Smoothing a Tera-word Language Model

Deniz Yuret
Koç University
dyuret@ku.edu.tr

Abstract

Frequency counts from very large corpora, such as the Web 1T dataset, have recently become available for language modeling. Omission of low frequency n-gram counts is a practical necessity for datasets of this size. Naive implementations of standard smoothing methods do not realize the full potential of such large datasets with missing counts. In this paper I present a new smoothing algorithm that combines the Dirichlet prior form of (Mackay and Peto, 1995) with the modified back-off estimates of (Kneser and Ney, 1995) that leads to a 31% perplexity reduction on the Brown corpus compared to a baseline implementation of Kneser-Ney discounting.

1 Introduction

Language models, i.e. models that assign probabilities to sequences of words, have been proven useful in a variety of applications including speech recognition and machine translation (Bahl et al., 1983; Brown et al., 1990). More recently, good results on lexical substitution and word sense disambiguation using language models have also been reported (Yuret, 2007).

The recently introduced Web 1T 5-gram dataset (Brants and Franz, 2006) contains the counts of word sequences up to length five in a 10^{12} word corpus derived from publicly accessible Web pages. As this corpus is several orders of magnitude larger than the ones used in previous language modeling studies, it holds the promise to provide more accurate domain independent probability estimates. However, naive application of the well-known smoothing methods do not realize the full potential of this dataset.

In this paper I present experiments with modifications and combinations of various smoothing methods using the Web 1T dataset for model building and the Brown corpus for evaluation. I describe a new smoothing method, Dirichlet-Kneser-Ney (DKN), that combines the Bayesian intuition of MacKay and Peto (1995) and the improved back-off estimation of Kneser and Ney (1995) and gives significantly better results than the baseline Kneser-Ney discounting.

The next section describes the general structure of n-gram models and smoothing. Section 3 describes the data sets and the experimental methodology used. Section 4 presents experiments with adaptations of various smoothing methods. Section 5 describes the new algorithm.

2 N-gram Models and Smoothing

N-gram models are the most commonly used language modeling tools. They estimate the probability of each word using the context made up of the previous $n-1$ words. Let abc represent an n-gram where a is the first word, c is the last word, and b represents ***zero or more words*** in between. One way to estimate $\Pr(c|ab)$ is to look at the number of times word c has followed the previous $n-1$ words ab,

$$\Pr(c|ab) = \frac{C(abc)}{C(ab)} \quad (1)$$

where $C(x)$ denotes the number of times x has been observed in the training corpus. This is the maximum likelihood (ML) estimate. Unfortunately it

Proceedings of ACL-08: HLT, Short Papers (Companion Volume), pages 141–144,
Columbus, Ohio, USA, June 2008.

does not work very well because it assigns zero probability to n-grams that have not been observed in the training corpus. To avoid the zero probabilities, we take some probability mass from the observed n-grams and distribute it to unobserved n-grams. Such redistribution is known as smoothing or discounting.

Most existing smoothing methods can be expressed in one of the following two forms:

$$\Pr(c|ab) = \alpha(c|ab) + \gamma(ab)\Pr(c|b) \quad (2)$$

$$\Pr(c|ab) = \begin{cases} \beta(c|ab) & \text{if } C(abc) > 0 \\ \gamma(ab)\Pr(c|b) & \text{otherwise} \end{cases} \quad (3)$$

Equation 2 describes the so-called interpolated models and Equation 3 describes the back-off models. The highest order distributions $\alpha(c|ab)$ and $\beta(c|ab)$ are typically discounted to be less than the ML estimate so we have some leftover probability for the c words unseen in the context ab. Different methods mainly differ on how they discount the ML estimate. The back-off weights $\gamma(ab)$ are computed to make sure the probabilities are normalized. The interpolated models always incorporate the lower order distribution $\Pr(c|b)$ whereas the back-off models consider it only when the n-gram abc has not been observed in the training data.

3 Data and Method

All the models in this paper are interpolated models built using the counts obtained from the Web 1T dataset and evaluated on the million word Brown corpus using cross entropy (bits per token). The lowest order model is taken to be the word frequencies in the Web 1T corpus. The Brown corpus was retokenized to match the tokenization style of the Web 1T dataset resulting in 1,186,262 tokens in 52,108 sentences. The Web 1T dataset has a 13 million word vocabulary consisting of words that appear 100 times or more in its corpus. 769 sentences in Brown that contained words outside this vocabulary were eliminated leaving 1,162,052 tokens in 51,339 sentences. Capitalization and punctuation were left intact. The n-gram patterns of the Brown corpus were extracted and the necessary counts were collected from the Web 1T dataset in one pass. The end-of-sentence tags were not included in the entropy calculation. For parameter optimization, numerical optimization was performed on a 1,000 sentence random sample of Brown.

