INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

The LeVoice Far-field Speech Recognition System for VOiCES from a Distance Challenge 2019

Yulong Liang, Lin Yang, Xuyang Wang, Yingjie Li, Chen Jia, Junjie Wang

Lenovo Research
Liangyl3@lenovo.com

Abstract

This paper describes our submission to the "VOiCES from a Distance Challenge 2019", which is designed to foster research in the area of speaker recognition and automatic speech recognition (ASR) with a special focus on single channel distant/far-field audio under noisy conditions. We focused on the ASR task under a fixed condition in which the training data was clean and small, but the development data and test data were noisy and unmatched. Thus we developed the following major technical points for our system, which included data augmentation, weighted-prediction-error based speech enhancement, acoustic models based on different networks, TDNN or LSTM based language model rescore, and ROVER. Experiments on the development set and the evaluation set showed that the front-end processing, data augmentation and system fusion made the main contributions for the performance increasing, and the final word error rate results based on our system scored 15.91% and 19.6% respectively.

Index Terms: VOiCES from a Distance Challenge 2019, speech enhancement, speech recognition, rescore, ROVER

1. Introduction

Since the accuracy of the close-talking and the noise-free speech recognition is approaching the best possible human speech recognition performance [1-4], more and more researchers have turned their attention to the far-field and noisy scenarios[5-8].

The "VOiCES from a Distance Challenge 2019'' [9][10] is such a competition designed to foster research in the area of speaker recognition and automatic speech recognition (ASR) with a special focus on single channel distant/far-field audio under noisy conditions. This challenge is based on the newly released Voices Obscured in Complex Environmental Settings (VOiCES) corpus, and the training data is an 80 hours subset of the Librispeech dataset. The VOiCES challenge has two tasks: speaker recognition and automatic speech recognition (ASR). Each task has fixed and open training conditions. The main difficulty of each task is that the training data is small, and there was mismatch between the training data and the evaluation data.

For far-field speech recognition, a lot of researches have been conducted. These researches can be divided into two categories. In the first category, researchers process the evaluation data in the front-end to make it more matchable with the model. In the second category, researchers train acoustic models(AM) and language models in the back-end to make model parameters match the data under the test conditions as much as possible. For the front-end processing, the main methods such as Optimal Modified Minimum Mean-Square Error Log-Spectral Amplitude and Improved Minimal Controlled Recursive Averaging (OMLSA-IMCRA)[11] and Weighted Prediction Error(WPE)[12][13] are used to realize de-reverberation and de-noising. For the back end, the mainly methods include applying different acoustic model architectures, such as Deep Neural Network(DNN), Time-delay Neural Network(TDNN)[5], factorized TDNN(TDNNF), Convolutional Neural Network(CNN), Long Short Term Memory(LSTM), model parameters optimization, Neural Network Language Model(NNLM) based rescore and multi-model fusion. The goal is to decrease the mismatch between the distant speech to be recognized with the training condition.

Because the training set given was clean speech, while the development set and the evaluation set were speech under complex conditions in which different kinds of noises and reverberation existed, we took several measures to optimize the recognition performance. Firstly, in order to solve the lacking of training data, we expanded the dataset by data augmentation strategies and adding reverberation and noises; Also we trained acoustic models with different network architectures; Thirdly a rescoring mechanism was added based on the one-pass decoding lattices; Finally, ROVER [14] was used to make full use of the complementarity among different systems.

The rest of this paper is organized as follows. Section 2 introduces each component of the system. Section 3 shows ASR results obtained using the VOiCES corpus. Section 4 is the conclusion of paper.

2. System description

A unified framework is given in Figure 1. As we can see, it is formed by several important elements, including Weighted Prediction Error, data augmentation, acoustic model, language model, rescore.

2.1. Front-end processing

Considering the fact that the task was single microphone distant speech recognition, we conducted experiments on signal processing block using one algorithms. Specifically, we used the WPE algorithm for de-reverberation.

