



Self-Normalized Importance Sampling for Neural Language Modeling

Zijian Yang¹, Yingbo Gao^{1,2}, Alexander Gerstenberger¹,
Jintao Jiang³, Ralf Schlüter^{1,2}, Hermann Ney^{1,2}

¹Human Language Technology and Pattern Recognition Group, Computer Science Department
RWTH Aachen University, 52074 Aachen, Germany ²AppTek GmbH, 52062 Aachen, Germany
³ AppTek, McLean, Virginia, USA

{zyang|ygao|schlueter|ney}@cs.rwth-aachen.de
alexander.gerstenberger@rwth-aachen.de, jjiang@apptek.com

Abstract

To mitigate the problem of having to traverse over the full vocabulary in the softmax normalization of a neural language model, sampling-based training criteria are proposed and investigated in the context of large vocabulary word-based neural language models. These training criteria typically enjoy the benefit of faster training and testing, at a cost of slightly degraded performance in terms of perplexity and almost no visible drop in word error rate. While noise contrastive estimation is one of the most popular choices, recently we show that other sampling-based criteria can also perform well, as long as an extra correction step is done, where the intended class posterior probability is recovered from the raw model outputs. In this work, we propose self-normalized importance sampling. Compared to our previous work, the criteria considered in this work are self-normalized and there is no need to further conduct a correction step. Through self-normalized language model training as well as lattice rescoring experiments, we show that our proposed self-normalized importance sampling is competitive in both research-oriented and production-oriented automatic speech recognition tasks.

Index Terms: self-normalization, importance sampling, training criterion, neural network, language modeling

1. Introduction

Nowadays, word-based neural language models (LMs) consistently give better perplexities than count-based language models [1, 2], and are commonly used for second-pass rescoring or first-pass decoding of automatic speech recognition (ASR) outputs [3, 4, 5]. One challenge to train such LMs, especially when the vocabulary size is large, is the traversal over the full vocabulary in the softmax normalization. During both training and testing, this brings inefficiencies and calls for sampling-based methods to ease the calculation burdens. Previously, many sampling-based training criteria are proposed and investigated. Some prominent examples include: hierarchical softmax [6], negative sampling [7], importance sampling (IS) [8], and noise contrastive estimation (NCE) [9].

While NCE is one of the most popular choices for language modeling [10, 11, 12], recently Gao et al. [13] show that other sampling-based criteria can also perform well, as long as an extra correction step is done, where the intended class posterior probability is recovered from the raw model outputs. The motivation of this work begins from there. We notice that by introducing simple modifications to the original IS training criteria, the additional correction step can be omitted and the class posteriors can be directly obtained from the model outputs. In other words, LMs trained with such criteria are also self-normalized -

straightforwardly, we call this type of training: self-normalized importance sampling.

Compared to Gao et al. [13], our method is simpler in the sense that no additional correction is required, and the LMs are directly trained to give the intended class posteriors. Compared to NCE, which is also a self-normalized criterion, our method is comparable and competitive, which we further show through extensive experiments on both research-oriented and production-oriented ASR tasks.

2. Related Work

Neural network LMs are shown to bring consistent improvements over count-based LMs [14, 1, 2]. These neural LMs are then used either in second-pass lattice rescoring or first pass decoding for ASR [3, 4, 5, 15, 16]. To mitigate the problem of having to traverse over the full vocabulary in the softmax normalization, various sampling-based training criteria are proposed and investigated [6, 7, 8, 9, 10, 11, 12]. These methods mainly deal with how to build up a contrastive loss such that the model is able to tell true target words from noisy ones. Related but originated from a different motivation, that is, to explicitly encourage normalization during training and hope for self-normalization during testing, variance regularization is also discussed in literature [17]. Recently, in [13], the authors identify the relationships between model optimums under various sampling-based training criteria and the contextual posterior probabilities. They show that with one additional correction step, the intended posteriors can be recovered. This work can be thought of as a follow-up and extension of [13], where we propose several simple modifications to the IS training criterion to directly enable self-normalized models, without having to do the additional correction step.

3. Methodology

In this section, we first describe several training criteria, including Binary Cross Entropy (BCE), NCE and IS. Then, we pinpoint the reason why IS is not self-normalized by default. Afterwards, we move on to discuss three modes of modifications to enable self-normalization of IS. In the following discussions, we define c as the running index in the target vocabulary size C , x as the history for next word prediction and n as the running index of word position N .

