Interspeech 2018
2-6 September 2018, Hyderabad

Lattice-free State-level Minimum Bayes Risk Training of Acoustic Models

Naoyuki Kanda, Yusuke Fujita, Kenji Nagamatsu

Hitachi Ltd., Japan

{naoyuki.kanda.kn,yusuke.fujita.su,kenji.nagamatsu.dm}@hitachi.com

Abstract

Lattice-free maximum mutual information (LF-MMI) training, which enables MMI-based acoustic model training without any lattice generation procedure, has recently been proposed. Although LF-MMI showed high accuracy in many tasks, its MMI criterion does not necessarily maximize the speech recognition accuracy. In this work, we propose a lattice-free state-level minimum Bayes risk training (LF-sMBR), which maximizes state-level expected accuracy without relying on a lattice generation procedure. As is the case with the LF-MMI, LF-sMBR avoids redundant lattice generation by exploiting forward-backward calculation on phone N-gram space, which enables a much simpler and faster training based on an sMBR criterion. We found that special care for silence phones was essential for improving the accuracy by LF-sMBR. In our experiments on the AMI, CSJ, and Librispeech corpora, LF-sMBR achieved small but consistent improvements over LF-MMI AMs, showing state-of-the-art results for each test set.

Index Terms: lattice-free maximum mutual information, speech recognition, acoustic model, sequence training

1. Introduction

Recent advances in automatic speech recognition (ASR) are largely owed to the progress made with deep neural network (DNN)-based acoustic models (AMs). In an earlier version of DNN-based ASR, AMs were trained using a frame-level cross-entropy (CE) loss criterion [1, 2, 3], which provided significant accuracy improvements compared to the Gaussian mixture model (GMM)-based AMs [3, 4, 5]. After the success of CE-based AM training, sequence-level training criteria such as maximum mutual information (MMI), state-level minimum Bayes risk (sMBR), and boosted MMI (bMMI) were introduced, showing much better accuracy than CE-AMs [6, 7]. Because MMI, sMBR, and bMMI all require error calculation over the entire hypothesis space, it is conventional to constrain the error calculation space with lattices generated by CE-AMs. This CE-AM-based lattice generation has two implicit problems. First, such a training procedure could easily fall into local optimum nearby CE-AMs, resulting in semi-optimum accuracy, and second, it requires a very high computational cost, which we want to avoid especially when the data size is quite large.

Recently, lattice-free maximum mutual information (LF-MMI) training of acoustic models has been proposed [8]. LF-MMI achieved MMI-based neural network training from scratch, i.e., without relying on the CE-based lattice generation procedure. Instead of lattice-based error calculation, LF-MMI uses forward-backward calculation on phone N-gram space. The key advantage here is that forward-backward calculation can be implemented in a highly parallelized way by using GPGPU techniques, which lower the computational cost of forward-backward calculation over the entire training procedure. LF-MMI-training also showed much better accuracy than CE- and sMBR-based training in many tasks [8, 9, 10].

One disadvantage of LF-MMI is that its MMI criterion does not necessarily maximize the speech recognition accuracy. Because of this property, sMBR training, which maximizes state-level expected recognition accuracy, often achieved the best results if applied after LF-MMI training. In previous literature [8, 11, 12], sMBR training after LF-MMI training showed a relative improvement of about 2%-6% [1] over original LF-MMI AMs. Importantly, these investigations into sMBR training were conducted using conventional lattice-based error calculation. Considering the advantages of LF-MMI training, it would be better if sMBR training could be realized in a lattice-free manner.

