INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

Estimation of gap between current language models and human performance

Xiaoyu Shen, Youssef Oualil, Clayton Greenberg, Mittul Singh, Dietrich Klakow

Spoken Language Systems (LSV)
Saarland University, Saarbrücken, Germany
xshen@lsv.uni-saarland.de, {firstname.lastname}@lsv.uni-saarland.de

Abstract

Language models (LMs) have gained dramatic improvement in the past years due to the wide application of neural networks. This raises the question of how far we are away from the perfect language model and how much more research is needed in language modelling. As for perplexity giving a value for human perplexity (as an upper bound of what is reasonably expected from an LM) is difficult. Word error rate (WER) has the disadvantage that it also measures the quality of other components of a speech recognizer like the acoustic model and the feature extraction. We therefore suggest evaluating LMs in a generative setting (which has been done before on selected hand-picked examples) and running a human evaluation on the generated sentences. The results imply that LMs need about 10 to 20 more years of research before human performance is reached. Moreover, we show that the human judgement scores on the generated sentences and perplexity are closely correlated. This leads to an estimated perplexity of 12 for an LM that would be able to pass the human judgement test in the setting we suggested.
Index Terms: language model, generative task, human judgement score, performance gap

1. Introduction

Statistical language modelling is the attempt to estimate the probability distribution over word sequences [1]. Recently, deep learning and recurrent neural networks (RNN) have greatly boosted language modelling research. In [2], by training on a larger corpus and exploiting modern GPUs, the best single model improved the state-of-the-art perplexity from 51 down to 30. Benefited from the rapid development of language modelling, remarkable advances have been witnessed over a lot of natural language processing tasks. In speech recognition, recent models have been reported to perform almost as well as humans [3, 4]. In machine translation, purely neural network-based models can already achieve performance comparable to the traditional phrase-based machine translation system Moses with a small vocabulary [5]. In text generation, coherent novel sentences can be generated by learning a holistic representation of every sentence [6].

Though much effort has been devoted to improve the performance of language models, little research has been done to examine how much exactly this improvement means on our way to an ideal language model. Do we still have a long way to go and all we have achieved is just a small portion of the total distance, or we are already quite close to the destination?

Perplexity [7] is usually used as a metric to measure the quality of language models. Nontheless, direct estimate of the inherent perplexity, which we expect from a perfect language model, in a language is rather difficult. When a language model can accurately estimate the probability of every word, its perplexity becomes the exponential of language entropy. The famous Shannon's game [8], Cover and King's variation [9] tried estimating the entropy of English, but both were in character-level based on pure human subjects. [10] proposed estimating the upper bound for the entropy of English by training a word trigram model, but it was also in character-level and language models have made dramatic breakthroughs since then. WER in speech recognition is shown strongly correlated with perplexity regardless of the type of used LMs [11]. However, language models serve only as an auxiliary component in speech recognition, aiming at distinguishing probable word sequences from all candidates. The signal quality, acoustic recognizer, scoring algorithm and many other factors contribute to the final result. Disentangling the effect of language modelling and thereby estimating the performance of an ideal model is infeasible.

In this paper, we propose estimating the gap between current language models and the perfect model based on a variation of Turing test [12]. Our assumptions are as follows.

- A perfect language model can fully understand the mechanisms beneath a language and assign the word probability in a way similar to the human intuition.
- When asked to complete sentences with provided contexts, by greedily generating the most probable word, the generated sentences should be absolutely plausible.
- The generated sentences, when mixed with normal sentences, should be at least indistinguishable or have even higher scores with respect to plausibility.

On account of these assumptions, we first trained a variety of language models, which range from the basic trigram count-based model to the state-of-the-art multi-layer long short-term memory (LSTM) networks. These models are applied to greedily generate words given a sentence start with fixed 8 words. The generated sentences, together with the original ones, are then randomly shuffled and judged by humans. Every model is assigned a human judgement score for its generated sentences. Our experiment exhibits a strong correlation between the language model performance and the human judgement score. Based on the correlation, we estimated performance of a human-comparable language model by polynomial regression. It is worth mentioning that our estimated performance is just a lower bound, not the exact value, of a perfect language model as every assumption we propose is a necessary but not sufficient condition of a previous one. Our experiment results show that there still exists a discrepancy between the current best model and our derived lower bound, implying more improvement is clearly needed in language modelling.

