INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

On Sampling-Based Training Criteria for Neural Language Modeling

Yingbo Gao[1,2], *David Thulke*[1,2], *Alexander Gerstenberger*[1], *Khoa Viet Tran*[1], *Ralf Schlüter*[1,2], *Hermann Ney*[1,2]

[1]Human Language Technology and Pattern Recognition Group, Computer Science Department, RWTH Aachen University, 52074 Aachen, Germany
[2]AppTek GmbH, 52062 Aachen, Germany
{ygao|thulke|schlueter|ney}@cs.rwth-aachen.de,
{alexander.gerstenberger|khoa.tran}@rwth-aachen.de

Abstract

As the vocabulary size of modern word-based language models becomes ever larger, many sampling-based training criteria are proposed and investigated. The essence of these sampling methods is that the softmax-related traversal over the entire vocabulary can be simplified, giving speedups compared to the baseline. A problem we notice about the current landscape of such sampling methods is the lack of a systematic comparison and some myths about preferring one over another. In this work, we consider Monte Carlo sampling, importance sampling, a novel method we call compensated partial summation, and noise contrastive estimation. Linking back to the three traditional criteria, namely mean squared error, binary cross-entropy, and cross-entropy, we derive the theoretical solutions to the training problems. Contrary to some common belief, we show that all these sampling methods can perform equally well, as long as we correct for the intended class posterior probabilities. Experimental results in language modeling and automatic speech recognition on Switchboard and LibriSpeech support our claim, with all sampling-based methods showing similar perplexities and word error rates while giving the expected speedups.

Index Terms: sampling, training criterion, NCE, LM, ASR

1. Introduction

Enjoying the benefit of large amounts of text-only training data, language models (LMs) remain an important part of the modern automatic speech recognition (ASR) pipeline [1, 2, 3]. However, the large quantity of available data is a double-edged sword, posing real challenges in training. For example, the popular BERT model [4] and the recent GPT-2 and GPT-3 models [5, 6] have millions of parameters and are trained on billions of tokens. For BERT and GPT, both systems use byte pair encoding [7] to mitigate the problem of potentially very large vocabularies. However, for ASR, it is not uncommon for LMs to operate on the word level with a vocabulary size in the order of several hundred thousands [8, 9].

In order to train the neural LMs more efficiently, many speedup methods are proposed. To name a few: hierarchical softmax is used in [10], which changes the flat classification over all words to a series of binary decisions to arrive at the correct word; negative sampling (NS) method in [11] sums over a few sampled words instead of the full vocabulary; the noise contrastive estimation (NCE) method is proposed in [12] and adapted later in [13] for efficient estimation of LMs. Note that this is in no way an exhaustive enumeration of the ideas and methods, because there exist other works that introduce interesting concepts to address the problem, e.g. the Monte Carlo (MC) sampling and importance sampling (IS) discussed in [14]. For an overview of approximations to softmax, we refer the readers to a comprehensive blog post by Sebastian Ruder [15].

Among these methods, we find an interesting appreciation for NCE. In [16], the authors discuss the self normalization properties of models trained with NCE. In another line of work that maximizes mutual information for representation learning [17, 18, 19, 20], the NCE concept is frequently used. In a preprint note by Chris Dyer [21], a short conclusion is drawn "Thus, if your goal is language modeling, you should use NCE; if your goal is word representation learning, you should consider both NCE and negative sampling." As a result, if one overlooks the math and takes it for granted, methods like NS may not sound very attractive for language modeling.

In this paper, we start from three fundamental criteria, namely mean squared error (MSE), binary cross-entropy (BCE), and cross-entropy (CE). We explicitly write out the sampling-based versions of them. Then, we derive the theoretical optimums of the model, and show that although for some sampling-based criteria we may not directly obtain the original class posterior probabilities from the model outputs, because there is a one-to-one mapping between the optimum and the posterior, we can correct the model outputs and obtain the desired probabilities. By doing so, the model also has self-normalization properties and gives reasonable performances. Note that we are not here to argue that any of the cited work is fundamentally wrong, our goal is simply to raise awareness that models trained with sampling-based criteria other than NCE can also perform well, given enough care.

