NTT Multi-Speaker ASR System for the DASR Task of CHiME-7 Challenge

*Naoyuki Kamo**, *Naohiro Tawara**, *Kohei Matsuura**, *Takanori Ashihara**, *Takafumi Moriya**,
Atsunori Ogawa, Hiroshi Sato, Tsubasa Ochiai, Atsushi Ando, Rintaro Ikeshita, Takatomo Kanou,
Marc Delcroix, Tomohiro Nakatani, Taichi Asami, Shoko Araki

NTT Corporation, Japan

{naoyuki.kamo, naohiro.tawara, kohei.matsuura, takanori.ashihara, takafumi.moriya, marc.delcroix}@ntt.com

Abstract

We introduce our submission to the Distant automatic speech recognition (DSAR) task of the CHiME 7 challenge. Our system uses end-to-end diarization with vector clustering (EEND-VC), guided source separation (GSS), and attention-based encoder-decoder and transducer-based ASR systems. Our submission exploits pre-trained self-supervised learning (SSL) models to build strong diarization and ASR modules. We also explore data augmentation using contrastive data selection based on representations from SSL models. Besides, we use self-supervised adaptation (SSA) to adapt these modules to the recording conditions of each session. Our DASR system achieves a 36 % diarization error rate (DER) reduction and 47 % word error rate reduction (WER) over the baseline on the main track of the evaluation set and ranked third in the challenge.

Index Terms: Robust ASR, speaker diarization, CHiME-7 DASR

1. Introduction

Recognizing conversational speech captured by distant microphones remains one of the major challenges for automatic speech recognition research (ASR). The CHiME challenge series has provided, over the years, datasets with increased levels of difficulty to measure the progress in the field of distant ASR (DASR), starting with single-talker DASR tasks in CHiME 1-4 [1, 2], then multi-talker tasks in CHiME challenges 5-6 [3, 4]. However, the evaluation data of the past editions covered relatively homogeneous recording conditions. CHiME-7 increased the difficulty by requiring the design of a single DASR system capable of handling multiple recording conditions varying in terms of the microphone array used (number and configuration) and the type of conversations (multi-talker home recordings and 2-speaker interviews) and recording environments.

In this paper, we propose a multi-talker DASR system for the CHiME-7 DASR task. A multi-talker DASR system requires identifying when each speaker speaks and transcribing the speech correctly even if multiple speakers speak at the same time and there is noise and reverberation. The problem requires combining speech enhancement (dereverberation, separation, and denoising), diarization, and ASR. Our proposed system follows a similar pipeline as the baseline system [5], i.e., speaker diarization, followed by speech enhancement (SE) with guided source separation (GSS) and then ASR. However, we replaced the baseline diarization system with an end-to-end diarization with vector clustering (EEND-VC) [6, 7] system and developed four powerful ASR back-ends.

Our system has the following characteristics:

1. **Robustness to recording conditions:** We exploit pre-trained self-supervised learning (SSL) models to build strong diarization and ASR systems. We used WavLM, which is trained on large amounts of noisy speech data, to extract features for the diarization and ASR modules. In addition, we also employ a pre-trained ECAPA-TDNN model to extract robust speaker embeddings for diarization.
2. **Adaptation to recording conditions:** We use self-supervised adaptation (SSA) for diarization and ASR to adapt to the recording conditions of each session. We use pseudo-labels generated from the combination of multiple systems or microphones.
3. **Independence to the microphone array configuration:** We use single-channel diarization and DOVER-LAP to combine the results of each channel. This allows exploiting multiple channels while keeping the diarization independent of the number and configuration of the microphones. Enhancement is performed with WPE and GSS using all microphones available, independently of the array configuration. Finally, ASR is performed on the single-channel output of GSS.
4. **Handling varying number of speakers:** EEND-VC can handle an arbitrary number of speakers as long as the maximum number of speakers in a segment can be fixed. Usually, the segment length has to be relatively short (a few seconds) to ensure that no more than the maximum number of speakers are active. Here, the characteristics of the CHiME-7 DASR task, which consists of conversations of up to four speakers, allow us to use much longer segments for diarization, i.e., 80 sec, by fixing the maximum number of speakers to four.

In addition, we employed a two-step training strategy for training the SSL-based diarization and ASR modules. We first trained the downstream models with fixed SSL model parameters and then retrained the whole system, including the SSL model parameters. Moreover, we also employed contrastive training data selection based on SSL models to increase the amount of training data [8]. We report detailed experiments showing the contribution of the different components of each module to justify our design choices.