2. Language Models

We divide the applied language models into 4 classes, ngram& feedforward neural network (FFNN), maximum entropy (ME), RNN with maximum entropy (RNNME) and RNN&LSTM. This section will briefly go through the definitions of them.

http://dx.doi.org/10.21437/Interspeech.2017-729

2.1. Ngram & Feedforward Neural Network

Traditionally, count-based models are used to approximate such a probability based on the frequency information extracted from training corpus[13]. With the Markov assumption[14] and smoothing techniques, probability can be easily estimated from the counting information of n-grams. We trained trigram and 5-gram models with Chen and Goodmans modified Kneser-Ney smoothing. Compared with the original KN smoothing, instead of using a single discounting parameter D, it has three different parameters D1, D2 and D3+ that are applied to n-grams with one, two and three or more counts. Experiments showed that this method outperformed other smoothing techniques [15].

FFNNs are the first neural network architecture introduced to language modelling [16]. Similar to count-based models, FFNNs assume only the most recent $n-1$ preceding words are considered for predicting the next word. Every word is mapped from a one-hot vector to a distributed representation where more semantic information can be encoded.

2.2. Maximum Entropy

Maximum entropy (ME) is another popular model. With features and constraints, it tries to maximize the entropy of the word probability distribution[17]. This model is computationally expensive but more features could be added to improve the performance. ME estimates the word probability as follows

$$P(x|h) = \frac{e^{\sum_i \gamma_i f_i(x,h)}}{Z(h)} \quad (1)$$

f is the feature function, γ is the parameters to be learned and Z is a normalizaiton term for a given history. We apply the ME extension of the srilm toolbox[18] for training. In our case, only up to 5-gram features are used and an $l_1 + l_2^2$ regularization is added to prevent overfitting. For parameter optimization, the Orthant-Wise Limited-memory Quasi-Newton (OWL-QN) method through the libLBFGS library is applied[19].

2.3. RNN with Maximum Entropy

In comparison to FFNNs, the RNN architecture has a recurrent layer to maintain the memory of all past information so that it can learn longer-range dependencies than FFNNs. Maximum entropy can also be incorporated as part of the RNN model (RNNME), which would further reduce the perplexity[20, 21]. The RNN part includes an input layer $x(t)$, a recurrent hidden layer $s(t)$ and an output layer $y(t)$ for each time step t. To speed up, we divides words into 50 classes and the output layer is factorized as a class probability and a word probability[22]. Words are assigned to classes simply proportionally respecting their frequencies. RNNME trains ME as part of the RNN model. Since ME models and the softmax outlayer layer are similar, they can be viewed as a direct connection between input and output layer and the direct parameters can be learned during the training phase of RNN[21]. In our experiment, only bigram and trigram features are used, a 1-billion-size hash[21] is used to map such features in order to reduce the complexity and speed up the training process.

2.4. Vanilla RNN & LSTM

The Vanilla RNN equals the RNNME without ME part. Though theoretically powerful, the training of RNNs is much more complex than FFNNs. Because of the gradient vanishing problem [23], vanilla RNNs fall short of learning weight parameters in a way that long-range dependencies can be captured. To solve this problem, LSTM neural network was proposed in [24] and further extended in [25] and [26]. The resulting structure utilized a gating mechanism to ensure backpropagation of useful information through many time steps. Unlike vanilla RNN, where the effective backpropagation ranges usually up to 6 steps, LSTM can propagate errors for more than 20 steps without losing validity.

We trained the FFNN, RNN and LSTM on multiple GPUs by separating training data into sequences of fixed length. An embedding layer and a projection layer is added to reduce complexity. Batch noise contrastive estimation [27] (FFNN and RNN) and Importance sampling [28] (LSTM) are implemented to scale the large vocabulary size.

3. Human Judgement

After training, 400 sentences, each of which contains at least 16 words are extracted from the test corpus. Sentences containing colons or quotation marks are filtered out beforehand because our experiment shows these punctuations severely interfere with human judgements and people themselves disagree with the use of them. For each sentence, we keep the first 8 words and apply language models to complete it by greedily generating the most probable word until an end token is reached. If a language model has not outputted an end token within 50 words, the generated sentence is regarded as incomplete. In our experiment, only trigram models generated a fair amount of incomplete sentences, the other language models finished almost all the sentences within the limit of 50 words. When incomplete sentences are sent to human judgement, the ellipsis is appended to the end. A snippet of sentence examples is shown in table 1

Table 1: *Example of generated sentences*

Context	Wednesday was the first day at school for
Trigram	the first time in the first time in the ...
5-gram	the first time in the history of the world .
ME	the first time in the history of the world .
RNN	the first time .
RNNME	the city 's history .
LSTM	the school 's president , who has been ...
Original	quadruplets Sarah , Peter , Lucy ...

