TOWARDS DISCRIMINATIVELY-TRAINED HMM-BASED END-TO-END MODELS FOR AUTOMATIC SPEECH RECOGNITION

Hossein Hadian[1] *, *Hossein Sameti*[1], *Daniel Povey*[2,3], *Sanjeev Khudanpur*[2,3]

[1]Department of Computer Engineering, Sharif University of Technology, Iran,
[2]Center for Language and Speech Processing, Johns Hopkins University, Baltimore, MD, USA,
[3]Human Language Technology Center of Excellence, Johns Hopkins University, Baltimore, MD, USA.

ABSTRACT

In recent years, end-to-end approaches to automatic speech recognition such as CTC have gained more popularity as they have been successfully used for large vocabulary speech recognition. While these approaches simplify the training procedure, their performance is still usually far from that of state-of-the-art HMM-DNN methods such as LF-MMI (Lattice-free Maximum Mutual Information). In this study, we present our work on training LF-MMI models, in an end-to-end manner, i.e. without using alignments from a HMM-GMM model. We investigate the effect of MMI and also different HMM topologies, both in phoneme-based and character-based (i.e. lexicon-free) setups. Besides, we propose a trivial full biphone tree to enable context dependent (CD) modeling without building a decision tree, which further improves the results. Our experiments show that our end-to-end approach can achieve comparable results to regular LF-MMI and outperforms other end-to-end methods significantly in equal lexicon-free conditions.

Index Terms— Hidden Markov model, end-to-end, Automatic speech recognition, Lattice-free MMI, flat-start

1. INTRODUCTION

In recent years, end-to-end approaches to automatic speech recognition have received a lot of attention. These methods typically aim to train a neural-network-based acoustic model in one stage without relying on alignments from an initial model (usually an HMM-GMM model) [1] [2] [3]. Besides, it is desirable (for simplicity) not to use a lexicon or language model in these approaches, however using a language model significantly improves the results [4] [2] [5] [6].

On the other hand, conventional DNN-based speech recognition methods (i.e. CD-DNN-HMM) rely on alignments and phonetic decision trees from an HMM-GMM system[7]. These methods usually use a frame-level objective function - such as cross-entropy - for training the neural network using the mentioned alignments.

Although end-to-end models are appealing because of simplicity and faster production, they have not performed as well as conventional models in terms of word error rate for large vocabulary speech recognition tasks such as Switchboard [5].

Currently the two popular end-to-end approaches are CTC and sequence-to-sequence methods [8]. CTC (Connectionist Temporal Classification) introduces a new simple sequence-level objective function to enable training a neural network on sequences of speech signals without using prior alignments [9]. However, as shown in [10], CTC is a special case of HMM and similar results can be achieved using a sequence-level objective function that maximizes HMM likelihood.

On the other hand, sequence-to-sequence models use a whole different structure based on an encoder network which maps the input sequence into a fixed-sized vector and a decoder network which, using an attention mechanism, generates the output sequence using this vector as its input. These models have performed very well in a few tasks such as machine translation [11] but they are not as good for speech recognition tasks [6].

Recently, a new approach called LF-MMI (i.e. lattice-free MMI) was proposed which currently has the state-of-the-art results on many speech recognition tasks [12, 13, 14, 15]. This method, like CTC, uses a sentence-level posterior for training the neural network but unlike end-to-end approaches, still loosely relies on alignments from an HMM-GMM model. The objective function used in this method is maximum mutual information (MMI) in the context of hidden Markov models [16].

In the work presented here, we aim to train these powerful models without running the common HMM-GMM training and tree-building pipeline (i.e. in a flat-start manner). Previously two studies [17, 18] performed GMM-free training, but they used state-tying decision trees (created using alignments from the DNN model) for context dependent (CD) modeling. However, we do not use any kind of decision trees, and we do the whole training process in one stage (i.e. without doing re-alignments or building trees or performing prior estimation). Another difference is that we use LF-MMI objective function instead of ML (maximum likelihood) for training the network. We do both phoneme-based and character-based (i.e. lexicon-free) training and show that our end-to-end LF-MMI setup significantly outperforms other end-to-end approaches in equal conditions and performs almost as well as regular LF-MMI.

In the two following sections, regular LF-MMI and CTC will be briefly explained and then in Section 4, we will describe our end-to-end setup. The experimental setup will be explained in Section 5 and the experiments and results will be presented in Section 6. Finally the conclusions appear in Section 7.