2.1.1. WPE

The signal was generated according to the following equation.

$$y[k] = \sum_{\tau=1}^{J} H[\tau]\, s[k-\tau] + v[k] \quad (1)$$

http://dx.doi.org/10.21437/Interspeech.2019-1944

The WPE aimed to find a K_l taps reverberation filter which could be represented in frequency domain as below:

$$\hat{y}_{n,l} = \sum_{k=1+D_l}^{K_l+D_l} g_{k,l}^{*} y_{n-k,l} \quad (2)$$

Where $\hat{y}_{n,l}$ denoted the reverberation signal spectrogram, and $g_{k,l}^{*}$ denoted the reverberation filter coefficients.D_l was a time delay to remove early reverberation prediction, which was beneficial to speech recognition.

The de-reverberation signal spectrogram could be expressed by the equation below:

$$\hat{s}_{n,l} = y_{n,l} - \hat{y}_{n,l} \quad (3)$$

The loss function for the reverberation filter was also known as weighted prediction error. The equation was shown in equation (4),

$$L^{[i+1]}(\bar{g}_l) = \sum_{n=0}^{N-1} \frac{\left|y_{n,l} - \hat{y}_{n,l}^{i}\right|^2}{\left|\hat{s}_{n,l}[i+1]\right|^2} \quad (4)$$

where i denoted iteration index.

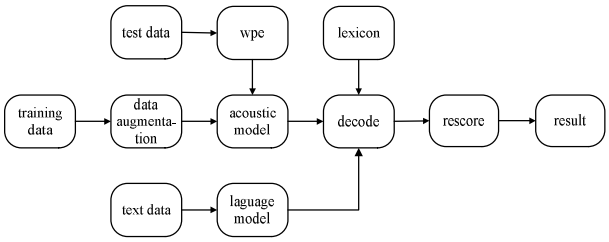


Figure 1: *the structure of our system*

The system includes data augmentation, WPE, acoustic model, language model and rescore.

2.2. Acoustic Model

2.2.1. Data augmentation

Data augmentation involved adding noises and reverberations to increase the amount of training data and to improve the robustness of the system. The augmentation methods were described in [15] and were implemented by Kaldi toolkit[16]. The noise source data and simulated impulse responses came from the dataset RIR_NOISES, which was freely available on http://www.openslr.org/28.

2.2.2. AM construction

Relying on Kaldi speech recognition toolkit we used Hidden Markov Model-Deep Neural Network (HMM-DNN) hybrid neural network acoustic models. The training procedure was as follows:

1. We first trained Gaussian Mixture Model(GMM) by using the clean 80 hours training data. Specially, two phoneme sets were used with the expectation that they were complementary for system fusion.

2. Before training a DNN AM, we conducted data cleaning and multiple data augmentation techniques explained in 2.2.1. We created the phone state alignment for cleaned and speed-perturbed training data based on the GMM AM and then copied them for full training set.

3. Next, based on the full training data, we trained the iVector extractor. Using the entire data and its iVector, we then trained an AM based on the Lattice-Free Maximum Mutual Information(LF-MMI) criterion. The input to the neural network was a 40-dimensional Mel-Frequency Cepstral Coefficients (MFCCs) appended with a 100-dimensional iVector.

4. Various network architectures were used through combinations of different types of layers, including TDNN, LSTM, Output-gate Projected Gated Recurrent Unit(BOPGRU)[17], CNN, TDNNF[18], Residual Bidirectional LSTM(RBiLSTM)[19] and other techniques, such as self-attention mechanism[20] and backstitch[21]. We evaluated different model architectures on the development set. To be specific, we investigated the following model architectures:

