INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Comparison of Lattice-Free and Lattice-Based Sequence Discriminative Training Criteria for LVCSR

Wilfried Michel, Ralf Schlüter, Hermann Ney

Human Language Technology and Pattern Recognition, Computer Science Department,
RWTH Aachen University, 52056 Aachen, Germany
{michel,schlueter,ney}@cs.rwth-aachen.de

Abstract

Sequence discriminative training criteria have long been a standard tool in automatic speech recognition for improving the performance of acoustic models over their maximum likelihood / cross entropy trained counterparts. While previously a lattice approximation of the search space has been necessary to reduce computational complexity, recently proposed methods use other approximations to dispense of the need for the computationally expensive step of separate lattice creation.

In this work we present a memory efficient implementation of the forward-backward computation that allows us to use unigram word-level language models in the denominator calculation while still doing a full summation on GPU. This allows for a direct comparison of lattice-based and lattice-free sequence discriminative training criteria such as MMI and sMBR, both using the same language model during training.

We compared performance, speed of convergence, and stability on large vocabulary continuous speech recognition tasks like Switchboard and Quaero. We found that silence modeling seriously impacts the performance in the lattice-free case and needs special treatment. In our experiments lattice-free MMI comes on par with its lattice-based counterpart. Lattice-based sMBR still outperforms all lattice-free training criteria.

Index Terms: speech recognition, acoustic modeling, sequence discriminative training, lattice-free MMI

1. Introduction

Hybrid hidden Markov models (HMM) [1] have been used for decades in state-of-the-art large vocabulary continuous speech recognition (LVCSR) systems. Gaussian hybrid HMMs have since been replaced with feed forward deep neural networks (DNN) [2] and recurrent long short-term memory neural networks (LSTM)[3]. All approaches have a common discrepancy that while they are modeling speech which is sequential data, the training criterion maximum likelihood or cross-entropy is defined on the frame level and minimizes the frame error rate (FER). However, the recognition is done on sequence-level and also incorporates additional knowledge sources like the language model. Furthermore, the evaluation criterion is the word error rate (WER). This mismatch between training and recognition suggests that using training criteria which are based on whole sequences and incorporate additional knowledge sources can be helpful.

Sequence discriminative training of neural networks for Automatic Speech Recognition (ASR) has been shown to provide a significant reduction in word error rates, in contrast to cross-entropy training for each prevalent modeling technique. Discriminative Gaussian HMMs (GHMMs) achieved best performance when trained using sequence-discriminative criteria like maximum mutual information (MMI) [4], minimum phone error (MPE) [5] or minimum Bayes risk (MBR) related criteria [6]. Significant improvements using sequence-discriminative criteria for feed forward networks have been reported in literature [7, 8]. These improvements hold true for deep bidirectional LSTM-RNNs (BLSTM), although BLSTMs inherently make use of the whole sequence and are able to incorporate more contextual information [9].

One shortcoming of these criteria is, that for the calculation of the gradients for model updates forward backward computations over the exponentially growing set of all possible HMM state sequences are necessary. To make these computations feasible, some approximations are necessary. The traditional way to approximate the search space is to only consider all likely HMM state sequences, which are obtained by decoding the training data with a pretrained model. The search space is thereby confined to the search lattices which have to be created beforehand and stored to disk. Implicit-lattice methods, which do not store lattices, have been explored in the past [10].

Recently a different, lattice-free approach has been proposed [11]. Here the search space is not confined to some predetermined state sequences, only the calculation of numerator statistics which encode the supervision information is restricted to loosely follow some fixed, predetermined alignment. Additionally the language model (LM) has been simplified by limiting the LM context to phone-level four-gram which allows for more frequent recombination of state paths. This approach has been adopted and adapted to different sequence criteria [12, 13], score-fusion and system combination [14, 15] and settings where we completely dispense with the need of initial GMM models [16].

In this work we aim to get a better understanding of the differences between lattice-based and lattice-free versions of sequence discriminative training criteria. We therefore propose two implementations of the forward backward computations which have a lower memory consumption of only $\mathcal{O}(\sqrt{T})$ and $\mathcal{O}(\log(T))$ respectively compared to the $\mathcal{O}(T)$ of a standard implementation [17, 18]. This allows us, at the expense of longer runtime, to dispense of the need of lattices while still avoiding many other approximations used in other lattice-free implementations. With this we can make use of full length utterances and/or more complex language models.

