A Simple Closed-Class/Open-Class Factorization for Improved Language Modeling

Fuchun Peng and Dale Schuurmans
Department of Computer Science
University of Waterloo
200 University Avenue West
Waterloo, Ontario, Canada, N2L 3G1
{f3peng,dale}@ai.uwaterloo.ca

Abstract

We describe a simple improvement to n-gram language models where we estimate the distribution over closed-class (function) words separately from the conditional distribution of open-class words given function words. In English, function words account for about 30% of written language, and also form a natural skeleton for most sentences. By factoring a language model into a function word model and a conditional model over open-class words given function words, we largely avoid the problem of sparse training data in the first phase, and localize the need for sophisticated smoothing techniques primarily to the second conditional model. We test our factored approach on the Brown and Wall Street Journal corpora and observe a 3.5% to 25.2% improvement in perplexity over standard methods, depending on the particular smoothing method and test set used. Compared to other proposals for improving n-gram language models, our factorization has the advantage of inherent simplicity and efficiency, and improves generalization between data sets.

1 Introduction

Statistical language modeling is concerned with determining the probability of naturally occurring word sequences in a language. Traditionally, the dominant motivation for language modeling has come from speech recognition. However, statistical language models have recently become more widely used in many other application areas, such as information retrieval, machine translation, optical character recognition, spelling correction, document classification, information extraction, and bioinformatics.

The goal of language modeling is to predict the probability of natural word sequences, $s = w_1 w_2 ... w_N$; or more simply, to put high probability on word sequences that actually occur (and low probability on word sequences that do not occur). Given a word sequence $w_1 w_2 ... w_N$ to be used as a test corpus, the quality of a language model can be measured by the empirical perplexity and entropy scores on this corpus (Bahl et al., 1983)

$$\begin{aligned} Perplexity &= \sqrt[N]{\prod_{i=1}^{N} \frac{1}{Pr(w_i|w_1...w_{i-1})}} \\ Entropy &= \log_2 Perplexity \end{aligned}$$

The goal is to obtain small values of these measures.

The simplest and most successful basis for language modeling is the n-gram model. Note that by the chain rule of probability we can write the probability of any word sequence as

$$Pr(w_1 w_2 ... w_N) = \prod_{i=1}^{N} Pr(w_i|w_1...w_{i-1}) \quad (1)$$

An n-gram model approximates this probability by assuming that the only words relevant to predicting $Pr(w_i|w_1...w_{i-1})$ are the previous $n-1$ words; that is, it assumes

$$Pr(w_i|w_1...w_{i-1}) = Pr(w_i|w_{i-n+1}...w_{i-1})$$

A straightforward maximum likelihood estimate of n-gram probabilities from a corpus is given by the observed frequency

$$\hat{Pr}(w_i|w_{i-n+1}...w_{i-1}) = \frac{\#(w_{i-n+1}...w_i)}{\#(w_{i-n+1}...w_{i-1})} \quad (2)$$

where $\#(.)$ is the number of occurrences of a specified gram in the training corpus. Although one could attempt to use these simple n-gram models to capture long range dependencies in language, attempting to do so directly immediately creates sparse data problems. Using grams of length up to

n entails estimating the probability of W^n events, where W is the size of the word vocabulary. This quickly overwhelms modern computational and data resources for even modest choices of n (beyond 3 to 6). Also, because of the heavy tailed nature of language (i.e. Zipf's law) one is likely to encounter novel n-grams that were never witnessed during training in any test corpus, and therefore some mechanism for assigning non-zero probability to novel n-grams is a central and unavoidable issue in statistical language modeling.

The standard approach to smoothing probability estimates to cope with sparse data problems (and to cope with potentially missing n-grams) is to use some sort of back-off or interpolated estimator (Katz, 1987; Ney et al., 1994; Chen and Goodman, 1998; Witten and Bell, 1991). The baseline model used in this paper is a discounted back-off n-gram model, which is defined as

$$Pr(w_i|w_{i-n+1}...w_{i-1}) = \begin{cases} \hat{Pr}(w_i|w_{i-n+1}...w_{i-1}), \\ \qquad \text{if } \#(w_{i-n+1}...w_i) > 0 \\ \beta(w_{i-n+1}...w_{i-1}) \times Pr(w_i|w_{i-n+2}...w_{i-1}), \\ \qquad \text{otherwise} \end{cases} \tag{3}$$

where

$$\hat{Pr}(w_i|w_{i-n+1}...w_{i-1}) = \frac{discount\ \#(w_{i-n+1}...w_i)}{discount\ \#(w_{i-n+1}...w_{i-1})}$$

and $\beta(w_{i-n+1}...w_{i-1})$ is a normalization constant.