All the generated sentences, together with the original ones from the test corpus, are then randomly shuffled and judged on the crowdsourcing website CrowdFlower[1]. People are told that the first 8 words of these sentences is the fixed context, the next words are generated either by human or machine. They should try to identify them by assigning a score to each sentence. The score is designed as a 4-level Likert scale [29] with 3 (clearly human), 2 (slightly human), 1 (slightly unhuman) and 0 (clearly unhuman). We adopted 4-level scale instead of the more common 5-level or 7-level in hope of forcing people to demonstrate their preference rather than safely choosing neutral scores. For the experiment participants, only English native people with the highest trust level on Crowdflower are allowed to perform this task. Each sentence is judged by at least 3 different participants and the score supported by most people is adopted. If 3 participants all disagree with each other, more judgements are dynamically collected until at least half of them get a consensus. Trust Levels are accumulated based on their previous performance on

[1] http://www.crowdflower.com/

Crowdflower. In total, 288 participants contributed to the human judgement task.

4. Experiments and Results

4.1. Experimental Setup

The experiments are performed on the 1B Word Benchmark data set [30] collected from English newspapers with about 0.85B words. Sentences containing special tokens are pruned out in advance as mentioned in Section 3. This leads to a corpus containing approximately 0.7B words. We reserve a small subset as the test corpus. The rest training corpus is then randomly split into 100 segments. Since models are trained with different data size, we fix the vocabulary as all tokens existing in the first segment, which contains 158451 words. Other tokens are mapped to a special UNK token.

We trained the trigram, 5-gram and ME on both the first segment and the whole data set, this led to 6 language models in total. A 5-gram 500-1500-600 FFNN was trained only on the whole data, with 500, 1500 and 600 refer to the embedding size, hidden layer size and projection size respectively. RNNME was trained incrementally on the first 1, 3, 6 until 21 segments with the hidden layer size fixed as 600, which formed totally 8 models. Since RNNME can only be run sequentially on CPUs, training it on the whole corpus is too time-consuming, so we stopped at 21 segments, which constitutes around 20% of the whole corpus. A 500-1500-600 vanilla RNN was trained on both the first segment and the whole corpora. 7 LSTM models were trained on the whole corpora, all of which share the same embedding size 512 but differ in the state size, projection size and drop-out rate. A full list of the applied language models is shown in Table 2, where L-2-4096-1024-0.1 denotes a 2 (layer)-4096 (state size)-1024 (projection size)-0.1 (drop-out rate) LSTM language model. This is the largest model we were able to fit into a GPU (Titan X) memory.

4.2. Uncertainty of Data

All data points in our experiment are assumed to be independent with each other. According to central limit theorem, the arithmetic mean of a sufficiently large number of independent random variables is approximately Gaussian distributed[31] with standard deviation as $SD_{\bar{x}} = \frac{\sigma}{\sqrt{n}}$. Here we use sample standard deviation s to approximate the population standard deviation σ. The uncertainty of our measurement can then be defined as $x \pm 2SD\bar{x}$. The factor 2 yields a confidence interval with confidence 95%.

When dealing with functions, uncertainty can be transformed by

$$\Delta f(z_1 ... z_n) = \sqrt{\sum_{i=1}^{N} (\frac{\partial f}{\partial z_i})^2 (\Delta z_i)^2} \tag{2}$$

where z_i is the uncertainty of each parameter [32].

After fitting the data points with polynomial regression, we measure the goodness of fit with adjusted R square [33]

$$R^2_{adjusted} = 1 - \frac{(1 - R^2)(N - 1)}{N - k - 1} \tag{3}$$

where R^2 is the sampled R-square(coefficient of determination), N and K are the number of samples and regressors. Compared with normal R^2, $R^2_{adjusted}$ imposes a penalty as the number of regressors increases to prevent overfitting.