In this work, we propose lattice-free sMBR (LF-sMBR) as an extension of LF-MMI. The same as LF-MMI, LF-sMBR circumvents redundant lattice generation by using forward-backward calculation on phone N-gram space. The main part of forward-backward calculation is realized using GPGPU techniques, which enables a much simpler and faster training based on the sMBR criterion. We found that the special treatment of silence phones was essential for improving the accuracy by LF-sMBR. Experiments on the AMI meeting corpus [13], the corpus of spontaneous Japanese (CSJ) [14], and LibriSpeech [15] showed $\sim$2% relative improvements over LF-MMI AMs by LF-sMBR. Improvements were small but consistent across test sets, and our AM achieved state-of-the-art results for each test set. Our training procedure is all lattice-free while best results were obtained by LF-sMBR training starting from LF-MMI AMs.

We should point out that exact calculation of "expected word accuracy" is still difficult (though not impossible [16]), and is beyond the focus of this paper. Instead, we use "state-level" expected accuracy, which can be estimated on phone N-gram space, and focused on the improvement over naive LF-MMI. In the recent work most similar to ours [17], the authors proposed a lattice-free version of boosted MMI and achieved about 2 % relative improvement over LF-MMI. Our work achieved similar improvements over LF-MMI but differs in that it is based on an sMBR criterion. Comparison with other criteria remains our future work.

2. LF-MMI-based acoustic modeling

The training criterion for LF-MMI is defined as [2]

$$\mathcal{F}^{LFMMI} = \sum_u \sum_{\mathbf{S}} P(\mathbf{S}|\mathcal{G}_u^N, \mathbf{X}_u) \log P(\mathbf{S}|\mathcal{G}^D, \mathbf{X}_u), \quad (1)$$

where u is the index of training utterances. The term $\mathbf{X}_u$ indicates acoustic features for utterance u, and $\mathbf{S}$ indicates a hypoth-

[1] Note that these numbers are relative improvements from highly accurate LF-MMI AMs. LF-MMI itself achieved much better results than CE AMs and CE+sMBR trained AMs [8, 10].

[2] This is a numerator-graph ($\mathcal{G}_u^N$)-based extension of basic MMI-criterion $\mathcal{F}^{MMI} = \sum_u \log P(\mathbf{S}_u|\mathbf{X}_u)$ [6, 7], which uses a Viterbi-aligned reference state sequence $\mathbf{S}_u$ instead of $\mathcal{G}_u^N$.

10.21437/Interspeech.2018-79

esis state sequence for $\mathbf{X}_u$. The term $\mathcal{G}_u^N$ indicates a numerator (or reference) graph that represents a set of possible correct state sequences for utterance u. The term $\mathcal{G}^D$ represents a denominator graph, which represents a possible hypothesis space. The error signal w.r.t. the final layer's output $y(u,t)$ corresponding to a state $s(u,t)$ of the AM at the time frame t of utterance u is calculated as

$$\frac{\partial \mathcal{F}^{LFMMI}}{\partial y(u,t)} = \gamma^N_{s(u,t)} - \gamma^D_{s(u,t)}. \quad (2)$$

The term $\gamma^*_{s(u,t)}$ (* is N or D) represents a posterior probability of being in a state $s(u,t)$ calculated on numerator graph $\mathcal{G}_u^N$ or denominator graph $\mathcal{G}^D$.

$$\gamma^*_{s(u,t)} = P(s(u,t)|\mathcal{G}^*, \mathbf{X}_u) = \sum_{\mathbf{S}\in\mathcal{G}^*} \delta_{\mathbf{S}:s(u,t)} P(\mathbf{S}|\mathcal{G}^*, \mathbf{X}_u). \quad (3)$$

Here, $\delta_{\mathbf{S}:s(u,t)}$ is a delta function, which is 1 if the state $s(u,t)$ corresponding to $y(u,t)$ is in $\mathbf{S}$, and 0 otherwise.

