UTTERANCE-WISE RECURRENT DROPOUT AND ITERATIVE SPEAKER ADAPTATION FOR ROBUST MONAURAL SPEECH RECOGNITION

Peidong Wang[1] *DeLiang Wang*[1,2]

[1]Department of Computer Science and Engineering, The Ohio State University, USA
[2]Center for Cognitive and Brain Sciences, The Ohio State University, USA
{wang.7642, wang.77}@osu.edu

ABSTRACT

This study addresses monaural (single-microphone) automatic speech recognition (ASR) in adverse acoustic conditions. Our study builds on a state-of-the-art monaural robust ASR method that uses a wide residual network with bidirectional long short-term memory (BLSTM). We propose a novel utterance-wise dropout method for training LSTM networks and an iterative speaker adaptation technique. When evaluated on the monaural speech recognition task of the CHiME-4 corpus, our model yields a word error rate (WER) of 8.28% using the baseline language model, outperforming the previous best monaural ASR by 16.19% relatively.

Index Terms— WRBN, utterance-wise recurrent dropout, iterative speaker adaptation, CHiME-4

1. INTRODUCTION

Automatic speech recognition technology has been successfully used in many real-world scenarios. While microphone arrays are widely employed, their effectiveness as spatial filters are much reduced in far-field recordings with strong reverberation. Monaural ASR is easier to deploy and more desirable in many situations. This paper investigates monaural ASR in adverse real-world scenarios.

Recently, one of the most popular monaural acoustic model types is the convolutional, long short-term memory, fully connected deep neural networks (CLDNNs) [1]. Applying the wide residual (convolutional) network and bidirectional long short-term memory (BLSTM) layers in a CLDNN framework, wide residual BLSTM network (WRBN) yields the best performance on the monaural speech recognition task using the baseline language model in the 4th CHiME speech separation and recognition challenge (CHiME-4) [2].

WRBN may be improved using better LSTM dropout methods and speaker adaptation techniques.

Dropout for LSTM has shown to be effective to attenuate the overfitting problem in the LSTM training process [3]. For speech recognition tasks, Moon *et al.* propose a *rnnDrop* method [4]. It samples the dropout mask once per utterance and applies the mask on the cell vector. Gal *et al.*'s method samples the dropout masks similarly but applies them on the input and hidden vectors (Gal dropout) [5]. Semeniuta *et al.* compare the two dropout mask sampling approaches, per-step (frame-wise) and per-sequence (utterance-wise) [6]. They propose to apply dropout on the cell update vector (Semeniuta dropout). Cheng *et al.* conduct extensive experiments on the dropout methods for LSTMs and conclude that applying utterance-wise sampled dropout masks ("per-frame dropout" in their paper) on the output, forget, and input gates yields the best result (Cheng dropout) [7].

Speaker adaptation aims at attenuating the distribution mismatch between the training and test data caused by speaker differences. The techniques can be classified into three categories, feature-space, model-space, and feature augmentation based [8]. One of the dominant techniques in the feature space may be the feature-space maximum likelihood linear regression (fMLLR) [9]. To apply fMLLR to deep neural network (DNN) based acoustic models, a well-trained Gaussian mixture model is used to obtain the fMLLR features, upon which the DNN based system is built. An MLLR based iterative adaptation technique is also proposed to update the Gaussian parameters using the decoding result in the previous iteration [10]. Another popular feature-space technique is linear input network (LIN) [11, 12]. It learns a speaker-specific linear transformation of the acoustic model input. For commonly used model-space techniques, a subset of the DNN parameters are adapted. These include linear hidden network (LHN) [13], learning hidden unit contributions (LHUC) [14], and the recently proposed speaker adaptation for batch normalized acoustic models [15]. For feature augmentation based methods, auxiliary features, such as i-vectors and speaker-specific bottleneck features, are used as additional information for the acoustic model [16, 17].

The rest of this paper is organized as follows. In Section 2 we explain the utterance-wise recurrent dropout and the iterative speaker adaptation. In Sections 3 and 4, we show the experiment setup and results. Finally, we provide concluding remarks in Section 5.