4 Experiments

In this section, I describe several smoothing methods and give their performance on the Brown corpus. Each subsection describes a single idea and its impact on the performance. All methods use interpolated models expressed by $\alpha(c|ab)$ and $\gamma(ab)$ based on Equation 2. The Web 1T dataset does not include n-grams with counts less than 40, and I note the specific implementation decisions due to the missing counts where appropriate.

4.1 Absolute Discounting

Absolute discounting subtracts a fixed constant D from each nonzero count to allocate probability for unseen words. A different D constant is chosen for each n-gram order. Note that in the original study, D is taken to be between 0 and 1, but because the Web 1T dataset does not include n-grams with counts less than 40, the optimized D constants in our case range from 0 to 40. The interpolated form is:

$$\alpha(c|ab) = \frac{\max(0, C(abc) - D)}{C(ab*)} \quad (4)$$

$$\gamma(ab) = \frac{N(ab*)D}{C(ab*)}$$

The $*$ represents a wildcard matching any word and $C(ab*)$ is the total count of n-grams that start with the $n-1$ words ab. If we had complete counts, we would have $C(ab*) = \sum_c C(abc) = C(ab)$. However because of the missing counts in general $C(ab*) \leq C(ab)$ and we need to use the former for proper normalization. $N(ab*)$ denotes the number of distinct words following ab in the training data. Absolute discounting achieves its best performance with a 3-gram model and gives 8.53 bits of cross entropy on the Brown corpus.

4.2 Kneser-Ney

Kneser-Ney discounting (Kneser and Ney, 1995) has been reported as the best performing smoothing method in several comparative studies (Chen and Goodman, 1999; Goodman, 2001). The $\alpha(c|ab)$ and $\gamma(ab)$ expressions are identical to absolute discounting (Equation 4) for the highest order n-grams.

However, a modified estimate is used for lower order n-grams used for back-off. The interpolated form is:

$$\begin{aligned} \Pr(c|ab) &= \alpha(c|ab) + \gamma(ab)\Pr'(c|b) \quad (5) \\ \Pr'(c|ab) &= \alpha'(c|ab) + \gamma'(ab)\Pr'(c|b) \end{aligned}$$

Specifically, the modified estimate $\Pr'(c|b)$ for a lower order n-gram is taken to be proportional to the number of unique words that precede the n-gram in the training data. The α' and γ' expressions for the modified lower order distributions are:

$$\begin{aligned} \alpha'(c|b) &= \frac{\max(0, N(*bc) - D)}{N(*b*)} \quad (6) \\ \gamma'(b) &= \frac{R(*b*)D}{N(*b*)} \end{aligned}$$

where $R(*b*) = |c : N(*bc) > 0|$ denotes the number of distinct words observed on the right hand side of the $*b*$ pattern. A different D constant is chosen for each n-gram order. The lowest order model is taken to be $\Pr(c) = N(*c)/N(**)$. The best results for Kneser-Ney are achieved with a 4-gram model and its performance on Brown is 8.40 bits.

4.3 Correcting for Missing Counts

Kneser-Ney takes the back-off probability of a lower order n-gram to be proportional to the number of unique words that precede the n-gram in the training data. Unfortunately this number is not exactly equal to the $N(*bc)$ value given in the Web 1T dataset because the dataset does not include low count abc n-grams. To correct for the missing counts I used the following modified estimates:

$$\begin{aligned} N'(*bc) &= N(*bc) + \delta(C(bc) - C(*bc)) \\ N'(*b*) &= N(*b*) + \delta(C(b*) - C(*b*)) \end{aligned}$$

The difference between $C(bc)$ and $C(*bc)$ is due to the words preceding bc less than 40 times. We can estimate their number to be a fraction of this difference. δ is an estimate of the type token ratio of these low count words. Its valid range is between 1/40 and 1, and it can be optimized along with the other parameters. The reader can confirm that $\sum_c N'(*bc) = N'(*b*)$ and $|c : N'(*bc) > 0| = N(b*)$. The expression for the Kneser-Ney back-off estimate becomes

$$\begin{aligned} \alpha'(c|b) &= \frac{\max(0, N'(*bc) - D)}{N'(*b*)} \quad (7) \\ \gamma'(b) &= \frac{N(b*)D}{N'(*b*)} \end{aligned}$$

Using the corrected N' counts instead of the plain N counts achieves its best performance with a 4-gram model and gives 8.23 bits on Brown.

4.4 Dirichlet Form

MacKay and Peto (1995) show that based on Dirichlet priors a reasonable form for a smoothed distribution can be expressed as

$$\begin{aligned} \alpha(c|ab) &= \frac{C(abc)}{C(ab*) + A} \quad (8) \\ \gamma(ab) &= \frac{A}{C(ab*) + A} \end{aligned}$$

The parameter A can be interpreted as the extra counts added to the given distribution and these extra counts are distributed as the lower order model. Chen and Goodman (1996) suggest that these extra counts should be proportional to the number of words with exactly one count in the given context based on the Good-Turing estimate. The Web 1T dataset does not include one-count n-grams. A reasonable alternative is to take A to be proportional to the missing count due to low-count n-grams: $C(ab) - C(ab*)$.

$$A(ab) = \max(1, K(C(ab) - C(ab*)))$$

A different K constant is chosen for each n-gram order. Using this formulation as an interpolated 5-gram language model gives a cross entropy of 8.05 bits on Brown.