Many sophisticated language models have been proposed to improve this basic back-off n-gram model. These models include link grammars (Lafferty et al., 1992), sentence mixtures (Iyer, 1999), decision trees, clustering (Brown et al., 1992), caching (Jelinek et al., 1991; Clarkson, 1999), skipping models (Rosenfeld, 1994; Siu and Ostendorf, 2000), maximum entropy models (Rosenfeld, 1994; Khudanpur and Wu, 2000), latent semantic analysis (Bellegarda, 2000), structured language models (Chelba and Jelinek, 1998; Charniak, 2001), neural network models (Bengio et al., 2001), and web-data improved trigrams (Zhu and Rosenfeld, 2001). The two references (Rosenfeld, 2000; Goodman, 2000) provide a thorough overview and systematic investigation of current techniques. Most of these methods attack the problem of coping with sparse training data directly, although some techniques also focus on improving the basic model by capturing longer range dependencies in language. Goodman (2000) shows that most of these techniques improve perplexity of the baseline n-gram model by a factor of 10%-26% individually, but that a sophisticated combination of a few techniques can achieve a state-of-the-art model that obtains a 50% perplexity reduction (1 bit entropy reduction) over the standard baseline.

In the remainder of the paper, we present a surprisingly simple extension to the basic n-gram language model that incorporates a trivial linguistic notion, introduces no new statistical ideas, and yet achieves comparable perplexity reductions to the complex extensions mentioned above. Our approach is far simpler than other approaches, and entails almost no computational overhead over the basic smoothed n-gram model itself.

2 A closed-class/open-class factorization

Although words fall into many parts of speech categories, the categories themselves can be divided into two types, which are commonly referred to as closed-class and open-class words (Bradley, 1978). Closed-class words comprise the function words in a language, and are called closed-class because new words are rarely added to these categories. For example, the standard closed-class categories include prepositions, articles, pronouns, conjunctions and wh-words, which comprise a small set of distinct words in natural language that stay relatively fixed over time. By contrast, the open-class words—nouns, verbs, adjectives and adverbs—comprise a vastly larger portion of the vocabulary that is subject to constant change as new words get introduced and old words fall into disuse.

In human languages, the closed-class words play a distinct, syntactic role from open-class words. One way of viewing closed-class words is that they reveal the skeletal structure of language, and do so somewhat independently of open-class words. In fact, psychological research has suggested that the closed and open lexicons are accessed via different mechanisms in human language processing (Bradley, 1978; Herron, 1998). The difference between these types of words is also revealed in research on information retrieval, which almost universally remove closed-class words (stop words) from processing models because these words say little about the topic of a document. However, in language modeling the closed-class words cannot be simply ignored because they must be accounted for in the probabilistic predictions.

From the perspective of language modeling, the most important difference between these two classes of words is the relative sizes of each set, and the relative frequencies with which each of their members occur. Clearly, the closed-class words comprise a much smaller set of distinct words than the set of all open-class words, and yet in natural language closed-class words occur nearly as often as open-class words. For example, in the Brown corpus there are 47,703 distinct words in the vocabulary, but only 172 distinct closed-class words. Among the 1,171,008 words in the corpus, 397,146 of these are closed-class words, which implies that the average frequency of closed-class words is about 142 times that of open-class words. Additionally, the average distance between two closed-class words in the Brown corpus is 1.9 words.

Based on these facts, we propose a language model that distinguishes between closed- and open-class words and attempts to explicitly exploit their differences. To do so, we first build an m-gram model for closed-class words that ignores the open-class words in the corpus. Once constructed, we then build a second (conditional) n-gram model for the open-class words that depends also on the closed-class words. Although this sounds simple minded, a closed-class word model that does not depend on open-class words allows one to achieve much better statistical estimates, even if this compromises the potential predictive power of using open-class words. Our experimental results below verify that this tradeoff nevertheless works on our favor. Interestingly, the advantage comes not only from the raw frequency of closed-class words, but also from their internal predictive coherence.

The specific m-gram model we use for closed-class words predicts the probability of the next closed-class word conditioned only on the previous $m-1$ closed-class words, which can be easily identified from the text. We then build a n-gram model for open-class words that predicts the probability of the next open-class word conditioned on the previous $n-1$ words, including both closed-class and open-class words. In essence, we estimate the distribution over open-class words using completely standard n-gram modeling techniques, but train the m-gram model for closed-class words by first extracting a skeleton corpus, S_C, consisting of only the closed-class words from the given training corpus S.