2. SYSTEM DESCRIPTION

A DNN-HMM based monaural speech recognition system consists of two parts, an acoustic model and a decoder. Modifications on the system can be conducted in roughly three categories, acoustic model related, the interaction between the acoustic model and the decoder, and decoder related. We improve WRBN in all three categories. For the acoustic model training process, we use a new utterance-wise recurrent dropout method. To adapt the acoustic model using the decoder, we propose an iterative speaker adaptation technique. For the parameters related to the decoder, we enlarge the beamwidths in the decoding graph.

2.1. Utterance-Wise Recurrent Dropout

A typical LSTM layer can be expressed as the three formulas below.

$$\begin{pmatrix} \mathbf{i}_t \\ \mathbf{f}_t \\ \mathbf{o}_t \\ \mathbf{g}_t \end{pmatrix} = \begin{pmatrix} \sigma(\mathbf{W}_i\mathbf{x}_t + \mathbf{U}_i\mathbf{h}_{t-1} + \mathbf{b}_i) \\ \sigma(\mathbf{W}_f\mathbf{x}_t + \mathbf{U}_f\mathbf{h}_{t-1} + \mathbf{b}_f) \\ \sigma(\mathbf{W}_o\mathbf{x}_t + \mathbf{U}_o\mathbf{h}_{t-1} + \mathbf{b}_o) \\ f(\mathbf{W}_g\mathbf{x}_t + \mathbf{U}_g\mathbf{h}_{t-1} + \mathbf{b}_g) \end{pmatrix} \tag{1}$$

$$\mathbf{c}_t = \mathbf{f}_t \otimes \mathbf{c}_{t-1} + \mathbf{i}_t \otimes \mathbf{g}_t \tag{2}$$

$$\mathbf{h}_t = \mathbf{o}_t \otimes f(\mathbf{c}_t) \tag{3}$$

where $\mathbf{i}_t$, $\mathbf{f}_t$, and $\mathbf{o}_t$ are the input, forget, and output gates at step t; $\mathbf{g}_t$ is the vector of cell updates and $\mathbf{c}_t$ is the updated cell vector; $\mathbf{c}_t$ is used to update the hidden state $\mathbf{h}_t$; σ is the sigmoid function, $\otimes$ is the element-wise multiplication, and f is typically chosen to be $tanh$.

In WRBN, formula (1) is simplified to formula (4) below.

$$\begin{pmatrix} \mathbf{i}_t \\ \mathbf{f}_t \\ \mathbf{o}_t \\ \mathbf{g}_t \end{pmatrix} = \begin{pmatrix} \sigma(\mathbf{W}_i\mathbf{x}_t + \mathbf{U}_i\mathbf{h}_{t-1}) \\ \sigma(\mathbf{W}_f\mathbf{x}_t + \mathbf{U}_f\mathbf{h}_{t-1}) \\ \sigma(\mathbf{W}_o\mathbf{x}_t + \mathbf{U}_o\mathbf{h}_{t-1}) \\ f(\mathbf{W}_g\mathbf{x}_t + \mathbf{U}_g\mathbf{h}_{t-1}) \end{pmatrix} \tag{4}$$

One major difference between WRBN and conventional DNN based acoustic models is its emphasis on utterance-wise training [2, 18]. In order to train the LSTM in an utterance-wise fashion, the dropout method should be both recurrent and with little temporal information loss. We list the dropout methods satisfying both requirements in formulas (5) - (8), corresponding to *rnnDrop* by Moon *et al.*, Gal dropout, Semeniuta dropout, and Cheng dropout, respectively. Dropout is denoted as a $d()$ function.