4.5 Dirichlet with KN Back-Off

Using a modified back-off distribution for lower order n-grams gave us a big boost in the baseline results from 8.53 bits for absolute discounting to 8.23 bits for Kneser-Ney. The same idea can be applied to the missing-count estimate. We can use Equation 8 for the highest order n-grams and Equation 7 for lower order n-grams used for back-off. Such a 5-gram model gives a cross entropy of 7.96 bits on the Brown corpus.

5 A New Smoothing Method: DKN

In this section, I describe a new smoothing method that combines the Dirichlet form of MacKay and

Peto (1995) and the modified back-off distribution of Kneser and Ney (1995). We will call this new method Dirichlet-Kneser-Ney, or DKN for short. The important idea in Kneser-Ney is to let the probability of a back-off n-gram be proportional to the number of unique words that precede it. However we do not need to use the absolute discount form for the estimates. We can use the Dirichlet prior form for the lower order back-off distributions as well as the highest order distribution. The extra counts A in the Dirichlet form are taken to be proportional to the missing counts, and the coefficient of proportionality K is optimized for each n-gram order. Where complete counts are available, A should be taken to be proportional to the number of one-count n-grams instead. This smoothing method with a 5-gram model gives a cross entropy of 7.86 bits on the Brown corpus achieving a perplexity reduction of 31% compared to the naive implementation of Kneser-Ney.

The relevant equations are repeated below for the reader's convenience.

$$\begin{aligned}
\Pr(c|ab) &= \alpha(c|ab) + \gamma(ab)\Pr'(c|b) \\
\Pr'(c|ab) &= \alpha'(c|ab) + \gamma'(ab)\Pr'(c|b) \\
\alpha(c|b) &= \frac{C(bc)}{C(b*) + A(b)} \\
\gamma(b) &= \frac{A(b)}{C(b*) + A(b)} \\
\alpha'(c|b) &= \frac{N'(*bc)}{N'(*b*) + A(b)} \\
\gamma'(b) &= \frac{A(b)}{N'(*b*) + A(b)} \\
A(b) &= \max(1, K(C(b) - C(b*))) \\
&\text{or} \quad \max(1, K|c : C(bc) = 1|)
\end{aligned}$$

6 Summary and Discussion

Frequency counts based on very large corpora can provide accurate domain independent probability estimates for language modeling. I presented adaptations of several smoothing methods that can properly handle the missing counts that may exist in such datasets. I described a new smoothing method, DKN, combining the Bayesian intuition of MacKay and Peto (1995) and the modified back-off distribution of Kneser and Ney (1995) which achieves a significant perplexity reduction compared to a naive implementation of Kneser-Ney smoothing. This is a surprizing result because Chen and Goodman (1999) partly attribute the performance of Kneser-Ney to the use of absolute discounting. The relationship between Kneser-Ney smoothing to the Bayesian approach have been explored in (Goldwater et al., 2006; Teh, 2006) using Pitman-Yor processes. These models still suggest discount-based interpolation with type frequencies whereas DKN uses Dirichlet smoothing throughout. The conditions under which the Dirichlet form is superior is a topic for future research.

References

Lalit R. Bahl, Frederick Jelinek, and Robert L. Mercer. 1983. A maximum likelihood approach to continuous speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):179–190.

Thorsten Brants and Alex Franz. 2006. Web 1T 5-gram version 1. Linguistic Data Consortium, Philadelphia. LDC2006T13.

Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Frederick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85.

Stanley F. Chen and Joshua Goodman. 1996. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th Annual Meeting of the ACL*.

Stanley F. Chen and Joshua Goodman. 1999. An empirical study of smoothing techniques for language modeling. *Computer Speech and Language*.

S. Goldwater, T.L. Griffiths, and M. Johnson. 2006. Interpolating between types and tokens by estimating power-law generators. In *Advances in Neural Information Processing Systems*, volume 18. MIT Press.

Joshua Goodman. 2001. A bit of progress in language modeling. *Computer Speech and Language*.

R. Kneser and H. Ney. 1995. Improved backing-off for m-gram language modeling. In *International Conference on Acoustics, Speech, and Signal Processing*.

David J. C. Mackay and Linda C. Bauman Peto. 1995. A hierarchical Dirichlet language model. *Natural Language Engineering*, 1(3):1–19.

Y.W. Teh. 2006. A hierarchical Bayesian language model based on Pitman-Yor processes. In *Proceedings of the ACL*, pages 985–992.

Deniz Yuret. 2007. KU: Word sense disambiguation by substitution. In *SemEval-2007: 4th International Workshop on Semantic Evaluations*.