3.1. Binary Cross Entropy (BCE)

BCE is a traditional training criterion, which only requires the model outputs $q_\theta(x, c)$ to be bounded in $(0, 1)$, and not neces-

sarily a normalized distribution.

$$F_{\text{BCE}} = \frac{1}{N} \sum_{n=1}^N \left\{ \log q_{\theta}(x_n, c_n) + \sum_{c \neq c_n} \log(1 - q_{\theta}(x_n, c)) \right\} \quad (1)$$

Given enough amount of training pairs (x_n, c_n) , it can be shown that the optimum $\hat{q}_{\theta}(x, c)$ when maximizing F_{BCE} is the empirical posterior distribution $p(c|x)$, which indicates that the model outputs are self-normalized.

3.2. Fundamentals of Sampling-Based Training Criteria

In sampling-based training criteria, for each training pair (x_n, c_n) , given the number of samples K and the noisy distribution D for sampling, the summation of some function $f(x_n, \tilde{c}_k, \mathbb{E}(\mathbb{C}_K(c)))$ over K random samples \tilde{c}_k , can approximate the summation of f over all the classes. The expected count $\mathbb{E}(\mathbb{C}_K(c))$ is the expectation of the number of the appearance of class c in all samples. For instance, if we do sampling with replacement, the expected count would be $KD(c)$. The term $\mathbb{E}(\mathbb{C}_K(c))$ in function f is usually used to control the variance, i.e. with the increase of K , the variance goes down and the approximation is more precise. In the following discussions, to simplify the notation, we define

$$\mathbb{E}(\mathbb{C}_K(c)) \equiv E_c \quad (2)$$

Omitting the summation over n for simplicity, the approximation of the sampling method is:

$$\sum_{k=1}^K f(x_n, \tilde{c}_k, E_{\tilde{c}_k}) \approx \sum_{c=1}^C E_c \cdot f(x_n, c, E_c) \quad (3)$$

3.3. Noise Contrastive Estimation (NCE)

NCE is a popular BCE-style criterion, with the function f in the following form:

$$f(x_n, \tilde{c}_k, E_{\tilde{c}_k}) = \log(1 - \frac{q_{\theta}(x_n, \tilde{c}_k)}{q_{\theta}(x_n, \tilde{c}_k) + E_{\tilde{c}_k}}) \quad (4)$$

More specifically, the NCE criterion can be written as:

$$F_{\text{NCE}} = \frac{1}{N} \sum_{n=1}^N \left\{ \log \frac{q_{\theta}(x_n, c_n)}{q_{\theta}(x_n, c_n) + E_{c_n}} + \sum_{k=1}^K \log(1 - \frac{q_{\theta}(x_n, \tilde{c}_k)}{q_{\theta}(x_n, \tilde{c}_k) + E_{\tilde{c}_k}}) \right\} \quad (5)$$

Here the first term in the summation over n corresponds to the target class c_n and the second term corresponds to the approximation shown in Equation (3). With enough amount of training data and samples K , the optimum of model output is $\hat{q}_{\theta}(x, c) \approx p(c|x)$.

3.4. Importance Sampling (IS)

IS is another sampling-based BCE-style criterion.

$$F_{\text{IS}} = \frac{1}{N} \sum_{n=1}^N \left\{ \log q_{\theta}(x_n, c_n) + \sum_{k=1}^K \frac{\log(1 - q_{\theta}(x_n, \tilde{c}_k))}{E_{\tilde{c}_k}} \right\} \quad (6)$$

Here, with enough amount of training data and samples K , the model optimum is not normalized.

$$\hat{q}_{\theta}(x, c) \approx \frac{p(c|x)}{1 + p(c|x)} \quad (7)$$

Nonetheless, as described in [13], $\hat{q}_{\theta}(x, c)$ can be transformed to $p(c|x)$ with an additional correction step.

In IS, the function f has a simpler form, which is the second log term in BCE criterion divided by the expected count.

$$f(x_n, \tilde{c}_k, E_{\tilde{c}_k}) = \frac{\log(1 - q_{\theta}(x_n, \tilde{c}_k))}{E_{\tilde{c}_k}} \quad (8)$$

According to Equation (3), the summation over samples is actually an approximation of the summation of the second log term in BCE over all classes,

$$\sum_{k=1}^K f(x_n, \tilde{c}_k, E_{\tilde{c}_k}) \approx \sum_{c=1}^C \log(1 - q_{\theta}(x_n, c)) \quad (9)$$

which makes IS very similar to the original BCE criterion, except that the sum is over all classes rather than excluding the target class (notice the summation over c in Equation (9) and (1)). Thus, our motivation is to modify the original IS criterion to have an approximation of the original BCE criterion, in order to directly obtain $p(c|x)$ as the optimum output, hence achieving self-normalized models.