2. Lattice-Free vs. Lattice-Based Training

In this work we want to highlight the similarities and the differences between the previously proposed lattice-free, our newly proposed lattice-free, and our lattice-based implementation using the MMI objective function as an example.

On a high level, the MMI criterion is given as the log posterior probability of the reference word sequence $\tilde{w}_1^{\tilde{N}}$ given the

Copyright © 2019 ISCA

http://dx.doi.org/10.21437/Interspeech.2019-2254

audio data x_1^T. A decomposition into similar contributions per training utterance is straight-forward.

$$F_{\text{MMI}} = \log\left(\tilde{w}_1^{\tilde{N}} \middle| x_1^T\right) \tag{1}$$

Then the usual decomposition into acoustic model and language model is employed and a sequence of time-aligned states is introduced as a hidden variable to arrive at the well-known result

$$F_{\text{MMI}} = \frac{\sum\limits_{\{s_1^T \in \tilde{w}_1^{\tilde{N}}\}} p\left(x_1^T, s_1^T \middle| \tilde{w}_1^{\tilde{N}}\right) p\left(\tilde{w}_1^{\tilde{N}}\right)}{\sum\limits_{N, \{w_1^N\}} \sum\limits_{\{s_1^T \in w_1^N\}} p\left(x_1^T, s_1^T | w_1^N\right) p\left(w_1^N\right)} \tag{2}$$

Where $\{w_1^N\}$ means the set of all possible word sequences of any length and $\{s_1^T \in w_1^N\}$ means the set of all HMM state sequences that are compatible with the given word sequence. Now different schemes apply different approximations.

2.1. Lattice-Based Training

In the classical, lattice-based approach, the sums in the denominator are approximated. Instead of summing over all possible word sequences the sum is restricted to a set of likely word sequences which are determined via a decoding run on the training data. An utterance specific lattice of hypotheses containing the word identities and word boundary times is written to disk and loaded every time this specific utterance is encountered during the training.

At training time, all possible state sequences for each word arc are constructed and aligned within the given word boundary times. For each hypothesis word arc the best state path is chosen. This can be seen as a Viterbi approximation in Equation 2, but formally this is incorrect since a single word sequence can appear with multiple state paths given that they have different word boundary times.

The probabilities can then be modeled by choice, e.g. use an n-gram count-based language model and the usual 3-state HMM topology.

2.2. Memory-Efficient Lattice-free Training

In the previously proposed approach for lattice-free MMI no utterance specific restrictions of the denominator sums apply. Instead, the search space is reduced by constraining the language model context to a phoneme n-gram and simplifying the HMM topology.

In [11], the sum in the numerator is constrained to loosely follow the reference GMM alignment up to a tolerance of a few frames. In [19], this constrained was relaxed. Here, the alignment only indirectly influences the state sequences, as the training utterances were cut into chunks with boundaries determined from the initial alignment.

In this work we aim at a direct comparison of lattice-free and lattice-based sequence discriminative training and intend to introduce as few constraints and approximations as possible. We therefore keep the original 3-state HMM topology with self loops and skips and also use a word-based language model (unigram). There is no restriction on or modification of the length of the training utterances.

This naturally leads to an increased search space, which directly translates to an increased usage of GPU resources. In a straightforward implementation of the forward backward computations one would do one forward pass while storing the α values to memory and one backwards pass while loading the α values and only keeping the current β value in memory. This leads to a linear memory complexity in terms of time and number of states which would be prohibitive given the full search graph.

Instead, while doing the forward pass we only store a selected few checkpoints to memory and immediately discard the other α values. During the backwards pass these checkpoints are loaded and only the α values for this specific block are recomputed and stored to memory. If we choose the block size to be $\sqrt{T}$ frames, then the memory consumption goes down to $\mathcal{O}\left(S \cdot 2\sqrt{T}\right)$ at the expense of having to do the forward pass twice.