Our contribution can be summarized into two points:

- First, we examine various sampling methods systematically, linking back to traditional criteria MSE, BCE and CE, and derive optimums for them.
- Second, we show from both a theoretical and a practical perspective, that all these sampling methods under consideration work, giving similar perplexities (PPLs), word error rates (WERs) and speedups.

2. Related Work

Neural LMs are commonly used in second-pass rescoring [1, 2, 22, 23] or first-pass decoding [3] in ASR systems. While for conventional research-oriented datasets like Switchboard the word-level vocabulary size is several dozens of thousands, for larger systems, especially commercially available systems, the vocabulary size can often go up to several hundred thousand. Numerous methods to speed up the training (and potentially testing) of word-level LMs are proposed in the past decades [10, 11, 12, 13, 14, 15]. These methods either exploit the statistical structure or perform sampling in the large target vocabulary. Among the sampling methods, NCE enjoys special appreciation [17, 18, 19, 20, 21], whose self-normalization property

http://dx.doi.org/10.21437/Interspeech.2021-1067

is also examined [16]. For self-normalization and variance regularization, it is shown that explicit additional losses can also be added to the cross-entropy training criterion [24]. Traditionally, MSE, BCE, and CE are three training criteria that give the correct class posterior probabilities [25, 26]. In this paper, we make a connection between sampling-based criteria and these traditional ones and show that as long as one corrects for the intended class posterior probabilities, these sampling methods all show similar PPLs and WERs, while giving the expected speedups.

3. Sampling-Based Training Criteria

In this section, we formally define the training criteria. To clarify the notations, in the following sections, n is a running index in the number of word positions N, c, c' and $\tilde{c}$ are running indices in the target vocabulary C, which is supposedly very large. The context or history for next word prediction is denoted with x, and the Kronecker delta δ is used to decide the identity of model prediction and ground truth. We use θ for model parameters, q for model outputs, p for class posterior probabilities, and $\hat{q}$ to represent the derived optimum. When model outputs are explicitly normalized, $q(c|x)$ is used, otherwise, $q(x, c)$ is used. F represents the training criterion to maximize. Lastly, the distribution from which we sample is denoted with $\mathcal{D}$, and k is a running index in the number of samples K. Due to the limited pages, we only show the criterion definition and the optimum, and comment on the important steps in derivations in this paper.

3.1. Traditional Criteria

3.1.1. Mean Squared Error (MSE)

MSE is a classic training criterion commonly used for regression problems. Intuitively, it corresponds to "counting the errors", but in the continuous sense.

$$F_{\text{MSE}}(\theta) := -\frac{1}{N}\sum_{n}^{N}\sum_{c}^{C}\left(q_\theta(x_n, c) - \delta(c_n, c)\right)^2 \quad (1)$$

$$\Longrightarrow \hat{q}_\theta(x, c) = p(c|x) \quad (2)$$

To obtain the optimum, one could rewrite the summation in N into a summation in x, expand the squared term, and single out the terms related to q_θ. By definition, q can be unbounded, but it is common to parametrize q to be positive. In our preliminary experiments, further constraining q to be between zero and one with sigmoid slightly boosts the performance. However, although we spend a great amount of effort to tune the MSE-based models[1], the PPLs are still much worse than BCE and CE, therefore we only describe the MSE criterion here for the sake of completeness and not mention it later.

3.1.2. Binary Cross Entropy (BCE)

BCE is another traditional training criterion. The motivation of BCE can be summarized as to "encourage the correct predictions and discourage the wrong predictions".

$$F_{\text{BCE}}(\theta) := \quad (3)$$

$$\frac{1}{N}\sum_{n=1}^{N}\Big(\log q_\theta(x_n, c_n) + \sum_{c'\neq c_n}^{C}\log(1 - q_\theta(x_n, c'))\Big)$$

$$\Longrightarrow \hat{q}_\theta(x, c) = p(c|x) \quad (4)$$

Here, divergence inequality can be used for derivation. Note that q is required to be bounded in (0, 1) and it is commonly done via a sigmoid operation.

