Context-Sensitive Statistics for Improved Grammatical Language Models

Eugene Charniak and Glenn Carroll

Department of Computer Science
Brown University
Providence, Rhode Island 02912

CS-94-07
Febuary 1994

Context-Sensitive Statistics for Improved Grammatical Language Models*

Eugene Charniak
Glenn Carroll
Department of Computer Science, Brown University

February 16, 1994

Abstract

We develop a language model using probabilistic context-free grammars (PCFGs) that is "pseudo context-sensitive" in that the probability that a non-terminal N expands using a rule r depends on N's parent. We derive the equations for estimating the necessary probabilities using a variant of the inside-outside algorithm. We give experimental results showing that, beginning with a high-performance PCFG, one can develop a pseudo PCSG that yields significant performance gains. Analysis shows that the benefits from the context-sensitive statistics are localized, suggesting that we can use them to extend the original PCFG. Experimental results confirm that this is both feasible and the resulting grammar retains the performance gains. This implies that our scheme may be useful as a novel method for PCFG induction.

1 Introduction

Like its non-stochastic brethren, probabilistic parsing has been based upon context-free grammars (CFGs), and for similar reasons: CFGs support a simple and efficient parsing mechanism while also accounting for most, if not all, of the natural language phenomena one encounters, particularly in word-order based languages such as English. In probabilistic parsing, of course, one does not use plain CFGs, but rather their probabilistic counterparts (PCFGs). In these each rule of the form $N^i \rightarrow \alpha^j$ has associated with it a probability $P(N^i \rightarrow \alpha^j)$ such that

$$\sum_j P(N^i \rightarrow \alpha^j) = 1$$

*This research was supported in part by NSF contract IRI-8911122 and ONR contract N0014-91-J-1202.

for all i, where N^i is the i'th non-terminal of the grammar. Note that the probabilities are context free in the sense that the probability associated with a rule is independent of the context in which the rule might find itself.

In this paper we investigate a scheme for introducing context sensitive statistics into stochastic parsing, with the aim of improving a grammar-based language model for English. Note that this goal is quite different from other uses of context sensitive statistics such as improving the speed of parsing [14] or improving the probability of the correct parse [2]. While we believe our statistics could be adapted to these purposes, our own interest lies in the area of language models.

A language model is a distribution over strings of (English) words, and a good model should accurately reflect the true distribution of English strings. Speaking more formally, we say the model defines a distribution over examples of English of length N, $P(w_{1,N})$, where $w_{1,N}$ ranges over all possible corpora of English of length N. A good language model is one which probability to strings according to their actual frequency of occurrence, high probability for common strings, and low probability for rare ones.

With a grammar-based model, one first parses the sentences using the grammar, and then uses the parse information to assign the probabilities to the actual words (See [6].). We make the standard assumption that sentences occur independently of each other, and thus, if $w_{1,N}$ are the words of the l sentences $s_{1,l}$,

$$P(w_{1,N}) = \prod_{i=1}^{l} P(s_i) \tag{1}$$

This assumption allows us to focus on individual sentences and their parses. Given some sentence s, consisting of n words, $w_{1,n}$, assume our model assigns τ parses (or trees) to s, $t_1 \ldots t_\tau$. We can then write,

$$P(s) = P(w_{1,n}) = \sum_{i=1}^{\tau} P(t_i) P(w_{1,n} \mid t_i)$$

Our grammar model constructs parses for strings of part-of-speech tokens, *not* words. The second term above, $P(w_{1,n} \mid t_i)$, is the probability of the words given the parse tree. The idea here is that more detailed knowledge of the syntactic structure in which the words find themselves will enable the model to better predict the probabilities of the words. This is a keen area for future research. Here, however, we are concerned with the first term. Our context sensitive statistics will be used to improve the probabilities of the parses, the $P(t_i)$.

Roughly speaking, we wish to maximize the probability of sentences. (Actually, we wish to maximize a product involving these probabilities, as equation 1 states.) Since, other things being equal, $P(w_{1,n})$ is maximized when $P(t_i)$ is as large as possible, this suggests the subgoal of maximizing the sum of probabilities of all possible parses. Letting v_i stand for the ith tag of s, we have

$$\sum_{i=1}^{\tau} P(t_i) = P(v_{1,n}) \tag{2}$$

Equation 2 states that maximizing the sum of the probability of the parses is equivalent to maximizing the probability of the tag sequence $v_{1,n}$. Commonly we do not deal with the probabilities of a language model directly, but rather try to minimize the model's per-word cross-entropy. In the same way, here we try to minimize the per-tag cross entropy of a grammar model. It can be shown that in the limit this is equivalent to minimizing

$$-\frac{1}{n}\log P(v_{1,n}) \tag{3}$$

For our purposes we simply take the quantity of Equation 3 as the per-tag cross entropy.