This reduction is usually sufficient to handle all training samples of reasonable length. If a handling of longer sequences is necessary we propose a more complicated scheme with multiple layers of checkpointing where for each sub-block we again use checkpoints within the block to further reduce memory consumption. Ideally, if in each iteration the block size is reduced by a factor $\frac{1}{2}$, we arrive at a scheme with logarithmic memory consumption. The disadvantage is, that in this case the α values have to be reconstructed and discarded more often so that the time requirement is increased by a factor $\log(T)$.

Further savings can be obtained by pruning away states with low occupation probability and only checkpointing a reduced number of states to memory [17]. But this approach would then reintroduce restrictions on the sums as in the lattice-based case and is therefore not applied.

An overview of the proposed methods can be seen in Table 1. For our experiments we used equidistant checkpoints if not noted otherwise.

Table 1: *Memory and runtime complexity of proposed algorithms in terms of total number of frames T, number of states S and edges E in the decoding graph*

Checkpoints	Runtime	Memory
none	$E \cdot T$	$S \cdot T$
equidistant	$E \cdot T$	$S \cdot \sqrt{T}$
logarithmic	$E \cdot T \log(T)$	$S \cdot \log(T)$

3. Experimental setup

For our experiments we used the Switchboard training data. Since our cross-entropy baseline on this task has already strongly been optimized, we also report results on data collected in the Quaero project [20]. The size of the training data is comparable. We use the 300h portion of the original Switchboard corpus and 250h of broadcast news and pod-casts for Quaero

The acoustic model is a 6-layer bidirectional LSTM with 500 cells per layer and direction. We use 4500/9000 clustered triphone states as targets (Quaero/Switchboard). We use L2 normalization, dropout [21], gradient noise [22], and focal loss [23] as regularization techniques. No speaker adaptation or data perturbation techniques were used. The models are first trained until convergence with the cross-entropy (CE) objective function. The CE Model is used to generate the lattices for lattice-based sequence discriminative training. Both lattice-based and lattice-free training is initialized with the CE Model and the learning rate is chosen to be smaller than in the CE training.

For lattice-free training, the search space is constructed as a static composition of a word-level unigram language model, the training lexicon, triphone context with across word coarticula-

Table 2: *Number of states and edges in the decoding graph for different corpora*

Corpus	States	Edges
Switchboard [11]	24k	220k
Switchboard	185k	660k
Quaero	375k	1.25M
Librispeech	550k	2.5M

tion, and the 3-state HMM topology. The final size of the search space automaton is many times larger compared to the previous approach [11]. Sizes are given in Table 2.

Decoding is done with a count-based fourgram language model.

4. Results

4.1. Denominator Language Model

An important difference between lattice-free and lattice-based sequence training is the language model used in training. We would like to compare the influence of different language model context lengths on lattice-based sequence training performance. Tables 3 and 4 show the results for lattice-based MMI and sMBR training for different language models used in comparison with the best lattice-free result.

For the Quaero dataset, we see that for best performance a bigram language model should be used. This is in accordance to previous results [24]. For Switchboard we only saw improvements if a fourgram language model was used and even degradations if a unigram was used instead.

Table 3: *Influence of language model context on lattice-based sequence training performance on the Switchboard 300h dataset. Numbers are WER [%].*

Criterion	LM train	LM lattice	Total	SWB
CE	-	-	14.4	9.9
MMI	1gram	1gram	15.2	10.3
	1gram	4gram	15.4	10.5
	4gram	4gram	14.3	9.6
lattice-free	1gram	-	14.2	9.9
sMBR	1gram	1gram	14.4	9.7
	1gram	4gram	14.2	9.4
	4gram	4gram	13.9	9.5
lattice-free	1gram	-	14.3	10.0

Table 4: *Influence of language model context on lattice-based sequence training performance on the Quaero dataset. Numbers are WER [%].*

Criterion	LM train	LM lattice	dev'10	eval'13
CE	-	-	14.2	10.7
MMI	1gram	1gram	13.8	10.3
	2gram	2gram	13.4	10.2
	4gram	4gram	13.9	10.5
lattice-free	1gram	-	13.8	10.3
sMBR	1gram	1gram	13.5	10.2
	2gram	2gram	13.2	10.0
	4gram	4gram	13.7	10.4
lattice-free	1gram	-	13.8	10.4

An advantage of lattice-free sequence discriminative training is the reduced training time and disk space footprint. The disk space argument is certainly true, the lattices take up several gigabytes of storage while the search space automaton only needs a few ten megabytes and can be generated on the fly.