3.1.3. Cross Entropy (CE)

CE is arguably the most commonly used criterion nowadays and finds its roots in information theory and probabilistic theory. Intuitively, CE "encourages the model probability on the target word to be more exactly correct".

$$F_{\text{CE}}(\theta) := \frac{1}{N}\sum_{n=1}^{N}\log\frac{\exp q_\theta(x_n, c_n)}{\sum_{c'=1}^{C}\exp q_\theta(x_n, c')} \quad (5)$$

$$\Longrightarrow \frac{\exp\hat{q}_\theta(x, c)}{\sum_{c'=1}^{C}\exp\hat{q}_\theta(x, c')} = p(c|x) \quad (6)$$

Again, applying divergence inequality here would give us the optimum. We explicitly write out the softmax operation in this case to highlight the summation in the big vocabulary C in the denominator. In this case, q is unbounded and the softmax guarantees the normalized property.

3.2. Sampling-Based Criteria

According to the previous discussion about MSE, BCE, and CE, we see a summation in C in all three cases. This summation can be viewed as an expectation of some quantity Q_c under the uniform distribution $\frac{1}{C}$: $\sum_c^C Q_c = C\sum_c^C \frac{1}{C}Q_c = C\mathbb{E}(Q_c)$. Approximating this expectation is the core concept of the sampling methods.

3.2.1. Monte Carlo Sampling (MCS)

MCS approximates an expectation by some sample mean. Specifically, instead of summing over C, we sum over K random samples where $\tilde{c}_k \sim \mathcal{D}$. NS in [10] is a prominent example of MCS.

$$F_{\text{BCE-MCS}}(\theta) := \quad (7)$$

$$\frac{1}{N}\sum_{n=1}^{N}\Big(\log q_\theta(x_n, c_n) + \sum_{k=1}^{K}\log(1 - q_\theta(x_n, \tilde{c}_k))\Big)$$

$$\Longrightarrow \hat{q}_\theta(x, c) \approx (1 + \frac{K\mathcal{D}(c)}{p(c|x)})^{-1} \quad (8)$$

$$F_{\text{CE-MCS}}(\theta) := \quad (9)$$

$$\frac{1}{N}\sum_{n=1}^{N}\Big(q_\theta(x_n, c_n) - \log\sum_{k=1}^{K}\exp q_\theta(x_n, \tilde{c}_k)\Big)$$

$$\Longrightarrow \frac{\mathcal{D}(c)\exp\hat{q}_\theta(x, c)}{\sum_{c'=1}^{C}\mathcal{D}(c')\exp\hat{q}_\theta(x, c')} \approx p(c|x) \quad (10)$$

3.2.2. Importance Sampling (IS)

IS rewrites the expectation by introducing another distribution other than the uniform distribution. In our case, this new distri-

[1]For example, initializing the model with parameters from a converged NCE-trained model, changing optimizers and grid-searching over the learning rates. We also find that down-scaling the penalization MSE has on the rival classes improves the PPL a little, but similar experiments with BCE or CE do not give any improvements. Although authors of [27] claim that MSE should be preferred, we argue that for large number of target classes, it is hard for MSE to converge to the optimum with our current training recipes. Our empirical experience also agrees with that in [25].

bution is the noise distribution $\mathcal{D}$.

$$F_{\text{BCE-IS}}(\theta) := \tag{11}$$
$$\frac{1}{N}\sum_{n=1}^{N}\left(\log q_\theta(x_n, c_n) + \sum_{k=1}^{K}\frac{\log(1 - q_\theta(x_n, \tilde{c}_k))}{K\mathcal{D}(\tilde{c}_k)}\right)$$
$$\implies \hat{q}_\theta(x, c) \approx (1 + \frac{1}{p(c|x)})^{-1} \tag{12}$$
$$F_{\text{CE-IS}}(\theta) := \tag{13}$$
$$\frac{1}{N}\sum_{n=1}^{N}\left(q_\theta(x_n, c_n) - \log\sum_{k=1}^{K}\frac{\exp q_\theta(x_n, \tilde{c}_k)}{K\mathcal{D}(\tilde{c}_k)}\right)$$
$$\implies \frac{\exp\hat{q}_\theta(x, c)}{\sum_{c'=1}^{C}\exp\hat{q}_\theta(x, c')} \approx p(c|x) \tag{14}$$

