Distance-Aware DNNs for Robust Speech Recognition

Yajie Miao, Florian Metze

Language Technologies Institute, School of Computer Science, Carnegie Mellon University

{ymiao,fmetze}@cs.cmu.edu

Abstract

Distant speech recognition (DSR) remains to be an open challenge, even for the state-of-the-art deep neural network (DNN) models. Previous work has attempted to improve DNNs under constantly distant speech. However, in real applications, the speaker-microphone distance (SMD) can be quite dynamic, varying even within a single utterance. This paper investigates how to alleviate the impact of dynamic SMD on DNN models. Our solution is to incorporate the frame-level SMD information into DNN training. Generation of the SMD information relies on a universal extractor that is learned on a meeting corpus. We study the utility of different architectures in instantiating the SMD extractor. On our target acoustic modeling task, two approaches are proposed to build distance-aware DNN models using the SMD information: simple concatenation and distance adaptive training (DAT). Our experiments show that in the simplest case, incorporating the SMD descriptors improves word error rates of DNNs by 5.6% relative. Further optimizing SMD extraction and integration results in more gains.

Index Terms: Deep neural networks, speaker-microphone distance, robust acoustic modeling

1. Introduction

The pervasive deployment of speech interfaces requires automatic speech recognition (ASR) systems to handle environmental variability effectively. In recent years, the introduction of deep neural networks (DNNs) has achieved the state-of-the-art recognition accuracy on a wide range of acoustic modeling tasks [1, 2, 3]. In general, DNNs display superior generalization ability than the traditional Gaussian mixture models (GMMs) [4]. However, robustness remains to be a challenge for DNN models [5]. For example, in [6], it is revealed that the performance of DNNs degrades significantly as the SNR drops. Apart from noise, another common type of environmental variability is the distance between speakers and microphones. A performance degradation is typically observed when we port DNN models from close to distant speech [7]. A number of techniques have been developed to enhance the robustness of DNN models on far-field speech. For instance, in [7, 8], DNN models are improved by combining speech signals from multiple distant microphones via concatenation or beamforming [9]. Also for DNN models, [10] evaluates the existing environmental robustness methods on a distant-microphone meeting transcription task. A robust front-end is derived by integrating different enhancement approaches and feature types.

Although showing nice gains, these DSR techniques have the limitation that most of them deal with constantly distant speech. That is, the speaker-microphone distance (SMD) is assumed to be unchanged throughout the course of recording. However, in many real-world scenarios, the SMD can be quite dynamic. For example, a speaker is likely to walk around when talking to a far-field microphone located at a fixed position. In this case, the SMD varies a lot even within a single utterance, which poses special difficulty to acoustic modeling. This paper aims to build DNN models robust to rich SMD variability. Our solution takes advantage of DNNs' flexibility to integrate heterogeneous features under the same optimization objective. We propose to construct distance-aware DNNs by explicitly incorporating the SMD information into DNN training. Past work [11, 8, 12] has attempted to incorporate various types of context information (e.g., noise estimate [11], speaker identity [8] and visual clues [12]) for robust DNN acoustic modeling. However, to our knowledge, no previous work has dealt with the explicit incorporation of dynamic distance information.

In this work, we achieve the incorporation of the SMD information by learning a universal SMD extractor. Such an extractor is a DNN with a bottleneck layer in the architecture. It is trained on a comprehensive meeting corpus, using SMD types as the classification targets. Then, we apply this extractor to our target acoustic modeling task. Specifically, each speech frame is fed into the extractor and activations of the bottleneck layer are taken as SMD descriptors. Distance-aware DNNs are built by appending these descriptors to the acoustic features (e.g., filterbanks) as the DNN inputs. Based on this basic framework, we make further optimizations from two aspects.

- Besides DNNs, we study two alternative architectures, convolutional neural networks (CNNs) [13, 14, 15] and recurrent neural networks (RNNs) [16, 17, 18, 19], as the SMD extractor.
- A distance adaptive training (DAT) approach is proposed to utilize the SMD information more effectively.

