End-to-end speech recognition using lattice-free MMI

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

[1]Department of Computer Engineering, Sharif University of Technology, Tehran, 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, USA.

hhadian@jhu.edu, sameti@sharif.edu, dpovey@gmail.com, khudanpur@jhu.edu

Abstract

We present our work on end-to-end training of acoustic models using the lattice-free maximum mutual information (LF-MMI) objective function in the context of hidden Markov models. By end-to-end training, we mean flat-start training of a single DNN in one stage without using any previously trained models, forced alignments, or building state-tying decision trees. We use full biphones to enable context-dependent modeling without trees, and show that our end-to-end LF-MMI approach can achieve comparable results to regular LF-MMI on well-known large vocabulary tasks. We also compare with other end-to-end methods such as CTC in character-based and lexicon-free settings and show 5 to 25 percent relative reduction in word error rates on different large vocabulary tasks while using significantly smaller models.

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]. For simplicity, it is desirable to avoid using 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 DNN using the alignments.

Currently, three popular end-to-end approaches are Connectionist Temporal Classification (CTC), RNN-Transducers and attention-based methods [8]. CTC introduces a sequence-level objective function to enable training a neural network on sequences of speech signals without using prior alignments [9] and RNN-Transducer is an extension of CTC with two separate RNNs [10]. CTC was a pioneering approach in end-to-end speech recognition and state-of-the-art results were achieved on the challenging Fisher+Switchboard task [11] when it was used with deep recurrent neural networks.

By contrast, attention-based models use a novel 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 [12] but, unless the training data is very large, they have not been as effective for speech recognition tasks [6].

Currently the lattice-free MMI (i.e. LF-MMI) method [13] achieves state-of-the-art results on many speech recognition tasks [13, 14, 15, 16]. 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 [17].

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). Two prior studies [18, 19] performed GMM-free training, but used state-tying decision trees (created using alignments from the DNN model) for context dependent (CD) modeling. However, we do not use state-tying trees, and we perform the entire training process in one stage (i.e. without generating re-alignments, building trees, or performing prior estimation). Another difference is that we use the LF-MMI objective function instead of maximum likelihood (ML) for training the network. In our recently submitted journal paper [20], flat-start LF-MMI was investigated in a phoneme-based setting. In this study, we explore character-based training and lexicon-free decoding and show that the end-to-end LF-MMI setup outperforms other end-to-end approaches under similar conditions.

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

2. Regular LF-MMI

The hidden Markov model (HMM) is a generative model commonly used for speech recognition. It is usually used jointly with a Gaussian mixture model (GMM), or a DNN to model acoustic data. A common approach for learning the HMM parameters is through maximum likelihood (ML) estimation which has the following objective function:

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

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

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

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 (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:

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

The denominator can be approximated as:

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

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 [21][22]. That is because a full denominator graph can make the computations slow. Using a full denominator graph has been investigated in [23] with HMM-GMM models. More recently Povey et. al [13] used MMI training with HMM-DNN models using a full denominator graph by adopting a few different techniques such as using a phone language model (LM) (instead of a word LM) for the denominator graph and most importantly performing the denominator computation on GPU hardware. The phone LM for the denominator graph was a pruned n-gram LM trained using the phone alignments of the training data. Also, the composite HMM was not used as the numerator graph and instead a special acyclic graph was used which could exploit the alignment information from a previous HMM-GMM model. More specifically, the numerator graph in the regular 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 (i.e. it has no self-loops). The phone model used with regular LF-MMI is a 2-state HMM as shown in Figure 1c.

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]:

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

where $p(\pi_t|\boldsymbol{x}^{(u)})$ is the network output for label sequence 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 the 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 [24]. 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 was the first successful method used for end-to-end speech recognition, we will use its HMM topology in our setup to compare with the other HMM topologies shown in Figure 1.