To illustrate how our factored language model works in detail, consider the following hypothetical word sequence $S = c_1 o_{11} o_{12} o_{13} c_2 o_{21} o_{22} c_3 o_{31} c_4 o_{41} o_{42} c_5$, where c denotes a closed-class word and o denotes an open-class word. Here, the *skeleton* of the sequence is $S_C = c_1 c_2 c_3 c_4 c_5$. First, we train a standard m-gram model for the closed-class words

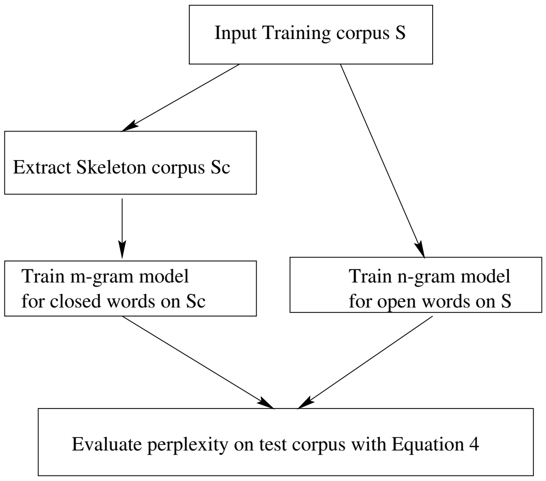

Figure 1: Training procedure for the closed/open m/n-gram word model

on S_C, and then train a standard n-gram model for the open-class words on all of S. We call the resulting combination a factored m/n-gram closed/open word model. Given this model the probability of a new test sequence $s = w_1 w_2 ... w_N$ can be computed by

$$Pr(s) = \prod_{i=1}^{N} Pr(w_i | w_1 ... w_{i-1})$$

where

$$Pr(w_i|w_1...w_{i-1}) = \begin{cases} \hat{P}r(w_i|\text{skeleton of } w_{i-n+2}...w_{i-1}) & \\ \qquad\qquad \text{if } w_i \text{ is a closed class word} & \\ \hat{P}r(w_i|w_{i-n+1}...w_{i-1}) \times \alpha(w_{i-n+1}...w_{i-1}) & \\ \qquad\qquad \text{if } w_i \text{ is an open class word} & \end{cases} \tag{4}$$

and $\alpha(w_{i-n+1}...w_{i-1})$ is a normalization constant such that

$$\alpha(w_{i-n+1}...w_{i-1}) = \frac{1 - \sum_{x \in \text{ closed words}} \hat{P}r(x|\text{skeleton of } w_{i-n+2}...w_{i-1})}{1 - \sum_{x \in \text{ closed words}} \hat{P}r(x|w_{i-n+2}...w_{i-1})}$$

(This ensures that the mass assigned to open-class words given $w_{i-n+1}...w_{i-1}$ is the same as the mass assigned given its skeleton.) The overall training regime for an m/n-gram closed/open word model is illustrated in Figure 1.

Although this extension to the standard n-gram language model is in some sense trivial, it nevertheless incorporates a small piece of concrete linguistic knowledge. What is more important is that this seemingly trivial extension gives a surprisingly significant and robust improvement to basic n-gram models with no real additional effort conceptually or computationally.

3 Experiments

To evaluate our factored approach to language modeling we conducted experiments on two large data sets: the Brown corpus (105,210 sentences, 1,171,008 words, 47,703 unique words) and the Wall Street Journal corpus (47,589 sentences, 1,232,350 words, 44,516 unique words). In each case, we converted the entire corpus to lower case text and then randomly selected 5000 sentences to serve as test examples. Specifically, we randomly selected 5000 sentences from the Brown corpus to serve as a test corpus, B1, and used the remainder of the corpus, B2, to train the language models. Similarly, we randomly selected 5000 sentences from the WSJ to serve as a test corpus, WSJ1, and used the remainder, WSJ2, for training.

To build a lexicon of closed-class words we automatically extracted all words that were given the part of speech tags DT, CC, IN, PRP, PRP$, TO, WDT, WP$, WRB, and WP in the Penn Treebank-3 tagged version of the Brown corpus. These words consist of all the prepositions, articles, pronouns, conjunctions, and wh-words occurring in the corpus. The final list of closed-class words had 172 distinct entries.