Our experiments are conducted on a task of transcribing amateur videos. Compared with a baseline DNN, our distance-aware DNN model results in a notable reduction on word error rates (WERs). The DAT approach achieves better WERs than the simple concatenation. Moreover, SMD descriptors can be combined with speaker representations to realize comprehensive adaptive training that encodes both SMD variability and speaker characteristics.

2. Review of DNNs

The architecture of the DNN we use is shown in Figure 1. A DNN is a multilayer perceptron (MLP) which consists of many hidden layers before the softmax output layer. Each hidden layer computes the outputs of hidden units given the input vector. We denote the feature vector at the t-th frame as $\mathbf{o}_t$. Normally $\mathbf{o}_t$ is the concatenation of multiple neighbouring frames surrounding t. The quantities shown in Figure 1 can be computed as:

$$\mathbf{a}_t^i = \mathbf{W}_i \mathbf{x}_t^i + \mathbf{b}_i \qquad \mathbf{y}_t^i = \sigma(\mathbf{a}_t^i) \qquad 1 \leq i \leq L \qquad (1)$$

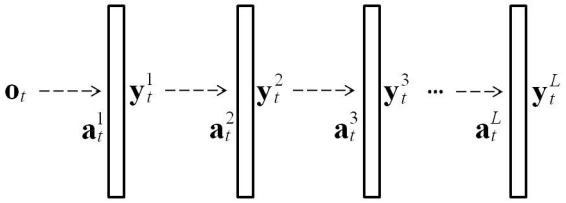

Figure 1: Architecture of the DNN model.

where L is the total number of layers, the weight matrix $\mathbf{W}_i$ connects the i-1-th and i-th layers, and $\mathbf{b}_i$ is the bias vector of the i-th layer. The inputs to the i-th layer $\mathbf{x}_t^i$ can be formulated as:

$$\mathbf{x}_t^i = \begin{cases} \mathbf{o}_t & i = 1 \\ \mathbf{y}_t^{i-1} & 1 < i \leq L \end{cases} \quad (2)$$

The activation function $\sigma(x)$ at the final layer $i = L$ takes the form of the softmax function. When used as a hybrid model, the DNN is trained to classify speech frames to context-dependent (CD) states. Outputs from the DNN represent the posterior probabilities of CD states given the input $\mathbf{o}_t$.

3. Extraction of SMD Descriptors

We formulate the extraction of the SMD information as a problem of bottleneck-feature (BNF) generation. Three architectures are investigated as the building block of the extractor.

3.1. SMD Extraction with DNNs

Our first SMD extractor is a DNN that has a bottleneck layer significantly narrower than the other hidden layers. This bottleneck layer squeezes the discriminative information into a low-dimensional space. The network is trained on a dataset where the SMD type (close-talking, distant, etc.) of each acoustic frame is known. For example, in the ICSI meeting corpus [20], each meeting session is recorded with multiple microphones, and details regarding the locations of the microphones are provided. Training of the network is to classify speech frames to the SMD types. After network training, activations from the bottleneck layer are treated as SMD descriptors that capture the SMD variability dynamically at the frame level.

Our previous work [21] has established the deep BNF (DBNF) architecture for better BNF extraction. DBNF differs from the previous BNF approaches [22, 23] in that the hidden layers are arranged in an asymmetric manner around the bottleneck layer. In particular, we insert multiple hidden layers prior to the bottleneck layer, and only one hidden layer between the bottleneck and the softmax layers. As discovered in [4], activations from higher layers of DNNs are more invariant to acoustic distortions. In this work, we apply this DBNF structure as the SMD extractor.