4.3. Metric-based Performance

We test the performance of langauge models with three automatic evaluation metrics: perplexity, mean log rank and percentage of the target word's probability being ranked as the 1st place [34]. All of them are measured on the generated part (the whole sentence excluding the first 8 words) of the original 400 sentences. Table 2 contains all results of applied models. As can be seen, these three metrics are highly consistent with each other. Language models with better scores on one metric usually also perform better on the other two metrics. A larger training data size significantly contributes to the improvement of performance. For RNNME, increasing the size from 1 to 21 segments continuously brings the perplexity from 196 down to 78, with only one fall back in 18 segments. As expected, trigram performs worst among all the language models. FFNN (83.0) performs even worse than 5-gram (73.7) despite consuming much more training time. ME trained on the whole corpus performs surprisingly well (68.8) with only up to 5-gram features included. The best LSTM model takes 2 weeks to converge and achieves a perplexity of 33.6, only slightly worse than the reported record (30.0) in [2].

Table 2: *Performance of Language Models*

Model	PPL	Rank	Top1(%)	Score
Trigram-1	303.2	3.48	19.7	0.11
Trigram-all	112.2	2.75	24.0	0.16
5gram-1	281.0	3.43	21.1	0.24
5-gram-all	73.7	2.43	31.2	0.60
ME-1	286.5	3.46	21.2	0.27
ME-all	68.8	2.40	31.8	0.64
FFNN-all	83.0	2.56	26.3	0.56
RNN-1	211.1	3.28	21.5	0.33
RNN-all	45.7	2.12	31.9	2.08
RNNME-1	196.3	3.21	22.2	0.44
RNNME-3	136.0	2.93	23.7	0.41
RNNME-6	109.7	2.78	24.8	0.43
RNNME-9	107.5	2.76	25.4	0.42
RNNME-12	103.1	2.72	25.0	0.40
RNNME-15	91.3	2.63	26.1	0.48
RNNME-18	106.9	2.76	24.0	0.44
RNNME-21	78.9	2.52	26.9	0.71
L-1-512-512-0.1	63.2	2.41	30.0	1.36
L-1-1024-512-0.1	54.5	2.29	31.8	1.86
L-1-2048-512-0.1	45.3	2.19	33.1	2.39
L-1-8192-2048-0.5	35.9	1.95	33.8	1.54
L-1-8192-2048-0	37.5	1.97	34.8	2.60
L-2-2048-512-0.1	39.8	2.09	35.0	2.91
L-2-4096-1024-0.1	33.6	1.94	36.2	3.51
Human (estimated)	12.0	1.14	40.5	7.95

4.4. Human Judgement Score

Let n_i denote the number of sentences being judged with score i. We noticed n_1 (slightly unhuman) or n_2 (slightly human) are quite similar over all language models. After manually examining these sentences, we found they were ambiguous and quite difficult to be clearly judged. In consequence, we define the human judgement score as the ratio of n_3 (clearly human) to n_0 (clearly unhuman). We believe these two values are more reliable and good language models should achieve higher scores by successfully fooling human judgers more often.