3.5. Self-Normalized Importance Sampling

3.5.1. Sampling including the Target Class

When the target class is allowed to be sampled, from the discussion above, the model is not self-normalized. Therefore, an intuitive way to exclude the target class from the summation is to simply subtract the target term $\log(1 - q_{\theta}(x_n, c_n))$.

$$F_{\text{Model1}} = \frac{1}{N} \sum_{n=1}^N \left\{ \log q_{\theta}(x_n, c_n) + \sum_{k=1}^K \frac{\log(1 - q_{\theta}(x_n, \tilde{c}_k))}{E_{\tilde{c}_k}} - \log(1 - q_{\theta}(x_n, c_n)) \right\} \approx F_{\text{BCE}} \quad (10)$$

With this simple modification, the sampling-based criterion is exactly an approximation of BCE criterion and the model optimum is now at $p(c|x)$. However, if the target class is not sampled, the additional $-\log(1 - q_{\theta}(x_n, c_n))$ term can result in a large gradient. When target class c_n does not appear in \tilde{c}_k , we can consider the gradient with respect to $\log q_{\theta}(x_n, c_n)$ of a single sentence n .

$$\frac{\partial F_{\text{Model1},n}}{\partial \log q_{\theta}(x_n, c_n)} = 1 + \frac{q_{\theta}(x_n, c_n)}{1 - q_{\theta}(x_n, c_n)} \quad (11)$$

Suppose that $p(c_n|x_n)$ is a large number (close to one) and $q_{\theta}(x_n, c_n)$ is close to its optimum $p(c_n|x_n)$, the gradient is large and makes it hard to converge. In the following experimental results, we show that when K is small, which means there is less chance to sample the target class out, this can lead to a bad performance.

3.5.2. Sampling excluding the Target Class

Another way to obtain the self-normalized output is to find some proper distribution or function f to directly approximate the summation without target class c_n .

$$\sum_k f(x_n, \tilde{c}_k, E_{\tilde{c}_k}) \approx \sum_{c \neq c_n} \log(1 - q_{\theta}(x_n, c)) \quad (12)$$

Considering equation (3), we propose two approaches: letting the expected count of c_n be zero, i.e. $E_{c_n} = 0$, or having a

function f_{c_n} that is specific for each target class and letting the value of the function be zero when the summation index c is c_n , i.e. $f_{c_n}(x_n, c_n, E_{c_n}) = 0$.

For the first approach (Mode2), $E_{c_n} = 0$ is equivalent to $D(c_n) = 0$, i.e. the probability to sample the target class being zero. To this end, we utilize different distributions $D_{c_n}(c)$ with $D_{c_n}(c_n) = 0$ for different target classes during training. Compared to IS, Mode2 has the same formula, but different noise distributions for sampling.

$$F_{\text{Mode2}} = \frac{1}{N} \sum_{n=1}^N \left\{ \log q(x_n, c_n) + \sum_{k=1}^K \frac{\log(1 - q(x_n, \tilde{c}_k))}{E_{\tilde{c}_k}} \right\} \quad (13)$$

Because in this way it is not possible to sample the target class c_n , there is no division-by-zero problem in the criterion.

For the second approach, we simply set f_{c_n} to zero when the target class gets sampled out.

$$F_{\text{Mode3}} = \frac{1}{N} \sum_{n=1}^N \left\{ \log q(x_n, c_n) + \sum_{k=1}^K f_{c_n}(x_n, \tilde{c}_k, E_{\tilde{c}_k}) \right\} \quad (14)$$

where

$$f_{c_n}(x_n, \tilde{c}_k, E_{\tilde{c}_k}) = \begin{cases} 0 & \text{if } \tilde{c}_k = c_n \\ \frac{\log(1 - q(x_n, \tilde{c}_k))}{E_{\tilde{c}_k}} & \text{else} \end{cases} \quad (15)$$

Compared to Mode2, Mode3 is more efficient since it can use one distribution for all training pairs (x_n, c_n) . For Mode2, on the other hand, it is necessary to use different distributions for different target classes. For Mode3, it is safer to do sampling without replacement, since if replacement is allowed, in extreme cases there can be many samples where f evaluates to zero, which adds no contribution to the summation and thus can influence the precision of the approximation.

