# The Academia Sinica Systems of Speech Recognition and Speaker Diarization for the CHiME-6 Challenge

*Hung-Shin Lee*[1], *Yu-Huai Peng*[1], *Pin-Tuan Huang*[1], *Ying-Chun Tseng*[2], *Chia-Hua Wu*[1], *Yu Tsao*[2], *Hsin-Min Wang*[1]

[1]Institute of Information Science, Academia Sinica, Taiwan
[2]Research Center for Information Technology Innovation, Academia Sinica, Taiwan
hungshinlee@gmail.com

## Abstract

This paper describes the Academia Sinica systems for the tracks of multiple-array ASR (Track 1) and diarization+ASR (Track2) in the 6th CHiME Challenge. For Track 1, we take a different approach from the official baseline to preprocess the Kinect data and derive the state-level alignment. In addition, we develop two LF-MMI-based acoustic models, the discriminative autoencoders (DcAE) and the feature-enhanced acoustic model (FEAM), which consider feature-level regularization and enhancement, respectively. For Track 2, we propose a new CNN-based training scheme, which develops speech representations by expanding the data into a set of segments, each of which contains more than one speaker. In training, a soft label is applied to each segment based on the speaker occupation ratio, and the standard cross entropy loss is used. In the evaluation set, our best system for Track 1 (Category A) achieves 46.8% WER, slightly better than the baseline performance (51.4%). For Track 2 (Category A), our system is also superior to the baseline while using the same TDNN-based acoustic model. The DER, JER, and WER are relatively improved by 13.24%, 12.60%, and 6.57%, respectively.

## 1. System Descriptions

We describe our systems for both tracks in the 6th CHiME Challenge (CHiME-6). For details of the CHiME-6 datasets and tasks, please refer to the official website[1] and [1].

### 1.1. Track 1: ASR

#### *1.1.1. Front-end data processing*

The training process of our ASR system is divided into two parts, front-end data processing and back-end acoustic modeling. As shown in the upper part of Figure 1, we first used the worn set and the Kinect set to train the GMMs. The worn set comes from the L and R channels in the worn microphone data, and is combined with the simulated reverberant speech using RIRs and point-source noises [2]. In the baseline program[2], the Kinect set consists of 400k utterances randomly selected from all Kinect channels without any enhancement. Our Kinect set, instead, comes from 1) all the first channel utterances of the Kinect data and 2) the corresponding enhanced utterances, where all channels with time annotations were passed to the front-end of weighted prediction error (WPE), guided source separation (GSS), and BeamformIt (BF) [3, 4, 5].

[1]https://chimechallenge.github.io/chime6/overview.html

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

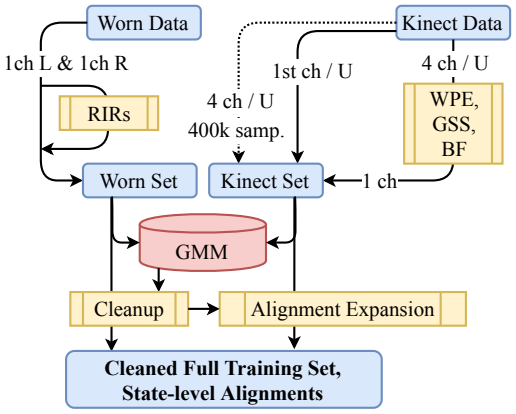


Figure 1: *Flowchart of data preparation in our system, where "U" and "ch" denote "Kinect" and "channel", respectively. The dotted line path is the method of forming the Kinect set in the baseline program.*

Following the model structure and training steps of the baseline program[2], we first created the phone alignment for the worn set based on the GMMs, and performed a data cleanup procedure. We then created the alignment and lattice for the complete training set for the NN-based acoustic models (AMs) by copying the alignment of the corresponding L channel in the worn set, i.e., the alignment expansion in [6].

#### *1.1.2. Back-end acoustic modeling*

To train the NN-based AMs, the training set was augmented by two data augmentation techniques, namely speed perturbation and volume perturbation. Bandpass perturbation [6] was not successful in our experiments.

