An Analysis of the Ability of Statistical Language Models to Capture the Structural Properties of Language

Aneiss Ghodsi and **John DeNero**
Computer Science Division
University of California, Berkeley
{aneiss, denero}@berkeley.edu

Abstract

We investigate the characteristics and quantifiable predispositions of both n-gram and recurrent neural language models in the framework of language generation. In modern applications, neural models have been widely adopted, as they have empirically provided better results. However, there is a lack of deep analysis of the models and how they relate to real language and its structural properties. We attempt to perform such an investigation by analyzing corpora generated by sampling from the models. The results are compared to each other and to the results of the same analysis applied to the training corpus. We carried out these experiments on varieties of Kneser-Ney smoothed n-gram models and basic recurrent neural language models. Our results reveal a number of distinctive characteristics of each model, and offer insights into their behavior. Our general approach also provides a framework in which to perform further analysis of language models.

1 Introduction

Statistical language modelling is critical to natural language processing and many generation systems. In recent years use has shifted from the previously prevalent n-gram model to the recurrent neural network paradigm that now dominates in most applications. Researchers have long sought to find the best language modeling solutions for particular applications, but it is important to understand the behavior of language models in a more generalizable way. This is advantageous both in developing language models and in applying them practically. Whether in tasks where statistical models are used to directly generate language or in cases where the model is used for ranking for surface realization, the statistical predispositions of the language model will be reflected in the results. In this paper we compare the behavior of n-gram models and Recurrent Neural Network Language Models (RNNLMs) with regard to properties of their generated language.

We use the SRILM toolkit for training and generating from n-gram models (Stolcke and others, 2002). Our n-gram model is a modified Kneser-Ney back-off interpolative model, unless otherwise stated (Chen and Goodman, 1999). We use Tomas Mikolov's implementation of an RNNLM, available at $rnnlm.org$ (Mikolov et al., 2010). This model has a single hidden recurrent layer, and three defining parameters: class size, hidden layer size, and backpropagation through time (BPTT) steps. Classes are used to factor the vocabulary mappings to improve performance, by predicting a distribution over classes of words and then over words in a class (Mikolov et al., 2011). BPTT steps determine how many times the recurrent layer of the network is unwrapped for training. Unless otherwise mentioned all neural models have class of 100 and use four BPTT steps. We use the Penn Tree Bank (PTB), constructed from articles from the Wall Street Journal, as our primary training corpus, with the standard training split of 42068 sentences (Marcus et al., 1993). Correspondingly, our generated language corpora also contain 42068 sentences. Novel sentences are easily sampled from trained language models by prompting with a start of sentence token,

Proceedings of The 9th International Natural Language Generation conference, pages 227–231,
Edinburgh, UK, September 5-8 2016.

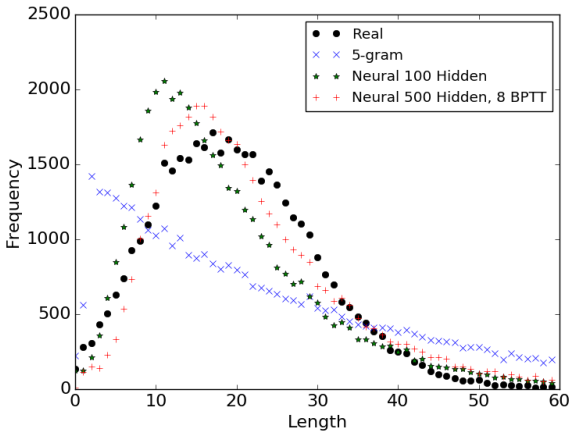

Figure 1: Sentence Length Distributions

sampling from the predicted distribution, using the result as context, and repeating until an end of sentence token is encountered.

We select three primary metrics with which to evaluate the various resulting corpora. The first is the distribution of sentence lengths. Sentence length is compared visually and through the sum of error as compared to the length distribution from the training corpus. The second metric is word frequency. Word frequency is analyzed by fitting a Zipfian distribution (Kingsley, 1932), and comparing between the distributions for each model. Third is pronoun frequency relative to distance from the start of a sentence. This was selected as a metric due to the fact that one-word pronouns are a small class fairly easily identifiable regardless of context (though there are a few that can be other parts of speech), partly avoiding the ambiguities and challenges that follow from part of speech taggers. This is especially useful in a corpus with a restricted vocabulary resulting in the replacement of uncommon tokens with a single token, such as the PTB, and with generated language that is not always semantically sound. These experiments were repeated multiple times with small variations, ensuring the key patterns in the results were not a product of chance.

Through these three metrics we seek to develop some insights into the behavior of standard stochastic models in language generation.

