IMPROVING LF-MMI USING UNCONSTRAINED SUPERVISIONS FOR ASR

Hossein Hadian[1,2]*, Daniel Povey*[2,3]*, Hossein Sameti*[1]*, Jan Trmal*[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

We present our work on improving the numerator graph for discriminative training using the lattice-free maximum mutual information (MMI) criterion. Specifically, we propose a scheme for creating unconstrained numerator graphs by removing time constraints from the baseline numerator graphs. This leads to much smaller graphs and therefore faster preparation of training supervisions. By testing the proposed unconstrained supervisions using factorized time-delay neural network (TDNN) models, we observe 0.5% to 2.6% relative improvement over the state-of-the-art word error rates on various large-vocabulary speech recognition databases.

Index Terms— lattice-free MMI, ASR, supervision, numerator graph.

1. INTRODUCTION

Maximum mutual information (MMI) is a commonly used objective function in automatic speech recognition (ASR) for sequence-discriminative training of acoustic models (AM). Unlike the maximum likelihood (ML) objective function which only maximizes the likelihood of the reference word sequence (usually represented by a composite hidden Markov model), MMI aims to maximize that likelihood while also minimizing the likelihood of all wrong word sequences. The composite hidden Markov model (HMM) graph is usually called the numerator graph in the context of MMI. The set of all word sequences is ideally represented by a denominator HMM graph which encodes all possible sequences of words. However, usually approximations are used instead of a full denominator graph as it can make the computations slow. Traditionally n-best lists (i.e., a list of top wrong word sequences) and later, lattices were used to approximate the denominator graph [1][2]. The denominator lattices are generated using a previously trained model such as a cross-entropy model (e.g., CD-HMM-DNN [3]) or a Gaussian-mixture-model (GMM) based model. A denominator lattice compactly encodes a small set of likely alternative word sequences for a training utterance. Using a full denominator graph was investigated in [4] with hidden Markov model (HMM)-GMM models.

Recently, the lattice-free MMI (LF-MMI) approach was proposed in [5], which uses a full denominator graph with deep neural network (DNN) based models. LF-MMI is basically the same as lattice-based MMI but uses a special numerator graph (which exploits alignment information) and a common denominator graph (instead of utterance-specific lattices). The derivatives for the LF-MMI objective function are computed by doing two forward-backward passes: one on the numerator graph and one on the denominator graph. To make the denominator forward-backwards efficient and fast, three main techniques are used in LF-MMI: (a) all utterances are split to fixed 1.5-second chunks (using the alignment information) and training is done on minibatches of these chunks. (b) the denominator graph is created using a pruned phone-level (instead of word-level) language model trained on the alignments from a previous HMM-GMM model. (c) the denominator computations are done on graphics processing units (GPU) instead of CPUs.

The numerator graph used in LF-MMI (which encodes the supervision information) is a special acyclic graph (i.e., a lattice), which can exploit the alignment information from a previous HMM-GMM model as time constraints on the phones. More specifically, the numerator graph is a finite state acceptor (FSA) where each phone can occur a certain number of frames sooner or later than its occurrence time in the corresponding alignment. Similar ideas were used in the context of CTC to reduce decoding latency [6].

In this study we propose unconstrained supervisions for LF-MMI. Specifically, we relax the supervision time constraints in each chunk, so that the numerator graph is not acyclic anymore (it will have self-loops). This provides more freedom in each chunk. We try this approach using the recently proposed state-of-the-art TDNN-F models [7] and show improvements on certain databases. We also show 2x speed-up in preparing supervisions for DNN training using this approach.

Two prior studies [8, 9] performed HMM-DNN training without using alignments from a GMM system (i.e., GMM-free training), showing improvements over baseline models. However, note that in our proposed method, we still use the alignments obtained from a previously trained system in order to be able to split the training utterances into equal-duration

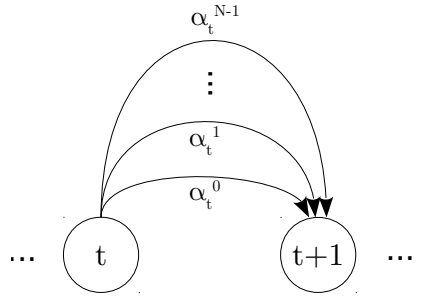

Fig. 1. Part of a time-enforcer FSA. $\{\alpha_t^0, \alpha_t^1, ..., \alpha_t^{N-1}\}$ is the set of pdf-ID's that are allowed at time-index t. This FSA is created on-the-fly.