2. System overview

Figure 1 is a schematic diagram of the proposed DASR system. As shown in the figure, it follows a similar processing flow as the baseline. However, we modified the different components. The details of the diarization and ASR modules are described in Section 3 and 4.

*Equal contribution

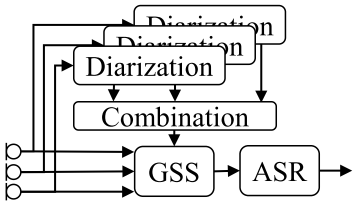

Figure 1: *Proposed recognition system for DASR track.*

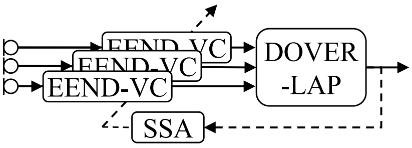

Figure 2: *Diarization module.*

For speech enhancement, we used the baseline GSS system [9]. We set the filter taps of WPE at five and the iterations of cACGMM [10] optimization with diarization guidance at 10. We call this system **SE1**. We also used WPE before diarization.

3. Diarization with EEND-VC

Our diarization system is shown in Fig 2. It processes each channel separately using EEND-VC [6, 7], and then combines the output using diarization output voting error reduction + Overlap (DOVER-LAP) [11]. We base our implementation on the publicly available implementation of EEND-VC[1] and DOVER-LAP.[2] EEND-VC estimates speaker activities and speaker embeddings on speech segments of a long recording and then clusters the speaker embeddings to stitch the segments together to form the diarization results.

We trained EEND-VC to handle up to four speakers in each segment. This enabled us to process long segments of 80 sec, which provided sufficient data to generate reliable speaker embeddings. We call this system **DIA1**. To further reduce speaker confusion, we used the pre-trained ECAPA-TDNN model [12] employed in the baseline system [5] to extract embeddings for each speaker in the segment, given the speech activity from EEND-VC. Here, we removed the speech overlapping regions to reduce speaker confusion. We call this system **DIA2**. Finally, we performed self-supervised adaptation (SSA) on each session to retrain the EEND-VC model using labels from the combination of the diarization results from all channels. We call these systems **DIA3** and **DIA4**. The details of our diarization systems are described in [13].

3.1. System configuration

We used the pre-trained WavLM-large to obtain the input speech features [7, 14]. The features consisted of the learnable weighted sum of all transformer layers of the WavLM, including weights for each feature dimension. The EEND-VC model consisted of six-stacked Transformer encoder blocks with eight attention heads. The input feature dimension was 1024, and the output dimension for each attention block was 256. We projected the encoder's output with a linear layer into four output streams, each consisting of the frame-by-frame speaker activity binary decisions and the speaker embedding of 256 dimensions.

We used a two-stage approach to train our system. First, we trained the system using simulated multi-talker recordings of up to four speakers, using the LibriSpeech [16], MUSAN for noise [17], and room impulse responses (RIRs) from SLR28 [18]. The simulated training data comprised 100,000 speech mixtures, 12,685 hours, and 2,338 speakers. We then retrained the model using randomly selected channels from the CHiME-6, Mixer 6 training sets, and part of DiPCo dev set.[3] This amounted to 182 recordings, 80.3 hours, and 114 speakers.

For the first training stage, we fixed the WavLM parameters and trained the model for 25 epochs with a learning rate of 10^{-3}, a batch size of 2048, and a segment size of 15 sec. For the second stage, we trained the full model (including WavLM) on segments of 80 sec, for three epochs with a learning rate of 10^{-5} and a batch size of one. For SSA, we used the labels obtained with the EEND-VC w/ ECAPA model and retrained the model for each session and each microphone independently for one epoch with a learning rate of 10^{-5}.

3.2. Diarization Results

Table 1 compares the performance of our proposed diarization systems with the baseline on the development set. For DIPCO, we report in parenthesis results on sessions S26 and S29, which were not used during training.

DIA1, which uses the original EEND-VC, significantly reduces the DER over the baseline for the three datasets. However, it still has relatively high speaker confusion errors, especially for the CHiME-6 data. This shows that the speaker embeddings of EEND-VC are not sufficiently discriminative. We can significantly reduce speaker confusion errors with DIA2, which replaces the embeddings of EEND-VC with those computed with the pre-trained ECAPA-TDNN model. The ECAPA-TDNN model has seen many more speakers during training than EEND-VC, which may explain the improved speaker discrimination. We used the results of DIA2 to generate pseudo-labels for SSA for DIA3. DIA3 also uses WPE as pre-processing to improve the segmentation. Finally, the best results were obtained with DIA4, which uses ECAPA-TDNN embeddings with the adapted system. It achieved a 35 % relative DER improvement over the baseline. We used DIA4 in our submission.

