Low latency modeling of temporal contexts

Vijayaditya Peddinti*, Yiming Wang, Daniel Povey, Sanjeev Khudanpur

***Abstract*—Bidirectional long short term memory (BLSTM) acoustic models provide significant word error rate reduction compared to their unidirectional counterpart, as they model both the past and future temporal contexts. However it is non-trivial to deploy bidirectional acoustic models for online speech recognition due to an increase in latency. In this paper we propose the use of temporal convolution, in the form of time-delay neural network (TDNN) layers, along with unidirectional LSTM (ULSTM) layers to outperform the BLSTMs, while limiting the latency to 200 ms. This also reduces the computational complexity both during training and inference. Further we also improve the state-of-the-art BLSTM acoustic models by operating them at different frame rates at different layers and show that the proposed model matches the performance of these mixed frame rate BLSTMs. We present results on the Switchboard 300 Hr LVCSR task and the AMI LVCSR task, in the three microphone conditions.**

***Index Terms*—time delay neural networks, recurrent neural networks, LSTM, acousticmodels**

I. Introduction

The use of future context information is typically shown to be helpful for acoustic modeling. This context is provided in feed-forward neural networks (FFNNs) by splicing a fixed set of future frames in the input representation [1] or through temporal convolution over the future context [2]. In unidirectional LSTM (ULSTM) acoustic models this is accomplished using a delayed prediction of the output labels [3], while in bidirectional LSTMs (BLSTMs) this is accomplished by processing the data in the backward direction using a separate LSTM layer [4], [5], [6].

Among the recurrent neural network acoustic models, including their variants like long short term memory (LSTM) and highway LSTM [7] networks, the bidirectional versions have been shown to outperform the unidirectional versions by a large margin [8], [7]. However the latency of these bidirectional models is significantly large, which makes them unsuitable for online speech recognition applications. To overcome this limitation chunk based training and decoding schemes [8], [9], [10] have been investigated. Further there has also been effort in using lower complexity recurrent neural networks e.g. simple RNN, for modeling the backward direction [11].

In this paper we propose the use of temporal convolution for modeling the future temporal context. The TDNN layers are interleaved with ULSTM layers and this model is shown to outperform BLSTMs in two different LVCSR tasks. An average relative improvement of 5% is reported compared to BLSTM acoustic models, while enabling online decoding with a maximum latency of 200 ms and also reducing the computational complexity. As the model just uses TDNN and ULSTM layers it can be used to estimate posteriors even with frame-level increments of audio rather than chunk-level increments, as in BLSTMs. Further we improve the BLSTM models by using a mixed frame-rate and show that the proposed model matches the performance of these improved BLSTMs.

In this paper, we limit our attention to the purely sequence-trained acoustic models based on the lattice-free maximal mutual information (LF-MMI) [12] criterion. This is of significance as the use of a sequence based cost function significantly impacts the design choices in the neural network architecture especially for feed-forward neural networks.

The paper is organized as follows : Section II presents the prior work, Section III presents the motivation for this effort, Section IV describes the proposed model, Section V describes the experimental setup, Section VI presents the results and finally the conclusion is presented in Section VII.

II. Prior Work

The superior performance of BLSTM acoustic models has motivated recent research efforts [10], [7], [11], [8] to make them amenable for online decoding. A common characteristic of these methods is the use of frame chunks in place of the entire utterance, and they differ in the way the recurrent states are initialized when processing these chunks.

Chen *et al.*, [10] proposed the use of context-sensitive chunks (CSC), where a fixed context of frames is appended to the chunk to the left and right. These frames are used to intialize the recurrent states of the network. Zhang *et al.*, [7] carried over the recurrent states for the forward LSTM from previous chunks reducing the computation on the left context. In these methods the overall latency is reduced to chunk-width (N_c) plus chunk right context (N_r).

Xue *et al.*, [11] proposed the use of a feed-forward DNN to estimate the initial state of the backward LSTMs, for a given chunk. They also proposed the use of a simple RNN in place of an LSTM for the backward direction.

Zeyer *et al.*, [8] proposed the use of overlapping chunks, estimating the posteriors using the BLSTM for each chunk and combining the posterior estimates from overlapping chunks. The initial recurrent states for a given chunk are initialized to $\mathbf{0}$.