With the two training corpora we ran two experiments. First, we trained our factored language model on the training set B2 extracted from the Brown corpus, and tested this model on both test sets B1 and WSJ1 respectively. We then repeated the experiment by training on the training set WSJ2 extracted from the WSJ corpus, and tested this model on both test sets (B1 and WSJ1) as well. The following two subsections report the results of these two experiments respectively.

3.1 Training on the Brown corpus

To establish a baseline, we first report the results of using traditional n-gram models trained on the sentences in B2 and evaluated on the sentences in B1 and WSJ1, but employing different smoothing (discounting) techniques (Table 1). The numbers in the table are the perplexity/entropy scores obtained on the two test corpora.

Table 2 shows the results obtained by using the factored m/n-gram closed/open word model described in the previous section. Table 3 shows the improvement of this factored model over the traditional n-gram approach, where for each dis-

Discounting	n	B1	WSJ1
Absolute	2	367.21/8.5204	700.75/9.4527
	3	**342.67/8.4206**	**698.17/9.4474**
	4	353.13/8.4641	711.13/9.4739
Linear	2	**407.28/8.6718**	**773.66/9.5955**
	3	409.28/8.6790	808.72/9.6590
	4	426.82/8.7374	826.90/9.6916
Good-Turing	2	362.82/8.5030	692.16/9.4349
	3	**339.26/8.4063**	**688.18/9.4266**
	4	350.33/8.4526	701.86/9.4550
Witten-Bell	2	377.04/8.5580	**716.98/9.4857**
	3	**370.65/8.5339**	783.47/9.6137
	4	392.37/8.6160	827.87/9.6932

Table 1: Baseline n-gram model trained on the Brown Corpus

counting technique the improvement is calculated by comparing the best standard n-gram model to the best m/n-gram closed/open model (shown in boldface in Tables 1 and 2).

Discounting	m/n	B1	WSJ1
Absolute	2/2	644.72/9.3325	1199.79/10.2285
	3/2	656.83/9.3593	1224.62/10.2581
	2/3	322.84/8.3347	651.35/9.3473
	3/3	328.91/8.3615	664.83/9.3768
	2/4	**307.11/8.2626**	626.72/9.2916
	2/5	308.19/8.2677	**625.68/9.2893**
Linear	2/2	687.40/9.4250	1284.40/10.3268
	3/2	721.16/9.4941	1343.68/10.3919
	2/3	354.97/8.4715	708.00/9.4676
	3/3	372.41/8.5407	740.67/9.5326
	2/4	334.70/8.3867	674.73/9.3981
	2/5	**334.68/8.3866**	**671.40/9.3910**
Good-Turing	2/2	639.91/9.3217	1188.68/10.2151
	3/2	650.28/9.3449	1210.92/10.2418
	2/3	319.84/8.3212	642.76/9.3281
	3/3	325.03/8.3444	654.78/9.3548
	2/4	**303.87/8.2473**	616.68/9.2683
	2/5	304.65/8.2510	**615.03/9.2645**
Witten-Bell	2/2	643.17/9.3290	1203.13/10.2325
	3/2	655.35/9.3561	1228.50/10.2626
	2/3	331.77/8.3740	**692.07/9.4347**
	3/3	338.06/8.4011	706.66/9.4648
	2/4	**329.36/8.3635**	702.99/9.4573
	2/5	335.17/8.3887	712.81/9.4773

Table 2: Factored closed/open m/n-gram model trained on the Brown Corpus

Discounting	B1	WSJ1
Absolute	10.4% (0.1580)	10.4% (0.1581)
Linear	17.8% (0.2852)	13.3% (0.2054)
Good-Turing	10.4% (0.1590)	10.6% (0.1622)
Witten-Bell	11.1% (0.1704)	3.5% (0.0510)

Table 3: Perplexity (entropy) improvement when trained on the Brown Corpus

These results show that a significant improvement in perplexity is achieved regardless of the type of smoothing employed and regardless of the corpus used for testing. The largest perplexity improvement came when using linear discounting, which was the weakest of the smoothing methods. Here the reduction was **17.8%** (0.2852 bits entropy) on the Brown test corpus B1, and **13.3%** (0.2054 bits entropy) on WSJ1. However, the simple closed/open factorization still led to improved performance even when considering the best smoothing method, which was Good-Turing smoothing in this case. In this case the perplexity reduction was still **10.4%** (0.1590 bits entropy) on the Brown test corpus B1, and **10.6%** (0.1622 bits entropy) on the WSJ1.