It seems reasonable to hope that the probabilities that minimize the per-tag cross entropy would assign higher probabilities to more common parses over uncommon ones, and thus, one would hope, the intended parse over those not intended. Such probabilities could also be used to guide the parsing process. Our own goal is simply to find ways to maximize this probability, or equivalently, to minimize the cross entropy.

Besides our differing goals, a distinguishing feature of this work is that we wish to be able to collect the parsing statistics without the use of a pre-parsed corpus. While such corpora are, of course, a valuable tool, they limit the choice of grammars to those that agree with the parses assigned in the corpus, and the volume of data available in such form is still quite limited. We show in this paper how our probabilities can be collected by an extension of the standard inside-outside algorithm [1,6].

2 Pseudo Context-Sensitivity

In this paper we propose to extend a standard PCFG by replacing the probability of each context-free rule with a set of probabilities, one for each non-terminal used by our grammar. Each of these new probabilities will reflect the probability of the rule occurring in a particular context, which in our case is simply the head of the parent rule. For example in figure 1, the rule $\overline{\mathsf{vbg}}\rightarrow\mathsf{adv}\ \mathsf{vbg}$ has the non-terminal $\overline{\mathsf{n}}$ as its parent. Formally, we write

$$P(N^i \rightarrow \alpha^j \mid \rho(N^i) = N^s) \tag{4}$$

where $\rho(x)$ is the non-terminal that immediately dominates x — its *parent.* We refer to N^i as the *child.* Continuing our example in Figure 1, we would require

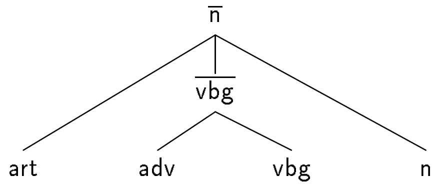

Figure 1: Application of a rule within a $\overline{\mathsf{n}}$

$$P(\overline{\mathsf{vbg}} \rightarrow \mathsf{adv\ vbg} \mid \rho(\overline{\mathsf{vbg}}) = \overline{\mathsf{n}})$$

This is the probability that we expand $\overline{\mathsf{vbg}}$, (a verb phrase headed by an "ing" verb) as an adverb followed by the verb, given that its parent is $\overline{\mathsf{n}}$, a noun phrase. Such a situation might occur in the parse of a phrase like "the slowly dripping faucet." (The base grammar for the experiments reported here is modeled after dependency grammars and thus has one non-terminal for each terminal in the language. Here $\overline{\mathsf{vbg}}$ is the non-terminal for the terminal vbg. We use this notation for all of the examples, although the techniques developed here work for all PCFGs, not just probabilistic dependency grammars.)

Note that by collecting the statistics of Equation 4 we have not, in fact, moved beyond what is expressible by PCFGs. The pseudo PCSG can be expanded into a PCFG roughly as follows. For each parent-child pair, begin by creating a new non-terminal to represent the chosen pair. For any rule with the parent as its head, substitute the new non-terminal for each occurrence of the child. For each rule headed by the child, add a new rule headed by the new non-terminal. With some trivial math, one can compute new probabilities for all the rules in the expanded grammar. We return to this process later.

Although technically we have not moved beyond PCFGs, clearly our formalism has something of a context-sensitive flavor. We like to think of it as gathering context-*sensitive* statistics for a context-*free* grammar, and thus we call our scheme "pseudo context-sensitivity." Clearly, the extra information provided by context allows us a good deal more flexibility in assigning probabilities to parses. For example, while the above rule for $\overline{\mathsf{vbg}}$ would be a not-uncommon one to find as part of a noun-phrase, consider instead the following rule for $\overline{\mathsf{vbg}}$

$$\overline{\mathsf{vbg}} \rightarrow \mathsf{vbg}\ \overline{\mathsf{n}}$$

This rule would be used in "Alice was planting the flowers." While this rule is a common one at the sentence level, at the noun-phrase level it would be quite uncommon. Our new probabilities would allow the system to capture this regularity. As just noted, this regularity could also be captured through the use of a different non-terminal dominating the gerund. However, our new scheme allows us to find and capture such regularities automatically.

To actually use this model requires first that one can efficiently estimate the probabilities specified in Equation 4 and second, that given these probabilities one can efficiently calculate the probability of a parse. The second of these is reasonably straight forward, and we leave it as an exercise for the reader. The former we cover in the next section.

3 Calculating the Probabilities

In this section we show how it is possible to calculate the rule probabilities using a variant of the inside-outside algorithm. Those who are not interested in actually implementing this scheme may safely skip to the next section, where we show how the application of these probabilities has led to a significant improvement in the probabilistic measure given in Equation 2.