In this paper we propose the use of temporal convolution, in place of a recurrent layer, for modeling the future temporal context. This not only reduces the latency but also reduces the computational complexity.

*corresponding author (p.vijayaditya@gmail.com)

This work was partially supported by DARPA LORELEI Grant No HR0011-15-2-0024, NSF Grant No CRI-1513128 and IARPA Contract No 2012-12050800010.

Vijayaditya Peddinti, Daniel Povey and Sanjeev Khudanpur are with the Center for Language and Speech Processing(CLSP) and Human Language Technology Center of Excellence (HLTCoE), Johns Hopkins University (JHU), USA

Yiming Wang is with CLSP, JHU, USA

III. Motivation

In this section we present a set of empirical results which motivate the model proposed in this paper. We initially detail modeling of large temporal contexts using TDNNs and compare their performance with unidirectional and bidirectional LSTMs. These results correspond to the Hub5'00 set (LDC2002S09) and the 300 hr Switchboard LVCSR task. Please see Section V for the detailed experimental setup.

A. Sub-sampled TDNNs

Time-delay neural networks [2], a predecessor to the convolutional neural networks, have been previously shown to be effective in modeling long-span temporal contexts. However there is a linear increase in parameters with increase in temporal context, for the same hidden-activation dimension. Further there is a linear increase in computation when they are trained using frame-level objective functions (e.g. cross-entropy) as this training is typically done with frame randomization. In the sub-sampled time-delay neural networks [13] the linear increase, both in parameters and computation, is eliminated using a non-uniform sub-sampling technique where the frame rate decreases with the depth of the network. This sub-sampled TDNN (TDNN-A), is the baseline acoustic model in this paper.

B. Sub-sampled TDNNs and pure sequence training

Recently Povey *et al*. [12], demonstrated that neural networks can be trained from scratch using a sequence level objective function. When only a sequence-level objective function is used, frame-shuffling is no longer applicable during training. This eliminates the linear increase in computation, even without sub-sampling, as the sequence level objective functions require the computation of contiguous outputs and the computation can be amortized over all the outputs in the sequence. Hence the first modification to the sub-sampled TDNN architecture is increasing the frame rate even at the deeper layers in the network to match the output frame rate.

Recently lower output frame rate models have been shown to outperform conventional frame rate models, while providing great savings in computation. In [14], [12] and [15] the authors propose the use of reduced frame rates of 25-33 Hz for the neural network outputs. Hence we change the frame rate at all the layers in the TDNN to match the output frame rate (33 Hz), as was done in [12]. This configuration is denoted TDNN-B.

1) Higher frame-rates at lower layers: Finally, we also explore the use of higher frame rate (100 Hz) at the lower layers of the neural network. We restrict the higher frame rates to the lower layers as this still preserves the computational efficiency and we observed negligible difference in performance when increasing the frame rate even at the higher layers. This TDNN is denoted TDNN-C. Finally we tuned the temporal contexts at the TDNN layers (TDNN-D).

We specify the TDNN architectures in terms of the splicing indices which define the input of the temporal convolution kernel at each layer. e.g. $\{-3,0,3\}$ means that the input to the temporal convolution at a given time step t is a spliced version of previous layer outputs at times $t-3$, t, $t+3$. The configurations of the sub-sampled TDNNs described above and their performance is shown in Table I.

It can be seen that the using higher frame rates at lower layers and tuning the temporal contexts of the layers provides 10.3% relative gain over the sub-sampled TDNNs proposed in [13] [1].

C. Comparison with LSTMs

We compare the performance of the best TDNN model (TDNN-D) with ULSTM and BLSTM. The hyper-parameters of both these recurrent architectures were individually tuned for the best performance. The ULSTM, denoted ULSTM-A, is a three layer stacked ULSTM. It operates at an output sampling rate of 33 Hz, similar to the TDNN models. Further all the three ULSTM layers also operate at 33 Hz. In our neural network toolkit this is accomplished by using a delay of 3 and computing the LSTM layer outputs every third time step. For the BLSTM, the forward and backward direction LSTMs are operated in a similar fashion. These modifications to the LSTMs are similar to LSTM architectures used in [14] and [15].