The architectures of our two newly proposed AMs are depicted in Figure 2. The first AM is discriminative autoencoders (DcAEs) [7], which attempts to effectively separate the phonetic part (P-Code) and the residual part (R-Code) in the embedding space. In this challenge, we not only corrected several minor mistakes in our previous implementation, but also upgraded its structure from "nnet3" to "chain". In this way, the LF-MMI criterion, the cross-entropy loss, and the mean squared error can be optimized simultaneously by Kaldi's training procedures.

The second AM is the feature-enhanced acoustic model (FEAM) as shown in Figure 2 (b). In FEAM-U, "-U" means that the U-Net is used. There are also two kinds of output layers, one is the phone-state scores for the LF-MMI criterion and the cross-entropy loss, and the other is the generated acoustic features. The acoustic features generated by the feature-enhanced networks (FENs) are expected to be close to the corresponding

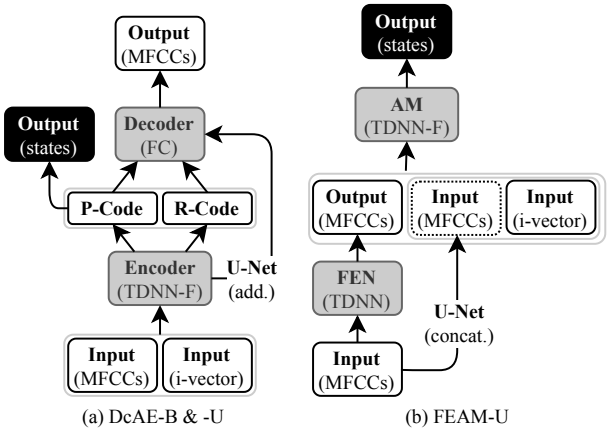


Figure 2: *Our NN-based acoustic models. In (a), FC denotes the fully-connected layers, and U-Net is used in DcAE-U but not in DcAE-B. In (b), FEN denotes the feature-enhanced networks using the TDNN layers.*

worn features during training. That is, we assume the worn set is almost clean, so that FENs can play a role in further enhancing the Kinect features.

#### 1.1.3. Objective function

The objective functions for training the DcAE-B, DcAE-U, and FEAM-U models involve the reconstruction or restoration error of acoustic features (MFCCs) and the phoneme-aware criteria, including the cross-entropy (XENT) and maximum mutual information (MMI). The error is to be minimized while the criteria are to be maximized during training.

Suppose the model $\mathcal{M}$ contains a deterministic mapping $f(\cdot)$, which is responsible for observation reconstruction (for DcAE) or restoration (for FEAM), respectively. Given a set of data $\mathcal{X}$ and the corresponding reference data $\mathcal{Y}$ ready to go through the inference-generation process $\mathcal{X} \xrightarrow{f} \mathcal{Y}$, the average reconstruction or restoration error, denoted by $\mathcal{L}_r$, is given by

$$\mathcal{L}_r(\mathcal{X}) = \frac{1}{|\mathcal{X}|} \sum_{\mathbf{x}\in\mathcal{X},\mathbf{y}\in\mathcal{Y}} \|f(\mathbf{x}) - \mathbf{y}\|_2^2, \tag{1}$$

where $\|\cdot\|_2^2$ is the 2-norm operator and $|\mathcal{X}|$ is the sample or mini-batch size. $f(\mathbf{x})$ corresponds to the "Output (MFCCs)" blocks in Figure 2. Note that, in DcAE, $\mathcal{Y}$ is equivalent to $\mathcal{X}$, which covers the MFCC vectors of the full training set shown in Figure 1. However, in FEAM-U, $\mathcal{Y}$ denotes the corresponding MFCC vectors of the worn set, which serves as the target to be restored from the MFCC vectors in $\mathcal{X}$.

