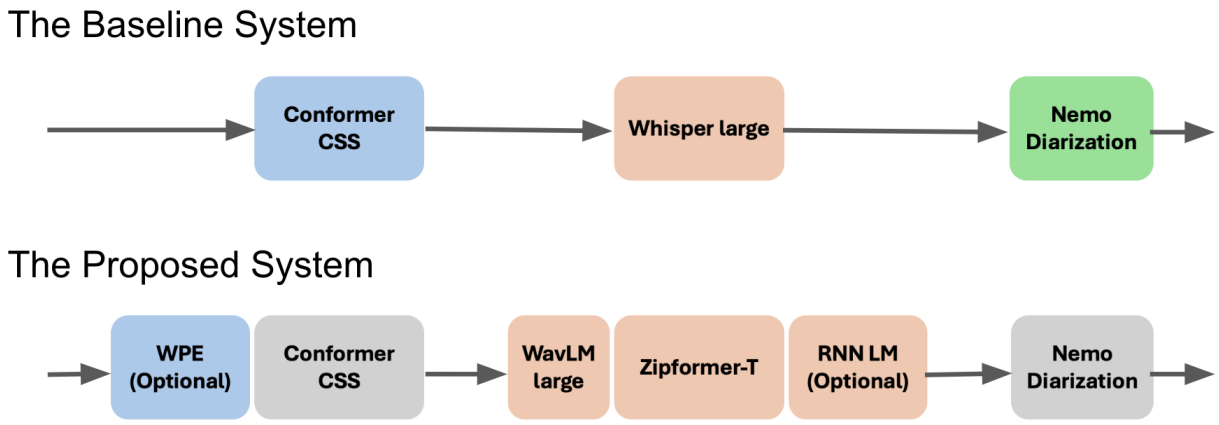

Figure 1: *The overall architecture of our proposed system, compared with the baseline system. It mainly consists of WavLM-large feature extractor and Zipformer transducer model. Unchanged modules from the baseline are grayed out. The optional WPE and RNN-LM modules are included in the System 2 version.*

Table 1: *Comparison of our submitted systems. ✓ represents "used" and blank space represents "not used."*

Component	Single-channel track		Multi-channel track	
	System 1	System 2	System 1	System 2
WPE				✓
Conformer CSS	✓	✓	✓	✓
WavLM-large	✓	✓	✓	✓
Zipformer transducer	✓	✓	✓	✓
RNN LM		✓		
Nemo word nmesc diarizer	✓	✓	✓	✓

Table 2: *WER [%] on the eval-set of AMI corpus. Whisper large was used without fine-tuning or adaptation.*

Model	Parameters	IHM	MDM
Zipformer (Fbank)	70.4 M	17.6	32.8
Zipformer (WavLM-large 24th)	70.0 M	13.2	26.3
Whisper large [3]	1.6 B	16.9	36.4

CHiME-7 challenge [10]. Typically, self-supervised learning (SSL) representations for downstream tasks involve a weighted sum with learnable weights, but we chose to use only the output from the last layer of WavLM-large. This decision was based on our preliminary experiments on the AMI corpus [11, 12], which revealed that the output from the last layer significantly improved recognition accuracy and outperformed Whisper, the baseline ASR module. Table 2 shows the result of our preliminary experiments. Zipformer models were trained on the augmented version of AMI corpus which is described in the section 3.1. Notably, we did not update any parameters of WavLM-large during training and inference.

2.2.2. *ASR Model*

For our ASR model, we adopted the Zipformer transducer model [5] due to its enhanced speed, reduced memory usage, and superior performance compared to traditional Transformers. Table 2 shows the structure of our Zipformer encoder. Unlike the Conformer, which processes sequences at a constant frame rate, the Zipformer utilizes a U-Net-like [13] architecture. Each Zipformer block runs in a different frame rate by operating down-sampling and up-sampling. This approach greatly improves the model's overall efficiency.

The model we utilized is based on a recipe[1] from the Icefall toolkit. It consists of five Zipformer layers as the encoder and a stateless decoder [14]. Pruned transducer loss [15] was used for training. We replaced the convolution subsampling module with a linear layer to enable the model to encode WavLM-large representations.