3.2. SMD Extraction with CNNs

Instead of fully-connected (FC) weight matrices, CNNs are characterized by local filters which capture locality along the frequency bands. On top of the convolution layers, max-pooling layers are usually added to improve the shift invariance in the frequency domain. Because of these configurations, CNNs have been shown to outperform DNNs on acoustic modeling tasks [13, 14, 15, 24]. In highly challenging acoustic conditions, the quality of the SMD descriptors might be undermined by spectral distortions such as noise and reverberation. Applying CNNs as the SMD extractor enables us to obtain SMD descriptors robust to these distortions. Following [15], our CNN-based extractor contains two convolution layers, on top of which multiple FC layers are added. One of the FC layers is a bottleneck layer for SMD descriptors generation. More details about our CNN settings are presented in Section 5.2.

3.3. SMD Extraction with RNNs

Both DNNs and CNNs can only model limited temporal dependency within the fixed-size context window. To resolve this limitation, previous work [18, 19, 25] has studied the application of RNNs to acoustic modeling. Unlike the standard feed-forward networks, the RNN has self-connections on its hidden layers. These connections allow temporal information to be propagated through many time steps. Given an input sequence $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_T)$, a recurrent layer iterates from $t = 1$ to T to compute the sequence of hidden states $\mathbf{Y} = (\mathbf{y}_1, ..., \mathbf{y}_T)$ via the following equations:

$$\mathbf{y}_t = \sigma(\mathbf{W}_{yx}\mathbf{x}_t + \mathbf{W}_{yy}\mathbf{y}_{t-1} + \mathbf{b}_y) \quad (3)$$

where $\mathbf{W}_{yx}$ is the input-to-hidden weight matrix, $\mathbf{W}_{yy}$ is the hidden-to-hidden (recurrent) weight matrix. We can see that the hidden activations $\mathbf{y}_{t-1}$ from the previous time step are recurrently fed to influence the hidden outputs $\mathbf{y}_t$ at the current time step.

In this paper, we use the more complicated Long Short-Term Memory (LSTM) [26] to construct RNNs. LSTM is a special recurrent layer that exploits memory cells to store temporal information and purpose-built gates to control the information flow. These modifications make RNNs particularly suited for modeling long-range temporal dynamics. In this work, we use a deep LSTM-RNN architecture, which stacks multiple LSTM layers, as the SMD extractor. Within each LSTM layer, following [19], we add a linear projection layer that transforms the memory cell activations into lower-dimensional outputs. In [19], adding this projection layer is found to reduce model parameters and generate better recognition accuracy. Interested readers can refer to [19] for more details.

4. Distance-Aware DNNs

The trained SMD extractor is transferred to our target acoustic modeling task on which distance-aware DNNs are then built. Note that the inputs to the SMD extractor and the inputs to the DNN acoustic model are not constrained to be identical. They may be trained with different feature types, e.g., MFCCs and filterbanks. Even with the same feature type, the SMD extractor and the DNN model may require different normalization on their front-end. At the t-th frame, the input vector of the DNN model is denoted as $\mathbf{o}_t$, and the inputs of the SMD extractor as $\mathbf{r}_t$. By feeding $\mathbf{r}_t$ to the extractor, we obtain its bottleneck layer activations $\mathbf{d}_t$ as the SMD descriptors. Two methods are investigated to incorporate these descriptors into DNN training.

4.1. Simple Concatenation

A simple way is to append these SMD descriptors to the original DNN inputs. Then, the DNN model is built over the augmented feature vectors $[\mathbf{o}_t, \mathbf{d}_t]$. During fine-tuning, the bottom layers are trained to fuse the SMD information and the acoustic features with non-linear transformations. The activations from these bottom layers become more invariant across SMD condi-

tions, and thus benefit the CD states classification performed by the upper layers.

4.2. Distance Adaptive Training

In our previous work [27, 28], we have presented a framework to perform speaker adaptive training (SAT) for DNN models. This approach requires an i-vector [29] to be extracted for each speaker. Based on the well-trained speaker-independent (SI) DNN, a separate *adaptation neural network* is learned to convert i-vectors into speaker-specific linear feature shifts. Adding these shifts to the original DNN inputs (i.e., $\mathbf{o}_t$) produces a speaker-normalized feature space. Parameters of the SI-DNN are re-updated in this new space, which finally generates the SAT-DNN model.