2 Sentence Lengths

The natural expectation is that a recurrent neural model, with its superior ability to 'remember' complex context, would vastly outperform even fairly high order n-gram models in modeling sentence length. While in training errors are only propagated as far back as truncated backpropagation is executed (the BPTT steps hyperparameter), the power of the recurrent layer seems to exceed its apparent depth during training, taking advantage of the ability of recurrent memory to retain subtle contextual information. As seen in Figure 1, even the four BPTT step model performs fairly well. Contrastingly, n-gram models perform very poorly. Table 1 notes the sum of the absolute errors across the full range of models. N-gram models exhibit no improvement with increasing order. In neural production, however, we see substantial improvements with increasing network complexity; specifically, with an increase in the size of the hidden layer and the number of BPTT steps. However, the neural models tested here are unable to replicate the precise shape of the distribution. All models overestimate the incidence of very long sentences.

Corpus	Sum of Error
Trigram	27736
5-gram	29694
Neural Hidden 100	19237
Neural Hidden 500	14132

Table 1: Sum of errors for sentence lengths, including normalized over total sentences.

3 Vocabulary Distribution

Zipf's Law states that, for N unique words and s as the defining parameter, the frequency of a word with rank k is given by the following (Kingsley, 1932):

$$f(k; s, N) = \frac{1/k^s}{\sum_{n=1}^{N}(1/n^s)}$$

There are two aspects of evaluation for word frequencies: First, the difference between the Zipf parameters of distributions fitted to various text sources; second, the error on the data set to which a Zipfian distribution is fitted, indicating how closely the data follows a distribution known to match natural language production.

As shown in table 2, n-gram smoothing techniques have a significant effect on the accuracy of the generated Zipf distribution. As an n-gram model approaches being a simple unigram model, it should

Corpus	s	LL
Real	0.99193	-104598
Unigram 0-Discount	0.99293	-104416
Trigram 0-Discounts	0.98348	-103967
Trigram Discounts	0.97921	-104049
Trigram Back-Off Only	0.93515	-102532
Neural Hidden 100	0.98707	-104332
Neural Hidden 500	0.99735	-104655

Table 2: Zipf fit parameters s with Log-Likelihood.

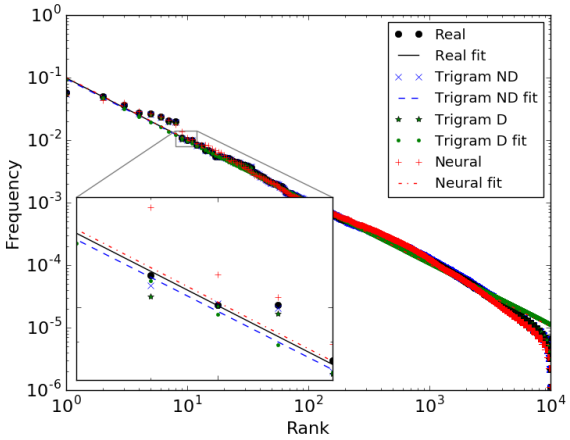

Figure 2: Zipf Distributions

approach the same distribution as real language, due to the fact that a unigram model behaves like direct sampling of words from the training corpus. Thus it is intuitive that the interpolated models, in which unigram information always influences generation, performs better than a simple Kneser-Ney back-off model. Critically, on any configuration, non-zero discounting seems to worsen the distribution. As discounting is a method by which probability is held out to distribute amongst less likely or unseen sequences or tokens, it is reasonable that it would affect the distribution. Figure 2 shows the distributions from a selection of models on a log-log scale, with the trigram model with non-zero discounts (D) and with zero discounts (ND).

4 Pronoun Frequency with Depth

Finally, we observe the probability of encountering a pronoun at an index according to the following expression:

$$\frac{\sum_{s \in sentences} s[i] \in pronouns}{\sum_{s \in sentences} len(s) \geq i+1}$$

We find that there is a spike in the probability

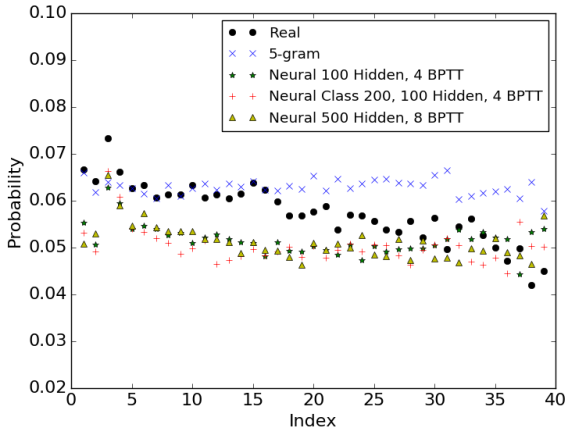

Figure 3: Pronoun Probability with Position

of encountering a pronoun as the first word in a sentence, to approximately 0.15, an intuitive result given the prevalence of pronouns as sentence subjects. All models captured this fairly well. More interestingly, the probability of generating or observing a pronoun decreases with depth into a sentence. This phenomenon is clearly observable in the training set, with a fairly linear slope, which we calculate to be approximately -6.9×10^{-4} when restricted to the first twenty indices, excluding zero, due to the low number of samples at further positions in the sentence causing noise to dominate. In order to verify this result, the slope was calculated by sampling 20 subsets of sentences and averaging the slope across subsets. A comparable slope exists even when the domain is restricted to a set of sentences all of the same length (for example fourteen word sentences). This means the phenomenon is not an artifact resulting from the distribution of sentence lengths and a relationship between pronoun occurrences and sentence endings.