(e.g., 1.5 second) chunks. The difference with our previous work on LF-MMI is that we now do not enforce time constraints within those chunks. The flat-start version of LF-MMI where no alignment information is used at all (i.e., the utterances are not split and there are no time constraints) was investigated in [10].

The rest of this paper is as follows. The baseline method for acquiring the time-constrained supervisions is explained in Section 2. In Section 3, we describe the proposed unconstrained supervisions. The experimental setup is explained in Section 4 and results are presented in Section 5. Finally the conclusions will appear in Section 6.

2. CONSTRAINED SUPERVISION

Since MMI training is well-known (for example see [11], [12]), we will not present equations for it and we will focus on the supervision creation process. In LF-MMI, the supervisions for a training utterance are created as follows: First, a lattice alignment of the utterance (containing alternative paths) is generated using a previously trained HMM-GMM model. Timing information of the phones are extracted from this lattice, in the form of a frame-by-frame mask which indicates what phones are allowed to appear on which frames with a tunable tolerance parameter (e.g., a tolerance of 5 frames). A time-enforcer FSA is created using this mask, where there is a transition from state t to $t+1$ with a particular *pdf-ID* (i.e., senone ID) on it, if that pdf-ID is from a phone that is allowed on the t^{th} frame. For illustration, part of a time-enforcer FSA is shown in Fig. 1. Then a composite HMM graph is created from the transcription (e.g., the kind of composite graph that is used in HMM-GMM training). This graph is converted to an FSA with pdf-ID as arc labels. Next, this composite HMM is composed with the time-enforcer FSA. This step will remove all the self-loops in the composite HMM, expanding them according to the timing information. Therefore, the resulting numerator FSA can be topologically sorted so that each state can be identified with a time-index. This allows splitting the supervision into chunks. Finally, this FSA (which is for the whole utterance) is split into fixed-size chunks. A sample numerator graph for a small 48-frame chunk is shown in Fig. 2. Since we use a frame-subsampling factor of 3, all paths in this FSA have a length of 16 arcs. Also note that to save space, we have used a tolerance of 3 for generating this FSA (we use 5 in experiments). We can see the expansion of self-loops, e.g., the run of states with pdf-ID 1737. This will be more clear when we compare with the unconstrained version of this FSA in Section 3.

Before the numerator graphs are used in training, there is one more step which is related to graph weights. As explained in [5], all numerator graphs are composed with the denominator graph to normalize the weights. This ensures that the MMI objective function values are never greater than zero. Additionally, the numerator graphs can benefit from the phone-level language model weights in the denominator graph.

Note that if we directly use the composite HMM (without composing with the time-enforcer FSA or splitting up) as supervision, this will lead to flat-start (i.e., from scratch) LF-MMI training [10][13].

3. PROPOSED UNCONSTRAINED SUPERVISION

In this study, we propose to relax the time constraints in each chunk. This is done as follows. All the steps in the baseline approach (as explained in Section 2) are performed, except for the fact that we use transition-ID's (i.e., non-tied HMM states) as the arc labels (instead of pdf-ID). To remove the time constraints from a chunk FSA, we first remove all arcs which their transition-ID indicates them as a self-loop (i.e., any arc that is an expanded version of the original self-loops in the composite HMM). Exceptions are the arcs that originate from the first state of the FSA. The reason for allowing them on the first frame, if they were already there, is because we want to allow phones to be cut in half on chunk boundaries. We do not have to do anything special on the last frame. Next, we add the self-loops to the FSA (according to the phone HMM topology and the transition-ID's on the arcs). Finally, all the transition-ID's are mapped to pdf-ID's. A sample unconstrained numerator FSA (corresponding to the same FSA in Fig. 2) is shown in Fig. 3. It can be seen that this numerator graph is considerably smaller than the constrained version shown in Fig. 2. This leads to substantial speed-ups in the final step of generating training examples, which is composing the numerator graphs with the denominator graph to normalize the transition weights (as explained in Section 2). By comparing Figs. 2 and 3, we can see how much different self-loops have been expanded.

