Microphone Distance Adaptation Using Cluster Adaptive Training for Robust Far Field Speech Recognition

Animesh Prasad, Khe Chai Sim

National University of Singapore, School of Computing
Computing 1, Singapore 117417

animesh@comp.nus.edu.sg, simkc@comp.nus.edu.sg

Abstract

Microphone distance adaptation is an important and challenging problem for far field speech recognition using a single distant microphone. This paper investigates the use of Cluster Adaptive Training (CAT) to learn a structured Deep Neural Network (DNN) that can be quickly adapted to cope with changes in the distance between the microphone and speaker at test time. A speech corpus was created by re-recording the Wall Street Journal (WSJ0) audio using far-field microphones with 8 different distances from the source. Experimental results show that unsupervised adaptation of the CAT-DNN model achieved up to 0.9% absolute word error rate reduction compared to the canonical model trained on multi-style data.

Index Terms: deep neural networks, speaker-microphone distance, acoustic modeling, adaptation

1. Introduction

Recent progress in acoustic modeling using context dependent Deep Neural Networks (DNNs) has shown promising results on many tasks beating the conventional Gaussian Mixture Models-Hidden Markov Model (GMM-HMM) systems [1]. However, there still exists a big performance degradation if the acoustic conditions of the testing data are very different from that of the training data. This mismatch may arise from one or many of the factors like speaker, channel, background noise, etc. Improving performance of DNN for classification job at the same time minimizing or at least preventing the degradation in the performance due to the mismatch is a challenging task and have gained a lot of interest among the research community. The broad strategy to tackle training-testing mismatch is to either adapt the model better to the testing conditions (model compensation) or to adapt the testing features to fit the model better (feature compensation).

Similar to the case of porting clean DNN models to noise or to new speakers, porting close talk DNN models to far field speech causes degradation in performance [2]. The techniques for speaker or noise adaptation like regularized or selective fine-tuning [3, 4], inserting a layer of linear transform [5, 6, 7], etc. might not be very helpful as these techniques require huge amount of data [8] to estimate the large number of parameters involved.

One of the prominent technique to adapt DNN models on far-field speech is to combine speech signals from multiple distant microphones via concatenation or beam forming [9, 10]. However, these signal processing techniques deal with constantly distant speech [2, 11]. Such systems require different setup for close and far field recognition. For automatic meeting transcription task in an ad-hoc fashion, microphone array setup might not be available, making it difficult to apply these techniques. Other proposed techniques try to calibrate the ad-hoc microphone array and localize the speaker [12, 13]; still these techniques require multiple microphones to perform the calibration. Also, most of these techniques assume the position of speaker to be invariant during an utterance and have little scope in terms of frame level adaptation. Ideally, the adaptation framework should be robust and capable enough to allow users to move freely and change distance within an utterance. Some proposed techniques include augmenting per-frame distance descriptor [11] similar to speaker descriptors as done in speaker adaptive techniques [14]. However, extracting good descriptors requires a separate system trained with enough data from multiple varying distance sources.

In this paper we aim to demonstrate a distance adaptation technique which is capable of addressing these requirements. The problem we try to investigate here is adapting to varying speaker distance to the microphone. This is not a microphone array scenario where multiple audio signals for same utterances are used; instead we only have access to a single microphone for this task. This scenario tries to imitate meeting transcription task using a single microphone, where speakers can be at variable distances from the microphone. Further, we don't assume that the speaker, or the speaker distance is known during the testing and hence each utterance is treated independently. The Cluster Adaptive Training (CAT) framework for DNN applied here, uses DNNs as multiple bases of a canonical parametric space. During adaptation, an interpolation vector (for this canonical parametric space), is estimated and used to combine the multiple DNN bases into a single adapted DNN, thereby minimizing the amount of adaptation data needed. The idea is to compensate the variation in the model space using few parameters. These few parameters can be estimated at utterance level or frame level allowing the scope for intra-utterance variation. The main contributions of the paper are:

- We present a new varying distance speech corpus for studying ad-hoc microphone distance adaptation. The unique property of the corpus is high inter channel variability in terms of source-microphone distance under reasonable limit of meeting scenarios.
- We demonstrate CAT as a basic model space source-microphone distance adaption technique on the collected data.