4.2. Runtime Performance

We now compare the runtime performance of lattice-based and lattice-free sequence training with the baseline cross-entropy training in Table 5. For a fair comparison we have to additionally include the time it takes to create and post-process the lattices in the timing information. We decided to not do it here as these calculations are trivially parallelizable over many cheap CPU cores, so if it is necessary to quickly train and deploy a model, this task may only take a fraction of overall training time. Also, in frame-wise cross entropy training we have the possibility to split all utterances into equally sized chunks of small length (e.g. 50 frames) to further speed up the computation. This is not realistic for sequence criteria without imposing strong approximations on the criterion [11]. We therefore give two numbers for cross-entropy, one with and one without usage of the chunking technique.

Table 5: *Training time and real time factor (rtf) per full epoch on the Switchboard data set for different training criteria*

Criterion	Time [h]	RTF
CE	3.5	0.012
– chunking	5.0	0.017
Lattice-based MMI	35.0	0.12
Lattice-free MMI	34.2	0.11

We found a tenfold increase in training times per epoch independent of the method used compared to the cross-entropy baseline. Interestingly, there is already an increase by factor 1.5 when going from a chunked to a non-chunked training procedure. The time used for the numerator and denominator calculations is about 85% of the total training time. This is in strong contrast to the results reported in [11], where these calculations only took up a small fraction of the total training time. But this discrepancy is to be expected, as the training time is proportional to the number of edges in the search graph and as our implementation does the forward computations twice (cf. Section 2).

This is, however, not a severe limitation of our approach, since we use sequence discriminative training only as a fine-tuning step after successfully training a cross-entropy model. The total time spent for discriminative training of the neural network is still comparable to the frame-wise training time.

4.3. Convergence Speed

We also want to compare lattice-based and lattice-free methods in terms of convergence speed. In Figure 1 the progression of word error rate over two epochs of training is shown.

We observe that the lattice-based curve not only converges faster to a minimum but also achieves slightly better error rates given the same conditions. For the lattice-based case, we also confirmed previous observations [15, 8] that most of the improvements from sequence discriminative training happen in a fraction of an epoch. The lattice-free curve shows first an increased error rate which then goes down below the cross-

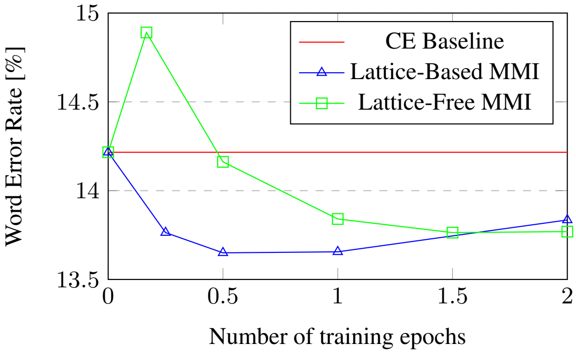

Figure 1: *Convergence speed of lattice-free and lattice-based MMI training on Quaero dataset, both using unigram word LM. WER on the dev2010 set.*

entropy starting point. This might indicate that the lattice-free model leaves the vicinity of a local optimum around the CE-model. It does, however, not lead to a lower error rate.

4.4. Silence

An important observation of our experiments was that silence requires far more attention in the lattice-free case than if lattices are used.

It is common practice to introduce a scale to each probability model to separately broaden or sharpen the distribution and to tune the relative weight of each model separately. When we use Viterbi approximation, only the relative scales matter. For sequence training and when using the sum also the value of the global scale matters.

In lattice-based sequence training it has empirically been found that setting the language model scale to 1 and reducing the acoustic scale by a factor on theorder of the best LM scale used in recognition gives generally good performance [8, 24]. For lattice-free training, this choice leads to a serious degradation in performance as more and more frames are classified as silence frames. Figure 2 shows how the silence ratio changes over the course of the training.