To summarize, the unconstrained numerator graph described above allows all the phone sequences that the alignment allows up to a tolerance, without containing any timing information (i.e., without enforcing any state-level alignment). However, this approach cannot be considered com-

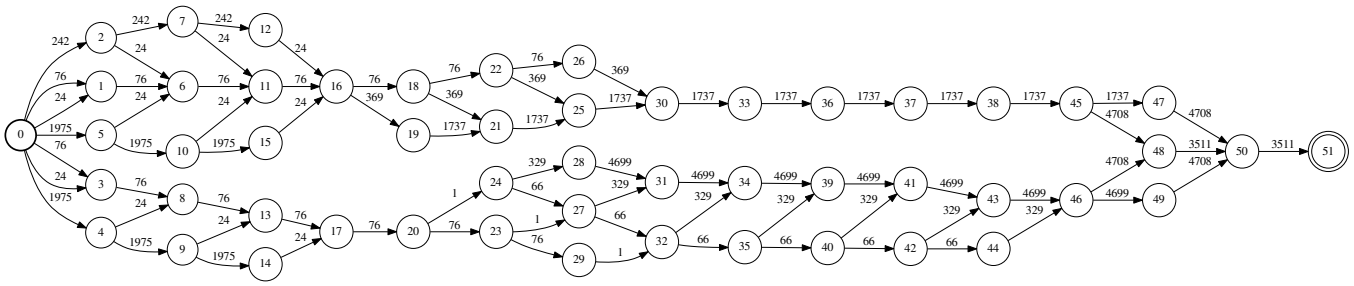

Fig. 2. Constrained numerator graph for a 48-frame chunk using tolerance = 3.

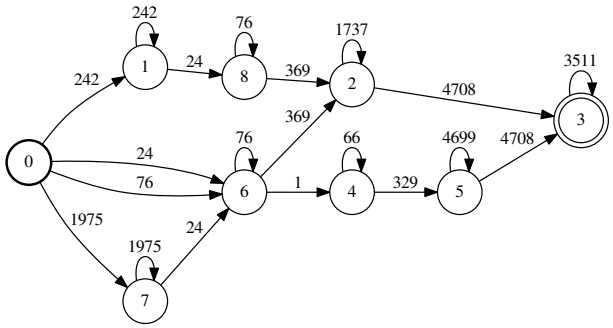

Fig. 3. Unconstrained numerator graph for a 48-frame chunk.

pletely unconstrained since we are still splitting the whole utterance to small chunks, i.e., we are globally enforcing constraints but locally inside the chunks there is no constraint. Note that we can't skip the lattice generation and the time-enforcer composition steps because otherwise we will not be able to split the utterances into chunks.

To compare this approach with flat-start LF-MMI, we can roughly consider them equivalent if the training data was already segmented to very short segments (e.g., 1.5 seconds).

4. EXPERIMENTAL SETUP

We use the open-source speech recognition toolkit Kaldi to run the experiments. The experiments presented in this paper are reproducible using this toolkit. We do most of our experiments on the 300-hour Switchboard database [14]. We evaluate on the Hub5 '00 set (also known as *eval2000*) and the RT03 test set. We also present results on TEDLIUM v2 [15], Wall Street Journal (WSJ) [16], AMI [17] and Librispeech [18].

4.1. Factorized TDNN

For the neural network, we use a factorized TDNN model. A factorized TDNN has a similar structure as a vanilla TDNN, except the weight matrices (of the layers) are factorized (using SVD) into two factors, with one of them constrained to be semi-orthonormal [7].