Rest of the paper is organized as follow: In Section 2, we discuss the CAT framework; in Section 3 we explain proposed CAT for distance adaptation and the adaptation technique. In Section 4, we explain the data. The results are discussed in Section 5, following which paper is concluded in Section 6.

2. Formulation of Cluster Adaptive Training

A simple feed forward deep neural network with layers $i = 1$ to L is a set of operations on input feature. For each input instance $\mathbf{x}$, each layer perform the operation as:

$$z^i(\mathbf{x}) = f\left(W^i h^i(\mathbf{x}) + b^i\right) \tag{1}$$

where,

$$h^i(\mathbf{x}) = \begin{cases} \mathbf{x}, & \text{if } i = 1 \\ z^{i-1}(\mathbf{x}), & \text{otherwise} \end{cases}$$

$$f = \begin{cases} \dfrac{\exp(x_k)}{\sum\limits_{j=0}^{dim(\mathbf{x})} \exp(x_j)}, & \text{if } i = L \\ \sigma = \dfrac{1}{1+e^{-x_k}}, & \text{otherwise} \end{cases}$$

A relatively complex level of mixing the information is by training individual models with some invariant speech characteristic, and then mixing the learned representation of some or many hidden layers. One such technique to mix representation is CAT. CAT [15] or multi-basis training [16] for DNN is a way to combine two or more basis DNNs to adapt to unseen condition with limited amount of data using limited number of parameters. Prior work has investigated the technique for speaker adaptation. We are first to propose and show that it can be applied for robust distance adaptation. In multi-basis model proposed in [16] each or some of the hidden unit activation (Activation-CAT) of the adapted models is interpolated as:

$$\hat{z}^i(\mathbf{x}) = \sum_{k=1}^{K} \lambda_k \sigma\left(W_k^i z^{i-1}(\mathbf{x}) + b_k^i\right) \tag{2}$$

In the case of the CAT model proposed in [15] the interpolation is done in the weight space (Weight-CAT) as:

$$\hat{z}^i(\mathbf{x}) = \sigma\left\{\sum_{k=1}^{K} \lambda_k \left(W_k^i z^{i-1}(\mathbf{x}) + b_k^i\right)\right\} \tag{3a}$$

$$= \sigma\left(\hat{W}^i z^{i-1}(\mathbf{x}) + \hat{b}^i\right) \tag{3b}$$

where,

$$\hat{W}^i = \sum_{k=1}^{K} \lambda_k W_k^i$$

$$\hat{b}^i = \sum_{k=1}^{K} \lambda_k b_k^i$$

Though both have shown similar performance for speaker adaptation, Weight-CAT model is more powerful as it is computationally more efficient. Once the λ_k are estimated the resulting weights ($\hat{W}^i$) can be calculated for whole utterance and only one overall transform per CAT layer need to be calculated. However, in case of the Activation-CAT all the transforms are calculated independently and resulting activation is interpolated, which is dependent on the individual frame.

3. Cluster Adaptive Training for Distance Adaptation

A distance independent DNN (DI-DNN or canonical) model can be converted into a CAT-DNN model by inserting distance dependent (DD) layers or modules. The DD modules can be inserted per training condition for one or many layers of the DI-DNN.

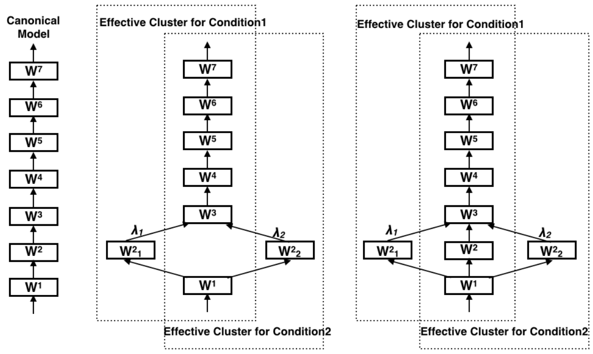

Figure 1: Turning a DI-DNN model into CAT-DNN model without the canonical module and CAT-DNN model with the canonical module