Table 2 shows the DER on the eval set. We observe a similar trend. Our system also achieved third place in terms of DER.

4. ASR back-end

Figure 3 shows the configuration of the ASR system. We developed four ASR back-ends that use WavLM-Large as the upstream model. These four systems differ in the configuration of the downstream model. Interestingly, the four ASR back-ends achieve similar performance, and combining these systems improves performance. We also used a Transformer-based language model (LM), which was used for N-best re-scoring. Finally, we performed session-wise SSA.

[1] https://github.com/nttcslab-sp/EEND-vector-clustering

[2] https://github.com/desh2608/dover-lap

[3] We used the Mixer 6 interview set, which contains labels for the interviewee only. We thus applied an early-stage diarization system on the lapel mics to generate activity references for the interviewer. Our training set also included sessions S28, S33, and S34 of DiPCo dev set.

Table 1: *Diarization results in terms of confusion (CF), false alarm (FA), missed (MI), and DER computed with md-eval with collar of 0.25 sec on dev set. We used WPE [15] pre-processing for DIA 3 and 4. The numbers in parenthesis are for sessions S26, S29 of DiPCo not used for training.*

		CHiME-6				DiPCo (S26, S29)				Mixer 6				Macro
ID	Model	CF	FA	MI	DER	CF	FA	MI	DER	CF	FA	MI	DER	DER
	Baseline	14.5	3.2	22.3	40.0	13.0	4.7	12.0	29.8 (29.9)	1.7	1.0	13.8	16.6	28.8
DIA1	EEND-VC	15.1	3.6	17.9	36.6	8.5	5.0	9.4	22.8 (20.7)	0.6	1.7	8.0	10.3	23.2
DIA2	DIA1 w/ ECAPA	8.8	3.9	17.5	30.1	5.7	5.3	9.2	20.2 (20.6)	1.3	1.7	8.1	11.0	20.4
DIA3	DIA2 + SSA + WPE	15.4	3.8	16.4	35.7	7.1	3.8	8.8	19.7 (18.2)	0.2	1.8	7.6	9.6	21.7
DIA4	DIA3 w/ ECAPA	8.1	4.0	16.2	28.2	4.7	4.0	8.8	17.5 (17.6)	0.2	1.8	7.7	9.7	18.5

Table 2: *Diarization results on eval set.*

		CHiME-6				DiPCo				Mixer 6				Macro
ID	Model	CF	FA	MI	DER	CF	FA	MI	DER	CF	FA	MI	DER	DER
	Baseline	28.7	11.2	16.4	56.3	10.2	5.8	11.8	27.9	1.5	1.0	6.9	9.3	31.2
DIA1	EEND-VC	16.7	3.1	20.0	39.8	8.5	9.1	9.3	26.9	1.4	1.6	3.9	6.8	24.5
DIA2	DIA1 w/ ECAPA	11.4	3.5	19.1	34.0	6.3	9.4	9.1	24.8	0.3	1.6	3.9	5.7	21.5
DIA3	DIA2 + SSA + WPE	13.4	2.1	20.3	35.9	6.6	6.9	9.1	22.6	0.1	1.6	3.9	5.6	21.4
DIA4	DIA3 w/ ECAPA	10.0	2.3	19.7	32.0	5.9	7.1	9.0	22.0	0.1	1.6	3.9	5.6	19.9

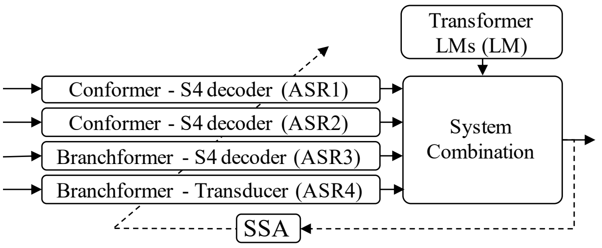
Transformer LMs (LM)
Conformer - S4 decoder (ASR1)
Conformer - S4 decoder (ASR2)
Branchformer - S4 decoder (ASR3)
Branchformer - Transducer (ASR4)
System Combination
SSA

Figure 3: *ASR module.*

4.1. Training data

We created several training sets for ASR; the first includes only the 91 hours of CHiME-7 training data (processed with GSS) [5], the second adds 960 hours of LibriSpeech training data [16], and the third also includes between 655 and 912 hours of VoxCeleb1+2 training data [19]. We selected the samples from the VoxCeleb dataset closest to the CHiME-6 data using the data selection algorithm that relies on discrete speech representation [8], which we describe in more detail below. Since VoxCeleb data is not transcribed, we used the Conformer-CTC model provided by NeMo [20][4] to generate the transcriptions. We did not use the CHiME-7 dev sets to train the ASR backends. Other data preparation procedures followed the CHiME-7 baseline.