2. REGULAR LF-MMI

Hidden Markov model (i.e. HMM) is a generative model commonly used for speech recognition. It is usually used jointly with a Gaussian Mixture Model, or a DNN (i.e. deep neural network) to model acoustic data. A common approach for learning the parameters of such models is through ML (i.e. maximum likelihood) estimation

*The first author performed the work while at CLSP, Johns Hopkins University.

The authors would like to thank Pegah Ghahremani and Vimal Manohar for their valuable comments.

which has the following objective function:

$$\mathcal{F}_{ML} = \sum_{u=1}^{U} \log p_\lambda(\boldsymbol{x}^{(u)}|\mathbb{M}_{\boldsymbol{w}^{(u)}}) = \sum_{u=1}^{U} \log \sum_{\boldsymbol{s}\in\mathbb{M}_{\boldsymbol{w}^{(u)}}} \prod_{t=0}^{T_u-1} p(s_{t+1}|s_t)p(x_t^{(u)}|s_t) \quad (1)$$

where λ is the set of all HMM parameters, U is the total number of training utterances, and $\boldsymbol{x}^{(u)}$ is the u^{th} speech utterance with transcription $\boldsymbol{w}^{(u)}$ and with length T_u. The composite HMM graph $\mathbb{M}_{\boldsymbol{w}^{(u)}}$ represents all the possible state sequences $\boldsymbol{s}$ pertaining to the transcription $\boldsymbol{w}^{(u)}$.

An alternative objective function is Maximum Mutual Information (i.e. MMI). MMI is a discriminative objective function which aims to maximize the probability of the reference transcription, while minimizing the probability of all other transcriptions:

$$\mathcal{F}_{MMI} = \sum_{u=1}^{U} \log \frac{p_\lambda(\boldsymbol{x}^{(u)}|\mathbb{M}_{\boldsymbol{w}^{(u)}})}{p_\lambda(\boldsymbol{x}^{(u)})} \quad (2)$$

The denominator can be written as:

$$p_\lambda(\boldsymbol{x}^{(u)}) = \sum_{\boldsymbol{w}} p_\lambda(\boldsymbol{x}^{(u)}|\mathbb{M}_{\boldsymbol{w}}) = p_\lambda(\boldsymbol{x}^{(u)}|\mathbb{M}_{den}) \quad (3)$$

where $\mathbb{M}_{den}$ is an HMM graph that includes all possible sequences of words. This is called the denominator graph, as opposed to $\mathbb{M}_{\boldsymbol{w}^{(u)}}$ which is called the numerator graph.

The denominator graph has traditionally been estimated using n-best lists and later using lattices [19]. That is because the full denominator graph can become quite large and it can make the computation significantly slow. Using a full denominator graph has been investigated in [19] with HMM-GMM models. More recently Povey et. al [12] used MMI training with HMM-DNN models using a full denominator graph (hence the name lattice-free) by adopting a few different techniques such as using a phone language model (instead of a word language model) and most importantly doing the denominator computation on GPU. The phone language model for the denominator graph is a pruned n-gram language model trained using the phone alignments of the training data. Also, they did not use the composite HMM as the numerator graph and instead used a special acyclic graph which can exploit the alignment information from a previous HMM-GMM model. Specifically, the numerator graph in the LF-MMI method, is an expanded version of the composite HMM , where the amount of expansion of the self-loops for each utterance is determined according to its alignment. As a result the numerator graph is an acyclic graph with only one initial and one final state, where the length of each path from the initial state to the final state is exactly the same and equals the number of frames of the utterance [1].

The HMM transitions are fixed and uniform in regular LF-MMI because adjusting them does not improve the results [12]. Also, the phone model used is a 2-state HMM as shown in Figure 1c.

[1] In fact, it equals the number of frames of the utterance divided by frame-subsampling-factor

3. CTC

The CTC method uses a blank label - which can appear between characters - to define an objective function which sums over all possible alignments of the reference label sequence with the input sequence of speech frames [9]:

$$\mathcal{F}_{CTC} = \sum_{u=1}^{U} \log p(\boldsymbol{w}^{(u)}|\boldsymbol{x}^{(u)}) = \sum_{u=1}^{U} \log \sum_{\boldsymbol{\pi}\in\mathcal{B}^{-1}(\boldsymbol{w}^{(u)})} \prod_{t=0}^{T_u-1} p(\pi_t|x_t^{(u)}) \quad (4)$$

where $p(\pi|x_t^{(u)})$ is the network output for label π at time t given utterance $\boldsymbol{x}^{(u)}$, and $\mathcal{B}$ is a many-to-one map that removes repetitive labels and then blanks from a label sequence.