The training procedure (version of retraining) for proposed CAT is similar to the proposed method for speaker adaptive training in the prior work [17, 16]. As shown in Fig.1 the canonical or the DI-DNN module trained on pooled multiple condition data acts as the initialization for the CAT-DNN system. Depending on the number of distance conditions and availability of data DD modules are initialized with the duplicate value as of the DI module (layer). The weights for DD modules (W_c^i) are learned only by the data from distance condition c. This is done by switching off all other DD modules when a frame from distance condition c is feed forward i.e. by assigning the λ_c as either 1 or 0. After the DD modules converge the λ_c and DD modules are updated in alternate epochs. Finally, the whole network is fine-tuned with small learning rate. The final set of λ_c is discarded.

During testing each of the DD module along with the common part of the network act as a cluster. The weight of all the DD modules are initialized equally to sum to 1. The interpolation weights (λ_c) of these DD modules can be estimated per utterance or per condition in supervised manner or unsupervised manner (by using the canonical model to generate pseudo transcripts). Further the weights can also be adapted at per frame level. The effective operations as performed by a CAT-DNN during testing for a CAT applied layer becomes:

$$\hat{z}^i(\mathbf{x}) = \sum_{c} \lambda_c \sigma\left(W_c^i z^{i-1}(\mathbf{x}) + b_c^i\right) \tag{4}$$

In certain scenarios it might be helpful to include the original canonical module along with the DD modules. This can be easily incorporated in the training. Instead of duplicating the original canonical module one for each resulting DD module, new DD modules are added per condition. These new DD modules are initialized with random small weights. The canonical module is always weighted 1. The newly added DD modules are learned by the same technique as discussed earlier by training the DD module only with λ_c as either 1 or 0 till convergence. Followed by updating DD modules and λ_c intermittently.

During testing weight of all the DD modules are initialized equally to a sum to 1 and adapted in same manner as explained earlier. However, the canonical module is always weighted 1 even during testing. In this case the effective expression becomes:

$$\hat{z}^i(\mathbf{x}) = \sigma\left(W^i z^{i-1}(\mathbf{x}) + b^i\right) + \sum_c \lambda_c \sigma\left(W_c^i z^{i-1}(\mathbf{x}) + b_c^i\right) \quad (5)$$

4. Varying Distance WSJ0 (VD-WSJ0)

For concentrating on the effect of distance factor in the context of meeting scenario we create our own speech corpus. The work uses re-recorded WSJ0 [18] as speech corpus for training and testing. The standard WSJ0 data is a single channel speech data recorded with a Sienhieser microphone. For making multiple channel distance varying data with this single channel data, while preserving the original speaker variations, we performed data duplication at various distance by the setup as shown in the Fig.2.

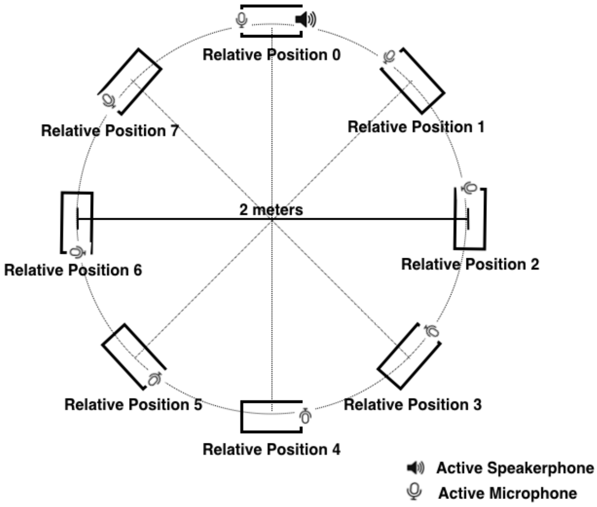

Figure 2: Data collection setup. The relative position shown is with respect to the playing iPad at the instance

We used 8 iPad Air for data collection. Each of the iPad is placed tangentially on a circle of diameter 2 meters at equal angles from the center. All the iPads are aligned in same order (left front end microphone and right top end speaker) while traversing the circle clockwise. Noise conditions in the room are nominal.