To leverage the VoxCeleb1+2 [19] data, we utilized the contrastive data selection algorithm in an unsupervised fashion [8]. This algorithm uses a general and a target LM to select data with high domain relevance scores on the target domain and low scores on the general domain. The LMs consist of 5-gram LMs trained on discrete acoustic units. We use Kneser-Ney smoothing during training. To generate the acoustic units, we discretized the hidden features of the 21st WavLM layer instead of using the w2v-BERT codebook as proposed in [8]. For the quantization, we run the k-means algorithm with 1024 clusters. To train the two LMs, we assigned all CHiME-7 training and development sets to the target domain and the VoxCeleb1+2 data to the general domain. We calculated the domain relevance score by measuring the probability differences between the target domain LM and the general domain LM to reduce the VoxCeleb1+2 data by a quarter, a choice made through experimentation.

We also levaraged the VoxCeleb1+2 transcriptions for training the Transformer-based LMs. We first trained a long short-term memory (LSTM)-based LM, which has the standard configuration[5], using the CHiME-7 training data. By using this LSTMLM, we calculated perplexities (PPLs) for each of the transcriptions. Then, by setting a PPL threshold, we selected the transcriptions that showed lower PPLs, i.e., the transcriptions that have similar characteristics to the CHiME-7 training data. Hereafter, we refer to the selected transcriptions as the VoxCelebLM data. Finally, by using the CHiME-7 & VoxCelebLM data, we trained Transformer-based LMs. The PPL threshold was optimized based on the Transformer-based LMs' PPLs for the CHiME-7 dev data.

4.2. Two-step training scheme

In this work, we found a novel training scheme that consisted of two steps. All ASR models used WavLM-Large for feature extractor, followed by four types of downstream models(see Section 4.3). First, we pre-trained ASR models while freezing SSL modules. We used Adam optimizer within the Noam scheduler with 40k warm-up steps, and the maximum learning rate was set to 2.5×10^{-3}. Then, the unfrozen SSL and the pre-trained ASR model were jointly fine-tuned with the ReduceLROnPlateau scheduler. That is, we halved the learning rate starting at 1×10^{-5} if the validation loss was not improved. We used early stopping with a patience of 5 for both training steps.

[4] https://huggingface.co/nvidia/stt_en_conformer_ctc_large

[5] https://github.com/espnet/espnet/blob/master/egs2/chime7_task1/asr1/conf/train_lm.yaml

4.3. Configuration of ASR back-ends

4.3.1. Conformer encoder - S4 decoder (ASR1&2):

The first two systems use a conformer encoder [21] and state-space (S4) [22] blocks for the decoder. The conformer configuration is identical to the *base model* described in Section 5.2 of [23]. We followed the S4 decoder configuration and the optimization setup of ESPnet's recipe[6]. We used the weighted sum of the layers of the WavLM models as input features. We averaged 5-best checkpoints for evaluation and decoded transcriptions with a beam width of 20. ASR 1 and 2 models are first trained using CHiME-7& Librispeech data in step 1 and then in step 2 with CHiME-7 data for ASR1 and CHiME-7&VoxCeleb data for ASR2.

4.3.2. Branchformer encoder - S4 decoder (ASR3):

For the third model (ASR3), we replaced the conformer of ASR 1 and 2 with a Branchformer encoder [24]. The configuration follows ESPnet's recipe [7], except we set the dimension of the convolutional gating MLP (cgMLP) to 2048. To focus on the linguistic information most relevant to ASR rather than the diverse representation [14, 25], we used the concatenation of the outputs of the 21st and 22nd layers of the WavLM model as input features instead of the weighted sum of all layers. This model was trained with CHiME-7&LibriSpeech&VoxCeleb in both training steps.