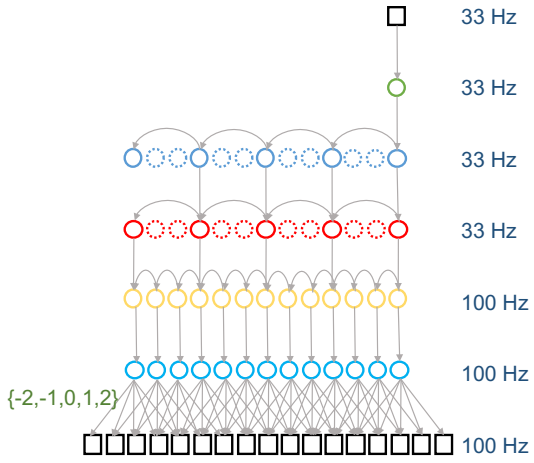

Fig. 1. Dependencies between activations at various layers and time-steps in the stacked ULSTM network with lowest layer operating at 100 Hz

1) Higher frame-rates at lower layer: As we observed that the use of a higher frame rate (100 Hz) was beneficial at the lower layers of the TDNNs, we explored increasing the frame rate to 100 Hz even in the lower layer of ULSTM and BLSTM models. This is accomplished by using a delay of 1 at the first LSTM layer, in place of 3. We denote these models as ULSTM-B and BLSTM-B. Figure 1 represents the dependencies between the activations across time steps and layers in the ULSTM-B.

Table II compares the models discussed in this section with the best TDNN model described in Section III-B. Firstly, it can be seen that operating the lower LSTM layers at a higher frame rate is beneficial (please compare ULSTM-A vs ULSTM-B and BLSTM-A vs BLSTM-B). Secondly, it can be seen that

[1] Part of this improvement was already realized in [12].

TABLE I
SUB-SAMPLED TDNN ARCHITECTURES

Model	Layer-wise context							Network context	WER (%)		
									SWBD	CHM	Total
TDNN-A	{-2,-1,0,1,2}	{-1,2}	{-3,3}	{-7,2}	{0}	{0}	{0}	[-13, 9]	11.1	21.8	16.5
TDNN-B	{-2,-1,0,1,2}	{-1,2}	{-3,0,3}	{-3,0,3}	{-3,0}	{0}	{0}	[-12, 10]	10.5	21.9	16.3
TDNN-C	{-2,-1,0,1,2}	{-1,0,1}	{-1,0,1}	{-3,0,3}	{-3,0,3}	{-3,0}	{0}	[-13, 10]	10.3	20.7	15.5
TDNN-D	{-1,0,1}	{-1,0,1}	{-1,0,1}	{-3,0,3}	{-3,0,3}	{-3,0,3}	{-3,0,3}	[-14, 14]	9.6	19.9	14.8

∗ Please note that the number of layers in the TDNN architectures is kept the same and the overall temporal context of the neural network is kept similar, except in TDNN-D, to make these architectures comparable. However due to the change in temporal convolution kernel size there is a slight increase in parameters as we move from TDNN-A to TDNN-D.

TABLE II
PERFORMANCE COMPARISON OF TDNN, ULSTM AND BLSTM ON SWBD LVCSR TASK

Model	WER (%)		
	SWBD	CHM	Total
TDNN-D	9.6	19.9	14.8
ULSTM-A	10.1	20.4	15.2
BLSTM-A	9.6	19.2	14.5
ULSTM-B	9.9	19.7	14.8
BLSTM-B	9.3	17.8	**13.5**

both TDNN and ULSTM models perform worse than both the BLSTM models. The use of a higher frame-rate at the lower layers results in a relative improvement of $\sim 7\%$ in BLSTM and $\sim 3\%$ in ULSTM. However the overall computational complexity increases by $\sim 35\%$ both during training and inference in these higher-frame rate models, compared to their lower frame-rate counterparts. Operating even the second LSTM layer at a higher frame rate did not lead to gains, while further increasing the overall computational complexity.

The superior performance of the bidirectional recurrent models compared to their unidirectional counterparts can be attributed to the modeling of the future context. To model the future context in the ULSTMs we propose the use of TDNN layers. These neural networks are discussed in Section IV.