3.2.3. Compensated Partial Summation (CPS)

CPS comes from a very simple motivation. If we replace the summation of C terms with a sub-summation of K terms, maybe we should compensate the partial summation by $\frac{C}{K}$. Essentially, this is still a Monte Carlo method with some correction term $\alpha = \frac{C}{K}$.

$$F_{\text{BCE-CPS}}(\theta) := \tag{15}$$
$$\frac{1}{N}\sum_{n=1}^{N}\Big(\log q_\theta(x_n, c_n) + \alpha\sum_{k=1}^{K}\log(1 - q_\theta(x_n, \tilde{c}_k))\Big)$$
$$\implies \hat{q}_\theta(x, c) \approx (1 + \frac{\alpha K\mathcal{D}(c)}{p(c|x)})^{-1} \tag{16}$$
$$F_{\text{CE-CPS}}(\theta) := \tag{17}$$
$$\frac{1}{N}\sum_{n=1}^{N}\Big(q_\theta(x_n, c_n) - \log\alpha\sum_{k=1}^{K}\exp q_\theta(x_n, \tilde{c}_k)\Big)$$
$$\implies \frac{\mathcal{D}(c)\exp\hat{q}_\theta(x, c)}{\sum_{c'=1}^{C}\mathcal{D}(c')\exp\hat{q}_\theta(x, c')} \approx p(c|x) \tag{18}$$

3.2.4. Noise Contrastive Estimation (NCE)

NCE changes the task of next-word prediction to a binary classification task of telling true samples from noisy ones. Originally, NCE is proposed in the context of BCE. Due to the property that the class posterior probabilities can be obtained directly from the model output, NCE is one of the most popular methods seen in literature [16, 17, 18, 19, 20, 21, 24].

$$F_{\text{BCE-NCE}}(\theta) := \frac{1}{N}\sum_{n=1}^{N}\left(\log\frac{q_\theta(x_n, c_n)}{q_\theta(x_n, c_n) + K\mathcal{D}(c_n)}\right. \tag{19}$$
$$\left. + \sum_{k=1}^{K}\log(1 - \frac{q_\theta(x_n, \tilde{c}_k)}{q_\theta(x_n, \tilde{c}_k) + K\mathcal{D}(\tilde{c}_k)})\right)$$
$$\implies \hat{q}_\theta(x, c) \approx p(c|x) \tag{20}$$
$$F_{\text{CE-NCE}}(\theta) := \frac{1}{N}\sum_{n=1}^{N}\left(\frac{q_\theta(x_n, c_n)}{q_\theta(x_n, c_n) + K\mathcal{D}(c_n)}\right. \tag{21}$$
$$\left. - \log\sum_{k=1}^{K}\exp(\frac{q_\theta(x_n, \tilde{c}_k)}{q_\theta(x_n, \tilde{c}_k) + K\mathcal{D}(\tilde{c}_k)})\right)$$
$$\implies \frac{\mathcal{D}(c)\exp(\frac{q_\theta(x,c)}{q_\theta(x,c)+K\mathcal{D}(c)})}{\sum_{c'}\mathcal{D}(c')\exp(\frac{q_\theta(x,c')}{q_\theta(x,c')+K\mathcal{D}(c')})} \approx p(c|x) \tag{22}$$

3.2.5. Notes on Derivation and Implications

The term $K\mathcal{D}(c)$ often shows up in our derivation due to rewriting the summation in K into a summation in C: $\sum_n^N\sum_k^K Q_{n,k} \approx \sum_n^N\sum_c^C K\mathcal{D}(c)Q_{n,c}$. Approximately, the number of terms that show up in the two summations should be equal, when given n and K is large enough. This trick is used in many of the derivations. Note that the model outputs q have different constraints in different criteria, the logits we plug into the criteria are activated according, e.g. applying sigmoid if q is bounded between zero and one.