4. End-to-end LF-MMI

In regular LF-MMI, the DNN outputs correspond to tied biphone or triphone HMM states, where the tying is done according to a context-dependency tree. This tree is in turn created using alignments from an HMM-GMM system [25]. We remove this prerequisite by using monophones or full biphones (cf. Section 4.1). Moreover, 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. Since we do not have alignments for the training data, we estimate the phone language model for the denominator graph using the training transcriptions (choosing a random pronunciation for words with alternative pronunciations in the phoneme-based setting), 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:

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

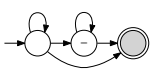
where $y_t^{(u)}(s)$ is the network output for state s at time t given input utterance u which we interpret as the logarithm of an HMM state likelihood (i.e. $\log p(x_t|s)$) since state priors have no effect in MMI training [13]. ${}^{NUM}\gamma_t^{(u)}(s)$ is the numerator HMM occupation probability for state s at time t for utterance u, and ${}^{DEN}\gamma_t^{(u)}(s)$ is defined similarly for the denominator graph.

The HMM transition probabilities are fixed in our setup. Training these shouldn't make a difference as long as there is no state-tying because their effect can be fully replicated by the neural network output (i.e. the transition probabilities act like a scale for the state likelihoods). In other words, the network will ignore them.

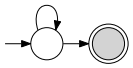
4.1. Tree-free context-dependent modeling

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. To enable context-dependent modeling in an end-to-end manner, we adopt a simple approach where we use full left biphones (or *bichars* in the character-based case). This is implemented as 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 may be considered end-to-end

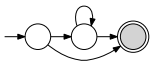
(in the sense of not requiring any previously trained models). In other words, we assume a separate HMM model for each and every possible pair of phonemes (or characters in character-based conditions). [1] This will create biphones that never occur in the training data, but they are never activated during training and the network learns to ignore them.



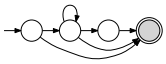
(a) CTC's HMM topology



(b) 1-state HMM topology



(c) 2-state HMM topology



(d) 3-state HMM topology

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

5. Experiments

5.1. Experimental setup

We do most of our experiments on two ASR corpora: Switchboard [26], and WSJ (Wall Street Journal) [27]. Switchboard is a database with 300 hours of transcribed speech. We evaluate on the Hub5 '00 set (also known as *eval2000*). We report word error rates (WER) on the "switchboard" portion of eval2000 (indicated as *SW*) but where stated, we also report WER on the Callhome subset (indicated as *CH*). Where stated, we use the Fisher data for acoustic modeling as well (together with Switchboard, a total of 2000 hours). For decoding we use the Fisher+Switchboard training transcriptions to train a 4-gram word LM (in lexicon-based settings) or a 9-gram character LM (in lexicon-free settings). WSJ is a database with 80 hours of transcribed speech. We test on the "eval92" subset. For lexicon-based decoding, we use a 3-gram LM trained on the WSJ training set transcriptions using an extended lexicon as in [4].

For running the experiments, we use Kaldi [28][2]. We do not use i-vectors or other speaker adaptation techniques in any of the experiments. In all the experiments we use a TDNN-LSTM structure [14], which has interleaving LSTM [29] and TDNN layers [30]; please refer to [14] for more details. As in [13], we use a frame subsampling factor of 3 which speeds up training by a factor of 2. We also augment the data with 2-fold *speed perturbation* in all the experiments [31] unless otherwise stated.

In all the end-to-end experiments, we use SGD 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; no other feature normalization or feature transform is used. The network parameters are initialized randomly to have zero mean and a small variance. Unlike other work, we do not perform re-alignments during training.

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 one of around 30 distinct lengths. When using speed perturbation, we modify the length of each utterance to the nearest of the distinct lengths. Otherwise, we can pad each utterance with silence to reach one of the distinct lengths.

[1] For example, on WSJ, which has 42 phonemes (including silence), we will have a total of 43*42*2 = 3612 HMM states (which are not tied) when using a 2-state HMM topology.

[2] This toolkit is open-source and the source code related to this study are available online for reproducing the results.

