The Hitachi/JHU CHiME-5 system: Advances in speech recognition for everyday home environments using multiple microphone arrays

Naoyuki Kanda[1], *Rintaro Ikeshita*[1], *Shota Horiguchi*[1], *Yusuke Fujita*[1], *Kenji Nagamatsu*[1], *Xiaofei Wang*[2], *Vimal Manohar*[2], *Nelson Enrique Yalta Soplin*[2], *Matthew Maciejewski*[2], *Szu-Jui Chen*[2], *Aswin Shanmugam Subramanian*[2], *Ruizhi Li*[2], *Zhiqi Wang*[2], *Jason Naradowsky*[2], *L. Paola Garcia-Perera*[2], *Gregory Sell*[2]

[1]Hitachi, Ltd.
[2]Johns Hopkins University

naoyuki.kanda.kn@hitachi.com

Abstract

This paper presents Hitachi and JHU's efforts on developing CHiME-5 system to recognize dinner party speeches recorded by multiple microphone arrays. We newly developed (1) the way to apply multiple data augmentation methods, (2) residual bidirectional long short-term memory, (3) 4-ch acoustic models, (4) multiple-array combination methods, (5) hypothesis deduplication method, and (6) speaker adaptation technique of neural beamformer. As the results, our best system in category B achieved 52.38% of word error rates (WERs) for development set, which corresponded to 35% of relative WER reduction from the state-of-the-art baseline. Our best system also achieved 48.20% of WER for evaluation set, which was the 2nd best result in the CHiME-5 competition.

1. Background

This paper describes our contribution for the 5th CHiME Challenge (CHiME-5). Our system was designed for both category A and B, and both single and multiple array settings. Fig. 1 shows all of our contributions on the CHiME-5. According to this graph, we explain acoustic modeling in section 2.1, frontend processing in section 2.2, language modeling in section 2.3, and decoding techniques in section 2.4.

2. Contributions

2.1. Acoustic modeling

2.1.1. Overview

Our acoustic model (AM) training procedure is depicted in Fig. 2.

1. We first trained Gaussian mixture model (GMM) by using the combination of L, R and L+R channel of worn microphone training data. Its training procedure is the same with the baseline [1].
2. We then created the phone alignment for L+R mixture of worn microphone data based on the GMM-AM, succeeding the data cleanup procedure as with the baseline program [1]. We then created the alignment for full training set for 1-ch AM by copying the alignment of worn microphone data. Here, full training set was created by applying multiple data augmentation techniques explained in Section 2.1.2.
3. Next, we trained iVector extractor by using the full training set. We then trained 1-ch AM by using the full training data and its iVector. Training was conducted based on the LF-MMI criterion.
4. Finally, we trained 4-ch AM initialized from 1-ch AM. Training was first conducted based on the LF-MMI criterion, and then continued based on the LF-sMBR criterion [2]. In this step, we used 4-ch array training data without data augmentation. In addition, we used weighted iVector extraction procedure, in which iVector was updated only on the single-speaker regions. Architecture of the 4-ch AM is described in Section 2.1.3.

2.1.2. Data augmentation for 1-ch AM training

We applied multiple types of data augmentation methods for the training data of 1-ch AMs. This data augmentation procedure finally produced about 4,500 hours of training data.

For worn microphone training data, we mixed L and R channels to create centered (=L+R) channel. Then L, R and center channels were augmented by speed perturbation [3] (x3), volume perturbation (x1), reverberation and noise perturbation (x2), and bandpass perturbation (x2). To simulate the reverberation conditions, we applied randomly generated impulse responses simulated by the image method by following the small and middle sized room settings in [4]. We also randomly added non-speech region extracted from microphone array training data in order to simulate the noisy condition. Bandpass perturbation was our original contribution in which randomly-selected frequency band was cut off under the constraint of leaving at least 1,000 Hz band within the range of less than 4,000 Hz. These procedure finally produced about 1,500 hours of data (36 times of the original data size).