In LF-MMI modeling, the numerator graph is constructed by loosely following GMM-AM-based reference alignments, while the denominator graph is created from the phone 4-gram language model (LM) trained using phone-level transcription of training data. "Lattice-free" means that the denominator computation is done without conducting any lattice generation procedure, which is required when using the conventional MMI training [6, 7]. Importantly, the number of states in phone 4-gram space is fixed during the entire training process, i.e., independent of utterance u. This enables highly parallelized implementation of the forward-backward calculation using GPGPU techniques. As a result, the computation of the forward-backward calculation is no longer dominant over the entire training process, which results in very fast training. In addition, network parameters can be trained from scratch, which means local optimum near CE-AMs can be avoided. LF-MMI exhibited much better results than CE and CE+sMBR AMs in many tasks [8, 10].

3. Lattice-free sMBR

3.1. Overview

The LF-sMBR training criterion is defined similarly to the conventional sMBR, as

$$\mathcal{F}^{LFsMBR} = \sum_u \sum_{\mathbf{S}\in\mathcal{G}^D} P(\mathbf{S}|\mathcal{G}^D, \mathbf{X}_u)\mathcal{A}(\mathbf{S}, \mathcal{G}_u^N), \quad (4)$$

which represents the state-level expected ASR accuracy for training data. Here, $\mathcal{A}(\mathbf{S}, \mathcal{G}_u^N)$ is the state-level accuracy of the hypothesis $\mathbf{S}$ calculated on the numerator graph $\mathcal{G}_u^N$. In conventional sMBR, $\mathcal{A}(\cdot,\cdot)$ is usually calculated by counting the frame-wise coincidence of the Viterbi-aligned reference state and the hypothesis state. In LF-sMBR, we instead use a summation of posterior probabilities estimated by the numerator graph, as

$$\mathcal{A}(\mathbf{S}, \mathcal{G}_u^N) = \sum_t \gamma^N_{s(t)}, \quad (5)$$

where $s(t)$ is the state in sequence $\mathbf{S}$ at time frame t. Then, the error signal is calculated as

$$\frac{\partial \mathcal{F}^{LFsMBR}}{\partial y(u,t)} = \gamma^D_{s(u,t)}\{\bar{\mathcal{A}}_u(t) - \bar{\mathcal{A}}_u\}, \quad (6)$$

where

$$\bar{\mathcal{A}}_u(t) = \frac{\sum_{\mathbf{S}\in\mathcal{G}^D} \delta_{\mathbf{S}:s(u,t)} P(\mathbf{S}|\mathcal{G}^D, \mathbf{X}_u)\mathcal{A}(\mathbf{S}, \mathcal{G}_u^N)}{\sum_{\mathbf{S}\in\mathcal{G}^D} \delta_{\mathbf{S}:s(u,t)} P(\mathbf{S}|\mathcal{G}^D, \mathbf{X}_u)}, \quad (7)$$

$$\bar{\mathcal{A}}_u = \sum_{\mathbf{S}\in\mathcal{G}^D} P(\mathbf{S}|\mathcal{G}^D, \mathbf{X}_u)\mathcal{A}(\mathbf{S}, \mathcal{G}_u^N). \quad (8)$$

The same as the case of LF-MMI, LF-sMBR uses phone 4-gram space for denominator graph $\mathcal{G}^D$. Basic statistics $\gamma^D_{s(u,t)}$ and $\gamma^N_{s(u,t)}$ are first estimated by using the forward-backward procedure, exactly the same as LF-MMI. Then, these statistics are used to compute expected accuracy $\bar{\mathcal{A}}_u(t)$ and $\bar{\mathcal{A}}_u$ by one more forward-backward procedure on phone 4-gram space. Therefore, the only difference between LF-sMBR and LF-MMI is the second forward-backward procedure. The algorithm of the second forward-backward calculation is the same as the conventional sMBR training, so we skip going into details here. Importantly, both the first and second forward-backward calculations can be implemented in a highly parallelized way by using GPGPU techniques because the number of states is independent of utterances. We implemented LF-sMBR by modifying an LF-MMI implementation of the Kaldi toolkit [18]. In our experimental settings (Sec 4.1.1), the second forward-backward procedure for LF-sMBR caused just an 8-9% increase of training time compared to the original LF-MMI.