There are two kinds of phoneme-aware criteria related to the "Output (states)" blocks in Figure 2. The first one, denoted by $\mathcal{F}_{xent}$, is the expected cross-entropy between the distribution represented by the reference labels and the predicted distribution. The second one, denoted by $\mathcal{F}_{mmi}$, is the MMI criterion between the distributions of the true word sequence and predicted word sequence. For the general description of LF-MMI training, also named as "chain" modeling in the Kaldi Koolkit[3], readers can refer to [8, 9].

By combining the acoustic feature reconstruction or restoration error, the phoneme-aware cross-entropy, and the MMI criterion, the objective function to be minimized becomes

$$-\mathcal{F}_{mmi} - 5 \times \mathcal{F}_{xent} + \alpha\mathcal{L}_r. \tag{2}$$

[3] https://kaldi-asr.org/doc/chain.html

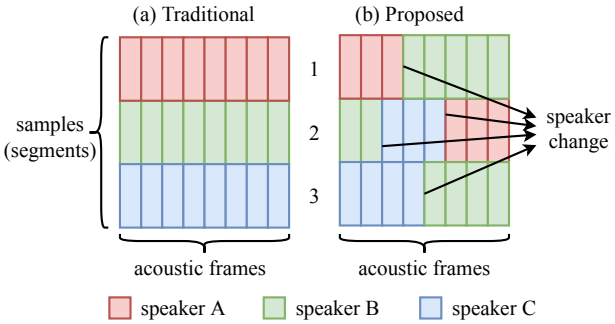


Figure 3: *Illustration of the traditional training scheme and our proposed training scheme with respect to the speaker distribution in a mini-batch. Each sample (segment) contains 8 acoustic frames and the mini-batch size is 3.*

The weight of $\mathcal{F}_{xent}$ is set to 5 following most recipes for "chain" modeling. $\alpha$ is an adjustable weight for increasing or decreasing the regularization strength of $\mathcal{L}_r$. It was set to $10^{-5}$ in this work. Actually, for the NN-based acoustic model as a whole, $\mathcal{F}_{xent}$ and $\mathcal{L}_r$ can be regarded as two regularizers.

#### 1.1.4. Recognition

In summary, we used five AMs, including DcAE-B, DcAE-U, FEAM-U, TDNN-F, and RBiLSTM [10, 6, 2]. All of them were trained on the "chain" structure using the Kaldi Toolkit, with the input combining 40-dimensional MFCCs and the 100-dimensional i-vector.

In the decoding phase, all Kinect channels were processed by WPE, GSS, and BF to form a single-channel utterance. Finally, we used the N-best ROVER method to combine the results from different AMs [11].

### 1.2. Tack 2: Diarization + ASR

For Track 2, we basically followed the baseline program[4], including the constitution of training set, dereverberation procedures, speech activity detection (SAD), and the back-end of PLDA and AHC. In the baseline program, BF is used to combine and enhance all channels of each Kinect into one channel. However, in our system, some failures for unknown reasons occurred when BF was performed on all channels of all Kinects. Therefore, we tried all possible combinations of channels and selected the set that contains the most compatible channels.

A typical speaker diarization system is composed of two components, a speaker model and a back-end processor, working for extraction and clustering of speaker representations, respectively. The main weakness of most speaker models might be the incompetence to discriminate short-duration segments, e.g., less than 2 seconds, and the ineffectiveness to extract a reliable speaker embedding when a segment contains more than one speaker. Speaker representation is crucial to speaker diarization especially when segment clustering is performed. Therefore, we propose a new training scheme to develop speaker representations by randomly augmenting the training data with segments that contain more than one speaker. That is, we attempt to produce one or more "speaker change" in each mini-batch while training. Thereinto, a soft label was applied to each segment (sample) based on the speaker occupation ratio and the standard cross entropy loss was used. Take Figure