This idea can be naturally ported to the SMD descriptors, and we conduct distance adaptive training (DAT) for DNNs. Specifically, we replace the i-vector representations with the SMD descriptors. A distance-normalized feature space is similarly derived by learning the adaptation network. Note that in this case, the linear feature shifts generated by the adaptation network are frame-specific rather than speaker-specific. Re-updating the parameters of the DNN in the normalized feature space gives us the adaptively trained DAT-DNN model. This DAT-DNN model becomes independent of specific SMD conditions, and thus generalizes better to unseen distance variability.

5. Experiments

5.1. Experimental Setup

5.1.1. Dataset

Following [28, 30], our target acoustic modeling task is to transcribe real-world *instructional videos*. To create the dataset, we download a collection of English videos from online video archives. These videos are uploaded by social media users to share expertise on specific tasks (e.g., oil change, sandwich making, etc.). Unlike broadcast news videos that are produced professionally, these amateur videos are shot using various types of portable devices (e.g., cameras, cellphones, etc.) with far-field microphones. Also, in many of the videos, the speakers frequently change their locations relative to the microphones. For example, in a video about soccer skills, the speaker has to move around in order to perform various actions. All these factors make the SMD vary a lot, not only within the same video but also within a single utterance. This SMD variability poses special challenges to acoustic modeling on this task. On average, the downloaded videos have the duration of 90 seconds. For each video, the manual transcripts have been provided by the uploading user. We take several steps to convert the collected data into an ASR corpus. These steps include transcripts normalization, utterance filtering, audio downsampling, and lexicon expansion. We finally get 94 hours of speech, out of which 90 hours are selected for training and 4 hours for testing. A video is taken as a speaker. For decoding, a trigram language model (LM) is trained on the training transcripts. This LM is then interpolated with another trigram LM trained on an additional set of 300 hours of instructional-video transcripts.

We train the SMD extractor on the ICSI meeting corpus [20]. In this corpus, each meeting session has been recorded with microphones laid out at different distances from the speakers. However, the total number of channels is not constant across meeting sessions. Moreover, not enough details regarding the channels are provided in the corpus that enables us to align channels across meetings. For instance, the channels marked with "#1" in two different meetings are not necessarily referring to the same microphone. Therefore, instead of only distance types, the combinations of speakers and distance types are taken as the labels, which totally results in 2311 classes. A SMD extractor can be learned by taking these classes as the targets.

5.1.2. Baseline GMM and DNN Models

Our GMM models are built with the open-source Kaldi toolkit [31]. We first train the initial MLE model using 39-dimensional MFCC+$\triangle$+$\triangle\triangle$. Then 7 frames of MFCCs are spliced together and projected down to 40 dimensions with linear discriminant analysis (LDA). A maximum likelihood linear transform (MLLT) is applied on the LDA features and generates the LDA+MLLT model. Discriminative training with the boosted maximum mutual information (BMMI) objective [32] is finally performed over the LDA+MLLT model. The GMM model has 3819 tied triphone states and an average of 16 Gaussian components per state.

We construct DNN models using our PDNN framework [33]. DNN inputs include 11 neighbouring frames (5 on each side of the center frame) of 40-dimensional log-scale filterbank coefficients, with per-video mean and variance normalization. The DNN model has 6 hidden layers each of which contains 1024 neurons. Parameters of the network are initialized randomly. DNN fine-tuning uses frame labels generated through forced alignment with the LDA+MLLT GMM model. The cross-entropy (CE) objective is optimized based on a decaying "newbob" learning rate schedule. Specifically, the learning rate starts from 0.08 and remains unchanged until the increase of the frame accuracy on a cross-validation set between two consecutive epochs falls below 0.2%. Then the learning rate is decayed by a factor of 0.5 at each of the subsequent epochs. The whole learning process terminates when the frame accuracy fails to improve by 0.2% between two successive epochs. We adopt the mini-batch size of 256 and the momentum of 0.5 for stochastic gradient descent (SGD). Table 1 shows the WERs of the BMMI-GMM and DNN models on the 4-hour testing set.