Chunk size (seconds)	Constrained 1.5	Unconstrained 1.5	Unconstrained 3.0
eval2000	13.1	**12.8**	12.9
RT03	15.4	**15.0**	15.1

Table 1. Impact of chunk size. Word error rates (in %) are shown for 2 test sets on the 300-hour Switchboard task.

In the experiments, we use exactly the same network and hyper-parameters for comparing constrained and unconstrained supervisions.

5. RESULTS

5.1. Impact of chunk length

Since regular LF-MMI supervisions are constrained, the final word error rates are not affected by chunk size. However, chunk size can impact the training process for the proposed unconstrained supervisions. Table 1 shows the results of using unconstrained supervisions on the 300-hour Switchboard task for two different chunk sizes. We see a slight degradation when using a larger chunk size (i.e., 3 seconds). That is expected, because of the extra freedom (and therefore uncertainty) in each chunk.

5.2. Noisy data

Table 2 shows the effect of using unconstrained supervisions on AMI – single distant microphone (SDM) case – which is a noisy database. We can see the relative improvements (in the first 2 rows) are small. Also, the last two rows show a case where we use a HMM-GMM model trained on the individual headset microphones (IHM) training data to get the lattice alignments for LF-MMI supervisions. Clearly, these alignments are better; however, it seems that in this case, removal of alignment information from the numerator graphs has degraded the word error rate; perhaps because they provide a good starting point.

		Constrained	Unconstrained
AMI-SDM	dev	37.1	**36.8**
	eval	40.7	**40.5**
AMI-SDM IHM-ali	dev	**35.9**	36.2
	eval	**39.7**	40.0

Table 2. Effect of unconstrained supervisions on word error rates (in %) for noisy data (AMI). IHM-ali means the alignments for creating supervisions are based on IHM data.

		Constrained	Unconstrained
Supervision prep time	Switchboard	278	**135**
	AMI-SDM	885	**254**
Training time	Switchboard	**99**	110
	AMI-SDM	116	**81**
Overall training time	Switchboard	**15.1 hr**	16.5 hr
	AMI-SDM	5.8 hr	**4.4 hr**

Table 3. Comparing supervision preparation and DNN training speed with constrained and unconstrained graphs. The timings shown are for processing 10,000 chunks (each chunk having 150 frames) in seconds, except for the last two rows which show the overall training time (using 8 GPUs).

5.3. Impact on speed

As explained in Section 2, using constrained supervisions leads to much smaller graphs which reduces disk usage and speeds up composition with the denominator graph (for weight normalization). As a result, we can prepare the supervisions in less time. Timing information is shown in Table 3 for Switchboard and AMI. We see 2x speed-up in preparing supervisions for Switchboard and almost 4x speed-up for AMI-SDM. The reason for achieving larger speed-ups on AMI, is that the aligned lattices are much bigger compared to Switchboard due to noise. In other words, there are many more alternative paths in the numerator graph resulting in bigger constrained graphs, which in turn lead to slower composition with the denominator graph.

The unconstrained graphs also have an effect on training speed. That is because in the constrained case, each state is active at exactly one time-index and we can take advantage of this to do forward-backward in $O(N)$ where N is the number of arcs in the constrained graph. By comparison, in the unconstrained case we need to do a full forward-backward, which may or may not be slower depending on the size of the constrained graph. For example, as shown in Table 3, training is almost 10% faster using the constrained setup for Switchboard, while for AMI (which is a noisy database) the unconstrained graphs lead to almost 30% speed-up in training.

5.4. Results on various databases

Finally, Table 4 summarizes the results of using unconstrained supervisions on various databases. We can see improvements in word error rate in most cases, with relative improvements ranging from 0.5% (on AMI-SDM) to 2.6% (on Switchboard). Although the improvements are small, they are consistent. On WSJ and mini-librispeech (which is a subset of Librispeech with 5.3 hours of training data[1]), there is either no improvement or some small degradation. Part of the reason could be that for small data, the constraints from the GMM system help because the DNN can overfit to the training data too much and learn bad alignments. We also see improvements on Librispeech – which is a large database (1000 hours) – especially on the harder test sets ("dev-other" and "test-other"). This is particularly useful, because the gains from supervision preparation speed-up are more significant on larger databases.