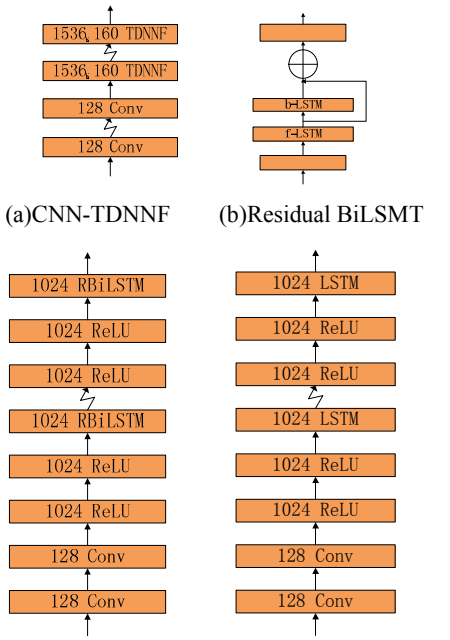
CNN-TDNNF (baseline)(Fig.2a): 7 layers CNN succeeded with 9 layers TDNNF.

CNN-attention-TDNNF: one-layer 15 heads self-attention was inserted between CNN and TDNNF as mentioned above.

CNN-TDNNF-BOPGRU: the baseline CNN-TDNNF model followed by two-layer BOPGRU.

CNN-TDNN-RBiLSTM(Fig.2c): the architecture was proposed in [17] which backward (b)-LSTM was applied on top of the forward(f)-LSTM and directly appending the outputs of f-LSTM and b-LSTM(Fig.2b).

CNN-TDNN-LSTM(Fig.2d): we did not tune this model much, it was a two-layer CNN, nine-layer TDNN and three-layer LSTMP interleaved model.



(a)CNN-TDNNF (b)Residual BiLSMT

(c)CNN-TDNN-RBiLSTM (d)CNN-TDNN-LSTM

Figure 2: *the structure of AM*

2.3. Language Model

The language model was composed of two parts. One was the traditional trigram language model, and the other was the language model based on TDNN with 3 layers. Both models were trained on the transcription of the training set. The rescoring technology would improve the performance.

2.4. System Fusion

In decoding phase, we used ROVER method to combine results from different systems. The finding was that the combination of two recognition results, derived from two weights of LM rescoring for the same system, was very effective to improve accuracy. To be detailed, the development data was decoded using each AM described in Section 2.2.2 independently. Then, we used language models with different weights (0.5 and 0.75 respectively) for rescoring. Finally, all results were combined into one using the ROVER method.

Finally, based on the performance on the development set, we selected 9 systems for submission results, including TDNNF, CNN_TDNNF, CNN_TDNNF_BOPGRU, CNN_TDNN_RBiLSTM and CNN_TDNN_LSTM.

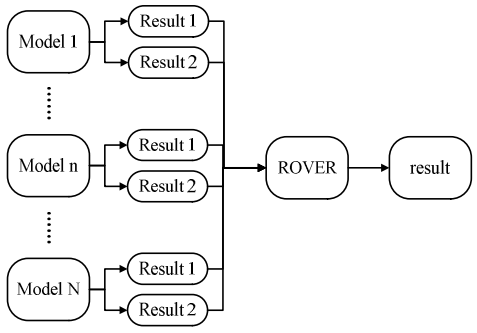


Figure 2: *the structure of ROVER*

3. Experiment setup and evaluation

3.1. Front-end experiments and data augmentation

Since the training data did not contain noises and reverberations, the front-end processing was mainly for the development set and the evaluation set. We used the WPE algorithm for the development set and the evaluation set in which the filter tap length is 40, the prediction time delay is 3 and the number of iterations on development is 3. We selected the Scaled Identity Matrix method as our spatial correlation matrix estimation method and the forgetting factor was chosen to be 0.5. The training data was augmented by 12 times, including data speed perturbation, adding point noise (included in RIR_NOISES dataset) and simulated reverberation. The results of the development set are in Table 1:

