Improved Smoothing for N-gram Language Models Based on Ordinary Counts

Robert C. Moore **Chris Quirk**
Microsoft Research
Redmond, WA 98052, USA
{bobmoore,chrisq}@microsoft.com

Abstract

Kneser-Ney (1995) smoothing and its variants are generally recognized as having the best perplexity of any known method for estimating N-gram language models. Kneser-Ney smoothing, however, requires nonstandard N-gram counts for the lower-order models used to smooth the highest-order model. For some applications, this makes Kneser-Ney smoothing inappropriate or inconvenient. In this paper, we introduce a new smoothing method based on ordinary counts that outperforms all of the previous ordinary-count methods we have tested, with the new method eliminating most of the gap between Kneser-Ney and those methods.

1 Introduction

Statistical language models are potentially useful for any language technology task that produces natural-language text as a final (or intermediate) output. In particular, they are extensively used in speech recognition and machine translation. Despite the criticism that they ignore the structure of natural language, simple N-gram models, which estimate the probability of each word in a text string based on the $N-1$ preceding words, remain the most widely used type of model.

The simplest possible N-gram model is the maximum likelihood estimate (MLE), which takes the probability of a word w_n, given the preceding context $w_1 \ldots w_{n-1}$, to be the ratio of the number of occurrences in a training corpus of the N-gram $w_1 \ldots w_n$ to the total number of occurrences of any word in the same context:

$$p(w_n|w_1 \ldots w_{n-1}) = \frac{C(w_1 \ldots w_n)}{\sum_{w'} C(w_1 \ldots w_{n-1} w')}$$

One obvious problem with this method is that it assigns a probability of zero to any N-gram that is not observed in the training corpus; hence, numerous smoothing methods have been invented that reduce the probabilities assigned to some or all observed N-grams, to provide a non-zero probability for N-grams not observed in the training corpus.

The best methods for smoothing N-gram language models all use a hierarchy of lower-order models to smooth the highest-order model. Thus, if $w_1 w_2 w_3 w_4 w_5$ was not observed in the training corpus, $p(w_5|w_1 w_2 w_3 w_4)$ is estimated based on $p(w_5|w_2 w_3 w_4)$, which is estimated based on $p(w_5|w_3 w_4)$ if $w_2 w_3 w_4 w_5$ was not observed, etc.

In most smoothing methods, the lower-order models, for all $N > 1$, are recursively estimated in the same way as the highest-order model. However, the smoothing method of Kneser and Ney (1995) and its variants are the most effective methods known (Chen and Goodman, 1998), and they use a different way of computing N-gram counts for all the lower-order models used for smoothing. For these lower-order models, the actual corpus counts $C(w_1 \ldots w_n)$ are replaced by

$$C'(w_1 \ldots w_n) = |\{w'|C(w' w_1 \ldots w_n) > 0\}|$$

In other words, the count used for a lower-order N-gram is the number of distinct word types that precede it in the training corpus.

The fact that the lower-order models are estimated differently from the highest-order model makes the use of Kneser-Ney (KN) smoothing awkward in some situations. For example, coarse-to-fine search using a sequence of lower-order to higher-order language models has been shown to be an efficient way of constraining high-dimensional search spaces for speech recognition (Murveit et al., 1993) and machine translation (Petrov et al., 2008). The lower-order models used in KN smoothing, however, are very poor estimates of the probabilities for N-grams that *have* been observed in the training corpus, so they are

Proceedings of the ACL-IJCNLP 2009 Conference Short Papers, pages 349–352,
Suntec, Singapore, 4 August 2009.

$$p(w_n|w_1 \dots w_{n-1}) = \begin{cases} \alpha_{w_1 \dots w_{n-1}} \frac{C_n(w_1 \dots w_n) - D_{n,C_n(w_1 \dots w_n)}}{\sum_{w'} C_n(w_1 \dots w_{n-1} w')} & \\ + \beta_{w_1 \dots w_{n-1}} p(w_n|w_2 \dots w_{n-1}) & \text{if } C_n(w_1 \dots w_n) > 0 \\ \\ \gamma_{w_1 \dots w_{n-1}} p(w_n|w_2 \dots w_{n-1}) & \text{if } C_n(w_1 \dots w_n) = 0 \end{cases}$$

Figure 1: General language model smoothing schema

not suitable for use in coarse-to-fine search. Thus, two versions of every language model below the highest-order model would be needed to use KN smoothing in this case.

Another case in which use of special KN counts is problematic is the method presented by Nguyen et al. (2007) for building and applying language models trained on very large corpora (up to 40 billion words in their experiments). The scalability of their approach depends on a "backsorted trie", but this data structure does not support efficient computation of the special KN counts.

In this paper, we introduce a new smoothing method for language models based on ordinary counts. In our experiments, it outperformed all of the previous ordinary-count methods we tested, and it eliminated most of the gap between KN smoothing and the other previous methods.

2 Overview of Previous Methods

All the language model smoothing methods we will consider can be seen as instantiating the recursive schema presented in Figure 1, for all n such that $N \geq n \geq 2$,[1] where N is the greatest N-gram length used in the model.