3.1. Relation to HMM

The CTC objective function can be thought of as HMM likelihood over a composite HMM, where each label (e.g. a character, in character-based CTC) has a special 2-state HMM topology as shown in Figure 1a. If we create the composite HMM by starting with a blank state (with a self-loop and a forward null transition) and concatenate the label HMMs, while inserting a single blank state between repetitive labels, we can see that the set of all paths in this composite HMM is identical to the set $\{\boldsymbol{\pi}|\boldsymbol{\pi} \in \mathcal{B}^{-1}(\boldsymbol{w}^{(u)})\}$. Therefore, comparing equations 1 and 4 we can see that CTC is a special case of HMM, when the state priors, observation priors, and transition probabilities are all uniform and fixed. Since CTC is the first successful method used for end-to-end speech recognition, we will use its HMM topology in our proposed setup to compare it with other more common HMM topologies shown in Figure 1.

4. END-TO-END LF-MMI

In this work, we remove any dependencies on HMM-GMM alignments and the state-tying tree to make the LF-MMI method end-to-end. In LF-MMI, the output neurons correspond to tied triphone HMM states, where the tying is done according to a decision tree. This decision tree is in turn created using alignments from an HMM-GMM system [20]. We remove this dependency by using monophones or full biphones (please see Section 4.1) instead of triphones. Besides, we use the composite HMM (with self-loops) as the numerator graph instead of the special acyclic graph used in regular LF-MMI.

As a result, unlike regular LF-MMI, there is no prior alignment information in the numerator graph and there is no restriction on the self-loops so there is much more freedom for the neural network to learn the alignments. Besides, since we do not have alignments for the training data, to estimate the phone language model for the denominator graph, we use the training transcriptions (choosing a random pronunciation for words with alternative pronunciations), after inserting silence phones with probability 0.2 between the words and with probability 0.8 at the beginning and end of the sentences.

The derivatives for MMI are as follows:

$$\frac{\partial \mathcal{F}_{MMI}}{\partial y_t^{(u)}(s)} = {}^{NUM}\gamma_t^{(u)}(s) - {}^{DEN}\gamma_t^{(u)}(s) \quad (5)$$

where $y_t^{(u)}(s)$ is the network output for state s at time t given input utterance u which we interpret as the logarithm of HMM state likelihood (i.e. $\log p(x_t|s)$) since state priors have no effect in MMI training [12]. ${}^{NUM}\gamma_t^{(u)}(s)$ is the numerator HMM occupation probability for state s at time t for utterance u, and ${}^{DEN}\gamma_t^{(u)}(s)$ is the same but for the denominator graph. For later reference, the derivatives for ML (maximum likelihood) are as follows:

$$\frac{\partial \mathcal{F}_{ML}}{\partial y_t^{(u)}(s)} = {}^{NUM}\gamma_t^{(u)}(s) \tag{6}$$

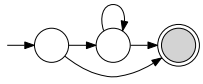
The HMM transitions are fixed in our setup. Training them won't make any difference as long as there is no state tying because their effect can be fully replicated by the neural network output. In other words, the network will ignore them.

4.1. Tree-free context-dependent modeling

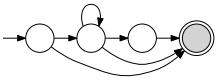
Our initial experiments with monophone end-to-end LF-MMI showed a remarkable gap between the results of end-to-end and regular LF-MMI. We speculate that part of this gap might be due to the lack of phone context in our end-to-end setup. Therefore, we propose a simple yet effective approach to model left biphones (or *bichars* in the lexicon-free case) by creating a *trivial full biphone tree*. This tree is not pruned at all (and does not do any tying), so there is no need for alignments and the approach is still end-to-end (in the sense of not requiring any previously trained models). In other words, we assume a separate HMM model for each and every possible pair of phonemes (or characters in lexicon-free conditions). [2] This will create biphones that never occur in the training data, but as we will see in Section 6.3, the network learns to ignore them.

(a) CTC's HMM topology

(b) 1-state HMM topology



(c) 2-state HMM topology



(d) 3-state HMM topology

Fig. 1. Different HMM topologies. The state marked with "-" is CTC's blank state and is shared across all the labels.