3.2. Treatment of silence in accuracy calculation

In the conventional sMBR training, it is common practice to evaluate Eq. (5) by only using non-silence frames according to the reference label, which makes the training criterion insensitive to insertion errors. We emulate this measure by resetting all posteriors for silence phones to 0 when calculating Eq. (5). On the other hand, a recent paper [19] proposed "one-silence-class" modification, in which all silence states (vocalized noise, non-spoken noise, etc.) are summarized into one class and counted in Eq. (5). We emulate this measure by replacing the posterior for each silence state into the sum of the posteriors of all silence states.

In the experimental section, we show the results for three cases: (1) counting all silence independently, (2) not counting silence phones, and (3) counting silence as one-silence-class.

3.3. Regularization

Since sMBR training is known to be sensitive to overfitting, we introduce three types of regularization techniques for LF-sMBR. The first and second techniques are $l2$ regularization and CE regularization, both of which were introduced in the original LF-MMI paper [8]. The $l2$ regularization adds a penalty term for the squared $l2$-norm of the network output. CE regularization is a technique to add one more output layer at the top of the neural network, which is updated via CE loss criterion.

The third technique is an MMI-based regularization that uses a combination of LF-sMBR and LF-MMI criteria similar to I-smoothing [20] or F-smoothing [7], as

$$(1-\lambda)\cdot\mathcal{F}^{LFsMBR} + \lambda\cdot\mathcal{F}^{LFMMI}, \quad (9)$$

where λ is a scaling parameter.

Note that the leaky hidden Markov model (leaky-HMM) technique, which adds transitions among all HMM-states, was also proposed for LF-MMI [8] to show its effectiveness as a regularizer. However, this technique is difficult to efficiently implement for LF-sMBR (though it is possible for LF-MMI), so we did not evaluate leaky-HMM for LF-sMBR.

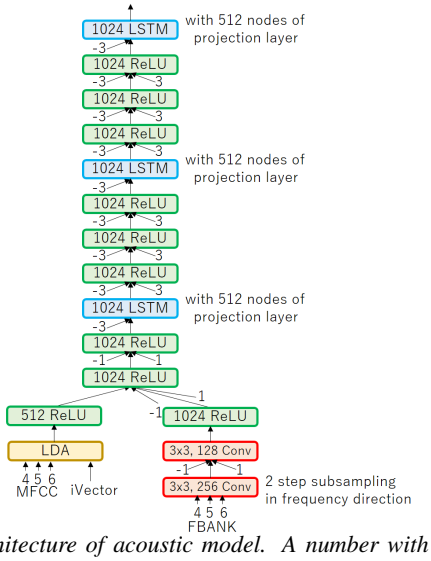

Figure 1: *Architecture of acoustic model. A number with an arrow indicates a time splicing index of TDNN [26].*

4. Experiments

4.1. Evaluation on AMI

4.1.1. Settings

We performed our first experiment on individual headset microphone (IHM) data from the AMI meeting corpus [13]. The training and evaluation datasets were prepared according to the instructions in Kaldi [18]. The training data included 77 hours of meeting speech and was augmented six times using speed perturbation (x3)[21] and noise/reverberation perturbation (x2) [22]. The development and evaluation data totaled 8.9 hours and 8.7 hours, respectively. A 3-gram LM trained by AMI transcription (49K vocabulary) was used for decoding. All parameters were tuned using the development data, and the best settings were used for decoding the evaluation data.

We trained an acoustic model consisting of a convolutional neural network (CNN), time-delay neural network (TDNN) [23], and long short-term memory (LSTM) [24]. The architecture of the AM, called CNN-TDNN-LSTM, is shown in Fig. 1. Input features for the network were 40-dim Mel-frequency cepstral coefficients (MFCCs) and 40-dim log-Mel-filterbank (FBANK), both without normalization. In addition, a 100-dim iVector was extracted every 100 msec and appended to the input features for online speaker/environment adaptation [25]. The input features were advanced by five frames, which has the same effect as reference label delay.