In this schema, C_n denotes the counting method used for N-grams of length n. For most smoothing methods, C_n denotes actual training corpus counts for all n. For KN smoothing and its variants, however, C_n denotes actual corpus counts only when n is the greatest N-gram length used in the model, and otherwise denotes the special KN C' counts.

In this schema, each N-gram count is discounted according to a D parameter that depends, at most, on the N-gram length and the the N-gram count itself. The values of the α, β, and γ parameters depend on the context $w_1 \dots w_{n-1}$. For each context, the values of α, β, and γ must be set to produce a normalized conditional probability distribution. Additional constraints on the previous models we consider further reduce the degrees of freedom so that ultimately the values of these parameters are completely fixed by the values selected for the D parameters.

The previous smoothing methods we consider can be classified as either "pure backoff", or "pure interpolation". In pure backoff methods, all instances of $\alpha = 1$ and all instances of $\beta = 0$. The pure backoff methods we consider are Katz backoff and backoff absolute discounting, due to Ney et al.[2] In Katz backoff, if $C(w_1 \dots w_n)$ is greater than a threshold (here set to 5, as recommended by Katz) the corresponding $D = 0$; otherwise D is set according to the Good-Turing method.[3]

In backoff absolute discounting, the D parameters depends, at most, on n; there is either one discount per N-gram length, or a single discount used for all N-gram lengths. The values of D can be set either by empirical optimization on held-out data, or based on a theoretically optimal value derived from a leaving-one-out analysis, which Ney et al. show to be approximated for each N-gram length by $N_1/(N_1 + 2N_2)$, where N_r is the number of distinct N-grams of that length occuring r times in the training corpus.

In pure interpolation methods, for each context, β and γ are constrained to be equal. The models we consider that fall into this class are interpolated absolute discounting, interpolated KN, and modified interpolated KN. In these three methods, all instances of $\alpha = 1$.[4] In interpolated absolute discounting, the instances of D are set as in backoff absolute discounting. The same is true for inter-

[1] For $n = 2$, we take the expression $p(w_n|w_2 \dots w_{n-1})$ to denote a unigram probability estimate $p(w_2)$.

[2] For all previous smoothing methods other than KN, we refer the reader only to the excellent comparative study of smoothing methods by Chen and Goodman (1998). References to the original sources may be found there.

[3] Good-Turing discounting is usually expressed in terms of a discount ratio, but this can be reformulated as $D_r = r - d_r r$, where D_r is the subtractive discount for an N-gram occuring r times, and d_r is the corresponding discount ratio.

[4] Jelinek-Mercer smoothing would also be a pure interpolation instance of our language model schema, in which all instances of $D = 0$ and, for each context, $\alpha + \beta = 1$.

polated KN, but the lower-order models are estimated using the special KN counts.

In Chen and Goodman's (1998) modified interpolated KN, instead of one D parameter for each N-gram length, there are three: D_1 for N-grams whose count is 1, D_2 for N-grams whose count is 2, and D_3 for N-grams whose count is 3 or more. The values of these parameters may be set either by empirical optimization on held-out data, or by a theoretically-derived formula analogous to the Ney et al. formula for the one-discount case:

$$D_r = r - (r+1)Y\frac{N_{r+1}}{N_r},$$

for $1 \leq r \leq 3$, where $Y = N_1/(N_1 + 2N_2)$, the discount value derived by Ney et al.

3 The New Method

Our new smoothing method is motivated by the observation that unsmoothed MLE language models suffer from two somewhat independent sources of error in estimating probabilities for the N-grams observed in the training corpus. The problem that has received the most attention is the fact that, on the whole, the MLE probabilities for the observed N-grams are overestimated, since they end up with all the probability mass that should be assigned to the unobserved N-grams. The discounting used in Katz backoff is based on the Good-Turing estimate of exactly this error.

Another source of error in MLE models, however, is quantization error, due to the fact that only certain estimated probability values are possible for a given context, depending on the number of occurrences of the context in the training corpus. No pure backoff model addresses this source of error, since no matter how the discount parameters are set, the number of possible probability values for a given context cannot be increased just by discounting observed counts, as long as all N-grams with the same count receive the same discount. Interpolation models address quantization error by interpolation with lower-order estimates, which should have lower quantization error, due to higher context counts. As we have noted, most existing interpolation models are constrained so that the discount parameters fully determine the interpolation parameters. Thus the discount parameters have to correct for both types of error.[5]

Our new model provides additional degrees of freedom so the α and β interpolation parameters can be set independently of the discount parameters D, with the intention that the α and β parameters correct for quantization error, and the D parameters correct for overestimation error. This is accomplished by relaxing the link between the β and γ parameters. We require that for each context, $\alpha \geq 0$, $\beta \geq 0$, and $\alpha + \beta = 1$, and that for every $D_{n,C_n(w_1 \ldots w_n)}$ parameter, $0 \leq D \leq C_n(w_1 \ldots w_n)$. For each context, whatever values we choose for these parameters within these constraints, we are guaranteed to have some probability mass between 0 and 1 left over to be distributed across the unobserved N-grams by a unique value of γ that normalizes the conditional distribution.