Table 1: *Results (% WER) of the BMMI-GMM and baseline DNN models on the testing set.*

Model	WER%
BMMI-GMM	26.1
DNN	23.4

5.2. Experiments of SMD Extractors

In this section, we firstly investigate the optimal configurations for the SMD extractor. Training of the extractor uses filterbanks as the features. However, instead of per-video normalization, we apply global mean and variance normalization to preserve channel and speaker variation across videos. The trained SMD extractor is applied to our video-transcribing dataset, generating the SMD descriptors from its bottleneck layer. We employ the simple concatenation method for incorporating the SMD information. When formulated as a DNN, the extractor contains 6 hidden layers in which the 5-th layer is the bottleneck layer. All the non-bottleneck hidden layers have 1024 units. Table 2 shows the WERs of the resulting distance-aware DNNs when the SMD descriptors have different dimensions.

In the best case, we place 100 units at the bottleneck layer and have the WER of 22.1%. As a comparison, we train the DNN-based SMD extractor using the 468 speakers (instead of speaker-channel combinations) as the targets. In this case, the distance-aware DNN gets a much worse WER 23.0%. This verifies the necessity of adding channel (distance) targets in SMD extractor training.

Section 3 presents three architectures to instantiate the SMD extractor. We compare the performance of these architectures in Table 2. The CNN architecture follows [15, 28], consisting of 2 convolution layers. The convolution operation is applied over both time and frequency. Atop of the convolution layres, 4 FC hidden layers and finally the softmax layer are placed. The 5-th hidden layer, i.e., the 3-rd FC layer, is the bottleneck layer which has 100 neurons. In our LSTM-RNN architecture, we have 2 LSTM layers, each of which contains 800 memory cells and 512 output units. On top of these two LSTM layers, we have a bottleneck layer with 100 units which is followed by the final softmax layer. Unlike DNNs and CNNs, the LSTM-RNN takes single frames of filterbanks as its inputs, without any context splicing.

From Table 2, we observe that applying the CNN as the SMD extractor gives no improvement over the DNN extractor. This is partly because although showing rich SMD variability, our dataset is relatively clean, without much noise and reverberation. The advantage of CNNs in normalizing spectral distortions cannot be manifested under this condition. In comparison, the application of the LSTM-RNN results in more obvious gains (0.3% absolute). This demonstrates the ability of LSTM-RNNs to learn more accurate SMD descriptors by exploiting long-term temporal dependency. To this end, the distance-aware DNN achieves the WER of 21.8%, which translates to 6.8% relative improvement over the DNN baseline (23.4%).

Table 2: *Results (% WER) of the SMD extractors with different configurations and architectures. "SMD Dim" refers to the dimension of the SMD descriptors, i.e., the size of the bottleneck layer in the SMD extractor.*

SMD Extractor	SMD Dim	WER%
DNN	50	22.4
DNN	100	22.1
DNN	150	22.3
CNN	100	22.1
LSTM-RNN	100	21.8

5.3. Results of DAT-DNNs

In addition to the simple concatenation, Section 4 proposes DAT for incorporation of the SMD descriptors. In this section, we experimentally study the performance of DAT-DNN models. As with SAT of DNNs [27, 28], building of DAT-DNN models starts from the DNN baseline which has been well trained in Section 5.1.2. An adaptation network is learned to convert the SMD descriptors into frame-level linear features shifts. This adaptation network contains 3 hidden layers, each of which has 512 units and uses the sigmoid activation function. Its output layer has 440 (the dimension of the filterbank features) units and uses the identity function $f(x) = x$. After adding the shifts to the original DNN inputs, we obtain the distance-normalized feature space, in which the DNN model is re-updated. This gives us the DAT-DNN model. Training of the adaptation network and updating of the DNN use the standard back-propagation.