$$\mathbf{c}_t = d(\mathbf{f}_t \otimes \mathbf{c}_{t-1} + \mathbf{i}_t \otimes \mathbf{g}_t) \tag{5}$$

$$\begin{pmatrix} \mathbf{i}_t \\ \mathbf{f}_t \\ \mathbf{o}_t \\ \mathbf{g}_t \end{pmatrix} = \begin{pmatrix} \sigma(\mathbf{W}_i d_x(\mathbf{x}_t) + \mathbf{U}_i d_h(\mathbf{h}_{t-1})) \\ \sigma(\mathbf{W}_f d_x(\mathbf{x}_t) + \mathbf{U}_f d_h(\mathbf{h}_{t-1})) \\ \sigma(\mathbf{W}_o d_x(\mathbf{x}_t) + \mathbf{U}_o d_h(\mathbf{h}_{t-1})) \\ f(\mathbf{W}_g d_x(\mathbf{x}_t) + \mathbf{U}_g d_h(\mathbf{h}_{t-1})) \end{pmatrix} \tag{6}$$

$$\mathbf{c}_t = \mathbf{f}_t \otimes \mathbf{c}_{t-1} + \mathbf{i}_t \otimes d(\mathbf{g}_t) \tag{7}$$

$$\begin{pmatrix} \mathbf{i}_t \\ \mathbf{f}_t \\ \mathbf{o}_t \\ \mathbf{g}_t \end{pmatrix} = \begin{pmatrix} d_i(\sigma(\mathbf{W}_i\mathbf{x}_t + \mathbf{U}_i\mathbf{h}_{t-1})) \\ d_f(\sigma(\mathbf{W}_f\mathbf{x}_t + \mathbf{U}_f\mathbf{h}_{t-1})) \\ d_o(\sigma(\mathbf{W}_o\mathbf{x}_t + \mathbf{U}_o\mathbf{h}_{t-1})) \\ f(\mathbf{W}_g\mathbf{x}_t + \mathbf{U}_g\mathbf{h}_{t-1}) \end{pmatrix} \tag{8}$$

A possible problem of (5) is that the cells that are dropped out may be completely excluded from the whole training process of the utterance. (6) may suffer from the same problem since different gates share the same masks in this method. (7) and (8) apply dropout only on part of the vectors, which may make the remaining part vulnerable to overfitting. Our utterance-wise recurrent dropout, shown in formula (9), tries to avoid the problems in the above dropout methods.

$$\begin{pmatrix} \mathbf{i}_t \\ \mathbf{f}_t \\ \mathbf{o}_t \\ \mathbf{g}_t \end{pmatrix} = \begin{pmatrix} \sigma(\mathbf{W}_i d_{xit}(\mathbf{x}_t) + \mathbf{U}_i d_{hi}(\mathbf{h}_{t-1})) \\ \sigma(\mathbf{W}_f d_{xft}(\mathbf{x}_t) + \mathbf{U}_f d_{hf}(\mathbf{h}_{t-1})) \\ \sigma(\mathbf{W}_o d_{xot}(\mathbf{x}_t) + \mathbf{U}_o d_{ho}(\mathbf{h}_{t-1})) \\ f(\mathbf{W}_g d_{xgt}(\mathbf{x}_t) + \mathbf{U}_g d_{hg}(\mathbf{h}_{t-1})) \end{pmatrix} \tag{9}$$

Four independently sampled utterance-wise masks are applied to all of the four hidden vectors. For the dropout on the input vectors, we opt for the conventional frame-wise method since applying utterance-wise dropout may completely lose the information in some feature dimensions.

2.2. Iterative Speaker Adaptation

Speaker adaptation is commonly used in the winning systems of the CHiME-4 challenge. Using the decoded path as the label, the acoustic model can be adapted to specific test speakers, attenuating the mismatch between the training and test data. In our work, we apply the unsupervised LIN speaker adaptation [11]. For each speaker in the test set, we train an 80×80 matrix as the linear input layer. This layer is shared among the three input channels in WRBN, corresponding to static, delta, and delta-delta features, respectively.

Observing that the improvement brought by speaker adaptation is significant, we propose to iterate the adaptation process by using the newly generated decoding result as the label for another adaptation iteration. Note that the decoding result here is the final result after the RNN language model rescoring. This iterative adaptation method is similar to a prior work using MLLR [10], but our work is in the context of the LIN adaptation for a DNN based acoustic model.

There are two ways to conduct the iterative speaker adaptation, by simply changing the label and keeping all other settings the same, or by stacking an additional linear input layer in each iteration. Note that although mathematically, multiple linear layers have the same effect as a single layer, the second method ensures that the "acoustic model" (the stacked linear layer(s) and the original acoustic model) being adapted is the

same one that generated the adaptation label. We conduct experiments on both methods and compare them in this work.