In the inference phase, we employed block-wise beam search decoding with a beam size of 8 on the output signals from the CSS module. For block-wise inference, we segmented the input audio streams into 40-second chunks with a 4-second overlap. After obtaining recognition results for all chunks, we assembled them based on word start timestamps derived from decoding and referencing input audio frame indexes. Once all chunk transcriptions were combined, we determined the end timestamp for each word. Typically, we used the start timestamp of the subsequent word as the end timestamp for the current word. However, this method could result in excessively long word durations if the next word followed a prolonged silence, leading to out-of-memory issues or suboptimal performance during speaker diarization inference. To address this, we limited the word duration to 1.0 second if the difference between the current word's start time and the next word's start time exceeded 1.0 second.

In addition to the above system, we also performed shallow fusion using a recurrent neural network language model

[1] https://github.com/k2-fsa/icefall/tree/master/egs/librispeech/ASR/pruned_transducer_stateless7

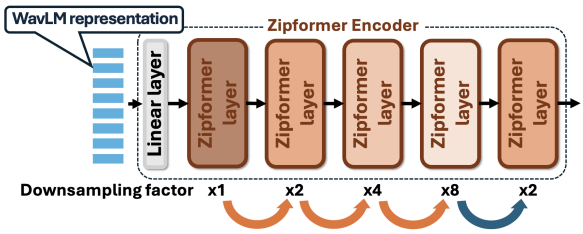

Figure 2: *The structure of our Zipformer encoder. Color intensity represents high frame rate. The convolution subsampling module was replaced with a linear layer.*

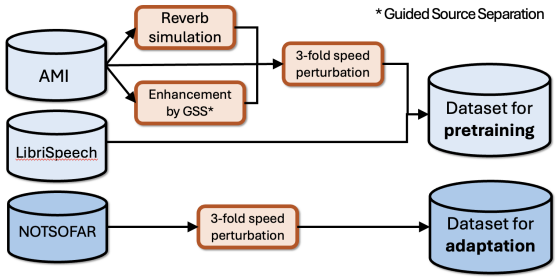

Figure 3: *The overall flow of data augmentation for training of our ASR model.*

(RNN LM). Our RNN LM consists of three layers of 2084-dimensional LSTMs [16]. The system that includes the RNN LM is referred to as System 2.

2.3. Speaker Diarization

We utilized the speaker diarization module from the baseline system. Given the transcription results with word boundaries from the ASR module, the speaker diarization module assigns speaker labels to each word by clustering the speaker embeddings using the normalized maximum eigengap-based spectral clustering (NME-SC) algorithm [17]. The speaker embeddings are extracted using the large TitaNet model [18] based on the word boundaries.

3. Experimental Setup

3.1. Data

The NOTSOFAR-1 task provides two new benchmarking datasets: natural meeting recordings and simulated training datasets. Participants can utilize the data available in the simulated training set, the training portion of the benchmarking dataset, as well as commonly used open-source datasets and pre-trained models defined in their rules. Here, we only utilized the first benchmark dataset of natural meeting recordings and two external datasets, LibriSpeech [19] and AMI [11, 12], from the list of allowed external data.

The LibriSpeech corpus comprises 960 hours of speech derived from read audiobooks from the LibriVox project, while the AMI corpus consists of 100 hours of meeting recordings with three types of audio data: IHM, SDM, and MDM. IHM (Individual Headset Microphone) data consists of audio recorded using microphones placed near the mouth. SDM (Single Distant Microphone) data consists of audio recorded using a single microphone placed further from the speakers. MDM (Multiple Distant Microphones) data consists of audio recorded using multiple microphones positioned at a distance from the speakers.

Note that the core datasets of the DASR task [20] (CHiME6 [21] , DiPCo [22], Mixer6 [23]) were not used in our experiment because they are not listed in the allowed external data section of the CHiME8 DASR webpage, which we interpret to mean they are forbidden.

3.2. Data Augmentation

Figure 3 shows the flow of data augmentation for training of our ASR model. To increase the amount of meeting recording data, we applied several data augmentation methods to the AMI corpus, while using the 960 hours of LibriSpeech data as is. The data augmentation methods were as follows. First, we added simulated reverberation to the IHM audio data using the lhotse library [24]. Second, we applied audio enhancement using GPU-accelerated guided source separation [25] on the MDM audio data. Third, we performed 3-fold speed perturbation on the entire AMI corpus, including both the reverberated IHM data and the GSS-enhanced MDM data. We used 500 byte-pair encoding (BPE) sub-word units generated from Librispeech and AMI as output token. Finally, we applied SpecAugment [26] on Librispeech, AMI, and NOTSOFAR-1 meeting dataset. After data augmentation, the size of AMI corpus increased to 881 hours, which is nearly same amount as LibriSpeech. All of the text data for supervision was normalised before model training using the text normalizer in the baseline implementation [2].