	Test-set	Constr.	Unconstr.	Rel.
AMI-SDM	dev	37.1	**36.8**	0.8%
	eval	40.7	**40.5**	0.5%
Switchboard	eval2000	13.1	**12.8**	2.3%
	RT03	15.4	**15.0**	2.6%
TEDLIUM	dev	7.4	**7.3**	1.4%
	test	7.7	**7.6**	1.3%
Librispeech	dev	**3.3**	**3.3**	0%
	dev-other	8.8	**8.7**	1.1%
	test	**3.8**	**3.8**	0%
	test-other	9.0	**8.8**	2.2%
WSJ	dev93	**4.3**	4.4	-2.1%
	eval92	**2.5**	**2.5**	0%
Mini-librispeech	dev	**8.5**	8.6	-1.1%

Table 4. Summary of results on various databases. Last column shows the relative improvement in word error rates.

6. CONCLUSIONS AND FUTURE WORK

In this study, we investigated the effect of removing time constraints from the MMI supervisions (specifically from the numerator graphs of the chunks) for large vocabulary speech recognition. This should not be confused with flat-start LF-MMI, where there is no constraint at all; here we still use the alignments from a previously trained model to split the utterances into small chunks; the difference however is that there are no constraints inside the chunks. Through experiments on various databases, we showed that the proposed unconstrained supervisions slightly improve the word error rate in most cases, and more importantly they are faster to prepare. In particular, we observed 2x and 4x speed-up in preparing supervisions for the Switchboard and AMI tasks, respectively. We also observed slight degradations in word error rate on smaller databases, which could be due to over-fitting.

A potential future work is to investigate creating supervisions using a simple alignment (which contains a single

[1]It is accessible from http://openslr.org/31/.

path only) instead of a lattice alignment. This can further speed up supervision preparation; however, preliminary experiments have shown some degradation in word error rate.

7. REFERENCES

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

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

[3] George E Dahl, Dong Yu, Li Deng, and Alex 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.

[4] 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.

[5] 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 *Proceedings of INTERSPEECH*, 2016.

[6] Andrew Senior, Haşim Sak, Félix de Chaumont Quitry, Tara Sainath, and Kanishka Rao, "Acoustic modelling with CD-CTC-SMBR LSTM RNNs," in *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2015, pp. 604–609.

[7] Daniel Povey, Gaofeng Cheng, Yiming Wang, Ke Li, Hainan Xu, Mahsa Yarmohamadi, and Sanjeev Khudanpur, "Semi-orthogonal low-rank matrix factorization for deep neural networks," in *Proceedings of INTERSPEECH*, 2018.

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

[9] Chao Zhang and Philip 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)*, 2014, pp. 5597–5601.

[10] Hossein Hadian, Hossein Sameti, Daniel Povey, and Sanjeev Khudanpur, "Flat-start single-stage discriminatively trained HMM-based models for ASR," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2018.

[11] 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)*, 1986, pp. 701–704.

[12] Daniel Povey, *Discriminative training for large vocabulary speech recognition*, Ph.D. thesis, University of Cambridge, 2005.

[13] Hossein Hadian, Hossein Sameti, Daniel Povey, and Sanjeev Khudanpur, "End-to-end speech recognition using lattice-free MMI," in *Proceedings of INTERSPEECH*, 2018.

[14] John J. Godfrey, Edward C. Holliman, and Jane McDaniel, "SWITCHBOARD: Telephone speech corpus for research and development," in *Proceedings of the 1992 IEEE International Conference on Acoustics, Speech and Signal Processing - Volume 1*, Washington, DC, USA, 1992, ICASSP'92, pp. 517–520, IEEE Computer Society.

[15] Anthony Rousseau, Paul Deléglise, and Yannick Estève, "Enhancing the TED-LIUM corpus with selected data for language modeling and more ted talks.," in *LREC*, 2014, pp. 3935–3939.

[16] 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*, Stroudsburg, PA, USA, 1992, HLT '91, pp. 357–362, Association for Computational Linguistics.

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

[18] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: an ASR corpus based on public domain audio books," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.