In training, the AM was first trained by LFMMI and then further trained by either conventional lattice-based sMBR or the proposed LF-sMBR. In LFMMI training, the initial learning rate was set to 0.001 and exponentially decayed to 0.0001 by the end of the training. We applied $l2$-regularization and CE-regularization [8] with scales of 0.00005 and 0.1, respectively. Leaky-HMM coefficient was set to 0.1. In addition, a backstitch technique [27] with the backstitch scale 1.0 and backstitch interval 4 was used. In the conventional lattice-based sMBR training, the learning rate was fixed to 0.000000125, and the $l2$-regularization and cross-entropy-regularization were set to 0.00005 and 0.1, respectively. We used "one-silence-class" counting method [19], which showed slightly better result than "silence-uncounting" method in our preliminary experiment. Settings for LF-sMBR are discussed in succeeding sections.

4.1.2. Baseline results with LF-MMI and conventional sMBR

We first evaluated the accuracy of the LF-MMI and the conventional lattice-based sMBR. The word error rates (WERs) with various combinations of training epochs are presented in Table 1. The two main findings are:

- LF-MMI training showed sufficient convergence after four epochs of iteration, and further training caused overfitting.
- Conventional sMBR training clearly boosted the accuracy, yielding 2-3% relative improvement over LF-MMI.

Note that, although sMBR achieved WER improvements, its computational cost for lattice generation was quite high. Because of data augmentation [3] and the large size of the AM, about 2,900 CPU-hours of computation (including I/O) was required to generate lattices for this 77-hours training data. On the other hand, 1-epoch of sMBR training after lattice generation required just about 12 hours of computation with CPU and GPU (NVIDIA®Tesla®P100). This primarily formed our motivation to propose LF-sMBR, which can eliminate the need to generate lattices. Benefit of the lattice-free method becomes much larger when the data size becomes large (such as 1,000-10,000 hours).

Table 1: *Baseline WERs (%) for AMI-IHM with LF-MMI and conventional lattice-based sMBR training.*

Criterion	dev	eval
LF-MMI (1-epoch)	20.71	20.32
LF-MMI (2-epoch)	19.75	19.05
LF-MMI (3-epoch)	**19.08**	18.61
LF-MMI (4-epoch)	19.11	**18.41**
LF-MMI (5-epoch)	19.15	18.81
LF-MMI (6-epoch)	19.13	18.82
LF-MMI (4-epoch) → sMBR (1-epoch)	18.75	17.91
LF-MMI (4-epoch) → sMBR (2-epoch)	**18.72**	**17.84**
LF-MMI (4-epoch) → sMBR (3-epoch)	18.74	17.85

4.1.3. Effect of different treatments of silence phones

Next, we evaluated the proposed LF-sMBR training. We first examined the effect of different treatment of silence phones in expected error calculation (Section 3.2). In this experiment, no regularization was applied. We used the fixed learning rate of 0.000000125 to conduct 1-epoch LF-sMBR training. [4]

Results are presented in Table 2. We found that uncounting the silence phones was essential for LF-sMBR. In this case, LF-sMBR achieved similar WER improvement as the conventional lattice-based sMBR. Unexpectedly, the naive "counting" method and the "one-silence counting" method were not effective for LF-sMBR while the training objective was improved appropriately in all three cases. One possible reason for this phenomenon is that the numerator calculation is done per model update in LF-sMBR, which could cause severe overfitting of silence phones. This hypothesis is partly reviewed in Section 4.1.5.