Inevitably, more iterations may lead to a longer adaptation time. Such increase in adaptation time, however, may be tackled with by well-designed system architectures that conduct speech recognition and model adaptation asynchronously, more powerful machines, and faster Internet routings.

2.3. Large Decoding Beamwidths

Due to the differences in training platforms (theirs Chainer and ours TensorFlow) and decoding systems, our result using the original WRBN is slightly worse than the one reported in the paper [2]. To compensate this system bias, we keep the WRBN acoustic model fixed and adjust the decoding parameters in the Kaldi scripts [19]. Specifically, we make the beamwidth and lattice beamwidth ten times larger than those used in the original WRBN. We also enlarge the lower and upper boundaries of the number of active tokens.

Unlike segment-wise trained conventional DNN acoustic models, WRBN takes as input complete utterances. The decoders in WRBN based systems, in our opinion, may also need to be adjusted such that relatively long-term dependencies are kept. Note that although the decoding beamwidths are larger, an empirical observation is that the decoding speed may not be influenced greatly.

3. EXPERIMENTAL SETUP

3.1. Dataset

Our experiments are conducted on the CHiME-4 corpus. It is a read speech corpus with a target of distant-talking automatic speech recognition. There are two types of data, real recorded and artificially simulated. The real data is recorded in real noisy environments, including bus, cafe, pedestrian area, and street junction. The simulated data, on the other hand, is generated by artificially mixing clean speech with noisy backgrounds. The ultimate goal of the CHiME-4 challenge is to recognize the real recorded utterances.

The training set of the CHiME-4 corpus contains 1600 real utterances and 7318 simulated utterances for each of the six microphone channels. The real utterances are uttered by 4 speakers and the simulated utterances are from the 83 speakers of the WSJ0 SI-84 training set. For the monaural task, the development set consists of 410 real utterances and 410 simulated utterances for each of the four audio environments, bus, cafe, pedestrian area, and street junction. Similarly, the monaural test set has 330 real recordings and 330 simulated utterances for each environment. The speakers in the training, development, and test set do not overlap. The utterances in the development and test set are randomly chosen from the six utterances recorded by the corresponding six microphones. Note that channels with hardware issues or masked by the user's hands or clothes, i.e. failed channels, are not selected.

3.2. Implementation Details

Using the same decoding parameters as those in the original WRBN based system, our result is 0.3% absolutely worse than the one reported in the paper [2]. We think this may be caused by the differences in training platforms and decoding systems. So we keep the WRBN model fixed and adjust the decoding parameters. Setting the decoding beamwidth to 180.0, lattice beamwidth 120.0, the minimal number of active tokens 20000 and the maximal number of tokens 80000, we are able to get a WER of 10.43%. Since the reported WER without speaker adaptation is 10.4%, we think that the system bias may be compensated with the new decoding parameters.

We fine tune the WRBN using our utterance-wise recurrent dropout method for five epochs. In addition to the dropout on LSTM, we also apply conventional dropout in the residual blocks [2]. All dropout rates are set to 0.2. We use the Adam optimizer and set the initial learning rate to 10^{-5}.

After language model rescoring, we apply LIN based iterative speaker adaptation. For each of the real and simulated set, we train a linear layer for each speaker for ten epochs. The optimizer is Adam and the initial learning rate is 10^{-4}. The linear layers, i.e. the 80×80 weight matrices, are initialized to be identity matrices. After the first adaptation process, we get the language model rescored result and use it as the label for the next iteration. For the straightforward method, i.e. simply replacing the adaptation label with a new one, we reuse the network structure and reinitialize the linear layers to identity matrices. For the method of stacking an additional layer in each iteration, we take the combination of the stacked layer(s) in the previous iteration(s) and the original acoustic model as the new acoustic model, keep them fixed, and train a new linear layer for them. In this work, we apply iterative speaker adaptation for three iterations, including the first adaptation process. During the WER calculation, language model rescoring, and speaker adaptation, all of the language model weights are chosen based on the development set results.