For lattice-based training the ratio stays constant. In the lattice-free case, the ratio increases depending on the scales. The best performance for lattice-free training was achieved by not scaling the models, i.e. setting all scales to 1.

4.4.1. Silence weight for sMBR

In our experiments with lattice-free sMBR criterion we noticed a severe degradation compared with the lattice-based case. In [12] it was suggested to downweight silence frames in the accuracy computation by a constant factor. We also tried this (Table 6) but found that this weighting did not influence the result.

Table 6: *Influence of weighting silence frame accuracy by a constant factor for lattice-free sMBR training. Numbers represent WER [%].*

Silence weight	Total	SWB
10^0	14.3	10.0
10^{-1}	14.3	10.0
10^{-2}	14.3	10.0
10^{-3}	14.4	10.0

4.4.2. Acoustic and Language Model scales

As already shown in Figure 2 the scales of our models used during training can have a strong influence on the convergence of lattice-free training. In Table 7 we present our results.

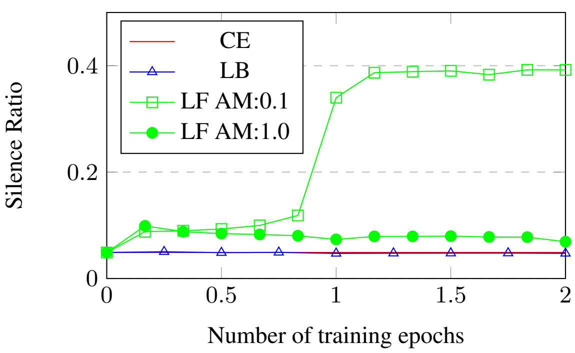

Figure 2: *Fraction of the training frames classified as silence over time for the cross-entropy (CE) baseline and lattice-based (LB) and lattice-free (LF) MMI trained models using different acoustic model scales (AM).*

While a low acoustic scale can seriously degrade the performance of the trained model, if it falls under a certain threshold, the language model scale seems to be of minor influence.

Table 7: *Influence of acoustic (AM) and language model (LM) scale on training performance for lattice-free MMI training. The other scale has been kept constant at* 1.0*. Numbers represent WER [%] on the total Hub5'00 set.*

AM scale	Total
0.1	26.0
0.2	14.7
1.0	14.2

LM scale	Total
0.2	14.2
1.0	14.2
10.0	14.3

5. Conclusions

In this work we presented a memory-efficient approach for calculating lattice-free sequence training criteria on GPU despite using a full word language model and full length utterances.

Comparing lattice-based and lattice-free training criteria we found that with careful tuning the lattice-free MMI trained model approaches the performance of a comparable lattice-based MMI model. The Lattice-based sMBR criterion still outperforms all lattice-free training criteria while the lattice-free version consistently performed worst.

Special attention is required with respect to silence frames in the lattice-free case. Without restrictions on set word-boundaries silence tends to dominate the model. Similar problems have been encountered in the case of frequent realignments using LSTM acoustic models in the context of full-sum training or CTC [25].

Acknowledgements

This work has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 694537, project "SEQCLAS" and Marie Skłodowska-Curie grant agreement No 644283, project "LISTEN") and from a Google Focused Award. The work reflects only the authors' views and none of the funding parties is responsible for any use that may be made of the information it contains. The GPU cluster used for the experiments was partially funded by Deutsche Forschungsgemeinschaft (DFG) Grant INST 222/1168-1.

The authors would like to thank Albert Zeyer, Eugen Beck, and Wei Zhou for many helpful discussions.

References

[1] H. A. Bourlard and N. Morgan, "Connectionist speech recognition: a hybrid approach," vol. 247, 1994.

[2] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," in *INTERSPEECH 2011, 12th Annual Conference of the International Speech Communication Association, Florence, Italy, August 27-31, 2011*, 2011, pp. 437–440.

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

[4] Y. Normandin, "Maximum mutual information estimation of hidden markov models," *In Automatic Speech and Speaker Recognition*, vol. 355, pp. 57–81, 01 1996.