Table 2: *WERs (%) for AMI-IHM using LF-sMBR with various silence treatments in expected accuracy calculation.*

Criterion	Silence	dev	eval
LF-MMI (4-epoch)	-	19.11	18.41
LF-MMI → LF-sMBR (1-epoch)	count	19.11	18.41
LF-MMI → LF-sMBR (1-epoch)	uncount	**18.82**	**18.10**
LF-MMI → LF-sMBR (1-epoch)	one-silence count	19.13	18.46

[3] In addition to the speed (x3) and reverberation (x2) perturbation, input frame shift variation (x3) was also applied in lattice generation for sMBR in Kaldi, which largely increased the computational cost.

[4] We examined various learning rates in our preliminary experiments and found that the effective range of learning rates was the same as the case of the conventional lattice-based sMBR training.

4.1.4. Effect of regularization

We tried various regularization techniques, the results of which are presented in Table 3. Here, we applied $l2$-regularization with scales of 0.00005 and CE-regularization with scales of 0.1. MMI-based regularization was applied with $\lambda = 0.1$. The results in this table demonstrate that each regularization ($l2$, CE, MMI) had a marginal effect to mitigate overfitting when iterating LF-sMBR. For example, while 2-epoch LF-sMBR without any regularization produced 18.22% WER, $l2$ regularization mitigated the degradation to 18.14%. Although the impact of regularization was much smaller than we expected from LF-MMI experiments [8], the combination of three regularizations produced the best results (18.80% and 18.04% for dev and eval).

Table 3: *WERs (%) for AMI-IHM using LF-sMBR with various regularizations.*

Criterion	Regularization			dev	eval
	$l2$	CE	MMI		
LF-MMI (4-epoch)				19.11	18.41
LF-MMI → LF-sMBR (1-epoch)				18.82	18.10
LF-MMI → LF-sMBR (2-epoch)				18.88	18.22
LF-MMI → LF-sMBR (1-epoch)	√			18.83	18.09
LF-MMI → LF-sMBR (2-epoch)	√			18.83	18.14
LF-MMI → LF-sMBR (1-epoch)		√		18.83	18.06
LF-MMI → LF-sMBR (2-epoch)		√		18.83	18.14
LF-MMI → LF-sMBR (1-epoch)			√	18.88	18.08
LF-MMI → LF-sMBR (2-epoch)			√	18.83	18.16
LF-MMI → LF-sMBR (1-epoch)	√	√	√	18.86	18.06
LF-MMI → LF-sMBR (2-epoch)	√	√	√	**18.80**	**18.04**

4.1.5. Freezing numerator calculation

As pointed out in Section 4.1.3, the per-update-based calculation of numerator posteriors might be what caused the overfitting in LF-sMBR. To investigate this hypothesis, we tested "freezing" the numerator posterior calculation by using original LF-MMI AM for the numerator posterior calculation (i.e., not by using the last LF-sMBR-updated AM). The results (listed in Table 4) demonstrate that small improvements of WER were achieved by freezing the numerator posterior calculation.

Table 4: *WERs (%) for AMI-IHM using LF-sMBR with/without numerator freezing.*

Criterion	Freeze numerator	dev	eval
LF-MMI (4-epoch)		19.11	18.41
LF-MMI → LF-sMBR (1-epoch)		18.86	18.06
LF-MMI → LF-sMBR (2-epoch)		18.80	18.04
LF-MMI → LF-sMBR (3-epoch)		18.83	18.07
LF-MMI → LF-sMBR (1-epoch)	√	18.80	18.07
LF-MMI → LF-sMBR (2-epoch)	√	**18.73**	18.02
LF-MMI → LF-sMBR (3-epoch)	√	18.77	**18.01**

4.1.6. Balancing LF-MMI and LF-sMBR

Finally, we examined whether it is possible to conduct LF-sMBR from an earlier stage of the LF-MMI training, the results of which are presented in Table 5. Here we conducted two epochs of LF-sMBR training after LF-MMI training. Unfortunately, the best performance was obtained only when we started the LF-sMBR training from sufficiently converged LF-MMI AM. Better results could have been obtained if we appropriately set the interpolation weight between LF-MMI and LF-sMBR, similar to a previously proposed interpolation weight scheduling between CE and sMBR [28]. This remains our future work.