In the setup, the speaker-phone of the iPads acts as the source and the primary microphone acts as the receiver. Most of the iPads comes with in-build microphone arrays for background noise cancellation. We only used the primary microphone without any signal pre-processing by the microphone array. For generating the multichannel speech data each of the iPads played $\frac{1}{8}$th of the training data and $\frac{1}{8}$th of the testing data while all other iPads recorded the audio. The data is then sorted according to the relative position from the source clockwise (or effectively counter-clockwise). Recording data is such circular fashion with iPad taking turn to play makes audio slightly different from the case when only single iPad is playing and all other are recording. The effect of the speaker-phone and the microphone quality mismatch is arguably reduced. As speaker-phone and microphone pair effect is now equally present in all the 8 relative positions. This makes data easy to analyze from the distance perspective.

The WSJ0 SI-84 training set has 7138 utterances spoken by 83 speakers to be used as training and development set. While 330 separate utterances spoken by 12 speakers used as testing set. In our version of data we have same number of utterances per relative distance. Hence, a particular utterance has 8 different distance based versions, which refers to scenario of a single speaker repeating speech at different distances. Also, all the utterances as recorded from a microphone refers to all the speakers speaking from a fixed distance to the microphone. Pooling various non repeating data from different distances lead to interesting meeting scenarios.

Table 1: Approximate distance (in meters) of the playing speaker-phone from the micro-phone

Position	RP_0	RP_1	RP_2	RP_3	RP_4	RP_5	RP_6	RP_7
Distance	0.2	0.6	1.2	1.6	2.0	2.0	1.6	0.8

Table 1 gives the approximate distance of the various microphone from the playing speaker-phone. RP_0 (relative position 0) refers to the position of the iPad playing the utterance. RP_1 refers to the immediate clockwise position to the playing iPad and so on. The RP_7, hence refers to the last clockwise position and the immediate anti-clockwise position to the playing iPad. Though some condition have similar distance the orientation of the micro-phone is different and hence all the testing results are reported for all 8 positions separately.

5. Experiments

5.1. Setup

A single HMM-DNN system is trained for relative positions RP_0 and RP_4 using 39 dimensional MFCC with delta and delta-delta features with a splicing context window of 5 to the left and 5 to the right followed by single global CMVN transform per relative position. Training alignment is obtained by HMM-GMM cMLLR system trained on Kaldi toolkit[19] on MFCC features with same configuration. The point to notice is that we are not doing speaker adaptation; speaker labels are just used to get better training sequence to develop a stronger DNN baseline to show the applicability of our method. During DNN training and testing the speaker labels are not used. The DNN are trained using cross entropy criteria, with per layer discriminative pre-training initialization with dropouts using CNTK [20]. During testing the data are normalized per utterance.

Table 2: WER on varying distance from speaker to microphone

	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	Avg
M_0	19.1	20.5	25.9	30.0	30.5	31.0	29.2	29.0	**26.9**
M_4	24.1	23.3	25.4	26.0	26.4	25.9	26.1	25.7	**25.4**

Hereafter, D_i refers to the data recorded by microphone of RP_i and M_i refers to the model trained on training subset of D_i for $i = 0$ to 7. Table 2 shows the performance of two extreme baseline models trained with only one seen condition. The trend indicates the performance of an unadapted system with only single distance condition training data. It suggests that the WER increases as the distance between training and testing condition increases. Also, as distance of the source from microphone increases WER increases even for the matching conditions.

Table 3: WER on multi-style trained models

	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	Avg
M_{04}	20.7	21.0	25.2	27.4	27.9	27.2	27.1	26.7	**25.4**
M_{0246}	22.8	22.1	25.4	25.9	25.7	25.2	25.9	24.8	**24.7**

Table 3 shows the WER on the models trained with multi-style/pooled data. Here pooled data has same number of utterances as single condition case however the utterances are pooled from different conditions. For example M_{04} model refers to a training meeting scenario with two group of speakers, one at distance RP_0 and other at distance RP_4, however during testing again the speakers can be at any of the 8 relative positions.