4. EVALUATION RESULTS

4.1. Results and Comparisons

Our model obtains a WER of 8.28% on the real recorded data of the CHiME-4 evaluation set. The results and the comparisons with the best monaural speech recognition systems are shown in Table 1. *Baseline* and *Unconstrained* denote the baseline RNN language model and unconstrained language models, respectively.

Our model outperforms the previous best model using the baseline RNN language model by 16.19% relatively. It is even better than the best model using an unconstrained language

Table 1. WER (%) Comparisons of Our Model and The Best Monaural Speech Recognition Systems

systems	Baseline		Unconstrained	
	simu	real	simu	real
Du *et al.* [20]	13.62	11.15	11.81	**9.15**
Heymann *et al.* (WRBN)	11.68	**9.88**	11.11	9.34
Proposed	11.14	**8.28**	-	-

model by 9.51% relatively. We expect the WER of our system to be further reduced using a better language model, especially when combined with our iterative speaker adaptation technique.

4.2. Results In Different Environments

We test the generalization ability of our model by comparing it with the original WRBN in all four environments. The comparisons are shown in Table 2. Note that the WRBN results are those using the unconstrained language model [2]. *WRBN* denotes the original WRBN model and *Proposed* denotes the WRBN improved by our utterance-wise recurrent dropout and iterative speaker adaptation. *bus*, *caf*, *ped*, and *str* denote the four audio environments.

Table 2. WER (%) Comparisons In Different Environments

environments	WRBN		Proposed	
	simu	real	simu	real
bus	8.07	13.22	**8.03**	**11.87**
caf	13.17	9.45	**12.94**	**8.65**
ped	**10.22**	7.75	10.44	**6.65**
str	**12.98**	6.93	13.15	**5.96**
average	**11.11**	9.34	11.14	**8.28**

The results show that our model is more robust than the original WRBN in all real scenarios by substantial margins. For the simulated data, in addition to language model differences, we think the limitations of current simulation techniques may also be part of the reason why the results in the two columns are close [21].

4.3. Step-by-Step Results

The results on the test set after each step are shown in Table 3. Note that we add the results after one iteration of the speaker adaptation in the *speaker adaptation* row.

Table 3. Step-by-Step WERs (%)

steps	simu	real
original WRBN	13.03	10.74
+ large beamwidths	12.72	10.43
+ modified Gal dropout	**12.40**	**9.72**
+ speaker adaptation	11.52	8.81
+ iterative speaker adaptation	**11.14**	**8.28**

The system bias is compensated by enlarging the decoding beamwidths. After applying the utterance-wise recurrent dropout, the WER is reduced to 9.72%, which is already better than the final result using the baseline language model in the paper [2]. One iteration of the speaker adaptation yields a WER of 8.81%, outperforming the corresponding result 9.88% by 10.83% relatively. Using the speaker adaptation for two more iterations, we observe a further improvement and get our best result 8.28%.

4.4. Results of The Two Iterative Speaker Adaptation Methods

The results of the two iterative speaker adaptation methods are shown in Table 4. The first adaptation method, denoted as *Iter*, simply changes the label and reuses the structure of the previous iteration. The second method stacks an additional linear layer in each iteration and is thus denoted as *Stack*.

Table 4. WER (%) Comparisons of The Two Iterative Speaker Adaptation Methods

iterations	Iter		Stack	
	tri-gram	RNN	tri-gram	RNN
1	11.01	8.81	11.01	8.81
2	10.59	**8.51**	**10.32**	8.52
3	10.42	**8.28**	**10.08**	8.45

While *Iter* yields better results after RNN language model rescoring, *Stack* performs better when using the simple tri-gram language model. The better tri-gram results and smaller improvements brought by the language model rescoring process may indicate that *Stack* is better at incorporating the language-level information into the acoustic model.