4.3.3. Branchformer-Transducer (ASR4):

The last model also uses a Branchformer encoder but a transducer decoder [26]. The ASR encoder contains two-layer 2D-CNNs followed by 18 Branchformer blocks [24] with the cgMLP linear layer of 2048 units. The ASR encoder received the weighted sum of all WavLM layer outputs. The prediction and joint networks had two-layer 640-dimensional LSTMs and a 512-dimensional feed-forward network, respectively. Our transducer model was optimized by the combined loss [27], which consisted of RNNT [26], CTC [28], and internal LM (ILM) [29] training objectives. The weights of CTC and ILM losses in both stages were set to 0.5 and 0.1, respectively. Scheduled Sampling (SS) using ILM [27] was applied to reduce exposure bias in both steps. We used CHiME-7 and LibriSpeech datasets for step 1 and added the selected VoxCeleb1+2 data to them for step 2. Note that we used the top one-third of the selected VoxCeleb1+2 data for step 2. The batch sizes in the first and second stages were set to 64 and 32, respectively.

4.3.4. Transformer-LMs (LM):

We performed LM rescoring using Transformer-based LMs trained on CHiME-7 & VoxCelebLM data. We adopted two LMs, i.e., the forward and backward Transformer-based LMs [30]. Both models have a standard configuration[8], except that we reduced the number of layers to 8 and the number of units to 1024.

Table 3: *Effect of data augmentation with contrastive data selection from VoxCeleb1+2. Note that we here report WERs (not DA-WERs).*

VoxCeleb1+2	CHiME-6	DiPCo	Mixer 6	Macro
-	28.9	29.6	16.9	25.1
all	28.5	29.6	17.6	25.2
quarter	29.1	29.1	16.3	24.8

4.4. Decoding and rescoring settings

The diarization sometimes produces long speech segments. This affects the decoding speed. We thus applied the VAD from pyanote [31, 32] on the enhanced speech to cut segments longer than 60 sec into shorter ones before ASR.

In the decoding step, we generated 32-best hypotheses for each ASR system and performed LM rescoring using forward and backward LMs. Thus, the final hypotheses were determined by the best score for each utterance. Each hypothesis generated from each ASR back-end with forward and backward LMs is determined as follows:

$$\begin{aligned}\hat{Y} &= \arg\max_{Y} \Big\{ \log p_{\text{ASR}}(Y|\boldsymbol{X}) + \mu_1 \log p_{\text{FLM}}(Y) \\ &+ \mu_2 \log p_{\text{BLM}}(Y) + \mu_3 \, |Y| \Big\}, \end{aligned} \tag{1}$$

where $\boldsymbol{X}$ is the input speech, Y is the hypothesis, $\log\ p_{\text{ASR}}(Y|\boldsymbol{X})$ is the ASR model score for Y given $\boldsymbol{X}$, $\log\ p_{\text{FLM}}(Y)$ and $\log\ p_{\text{BLM}}(Y)$ are the forward and backward LM (FLM and BLM) scores for Y, μ_* are the weights for FLM, BLM, and length penalty $|Y|$ scores (these weights were tuned using the development set), and $\hat{Y}$ is the best hypothesis.

4.5. Self-supervised adaptation (SSA)

For further improvements of ASR performance, we also performed session-wise SSA for each ASR back-end, using labels obtained from the first recognition pass using the combination of the four ASR systems. For SSA, we retrained the ASR models for each session for one epoch with AdamW optimizer using a learning rate of 5×10^{-6}. After SSA, we performed the system combination again.

4.6. ASR Results

4.6.1. Effect of data selection

Table 3 shows the results of preliminary experiments to check the effectiveness of data selection described in Section 4.1. We find that the model trained by adding all VoxCeleb1+2 data achieved worse performance than the baseline model trained with no additional data on the macro WER. On the other hand, the model trained with a quarter of VoxCeleb1+2 data achieved better overall performances than the other models.

4.6.2. Effect of two-step training

Table 4 shows the effects of the two-step training in our preliminary experiments on the dev set. The second training step improves the performance for both encoder-decoder and transducer architectures for all datasets. Overall, retraining the model, including the WavLM parameters, achieves more than

[6] https://github.com/espnet/espnet/blob/master/egs2/librispeech/asr1/conf/tuning/train_asr_s4_decoder.yaml

[7] https://github.com/espnet/espnet/blob/master/egs2/librispeech/asr1/conf/tuning/train_asr_branchformer_hop_length160_e18_linear3072.yaml

[8] https://github.com/espnet/espnet/blob/master/egs2/librispeech/asr1/conf/tuning/train_lm_transformer2.yaml

Table 4: *Effect of the second step on the encoder-decoder (E-D) and transducer (Tran) architectures. Note that we here report WERs (not DA-WERs).*