4. Experimental Results

4.1. Experimental Setup

We conduct experiments on public datasets including Switchboard and Librispeech. Additionally, we experiment with one in-house dataset: 8kHz US English (EN). The detailed statistics of these three datasets can be found in Table 1. For EN, 7473M in Table 1 is the total amount of running words used for baseline count-based LM training, while roughly 694M running words are used for neural network training.

We follow the setup in [13] in terms of model architectures. For Switchboard and EN, we use LSTM LMs with 2 layers, 2048 hidden dimension and 128 embedding dimension. For Librispeech, we use Transformer [18] LMs, with 42 layers, 2048 feed-forward hidden dimension, 512 key/query/value dimension, 8 attention heads, and 512 input embedding dimension, following the architecture in [2]. The acoustic models are hybrid hidden Markov models, with 7 bidirectional LSTM layers and 500 dimension. We refer the reader to [19] and [20] for more details.

We evaluate our models with perplexities (PPLs) and second-pass rescoring word error rates (WERs). For Switchboard and EN, the lattices for rescoring are generated with 4-gram Kneser-Ney LMs [19]. For librispeech, the lattice is generated with a LSTM LM [4]. All the model outputs are normalized properly when computing PPLs in order to have a fair comparison. In rescoring, the outputs of models trained with

Table 1: Data statistics of the three datasets.

corpus	vocabulary size	running words
Switchboard	30k	24M
Librispeech	200k	812M
EN	250k	7473M

normalized or self-normalized criteria are used as is and no additional renormalization is done. For Importance Sampling, as the output cannot directly be used as scores, we obtain scores by a proper mapping of outputs [13]. We report rescoring results on clean and other test sets for Librispeech and on Switchboard (SWB) and CallHome (CH) Hub5'00 test sets for Switchboard. For EN, we use an in-house test set.

For sampling-based criteria, we use log-uniform distributions as noise distributions, following [13]. If not mentioned otherwise, the number of samples K is 8000. We do sampling with replacement, except Mode3. During training, each batch shares the same sampled classes for efficiency [21, 22], except Mode2. For Mode2, to build up the desired distribution for each training pair (x_n, c_n) , we use $C - 1$ as the number of labels in log-uniform, and map the sampled indices \tilde{c}'_k to \tilde{c}_k .

$$\tilde{c}_k = \begin{cases} \tilde{c}'_k & \text{if } \tilde{c}'_k < c_n \\ \tilde{c}'_k + 1 & \text{else} \end{cases} \quad (16)$$

The average training time per batch is evaluated on NVIDIA GeForce GTX 1080 Ti GPUs, with a batch size of 32 for Switchboard, 64 for LibriSpeech and EN.

4.2. Main Results

We first compare three modes of self-normalized IS on Switchboard, employing NCE as the baseline. As shown in Table 2, for Mode1, when $K = 100$, the performance is much worse than other methods. When $K = 8000$, on the other hand, the performance is on par with others, which verifies our statement about the big gradient problem in Section 3.5.1. For both $K = 100$ and $K = 8000$ in Mode2 and Mode3, the performance is comparable with NCE. Considering that Mode3 is more efficient than Mode2 in terms of the noise distribution, we choose Mode3 to be our main method. For both Mode2 and Mode3, we contribute the relatively small difference in PPLs to the small vocabulary size.

Table 2: PPLs of NCE and Mode{1,2,3} on Switchboard.

criterion	K	PPL
NCE	100	54.4
	8000	52.7
Mode1	100	87.2
	8000	53.0
Mode2	100	53.9
	8000	53.5
Mode3	100	54.0
	8000	53.0

We further investigate the influence of the number of samples K on Switchboard in Mode3. Figure 1 shows that when K is relatively small, the increase of K almost does not influence the training speed, as the computation effort for the sum over K is too small to influence the whole training procedure. When K is large enough, increasing K exponentially will slow

down the training speed dramatically. On the other hand, when K is relatively small, the increase of K brings huge improvements in PPL. Because when K is small, the sampling without replacement is similar to the sampling with replacement case, where the variance is proportional to $1/K$ and thus goes down fast with the increase of K , leading to a better approximation of the summation and fast decrease of PPL. When K is large enough, the approximation is already good, so with the increase of K , the PPL only goes down slowly, and gradually converges. One can find the best trade-off between training speed and PPL according to the specific task. Surely, the trade-off also depends on the model architecture and the vocabulary size, and thus we do not do further experiments to investigate this relationship on other datasets.