We also used the first channel of all microphone array data and the data after applying BeamFormIt [5]. We applied the same data augmentation techniques except the reverberation and noise perturbation. These procedure produced about 3,000 hours of data (72 times of the original data size). Finally, we combined 1,500 and 3,000 hours of data to create full training set for 1-ch AM.

2.1.3. 4-ch AM with RBiLSTM

The model architectures of our AM are depicted in Fig. 3. For the 4-ch acoustic features, we used two types of features. One is log amplitude $\log|x_{i,f,t}|$ of the observation for each microphone i $(=1,2,3,4)$, time frame t, and frequency bin f. Another feature is the phase difference between each and the 1st

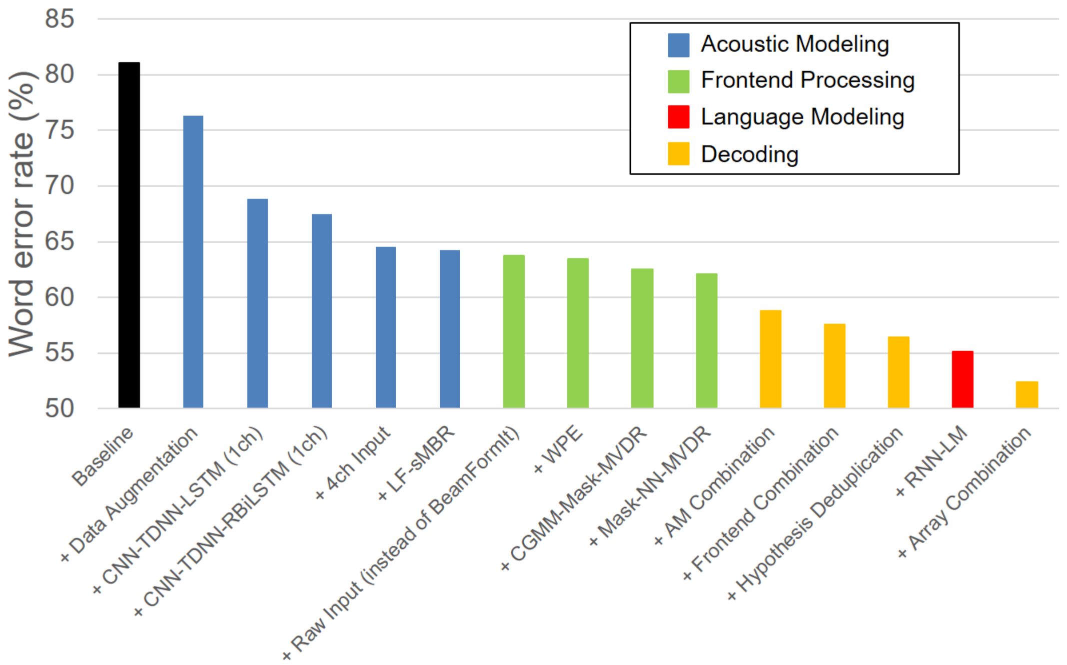

Figure 1: *Step-by-step improvements for development set*

Table 1: *Effect of data augmentation for baseline model.*

Track	Data	Epochs	rp/np	bp	Worn	Ref-Array
Single	$W + R_1$	4			44.05	79.65
Single	$W + R_1 + B_1$	4			44.49	78.72
Single	$W + R_{1..6} + B_{1..6}$	4			48.92	78.51
Single	$W + R_{1..6} + B_{1..6}$	2	√		45.82	77.26
Single	$W + R_{1..6} + B_{1..6}$	1	√	√	45.37	**76.31**

rp: reverberation perturb., np: noise perturb., bp: bandpass perturb.

In data column, W: worn mic., R_i: raw 1ch of i-th array, B_i: BeamFormIt 1ch of i-th array.