		CHiME-6	DiPCo	Mixer 6	Macro
E-D	step 1	29.6	30.7	17.4	25.9
	+ step 2	24.3	29.0	16.0	23.1
Tran	step 1	29.8	30.1	16.5	25.5
	+ step 2	24.9	28.7	14.9	22.8

Table 5: *DA-WER on the dev set with oracle diarization and SE1 for far-field acoustic robustness task (sub-track1). The numbers in parenthesis are for sessions S26, S29 of DiPCo.*

ID	Model	CHiME-6	DiPCo (S26,29)	Mixer6	Macro
(0)	Baseline w/ SE1	32.2	33.1 (35.0)	20.2	28.5
(1)	ASR1	22.5	27.1 (28.2)	12.3	20.7
(2)	ASR2	23.0	26.1 (26.5)	12.4	20.5
(3)	ASR3	21.9	26.6 (27.4)	12.8	20.4
(4)	ASR4	22.1	26.3 (27.2)	12.6	20.3
(5)	(1)+(2)+(3)+(4)	21.0	25.7 (26.2)	11.9	19.5
(6)	(5)+LM rescoring	20.9	25.5 (26.2)	11.9	19.4
(7)	ASR1+SSA	21.3	25.4 (25.4)	12.0	19.6
(8)	ASR2+SSA	21.8	25.5 (25.7)	12.1	19.8
(9)	ASR4+SSA	21.1	25.5 (26.1)	12.0	19.5
(10)	(7)+(8)+(9)	20.7	25.2 (25.4)	11.7	19.2
(11)	(10)+LM rescoring	20.7	25.0 (25.4)	11.7	19.2

10 % relative Macro-WER improvement. This result suggests that WavLM (SSL) fine-tuning is crucial in achieving competitive ASR performance.

4.6.3. Overall evaluation on the sub-track with oracle diarization

Table 5 shows the ASR results using our proposed ASR systems for the development set of the sub-track using Oracle diarization and SE1. The different individual ASR back-ends perform similarly and achieve a relative WER improvement of about 28.8 % over the challenge baseline. Combining these systems further improves the macro WER by about 1 %, and LM rescoring achieves a small but consistent improvement. We use the pseudo label obtained from system ID (6) to perform session-wise SSA for ASR back-ends 1, 2, and 4. SSA improves ASR by about 1%. Finally, the combinations of these models achieve a WER of 19.2 %. We submitted system (11), which achieves a relative WER reduction over the baseline of 32.6 %.

Table 6 shows the ASR results for the evaluation set of the sub-track using Oracle diarization. Our proposed system achieves a relative WER improvement of 38.6 % over the baseline.

5. Overall results for main track

Table 7 compares the results of the baseline and our pipeline on the main track for the dev and eval sets. We submitted system (9), which achieves a relative WER reduction over the baseline of 44 % and 47 % for the dev and eval sets, respectively. Our system ranked third in terms of Macro WER on the eval set and second on the Mixer 6 eval set.

Table 6: *DA-WER on the eval set with oracle diarization and SE1 for far-field acoustic robustness task (sub-track1).*

ID	Model	CHiME-6	DiPCo	Mixer6	Macro
(0)	Baseline	35.5	36.3	28.6	33.4
(7)	ASR1+SSA	24.3	23.3	14.7	20.8
(8)	ASR2+SSA	25.5	23.6	15.0	21.3
(9)	ASR4+SSA	24.4	23.2	14.6	20.7
(10)	(7)+(8)+(9)	24.0	22.8	14.6	20.5
(11)	(10)+LM rescoring	24.0	22.8	14.6	20.5

6. Conclusion

We presented the multi-talker recognition system that we submitted to the DASR task of the CHiME-7 challenge. The system performs diarization with EEND-VC, and recognition with a combination of four powerful ASR systems. We exploit SSL models in different parts of our systems to build robust systems.

There are several future work directions to improve performance and generalization further. For diarization, we should explore ways to improve speaker embeddings to reduce speaker confusion when using short segments. This would allow relaxing the constraint of our current systems to have at most four speakers in an 80-second segment. We would also like to explore target speaker extraction [33] that was successful in other submissions [34]. For ASR, we would like to investigate more diverse ASR systems such as DNN-HMM hybrid systems or target-speaker ASR [35, 36].

7. References

[1] J. Barker, E. Vincent, N. Ma, H. Christensen, and P. Green, “The pascal chime speech separation and recognition challenge,” *Computer Speech & Language*, vol. 27, no. 3, pp. 621–633, 2013.