Previous smoothing methods suggest several approaches to setting the D parameters in our new model. We try four such methods here:

1. The single theory-based discount for each N-gram length proposed by Ney et al.,
2. A single discount used for all N-gram lengths, optimized on held-out data,
3. The three theory-based discounts for each N-gram length proposed by Chen and Goodman,
4. A novel set of three theory-based discounts for each N-gram length, based on Good-Turing discounting.

The fourth method is similar to the third, but for the three D parameters per context, we use the discounts for 1-counts, 2-counts, and 3-counts estimated by the Good-Turing method. This yields the formula

$$D_r = r - (r+1)\frac{N_{r+1}}{N_r},$$

which is identical to the Chen-Goodman formula, except that the Y factor is omitted. Since Y is generally between 0 and 1, the resulting discounts will be smaller than with the Chen-Goodman formula.

To set the α and β parameters, we assume that there is a single unknown probability distribution for the amount of quantization error in every N-gram count. If so, the total quantization error for a given context will tend to be proportional to the

[5] Jelinek-Mercer smoothing is an exception to this generalization, but since it has only interpolation parameters and no discount parameters, it forces the interpolation parameters to do the same double duty that other models force the discount parameters to do.

number of distinct counts for that context, in other words, the number of distinct word types occurring in that context. We then set α and β to replace the proportion of the total probability mass for the context represented by the estimated quantization error with probability estimates derived from the lower-order models:

$$\beta_{w_1 \ldots w_{n-1}} = \delta \frac{|\{w' | C_n(w_1 \ldots w_{n-1} w') > 0\}|}{\sum_{w'} C_n(w_1 \ldots w_{n-1} w')}$$
$$\alpha_{w_1 \ldots w_{n-1}} = 1 - \beta_{w_1 \ldots w_{n-1}}$$

where δ is the estimated mean of the quantization error introduced by each N-gram count.

We use a single value of δ for all contexts and all N-gram lengths. As an *a priori* "theory"-based estimate, we assume that, since the distance between possible N-gram counts, after discounting, is approximately 1.0, their mean quantization error would be approximately 0.5. We also try setting δ by optimization on held-out data.

4 Evaluation and Conclusions

We trained and measured the perplexity of 4-gram language models using English data from the WMT-06 Europarl corpus (Koehn and Monz, 2006). We took 1,003,349 sentences (27,493,499 words) for training, and 2000 sentences each for testing and parameter optimization.

We built models based on six previous approaches: (1) Katz backoff, (2) interpolated absolute discounting with Ney et al. formula discounts, backoff absolute discounting with (3) Ney et al. formula discounts and with (4) one empirically optimized discount, (5) modified interpolated KN with Chen-Goodman formula discounts, and (6) interpolated KN with one empirically optimized discount. We built models based on four ways of computing the D parameters of our new model, with a fixed $\delta = 0.5$: (7) Ney et al. formula discounts, (8) one empirically optimized discount, (9) Chen-Goodman formula discounts, and (10) Good-Turing formula discounts. We also built a model (11) based on one empirically optimized discount $D = 0.55$ and an empircially optimized value of $\delta = 0.9$. Table 1 shows that each of these variants of our method had better perplexity than every previous ordinary-count method tested.

Finally, we performed one more experiment, to see if the best variant of our model (11) combined with KN counts would outperform either variant of interpolated KN. It did not, yielding a perplexity of 53.9 after reoptimizing the two free parameters of the model with the KN counts. However, the best variant of our model eliminated 65% of the difference in perplexity between the best previous ordinary-count method tested and the best variant of KN smoothing tested, suggesting that it may currently be the best approach when language models based on ordinary counts are desired.

	Method	PP
1	Katz backoff	59.8
2	interp-AD-fix	62.6
3	backoff-AD-fix	59.9
4	backoff-AD-opt	58.8
5	KN-mod-fix	52.8
6	KN-opt	53.0
7	new-AD-fix	56.3
8	new-AD-opt	55.6
9	new-CG-fix	57.4
10	new-GT-fix	56.1
11	new-AD-2-opt	54.9

Table 1: 4-gram perplexity results

References

Chen, Stanley F., and Joshua Goodman. 1998. An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Harvard University.

Kneser, Reinhard, and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. In *Proceedings of ICASSP-95*, vol. 1, 181–184.

Koehn, Philipp, and Christof Monz. 2006. Manual and automatic evaluation of machine translation between European languages. In *Proceedings of WMT-06*, 102–121.

Murveit, Hy, John Butzberger, Vassilios Digalakis, and Mitch Weintraub. 1993. Progressive search algorithms for large-vocabulary speech recognition. In *Proceedings of HLT-93*, 87–90.

Nguyen, Patrick, Jianfeng Gao, and Milind Mahajan. 2007. MSRLM: a scalable language modeling toolkit. Technical Report MSR-TR-2007-144. Microsoft Research.

Petrov, Slav, Aria Haghighi, and Dan Klein. 2008. Coarse-to-fine syntactic machine translation using language projections. In *Proceedings of ACL-08*. 108–116.