Table 1: *Result of data augmentation and front-end process*

	Utt nums	4318	1077	1074	1075	1092
TDNNF	noise type	All	None	Babb	Musi	tele
	1rvb	31.93	24.28	36.05	34.12	33.26
	3rvb	28.73	21.97	32.27	30.76	29.92
	3rvb +noise	24.97	20.64	27.19	26.09	25.95
	3rvb +noise +wpe	21.67	17.63	24.23	21.95	22.83

noise type includes none, babble, music and television. None means no noise. 1rvb means reverberation 1 time, 3rvb+noise means reverberation 3 times and adding noise, WPE means we used WPE to the development set.

According to the table 1, the best performance was achieved by adding reverberation and noise and expanding the data three times. The reason was that the data added reverberation and noise were closer to the development set data.

The performance was significantly improved by the de-reverberation and showed 10%-15% relative performance improvement respectively for different systems, which implies that there is a significant mismatch between the original development data and the training data.

3.2. Acoustic models

Different acoustic model structures and different methods were explored. The performance is shown in the following table.

Table 2: *Comparison of model architectures(WER)*

Acoustic model	Development set
TDNNF	21.67
CNN-TDNNF	21.06
CNN-TDNNF-BOPGRU	20.96
CNN-TDNN-RBiLSTM	21.01
CNN-TDNN-LSTM	22.88

Table 3: *Comparison of different methods*

Acoustic model	Development set
CNN-TDNNF (39phone)	21.06
CNN-TDNNF (39phone+attention)	21.03
CNN-TDNNF (84phone+attention)	20.58
CNN-TDNNF (84phone+attention+mix-up)	19.82

39phone and 84phone means the number of phones we used to training the model. Attention and mix-up[22] are the method when we trained the model.

From the table 2, we can see that the performances of different networks are similar. We used TDNNF model as our baseline and it turned out to be effective in several Kaldi examples. The CNN-TDNNF-BOPGRU model gained a 3.2% reduction in WER and got the best result on the development set after WPE processing. The CNN-TDNNF model gained a 2.8% reduction in WER. Similarly, as mentioned in [19], we observed performance improvement by using CNN-TDNN-RBiLSTM model which gained a 3.0% reduction in WER.

From the table 3, we can see that the performance of 39 phonemes was not as good as that of 84 phonemes. The network structure of CNN-TDNNF had the better performance. The mixed-up and attention mechanism could also improve system performance.

3.3. Language models

Language model testing consisted of two parts. One was to test the performance of the trigram and 4-gram models based on the transcription of the training data; the other was to test

the rescore performance of the TDNN language model and LSTM language model. The experimental results are as follows:

Table 4: *Comparison of different language models*

Language model	Development set
Baseline(trigram)	19.82
4-gram	20.01
NNLM(TDNN)+0.5weight	17.33
NNLM(TDNN)+0.75weight	17.17
NNLM(LSTM)+0.5weight	17.89
NNLM(LSTM)+0.75weight	17.77

The baseline model is the CNN_TDNNF in section 3.2. 0.5 and 0.75 mean the weight of NNLM.TDNN and LSTM is the structure of the NNLM.

From the table 4, we can see that the performance of trigram LM was slightly better than that of 4-gram LM. The reason may be that the transcription of the training data was small and it was difficult to train 4-gram LM sufficiently.

The performance of the TDNN-based LM was better than that of LSTM-based LM, which may be due to the small number of the training data.