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). These results are without CD modeling.*

	Phoneme			Character			
	1state	2state	3state	CT	1state	2state	3state
Switchboard	11.7	**10.7**	**10.7**	14.5	14.2	**13.3**	**13.2**
WSJ	3.1	**3.1**	3.3	5.4	5.3	**5.2**	5.4

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

	Switchboard		WSJ	
	Phone	Char	Phone	Char
Regular LF-MMI	9.1	10.4	2.8	3.5
EE-LF-MMI (monophone)	10.7	13.3	3.1	5.2
EE-LF-MMI (full biphone)	9.6	10.9	3.0	4.1
EE-LF-MMI (regular biphone)*	9.3	10.5	2.9	3.7

* This uses regular LF-MMI's context-dependency tree.

5.2. 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 setups. For the character-based setup, we also test with CTC's equivalent HMM topology (Figure 1a). 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.

5.3. Tree-free full biphone modeling

The first two rows of Table 2 compare monophone end-to-end LF-MMI results with regular LF-MMI (using regular pruned biphone tree[3]) results. 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 2.4% in character-based setups. This means that in a phoneme-based setup, end-to-end LF-MMI is only 0.5% worse than regular LF-MMI on the 300hr Switchboard task and almost the same on WSJ. For WSJ, there is no improvement in the phoneme-based setup but the WER has been improved more than 1% in the 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

[3] Note that with regular LF-MMI, conventional biphone and triphone trees lead to similar WERs (not shown).

Table 3: *Comparison of WER for character-based end-to-end LF-MMI (EE-LF-MMI) and CTC on the 300hr Switchboard.*

Method	Parameters	Lexicon	LM	SW	CH
CTC [32]	50M	N	Char NG	19.8	32.1
EE-LF-MMI	26M	N	Char NG	14.4	25.2
EE-LF-MMI	26M	N	Char RNN	**13.0**	**23.6**
CTC [32]	50M	Y	Word NG	15.1	26.3
EE-LF-MMI	26M	Y	Word NG	**10.9**	**20.6**
CTC [32]	50M	Y	Word RNN	14.0	25.3
EE-LF-MMI	26M	Y	Word RNN	**9.3**	**18.9**
EE-LF-MMI no-SP	26M	Y	Word RNN	10.2	20.0

Table 4: *Comparison of WER for character-based end-to-end LF-MMI (EE-LF-MMI) and related methods on the 2000hr Fisher+Switchboard task. The last two rows show the phoneme-based results. no-SP means no speed perturbation. Tot means on all of eval2000.*

Method	Params	Lex.	LM	SW	CH	Tot†
CTC [32]	50M	N	Char NG	13.8	21.8	17.8
Attention* [33]	100M	N	N	8.6	17.8	13.2
RNN-T* [33]	120M	N	N	**8.5**	**16.4**	**12.5**
EE-LF-MMI	26M	N	Char NG	12.1	21.7	16.9
EE-LF-MMI	26M	N	Char RNN	12.0	21.9	17.0
CTC [32]	50M	Y	Word NG	11.3	18.7	15.0
RNN-T* [33]	120M	Y	Word NG	8.1	17.5	12.8
EE-LF-MMI	26M	Y	Word NG	9.3	18.6	14.0
EE-LF-MMI no-SP	26M	Y	Word NG	9.7	19.0	14.4
CTC [32]	50M	Y	Word RNN	10.2	17.7	14.0
EE-LF-MMI	26M	Y	Word RNN	8.0	17.6	12.8
Phone CTC [34]	–	Y	Word NG	10.2	16.5	13.3
Phone EE-LF-MMI	26M	Y	Word NG	8.6	15.5	12.0
Phone EE-LF-MMI	26M	Y	Word RNN	**7.5**	**14.6**	**11.0**

* These use data augmentation by adding background noise.
† The total eval2000 WER for CTC and Attention is the average of SW and CH (as it is not reported).

that our simple full CD technique performs almost as well as common tree-based CD modeling.