In the generic inside-outside algorithm one re-estimates the probability of an event e by seeing how often it occurs in a training corpus. Our events will be rule invocations, or uses, in sentence parses. Typically each sentence in the corpus has many parses, and it is possible that e occurs zero, one, or more times in any particular parse. One estimates the e-counts by adding up, for each occurrence of e the probability of the parse in which e occurs, given the sentence. Our goal in this section is to come up with the equations for this sum. We show the equations for the case where the grammar is in Chomsky-normal form.

We want to count the number of times an event occurs ($C(e)$).

$$\begin{aligned} C(e) &= \sum_e P(e \mid w_{1,n}) \\ &= \frac{1}{P(w_{1,n})} \sum_e P(e, w_{1,n}) \end{aligned}$$

More specifically, we wish to count the occurrence of the rule $N^i \rightarrow N^p\ N^q$ in the context of $N^i\ N^s$. The sum over all possible ways this event could occur includes (1) the positions N^i, N^p, and N^q in the parse, (2) the position of N^s in the parse, and (3) the rule that relates N^s to N^i. These are shown in Figure 2. Note that for part (3) there are two cases we need to consider, where N^i occurs as the left and right constituents of the higher level rule. Figure 2 only shows the second of these. These two possibilities correspond to the two sums in the following equations.

$$\begin{aligned} & C(N^i \rightarrow N^p N^q, \rho(N^i) = N^s) \\ &= \frac{1}{P(v_{1,n})} \sum_{j,k,t,h,f} P(N^i_{j,k}, N^p_{j,f}, N^q_{f+1,k}, N^s_{h,k}, N^t_{h,j-1}, v_{1,n}) \\ &\qquad + P(N^i_{j,k}, N^p_{j,f}, N^q_{f+1,k}, N^s_{j,h}, N^t_{k+1,h}, v_{1,n}) \end{aligned} \tag{5}$$

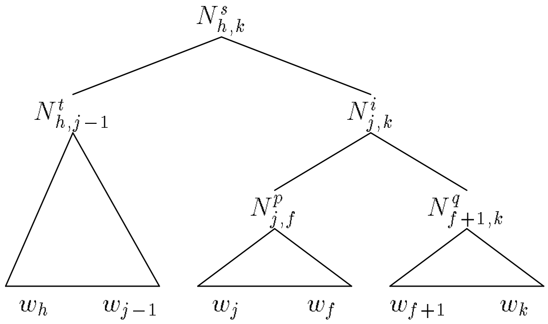

Figure 2: The situations for $N^i \rightarrow N^p \; N^q$ under N^s

$$
\begin{aligned}
&= \frac{1}{P(v_{1,n})} \sum_{j,k,t,h,f} P(v_{1,h-1}, N^s_{h,k}, v_{k+1,n}) \\
&\qquad P(N^i_{j,k}, N^t_{h,j-1} \mid v_{1,h-1}, N^s_{h,k}, v_{k+1,n}) \\
&\qquad P(v_{h,j-1} \mid N^i_{j,k}, N^t_{h,j-1}, v_{1,h-1}, N^s_{h,k}, v_{k+1,n}) \\
&\qquad P(N^p_{j,f}, N^q_{f+1,k} \mid v_{1,j-1}, N^i_{j,k}, N^t_{h,j-1}, N^s_{h,k}, v_{k+1,n}) \\
&\qquad P(v_{j,f} \mid N^p_{j,f}, N^q_{f+1,k}, v_{1,j-1}, N^i_{j,k}, N^t_{h,j-1}, N^s_{h,k}, v_{k+1,n}) \\
&\qquad P(v_{f+1,k} \mid N^p_{j,f}, N^q_{f+1,k}, v_{1,j-1}, N^i_{j,k}, N^t_{h,j-1}, N^s_{h,k}, v_{f+1,n}) \\
&\qquad + P(v_{1,j-1}, N^s_{j,h}, v_{h+1,n}) \\
&\qquad P(N^i_{j,k}, N^t_{k+1,h} \mid v_{1,j-1}, N^s_{j,h}, v_{h+1,n}) \\
&\qquad P(v_{k+1,h} \mid N^i_{j,k}, N^t_{k+1,h}, v_{1,j-1}, N^s_{j,h}, v_{h+1,n}) \\
&\qquad P(N^p_{j,f}, N^q_{f+1,k} \mid N^i_{j,k}, N^t_{k+1,h}, v_{1,j-1}, N^s_{j,h}, v_{k+1,n}) \\
&\qquad P(v_{j,f} \mid N^p_{j,f}, N^q_{f+1,k}, N^i_{j,k}, N^t_{k+1,h}, v_{1,j-1}, N^s_{j,h}, v_{k+1,n}) \\
&\qquad P(v_{f+1,k} \mid N^p_{j,f}, N^q_{f+1,k}, N^i_{j,k}, N^t_{k+1,h}, v_{1,f}, N^s_{j,h}, v_{k+1,n}) \\
&= \frac{1}{P(v_{1,n})} \sum_{j,k,t,h,f} P(v_{1,h-1}, N^s_{h,k}, v_{k+1,n}) P(N^i_{j,k}, N^t_{h,j-1} \mid N^s_{h,k}) \\
&\qquad P(v_{h,j-1} \mid N^t_{h,j-1}, N^s_{h,k}) P(N^p_{j,f}, N^q_{f+1,k} \mid N^i_{j,k}, N^s_{h,k}) \\
&\qquad P(v_{j,f} \mid N^p_{j,f}, N^i_{j,k}) P(v_{f+1,k} \mid N^q_{f+1,k}, N^i_{j,k}) \\
&\qquad + P(v_{1,j-1}, N^s_{j,h}, v_{h+1,n}) P(N^i_{j,k}, N^t_{k+1,h} \mid N^s_{j,h}) \\
&\qquad P(v_{k+1,h} \mid N^t_{k+1,h}, N^s_{j,h}) P(N^p_{j,f}, N^q_{f+1,k} \mid N^i_{j,k}, N^s_{j,h}) \\
&\qquad P(v_{j,f} \mid N^p_{j,f}, N^i_{j,k}) P(v_{f+1,k} \mid N^q_{f+1,k}, N^i_{j,k}) \qquad (6) \\
&= \frac{1}{P(v_{1,n})} \sum_{j,k,t,h,f} \alpha_s(h,k) P(N^s \rightarrow N^i N^t) \beta^s_t(h, j-1)
\end{aligned}
$$

