The Qdreamer Systems for CHiME-6 Challenge

Haoyuan Tang[1], Huanliang Wang[1], Jiajun Wang[1], Li Zhang[1], JiaBin Xue[2], Zhi Li[1]

[1]Qdreamer Research, Suzhou, JiangSu, P.R. China
[2]School of Computer Science and Technology, Harbin Institute of Technology, Harbin, P.R. China

haoyuan.tang@qdreamer.com, huanliang.wang@qdreamer.com, jiajun.wang@qdreamer.com

Abstract

This paper presents our discription to the Chime-6 ASR system. We experimented different ways to improve the performance of our ASR system[1][2], including 1) training data augmentation via different version of enhanced training data. 2) state-level minimum bayes risk (sMBR) training. 3) acoustic model fusion. 4) system combination of different version of ehanced testing data using minimum bayes risk (MBR) decoding. 5) the forward and backward long short-term memory (LSTM) based language modeling. Experiments shows our best system in category A achieved 37.6 and 39.0 of word error rates (WER) for development and evaluation set for track1 in category A.

Index Terms: speech recognition, human-computer interaction, computational paralinguistics

1. Background

This paper presents our experiment for CHiME-6 challenge. We describe our effort to improve system performance for track1 in category A and B. Our system compose the following parts: 1). A front end including Source Activity Detector (SAD), weighted prediction error dereverberation (WPE)[3], guided source separation (GSS)[4] and Minimum Variance Distortionless Response (MVDR)[5]. 2). Acoustic modeling trained by lattice-free maximum mutual information (MMI) criterion[6][7] and sMBR[8]. 3). LSTM based language modeling trained on original and reversed text for rescoring[9].

2. Contributions

2.1. Front-end

For frontend processing, we using the baseline frontend system including SAD, SWPE, GSS and MVDR as shown in Figure 1. Multiple array data is first sent into channel selection block with different channel selection methods. After that, the selected channels of different arrays are merged together to form a mulit-channel signal. To imporve the accuracy of SAD, besides the time annotations given by the organizers, we also take advantage of non silence alignments generated by acoustic model. $\alpha_{t,k}$ in Figure 1 stands for the time annonatation. D stands for the original number of channels of array data. d stands for the number of channels after channel selection block. We also found that replacing the window of inverse fourier transform (IFFT) with ones that is orthogonal to window applied to fourier transform (FFT) leads performace improvement. Table 1 shows the performance with offical acoustic model. However, due to time limitation, this modification was not included into latter experiment.

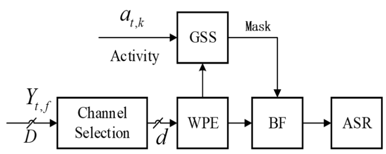

Figure 1: Front-end System.

Table 1: Experiment results of front-end with offical acoustic model

Data	dev
baseline	51.73
+ ifft window replacement	50.93

2.2. Training data augmentation

We applied multiple types of data augmentation to enlarge the data coverage. For worn microphone training data, we choose 2 type of channel selection to do front-end processing (x2). Then the signal is augmented with speed perturbation (x3), volume perturbation (x1), reverberation and noise perturbation (x2)[10].

For multiple array data, we choose 5 types of channel selection to do front-end processing (x5). Then speed perturbation (x2) and volume perterbation (x1) is applied. The channel selection of worn microphone and multiple array training data is listed in Table 2. 'L' represents the left channel of each worn microphone data is selected, while the 'R' stands for right channel. 'ch1+ch4' means the first and last channel of each multiple array data is selecte, while 'ch2+ch4' stands for the second and last channel, 'ch3+ch4' stands for the third and the last channel. 'all' stands for all 4 channels of each multiple array data is selected. 'ref-array' stands for only all channels of the reference are selected. For training data the reference array is manually set to be array ID of 'U02'. Finally, the training data is composed by the following parts.

- D1) Original worn microphone data.
- D2) Multiple array Data with 5 types of channel selection after front-end processing along with its augmented data.
- D3) Worn microphone data with 2 types of channel selection along with its augmented data.

These procedure finally result in 940 hours of training data. We have investigated the impact of training data based on official TDNN-F structures. Table 4 shows the effect of data augmenation.

Table 2: channel selection of training data

worn	mutiple array
	ch1+ch4
L	ch2+ch4
	ch3+ch4
L+R	all
	ch1
	ref-array

Table 3: channel selection of test data

mutiple array
ch1+ch4
ch2+ch4
ch3+ch4
all

Table 4: Comparison of acoustic models trained with different data

Data	dev
baseline	51.73
D1+D2	46.48
D1+D2+D3	45.87

2.3. Acoustic models

In the back-end, we use 3 different kinds of acoustic models, all trained on LF-MMI criterion using kaldi toolkit. The ASR system include TDNN-F (30 layers) network[11], CNN-TDNN (11-layer CNN + 20-layer TDNN) trained and CNN-TDNN-LSTM[12]. The model architecture of CNN-TDNN-LSTM model is shown in figure 1. TDNN-F network is trained with official MFCC features and 100-dimension online ivector. CNN-TDNN is trained with 80-dimensional logmel-filterbank (LMFB) features and online ivector. CLDNN is trained with MFCC, LMFB and online ivector feautures. The 3 models are first trained with full dataset (D1+D2+D3) with LF-MMI criterion, and further fine tuned with sMBR criterion on small training dataset (D1+D2). The 3 acoustic models show strong complementarity when fused together. The comparision of the performance of the 3 models is shown in Table 5.