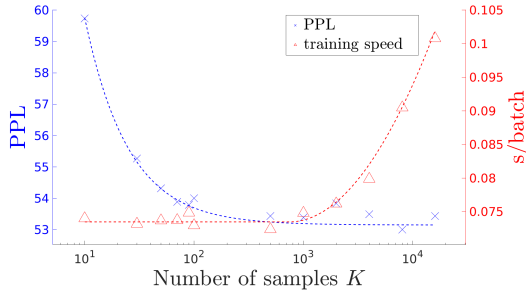


Figure 1: The influence of number of samples K .

Table 3 shows the comparison of different criteria on Switchboard. In this task, LSTM LMs are interpolated with count-based LM for evaluation. In terms of PPLs, the CE baseline is slightly better than others, because the model outputs are well-normalized during training, which is more consistent with the evaluation on PPL. All other methods have similar performance. In terms of WER, CE baseline is slightly better, but the advantage is not large enough to draw the conclusion that CE outperforms sampling-based methods. In terms of training speed, sampling-based methods are faster than the traditional methods, since for sampling-based methods, the summation is over K , which is less than C . However, since the vocabulary size of Switchboard is not so large, the training speedup is limited. Compared to the original IS, the output of self-normalized IS (Mode3) can be used as scores directly in rescoring without correction, giving comparable training speed and performance.

Table 3: WERs of different criteria on Switchboard.

criterion	train time (s/batch)	PPL	WER		
			All	SWB	CH
CE	0.100	49.9	10.1	6.8	13.4
BCE	0.107	52.3	10.3	6.9	13.7
IS	0.079	51.5	10.3	7.0	13.7
NCE	0.079	51.4	10.2	6.9	13.6
Mode3	0.090	51.7	10.2	6.9	13.6

Table 4 exhibits similar results on Librispeech. Since the vocabulary size is much larger than Switchboard, the training speedup gained from sampling is more significant, while the performance are still similar to traditional methods. With different model architectures and larger vocabulary sizes, we show that Mode3 is still competitive compared to NCE.

Experimental results on EN are presented in Table 5. Here, the baseline LM used for first-pass decoding is count-based.

Table 4: WERs of different criteria on Librispeech.

criterion	train time (s/batch)	PPL	WER	
			clean	other
CE	0.302	57.7	2.5	5.4
BCE	0.358	58.5	2.5	5.4
IS	0.206	58.4	2.6	5.5
NCE	0.206	57.9	2.5	5.4
Mode3	0.216	58.3	2.5	5.4

Table 5: WERs of different criteria on EN.

corpus	criterion	K	train speed (s/batch)	PPL	WER
EN	baseline	-	-	82.3	13.7
	NCE	100	0.092	65.9	13.3
		8000	0.098	55.9	13.1
	Mode3	100	0.089	68.0	13.3
		8000	0.114	55.8	13.1

With LSTM LM rescoring, we observe a great improvement in WER, which is consistent with the common knowledge about neural network LMs outperforming count-based LMs for ASR.

Besides, we observe that Mode3 has similar performance to NCE for both $K = 100$ and $K = 8000$ on the production-oriented ASR dataset, validating our claim that the proposed self-normalized IS training criterion is a competitive sampling-based training criterion for neural LMs.

5. Conclusion

We propose self-normalized importance sampling for training neural language models. Previously, noise contrastive estimation is a popular choice for sampling-based training criterion. In our recent work, we show that other sampling-based training criteria, including importance sampling, can perform on par with noise contrastive estimation. However, one caveat in that work is an additional correction step to recover the posterior distribution. In this work, we completely negate the need for this step by modifying the importance sampling criterion to be self-normalized. Through extensive experiments on both research-oriented and production-oriented datasets, we obtain competitive perplexities as well as word error rates with our improved method.

6. 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”). 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. This work was partially supported by the project HYKIST funded by the German Federal Ministry of Health on the basis of a decision of the German Federal Parliament (Bundestag) under funding ID ZMV11-2520DAT04A. We thank Christoph Lüscher for providing the LibriSpeech acoustic model and Eugen Beck for the lattices, and Markus Kitza for the SwitchBoard lattices.