3.4. ROVER

The output results based on all the model structures were fused, and the experimental results are as follows:

Table 5: Comparison of different ROVER methods

	Development set	Evaluation set
Baseline	17.17	21.66
System-0	16.27	20.06
System-1	15.91	19.60
System-2	15.96	19.66

The baseline is the model of CNN_TDNNF in section 3.3. System-0 fused 8 models(1+2+3+4+6+7+8+9), System-1 fused 9 model(1+2+3+4+5+6+7+8+9), System-2 fused 8 model(1+2+3+5+6+7+8+9), the model came from table 6. 1-9 is the number of model,

We could see from the table 5 that the ROVER was very helpful in improving the performance. We got three systems by fusing. System 0 was the result of fusing 8 models(1+2+3+4+6+7+8+9) we have trained. System 1 fused 9 model(1+2+3+4+5+6+7+8+9), and System 2 fused 8 model(1+2+3+5+6+7+8+9). The experimental results showed that the performance of System 1 was similar with that of System 2, and the result of System 0 is poor. We concluded that several systems with poor performance would lead to the decline of the overall system performance, and it would be better to fuse the models which had better performance.

3.5. Result summary

The table 6 contains all the results. From the table 6 we could find that the number of the phone, WPE, data augmentation, acoustic model such as CNN, TDNNF, RiBLSTM, BOPGRU, rescore and the methods such as attention, mix-up that used to training the acoustic model could improve the performance. This discovery is very helpful for our future work.

Table 6: *all result of our system*

AM	rescore	Dev set	Dev WPE
TDNNF 39phone+rvb1	no	31.93	
TDNNF 39phone+rvb3	no	28.73	
TDNNF 39phone+rvb3+noise(1)	no	24.97	21.67
	0.5	22.66	19.56
	0.75	22.65	19.48
CNN_TDNNF 39phone+rvb3+noise(2)	no	24.16	21.06
	0.5	21.4	18.67
	0.75	21.28	18.61
CNN_TDNNF 39phone+rvb3+noise (attention)(3)	no	23.89	21.03
	0.5	21.32	18.75
	0.75	21.29	18.74
CNN_TDNNF 84phone+rvb3+noise (attention)(4)	no		20.58
	0.5	---	18.16
	0.75		18.27
CNN+TDNNF 84phone+rvb3+noise (attention + mix-up)(5)	no		19.82
	0.5	---	17.33
	0.75		17.17
CNN_TDNNF_BOPGRU 39phone+rvb3+noise(a)(6)	0.5	---	19.18
	0.75		19.10
CNN_TDNNF_BOPGRU 39phone+rvb3+noise(b)(7)	0.5	---	18.67
	0.75		18.61
CNN_TDNNF_RBLSTM 39phone+rvb3+noise(a)(8)	0.5	---	19.10
	0.75		19.00
CNN_TDNN_RBLSTM 39phone+rvb3+noise(b)(9)	0.5	---	19.02
	0.75		18.91

Dev set is the development set. Dev WPE means that we used the WPE to the development set. 39phone and 84phone means the phone we used to train the model. Rvb3 means doing reverberation 3 times to the training data. Noise means adding noise to the training data, the different of (a) and (b) is the number of layer in the neural network. (1)-(9) is the number of the model of our system. Attention means training model with it. Mix-up means training model with it.

4. Conclusion

By taking part in the "VOiCES from a Distance Challenge 2019", we discovered that methods such as data augmentation, WPE, rescoring, ROVER effectively improved the system performance. For the following researches and experiments, some strategies of denoising could be adopted to further improving the ASR performance.

5. Acknowledgements

We would like to thank our team members for their tremendous support for the system development during the competition.

6. References

[1] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks." in Proc. INTERSPEECH, 2011, pp. 437-440.

[2] G. E. Dahl, D. Yu, L. Deng and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," IEEE Trans. SAP, vol. 20, no. 1, pp. 30-42, 2012.

[3] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A.Senior, V.Vanhoucke, P.Nguyen, T.N.Sainath, and B.Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," Signal Processing Magazine, IEEE, vol. 29, no. 6, pp. 82-97, 2012.

[4] G.Saon, H.Soltau, D.Nahamoo, and M.Picheny, "Speaker adaptation of neural network acoustic models using i-vectors." in Proc. ASRU, 2013, pp. 55-59.