IV. PROPOSED MODEL

A. Temporal convolution in ULSTMs

In this section we detail the use of temporal convolution in the recurrent neural networks. The TDNN layers, which perform temporal convolution, are used to model the right temporal context. We explore three different ways of combining temporal convolution and ULTSMs viz.,

- Stacking ULSTMs over TDNNs (TDNN-ULSTM-A)
- Stacking TDNNs over ULSTMs (TDNN-ULSTM-B)
- Interleaving TDNNs and ULSTMs (TDNN-ULSTM-C)

Figure 2 represents the dependencies between activations at various layers and at various time-steps, in the TDNN-ULSTM-C network. The layers with lateral connections represent the LSTM layers. To draw a correspondence with a BLSTM we could assume that the two TDNN layers between the consecutive forward direction LSTMs replace the backward direction LSTM of a BLSTM.

It can be seen that all the LSTM layers, which are computationally expensive, operate at a 33 Hz frame rate. Compared with the stacked ULSTM or BLSTM where the lowest layer operates at 100 Hz frame rate (ULSTM-B and BLSTM-B) the computational benefits are obvious.

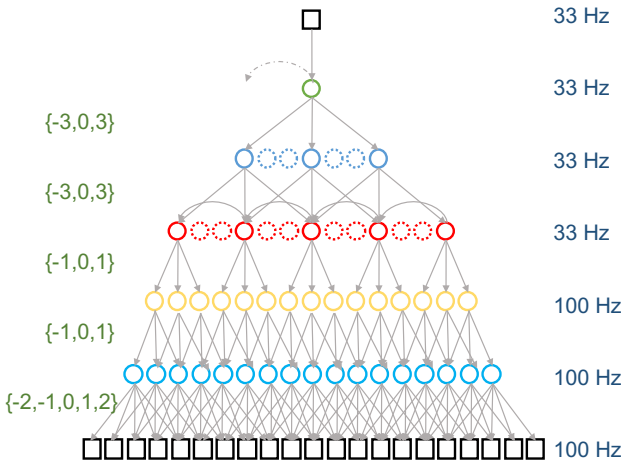

Fig. 2. Dependencies between activations at various layers and time-steps in a stacked TDNN-ULSTM network with interleaved temporal convolutions. The numbers on the left of the figure represent the temporal convolution kernel and the numbers on the right represent the frame-rate at the layer.

Further, to compare the benefits of performing temporal convolution in BLSTM models, we also interleave temporal convolution with the forward and backward LSTMs in a BLSTM network. This network is denoted TDNN-BLSTM-A.

As described above only the lower temporal convolution layers operate at a 100 Hz frame rate in TDNN-ULSTM-C. We also verify if additional gains can be had by operating even the lowest recurrent layer at 100 Hz frame rate, similar to ULSTM-B. This model is denoted TDNN-ULSTM-D.

V. EXPERIMENTAL SETUP

Experiments were conducted using the Kaldi toolkit [16]. We report the main set of results on 300 hour Switchboard conversational telephone speech task. The GMM-HMM system, used to generate the lattices for neural network training, the language model, the neural network training data preparation and decoding setup is similar to [13], [12]. For the convenience of the reader we describe this setup briefly.

We also present results on the AMI LVCSR task [17], [18], [19], for the individual headset microphone (IHM), single distant microphone (SDM) and multiple distant microphone

TABLE III
PERFORMANCE COMPARISON OF VARIOUS MODELS IN THE SWBD LVCSR TASK

Model	Architecture	WER (%)		
		SWBD	CHM	Total
TDNN-D	$T^{100}T^{100}T^{100}TTTT$	9.6	19.9	14.8
ULSTM-B*	$L_f^{100}L_fL_f$	9.9	20.0	15.0
BLSTM-B*	$[L_f^{100}, L_b^{100}][L_f, L_b][L_f, L_b]$	9.3	17.8	13.5
TDNN-ULSTM-A	$T^{100}T^{100}T^{100}TTTTL_fL_fL_f$	9.5	19.5	14.6
TDNN-ULSTM-B	$L_fL_fL_fT^{100}T^{100}T^{100}TTTT$	9.4	19.1	14.3
TDNN-ULSTM-C	$T^{100}T^{100}T^{100}L_fTTL_fTTL_f$	9.2	18.2	13.8
TDNN-ULSTM-D	$T^{100}T^{100}T^{100}L_f^{100}TTL_fTTL_f$	9.0	18.8	13.9
BLSTM-A *	$[L_f, L_b], [L_f, L_b], [L_f, L_b]$	9.6	19.2	14.5
TDNN-BLSTM-A	$T^{100}T^{100}T^{100}[L_fT, L_bT][L_fT, L_bT][L_f, L_b]$	9.2	18.8	14.1

∗ The depth and other hyper-parameters of the ULSTM and BLSTM models have been tuned for best performance.