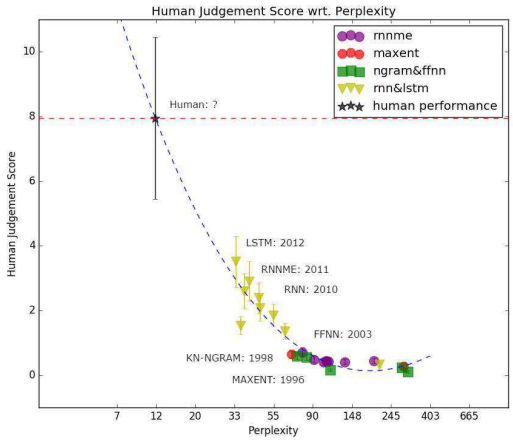

Figure 1: *Human Judgement Score wrt. Perplexity Adjusted R-square: 0.953*

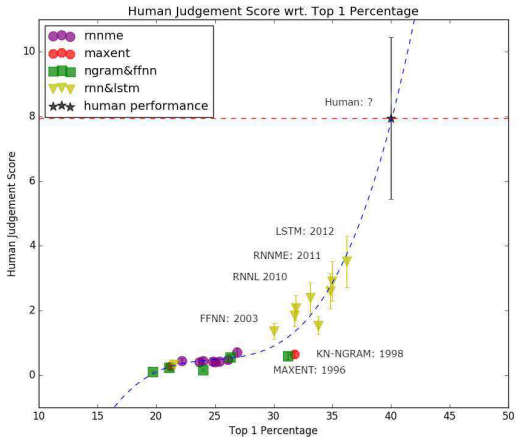

Figure 2: *Human Judgement Score wrt. Top 1 Percentage Adjusted R-square: 0.955*

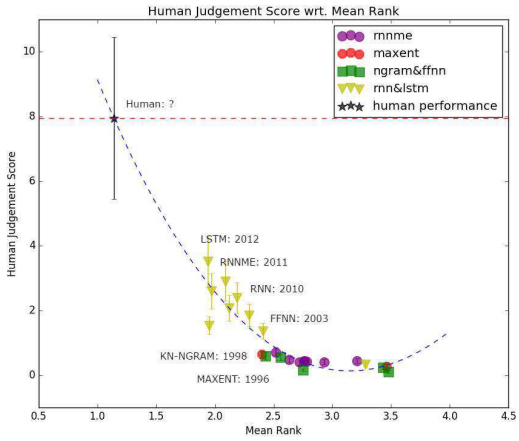

Figure 3: *Human Judgement Score wrt. Mean Rank Adjusted R-square: 0.934*

The last column of Table 2 shows the human judgement scores. We can see models with better metric scores normally, though not always, have better chances of successfully fooling humans. One major exception comes from the trigram model trained on full corpus, who achieved a very low human judgement score (0.16) with a mediate perplexity (112.22). Original sentences received a human judgement score of 7.95, which is a big lead over all language models, even to the best LSTM model (3.51). It is worth noting that even for original sentences, quite a few of them are judged as clearly unhuman since they are novel, contain professional terms or noise distractions resulting from the inappropriate tokenization. The real human performance should be higher, which further explains why our estimation stands only for a lower bound.

Figure 1-3 pictorize the correlation between human judgement score and metric-based performance. The least square polynomial fit[35] with degree 3 is applied to fit all the data points (blue curve). The horizontal dash line stands for the human judgement score from original sentences. It can be viewed as the lower bound of a perfect language model. The intersecting point of the dash line and the fitted curve could then be used to approximate the metric-based performance of such language model. We further labelled the year number every model is first applied so that the evolving development of language modelling can be easily seen. Perplexity is scaled logarithmically for curve fitting while the other two metrics stay the same. As shown in the figures, all curves fit the data points pretty well. The adjusted R square is more than 0.9 for all three automatic measurement metrics. The fitted curve suggests an estimated perplexity 12 for a human-comparable language model in our task. The estimated value for mean log rank and top 1 percentage is 1.14 and 40.5% respectively. Interestingly, given the average word length of 5.5, Shannon estimated the lower bound on human-level word perplexity as $2^{0.648\times5.5} = 11.8$[8], which is consistent with our result.

5. Conclusion and Future Work

This work attempts to detect and measure the gap between current language models and human performance. In our sentence completion task, by defining a metric for human judgement score and taking the original sentences as a target, this gap can be estimated. Our findings show that neural networks do bring a remarkable improvement to language models, whether for metric-based performance or human judgement scores. However, current language models are still far from perfect, more research is needed in this field.

This experiment is a preliminary study and a few shortcomings exist. The human judgement task is rather unsupervised and the results are subject to more uncertainty. In the future we will collect more experiments and control the human judgement in a more fine-grained way.

6. Acknowledgements

We thank Andrea Fisher, Thomas Trost and all the anonymous reviewers for useful discussions and suggestions. Xiaoyu Shen is supported by computer science graduate school of Saarland University. This work is funded by the DFG collaborative research center SFB 1102.

7. References

[1] P. F. Brown, J. Cocke, S. A. D. Pietra, V. J. D. Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin, "A statistical approach to machine translation," *Computational linguistics*, vol. 16, no. 2, pp. 79–85, 1990.

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

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

[4] G. Saon, G. Kurata, T. Sercu, K. Audhkhasi, S. Thomas, D. Dimitriadis, X. Cui, B. Ramabhadran, M. Picheny, L.-L. Lim *et al.*, "English conversational telephone speech recognition by humans and machines," *arXiv preprint arXiv:1703.02136*, 2017.