[4] https://github.com/kaldi-asr/kaldi/tree/master/egs/chime6/s5_track2/

Table 1: *The architecture and specifications of ResNet-34, where* `res1` *to* `res5` *denote 5 ResNet-based layers,* `pooling` *denotes an average pooling layer, and $B$, $T$, $D$, and $C$ represent the batch size, temporal length, feature dimensionality, and number of training speakers, respectively.*

| Layer | Feature Size | Downsample | # Blocks |
|---|---|---|---|
| `input` | $B \times 1 \times T \times D$ | - | - |
| `conv` | $B \times 16 \times T \times D$ | False | - |
| `res1` | $B \times 16 \times T \times D$ | False | 3 |
| `res2` | $B \times 32 \times \frac{T}{2} \times \frac{D}{2}$ | True | 4 |
| `res3` | $B \times 64 \times \frac{T}{4} \times \frac{D}{4}$ | True | 6 |
| `res4` | $B \times 128 \times \frac{T}{8} \times \frac{D}{8}$ | True | 3 |
| `res5` | $B \times 256 \times \frac{T}{16} \times \frac{D}{16}$ | True | 3 |
| `pooling` | $B \times 256$ | - | - |
| `linear1` | $B \times 256$ | - | - |
| `linear2` | $B \times C$ | - | - |

3 for example, if the ground truths of samples 1 and 2 in the traditional case are [1, 0, 0] and [0, 1, 0], they can be [3/8, 5/8, 0] and [3/8, 2/8, 3/8], respectively, in our proposed case. The ratio of the number of multi-speaker segments to the number of single-speaker segments is about 13.64%.

To build a speaker model, we employed the CNN-based ResNet-34 architecture, where the feature kind, training hyperparameters, specification of layers, aggregation type, and loss function are almost the same as the baseline system described in [12]. As shown in Table 1, the only difference is that we added one additional res-layer with 256 channels and 3 blocks to the original model in order to extract the 256-dimensional speaker embeddings. The VoxCeleb-2 Corpus was used for training the speaker model. The best model checkpoint was determined by JERs of the CHiME-6 development set.

The initial speaker label for each segment was given through clustering the speaker embeddings. Resegmentation was subsequently performed with variational Bayes (VB) diarization [13], where a 2048-component UBM-GMM with diagonal covariance matrices and 400 eigenvoice bases were trained in advance with 30-dimensional MFCCs. Moreover, the initial speaker label was used for initialization in the VB diarization model. The tunable parameters, such as the minimum duration, loop probability, downsampling factor, and maximum number of iteration, were determined by the development set, and were set to 1, 0.998, 1, and 1, respectively.

## 2. Experiment Results

First, we evaluate the ASR experiments (Track 1). The results for the development and evaluation sets are presented in Table 2 (cf. the columns marked as "Track 1"). From the table, we have the following observations. First, our TDNN-F system outperforms the baseline system. Both systems are based on the same TDNN-F model architecture and have the same training strategy, but the front-end data processing is different. Compared with the front-end data processing used in the baseline system, our front-end data processing, using WPE, GSS, and BF on the Kinect training data, produces relative reductions of 2.34% and 3.89% in WER for the development and evaluation sets, respectively. Second, our DcAE-B and DcAE-U models obtain comparable results to the TDNN-F model. In addition, DcAE-U seems to be superior to DcAE-B, but the difference is small. Third, our FEAM-U model that adopts joint training for feature enhancement and acoustic modeling does not perform as expected. The reason is worth further studying in the future. Fourth, the N-best ROVER method is quite successful. Overall, compared with the baseline system, our system obtain relative reductions of 7.87% and 8.84% in WER for the development and evaluation sets, respectively. The detailed WERs for Track 1 with respect to different acoustic models and test environments (Kinect recordings in different locations) are shown in Table 4.

Table 2: *Average WERs (%) for Track 1 and Track 2 (Category A only).*

| | **Track 1** | | **Track 2** | |
|---|---|---|---|---|
| **Model** | **Dev** | **Eval** | **Dev** | **Eval** |
| Baseline | 51.32 | 51.36 | 84.25 | 77.94 |
| TDNN-F | 50.23 | 49.53 | 75.89 | 73.68 |
| RBiLSTM | 52.23 | 50.38 | 76.90 | 73.39 |
| DcAE-B | 50.30 | 49.80 | 75.90 | 73.66 |
| DcAE-U | 49.95 | 49.96 | 75.78 | 73.54 |
| FEAM-U | 53.81 | 53.05 | 78.70 | 76.20 |
| ROVER | **47.28** | **46.82** | **74.36** | **71.56** |