Table 5: *Effect of LF-sMBR on earlier stage of LF-MMI.*

# of LF-MMI epochs	before LF-sMBR		after LF-sMBR	
	dev	eval	dev	eval
2	19.75	19.05	19.50	18.76
3	19.08	18.61	18.88	18.25
4	19.11	18.41	**18.73**	**18.02**

4.2. Experiments on CSJ and LibriSpeech

We also conducted evaluations on CSJ [14] and LibriSpeech [15].

CSJ is one of the most widely used evaluation sets for Japanese speech recognition. It consists of about 600 hours of Japanese lecture recordings. We used the three official evaluation sets, E1, E2, and E3 [29], each of which includes different types of groups of ten lectures (5.6 hours of 30 lectures in total). For the development set to tune the decoding parameters, we selected 7.1 hours of 40 lecture recordings. The remainder of the 577 hours of lecture recordings (excluding the same speaker's lectures with evaluation and development sets) was used for AM and LM training. In AM training, training data were augmented by using speed perturbation (x3) [21]. We used 4-gram and recurrent neural network-based LMs for decoding, the details of which are shown in [10].

Librispeech consists of about 1,000 hours of read English speech. Training, development, and evaluation sets were prepared according to the Kaldi scripts. The evaluation set consisted of two groups named "clean" and "other" in accordance with the difficulty of recognizing the speech; their durations were 5.4 hours and 5.3 hours, respectively. Training data were augmented by using speed perturbation (x3) [21]. We used officially provided large 4-gram LM for decoding.

For both CSJ and Librispeech, we used the same AM architecture as in the AMI experiments. We also used the best training parameters (learning rate, regularization, etc.) from the AMI experiments, except for the number of iterations for LF-MMI and LF-sMBR. For CSJ, we conducted four epochs of LF-MMI and one epoch of LF-sMBR. For Librispeech, we conducted two epochs of LF-MMI and 0.5 epochs of LF-sMBR.

Results for CSJ and Librispeech are presented in Tables 6 and 7, respectively. Note that decoding parameters (language model scale, insertion penalty, etc.) for these evaluations were all tuned by a development set. Aside from just one case of "E1 test set of CSJ with RNN-LM rescoring", LF-sMBR showed $\sim$ 2% relative improvements compared to LF-MMI. To the best of our knowledge, the numbers in Tables 6 and 7 are the best results ever reported with these data sets.

Table 6: *WERs (%) for CSJ evaluation set.*

Criterion	E1	E2	E3	avg.
(4gram-LM)				
LF-MMI	8.51	6.94	6.94	7.46
LF-MMI → LF-sMBR	**8.41**	**6.72**	**6.86**	**7.33**
(4gram-LM + RNN-LM rescoring)				
LF-MMI	**7.43**	6.38	6.41	6.74
LF-MMI → LF-sMBR	7.49	**6.22**	**6.16**	**6.62** (*)

(*) Character error rate (CER) was 5.03% (E1: 5.67%, E2: 4.90%, E3: 4.53%)

Table 7: *WERs (%) for LibriSpeech evaluation set.*

Criterion	clean	other
LF-MMI	3.72	8.69
LF-MMI → LF-sMBR	**3.68**	**8.57**

5. Conclusion

We proposed LF-sMBR training that enables sMBR criterion-based training in a lattice-free manner. Instead of the redundant lattice generation procedure, LF-sMBR exploits forward-backward calculation on phone N-gram space, thus enabling a much simpler and faster training than conventional lattice-based sMBR training. In our experiments on the AMI, CSJ, and Librispeech corpora, LF-sMBR achieved small but consistent improvements over LF-MMI AMs, showing state-of-the-art results for each test set.