[5] D. Povey, "Discriminative training for large vocabulary speech recognition," Ph.D. dissertation, Cambridge University, 01 2003.

[6] M. Gibson and T. Hain, "Hypothesis spaces for minimum bayes risk training in large vocabulary speech recognition," in *INTERSPEECH 2006 - ICSLP, Ninth International Conference on Spoken Language Processing, Pittsburgh, PA, USA, September 17-21, 2006*, 2006.

[7] B. Kingsbury, T. N. Sainath, and H. Soltau, "Scalable minimum bayes risk training of deep neural network acoustic models using distributed hessian-free optimization," pp. 10–13, 2012.

[8] K. Veselý, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks," in *INTERSPEECH 2013, 14th Annual Conference of the International Speech Communication Association, Lyon, France, August 25-29, 2013*, 2013, pp. 2345–2349.

[9] P. Voigtlaender, P. Doetsch, S. Wiesler, R. Schlüter, and H. Ney, "Sequence-discriminative training of recurrent neural networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, April 19-24, 2015*, 2015, pp. 2100–2104.

[10] S. 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 & Language Processing*, vol. 14, no. 5, pp. 1596–1608, 2006.

[11] D. Povey, V. Peddinti, D. Galvez, P. Ghahremani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely sequence-trained neural networks for ASR based on lattice-free MMI," in *Interspeech 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*, 2016, pp. 2751–2755.

[12] N. Kanda, Y. Fujita, and K. Nagamatsu, "Lattice-free state-level minimum bayes risk training of acoustic models," in *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018.*, 2018, pp. 2923–2927.

[13] C. Weng and D. Yu, "A comparison of lattice-free discriminative training criteria for purely sequence-trained neural network acoustic models," in *arXiv:1811.03700*, 2019.

[14] N. Kanda, Y. Fujita, and K. Nagamatsu, "Investigation of lattice-free maximum mutual information-based acoustic models with sequence-level kullback-leibler divergence," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2017, Okinawa, Japan, December 16-20, 2017*, 2017, pp. 69–76.

[15] W. Xiong, L. Wu, F. Alleva, J. Droppo, X. Huang, and A. Stolcke, "The microsoft 2017 conversational speech recognition system," Tech. Rep., 2018.

[16] H. Hadian, H. Sameti, D. Povey, and S. Khudanpur, "Flat-start single-stage discriminatively trained hmm-based models for ASR," *IEEE/ACM Trans. Audio, Speech & Language Processing*, vol. 26, no. 11, pp. 1949–1961, 2018.

[17] G. Zweig and M. Padmanabhan, "Exact alpha-beta computation in logarithmic space with application to MAP word graph construction," in *INTERSPEECH*, 2000.

[18] J. Binder, K. P. Murphy, and S. J. Russell, "Space-efficient inference in dynamic probabilistic networks," in *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence, IJCAI 97, Nagoya, Japan, August 23-29, 1997, 2 Volumes*, 1997, pp. 1292–1296.

[19] H. Hadian, D. Povey, H. Sameti, J. Trmal, and S. Khudanpur, "Improving LF-MMI using unconstrained supervisions for ASR," in *2018 IEEE Spoken Language Technology Workshop, SLT 2018, Athens, Greece, December 18-21, 2018*, 2018, pp. 43–47.

[20] "Quaero." [Online]. Available: http://www.quaero.org/

[21] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[22] A. Neelakantan, L. Vilnis, Q. V. Le, I. Sutskever, L. Kaiser, K. Kurach, and J. Martens, "Adding gradient noise improves learning for very deep networks," *CoRR*, vol. abs/1511.06807, 2015. [Online]. Available: http://arxiv.org/abs/1511.06807

[23] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," pp. 2999–3007, 2017.

[24] R. Schlüter, B. Müller, F. Wessel, and H. Ney, "Interdependence of language models and discriminative training," in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, vol. 1, Keystone, CO, Dec 1999, p. 119–122.

[25] A. Zeyer, E. Beck, R. Schlüter, and H. Ney, "CTC in the context of generalized full-sum HMM training," in *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*, 2017, pp. 944–948.