Table 3: *Results for Track 2. The acoustic models are the same.*

| | **Dev** | | | **Eval** | | |
|---|---|---|---|---|---|---|
| **Model** | **DER** | **JER** | **WER** | **DER** | **JER** | **WER** |
| Baseline | 63.42 | 70.83 | 84.25 | 68.20 | 72.54 | 77.94 |
| Proposed | **56.77** | **60.62** | **75.57** | **59.17** | **63.40** | **72.82** |

Next, we evaluate the effectiveness of our speaker diarization method. We applied the TDNN-F model of the baseline system to the baseline speaker diarization results and our speaker diarization results. That is, the same TDNN-F model was used in the experiment. As shown in Table 3, our speaker diarization system is superior to the baseline speaker diarization system. For the evaluation set, the DER and JER are relatively reduced by 13.24% and 12.60%, respectively. In addition, our speaker diarization results also yield better ASR performance. The WER is relatively reduced by 6.57% for the evaluation set.

Finally, we jointly evaluate our speaker diarization and ASR systems (Track 2). The results are shown in the right columns of Table 2. It is clear that based on better speaker diarization results, all our single ASR model systems (including TDNN-F, RBiLSTM, DcAE-B, DcAE-U, and FEAM-U) are superior to the baseline system. In addition, The ROVER-based fusion system can relatively reduce the WER by 8.19% for the evaluation set (from 77.94% to 71.56%). In summary, for the evaluation set of Track 2, the WER of the baseline system is 77.94%, which is reduced to 72.82% when our speaker diarization system is applied. The WER is further reduced to 71.56% when our complete (speaker diarization plus ASR) system is applied.

## 3. Conclusions and Future Work

Although the combination of our front-end data processing, speaker diarization, and acoustic modeling (with assistance of ROVER) methods outperforms the baseline system, there are still several problems that need to be solved:

1. There is a need to consider using multiple-stage GSS, where preliminary ASR is used to rectify silence align-

Table 4: *Detailed WERs (%) for Track 1 (only Category A) with respect to different acoustic models and test environments.*

| Model | Session | | Kitchen | Dining | Living | Overall |
|---|---|---|---|---|---|---|
| TDNN-F | Dev | S02 | 52.87 | 55.16 | 45.93 | 50.23 |
| | | S09 | 52.09 | 49.42 | 47.48 | |
| | Eval | S01 | 42.17 | 63.90 | 57.95 | 49.53 |
| | | S21 | 42.44 | 57.03 | 42.35 | |
| DcAE-B | Dev | S02 | 53.69 | 55.29 | 46.11 | 50.30 |
| | | S09 | 52.01 | 48.92 | 47.19 | |
| | Eval | S01 | 42.20 | 64.78 | 57.40 | 49.80 |
| | | S21 | 42.70 | 57.51 | 43.13 | |
| DcAE-U | Dev | S02 | 53.06 | 55.28 | 46.41 | 49.95 |
| | | S09 | 50.52 | 48.22 | 46.82 | |
| | Eval | S01 | 42.63 | 64.49 | 57.14 | 49.96 |
| | | S21 | 44.04 | 57.78 | 42.61 | |
| RBiLSTM | Dev | S02 | 56.38 | 57.75 | 48.23 | 52.23 |
| | | S09 | 53.67 | 50.27 | 47.92 | |
| | Eval | S01 | 43.40 | 64.24 | 57.93 | 50.38 |
| | | S21 | 44.32 | 57.33 | 43.62 | |
| FEAM-U | Dev | S02 | 57.33 | 59.64 | 50.01 | 53.81 |
| | | S09 | 54.8 | 51.74 | 49.71 | |
| | Eval | S01 | 45.59 | 68.81 | 60.29 | 53.05 |
| | | S21 | 45.40 | 61.68 | 45.44 | |
| **ROVER** | Dev | S02 | 49.97 | 52.48 | 43.29 | 47.28 |
| | | S09 | 49.04 | 44.77 | 45.36 | |
| | Eval | S01 | 40.58 | 61.21 | 53.37 | 46.82 |
| | | S21 | 40.13 | 53.99 | 39.67 | |

ments.