5.2. CAT on Single Layer

We take M_{04} multi-style trained model as the canonical model and perform CAT at each of the layers of the canonical model one at a time. We consider two DD modules one each for D_0 and D_4. While fine-tuning the DD module the canonical module is frozen; and while re-estimating the interpolation parameters whole network is frozen. The interleaving update for DD modules and interpolation parameters is done for 2 epochs each with learning rate of 0.05, halved after each epoch. Whole network is fine tuned with small learning rate for single iteration in the end. During testing unsupervised adaptation is performed by taking the hypothesis and alignment from the decoding of the canonical model. The parameters are initialized to sum to 1 and learned per utterances with learning rate of 0.01. The results for adapted CAT Model without the canonical module are shown in the Table 4. Table 5 shows the result of CAT under same configuration but with the canonical module. Each row refers to the layers on which the CAT is applied. All these configuration requires only 2 parameters per model to be adapted during testing.

Table 4: WER for CAT on single layer without the canonical module

	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	Avg
L_1	19.9	20.7	25.2	26.5	27.2	26.9	27.0	26.5	**25.0**
L_2	20.3	20.9	25.5	26.4	27.5	26.8	27.2	26.4	**25.1**
L_3	20.5	21.2	25.6	26.8	27.4	26.8	27.4	26.6	**25.3**
L_4	20.5	21.3	25.7	27.2	27.3	26.9	27.5	26.7	**25.4**

Table 5: WER for CAT on single layer with the canonical module

	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	Avg
L_1	19.5	20.3	24.9	26.3	26.8	26.7	26.9	26.3	**24.7**
L_2	19.9	20.5	25.0	26.5	27.0	26.8	27.1	26.5	**24.9**
L_3	20.2	21.1	25.3	26.7	27.2	26.7	27.3	26.5	**25.1**
L_4	20.3	21.2	25.5	27.0	27.1	26.9	27.4	26.6	**25.2**

The results suggest that CAT for distance adaptation works better with layers closer to the feature with maximum gain being observed in the case of L_1. Further, as we move towards higher layers, the variation in input features decreases due to normalization from previous layers hence resulting into lesser gains. CAT models with the canonical module perform better than their counterpart without the canonical module. This might be because of the fact that the canonical module learns better generalization from all the data, while DD module only learns condition specific transform.

5.3. CAT on Multiple Layers

Similarly CAT can be applied on multiple layers sequentially starting form the input layer towards the regression layer. The training procedure starts with taking the canonical model adding the DD modules to lowermost layer, fine-tuning the DD module and parameters in alternate epochs followed by fine-tuning the whole network with small learning rate and repeating the same procedure for next layer till required. While introducing CAT at higher layer the lower layer DD modules as well as the interpolation parameters are used fixed (as from the previous model) and only the weights and interpolation parameters of the current layer are updated. Table 6 shows the results of applying CAT sequentially on all the layers. Each row of result refers to the range of layers on which the CAT is applied. The results show diminishing gains as we apply CAT to on multiple layers. However, the best result from model L_{1-3} suggest that with CAT even with few seen conditions (0 and 4) models can out perform multi-style model M_{0246} trained on equal data from more seen variations (0,2,4,6). Table 7 shows the breakup of the gain of the best model L_{1-3} on seen (0 and 4) and unseen conditions (1,2,3,5,6,7), suggesting more gain is observed in the testing conditions same as the training conditions.

Table 6: WER for CAT on multiple layers with the canonical module

	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	Avg
L_{1-2}	19.4	20.2	24.8	26.2	26.6	26.5	26.7	26.3	**24.6**
L_{1-3}	19.3	20.1	24.7	26.2	26.6	26.4	26.7	26.3	**24.5**
L_{1-4}	19.4	20.2	24.7	26.3	26.6	26.5	26.7	26.3	**24.6**

Table 7: Average absolute gain in WER for L_{1-3} with respect to M_{04} on seen and unseen distance conditions

Data	Seen	Unseen	Avg
Absolute Gain	1.3	0.7	**0.9**

6. Conclusions

For realization of hands-free speech recognition, distance adaptation is a crucial aspect as evident from the performance degradation in the case of mismatch. The speech corpus variation presented here might be interesting to explore various adaptation techniques specific to varying distance in meeting scenario. Further, the proposed Cluster Adaptive Training seems promising model space adaptation demonstrating 0.9% absolute WER improvement on top of the strong canonical model with dropout, using only few parameters which can be easily estimated during testing. Future work includes applying this technique in weight space instead of activation space, estimating the adaptation parameters at frame level and comparing the technique with other existing adaptation techniques. Also, as CAT is already applied to speaker adaptation, an interesting study would be to jointly adapt speaker and distance.