3.2 Training on the WSJ corpus

Qualitatively similar results are obtained when training on the WSJ corpus. First, to re-establish baseline performance, Table 4 shows the results of traditional n-gram models trained on WSJ2 and evaluated on B1 and WSJ1 using each of the different discounting methods.

Discounting	n	B1	WSJ1
Absolute	2	**637.46/9.3162**	270.96/8.0819
	3	650.19/9.3447	**226.51/7.8234**
	4	659.68/9.3656	230.41/7.8481
Linear	2	**694.96/9.4407**	297.44/8.2164
	3	728.82/9.5094	**268.90/8.0709**
	4	739.38/9.5300	279.34/8.1258
Good-Turing	2	**626.87/9.2920**	267.79/8.0649
	3	637.56/9.3164	**224.62/7.8113**
	4	646.48/9.3364	229.40/7.8417
Witten-Bell	2	**640.94/9.3241**	277.13/8.1144
	3	712.35/9.4764	**237.86/7.8940**
	4	753.89/9.5582	243.22/7.9261

Table 4: Baseline n-gram model trained on the WSJ

Table 5 shows the results obtained by using the factored m/n-gram closed/open word model, and Table 6 shows the perplexity (entropy) improvement of this factored model over the traditional n-gram approach.

Discounting	m/n	B1	WSJ1
Absolute	2/2	867.79/9.7612	488.89/8.9333
	3/2	886.85/9.7925	495.76/8.9535
	2/3	514.26/9.006	225.86/7.8192
	3/3	525.56/9.0377	229.03/7.8394
	2/4	500.82/8.9681	**213.40/7.7374**
	2/5	**500.26/8.9665**	216.05/7.7552
Linear	2/2	898.51/9.8114	521.68/9.0270
	3/2	963.81/9.9126	542.63/9.0838
	2/3	539.30/9.0749	252.81/7.9819
	3/3	578.49/9.1761	262.96/8.0387
	2/4	521.85/9.0274	**240.75/7.9114**
	2/5	**520.16/9.0228**	243.94/7.9303
Good-Turing	2/2	854.09/9.7382	483.86/9.7382
	3/2	872.26/9.7686	489.82/8.9361
	2/3	504.27/8.9780	224.03/7.8075
	3/3	515.00/9.0084	226.79/7.8252
	2/4	489.96/8.9365	**211.92/7.7274**
	2/5	**489.03/8.9337**	214.64/7.7458
Witten-Bell	2/2	844.50/9.7219	490.59/8.9383
	3/2	861.40/9.7505	497.59/8.9588
	2/3	**530.54/9.0513**	228.76/7.8376
	3/3	541.16/9.0799	232.02/8.9329
	2/4	542.01/9.0821	**220.22/7.7828**
	2/5	547.75/9.0973	222.87/7.8001

Table 5: Factored closed/open m/n-gram model trained on the WSJ

Once again, a noticeable improvement in perplexity is observed regardless of smoothing technique and test corpus. Linear discounting was again the weakest smoothing method, and led to the largest improvements in perplexity using the factored closed/open model. Here the reduction was **25.2%** (0.4187 bits entropy) on the Brown test corpus B1, and **10.5%** (0.1595 bits entropy) on WSJ1. However, we also achieve an improvement even for the best smoothing method, which again was Good-Turing smoothing. In this case, a reduction of **22.0%** (0.3583 bits entropy) was obtained on B1, and **5.7%** (0.0839 bits entropy) on WSJ1.

Discounting	B1	WSJ1
Absolute	21.6% (0.3504)	5.8% (0.0860)
Linear	25.2% (0.4187)	10.5% (0.1595)
Good-Turing	22.0% (0.3583)	5.7% (0.0839)
Witten-Bell	17.2% (0.2728)	7.4% (0.1112)

Table 6: Perplexity (entropy) improvement when trained on the WSJ

4 Discussion

A long standing trend in statistical language modeling research is to focus on the problem of sparse training data and pursue sophisticated techniques for dimensionality reduction in an attempt to achieve more reliable estimates. This trend encompasses research into various smoothing techniques (and extensions) such as vocabulary clustering methods, maximum entropy models, neural network models, and decision trees, among others. Another recent trend is to use additional training data from auxiliary sources to help improve the model; for example, by using additional web data (Zhu and Rosenfeld, 2001). In contrast to this work, we are exploiting a trivial piece of prior linguistic structure which also happens to be relevant to limiting the deleterious effects of sparse data. By factoring the vocabulary into closed- and open-classes and training a distinct model for closed-class words, we obtain a 3.5%-25.2% improvement in perplexity, depending on discounting method used and on the training/testing set. There are several observations to make about these experimental results:

The first observation is that the improvement obtained by factoring is largely due to improved prediction on closed-class words. That is, since the factored model treats open-class words almost the same way as the traditional n-gram model, it must make similar predictions on open-class words. The perplexity improvement is therefore primarily achieved from predicting closed-class words more accurately. Our results show that the gains obtained from improving the quality of the statistical estimates of closed-class words, by considering only closed-class words alone, are greater than the losses incurred by predicting closed-class words without any reference to open-class words in the surrounding context. Clearly, some of the advantage comes from reducing the sparse data problem to a point where it is not a significant impediment. However, more interestingly, our results also show that the closed-class (function) words themselves exhibit a significant amount of predictive coherence that is independent of the surrounding open-class context. Surprisingly, we conclude that, from the perspective of n-gram language modeling, open-class words do more damage than good when it comes to predicting closed-class function words—since the sparse data problems they introduce are more profound than the predictive benefits they offer.

Second, we make the simple observation that the improvement obtained by the factored model depends on the specific method used to smooth the probability estimates. However, some improvement is obtained in every case we examined. Generally speaking, the largest perplexity improvements are obtained when using the weakest smoothing method (linear discounting in our case), and the smallest improvements (but still positive) are obtained when using more effective smoothing methods such as absolute discounting or Good-Turing smoothing. Nevertheless, some improvement is robustly achieved in every case.

It is interesting to contrast the results of testing on the Brown corpus to the results of testing on the WSJ corpus. From Tables 3 and 6, we can see that the improvement in perplexity is greater when training on the Brown corpus than when training on WSJ. A partial explanation for this fact is simply that the closed-class words are more prevalent in the Brown corpus than in WSJ. (In the Brown corpus, closed-class words comprise 34% of all word occurrences, whereas in the WSJ the frequency of closed-class words is only 28%.) It seems intuitive that the more frequently closed-class words occur, the greater improvement we should expect to see in the test corpora.

However, the mere prevalence of closed-class words does not fully explain their utility in language modeling. In fact, if their benefit was solely to reduce the effects of sparse training data, then a more direct approach would be more effective: one could simply take the most frequent English words, regardless of their part of speech categories, and learn a factored model in exactly the same way as indicated in Section 2—using these frequent words instead of closed-class words to factor the model. In fact, we did just this, to validate the value of using the linguistic notion of closed- versus open- part of speech classes. Specifically, we re-ran the previous experiments by taking the 172 most frequent English words (which is the number of closed-class words, regardless of part of speech category—calculated on the training segment of the Brown corpus, B2) and learned an m-gram model over these words alone, while estimating the model over the remainder of the vocabulary using a standard n-gram model which considered all previous $n-1$ words of any type. Table 7 shows the results of running this control experiment by training on the Brown corpus, B2, and testing on sentences from both the Brown and WSJ corpora. Interestingly, we found that using the 172 most frequent words to factor the language model does indeed lead to an improved perplexity score over standard n-gram modeling (Table 8). However, this improvement was generally not as great as that of using closed-class (function) words (Table 9). Here, a weaker improvement is obtained for the stronger smoothing models (Good-Turing smoothing and absolute smoothing) even though the 172 most frequent words account for 53% of the Brown corpus (whereas the closed-class words only account for 34%). This result shows that the improvement obtained by the closed/open class factorization is not solely due to reducing the sparse data problem, but also due to exploiting the non-trivial predictive coherence that exists between function words, independent of their surrounding open-class context.

	m/n	B1	WSJ1
Absolute	2/2	543.40/9.0858	1000.01/9.9658
	3/2	531.87/9.0549	1016.81/9.9898
	2/3	344.05/8.4264	666.40/9.3802
	3/3	336.75/8.3955	677.59/9.4042
	2/4	329.36/8.3635	648.15/9.3401
	2/5	**328.79/8.3610**	**646.42/9.3363**
Linear	2/2	566.24/9.1452	1051.85/10.0387
	3/2	587.26/9.1978	1118.12/10.1268
	2/3	349.49/8.4491	684.06/9.4179
	3/3	362.46/8.5017	727.15/9.5061
	2/4	329.31/8.3633	655.02/9.3554
	2/5	**327.43/8.3550**	**651.24/9.3470**
Good-Turing	2/2	558.61/9.1256	1030.50/10.0091
	3/2	545.07/9.0903	1045.33/10.0297
	2/3	343.51/8.4242	664.47/9.3760
	3/3	335.18/8.3888	674.04/9.3966
	2/4	327.11/8.3536	642.87/9.3283
	2/5	**326.11/8.3492**	**640.50/9.3230**
Witten-Bell	2/2	525.39/9.0372	969.94/9.9217
	3/2	513.57/9.0044	980.48/9.9373
	2/3	346.79/8.4379	**687.61/9.4254**
	3/3	**338.99/8.4051**	695.09/9.4410
	2/4	349.92/8.4509	707.62/9.4668
	2/5	356.14/8.4763	717.77/9.4873