2. Kinect data selection is necessary because not all channels contribute to the performance of ASR.
3. The recently widely used acoustic models suitable for tasks containing dozens of hours of training speech are worth investigating, such as CNN-TDNN-F.
4. Some back-end processing techniques can be implemented, such as state-level minimum Bayes risk (sMBR) [14] training and MBR decoding with lattice combination [15].
5. Speaker diarization techniques that can better handle overlapping speech segments are needed.
6. There is still much room for improvement in the performance of the FEAM-U model. We will try to find out the best settings of the model structure and parameters.

## 4. Acknowledgements

This work was supported in part by the Ministry of Science and Technology of Taiwan under Grants: MOST 109-2634-F-008-006 and MOST 109-2634-F-001-009.

## 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] V. Manohar, S.-j. Chen, Z. Wang, Y. Fujita, S. Watanabe, and S. Khudanpur, "Acoustic modeling for overlapping speech recognition: JHU CHiME-5 challenge system," in *Proc. ICASSP*, 2019.

[3] 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 *Proc. ITG Symposium on Speech Communication*, 2018.

[4] C. Boeddecker, J. Heitkaemper, J. Schmalenstroeer, L. Drude, J. Heymann, and R. Haeb-umbach, "Front-end processing for the CHiME-5 dinner party scenario," in *Proc. Workshop on Speech Processing in Everyday Environments (CHiME)*, 2018.

[5] M. Kitza, W. Michel, C. Boeddeker, J. Heitkaemper, T. Menne, R. Schlüter, H. Ney, J. Schmalenstroeer, L. Drude, J. Heymann, and R. Haeb-Umbach, "The RWTH/UPB system combination for the CHiME 2018 workshop," in *Proc. Workshop on Speech Processing in Everyday Environments (CHiME)*, 2018.

[6] N. Kanda, R. Ikeshita, S. Horiguchi, Y. Fujita, K. Nagamatsu, X. Wang, V. Manohar, N. E. Yalta Soplin, M. Maciejewski, S.-J. Chen, A. S. Subramanian, R. Li, Z. Wang, J. Naradowsky, L. P. Garcia-Perera, and G. Sell, "The Hitachi/JHU CHiME-5 system: Advances in speech recognition for everyday home environments using multiple microphone arrays," in *Proc. Workshop on Speech Processing in Everyday Environments (CHiME)*, 2018.

[7] P.-T. Huang, H.-S. Lee, S.-S. Wang, K.-Y. Chen, Y. Tsao, and H.-M. Wang, "Exploring the encoder layers of discriminative autoencoders for LVCSR," in *Proc. Interspeech*, 2019.

[8] K. Vesely, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks," in *Proc. Interspeech*, 2013.

[9] D. Povey, V. Peddinti, D. Galvez, P. Ghahrmani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely sequence-trained neural networks for ASR based on lattice-free MMI," in *Proc. Interspeech*, 2016.

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

[11] J. G. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer Output Voting Error Reduction (ROVER)," in *Proc. ASRU*, 1997.

[12] W. Cai, J. Chen, and M. Li, "Exploring the encoding layer and loss function in end-to-end speaker and language recognition system," in *Proc. Odyssey*, 2018.

[13] M. Diez, L. Burget, and P. Matejka, "Speaker diarization based on Bayesian HMM with eigenvoice priors," in *Proc. Odyssey*, 2018.

[14] P. Voigtlaender, P. Doetsch, S. Wiesler, R. Schluter, and H. Ney, "Sequence-discriminative training of recurrent neural networks," in *Proc. Interspeech*, 2013.

[15] H. Xu, D. Povey, L. Mangu, and J. Zhu, "Minimum Bayes risk decoding and system combination based on a recursion for edit distance," *Computer Speech and Language*, vol. 25, no. 4, pp. 802–828, 2011.