Table 2: *Comparison of acoustic model architectures.*

Track	Model	Worn	Ref-Array
Single	Baseline	45.37	76.31
Single	1-ch CNN-TDNN-LSTM (LF-MMI)	39.22	68.87
Single	1-ch CNN-TDNN-BiLSTM (LF-MMI)	40.04	68.42
Single	1-ch CNN-TDNN-RBiLSTM (LF-MMI)	39.21	67.46
Single	4-ch CNN-TDNN-RBiLSTM (LF-MMI)	n/a	64.54
Single	4-ch CNN-TDNN-RBiLSTM (LF-sMBR)	n/a	**64.25**

microphone as follows.

$$\cos(\angle(x_{i,f,t}) - \angle(x_{1,f,t})) \qquad (i = 2, 3, 4), \tag{1}$$

$$\sin(\angle(x_{i,f,t}) - \angle(x_{1,f,t})) \qquad (i = 2, 3, 4). \tag{2}$$

Our AM training procedure is as follows. We first trained the AM without 4-ch branch based on LF-MMI by using augmented training data described in section 2.1.2. We then added randomly initialized 4-ch branch to the AM and continued the LF-MMI training. In this phase, we updated only newly added parameters. Finally, entire parameters of the 4-ch AM were updated based on LF-sMBR criterion.

For the network architecture, we proposed residual bidirectional long short-term memory (RBiLSTM) in which backward(b)-LSTM is applied on top of the forward(f)-LSTM while directly appending the outputs of f-LSTM and b-LSTM (Fig. 4). WERs with different model architectures are shown in Table 2. As shown in the table, we observed performance improvements by both RBiLSTM and 4-ch input branch.

2.2. Frontend processing

For frontend processing, we newly developed minimum variance distortion-less response (MVDR) beamforming techniques with two types of speaker adaptive mask estimation methods: 1) speaker-adapted neural networks and 2) speaker-aware complex Gaussian mixture model (CGMM).

2.2.1. Speaker adaptive mask estimation neural network

In this method, we first trained a speaker-independent mask estimation network by using artificially mixed training data [6, 7]. The network was then retrained for each speaker using non-overlapped speech segments by following the start and end time stamps in the transcriptions. A loss function to adapt to the target speaker p is

$$J^{(p)}_{adapt} = \sum_{f,t} \mathsf{BCE}\left(\rho_{t,p} M^{(p)}_{f,t}, \quad M^{(p)}_{f,t}\right), \tag{3}$$

where $M^{(p)}_{f,t}$ is the network output for speaker p's mask at time t and frequency f, and BCE is the binary cross entropy function. The term $\rho_{t,p}$ is 1 if speaker p is present at time t, and otherwise 0. This loss function is designed for enhancing the difference between the target speech and interference speeches, as well as between the target speech and real background noises.

At inference stage, the mask for the target speaker p was estimated using both target and interference speakers' networks.

$$\hat{M}^{(p)}_{f,t} = \begin{cases} M^{(p)}_{f,t} & \text{if } \arg\max_q \rho_{t,q} M^{(q)}_{f,t} = p \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