During decoding, we adapt the DAT-DNN model simply by extracting the SMD descriptors on the testing speech frames, feedforwarding the descriptors through the adaptation network, and adding the feature shifts to the original filterbank features. The DAT-DNN model is finally decoded in the normalized feature space. We use the SMD descriptors generated by the LSTM-RNN. Table 3 shows the results of the DAT-DNN model. Due to more complicated integration of the SMD information, our DAT approach outperforms the simple concatenation.

For complete evaluation, we also build the SAT-DNN model by extracting a 100-dimensional i-vector for each speaker. From Table 3, we observe that the improvement (22.0% vs 23.4%) of the SAT-DNN over the baseline DNN is not as large as the improvement reported in [28]. This is because the video-level i-vectors are insufficient to capture the rich SMD variability within videos/utterances. In contrast, DAT can model the SMD dynamics by taking advantage of the frame-level SMD descriptors. As a result, the DAT-DNN model ends up to achieve better results than the SAT-DNN. Finally, we concatenate the i-vectors and SMD descriptors into an expanded representation, which encodes both the video-level speaker characteristics and the frame-level SMD dynamics. Adaptive training is similarly carried out over these new representations. Table 3 shows that the resulting *SAT+DAT-DNN* model obtains a better WER than both the SAT-DNN and the DAT-DNN. Compared with the DNN baseline, the SAT+DAT-DNN results in 9.0% relative improvement (21.3% vs 23.4%).

Table 3: *Results (% WER) of the adaptively trained DNN models. "Feat" means the type of additional descriptors used in adaptive training, e.g., SMD descriptors for DAT.*

Adaptive Model	Feat	WER%
DAT-DNN	SMD descriptors	21.5
SAT-DNN	I-vectors	22.0
SAT+DAT-DNN	SMD descriptors + I-vectors	21.3

6. Conclusions and Future Work

In this paper, we have investigated the impact of dynamic SMD on the performance of DNN acoustic models. To alleviate the effect of SMD, we build distance-aware DNN models by explicitly incorporating SMD descriptors into DNN training. Extraction of the SMD information is achieved by a universal extractor that is learned on a meeting corpus. Our experiments show that distance-aware DNNs achieve notable WER improvement over the DNN baseline. Optimizations to the SMD information extraction and integration result in further WER reduction.

In our future work, we plan to quantity the SMD variability of a video or utterance based on the frame-level SMD descriptors. This allows us to more closely examine the correlation between the SMD variability and the difficulty of ASR. Also, we will explore other architectures to extract SMD descriptors with special characteristics, e.g., deep maxout networks (DMNs) for sparse SMD features [34, 24].

7. Acknowledgements

This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number OCI-1053575.

8. References

[1] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 1, pp. 30–42, 2012.

[2] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. 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.

[3] 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.

[4] D. Yu, M. L. Seltzer, J. Li, J.-T. Huang, and F. Seide, "Feature learning in deep neural networks-studies on speech recognition tasks," *arXiv preprint arXiv:1301.3605*, 2013.

[5] 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 (TASLP)*, vol. 22, no. 4, pp. 745–777, 2014.

[6] Y. Huang, D. Yu, C. Liu, and Y. Gong, "A comparative analytic study on the gaussian mixture and context dependent deep neural network hidden markov models," in *Fifteenth Annual Conference of the International Speech Communication Association (INTERSPEECH)*. ISCA, 2014.

[7] P. Swietojanski, A. Ghoshal, and S. Renals, "Hybrid acoustic models for distant and multichannel large vocabulary speech recognition," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 2013, pp. 285–290.

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

[9] D. Marino and T. Hain, "An analysis of automatic speech recognition with multiple microphones." in *Twelfth Annual Conference of the International Speech Communication Association (INTERSPEECH)*. ISCA, 2011, pp. 1281–1284.

[10] T. Yoshioka and M. J. Gales, "Environmentally robust asr front-end for deep neural network acoustic models," *Computer Speech & Language*, vol. 31, no. 1, pp. 65–86, 2015.

[11] M. L. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7398–7402.