From the derived optimums, if the model output $q(x, c)$ is not strictly the intended class posterior probability $p(c|x)$ (unlike in the case of three traditional criteria and BCE-NCE), we confirm the statement in [21] that such models are not directly applicable for language modeling. However, for all cases, it is clear that there is a one-to-one mapping between q and p. Therefore, given each model output $q(x, c)$, we can calculate the desired $p(c|x)$ (optionally querying the noise distribution $\mathcal{D}$), and use this quantity for rescoring in ASR.

4. Experiments

4.1. Experimental Setup

For the experimental validation, we train word-based LMs on LibriSpeech and SwitchBoard and evaluate the resulting models in well-tuned ASR systems with second-pass lattice rescoring. The vocabulary contains about 200k words for LibriSpeech and about 30k words for SwitchBoard, respectively.

For all training criteria, we make use of the same model architecture[2]. For LibriSpeech, our LMs make use of the Transformer [28] architecture, which is motivated by recent state-of-the-art results outperforming LSTM-based LMs on this task by a large margin [9, 29]. We use 42 layers, 512 input embedding dimension, 2048 feed-forward dimension, 8 attention heads, and 512 residual and key/query/value dimension. We do not use a positional encoding, following the architecture presented in [29]. For SwitchBoard, we use LSTM LMs with 2 layers and 2048 hidden units, and 128 embedding dimension.

For sampling-based approaches, we sample 8192 samples from log-uniform noise distributions, which performs well in our preliminary experiments. The noise samples are shared over the whole batch for computational efficiency in all cases [30, 31]. The parameters are learned using stochastic gradient descent with a learning rate of 1. The global gradient norms are clipped to 1. We implement our models using the open-source toolkit RETURNN[3] [32], based on TensorFlow [33].

The acoustic models are based on hybrid hidden Markov models. For detailed information on training setups, including discriminative training approaches, adaptation, and model architectures we refer the reader to [34] for the SwitchBoard and [9] for the LibriSpeech systems. For LibriSpeech the lattices are generated using a well-tuned LSTM LM in the first pass [3]. For SwitchBoard, the lattices are generated using a 4-gram Kneser-Ney LM [34]. We interpolate the LSTM LMs with count-based LMs for evaluation. We report on clean and other test sets for LibriSpeech and on SwitchBoard (SWB) and CallHome (CH) Hub5'00 test sets for SwitchBoard. During LM training we use a separate validation dataset. PPLs and WERs are always obtained with proper normalization over the full vocabulary unless

[2]Ideally, one should tune the hyperparameters for each training criterion individually. Given the computational and time constraints, we believe that this is still a reasonable approach.

[3]Example config is available at `https://git.io/Jnq3Q`.

noted otherwise. The reported average training time per batch is obtained on NVIDIA GeForce GTX 1080 Ti GPUs, with a batch size of 64 for LibriSpeech and 32 for Switchboard respectively.

4.2. Main Results

Below we present the main results. Table 1 gives the PPLs and WERs of our baseline models [3, 34]. In our opinion, these are competitive systems and serve as reasonable baselines for our purpose of judging different sampling methods.

Table 1: *Baseline PPLs and WERs.*

dataset	model	PPL	WER		
LibriSpeech	LSTM	64.3	-	clean	other
			-	2.6	5.8
SwitchBoard	4-gram	74.6	ALL	SWB	CH
			11.8	8.1	15.4

In Table 2, we present the PPLs and WERs of the sampling-based models on the LibriSpeech dataset. Since we sample 8k samples out of a 200k vocabulary, for all sampling methods under consideration, there is a significant relative training time speedup of over 40%, which is expected. Because of limited computational resources and our purpose not being to obtain the best trade-off between speed and quality, we do not sweep over different sampling sizes. Note that we do not include CE-NCE results due to some numerical problems. In terms of PPLs and WERs, we see a consistent improvement over the baseline, and we attribute this to using the Transformer and not the LSTM architecture [9, 29]. Comparing different sampling methods, we see a small variation in PPLs and even less in WERs. This justifies our statement that all these sampling methods work equally well when the model outputs are corrected accordingly.