Table 7: Factored closed/open m/n-gram word model trained on the Brown Corpus, using the 172 most frequent words (instead of closed-class words) to factor the model

	B1	WSJ1
Absolute	4.1% (0.0596)	7.4% (0.1111)
Linear	19.6% (0.3168)	15.9% (0.2492)
Good-Turing	3.9% (0.0571)	6.9% (0.1038)
Witten-Bell	8.5% (0.1288)	4.1% (0.0603)

Table 8: Perplexity (entropy) improvement of using the 172 word factorization over the baseline n-gram model

	B1	WSJ1
Absolute	-7.1% (-0.0984)	-3.3% (-0.047)
Linear	2.2% (0.0316)	3% (0.0438)
Good-Turing	-7.3% (-0.1019)	-4.1% (-0.0584)
Witten-Bell	-3% (-0.0416)	0.6% (0.0093)

Table 9: Perplexity (entropy) difference in using the top 172 word factorization compared to the closed-class word factorization

Returning to the experiments in Section 3, we observe that one of the main benefits of our factored approach is that it leads to significantly better generalization between data sets than the traditional n-gram model. For example, compare the results of testing on the Brown corpus when training on the Brown and WSJ corpora respectively. That is, compare column B1 in Table 3 with column B1 in Table 6. Here we see that a much greater improvement is obtained when training on a different data source from the test corpus. That is, the perplexity reduction when testing on the Brown corpus is much greater when training on the WSJ than it is when training on the Brown corpus itself. The same phenomenon is observed when we consider the test results for the WSJ corpus. Compare column WSJ1 in Table 3 with column WSJ1 in Table 6. Again, we see a much greater reduction in perplexity when training on the Brown corpus and testing on WSJ than we see when both training and testing on WSJ (except the results on Witten-Bell smoothing). These results suggest that the model learned over closed class words is not only more accurate than traditional n-gram modeling, but also generalizes better between corpora. This is an important advantage when one considers exploiting auxiliary data sources like the web (Zhu and Rosenfeld, 2001).

Finally, our last observation is that a simple m-gram model may not be the most effective technique for capturing the predictive dependence between closed-class words. Evidence for this assertion comes from the fact that we ran experiments with m-gram models for $m = 1...6$ and found that simple 2-grams or 3-grams were optimal over this range. This outcome contradicted our prior intuition that since closed-class words were so frequent in the corpus they should be better modeled by long m-grams. In fact, because of the prevalence of closed-class words in our training corpora, this failure of the larger m-gram models cannot be entirely attributed to the sparse data problem. We suspect that perhaps a simple m-gram model is not sufficient to model the predictive relationship between function words in a language, due to their close relationship to syntactic structure. One idea we are currently exploring is whether this syntactic structure might be better modeled by a structural language model (Chelba and Jelinek, 1998; Charniak, 2001).

5 Conclusions

We have described a simple variant of the traditional n-gram model that factors the vocabulary into closed and open classes. We showed that this factored model demonstrates improvements over traditional n-gram models on the Brown and WSJ corpora, where the observed perplexity improvement ranges from 3.5%-25.2% depending on the discounting method used and the training/test set. We conclude that traditional n-gram models are not as effective at modeling closed-class words given open-class words, apparently because this entails coping with significant sparse data problems and ignores the fact that closed-class words have an internally coherent predictive structure.

Our factored closed/open model is among simplest variant of n-gram models that has recently been investigated in the literature; it is much simpler than long range n-gram models (also called link grammars) (Lafferty et al., 1992), class-based n-gram models (Brown et al., 1992), and variable length n-gram models (Siu and Ostendorf, 2000). Of course, all of these methods themselves are simpler than complex techniques such as maximum entropy models (Rosenfeld, 1994; Khudanpur and Wu, 2000) and neural network language models (Bengio et al., 2001). However, beyond the simplicity of our factored m/n-gram approach, an additional strength is that it systematically improves standard n-gram models under a variety of conditions while consistently improving generalization across data sets.