[12] G. Gravier, G. Potamianos, and C. Neti, "Asynchrony modeling for audio-visual speech recognition," in *Proceedings of the second international conference on Human Language Technology Research*. Morgan Kaufmann Publishers Inc., 2002, pp. 1–6.

[13] T. N. Sainath, A.-r. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for lvcsr," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 8614–8618.

[14] O. Abdel-Hamid, A.-R. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 22, no. 10, pp. 1533–1545, 2014.

[15] H. Soltau, G. Saon, and T. N. Sainath, "Joint training of convolutional and non-convolutional neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 5572–5576.

[16] O. Vinyals, S. V. Ravuri, and D. Povey, "Revisiting recurrent neural networks for robust asr," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 4085–4088.

[17] A. L. Maas, Q. V. Le, T. M. O'Neil, O. Vinyals, P. Nguyen, and A. Y. Ng, "Recurrent neural networks for noise reduction in robust asr," in *Thirteenth Annual Conference of the International Speech Communication Association (INTERSPEECH)*. ISCA, 2012.

[18] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 6645–6649.

[19] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Fifteenth Annual Conference of the International Speech Communication Association (INTERSPEECH)*. ISCA, 2014.

[20] A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke *et al.*, "The icsi meeting corpus," in *Acoustics, Speech and Signal Processing (ICASSP), 2003 IEEE International Conference on*. IEEE, 2003, pp. 364–367.

[21] J. Gehring, Y. Miao, F. Metze, and A. Waibel, "Extracting deep bottleneck features using stacked auto-encoders," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 3377–3381.

[22] F. Grezl and P. Fousek, "Optimizing bottle-neck features for lvcsr," in *Acoustics, Speech and Signal Processing (ICASSP), 2008 IEEE International Conference on*. IEEE, 2008, pp. 4729–4732.

[23] D. Yu and M. L. Seltzer, "Improved bottleneck features using pretrained deep neural networks." in *Twelfth Annual Conference of the International Speech Communication Association (INTERSPEECH)*. ISCA, 2011, pp. 237–240.

[24] Y. Miao and F. Metze, "Improving language-universal feature extraction with deep maxout and convolutional neural networks," in *Fifteenth Annual Conference of the International Speech Communication Association (INTERSPEECH)*. ISCA, 2014.

[25] H. Sak, O. Vinyals, G. Heigold, A. Senior, E. McDermott, R. Monga, and M. Mao, "Sequence discriminative distributed training of long short-term memory recurrent neural networks," in *Fifteenth Annual Conference of the International Speech Communication Association (INTERSPEECH)*. ISCA, 2014.

[26] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[27] Y. Miao, H. Zhang, and F. Metze, "Towards speaker adaptive training of deep neural network acoustic models," in *Fifteenth Annual Conference of the International Speech Communication Association (INTERSPEECH)*. ISCA, 2014.

[28] Y. Miao, L. Jiang, H. Zhang, and F. Metze, "Improvements to speaker adaptive training of deep neural networks," in *2014 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2014.

[29] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 4, pp. 788–798, 2011.

[30] J. Chiu, Y. Miao, A. Black, and A. Rudnicky, "Distributed representation-based spoken word sense induction," in *Sixteenth Annual Conference of the International Speech Communication Association (INTERSPEECH) (To Appear)*. ISCA, 2015.

[31] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlíček, Y. Qian, P. Schwarz, J. Silovský, G. Stemmer, and K. Veselý, "The kaldi speech recognition toolkit," in *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*. IEEE, 2011, pp. 1–4.

[32] D. Povey, "Discriminative training for large vocabulary speech recognition," Ph.D. dissertation, University of Cambridge, 2005.

[33] Y. Miao, "Kaldi+pdnn: building dnn-based asr systems with kaldi and pdnn," *arXiv preprint arXiv:1401.6984*, 2014.

[34] Y. Miao, F. Metze, and S. Rawat, "Deep maxout networks for low-resource speech recognition," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 2013, pp. 398–403.