Table 2: *Sampling-Based Transformer LMs on LibriSpeech.*

criterion	sampling	train time (ms/batch)	PPL	WER	
				clean	other
BCE	-	0.358	58.5	2.5	5.4
	MCS	0.213	58.0	2.6	5.4
	IS	0.206	58.4	2.6	5.5
	CPS	0.205	58.4	2.5	5.4
	NCE	0.206	57.9	2.5	5.4
CE	-	0.302	57.7	2.5	5.4
	MCS	0.206	57.9	2.5	5.4
	IS	0.201	58.7	2.5	5.4
	CPS	0.203	62.2	2.5	5.4

In Table 3, similar results are presented for the SwitchBoard dataset. In this case, all sampling-based methods give consistent relative training time speedup of over 20%. Considering that here 8k samples are drawn out of a 30k vocabulary, instead of 8k being drawn from a 200k vocabulary like that for LibriSpeech, the relative speedup being smaller is thus expected. PPL-wise, significant improvements over count-based baseline LM are achieved, which is also confirmed numerous times in different works on different datasets. The CE baseline is slightly better than the sampling-based LMs because our training recipe is well-tuned for the CE baseline. Comparing different sampling-based criteria, NCE and IS are slightly better, but the advantage is not large enough to justify one method being strictly superior than another. In terms of WERs, all sampling methods perform similarly.

Table 3: *Sampling-Based LSTM LMs on Switchboard.*

criterion	sampling	train time (ms/batch)	PPL	WER		
				All	SWB	CH
BCE	-	0.107	52.3	10.3	6.9	13.7
	MCS	0.077	52.6	10.3	7.0	13.6
	IS	0.079	51.5	10.3	7.0	13.7
	CPS	0.079	52.4	10.3	7.1	13.6
	NCE	0.078	51.4	10.3	7.0	13.6
CE	-	0.100	49.9	10.1	6.8	13.4
	MCS	0.078	52.4	10.4	7.0	13.7
	IS	0.076	52.3	10.2	6.9	13.6
	CPS	0.077	52.3	10.3	7.0	13.6

While PPLs and WERs reported above are properly normalized and give expected training time speedups, what makes these sampling methods especially attractive is the use case without explicit normalization. To this end, we look at the BCE sampling variants and directly rescore with the model outputs of these LMs, and report the WERs on Switchboard. As seen in Table 4, BCE-IS performs on par with BCE-NCE, as well as the CE baseline shown in Table 3, which shows its competitiveness and self-normalization properties. We notice that when plugging in the log-uniform distribution for $\mathcal{D}$, the BCE-MCS and BCE-CPS performance without normalization can get much worse to around 11.8% WER, which is similar to the count-based model. We attribute this to having to query the unreliable noise distribution $\mathcal{D}$ during search (which is not the case for BCE-IS and BCE-NCE) and therefore use a smoothed empirical unigram distribution for $\mathcal{D}$ instead.

Table 4: *WERs without Explicit Normalization on SwitchBoard.*

WER	BCE-MCS	BCE-IS	BCE-CPS	BCE-NCE
ALL	11.0	10.2	11.3	10.2
SWB	7.3	6.9	7.5	6.9
CH	14.7	13.6	15.1	13.6

5. Conclusion

For language modeling with large vocabularies, we consider different sampling-based training criteria. We start from three traditional criteria and formulate sampling-based versions of them. We derive optimums and argue that when model outputs are corrected for the intended class posteriors, these methods perform equally well compared to the popular noise contrastive estimation. Experimental evidence of perplexity and word error rate results on LibriSpeech and SwitchBoard support our claim. For direct rescoring without explicit normalization, we show the self-normalization properties of such sampling-based models.

6. Acknowledgements

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement n° 694537, project "SEQCLAS"). The work reflects only the authors' views and the European Research Council Executive Agency (ERCEA) is not responsible for any use that may be made of the information it contains. Simulations were performed with computing resources granted by RWTH Aachen University under project `rwth0582`. 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] X. Liu, Y. Wang, X. Chen, M. J. 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)*. IEEE, 2014, pp. 4908–4912.

[2] K. Irie, Z. Tüske, T. Alkhouli, R. Schlüter, H. Ney *et al.*, "Lstm, gru, highway and a bit of attention: an empirical overview for language modeling in speech recognition," in *Interspeech*, 2016, pp. 3519–3523.