[5] F. Meng, Z. Lu, Z. Tu, H. Li, and Q. Liu, "A deep memory-based architecture for sequence-to-sequence learning," *ICLR*, 2016.

[6] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S. Bengio, "Generating sentences from a continuous space," *CoNLL*, 2016.

[7] L. R. Bahl, F. Jelinek, and R. L. Mercer, "A maximum likelihood approach to continuous speech recognition," *IEEE transactions on pattern analysis and machine intelligence*, no. 2, pp. 179–190, 1983.

[8] C. E. Shannon, "Prediction and entropy of printed english," *Bell Labs Technical Journal*, vol. 30, no. 1, pp. 50–64, 1951.

[9] T. Cover and R. King, "A convergent gambling estimate of the entropy of english," *IEEE Transactions on Information Theory*, vol. 24, no. 4, pp. 413–421, 1978.

[10] P. F. Brown, V. J. D. Pietra, R. L. Mercer, S. A. D. Pietra, and J. C. Lai, "An estimate of an upper bound for the entropy of english," *Computational Linguistics*, vol. 18, no. 1, pp. 31–40, 1992.

[11] 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 (TASLP)*, vol. 23, no. 3, pp. 517–529, 2015.

[12] A. M. Turing, "Computing machinery and intelligence," *Mind*, vol. 59, no. 236, pp. 433–460, 1950.

[13] S. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer," *IEEE transactions on acoustics, speech, and signal processing*, vol. 35, no. 3, pp. 400–401, 1987.

[14] H. Ney, S. Martin, and F. Wessel, "Statistical language modeling using leaving-one-out," in *Corpus-based methods in Language and Speech processing*. Springer, 1997, pp. 174–207.

[15] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," in *Proceedings of the 34th annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 1996, pp. 310–318.

[16] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *journal of machine learning research*, vol. 3, no. Feb, pp. 1137–1155, 2003.

[17] R. Rosenfeld, "A maximum entropy approach to adaptive statistical language modeling," 1996.

[18] A. Stolcke *et al.*, "Srilm-an extensible language modeling toolkit." in *Interspeech*, vol. 2002, 2002, p. 2002.

[19] T. Alumäe and M. Kurimo, "Efficient estimation of maximum entropy language models with n-gram features: an srilm extension." in *INTERSPEECH*, 2010, pp. 1820–1823.

[20] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur, "Recurrent neural network based language model." in *Interspeech*, vol. 2, 2010, p. 3.

[21] T. Mikolov, A. Deoras, D. Povey, L. Burget, and J. Černockỳ, "Strategies for training large scale neural network language models," in *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*. IEEE, 2011, pp. 196–201.

[22] T. Mikolov, S. Kombrink, L. Burget, J. Černockỳ, and S. Khudanpur, "Extensions of recurrent neural network language model," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2011, pp. 5528–5531.

[23] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

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

[25] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," *Neural computation*, vol. 12, no. 10, pp. 2451–2471, 2000.

[26] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning precise timing with lstm recurrent networks," *Journal of machine learning research*, vol. 3, no. Aug, pp. 115–143, 2002.

[27] M. Gutmann and A. Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models." in *AISTATS*, vol. 1, no. 2, 2010, p. 6.

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

[29] R. Likert, "A technique for the measurement of attitudes." *Archives of psychology*, 1932.

[30] C. Chelba, T. Mikolov, M. Schuster, Q. Ge, T. Brants, P. Koehn, and T. Robinson, "One billion word benchmark for measuring progress in statistical language modeling," *arXiv preprint arXiv:1312.3005*, 2013.

[31] I. Barany and V. Vu, "Central limit theorems for gaussian polytopes," *The Annals of Probability*, pp. 1593–1621, 2007.

[32] D. Klakow and J. Peters, "Testing the correlation of word error rate and perplexity," *Speech Communication*, vol. 38, no. 1, pp. 19–28, 2002.

[33] A. C. Cameron and F. A. Windmeijer, "R-squared measures for count data regression models with applications to health-care utilization," *Journal of Business & Economic Statistics*, vol. 14, no. 2, pp. 209–220, 1996.

[34] P. Clarkson, T. Robinson *et al.*, "Towards improved language model evaluation measures." in *EUROSPEECH*, 1999.

[35] J. Gergonne, "The application of the method of least squares to the interpolation of sequences," *Historia Mathematica*, vol. 1, no. 4, pp. 439–447, 1974.