3.3. Training Setup

We first pre-trained our Zipformer transducer model on the Librispeech and augmented AMI datasets. The learning rate was warmed up to 0.05 for 2000 iteration. The models to fine-tune was selected based on time-constrained minimum permutation word error rate (tcpWER) on the dev-set-2. Afterwards, we fine-tuned the model on the augmented NOTSOFAR-1 meeting dataset. On the other hand, our RNN LM was trained exclusively on the LibriSpeech dataset. The same learning rate setting as pre-training was used for fine-tuning. All model components were trained using two NVIDIA A100 40GB GPUs.

4. Experimental Results

Following the challenge's rules, time-constrained minimum permutation word error rate (tcpWER) was used for evaluation. Unlike ordinary WER, tcpWER considers speaker-attribution and temporal annotation, making it more suitable for evaluating meeting transcription systems. Additionally, we provide the total inference time of each system to measure efficiency.

4.1. Task1: Single-Channel Track

Figure 4 shows the total inference time vs. tcpWER comparison for different systems: baseline, System 1, and System 2 in the single-channel scenario. An NVIDIA A10 24GB GPU was used for evaluation. By replacing Whisper with WavLM-large and Zipformer transducer, System 1 achieved a 11.1% relative tcpWER reduction and a 6-fold increase in inference speed. This result indicates that over 80% of the baseline system's inference time is consumed by its ASR module. Con-

[2]https://github.com/microsoft/NOTSOFAR1-Challenge/tree/main/utils/text_norm_whisper_like

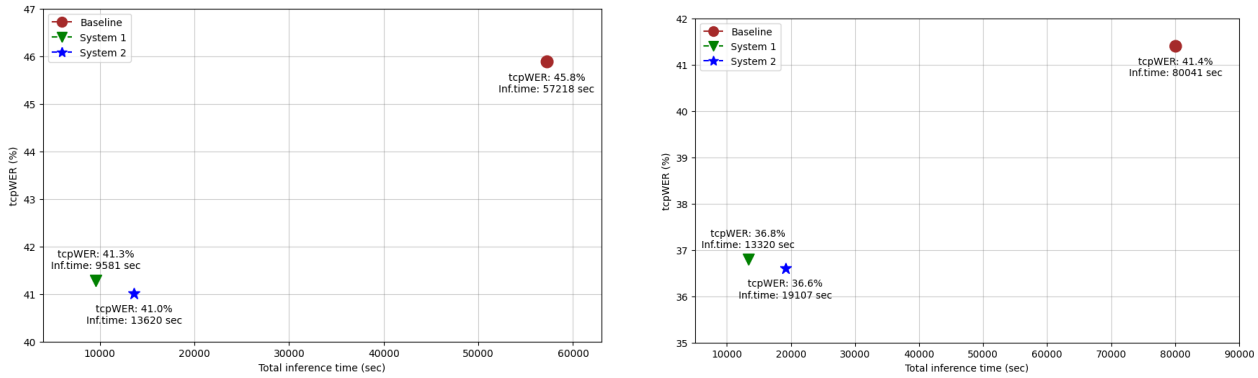

Figure 4: *The total inference time vs. tcpWER comparison for different models: baseline, System 1, and System 2 in the* ***single-channel track****. The left figure shows the results on the dev-set-2, and the right figure shows the results on the eval-set.*