7. Acknowledgment

The author would like to thank Prof. Mark Gales, University of Cambridge for his insightful feedback on the paper.

8. References

[1] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "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.

[2] P. Swietojanski, A. Ghoshal, and S. Renals, "Hybrid acoustic models for distant and multichannel large vocabulary speech recognition," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding, Olomouc, Czech Republic, December 8-12, 2013*, 2013, pp. 285–290.

[3] X. Li and J. Bilmes, "Regularized adaptation of discriminative classifiers," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 1. IEEE, 2006, pp. I–I.

[4] S. Wager, S. Wang, and P. S. Liang, "Dropout training as adaptive regularization," in *Advances in Neural Information Processing Systems*, 2013, pp. 351–359.

[5] J. Trmal, J. Zelinka, and L. Müller, "Adaptation of a feedforward artificial neural network using a linear transform," in *Text, Speech and Dialogue*. Springer, 2010, pp. 423–430.

[6] K. Yao, D. Yu, F. Seide, H. Su, L. Deng, and Y. Gong, "Adaptation of context-dependent deep neural networks for automatic speech recognition," in *Spoken Language Technology Workshop (SLT), 2012 IEEE*. IEEE, 2012, pp. 366–369.

[7] D. Albesano, R. Gemello, P. Laface, F. Mana, and S. Scanzio, "Adaptation of artificial neural networks avoiding catastrophic forgetting," in *Neural Networks, 2006. IJCNN'06. International Joint Conference on*. IEEE, 2006, pp. 1554–1561.

[8] D. Yu and L. Deng, *Automatic Speech Recognition - A Deep Learning Approach*. Springer, October 2014.

[9] Y. Liu, P. Zhang, and T. Hain, "Using neural network front-ends on far field multiple microphones based speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2014, Florence, Italy, May 4-9, 2014*, 2014, pp. 5542–5546.

[10] D. Marino and T. Hain, "An analysis of automatic speech recognition with multiple microphones," in *INTERSPEECH 2011, 12th Annual Conference of the International Speech Communication Association, Florence, Italy, August 27-31, 2011*, 2011, pp. 1281–1284.

[11] Y. Miao and F. Metze, "Distance-aware DNNs for robust speech recognition," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015, pp. 761–765.

[12] I. Himawan, I. McCowan, and S. Sridharan, "Clustered blind beamforming from ad-hoc microphone arrays," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 4, pp. 661–676, 2011.

[13] M. Chen, Z. Liu, L.-W. He, P. Chou, and Z. Zhang, "Energy-based position estimation of microphones and speakers for ad hoc microphone arrays," in *Applications of Signal Processing to Audio and Acoustics, 2007 IEEE Workshop on*. IEEE, 2007, pp. 22–25.

[14] 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.

[15] T. Tan, Y. Qian, M. Yin, Y. Zhuang, and K. Yu, "Cluster adaptive training for deep neural network," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4325–4329.

[16] C. Wu and M. J. Gales, "Multi-basis adaptive neural network for rapid adaptation in speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4315–4319.

[17] T. Ochiai, S. Matsuda, X. Lu, C. Hori, and S. Katagiri, "Speaker adaptive training using deep neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 6349–6353.

[18] D. B. Paul and J. M. Baker, "The design for the wall street journal-based CSR corpus," in *Proceedings of the workshop on Speech and Natural Language*. Association for Computational Linguistics, 1992, pp. 357–362.

[19] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi speech recognition toolkit," in *Proc. of ASRU*, Dec. 2011.

[20] D. Yu, A. Eversole, M. L. Seltzer, K. Yao, B. Guenter, O. Kuchaiev, F. Seide, H. Wang, J. Droppo, Z. Huang, G. Zweig, C. J. Rossbach, and J. Currey, "An introduction to computational networks and the computational network toolkit (invited talk)," in *INTERSPEECH 2014, 15th Annual Conference of the International Speech Communication Association, Singapore, September 14-18, 2014*, 2014.