5.4. Comparison to other end-to-end approaches

In this section we compare end-to-end LF-MMI with other end-to-end methods. Tables 3 and 4 show the results on the 300hr Switchboard and 2000hr Fisher+Switchboard tasks respectively, and Table 5 shows the results on WSJ. The characters we use in character-based modeling are the digits, the letters, apostrophe, and space. We report WER for both lexicon-based and lexicon-free decoding. In lexicon-free decoding we decode characters and separate the words by the decoded space characters. The language models we use for lexicon-free decoding are character n-grams (Char NG) and character RNN-LMs. We use a 9-gram character LM trained on the training transcriptions.

On the larger 2000hr Fisher+Switchboard, end-to-end LF-MMI has achieved around 1% improvement (on all of eval2000) over CTC in the lexicon-free decoding case but the best results are for RNN-Transducer. When using lexicon-based decoding, character-based end-to-end LF-MMI and RNN-Transducer achieve the same result (12.8) outperforming CTC (14.0). However, the best overall results are for the phoneme-based end-to-end LF-MMI achieving a total WER of 11.0 on eval2000 (7.5 on the Switchboard subset). Note that the models used in end-to-end LF-MMI are considerably smaller. Also note that in training the attention-based and RNN-Transducer models (which are substantially larger) [33], data augmentation with background noise has been applied. For comparison, we have included results without speed perturbation (no-SP) too. Meanwhile, we see significant improvements on the smaller 300hr Switchboard and 80hr WSJ task, in both lexicon-free and lexicon-based decoding. Specifically, we see 4 to 5 percent absolute improvement in WER on the 300hr Switchboard task, and 1.4 percent improvement on WSJ in similar conditions (i.e. no-SP).

Table 5: *Comparison of WER for character-based end-to-end LF-MMI (EE-LF-MMI) and related methods on WSJ.*

Method	Parameters	Lexicon	LM	WER
Phone CTC [4]	–	Y	Word NG	7.3
Attention [35]	6.6M	Y	Word NG	6.7
EE-LF-MMI	8.2M	Y	Word NG	**4.1**
EE-LF-MMI no-SP	8.2M	Y	Word NG	5.3
EE-LF-MMI	8.2M	N	Char NG	5.4

5.5. Training and decoding speed

Even though the LF-MMI objective function requires a denominator computation which is nontrivial, the training is quite fast because we can use considerably smaller models (compared to other end-to-end models or CD-HMM-DNN models which use cross-entropy) while achieving better results. For example, the training speed of end-to-end LF-MMI on the 2000hr Fisher+Switchboard task is approximately 2.1 hours of speech per minute on a *GeForce GTX 1080 Ti* GPU. The overall data preparation and feature extraction takes about 20 hours on a 32-core machine and the overall network training lasts about 3 days on a machine with 8 GTX 1080 Ti GPUs. The decoding real-time factor is 0.9 on Switchboard and 0.4 on WSJ (on CPU).

6. Conclusions

In this study, we described a simple HMM-based end-to-end method for ASR and evaluated it on well-known large vocabulary speech recognition tasks. This acoustic model is all-neural, GMM-free, tree-free, and is trained in a flat-start manner in one stage (using lattice-free MMI) without requiring any initial alignments, pre-training, prior estimation, or transition training. Through experiments, we showed that our end-to-end method outperforms other end-to-end methods in similar settings, especially on smaller databases such as the 300hr Switchboard or 80hr WSJ tasks where the relative improvements in WER range from 15 to 25 percent. By training our end-to-end model on the 2000hr Fisher+Switchboard database, we achieved a WER of 12.8 on all of eval2000 (8.0 on the Switchboard subset) in the character-based case, and a WER of 11.0 on all of eval2000 (7.5 on the Switchboard subset) in the phoneme-based setting. We also showed that by using a full biphone modeling technique, our approach can perform almost as well as regular LF-MMI (only 0.5% worse).

7. Acknowledgements

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

8. References

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

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

[3] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates *et al.*, "Deep speech: Scaling up end-to-end speech recognition," *arXiv preprint arXiv:1412.5567*, 2014.

[4] Y. Miao, M. Gowayyed, and F. Metze, "EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding," in *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2015, pp. 167–174.