Neither class of model does particularly well at capturing this property, as can been seen in Figure 3. N-gram models were able to effectively capture the pronoun probability at the first word, as expected given the model should more or less reproduce the first-word distribution of the training data. They also appear to reflect the probabilities at the next several indices, but as with sentence length, they fail at any significant sentence depth regardless of n-gram order. The distribution in the n-gram generated language becomes approximately uniform. Neural models seem to capture some negative slope in the

first ten to twenty words, but with depressed overall probabilities, and a loss of the pattern after a certain depth. Figure 3 also shows that increasing RNNLM complexity, whether in class, hidden size, or number of BPTT steps, does little to change the performance of the model in this metric.

This is concerning regarding the ability of this form of RNNLM to capture certain complex structural patterns, and indicates that the structure is inherently limited. It may be that a model with a Long-Short Term Memory unit (LSTM) as the recurrent component could perform better, with its superior ability to capture longer term contextual dependencies (Hochreiter and Schmidhuber, 1997). Indeed, LSTMs have become highly popular in many sequential learning tasks. However, given that these same basic RNNLMs performed well in the position-dependent sentence length metric, this result is disappointing.

5 Future Work

There are a number of clear steps to expand on this line of research, including experimenting with a greater variety of language models. In particular, a recurrent model with a Long Short-Term Memory unit (LSTM) might improve on the weaknesses of the simple RNNLM demonstrated here.

Additionally, further diversification of data sets is important to learning about patterns as they differ or remain consistent across sources. For example, preliminary analysis of the more stylistically diverse Brown corpus (Francis, 1964) indicates that the pronoun trend observed in the PTB may not be present in other domains, at least not as clearly. Additionally to profiling models on specific text genres, the experiments must be recreated on a far more sizeable dataset, such as the Wikipedia text corpus.

Finally, the introduction of new metrics to the language model analysis could add further value. Automatic tagging and parsing systems are likely to suffer from significant inaccuracy on the often flawed text produced by stochastic models; however, the results from applying such systems could prove informative about language model quality, as a model is not effectively capturing structural and semantic properties of language if parsing and tagging results statistics are not comparable to those of real language. Statistical analysis of parsing results would help expand the quantitative portrait of a language model.

6 Conclusion

Our work characterizes some key structural properties of language generated from two common statistical models. The results presented here verify many of the expectations regarding the behavior of n-gram and RNN techniques, and also introduce some new observations. RNNs have a structural capacity largely missing from n-gram models, which is particularly apparent in sentence length distributions. The recurrent model used here, however, struggled in reproducing the more complex pattern represented by the pronoun distribution over position. The results of the Zipfian distribution analysis indicate that neural networks with reasonable complexity are capable of approaching the correct vocabulary distribution, and competing favorably with the most vocabulary-optimized n-gram models. We found some interesting phenomena where smoothing, especially with high order n-gram models, flattened the Zipf distribution. At the very least we see that basic RNNLMs exhibit no real weaknesses next to n-gram models, beyond training time.

Overall, the methods we present here comprise an approach to language model analysis that is more independent from specific applications than previous reviews of language model performance. By selecting structural properties of language that are measurable and ideally equally valid on real and sampled language, it is possible to characterize language models and examine their learning capacities and predispositions in generation and ranking. Future avenues of investigation in line with this paradigm can provide more detailed portraits and serve as guidance both in the selection of models for applications and for further developments in statistical language modeling.

Acknowledgments

We would like to thank the members of the NLP group at the University of California, Berkeley for their contributions to discussions, as well as the two anonymous reviewers of this paper for their suggestions.

References

Stanley F Chen and Joshua Goodman. 1999. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4):359–393.

Winthrop Nelson Francis. 1964. A standard sample of present-day english for use with digital computers.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Zipf George Kingsley. 1932. Selective studies and the principle of relative frequency in language.

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *INTERSPEECH*, volume 2, page 3.

Tomáš Mikolov, Stefan Kombrink, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2011. Extensions of recurrent neural network language model. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5528–5531. IEEE.

Andreas Stolcke et al. 2002. Srilm-an extensible language modeling toolkit. In *INTERSPEECH*, volume 2002, page 2002.