Parsing with Context-Free Grammars and Word Statistics

Eugene Charniak

Department of Computer Science
Brown University
Providence, Rhode Island 02912

CS-95-28
September 1995

Parsing with Context-Free Grammars and Word Statistics*

Eugene Charniak
Department of Computer Science, Brown University

May 23, 1995

Abstract

We present a language model in which the probability of a sentence is the sum of the individual parse probabilities, and these are calculated using a probabilistic context-free grammar (PCFG) plus statistics on individual words and how they fit into parses. We have used the model to improve syntactic disambiguation. After training on Wall Street Journal (WSJ) text we tested on about 200 WSJ sentence restricted to the 5400 most common words from our training. We observed a 41% reduction in bracket-crossing errors compared to the performance of our PCFG without the use of the word statistics.

1 Introduction

Modern statistical language processing offers a new window on the old problem of syntactic disambiguation — finding the correct syntactic structure among the many that are possible. This research program typically starts with the notion of a "language model." This is an assignment of probabilities to all strings in the language. Suppose that a corpus consists of a sequence of n sentences $s_{1,n}$. We assume that the probabilities of sentences are independent, so the probability of our corpus is the product of the sentence probabilities:

$$p(s_{1,n}) = \prod_{i=1}^{n} p(s_i) \tag{1}$$

We now turn to the probability of an individual sentence s and relate it to the possible parses of the sentence. Let $p(s, \pi)$ be the probability of a sentence s

*This research was supported in part by NSF contract IRI-9319516 and ONR contract N0014-91-J-1202.

under a particular parse π, and $\mathcal{P}(s)$ be the set of all parses for s, then

$$p(s) = \sum_{\pi \in \mathcal{P}(s)} p(s, \pi) \tag{2}$$

From this we can define syntactic disambiguation as finding the parse that maximizes $p(s, \pi)$:

$$\arg \max_{\pi \in \mathcal{P}(s)} p(s, \pi) \tag{3}$$

The simplest example of this approach is using a probabilistic context-free grammar (PCFG) to assign parse probabilities. With PCFGs we calculate the probability in a very simple way. The probability of a parse is the product of the probabilities of the PCFG rules used in the parse. As such the technique is only able to distinguish between very common and uncommon constructions in the language. However, Equation 3 works for any method of calculating the parse probabilities, and if basing the probability on only the rules is insufficient then adding word information might improve things.

One example of how adding word-level information can aid disambiguation is shown in [9], which deals with prepositional phrase attachment in cases where the pp can attach to either a verb, or the direct object of the verb. They start by finding cases where the prepositional phrase attachment is unambiguous (or nearly so) and collect statistics on how often a particular preposition attached to a particular noun, C(n, prep), and how often it attached to a particular verb, C(v, prep). (E.g., in "The children sold cookies in the classroom" one would use C(cookies, in), and C(sold, in).) These counts are normalized to make probabilities and the probabilities are used to make decisions.

In this paper we take this idea further and apply word statistics at all points in the parse. While this seems like an important thing to do, there are relatively few papers in this area [1,10,11,16]. Section 6 compares our work to earlier research.

2 The Model

First some notation. A sequence of words in a sentence from the j'th word to the k'th is denoted by $w_{j,k}$. A constituent, say an np, which dominates these words is denoted by $\mathsf{np}_{j,k}$. If we wish to make the constituent type a variable, $N^i_{j,k}$ denotes the non-terminal i that dominates $w_{j,k}$. Conversely, N^i denotes a constituent of type i without reference to the words it covers.

Next, consider a constituent, c, in the parse π. Let the parent constituent of c be $\rho(c)$. The grandparent of c, $\rho(\rho(c))$ we abbreviate to $\rho^2(c)$. For example, in Figure 1 if c is the constituent labeled "np:rocks" then $\rho(c)$ is the pp, and $\rho^2(c)$ is the vp. We also assume that within a parse each constituent has a *head.* The head of a terminal is its lexical item. The head of a non-terminal is, informally,

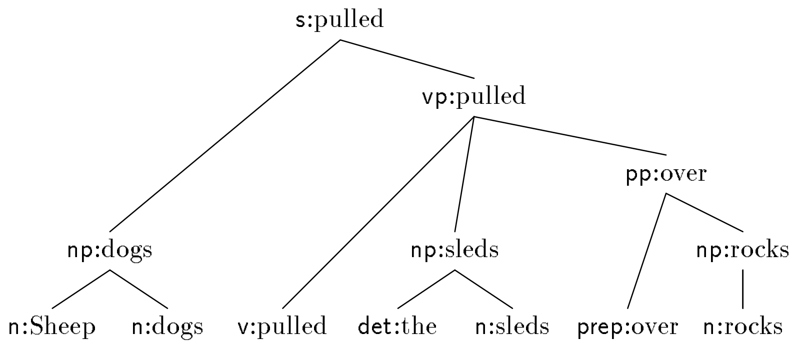

Figure 1: Phrase marker for "Sheep dogs pulled the sleds over rocks"