7. References

- [1] M. Sundermeyer, R. Schlüter, and H. Ney, "Lstm neural networks for language modeling," in *Thirteenth annual conference of the international speech communication association*, 2012.
- [2] K. Irie, A. Zeyer, R. Schlüter, and H. Ney, "Language modeling with deep transformers," in *Interspeech*, Graz, Austria, Sep. 2019, pp. 3905–3909, iSCA Best Student Paper Award. [slides]. [Online]. Available: <http://arxiv.org/pdf/1905.04226.pdf>
- [3] X. Liu, Y. Wang, X. Chen, M. J. F. Gales, and P. C. Woodland, "Efficient lattice rescoring using recurrent neural network language models," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 4908–4912.
- [4] E. Beck, W. Zhou, R. Schlüter, and H. Ney, "Lstm language models for lvcsr in first-pass decoding and lattice-rescoring," *arXiv preprint arXiv:1907.01030*, 2019.
- [5] E. Beck, R. Schlüter, and H. Ney, "Lvcsr with transformer language models," in *INTERSPEECH*, 2020, pp. 1798–1802.
- [6] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2013. [Online]. Available: <http://arxiv.org/abs/1301.3781>
- [7] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, C. J. C. Burges, L. Bottou, Z. Ghahramani, and K. Q. Weinberger, Eds., 2013, pp. 3111–3119. [Online]. Available: <https://proceedings.neurips.cc/paper/2013/hash/9aa42b31882ec039965f3c4923ce901b-Abstract.html>
- [8] Y. Bengio and J.-S. Senécal, "Quick training of probabilistic neural nets by importance sampling," in *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, C. M. Bishop and B. J. Frey, Eds., vol. R4. PMLR, 03–06 Jan 2003, pp. 17–24, reissued by PMLR on 01 April 2021. [Online]. Available: <https://proceedings.mlr.press/r4/bengio03a.html>
- [9] M. Gutmann and A. Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, Y. W. Teh and M. Titterton, Eds., vol. 9. Chia Laguna Resort, Sardinia, Italy: PMLR, 13–15 May 2010, pp. 297–304. [Online]. Available: <https://proceedings.mlr.press/v9/gutmann10a.html>
- [10] A. Mnih and Y. W. Teh, "A fast and simple algorithm for training neural probabilistic language models," in *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*. icml.cc / Omnipress, 2012. [Online]. Available: <http://icml.cc/2012/papers/855.pdf>
- [11] J. Goldberger and O. Melamud, "Self-normalization properties of language modeling," in *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 764–773. [Online]. Available: <https://aclanthology.org/C18-1065>
- [12] X. Chen, X. Liu, M. J. F. Gales, and P. C. Woodland, "Recurrent neural network language model training with noise contrastive estimation for speech recognition," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5411–5415.
- [13] Y. Gao, D. Thulke, A. Gerstenberger, K. V. Tran, R. Schlüter, and H. Ney, "On Sampling-Based Training Criteria for Neural Language Modeling," in *Proc. Interspeech 2021*, 2021, pp. 1877–1881.
- [14] M. Sundermeyer, H. Ney, and R. Schlüter, "From feedforward to recurrent lstm neural networks for language modeling," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 3, pp. 517–529, Mar. 2015. [Online]. Available: <http://dx.doi.org/10.1109/TASLP.2015.2400218>
- [15] S. Kumar, M. Nirschl, D. Holtmann-Rice, H. Liao, A. T. Suresh, and F. Yu, "Lattice rescoring strategies for long short term memory language models in speech recognition," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2017, pp. 165–172.
- [16] K. Li, D. Povey, and S. Khudanpur, "A parallelizable lattice rescoring strategy with neural language models," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6518–6522.
- [17] X. Chen, X. Liu, Y. Wang, M. J. F. Gales, and P. C. Woodland, "Efficient training and evaluation of recurrent neural network language models for automatic speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 11, pp. 2146–2157, 2016.
- [18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.
- [19] M. Kitza, P. Golik, R. Schlüter, and H. Ney, "Cumulative adaptation for blstm acoustic models," *arXiv preprint arXiv:1906.06207*, 2019.
- [20] C. Lüscher, E. Beck, K. Irie, M. Kitza, W. Michel, A. Zeyer, R. Schlüter, and H. Ney, "Rwth asr systems for librispeech: Hybrid vs attention-w/o data augmentation," *arXiv preprint arXiv:1905.03072*, 2019.
- [21] B. Zoph, A. Vaswani, J. May, and K. Knight, "Simple, fast noise-contrastive estimation for large rnn vocabularies," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 1217–1222.
- [22] R. Jozefowicz, O. Vinyals, M. Schuster, N. Shazeer, and Y. Wu, "Exploring the limits of language modeling," *arXiv preprint arXiv:1602.02410*, 2016.