$$
\begin{aligned}
&P(N^i \rightarrow N^p N^q \mid N^s)\beta_p^i(j,f)\beta_q^i(f+1,k) \\
+&\alpha_s(j,h)P(N^s \rightarrow N^i N^t)\beta_t^s(k+1,h) \\
&P(N^i \rightarrow N^p N^q \mid N^s)\beta_p^i(j,f)\beta_q^i(f+1,k)
\end{aligned} \tag{7}
$$

Here Equation 5 formally defines what we count as an example of the event we are looking for. The next version breaks it apart into the pieces we need, and the next, Equation 7 introduces independence assumptions appropriate for our pseudo context-sensitive PCFGs. These are the same as those for standard PCFGs, except whenever we have the opportunity to condition on the parent of the conditioning non-terminal, we do so. Finally, Equation 7 replaces the various terms of the equaiton with abbreviations for the required probabilities. The outside probabilities ($\alpha_l(m,n)$) should be familiar to those acquainted with the inside-outside algorithm. The probabilities of rules is unchanged, except that when we can, we condition on the parent of the right-hand-side non-terminal. Finally we have introduced a new symbol, $\beta_x^y(j,k)$, that is the inside probability $\beta_x(j,k)$, conditioned on the fact that the parent of N^x is N^y. It can be shown that this last probability is computable in polynomial time (and, to be specific, in the time required to parse the sentence).

Note how Equation 7 uses the term $P(N^i \rightarrow N^p N^q \mid N^s)$, the probability we wish to estimate. The same thing happens in the more traditional use of the inside-outside algorithm, where the expression for the counts on a rule's usage involves the probability of the rule. The idea is that one makes an initial estimate of this probability and the inside-outside algorithm modifies this estimate to bring it closer to what it sees in the training corpus. This revised number can be fed in again, leading to the iterative nature of the scheme. The same thing happens here. Fortunately there is a reasonably straight-forward number to use here for the initial estimate, namely the probability of the rule independent of the parent node N^s. Thus the first time through the algorithm we use the unmodified probabilities of the rules. Subsequently the estimates computed on the previous iteration would be used in the next go-round. (However, in our experiment our training corpus was too small for this. Overfitting of data occurred after the first iteration.)

It is not too hard to see how equation 7 translates into a form that is not dependent on the CFG being in Chomsky-normal form. We omit this transformation for the sake of brevity. The version implemented, however, is the general one.

4 Sparse Data

We now want to apply pseudo context-sensitivity to improving the performance of our grammars according to the measure given in Equation 2. Before we can do so, however, we need to overcome one more problem, sparse data.

While the context-sensitive technique we have developed work for any context-free grammar, it was developed in conjunction with our work on grammar induction. The grammars we produce in our learning scheme are typically quite large, as we have sacrificed expressiveness of our grammar formalism in exchange for ease of learning. Thus a typical grammar is about 3500 rules or so. As we have 20 non-terminals our pseudo context-sensitive rules require about 70,000 ($= 3500 \cdot 20$) parameters. As the corpus we have been using has about 300,000 words, and figuring about one rule application per word, it is clear that we do not have enough data to reliably estimate all of these parameters, particularly as some rules are quite rare.