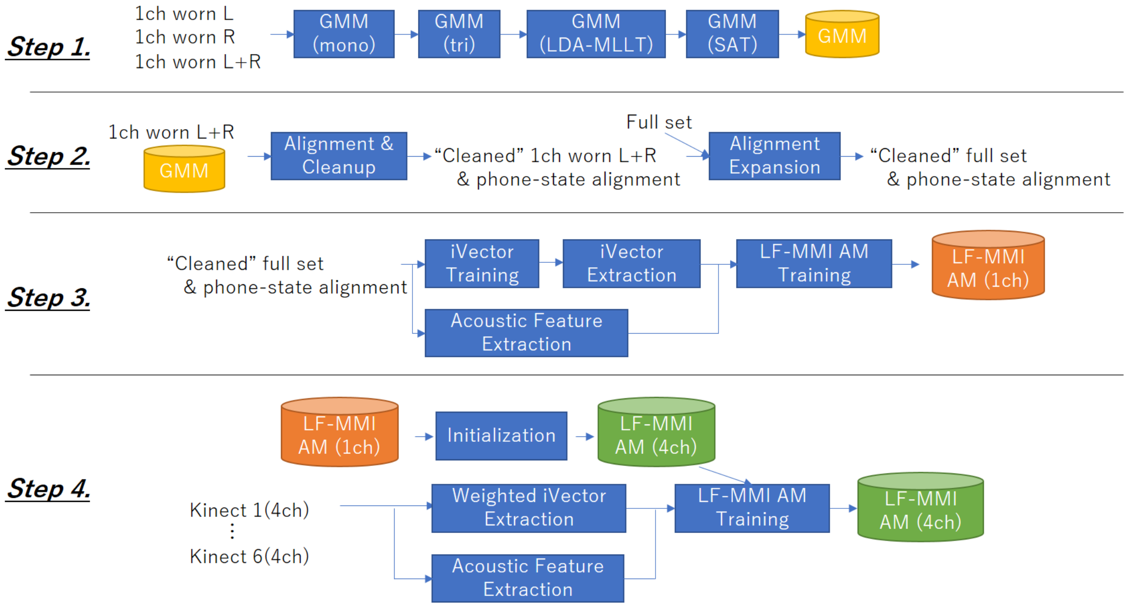

Figure 2: *Overview of acoustic model training procedure*

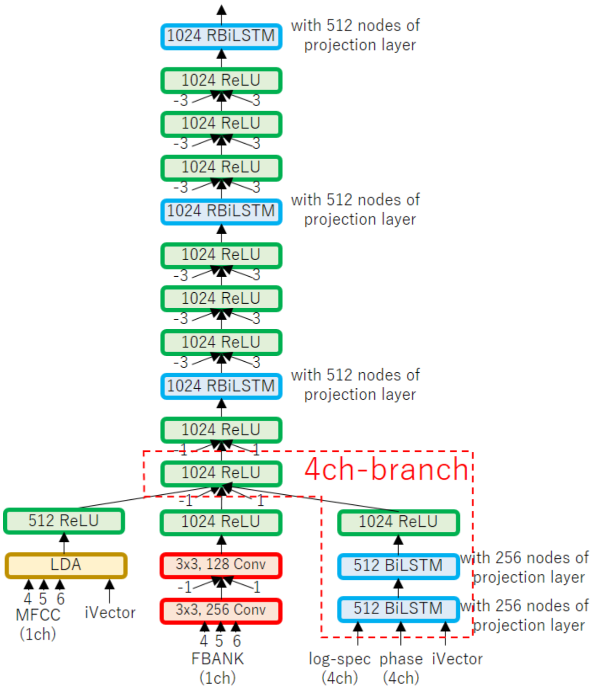

Figure 3: *Acoustic model architecture*

Motivation for this operation is to discard the time-frequency components that are not dominated by the target speaker. The estimated mask was used to estimate enhanced covariance matrices and steering vectors, which were then fed into the MVDR beamformer to extract the target speaker's speech.

2.2.2. *Speaker-aware CGMM-based MVDR*

We also developed the MVDR beamformer with CGMM-based mask estimation method [8, 9]. We used a block-wise approach to extract target speaker's speech. Namely, we segmented the multi-channel audios into small blocks, and performed offline CGMM-based MVDR beamforming for each block. Enhanced speech blocks were then concatenated to form the input for the backend decoder.