5. CONCLUDING REMARKS

We propose an utterance-wise recurrent dropout method and an iterative speaker adaptation technique for robust monaural speech recognition. Each of the proposed methods yields a substantial improvement on the CHiME-4 corpus. The WER of our best model is 8.28%, outperforming the previous best system by 16.19% relatively. Future directions on robust monaural speech recognition include adding speech separation frontends, upgrading the components of the CLDNN acoustic models, designing better decoders for utterance-wise trained acoustic models, and boosting the performances of end-to-end systems on small corpora.

6. ACKNOWLEDGMENTS

We would like to thank J. Heymann at Paderborn University for sharing the WRBN code, Z.Q. Wang for helpful discussions, and K. Tan for valuable comments on an early version of this paper. This work was supported in part by an AFRL contract (FA8750-15-1-0279), an NSF grant (IIS-1409431), and the Ohio Supercomputer Center.

7. REFERENCES

[1] T.N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4580–4584.

[2] J. Heymann, L. Drude, and R. Haeb-Umbach, "Wide residual blstm network with discriminative speaker adaptation for robust speech recognition," *submitted to the CHiME*, vol. 4, 2016.

[3] G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R.R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[4] T. Moon, H. Choi, H. Lee, and I. Song, "Rnndrop: A novel dropout for rnns in asr," in *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*. IEEE, 2015, pp. 65–70.

[5] Y. Gal and Z. Ghahramani, "A theoretically grounded application of dropout in recurrent neural networks," in *Advances in neural information processing systems*, 2016, pp. 1019–1027.

[6] S. Semeniuta, A. Severyn, and E. Barth, "Recurrent dropout without memory loss," *arXiv preprint arXiv:1603.05118*, 2016.

[7] G. Cheng, V. Peddinti, D. Povey, V. Manohar, S. Khudanpur, and Y. Yan, "An exploration of dropout with lstms," in *Proceedings of Interspeech*, 2017.

[8] J. Li, L. Deng, Y. Gong, and R. Haeb-Umbach, "An overview of noise-robust automatic speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 4, pp. 745–777, 2014.

[9] M.J.F. Gales, "Maximum likelihood linear transformations for hmm-based speech recognition," *Computer speech & language*, vol. 12, no. 2, pp. 75–98, 1998.

[10] P.C. Woodland, D. Pye, and M.J.F. Gales, "Iterative unsupervised adaptation using maximum likelihood linear regression," in *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*. IEEE, 1996, vol. 2, pp. 1133–1136.

[11] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*. IEEE, 2011, pp. 24–29.

[12] A. Narayanan and D.L. Wang, "Investigation of speech separation as a front-end for noise robust speech recognition," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 22, no. 4, pp. 826–835, 2014.

[13] B. Li and K.C. Sim, "Comparison of discriminative input and output transformations for speaker adaptation in the hybrid nn/hmm systems," in *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[14] P. Swietojanski, J. Li, and S. Renals, "Learning hidden unit contributions for unsupervised acoustic model adaptation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 8, pp. 1450–1463, 2016.

[15] Z.Q. Wang and D.L. Wang, "Unsupervised speaker adaptation of batch normalized acoustic models for robust asr," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 4890–4894.

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

[17] T. Tan, Y. Qian, D. Yu, S. Kundu, L. Lu, K.C. Sim, X. Xiao, and Y. Zhang, "Speaker-aware training of lstm-rnns for acoustic modelling," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 5280–5284.

[18] Z.Q. Wang and D.L. Wang, "A joint training framework for robust automatic speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 4, pp. 796–806, 2016.

[19] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, et al., "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, 2011, number EPFL-CONF-192584.

[20] J. Du, Y.H. Tu, L. Sun, F. Ma, H.K. Wang, J. Pan, C. Liu, J.D. Chen, and C.H. Lee, "The ustc-iflytek system for chime-4 challenge," *Proc. CHiME*, pp. 36–38, 2016.

[21] K. Kinoshita, M. Delcroix, S. Gannot, E.A.P. Habets, R. Haeb-Umbach, W. Kellermann, V. Leutnant, R. Maas, T. Nakatani, B. Raj, et al., "A summary of the reverb challenge: state-of-the-art and remaining challenges in reverberant speech processing research," *EURASIP Journal on Advances in Signal Processing*, vol. 2016, no. 1, pp. 7, 2016.