Thus, rather than use the context-sensitive parameters "raw", we smooth them by mixing them in with the non-context-sensitive rule probabilities. The equation we use for the smoothing is

$$P(N^i \rightarrow \alpha^j \mid \rho(N^i) = N^s) = \sigma_1 P(\rightarrow \alpha^j)\rho(N^i) = N^s + \sigma_2 P(N^i \rightarrow \alpha^j) \quad (8)$$

where $\sigma_1 + \sigma_2 = 1$ are the mixing constants. In a smoothing equation like Equation 8 the σ_is can be functions of the conditioning terms in the probability distribution we are calculating. In fact, as the probability of a rule is implicitly conditioned on the left-hand side of the rule (note that the probabilities of rules sum to one for each left-hand side), our σ_is can be functions of both N^i and N^s. The only requirement is that for each pair, N^i, N^s, we still have $\sigma_1(N^i, N^s) + \sigma_2(N^i, N^s) = 1$.

We have used a reasonably standard method for finding optimal settings for the σ_is. We split our training data into two pieces, and used one piece, together with the inside-outside algorithm to estimate our context-sensitive and context free probabilities, $P(N^i \rightarrow \alpha^j)\rho(N^i) = N^s$ and $P(N^i \rightarrow \alpha^j)$. The optimal settings for the σ_is are those which make their probability-weighted sums as high as possible; we used an iterative search procedure and the remaining data to find approximately optimal settings for the σ_is.

5 Results

Before giving the results it is necessary to establish some kind of yardstick for performance. We have suggested above that cross entropy per tag is the right number, but this figure is not suitable for comparing competing models. The difficulty is that it can vary widely with the training sentences, the tag set used, and the accuracy of the tagging. Since an absolute number is not suitable, we supply comparison figures between our model and a sort of industrial standard, the tri-tag model, trained on the same sentences, with the same tags. (The tri-tag model is one in which each tag is predicted according to the probability of getting that tag given the two previous tags.) The tri-tag model (or often a bi-tag model) has been very successful at language modeling, and is the typical model of choice in tagging models such as those in [3,7,8,9,12].

What would correspond to a good improvement in the cross-entropy of the tag sequence? To get some idea of this we took one of our best pure PCFGs and generated an artificial corpus from the grammar. We then compared the cross entropy the correct grammar assigned to the tag sequence with that assigned by a tri-tag model. We found that the correct grammar is only .15 bits/tag better than the tri-tag model (2.65 vs 2.80 bits/tag). In our learning work we have been aiming at an improvement over tri-tag of about half of that, in light of the difficulties presented by the complexity of real English, limited availability of data, and limited computational resources.

We derived our context-sensitive grammar from a PCFG developed from related work on grammar induction. This latter grammar was learned on the basis of a 300,000 word training corpus, consisting of all sentences in the tagged Brown Corpus [11] of length less than 23, and not containing certain terminals we wished to ignore (most notably parentheses, foreign words and titles). We built the context sensitive version of this grammar by applying Equation 7 and training over the same corpus from which the PCFG was learned. Our results are obtained using a corpus of 10,000 words drawn from the same source, reserved for testing. Both the context-sensitive and context-free grammars assigned some parse to 99.5% of the words in the testing corpus (99.6% of the sentences). Unparsed sentences are ignored when collecting further data. As the exact same sentences are unparsed by both the context-free and context-sensitive grammars, and the percentage of unparsable sentences is .4% it does not seem likely that these sentences are influencing the results given here.

The results of using our context-sensitive probabilities is shown in Figure 3. In all cases we show the improvement over the tri-tag model, which had per-tag cross-entropy of 2.738 bits/tag. (Thus 0 bits/tag in our graph would correspond to a grammar that is no better or worse than the tri-tag model on average.) The left-most entry (.039 bits/tag) shows the results we obtained prior to the use of the context-sensitive statistics. The right-most (.072 bits/tag) shows what was obtained after their use. The difference, .033 bits/tag, is quite large, at least when compared to the goal of a .075 bits/tag improvement.

The intermediate figures are also of some interest. The first, labeled "crude," was obtained getting the counts for the probabilities using the viterbi approximation rather than the correct Equation 7. The second, labeled "statistics fixed" was obtained using Equation 7 but smoothed the probabilities using a "seat of the pants" guess for the σ_i's ($\sigma_1 = 0$ if the combination of N^i and N^s was seen less than 1000 times, .6 otherwise). Using optimal σ_i's gave the right-most, final, figure. As can be seen, attending to such details does make a difference.