(MDM) conditions. This recipe is similar to the one described in [20] with the salient difference being that the SDM and MDM LVCSR systems are trained with lattices generated from IHM data. For this we identified parallel segments in IHM audio corresponding to the utterances in SDM/MDM data. The IHM SAT HMM-GMM system was used to generate lattices from these parallel utterances.

A. Training recipe

All these training recipes use *speed-perturbation* based data augmentation [21], iVector adaptation of the neural network [22] and volume perturbation [13]. Further they also use an enhanced lexicon which models the pronunciation probabilities and probability of silence before and after each pronunciation [23]. The recipes to reproduce the experiments in this paper are available at [24].

We perform pure sequence training, *i.e.*, without frame level pretraining, using the lattice-free MMI objective [12]. It is computed using fixed length chunks of duration 1.5 second, with an extra context of 400 ms on left and right. Inference also uses the same settings.

VI. RESULTS

In this section we provide a comparison of acoustic models in the SWBD and AMI LVCSR tasks. We provide a short hand description of each network we discuss here. A forward LSTM is denoted L_f, backward LSTM is denoted L_b and a TDNN layer is denoted T. By default all the layers are assumed to operate at 33 Hz frame rate, and if the frame rate is different it is denoted in the super-script. e.g. TDNN-ULSTM-C can be described as $T^{100}T^{100}T^{100}L_fTTL_fTTL_f$ in this notation. Further layers which are operate on the same input and whose outputs are appended before passing to the next layer are represented using $[.,.]$ notation, e.g. the forward and backward LSTMs in a BLSTM are shown as $[L_f, L_b]$.

Table III presents a comparison of the acoustic models on the 300 Hr Switchboard LVCSR task. The TDNN layers in these acoustic models have the same temporal contexts as TDNN-D in Section III-B. It can be seen that TDNN-ULSTM-C performs the best among the three different TDNN-ULSTM models. Further it also closely matches the BLSTB-B performance. Also, operating the lowest LSTM layer in TDNN-ULSTM-C at 100 Hz, i.e., TDNN-ULSTM-D, was not beneficial.

TABLE IV
PERFORMANCE COMPARISON IN THE AMI LVCSR TASK

Model	WER (%)					
	IHM		SDM		MDM	
	Dev	Eval	Dev	Eval	Dev	Eval
TDNN-D	21.7	22.1	39.9	43.9	36.6	40.1
BLSTM-A	21.0	20.9	38.8	42.0	35.4	38.4
BLSTM-B	20.6	20.3	37.4	40.5	34.5	37.3
TULSTM-C*†	20.8	20.5	37.3	40.4	34.1	36.8
TBLSTM-A*	20.7	20.7	37.0	40.4	34.2	36.6

* TDNN-ULSTM is denoted as TULSTM and TDNN-BLSTM is denoted as TBLSTM in this table.
† TULSTM-C has additional TDNN layers between successive ULSTM layers

Table IV presents the results on the AMI-LVCSR task. It can be seen that the TDNN-ULSTM-C model performs better than BLSTM-A and also BLSTM-B. In preliminary experiments we observed that using additional temporal context was beneficial in this task. This might be attributed to the fact that this data is reverberated. This additional context was provided using an additional TDNN layer between successive ULSTM layers.

Interleaving temporal convolution with the forward and backward ULSTM layers in BLSTMs *i.e.,* TDNN-BLSTM-A was shown to perform better than BLSTM-A. However there was no consistent different in performance compared to TDNN-ULSTM-C and BLSTM-B across the two LVCSR tasks.