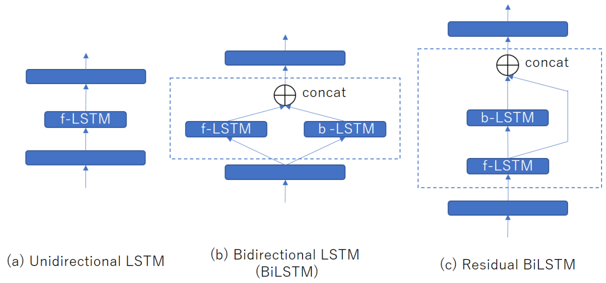

Figure 4: *LSTM, BiLSTM and Residual-BiLSTM*

We used three Gaussians to model the complex spectral coefficients; the first Gaussian for target speaker, the second one for interference speakers and the third one for background noise. To estimate the target speaker's speech accurately, we searched the nearest non-overlapped segment of the target speaker by following the start and end time stamps in the transcription, and used that segment to initialize the target speaker's Gaussian. Gaussians for interference speakers and background noise were initialized by the observation and identity matrix, respectively. After the convergence of the EM algorithm, the target speaker mask and background noise mask were both used to estimate enhanced covariance matrices and steering vectors, which were then fed into the MVDR beamformer to extract the target speaker's speech. Detailed parameters for the speaker-aware CGMM-based MVDR are given in Table 3.

Table 3: *Parameters for the block-wise CGMM-based MVDR beamformer.*

Block size = 6.4s
Frame length = 1024
Frame shift = 256
Number of Gaussians = 3
Number of iterations = 10

2.2.3. *Comparison of frontend processing*

WER with different frontends are shown in Table 4. We first found that raw input was better than BeamformIt which was used in the baseline method. We then found that weighted prediction error (WPE)-based dereverberation [10] slightly reduced the WER. Finally, we found that both the neural network based beamformer and the CGMM based beamformer significantly reduced the WER. Note that our final result was obtained by combining results with the neural network based and CGMM based MVDR beamformers.

Table 4: *Comparison of frontend processing.*

Track	Frontend for 1-ch input	Frontend for 4-ch input	Ref-Array
Single	Raw	Raw	63.79
Single	BeamFormIt	Raw	64.28 (*)
Single	WPE	WPE	63.49
Single	WPE + CGMM-MVDR	WPE	62.53
Single	WPE + NN-MVDR	WPE	**62.09**

(*) It is slightly different from the value in Table 2 due to a small difference of iVector extraction procedure in the final and preliminary systems.

2.3. Language modeling

We trained recurrent neural network language models (RNN-LMs) by using the official transcription of training data. We prepared two 2-layer LSTM-based models with forward and backward direction. In decoding, average score of the official LM, the forward RNN-LM and backward RNN-LM were used with the weighting of 0.5:0.25:0.25. Note that we submitted results without and with RNN-LM as shown in Tables 5 and 6.

2.4. Decoding

In decoding phase, we used N-best ROVER method to combine the results from different AMs. We also found that the combination of recognition results from different microphone arrays was very effective to improve the accuracy. Namely, we recognized each array with each AM independently. Then, we combined all results into the final result by using the N-best ROVER method.

For AM combination, we trained the AM in which all RBiLSTMs were replaced into conventional LSTMs or conventional BiLSTMs. We also trained AMs with 7,000 senones instead of 3,500 senones as baseline. Therefore, we finally used six types of AMs; {CNN-TDNN-RBiLSTM, CNN-TDNN-LSTM, CNN-TDNN-BiLSTM} x {3500, 7000} senones.

We also propose "hypothesis deduplication", in which if the same words were recognized for overlapped utterances, recognized words with lower confidence were excluded from the hypothesis. This produced about 1.3 point of absolute WER improvement for the final system.

3. Experimental evaluation

Our final results are shown in Table 5 (category A without RNN-LM) and in Table 6 (category B with RNN-LM). Our best system in category B achieved 55.15% (single array) and 52.38% (multiple array) of WERs for development set, which was about 35% better than the baseline [1]. In addition, our best system achieved 48.20% (single array) and 48.24% (multiple array) of WERs for evaluation set, which were the 2nd best result in the CHiME-5 competition.