[2] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, “The third ‘CHiME’ speech separation and recognition challenge: Dataset, task and baselines,” in *Proc. ASRU*, 2015, pp. 504–511.

[3] J. Barker, S. Watanabe, E. Vincent, and J. Trmal, “The fifth ’CHiME’ speech separation and recognition challenge: Dataset, task and baselines,” 2018, arXiv:1803.10609.

[4] 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,” in *Proc. 6th International Workshop on Speech Processing in Everyday Environments (CHiME 2020)*. ISCA, 2020, pp. 1–7.

[5] 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,” in *Proc. 7th International Workshop on Speech Processing in Everyday Environments (CHiME 2023)*, 2023.

[6] K. Kinoshita, M. Delcroix, and N. Tawara, “Advances in integration of end-to-end neural and clustering-based diarization for real conversational speech,” in *Proc. Interspeech*, 2021, pp. 3565–3569.

[7] ——, “Integrating end-to-end neural and clustering-based diarization: Getting the best of both worlds,” in *Proc. ICASSP*, 2021, pp. 7198–7202.

[8] Z. Lu, Y. Wang, Y. Zhang, W. Han, Z. Chen, and P. Haghani, “Unsupervised data selection via discrete speech representation for ASR,” in *Proc. Interspeech*, 2022, pp. 3393–3397.

[9] C. Boeddecker, J. Heitkaemper, J. Schmalenstroeer, L. Drude, J. Heymann, and R. Haeb-Umbach, “Front-end processing for the

Table 7: *Overall results in terms of DA-WER on the dev set. The number in parenthesis are for the remaining dev set.*

				Dev				Eval			
ID	Diar	SE	ASR	CHiME-6	DiPCo (S26, S29)	Mixer6	Macro	CHiME-6	DiPCo	Mixer6	Macro
(0)	Base	Base	Base	62.4	56.6	22.5	47.2	77.4	54.7	33.7	55.3
(1)	DIA4	SE1	ASR1	37.0	33.6 (34.9)	12.1	27.6	43.3	32.7	15.3	30.5
(2)	DIA4	SE1	ASR2	37.2	33.2 (34.0)	12.2	27.6	43.7	32.7	15.1	30.5
(3)	DIA4	SE1	ASR4	36.7	32.6 (34.3)	12.0	27.0	43.2	31.9	14.8	30.0
(4)	DIA4	SE1	(1)+(2)+(3)	36.2	32.5 (33.7)	11.8	26.8	42.8	31.6	14.9	29.8
(5)	DIA4	SE1	ASR1+SSA	36.2	32.4 (33.5)	11.8	26.8	42.6	31.7	14.6	29.6
(6)	DIA4	SE1	ASR2+SSA	36.6	32.3 (33.2)	12.0	27.0	43.1	31.5	15.1	29.9
(7)	DIA4	SE1	ASR4+SSA	36.6	32.4 (34.0)	11.9	26.9	43.1	31.8	14.6	29.8
(8)	DIA4	SE1	(5)+(6)+(7)	35.7	32.0 (33.2)	11.7	26.5	42.3	31.1	14.6	29.3
(9)	DIA4	SE1	(8)+LM rescoring	35.6	32.0 (32.9)	11.7	26.4	42.2	31.1	14.6	29.3

CHiME-5 dinner party scenario," in *Proc. 5th International Workshop on Speech Processing in Everyday Environments (CHiME 2018)*, 2018, pp. 35–40.

[10] N. Ito, S. Araki, and T. Nakatani, "Complex angular central gaussian mixture model for directional statistics in mask-based microphone array signal processing," in *Proc. EUSIPCO*, 2016.

[11] D. Raj, P. Garcia, Z. Huang, S. Watanabe, D. Povey, A. Stolcke, and S. Khudanpur, "DOVER-Lap: A method for combining overlap-aware diarization outputs," in *Proc. SLT*, 2021.

[12] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification," in *Proc. Interspeech*, 2020, pp. 3830–3834.

[13] N. Tawara, M. Delcroix, A. Ando, and A. Ogawa, "NTT speaker diarization system for CHiME-7: multi-domain, multi-microphone end-to-end and vector clustering diarizations," 2023, arXiv:1510.08484.

[14] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao, J. Wu, L. Zhou, S. Ren, Y. Qian, Y. Qian, J. Wu, M. Zeng, X. Yu, and F. Wei, "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.

[15] 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.

[16] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. ICASSP*, 2015, pp. 5206–5210.

[17] D. Snyder, G. Chen, and D. Povey, "MUSAN: A music, speech, and noise corpus," 2015, arXiv:1510.08484.