We do not indicate computational resources expended in context-free vs pseudo context-sensitive as there is no significant difference in this regard. The actual parsing is the same in both cases, the only difference appearing after the parse when calculating the probabilities of the tag sequence. While parsing and both probabilistic calculations have big-O complexity n^3, in fact actual time is

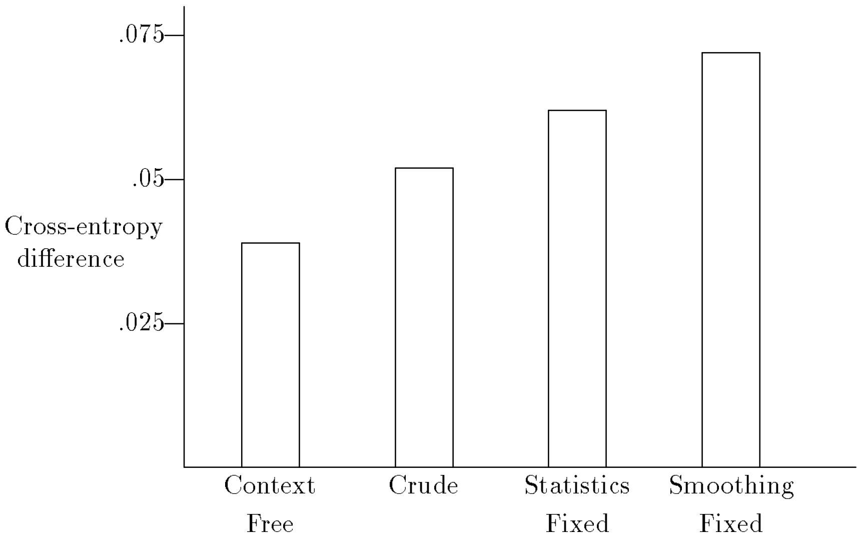

Figure 3: Per-word cross-entropy of held-out data with/without context-sensitive probabilities

dominated by the former, as the probabilistic calculations are quite simple.

6 Analysis

We now turn to the question of from whence this .033 bits/tag improvement arises. Roughly speaking there are two possible hypotheses. The first is that the context sensitivity sharpens the probabilities across the board. The second is that there are particular situations where it is important to know the context in which a rule occurs, and these provide the lion's share of the benefit. Our initial hypothesis was that the second of these would prove to be the case. In this section we offer evidence that this is so.

We start by remembering that for each parent s and child t there is a distinct distribution for the rules R_t that expand the non-terminal t. This distribution is $P(R_t \mid s, t)$. It gives the probability that t in the context of s is expanded using each $r \in R_t$. The question we pose for each s, t pair is "Is $P(R_t \mid s, t)$ significantly different from $P(R_t \mid \neg s, t)$?" If the difference is large, then the context sensitive technique is buying us a lot in the situation in which s is the parent of t. We estimate significant difference using a likelihood ratio analysis described in [10].

The data for our estimate are the number of times that each of the k rules $r \in R_t$ is used when s is the parent of t, which we designate $C_1(s, t) = \{c_{1,1}, c_{1,2}, \ldots, c_{1,k}\}$, and similarly for the number of times when s is *not* the parent of t, which we designate $C_2(s,t) = \{c_{2,1}, c_{2,2}, \ldots, c_{2,k}\}$.

We estimate $P(R_t \mid s,t)$ using the "obvious" choice:

$$P(r_i \mid s,t) = \frac{c_{1,i}}{\sum_{j=1}^{k} c_{1,j}} \tag{9}$$

(and similarly for $P(r_i \mid \neg s, t)$).

Loosely speaking, we compare the chance of seeing our data, C_1 and C_2, given that the distributions are distinct, versus the chance of seeing the data, given that the distributions are really the same. We name the former hypothesis

$$H(P(R_t \mid s,t), P(R_t \mid \neg s,t), C_1, C_2) \tag{10}$$

In the latter case, we have

$$H(P(R_t \mid t), P(R_t \mid t), C_1, C_2) \tag{11}$$

since in this case

$$P(R_t \mid s,t) = P(R_t \mid \neg s,t) = P(R_t \mid t)$$

Finally, following [10] we consider the quantity

$$-\log \lambda(s,t) = -\log \left[\frac{H(P(R_t \mid t), P(R_t \mid t), C_1, C_2)}{H(P(R_t \mid s,t), P(R_t \mid \neg s,t), C_1, C_2)} \right] \tag{12}$$

We lack space to show an exact form for H and $-\log \lambda(s,t)$ (but see [10] for details). Intuitively, however, this is a measure of how likely it is that the context sensitive probabilities for the rules given s, t are really just the context-free probabilities. The advantage of this quantity for our purposes is that it can be computed exactly, starting from the multinomial distribution, and thus is accurate even in the presence of rare events, which, if we may be excused the oxymoron, are quite common in our data. (Many of the rules occur less than ten times in our data. Thus the number of times we would expect them to occur with a particular parent s may well be less than one.)