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

[6] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4945–4949.

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

[8] J. Chorowski, D. Bahdanau, K. Cho, and Y. Bengio, "End-to-end continuous speech recognition using attention-based recurrent NN: first results," *arXiv preprint arXiv:1412.1602*, 2014.

[9] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of International Conference on Machine Learning*. ACM, 2006, pp. 369–376.

[10] A. Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.

[11] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates *et al.*, "Deep speech: Scaling up end-to-end speech recognition," *arXiv preprint arXiv:1412.5567*, 2014.

[12] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[13] D. Povey, V. Peddinti, D. Galvez, P. Ghahrmani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely sequence-trained neural networks for ASR based on lattice-free MMI," in *Proceedings of INTERSPEECH*, 2016.

[14] V. Peddinti, Y. Wang, D. Povey, and S. Khudanpur, "Low latency acoustic modeling using temporal convolution and LSTMs," *IEEE Signal Processing Letters*, 2017.

[15] K. J. Han, S. Hahm, B.-H. Kim, J. Kim, and I. Lane, "Deep learning-based telephony speech recognition in the wild," in *Proceedings of INTERSPEECH*, 2017, pp. 1323–1327.

[16] W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu, and G. Zweig, "The Microsoft 2016 conversational speech recognition system," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 5255–5259.

[17] L. Bahl, "Maximum mutual information estimation of hidden Markov model parameters for speech recognition," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1986, pp. 701–704.

[18] A. Senior, G. Heigold, M. Bacchiani, and H. Liao, "GMM-free DNN acoustic model training," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 5602–5606.

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

[20] H. Hadian, H. Sameti, D. Povey, and S. Khudanpur, "Flat-start single-stage discriminatively trained HMM-based models for ASR," *Submitted to IEEE Transactions on Audio, Speech, and Language Processing*, 2018.

[21] V. Valtchev, J. Odell, P. C. Woodland, and S. J. Young, "Lattice-based discriminative training for large vocabulary speech recognition," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 2. IEEE, 1996, pp. 605–608.

[22] P. C. Woodland and D. Povey, "Large scale discriminative training of hidden Markov models for speech recognition," *Computer Speech & Language*, vol. 16, no. 1, pp. 25–47, 2002.

[23] S. F. Chen, B. Kingsbury, L. Mangu, D. Povey, G. Saon, H. Soltau, and G. 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.

[24] A. Zeyer, E. Beck, R. Schlüter, and H. Ney, "CTC in the context of generalized full-sum HMM training," in *Proceedings of INTERSPEECH*, 2017, pp. 944–948.

[25] S. J. Young, J. J. Odell, and P. 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.

[26] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "SWITCHBOARD: Telephone speech corpus for research and development," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1. IEEE, 1992, pp. 517–520.

[27] 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.

[28] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The Kaldi speech recognition toolkit," in *Proceedings of IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, no. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.

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

[30] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, "Phoneme recognition using time-delay neural networks," in *Readings in Speech Recognition*. Elsevier, 1990, pp. 393–404.

[31] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Proceedings of INTERSPEECH*, 2015, pp. 3586–3589.

[32] G. Zweig, C. Yu, J. Droppo, and A. Stolcke, "Advances in all-neural speech recognition," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, 2017, pp. 4805–4809.

[33] E. Battenberg, J. Chen, R. Child, A. Coates, Y. Gaur, Y. Li, H. Liu, S. Satheesh, D. Seetapun, A. Sriram *et al.*, "Exploring neural transducers for end-to-end speech recognition," *arXiv preprint arXiv:1707.07413*, 2017.

[34] K. Audhkhasi, B. Ramabhadran, G. Saon, M. Picheny, and D. Nahamoo, "Direct acoustics-to-word models for english conversational speech recognition," in *Proceedings of INTERSPEECH*, 2017, pp. 959–963.

[35] J. Chorowski and N. Jaitly, "Towards better decoding and language model integration in sequence to sequence models," in *Proceedings of INTERSPEECH*, 2017.