[3] 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.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://www.aclweb.org/anthology/N19-1423

[5] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[6] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, 2020.

[7] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," *arXiv preprint arXiv:1508.07909*, 2015.

[8] Y. Wang, A. Mohamed, D. Le, C. Liu, A. Xiao, J. Mahadeokar, H. Huang, A. Tjandra, X. Zhang, F. Zhang *et al.*, "Transformer-based acoustic modeling for hybrid speech recognition," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6874–6878.

[9] 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.

[10] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[11] 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*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds., vol. 26. Curran Associates, Inc., 2013. [Online]. Available: https://proceedings.neurips.cc/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf

[12] 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*. JMLR Workshop and Conference Proceedings, 2010, pp. 297–304.

[13] A. Mnih and Y. W. Teh, "A fast and simple algorithm for training neural probabilistic language models," *arXiv preprint arXiv:1206.6426*, 2012.

[14] Y. Bengio, J.-S. Senécal *et al.*, "Quick training of probabilistic neural nets by importance sampling." in *AISTATS*, 2003, pp. 1–9.

[15] S. Ruder, "On word embeddings - Part 2: Approximating the Softmax," http://ruder.io/word-embeddings-softmax, 2016.

[16] J. Goldberger and O. Melamud, "Self-normalization properties of language modeling," *arXiv preprint arXiv:1806.00913*, 2018.

[17] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[18] B. Poole, S. Ozair, A. Van Den Oord, A. Alemi, and G. Tucker, "On variational bounds of mutual information," in *International Conference on Machine Learning*. PMLR, 2019, pp. 5171–5180.

[19] M. Tschannen, J. Djolonga, P. K. Rubenstein, S. Gelly, and M. Lucic, "On mutual information maximization for representation learning," *arXiv preprint arXiv:1907.13625*, 2019.

[20] L. Kong, C. d. M. d'Autume, W. Ling, L. Yu, Z. Dai, and D. Yogatama, "A mutual information maximization perspective of language representation learning," *arXiv preprint arXiv:1910.08350*, 2019.

[21] C. Dyer, "Notes on noise contrastive estimation and negative sampling," *arXiv preprint arXiv:1410.8251*, 2014.

[22] 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.

[23] K. Li, D. Povey, and S. Khudanpur, "A parallelizable lattice rescoring strategy with neural language models," *arXiv preprint arXiv:2103.05081*, 2021.

[24] X. Chen, X. Liu, Y. Wang, M. J. 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.

[25] P. Golik, P. Doetsch, and H. Ney, "Cross-entropy vs. squared error training: a theoretical and experimental comparison." in *Interspeech*, vol. 13, 2013, pp. 1756–1760.

[26] P. Golik, "Data-driven deep modeling and training for automatic speech recognition," Ph.D. dissertation, RWTH Aachen University, Computer Science Department, RWTH Aachen University, Aachen, Germany, Aug. 2020.

[27] L. Hui and M. Belkin, "Evaluation of neural architectures trained with square loss vs cross-entropy in classification tasks," *arXiv preprint arXiv:2006.07322*, 2020.

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[29] 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

[30] 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.

[31] R. Jozefowicz, O. Vinyals, M. Schuster, N. Shazeer, and Y. Wu, "Exploring the limits of language modeling," *arXiv preprint arXiv:1602.02410*, 2016.

[32] A. Zeyer, T. Alkhouli, and H. Ney, "RETURNN as a generic flexible neural toolkit with application to translation and speech recognition," in *Proceedings of ACL 2018, System Demonstrations*. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 128–133. [Online]. Available: https://www.aclweb.org/anthology/P18-4022

[33] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, and M. D. et al., "Tensorflow: A system for large-scale machine learning," in *Proc. USENIX Sympo. on Operating Systems Design and Impl. (OSDI 16)*, Savannah, GA, USA, Nov. 2016, pp. 265–283.

[34] M. Kitza, P. Golik, R. Schlüter, and H. Ney, "Cumulative adaptation for blstm acoustic models," in *Interspeech*, Graz, Austria, Sep. 2019, pp. 754–758.