Note, also, that in most normal circumstances $-\log \lambda(s,t)$ grows linearly in the number of times s, t are observed together, $C_1(s,t)$. Intuitively this captures the idea that more data allows one to make finer discriminations. The other contributing factor, naturally, is the difference between the observed distributions $P(R_t \mid s,t)$ and $P(R_t \mid \neg s,t)$. Because of this we decided to plot $-\log \lambda(s,t)$ against $C_1(s,t)$, with one point for each s, t combination. If the result were a straight line it would indicate that the various $P(R_t \mid s,t)$ distributions differed to approximately the same degree from their context-free equivalents, $P(R_t \mid t)$, and that the difference in the $-\log \lambda(s,t)$ is just due to having more data for some points, the larger $C(s,t)$'s, than others.

The results shown in Figure 4 are quite different. While there is clearly

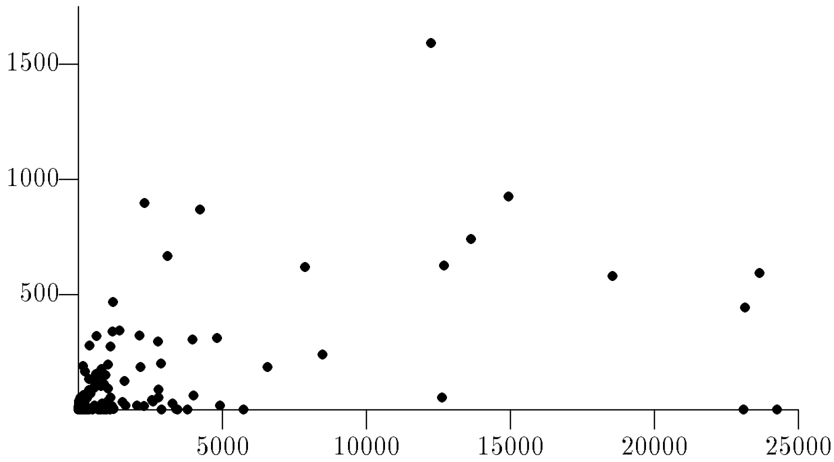

Figure 4: Plot of $-\log \lambda(s,t)$ against $C(s,t)$

a positive correlation between $C_1(s,t)$ and $-\log \lambda(s,t)$, it is hardly a straight line. Instead a quick glance at the chart suggests that a small number of s,t combinations account for the overwhelming majority of the context-sensitive effect.

To further test this hypothesis, we modified the smoothing equation 8 to assign a σ_1 of zero when calculating the probability of a rule given an s,t pair not in the top n pairs, when sorted by $-\log \lambda(s,t)$. Figure 5 shows that by the time we have considered 51 out of the 400 s,t combinations we have captured virtually all of the context-sensitive effect, and even by 21 s,t combinations (5% of the data) we have most (80%) of the effect. This suggests that our initial hypothesis, that the effect is concentrated in a small number of cases, is basically correct.

Recall that our pseudo PCSG is not truly context-sensitive, because it does not move out of the range of languages generated by PCFGs, and it is possible to "compile out" our context-sensitive statistics. Since performance benefits are concentrated in a small number of s,t pairs, and the compilation procedure can be carried out incrementally, on a per s,t pair basis, the transformation appears to be practical. The worry here is that the grammar might be so large as to be useless. Even limiting ourselves to the 20 best s,t pairs, adding a non-terminal for each pair doubles the number of non-terminals in our grammar, and, in the worst case, could cause an exponential blow-up in the number of rules. Further, we did not smooth the expanded grammar as we did for the pseudo PCSG.

Nonetheless, the observed localization was encouraging, and back-of-the-envelope calculations suggested that the expanded grammar would be only about 10,000 rules, which is a manageable size. We carried out the experiment of transforming the grammar and evaluating its performance. Ignoring

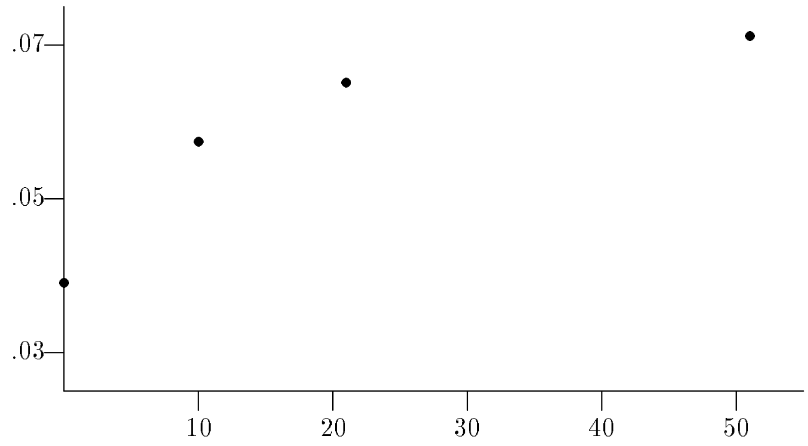

Figure 5: Cross entropy as a function of increasing numbers of s, t's

rules with zero probability, the transformation added 5384 rules to the existing 3500. This more than doubles the size, but it was actually less of an increase than expected. After two iterations of the inside-outside algorithm, the grammar began to overfit the training data, but performance reached the same level as that of the pseudo PCSG. To be precise, the trained PCFG showed an *improvement* over the pseudo PCSG of 0.001 bit per word, even though it had fewer parameters (about 8,000 vs. 17,000). We do not regard this improvement as significant, but the fact the PCFG can recover the missing 20% performance gain is very satisfying.