[5] S.Renals, T.Hain, and H.Bourlard, "Recognition and understanding of meetings the ami and amida projects," in Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU'07, Kyoto, 12 2007, iDIAP-RR 07-46.

[6] P. Swietojanski, A. Ghoshal, and S. Renals, "Convolutional neural networks for distant speech recognition," Signal Processing Letters, IEEE, vol. 21, no. 9, pp. 1120-1124, September 2014.

[7] A. Menon, C. Kim, and R. M. Stern, "Robust Speech Recognition Based on Binaural Auditory Processing," in INTERSPEECH 2017, Aug 2017.

[8] C. Kim, K.Chin, M.Bacchiani, and R.M. Stern, "Robust speech recognition using temporal masking and thresholding algorithm," in INTERSPEECH-2014, Sept 2014, pp. 2734-2738.

[9] C. Richey, M. Barrios, Z. Armstrong, C. Bartels, H. Franco, M. Graciarena, A. Lawson, M. K. Nandwana, A. Stauffer, J. van Hout, P. Gamble, J. Hetherly, C. Stephenson, K. Ni, "Voices Obscured in Complex Environmental Settings (VOICES) corpus," in ISCA INTERSPEECH 2018, pp. 1566-1570, 2018.

[10] M. K. Nandwana, J. Van Hout, M. McLaren, C. Richey, A. Lawson, M Barrios, "The VOiCES from a Distance Challenge 2019 Evaluation Plan," arXiv:1902.10828 [eess.AS], March 2019.

[11] Cohen I, "Noise spectrum estimation in adverse environments: Improved minima controlled recursive averaging[J]". IEEE Transactions on speech and audio processing, 2003, 11(5): 466-475.

[12] Yoshioka T, Nakatani T, "Generalization of multi-channel linear prediction methods for blind MIMO impulse response shortening[J]," IEEE Transactions on Audio, Speech, and Language Processing, 2012, 20(10): 2707-2720.

[13] F. Weninger, S. Watanabe, Y. Tachioka, and B. Schuller, "Deep recurrent de-noising auto-encoder and blind de-reverberation for reverberated speech recognition," in ICASSP, 2014, pp. 4656-4659.

[14] J. Fiscus. "Post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover)". In Proceedings of the 1997 IEEE ASRU Workshop, pages 347-354, Santa Barbara, CA, 1997.

[15] Tom Ko, Vijayaditya Peddinti, Daniel Povey, Michael L. Seltzer and Sanjeev Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition", ICASSP 2017

[16] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O.Glembek,N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarzet al., "The kaldi speech recognition toolkit," in Proc. ASRU, 2011.

[17] Cheng G, Povey D, Huang L, et al. Output-Gate Projected Gated Recurrent Unit for Speech Recognition[J]. Proc. Interspeech 2018, 2018: 1793-1797.

[18] Povey D, Cheng G, Wang Y, et al, "Semi-orthogonal low-rank matrix factorization for deep neural networks[C]"//Proceedings of the 19th Annual Conference of the International Speech Communication Association (INTERSPEECH 2018), Hyderabad, India. 2018.

[19] Kanda N, Ikeshita R, Horiguchi S, et al, "The Hitachi/JHU CHiME-5 system: Advances in speech recognition for everyday home environments using multiple microphone arrays[C],"//The 5th International Workshop on Speech Processing in Everyday Environments (CHiME 2018), Interspeech. 2018.

[20] Povey D, Hadian H, Ghahremani P, et al. "A time-restricted self-attention layer for asr[C],"//2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018: 5874-5878.

[21] Wang Y, Peddinti V, Xu H, et al, "Backstitch: Counteracting Finite-Sample Bias via Negative Steps[C],"//Interspeech. 2017: 1631-1635.

[22] Medennikov I, Khokhlov Y, Romanenko A, et al. An Investigation of Mixup Training Strategies for Acoustic Models in ASR[C]//Interspeech 2018. ISCA, 2018.