VII. CONCLUSION

In this paper we propose the use of temporal convolution layers for modeling the future temporal context. We show that this architecture performs better the conventional stacked BLSTM network. Further we improved the stacked BLSTM baseline by using a higher frame rate at the lowest BLSTM layer, and showed that the proposed neural network architecture matches the performance of this improved baseline.

The overall latency of the TDNN-ULSTM model is determined by the right context of the TDNN layers and the output delay of the ULSTM. For the TDNN-ULSTM-C model this was 200 ms. Further as the backward LSTM layers were replaced by temporal convolution layers, the computational complexity was also reduced.

REFERENCES

[1] H. A. Bourlard and N. Morgan, *Connectionist speech recognition: a hybrid approach*. Springer Science & Business Media, 1994, vol. 247.

[2] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang, "Phoneme recognition using time-delay neural networks," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 3, pp. 328–339, Mar. 1989.

[3] H. Sak, A. W. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling." in *Interspeech*, 2014, pp. 338–342.

[4] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[5] A. Graves, S. Fernández, and J. Schmidhuber, "Bidirectional LSTM networks for improved phoneme classification and recognition," *Artificial Neural Networks: Formal Models and Their Applications*, 2005.

[6] A. Graves, N. Jaitly, and A. R. Mohamed, "Hybrid speech recognition with Deep Bidirectional LSTM," *2013 IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU 2013 - Proceedings*, pp. 273–278, 2013.

[7] Y. Zhang, G. Chen, D. Yu, K. Yaco, S. Khudanpur, and J. Glass, "Highway long short-term memory rnns for distant speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 5755–5759.

[8] A. Zeyer, R. Schluter, and H. Ney, "Towards online-recognition with deep bidirectional LSTM acoustic models," *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, vol. 08-12-Sept, pp. 3424–3428, 2016.

[9] P. Doetsch, M. Kozielski, and H. Ney, "Fast and Robust Training of Recurrent Neural Networks for Offline Handwriting Recognition," *Proceedings of International Conference on Frontiers in Handwriting Recognition, ICFHR*, vol. 2014-Decem, pp. 279–284, 2014.

[10] K. Chen, Z.-J. Yan, and Q. Huo, "Training Deep Bidirectional LSTM Acoustic Model for LVCSR by a Context-Sensitive-Chunk BPTT Approach," in *Proceedings of the Interspeech*, 2015.

[11] S. Xue and Z. Yan, "Improving latency-controlled BLSTM acoustic models for online speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2017.

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

[13] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Proceedings of INTERSPEECH*, 2015.

[14] H. Sak, A. Senior, K. Rao, and F. Beaufays, "Fast and accurate recurrent neural network acoustic models for speech recognition," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[15] G. Pundak and T. N. Sainath, "Lower frame rate neural network acoustic models," in *Proceedings of Interspeech 2016*, 2016, pp. 22–26.

[16] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlíček, Y. Qian, P. Schwarz *et al.*, "The Kaldi speech recognition toolkit," in *Proceedings of ASRU*, 2011.

[17] I. McCowan, J. Carletta, W. Kraaij, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos *et al.*, "The AMI meeting corpus," in *Proceedings of the 5th International Conference on Methods and Techniques in Behavioral Research*, vol. 88, 2005.

[18] J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal *et al.*, "The AMI meeting corpus: A pre-announcement," in *International Workshop on Machine Learning for Multimodal Interaction*. Springer, 2005, pp. 28–39.

[19] S. Renals, T. Hain, and H. Bourlard, "Recognition and interpretation of meetings: The AMI and AMIDA projects," in *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU '07)*, 2007.

[20] V. Peddinti, V. Manohar, Y. Wang, D. Povey, and S. Khudanpur, "Far-field asr without parallel data," in *Proceedings of Interspeech*, 2016.

[21] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Proceedings of INTERSPEECH*, 2015.

[22] 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.

[23] G. Chen, H. Xu, M. Wu, D. Povey, and S. Khudanpur, "Pronunciation and silence probability modeling for asr." in *INTERSPEECH*, 2015, pp. 533–537.

[24] *Code to reproduce results of the experiments in this paper.*, 2017 (accessed March 23 , 2017), https://github.com/vijayaditya/kaldi/tree/tdnn_lstm.