6. References

[1] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks." in *Proc. INTERSPEECH*, 2011, pp. 437–440.

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

[3] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "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.

[4] N. Kanda, R. Takeda, and Y. Obuchi, "Elastic spectral distortion for low resource speech recognition with deep neural networks," in *Proc. ASRU*, 2013, pp. 309–314.

[5] G. Heigold, V. Vanhoucke, A. Senior, P. Nguyen, M. Ranzato, M. Devin, and J. Dean, "Multilingual acoustic models using distributed deep neural networks," in *Proc. ICASSP*, 2013, pp. 8619–8623.

[6] K. Veselỳ, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks." in *Proc. INTERSPEECH*, 2013, pp. 2345–2349.

[7] H. Su, G. Li, D. Yu, and F. Seide, "Error back propagation for sequence training of context-dependent deep networks for conversational speech transcription," in *Proc. ICASSP*, 2013, pp. 6664–6668.

[8] 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," *Proc. INTERSPEECH*, pp. 2751–2755, 2016.

[9] W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu, and G. Zweig, "Achieving human parity in conversational speech recognition," *arXiv preprint arXiv:1610.05256*, 2016.

[10] N. Kanda, Y. Fujita, and K. Nagamatsu, "Investigation of lattice-free maximum mutual information-based acoustic models with sequence-level kullback-leibler divergence," in *Proc. ASRU*, 2017, pp. 69–76.

[11] V. Manohar, D. Povey, and S. Khudanpur, "JHU Kaldi system for arabic MGB-3 ASR challenge using diarization, audio-transcript alignment and transfer learning," in *Proc. ASRU*, 2017, pp. 346–352.

[12] N. Kanda, Y. Fujita, and K. Nagamatsu, "Sequence distillation for purely sequence trained acoustic models," in *Proc. ICASSP*, 2018.

[13] 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.

[14] K. Maekawa, "Corpus of spontaneous japanese: Its design and evaluation," in *ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition*, 2003.

[15] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *Proc. ICASSP*, 2015.

[16] G. Heigold, W. Macherey, R. Schluter, and H. Ney, "Minimum exact word error training," in *Proc. ASRU*, 2005, pp. 186–190.

[17] Z. Chen, J. Droppo, J. Li, W. Xiong, Z. Chen, J. Droppo, J. Li, and W. Xiong, "Progressive joint modeling in unsupervised single-channel overlapped speech recognition," *IEEE/ACM Trans. on ASLP*, vol. 26, no. 1, pp. 184–196, 2018.

[18] 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 *Proc. ASRU*, 2011.

[19] V. Peddinti, G. Chen, D. Povey, and S. Khudanpur, "Reverberation robust acoustic modeling using i-vectors with time delay neural networks," in *Proc. Interspeech*, 2015, pp. 2440–2444.

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

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

[22] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *Proc. ICASSP*, 2017, pp. 5220–5224.

[23] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, "Phoneme recognition using time-delay neural networks," *IEEE Trans. ASSP*, vol. 37, no. 3, pp. 328–339, 1989.

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

[25] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors." in *Proc. ASRU*, 2013, pp. 55–59.

[26] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts." in *Proc. INTERSPEECH*, 2015, pp. 3214–3218.

[27] Y. Wang, V. Peddinti, H. Xu, X. Zhang, D. Povey, and S. Khudanpur, "Backstitch: Counteracting finite-sample bias via negative steps," *Proc. INTERSPEECH*, pp. 1631–1635, 2017.

[28] T. N. Sainath, V. Peddinti, O. Siohan, and A. Narayanan, "Annealed f-smoothing as a mechanism to speed up neural network training," *Proc. Interspeech*, pp. 3542–3546, 2017.

[29] T. Kawahara, H. Nanjo, T. Shinozaki, and S. Furui, "Benchmark test for speech recognition using the Corpus of Spontaneous Japanese," in *ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition*, 2003.