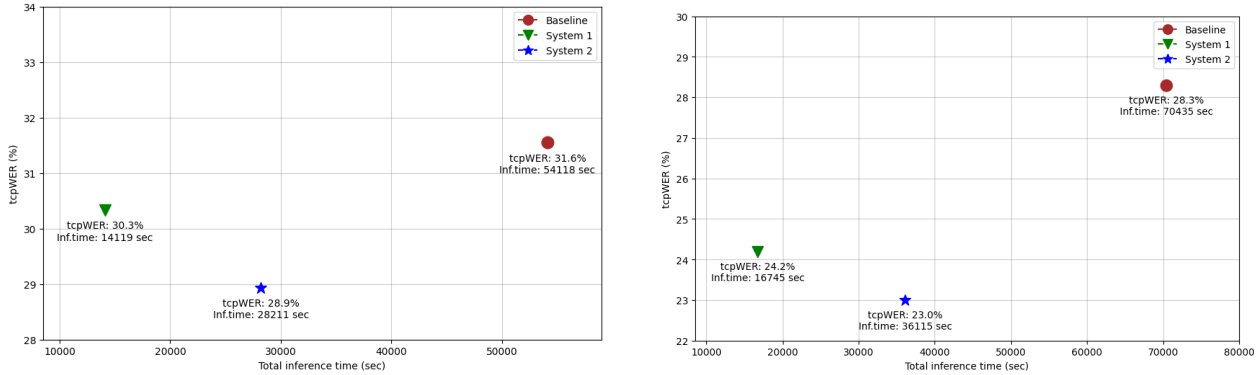

Figure 5: *The total inference time vs. tcpWER comparison for different models: baseline, System 1, and System 2 in the* ***known-geometry multi-channel track****.The left figure shows the results on the dev-set-2, and the right figure shows the results on the eval-set.*

sidering the performance gap between the baseline system and our proposed system on the dev-set-2 and eval-set, our system successfully avoided overfitting to dev-set-2.

By applying shallow fusion with the RNN LM, System 2 achieved slightly better tcpWER than System 1. This modest improvement may be attributed to a domain mismatch between the training and evaluation data. The System2 resulted in 3th place out of 6 teams in the official ranking.

4.2. Task2: Known-Geometry Multi-Channel Track

Figure 5 shows the total inference time vs. tcpWER comparison for different systems: baseline, System 1, and System 2 in the known-geometry multi-channel scenario. An NVIDIA A10 24GB GPU is used for evaluation as well as the single-channel track. By replacing Whisper by WavLM-large and Zipformer transducer, the System 1 achieved a relative tcpWER reduction of 14.5% and 4.2 times faster inference speed. By applying dereverberation by WPE, System 2 achieved a relative tcpWER reduction of 18.7% from the baseline system and resulted in 4th place out of 10 teams in the official ranking. Since a large number of channels in the microphone causes long inference time for WPE, System 2 took twice as long for inference as System 1. However, System 2 still maintained a much faster inference speed compared to the baseline system. Further reduction of inference time could be achieved by GPU-accelerated version of WPE [27]. Different from the result on the single-channel track, our system for the known-geometry multi-channel track worked very well in terms of tcpWER on the eval-set, while the system showed relatively limited improvement on the dev-set-2. Shallow fusion with the RNN LM did not provide any performance improvement in this multi-channel track.

5. Conclusion

We presented the NAIST system for the NOTSOFAR-1 task of the CHiME-8 challege. By adopting WavLM-large as the ASR frontend and the Zipformer transducer as the ASR model, respectively, we achieved both better tcpWER and significantly faster inference speed compared to the baseline system. However, we observed limited dereverberation capability in the Conformer CSS model, indicating that exploring more powerful speech separation models for CSS modules (e.g. TF-GridNet [28]) is a promising direction for future work. Additionally, we found data augmentation to be quite effective and will conduct ablation studies to identify which techniques had the most significant impact on enhancing system performance.

6. Acknowledgements

Part of this work is supported by JSPS KAKENHI Grant Numbers JP21H05054 and JP23K21681

7. References

[1] J. Li *et al.*, "Recent advances in end-to-end automatic speech recognition," *APSIPA Transactions on Signal and Information Processing*, vol. 11, no. 1, 2022.

[2] A. Vinnikov, A. Ivry, A. Hurvitz, I. Abramovski, S. Koubi, I. Gurvich, S. Peer, X. Xiao, B. M. Elizalde, N. Kanda, X. Wang, S. Shaer, S. Yagev, Y. Asher, S. Sivasankaran, Y. Gong, M. Tang, H. Wang, and E. Krupka, "Notsofar-1 challenge: New datasets, baseline, and tasks for distant meeting transcription," in *Interspeech 2024*, 2024, pp. 5003–5007.

[3] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *International conference on machine learning*. PMLR, 2023, pp. 28 492–28 518.

[4] A. Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.