5. EXPERIMENTAL SETUP

We do our experiments on two ASR corpora: Switchboard [21], and WSJ (Wall Street Journal) [22]. Switchboard is a database with 300 hours of transcribed speech. We do our evaluations on the "switchboard" portion of the Hub5 '00 set (also called "eval2000") but in the final results, we also report word error rates on the full eval2000 set (i.e. Callhome and switchboard). We use a 4-gram language model trained on Fisher+Switchboard training transcriptions. WSJ is a corpus of read-speech with 81 hours of transcribed speech. We do our tests on the "eval92" and "dev93" subsets using a 3-gram language model trained on the training transcriptions using an extended lexicon as used in [4] and [2].

5.1. Experimental setup

For running the experiments, we use the Kaldi speech recognition toolkit [23]. This toolkit is open-source and the source codes related to this study are available online for reproducing the results. We do not use i-vectors or other speaker adaptation techniques in any of the experiments. In all the experiments, by default we use a TDNN network structure [24] or where stated, a TDNN-LSTM structure [13]. The TDNN structure is a time-delay neural network with 7 hidden layers, while the TDNN-LSTM structure is a network with interleaving Long Short-Term Memory layers and TDNN layers. Please refer to [13] for more details. As in [12], we use a frame subsampling factor of 3. This speeds up training by a factor of 2. We also do 3-fold *speed perturbation* in all the experiments [25].

In all the end-to-end experiments, we use SGD (Stochastic Gradient Descent) to train the network (in a single stage, for 4 epochs), on 40-dimensional MFCC features extracted from 25ms frames every 10ms. The features are normalized on a per-speaker basis to have zero mean and unit variance. Other than this, no feature normalization or feature transform is used. The network parameters are initialized randomly to have zero mean and a small variance. Besides, unlike other works, we do not perform re-alignments during training, we do not change acoustic or language model scales during training, and we do not use state-tying trees of any kind when doing context dependent (CD) modeling (Section 4.1).

In regular LF-MMI, all utterances are split into chunks of 150 frames to make GPU computations efficient. However, in end-to-end LF-MMI, we can't split the utterances because we don't have alignments. Instead, we ensure that all the utterances are modified to be of around 20 distinct lengths. We use speed perturbation to modify the length of each utterance to the nearest of the distinct lengths. Alternatively, we can pad each utterance with silence to reach one of the distinct lengths.

6. EXPERIMENTS

6.1. ML vs. MMI

Ever since MMI was introduced, it has been usually used only after ML (i.e. maximum likelihood) training of the model. In other words, MMI estimation is not usually used in a flat-start manner but instead it is used when the parameters have been already estimated using ML estimation so as to tune them discriminatively. In our end-to-end setup, we can simply skip denominator computation, and only use the numerator occupation probabilities as the gradients which gives us ML estimation (Equation 6). Note that this is not cross-entropy.

Therefore, we investigate whether enabling MMI after a few iterations versus using MMI from the beginning makes a difference in the final word error rates. More specifically, in this experiment, we start training the network using the ML objective function and at some point in training we change the objective function to MMI. Figure 2 shows the word error rates for different cases of enabling MMI at the beginning and after a specific percent of the training epochs. We can see that, the performance is almost the same when

[2] For example, on Switchboard, which has 46 phonemes (including silence), we will have a total of 46*46*2 = 4232 HMM states (which are not tied) when using a 2-state phone model.

	Phoneme			Character			
	1state	2state	3state	CT	1state	2state	3state
Switchboard	13.0	12.0	12.0	18.1	17.8	15.9	15.8
WSJ	3.2	3.2	3.4	5.7	5.6	5.5	5.7

Table 1. Effect of using different HMM topologies in end-to-end LF-MMI. 1state means 1-state HMM topology and so on (as in Figure 1). CT means CTC's equivalent HMM topology (Figure 1a.)

we enable MMI sometime in the first half of training. Another point is that, the degradation (when we enable MMI at 90% or later) is much more for Switchboard vs. WSJ and for character-based vs. phoneme-based models. In the rest of the experiments presented in this paper, we use MMI from the beginning to the end.