[18] *Room Impulse Response and Noise Database*, https://www.openslr.org/28/.

[19] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman, "Voxceleb: Large-scale speaker verification in the wild," *Computer Speech & Language*, vol. 60, p. 101027, 2020.

[20] O. Kuchaiev, J. Li, H. Nguyen, O. Hrinchuk, R. Leary, B. Ginsburg, S. Kriman, S. Beliaev, V. Lavrukhin, J. Cook, P. Castonguay, M. Popova, J. Huang, and J. M. Cohen, "NeMo: A toolkit for building AI applications using neural modules," 2019, arXiv:1909.09577.

[21] A. Gulati, J. Qin, C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented transformer for speech recognition," in *Proc. Interspeech*, 2020, pp. 5036–5040.

[22] K. Miyazaki, M. Murata, and T. Koriyama, "Structured state space decoder for speech recognition and synthesis," in *Proc. ICASSP*, 2023.

[23] K. Matsuura, T. Ashihara, T. Moriya, T. Tanaka, A. Ogawa, M. Delcroix, and R. Masumura, "Leveraging large text corpora for end-to-end speech summarization," in *Proc. ICASSP*, 2023.

[24] K. Kim, F. Wu, Y. Peng, J. Pan, P. Sridhar, K. J. Han, and S. Watanabe, "E-Branchformer: Branchformer with enhanced merging for speech recognition," in *Proc. SLT*, 2023, pp. 84–91.

[25] T. Ashihara, T. Moriya, K. Matsuura, T. Tanaka, Y. Ijima, T. Asami, M. Delcroix, and Y. Honma, "SpeechGLUE: How well can self-supervised speech models capture linguistic knowledge?" in *Proc. Interspeech*, 2023, pp. 2888–2892.

[26] A. Graves, "Sequence Transduction with Recurrent Neural Networks," in *Proc. ICML Workshop on Representation Learning*, 2012.

[27] T. Moriya, T. Ashihara, H. Sato, K. Matsuura, T. Tanaka, and R. Masumura, "Improving scheduled sampling for neural transducer-based ASR," in *Proc. ICASSP*, 2023.

[28] A. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification : Labelling unsegmented sequence data with recurrent neural networks," in *Proc. ICML*, 2006, pp. 369–376.

[29] Z. Meng, N. Kanda, Y. Gaur, S. Parthasarathy, E. Sun, L. Lu, X. Chen, J. Li, and Y. Gong, "Internal language model training for domain-adaptive end-to-end speech recognition," in *Proc. ICASSP*, 2021, pp. 7338–7342.

[30] A. Ogawa, N. Tawara, M. Delcroix, and S. Araki, "Lattice rescoring based on large ensemble of complementary neural language models," in *Proc. ICASSP*, 2022, pp. 6517–6520.

[31] H. Bredin, R. Yin, J. M. Coria, G. Gelly, P. Korshunov, M. Lavechin, D. Fustes, H. Titeux, W. Bouaziz, and M.-P. Gill, "pyannote.audio: neural building blocks for speaker diarization," in *Proc. ICASSP*, 2020.

[32] H. Bredin and A. Laurent, "End-to-end speaker segmentation for overlap-aware resegmentation," in *Proc. Interspeech*, 2021.

[33] M. Delcroix, T. Ochiai, K. Zmolikova, K. Kinoshita, N. Tawara, T. Nakatani, and S. Araki, "Improving speaker discrimination of target speech extraction with time-domain speakerbeam," in *Proc. ICASSP*. IEEE, 2020, pp. 691–695.

[34] L. Ye, H. Lu, G. Cheng, Y. Chen, Z. Shang, and X. Li, "The IACAS-Thinkit System for CHiME-7 Challenge," in *Proc. 7th International Workshop on Speech Processing in Everyday Environments (CHiME 2023)*, 2023.

[35] T. Moriya, H. Sato, T. Ochiai, M. Delcroix, and T. Shinozaki, "Streaming Target-Speaker ASR with Neural Transducer," in *Proc. Interspeech*, 2022, pp. 2673–2677.

[36] T. Moriya, H. Sato, T. Ochiai, M. Delcroix, T. Ashihara, K. Matsuura, T. Tanaka, R. Masumura, A. Ogawa, and T. Asami, "Knowledge Distillation for Neural Transducer-based Target-Speaker ASR: Exploiting Parallel Mixture/Single-Talker Speech Data," in *Proc. Interspeech*, 2023, pp. 899–903.