One of the main benefits of our method is that it is completely orthogonal to other language modeling techniques, and can in fact be applied in conjunction with any of the above mentioned methods (clustering, caching, skipping, maximum entropy, latent semantic analysis) with the prospect of gaining further improvements. Here, because we factor words into two distinct classes, we can use different modeling techniques for each class; for example, using class-based n-gram models for open words and probabilistic context free grammars for closed-class words. Combining specific methods in this manner remains future work.

6 Acknowledgments

We would like to thank the Waterloo Statistical NLP group and the anonymous referees for their helpful comments. We also thank Xiaojin Zhu for insightful discussion. Research supported by Bell University Labs, MITACS and NSERC.

References

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

Bellegarda, J. 2000. Exploiting latent semantic information in statistical language modeling. *Proceedings of the IEEE 88(8)*, 1279-1296.

Bengio, Y., Ducharme, R. and Vincent, P. 2001. A Neural Probabilistic Language Model. In *Advances in Neural Information Processing Systems 13*.

Bradley, D. 1978. Computational distinctions of vocabulary type. *PhD thesis*, MIT.

Brown, P., Della Pietra, V., deSouza, P., Lai, J. and Mercer, R. 1992. Class-based n-gram models of natural language. In *Computational Linguistics 18*, 467-479.

Charniak, E. 2001. Immediate-Head Parsing for Language Models. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics.*

Chelba, C. & Jelinek, F. 1998. Exploiting syntactic structure for language modelling. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, Montreal, 225-231.

Chen, S. and Goodman, J. 1998. An Empirical Study of Smoothing Techniques for Language Modeling. *Technical Report TR-10-98*, Computer Science Group, Harvard University.

Clarkson, P. 1999. Adaptation of Statistical Language Models for Automatic Speech Recognition. *PhD thesis*, Cambridge University Engineering Department.

Goodman, J. 2000. A Bit of Progress in Language Modeling. *Extended Version DRAFT To be published as a technical report*, Microsoft Research.

Herron, D. 1998. Context, World-class, and Prosody in the Recognition of the Open and Closed-class Words. *PhD thesis*, University of California, San Diego.

Iyer, R. and Ostendorf, M. 1999. Modeling long distance dependence in language: Topic mixtures versus dynamic cache models. *IEEE Transactions on Speech and Audio Processing, 7(1)*, 30-39.

Jelinek, F., Merialdo, B., Roukos, S. and Strauss, M. 1991. A dynamic LM for speech recognition. In *Proceedings ARPA workshop on Speech and Natural Language*, 293-295.

Katz, S. 1987. Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer. In *IEEE Transactions on Acoustics, Speech and Signal Processing ASSP-35*, 400-401.

Khudanpur, S. and Wu, J. 2000. Maximum Entropy Techniques for Exploiting Syntactic, Semantic and Collocational Dependencies in Language Modeling. In *Computer Speech and Language*, 355-372.

Lafferty, J., Sleator, D. and Temperley, D. 1992. Grammatical trigrams: A probabilistic model of link grammar. In *Proceedings of the AAAI Fall Symposium on Probabilistic Approaches to Natural Language*, Cambridge, MA.

Ney, H., Essen, U. and Kneser R. 1994. On structuring probabilistic dependencies in stochastic language modeling. In *Computer Speech and Language 8(1)*, 1-28.

Rosenfeld, R. 1994. Adaptive Statistical Language Modeling: A Statistical Approach. *PhD Thesis*, School of Computer Science, Carnegie Mellon University. Published as Technical Report CMU-CS-94-138.

Rosenfeld, R. 2000. Two decades of Statistical Language Modeling: Where Do We Go From Here? In *Proceedings of the IEEE 88(8).*

Siu, M. and Ostendorf, M. 2000. Variable N-gram Language Modeling and Extensions for Conversational Speech. In *IEEE Transactions on Speech and Audio Processing 8*, 63-75.

Witten, I. and Bell, T. 1991. The Zero-Frequency Problem: Estimating the Probabilities of Novel Events in Adaptive Text Compression. In *IEEE Transactions on Information Theory 37(4).*

Zhu, X. and Rosenfeld, R. 2001. Improving Trigram Language Modeling with the World Wide Web. In *proceedings of International Conference on Acoustics, Speech, and Signal Processing.*