the most important word of the phrase, so the head of an np is the head noun, of a vp is the main verb, of a pp, the preposition, etc. More formally, every context-free grammar rule must specify which of its sub-constituents provides its head. We denote the head of a constituent by $h(c)$. We abbreviate $h(\rho(c))$ as $h\rho(c)$, and $h(\rho^2(c))$ as $h\rho^2(c)$. $t(c)$ denotes the constituent type of c (e.g., if c is a prepositional phrase, $t(c) = \mathsf{pp}$), $r(c)$ denotes the PCFG rule used to expand c (e.g., if c is the constituent in Figure 1 labeled "vp:pulled" then $r(c) = \mathsf{vp} \rightarrow \mathsf{v}$ np pp) and $a(c)$ indicates if c appears *after* the head in c's parent phrase (more on this point when we explain the reasoning behind this particular model). All of the definitions are relative to a parse.

We now formally define our probabilistic model for the probability of a parse π for the sentence s as follows:

$$p(\pi, s) = \prod_{c \in \pi} p(h(c) \mid t(c), h\rho(c), h\rho^2(c), a(c)) \cdot p(r(c) \mid h(c)) \tag{4}$$

We can think of this equation as saying that we predict the head of a constituent on the basis of information from higher constituents, and we then predict the rule used to expand the current constituent from the head just predicted.

There are two questions of interest with regard to Equation 4, (a) does it indeed define a probability distribution over sentences and their parses, and (b) can we compute the probability of a sentence in polynomial time? We will briefly consider the first here. We derive the equations for polynomial-time probability calculations in Appendix A.

To see that Equation 4 does, in fact, define a probability distribution over sentences and their parses, let us consider how this model handles the sentence "Sheep dogs pulled the sleds over rocks" under the parse π of Figure 1. We first note that the probability of the sentence and the parse is equivalent to the probability of having all of the constituents of the parse,all of the words, and the words in the correct order.

$$\begin{aligned} p(w_{1,7}, \pi) \quad = \quad & p(\mathsf{s}_{1,7},\ h(\mathsf{s}_{1,7}) = \text{pulled},\ \mathsf{np}_{1,2},\ h(\mathsf{np}_{1,2}) = \text{dogs}, \\ & \mathsf{n}_{1,1},\ h(\mathsf{n}_{1,1}) = \text{sheep},\ \mathsf{n}_{2,2},\ h(\mathsf{n}_{2,2}) = \text{dogs}, \\ & \mathsf{vp}_{3,7},\ h(\mathsf{vp}_{3,7}) = \text{pulled}, \ldots) \end{aligned} \tag{5}$$

The right-hand side specifies both the parse and the sentence. To see this, first note that it specifies all of the constituents of the parse, and thus the parse tree. Furthermore, we assume that all terminal symbols in a parse are the heads of at least one constituent, thus specifying the heads of all of the constituents pins down the words of the sentence. Finally the order of the words are fixed by the ordering of the constituents.

We can then break Equation 5 up into a sequence of conditional probabilities as follows:

$$\begin{aligned} & p(w_{1,7}, \pi) = \\ & \quad p(\mathsf{s}_{1,7}) \cdot p(h(\mathsf{s}_{1,7}) = \text{pulled} \mid \mathsf{s}_{1,7}) \\ & \quad p(\mathsf{np}_{1,2}, \mathsf{vp}_{3,7} \mid h(\mathsf{s}_{1,7}) = \text{pulled},\ \mathsf{s}_{1,7}) \\ & \quad p(h(\mathsf{np}_{1,2}) = \text{dogs} \mid \mathsf{np}_{1,2},\ \mathsf{vp}_{3,7},\ h(\mathsf{s}_{1,7}) = \text{pulled},\ \mathsf{s}_{1,7}) \\ & \quad p(\mathsf{n}_{1,1},\ \mathsf{n}_{2,2} \mid h(\mathsf{np}_{1,2}) = \text{dogs},\ \mathsf{np}_{1,2},\ \mathsf{vp}_{3,7},\ h(\mathsf{s}_{1,7}) = \text{pulled},\ \mathsf{s}_{1,7}) \\ & \quad p(h(\mathsf{n}_{1,1}) = \text{sheep} \mid \mathsf{n}_{1,1},\ \mathsf{n}_{2,2},\ h(\mathsf{np}_{1,2}) = \text{dogs},\ \mathsf{np}_{1,2},\ \mathsf{vp}_{3,7},\ \ldots) \\ & \quad \ldots \end{aligned} \tag{6}$$

We now use the independence assumptions implicit in Equation 4, namely:

1. the probability of expanding a constituent c using a given rule is independent of everything except the head of c
2. the probability of a constituent c having a given head is independent of everything except the head of c's parent, grandparent, the type of c, and whether c appears before or after the head of its parent.