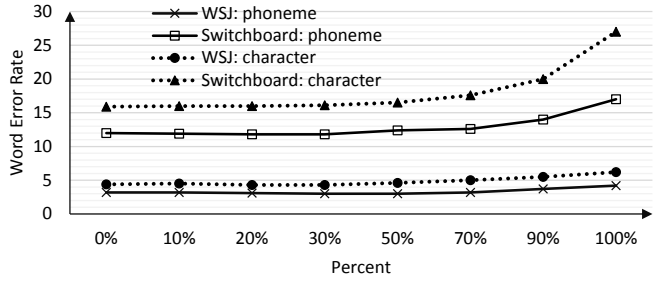


Fig. 2. ML vs MMI. This plot shows the final word error rate when we do a certain percent of the epochs in the beginning without MMI. 0% means we do MMI during all the epochs. 100% means we never do MMI. Total number of epochs in all cases is the same (i.e. 4).

6.2. Phone/Character HMM Topology

One of the advantages of using HMM is that we can potentially improve the alignment learning process by designing the HMM topology for the phones (or characters). We compare three topologies as shown in Figure 1{b,c,d} in Table 1, both in character-based and phoneme-based setup. For the character-based setup, we also test with CTC's equivalent HMM topology. It can be seen that CTC's topology performs similar to a 1-state HMM. Also a 2-state model performs remarkably better than a single state model but a 3-state model does not significantly outperform the 2-state model. For the rest of the experiments in this paper, we use the 2-state HMM topology.

6.3. Tree-free full biphone modeling

The first two rows of Table 2 compare monophone end-to-end LF-MMI results with regular LF-MMI results all using TDNNs. We can see there is a large gap between regular and end-to-end LF-MMI in all cases except in phoneme-based WSJ which is fairly easier than other tasks. The third row of Table 2 shows the impact of full CD (i.e. context-dependent) modeling using biphones/bichars as explained in Section 4.1, which has helped significantly. In particular, for Switchboard it has improved the WER by 1.1% in phoneme-based and 3.1% in character-based setups. This means that in phoneme-based setup, end-to-end LF-MMI is only 0.6% worse than regular LF-MMI on the challenging Switchboard task and almost the same on WSJ. For WSJ, there is no improvement in phoneme-based setup but the WER has been improved more than 1% in character-based setup. For comparison, we also show the result of using regular LF-MMI's tree (which is a pruned context-dependency tree built using HMM-GMM alignments) in our approach. Note that this is not end-to-end any more. We can see that our simple full CD technique performs almost as well as common tree-based CD modeling.

	Switchboard		WSJ	
	Phone	Char	Phone	Char
Regular LF-MMI	10.3	11.9	3.0	3.8
EE-LF-MMI	12.0	15.9	3.2	5.5
EE-LF-MMI (Full CD)	10.9	12.8	3.4	4.4
EE-LF-MMI (Regular CD)*	10.6	12.4	3.2	4.0

* this uses regular LF-MMI's context-dependency tree

Table 2. Effect of full tree-free biphone/bichar modeling (Full CD) in end-to-end LF-MMI (EE-LF-MMI).

	Switchboard		WSJ	
	swbd	eval2000	dev93	eval92
Sequence-to-sequence	-	-	-	9.3 [6]
CTC+BLSTM	20.0 [3]	25.9 [3]	-	7.3 [4]
EE-LF-MMI	15.9	21.8	10.0	5.5
EE-LF-MMI (Full CD)	12.8	18.4	7.5	4.4
EE-LF-MMI (Full CD) + LSTM	10.8	15.8	7.8	4.6

Table 3. Comparison of the WERs achieved with our end-to-end LF-MMI approach (EE-LF-MMI), with the best reported results of other end-to-end methods. The last row uses TDNN-LSTMs [13].

6.4. Comparison to other lexicon-free methods

Table 3 compares our method with CTC and sequence-to-sequence models as the current successful end-to-end approaches, in lexicon-free conditions. In this Table, we report the results on the full Hub5'00 set too. As we can see, our approach outperforms other end-to-end methods significantly. This Table also shows the impact of using LSTM in our setup (for acoustic modeling, not for LM). All our experiments were done using feed-forward TDNN networks except the last row of Table 3 which shows that when we use recurrence in our network, the results can be further improved significantly on the harder Switchboard task.