2.4. decoding

As for development and evaluation data, we choose one type of channel selection (namely 'ch1-ch4') to do front-end processing. The enhanced signal is sent to the 3 acoustic models to calculate posterior respectively. We first ensemble the 3 acoustic models via state posterior

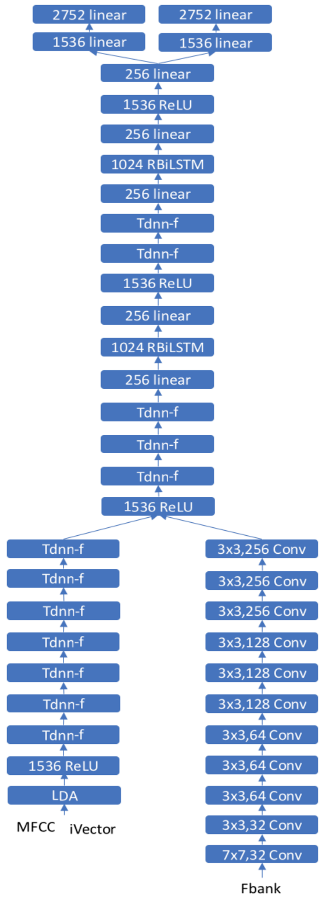

Figure 2: architecture of CNN-TDNN-LSTM network.

averaging[13], and send the averaged posterior to the decoder. Then we do a second pass decoding by introduce non silence alignments generated by the ensemble model to refine activity in the front-end. This time, we choose 4 types of channel selection ('ch1+ch4', 'ch2+ch4', 'ch3+ch4', 'all') (as shown in Table 3)to do front-end processing to generate 4 enhanced signal. Again, we decode the 4 signal with model ensemble and get 4 decoding results for each enhanced signal. Finally we use MBR decoding method[14] to combine the results of the 4 enhanced signal. The performance of model ensemble and MBR decoding is shown in Table 6.

2.5. Language models

We trained recurrent network for language models by using official original and reversed transcription of training data. We prepare two 2-layer LSTM models with forward and backward direction. In the rescoring stage, the language score of official LM, the forward LSTM and backward LSTM is weighted with 0.4:0.3:0.3. The performance of our language model in rescoring is shown in Table 7.

3. Experiment evaluation

Our final results is shown in Table 8 (category A without RNN-LM) and in Table 9 (category B with RNN-LM).

Table 5: Comparison of network structures

Model structure	dev
baseline	45.87
TDNN-F(30)	45.19
+sMBR	44.69
CNN-TDNN	44.48
+sMBR	44.05
CNN-TDNN-LSTM	44.97
+sMBR	44.06

Table 6: Performance of model ensemble and MBR decoding

method	dev
posterior averaging	41.52
+alignment	39.71
+MBR decoding	37.59

Table 7: Performance of model ensemble and MBR decoding

method	dev
baseline lm	39.71
+RNN-LM	37.99
+MBR decoding	35.95

Table 8: WER for category-A best system without RNN-LM

Track	Session	WER
track1	Dev	37.6
	Eval	39.0

Table 9: WER for category-B best system with RNN-LM

Track	Session	WER
track1	Dev	36.0
	Eval	37.5

Our best system in category A achived 37.6 of WER, and 36.0 of WER in category B for development set. In addition, our best system achived 39.0 of WER in category A, and 37.5 WER in category B for evaluation set.

4. References

[1] J. B. E. V. A. A. X. C. S. K. V. M. D. P. D. R. D. S. A. S. S. J. T. B. B. Y. C. B. Z. N. Y. F. S. H. N. K. T. Y. Shinji Watanabe, Michael Mandel, "Chime-6 challenge: Tackling multispeaker speech recognition for unsegmented recordings," in Proc. 6th International Workshop on Speech Processing in Everyday Environments (CHiME 2020), 2020.

[2] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," Proceedings of the IEEE, vol. 77, no. 2, pp. 257–286, 1989.

[3] T. Yoshioka and T. Nakatani, "Generalization of multichannel linear prediction methods for blind mimo impulse response shortening," IEEE Transactions on Audio Speech & Language Processing, vol. 20, no. 10, pp. 2707–2720, 2012.

[4] C. Boeddeker, J. Heitkaemper, J. Schmalenstroeer, L. Drude, J. Heymann, and R. Haebumbach, "Front-end processing for the chime-5 dinner party scenario," 2018.

[5] M. Souden, J. Benesty, and S. Affes, "On optimal frequency-domain multichannel linear filtering for noise reduction," IEEE Transactions on Audio Speech & Language Processing, vol. 18, no. 2, pp. 260–276, 2010.

[6] D. Povey, V. Peddinti, D. Galvez, P. Ghahremani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely sequence-trained neural networks for asr based on lattice-free mmi." pp. 2751–2755, 2016.

[7] Y. Wang, V. Peddinti, H. Xu, X. Zhang, D. Povey, and S. Khudanpur, "Backstitch: Counteracting finite-sample bias via negative steps." pp. 1631–1635, 2017.

[8] K. Veselý, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks," pp. 2345–2349, 2013.

[9] Y. Wang, D. Snyder, H. Xu, V. Manohar, P. S. Nidadavolu, D. Povey, and S. Khudanpur, "The jhu asr system for voices from a distance challenge 2019," 2019.

[10] 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 Icassp IEEE International Conference on Acoustics, 2017.

[11] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts." pp. 3214–3218, 2015.

[12] T. N. Sainath, O. Vinyals, A. W. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," pp. 4580–4584, 2015.

[13] J. Du, T. Gao, L. Sun, F. Ma, and J. D. Chen, "The ustc-iflytek systems for chime-5 challenge," in CHiME 2018 Workshop on Speech Processing in Everyday Environments, 2018.

[14] H. Xu, D. Povey, L. Mangu, and Z. Jie, "Minimum bayes risk decoding and system combination based on a recursion for edit distance," Computer Speech & Language, vol. 25, no. 4, pp. p.802–828, 2011.