This experiment shows that our scheme can be used as a novel form of PCFG induction, one which adds both rules and non-terminals, and revises the probabilities to produce significantly lower cross entropy. Adding non-terminals is a particularly sticky problem for grammer learners, as the unconstrained space is too large to search. What is usually done is to fix the number of non-terminals in some other way, either using outside sources of information [4], or, as we do, via a restricted formalism [5]. Another approach, suggested in [13] is to use a CNF grammar, simply guess an upper bound on the number of non-terminals, and deploy a grammar minimization procedure periodically during the grammar training. The appeal of our approach is that it does not require guesses, but can automatically identify a set of promising new non-terminals, and associated rules and probabilities. It is, admittedly, highly constrained, but we regard this as more of a feature than a drawback. Overly large search spaces require learners to deploy constraints. Our procedure is restricted enough to be feasible for a large problem (English), but loose enough to allow significant performance gains.

7 Conclusion

We have presented a PCFG model in which the probability of a rule also depends on the parent of the node being expanded. The scheme is applicable to any PCFG and the equations we have derived allow one to collect the necessary statistics without requiring preparsed data. In the experiment we ran, the improvement over the context-free version is quite large, given the expected range. We have analyzed the context-sensitive statistics, and shown that most of the effect is fairly localized.

This localization encouraged us to attempt to use statistics gathered for our pseudo PCSG to extend our original PCFG. By using the most promising s, t pairs, we demonstrated that expanding the grammar retains the performance gains of the pseudo PCSG, despite the reduction in the number of parameters. This implies that our scheme is not only good for improving performance by means of a pseudo PCSG, but it may also be viewed as a systematic means for inducing non-terminals, rules, and probabilities for a PCFG.

References

1. Baum, L. E. An inequality and associated maximization technique in statistical estimation for probabilistic functions of a Markov process. *Inequalities 3* (1972), 1–8.
2. Black, E., Jelinek, F., Lafferty, J., Magerman, D., Mercer, R. and Roukos, S. *Towards history-based grammars: using richer models for probabilistic parsing.* In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics.* 1993, 31–37.
3. Boggess, L., Agarwal, R. and Davis, R. *Disambiguation of prepositional phrases in automatically labeled technical text.* In *Proceedings of the Ninth National Conference on Artificial Intelligence.* AAAI Press/MIT Press, Menlo Park, 1991, 155–159.
4. Briscoe, T. and Waegner, N. *Robust stochastic parsing using the inside-ouside algorithm.* In *Workshop Notes, Statistically-Based NLP Techniques.* AAAI, 1992, 30–53.
5. Carroll, G. and Charniak, E. *Learning probabilistic dependency grammars from labeled text.* In *Working Notes, Fall Symposium Series.* AAAI, 1992, 25–32.
6. Charniak, E. *Statistical Language Learning.* MIT Press, Cambridge, 1993.
7. Charniak, E., Hendrickson, C., Jacobson, N. and Perkowitz, M. *Equations for part-of-speech tagging.* In *Proceedings of the Eleventh National Conference on Artificial Intelligence.* AAAI Press/MIT Press, Menlo Park, 1993, 784–789.

8. Church, K. W. *A stochastic parts program and noun phrase parser for unrestricted text.* In *Second Conference on Applied Natural Language Processing*. ACL, 1988, 136–143.

9. DeRose, S. J. Grammatical category disambiguation by statistical optimization. *Computational Linguistics 14* (1988), 31–39.

10. Dunning, T. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics 121* (1993), 61–74.

11. Francis, W. N. and Kučera, H. *Frequency Analysis of English Usage: Lexicon and Grammar*. Houghton Mifflin, Boston, 1982.

12. Jelinek, F. *Markov source modeling of text generation.* In *The Impact of Processing Techniques on Communications*, J. K. Skwirzinski, Ed. Nijhoff, Dordrecht, 1985.

13. Lari, K. and Young, S. J. *The estimation of stochastic context-free grammars using the Inside-Outside algorithm.* In *Computer Speech and Language*. vol. 4, 1990, 35–56.

14. Magerman, D. M. and Weir, C. *Efficiency, robustness and accuracy in Picky chart parsing.* In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*. 1992, 40–47.