7. CONCLUSIONS AND FUTURE WORK

In this study, we introduced a simple HMM-based end-to-end method for ASR. In other words, this method is all-neural, GMM-free, tree-free, and is trained in a flat-start manner in a single stage without requiring any initial alignments, pre-training, prior estimation, or transition estimation. We train this neural network using lattice free MMI. Through experiments, we showed that our end-to-end method outperforms other end-to-end methods significantly in equal phoneme-based and character-based (i.e lexicon free) conditions. By training our model on the 300 hour Switchboard database, we achieved a WER of 10.8 on the Switchboard portion of Hub5 '00 test set in lexicon-free end-to-end conditions. To the best of our knowledge, this is the best end-to-end WER reported for Switchboard. We also showed that by using a full biphone modeling technique, our approach can perform almost as well as regular LF-MMI (only 0.6% worse). A future work that is desirable is to train our model on raw speech signals (as in [26]) so that no MFCC feature extraction is needed.

8. REFERENCES

[1] Alex Graves, Abdel-rahman Mohamed, and Geoffrey 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.

[2] Alex Graves and Navdeep Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014, pp. 1764–1772.

[3] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al., "Deep speech: Scaling up end-to-end speech recognition," *arXiv preprint arXiv:1412.5567*, 2014.

[4] Yajie Miao, Mohammad Gowayyed, and Florian Metze, "Eesen: End-to-end speech recognition using deep rnn models and wfst-based decoding," in *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*. IEEE, 2015, pp. 167–174.

[5] Andrew Maas, Ziang Xie, Dan Jurafsky, and Andrew Ng, "Lexicon-free conversational speech recognition with neural networks," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2015, pp. 345–354.

[6] Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio, "End-to-end attention-based large vocabulary speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 4945–4949.

[7] Geoffrey , Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[8] Jan Chorowski, Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, "End-to-end continuous speech recognition using attention-based recurrent nn: first results," *arXiv preprint arXiv:1412.1602*, 2014.

[9] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 369–376.

[10] Albert Zeyer, Eugen Beck, Ralf Schlüter, and Hermann Ney, "Ctc in the context of generalized full-sum hmm training," *Proc. Interspeech 2017*, pp. 944–948, 2017.

[11] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[12] Daniel Povey, Vijayaditya Peddinti, Daniel Galvez, Pegah Ghahrmani, Vimal Manohar, Xingyu Na, Yiming Wang, and Sanjeev Khudanpur, "Purely sequence-trained neural networks for asr based on lattice-free mmi," in *Interspeech*, 2016.

[13] Vijayaditya Peddinti, Yiming Wang, Daniel Povey, and Sanjeev Khudanpur, "Low latency acoustic modeling using temporal convolution and lstms," *IEEE Signal Processing Letters*, 2017.

[14] Kyu J Han, Seongjun Hahm, Byung-Hak Kim, Jungsuk Kim, and Ian Lane, "Deep learning-based telephony speech recognition in the wild," *Training*, vol. 22, no. 46M, pp. 93M, 2017.

[15] Wayne Xiong, Jasha Droppo, Xuedong Huang, Frank Seide, Mike Seltzer, Andreas Stolcke, Dong Yu, and Geoffrey Zweig, "The microsoft 2016 conversational speech recognition system," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 5255–5259.

[16] A.B. Smith, C.D. Jones, and E.F. Roberts, "Article title," *Journal*, vol. 62, pp. 291–294, January 1920.

[17] Andrew Senior, Georg Heigold, Michiel Bacchiani, and Hank Liao, "Gmm-free dnn acoustic model training," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 5602–5606.

[18] Chao Zhang and Philip C Woodland, "Standalone training of context-dependent deep neural network acoustic models," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 5597–5601.

[19] Stanley F Chen, Brian Kingsbury, Lidia Mangu, Daniel Povey, George Saon, Hagen Soltau, and Geoffrey Zweig, "Advances in speech transcription at ibm under the darpa ears program," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1596–1608, 2006.

[20] Steve J Young, Julian J Odell, and Philip C Woodland, "Tree-based state tying for high accuracy acoustic modelling," in *Proceedings of the workshop on Human Language Technology*. Association for Computational Linguistics, 1994, pp. 307–312.

[21] John J Godfrey, Edward C Holliman, and Jane McDaniel, "Switchboard: Telephone speech corpus for research and development," in *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on*. IEEE, 1992, vol. 1, pp. 517–520.

[22] Douglas B Paul and Janet 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.

[23] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr 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.

[24] Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[25] Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur, "Audio augmentation for speech recognition.," in *INTERSPEECH*, 2015, pp. 3586–3589.

[26] Pegah Ghahremani, Vimal Manohar, Daniel Povey, and Sanjeev Khudanpur, "Acoustic modelling from the signal domain using cnns.," in *INTERSPEECH*, 2016, pp. 3434–3438.