One notable point is that, contrary to our expectation, multiple array combination had almost no effect on evaluation set. Especially, array combination even degraded the performance for kitchen and dining scenarios of session S21. It could be because the reference-array position was almost optimal for that session. One another notable point is that RNN-LM was effective for all environments without no exception. We were able to confirm the robustness of RNN-LM for this highly natural conversation. Overall, our system showed very competitive results in the competition, which supported the effectiveness of our proposed techniques.

Table 5: *WERs (%) for the* **category-A** *best system* **without** *RNN-LM.*

Track	Session		Kitchen	Dining	Living	Overall
Single	Dev	S02	66.37	56.79	50.89	56.40
		S09	55.89	55.94	51.57	
	Eval	S01	59.42	44.18	63.85	50.36
		S21	52.11	42.14	46.71	
Multiple	Dev	S02	61.05	54.56	50.47	54.00
		S09	51.87	52.46	52.48	
	Eval	S01	59.82	43.59	62.28	50.59
		S21	54.70	44.12	45.95	

Table 6: *WERs (%) for the* **category-B** *best system* **with** *RNN-LM.*

Track	Session		Kitchen	Dining	Living	Overall
Single	Dev	S02	65.13	55.42	49.54	55.15
		S09	55.24	54.37	50.15	
	Eval	S01	57.62	41.81	62.33	48.20
		S21	49.68	39.78	44.59	
Multiple	Dev	S02	59.31	52.96	48.95	52.38
		S09	50.64	50.69	50.46	
	Eval	S01	57.01	41.22	60.67	48.24
		S21	51.59	42.17	43.82	

4. Acknowledgments

We would like to thank Shinji Watanabe, Samik Sadhu, Jan Trmal, Hynek Hermansky, Sanjeev Khudanpur, Najim Dehak and Daniel Povey for their tremendous support for the system development during the competition period.

5. References

[1] J. Barker, S. Watanabe, E. Vincent, and J. Trmal, "The fifth CHiME Speech Separation and Recognition Challenge: Dataset, task and baselines," in *Proc. INTERSPEECH*, 2018.

[2] N. Kanda, Y. Fujita, and K. Nagamatsu, "Lattice-free state-level minimum Bayes risk training of acoustic models," in *Proc. INTERSPEECH*, 2018.

[3] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, “Audio augmentation for speech recognition.” in *Proc. INTERSPEECH*, 2015, pp. 3586–3589.

[4] 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 *Proc. ICASSP*, 2017, pp. 5220–5224.

[5] X. Anguera, C. Wooters, and J. Hernando, “Acoustic beamforming for speaker diarization of meetings,” *IEEE Trans. on ASLP*, vol. 15, no. 7, pp. 2011–2022, 2007.

[6] J. Heymann, L. Drude, and R. Haeb-Umbach, “Neural network based spectral mask estimation for acoustic beamforming,” in *Proc. ICASSP*, 2016, pp. 196–200.

[7] H. Erdogan, J. R. Hershey, S. Watanabe, M. I. Mandel, and J. Le Roux, “Improved MVDR beamforming using single-channel mask prediction networks.” in *Proc. Interspeech*, 2016, pp. 1981–1985.

[8] T. Higuchi, N. Ito, T. Yoshioka, and T. Nakatani, “Robust MVDR beamforming using time-frequency masks for online/offline ASR in noise,” in *Proc. ICASSP*, 2016, pp. 5210–5214.

[9] T. Higuchi, N. Ito, S. Araki, T. Yoshioka, M. Delcroix, and T. Nakatani, “Online mvdr beamformer based on complex gaussian mixture model with spatial prior for noise robust asr,” *IEEE/ACM Trans. on ASLP*, vol. 25, no. 4, pp. 780–793, 2017.

[10] T. Nakatani, T. Yoshioka, K. Kinoshita, M. Miyoshi, and B.-H. Juang, “Speech dereverberation based on variance-normalized delayed linear prediction,” *IEEE Trans. on ASLP*, vol. 18, no. 7, pp. 1717–1731, 2010.