[5] Z. Yao, L. Guo, X. Yang, W. Kang, F. Kuang, Y. Yang, Z. Jin, L. Lin, and D. Povey, "Zipformer: A faster and better encoder for automatic speech recognition," in *The Twelfth International Conference on Learning Representations*, 2023.

[6] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *arXiv preprint arXiv:2005.08100*, 2020.

[7] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao *et al.*, "Wavlm: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.

[8] T. Nakatani, T. Yoshioka, K. Kinoshita, M. Miyoshi, and B.-H. 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.

[9] 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 *Speech Communication; 13th ITG-Symposium*. VDE, 2018, pp. 1–5.

[10] S. Cornell, M. Wiesner, S. Watanabe, D. Raj, X. Chang, P. Garcia, Y. Masuyama, Z.-Q. Wang, S. Squartini, and S. Khudanpur, "The chime-7 dasr challenge: Distant meeting transcription with multiple devices in diverse scenarios," *arXiv preprint arXiv:2306.13734*, 2023.

[11] J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal *et al.*, "The ami meeting corpus: A pre-announcement," in *International workshop on machine learning for multimodal interaction*. Springer, 2005, pp. 28–39.

[12] W. Kraaij, T. Hain, M. Lincoln, and W. Post, "The ami meeting corpus," in *Proc. International Conference on Methods and Techniques in Behavioral Research*, 2005, pp. 1–4.

[13] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer, 2015, pp. 234–241.

[14] M. Ghodsi, X. Liu, J. Apfel, R. Cabrera, and E. Weinstein, "Rnn-transducer with stateless prediction network," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7049–7053.

[15] F. Kuang, L. Guo, W. Kang, L. Lin, M. Luo, Z. Yao, and D. Povey, "Pruned rnn-t for fast, memory-efficient asr training," *arXiv preprint arXiv:2206.13236*, 2022.

[16] S. Hochreiter, "Long short-term memory," *Neural Computation MIT-Press*, 1997.

[17] T. J. Park, K. J. Han, M. Kumar, and S. Narayanan, "Auto-tuning spectral clustering for speaker diarization using normalized maximum eigengap," *IEEE Signal Processing Letters*, vol. 27, pp. 381–385, 2019.

[18] N. R. Koluguri, T. Park, and B. Ginsburg, "Titanet: Neural model for speaker representation with 1d depth-wise separable convolutions and global context," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 8102–8106.

[19] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 2015, pp. 5206–5210.

[20] S. Cornell, T. Park, S. Huang, C. Boeddeker, X. Chang, M. Maciejewski, M. Wiesner, P. Garcia, and S. Watanabe, "The chime-8 dasr challenge for generalizable and array agnostic distant automatic speech recognition and diarization," *arXiv preprint arXiv:2407.16447*, 2024.

[21] S. Watanabe, M. Mandel, J. Barker, E. Vincent, A. Arora, X. Chang, S. Khudanpur, V. Manohar, D. Povey, D. Raj *et al.*, "Chime-6 challenge: Tackling multispeaker speech recognition for unsegmented recordings," *arXiv preprint arXiv:2004.09249*, 2020.

[22] M. Van Segbroeck, A. Zaid, K. Kutsenko, C. Huerta, T. Nguyen, X. Luo, B. Hoffmeister, J. Trmal, M. Omologo, and R. Maas, "Dipco–dinner party corpus," *arXiv preprint arXiv:1909.13447*, 2019.

[23] L. Brandschain, D. Graff, C. Cieri, K. Walker, C. Caruso, and A. Neely, "Mixer 6." 01 2010.

[24] P. Żelasko, D. Povey, J. Trmal, S. Khudanpur *et al.*, "Lhotse: a speech data representation library for the modern deep learning ecosystem," *arXiv preprint arXiv:2110.12561*, 2021.

[25] D. Raj, D. Povey, and S. Khudanpur, "Gpu-accelerated guided source separation for meeting transcription," *arXiv preprint arXiv:2212.05271*, 2022.

[26] 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.

[27] D. Raj. (2022) Gpu-accelerated guided source separation, github. [Online]. Available: https://github.com/desh2608/gss/blob/master/gss/wpe/wpe.py

[28] Z.-Q. Wang, S. Cornell, S. Choi, Y. Lee, B.-Y. Kim, and S. Watanabe, "Tf-gridnet: Integrating full-and sub-band modeling for speech separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.