This gives us the following:

$$\begin{aligned} & p(w_{1,7}, \pi) = \\ & \quad p(\mathsf{s}_{1,7}) \cdot p(h(\mathsf{s}_{1,7}) = \text{pulled} \mid \mathsf{s}_{1,7}) \\ & \quad p(\mathsf{s} \ \rightarrow\ \mathsf{np\ vp} \mid h(\mathsf{s}_{1,7}) = \text{pulled}) \\ & \quad p(h(\mathsf{np}_{1,2}) = \text{dogs} \mid \mathsf{np}_{1,2},\ h(\mathsf{s}_{1,7}) = \text{pulled}) \\ & \quad p(\mathsf{np} \ \rightarrow \mathsf{n\ n} \mid h(\mathsf{np}_{1,2}) = \text{dogs}) \\ & \quad p(h(\mathsf{n}_{1,1}) = \text{sheep} \mid \mathsf{n}_{1,1},\ h(\mathsf{np}_{1,2}) = \text{dogs},\ h(\mathsf{s}_{1,7})\ \text{pulled},\ a(\mathsf{n}_{1,1}) = 0) \\ & \quad \ldots \end{aligned} \tag{7}$$

This can be seen to be in the form of Equation 4

Word	Rule	Example
gave	vp $\rightarrow$ v np pp	"gave the cookie to Bred"
	vp $\rightarrow$ v pron np	"gave him the cookie"
	vp $\rightarrow$ v np np	"gave Fred the cookie"
	vp $\rightarrow$ v np np pp	"gave Fred the cookie on Monday"
tell	vp $\rightarrow$ v pron s	"tell her I left"
	vp $\rightarrow$ v pron wh-phrase	"tell her what you know"
	vp $\rightarrow$ v pron pp	"tell her on Tuesday"
	vp $\rightarrow$ v pron	"tell her"
	vp $\rightarrow$ v pron wh to-phrase	"tell her what to do"
	vp $\rightarrow$ v wh-phrase	"tell what you know"
	vp $\rightarrow$ v pron np	"tell her the story"

Figure 2: Significant rules for the verbs "gave" and "tell"

Another way to understand Equation 4 is to see how the extra conditioning information allows us to sharpen the probabilities of words and rules. Let us start with the simpler term on the right, $p(r(c) \mid h(c))$. Here we are conditioning the rule used to expand c on its head. The most common examples of this dependence is for verb case frames. For example, the rule vp $\rightarrow$ v np np is not all that common, and thus the probability assigned to it by a pure PCFG is low (.0038 in our grammar). However, if the head (the main verb) is "gave" $p(\mathsf{vp} \rightarrow \mathsf{v\ np\ np} \mid gave)$ should be relatively high (and in our grammar is, .052). Indeed, we pulled out rule/verb pairs where the probability of the rule given the verb was significantly higher than the prior on the rule. (The exact measure used here is that described in [8].) The results looked much like the studies directed at finding verb case-frames such as [2,12]. Figure 2 shows the four rules for "gave" and the seven for "tell" which differ most from the original PCFG probabilities for the rules.

Let us now consider in detail the first term on the right of Equation 4: $p(h(c) \mid t(c), h\rho(c), h\rho^2(c), a(c))$. We focus our discussion by seeing what happens as we condition on more and more information. The need to condition on $t(c)$ should be obvious. Clearly $p(the)$ will not be as accurate as $p(the \mid \mathsf{det})$, that is, the probability that the head is "the" given that we know it is a determiner.

The most important step is conditioning on the head of the parent as well: $p(h(c) \mid t(c), h\rho(c))$. It is this step that gives the prepositional-phrase attachment work of [9] its power. For example, should the pp "over rocks" in Figure 1 attached to "pulled" or "sleds?" Equation 4 applied to the pp in effect contrasts the two conditional probabilities $p(over \mid \mathsf{pp}, pulled)$ and $p(over \mid \mathsf{pp}, sleds)$. This is exactly what [9] does with its counts of how often a particular pp attaches to either the preceding noun or verb. In a similar way, this sort of conditioning

fire weapon: bullets cannon gas guns missile missiles rifles rockets shot shots weapons

fire employee: agency aide auditor chairman cities consultants controllers director employee employees executives firm house iacocca inc manager marwick north o'donnell officer officers officials oliver people president saunders secretary worker workers

non-object np's: day december month months tuesday week weeks year yesterday

pronoun np's: 16 anyone two everyone

Figure 3: Heads of noun-phrases following the verb "fired"

will help to sharpen the probability of seeing "sheep" as a modifier of "dog" in the np "sheep dogs" in Figure 1 since presumably $p(sheep \mid \mathsf{n}, dog)$ is much larger than $p(sheep \mid \mathsf{n})$.

To take another example of the kinds of information captured by this term, consider the following:

$$p(h(c) \mid t(c) = \text{noun},\ h\rho(c) = \textit{fired},\ a(c) = 1) \tag{8}$$

Intuitively this would correspond to the heads of noun phrases which follow the verb "fired." Figure 3 lists all such words who's associated probability is greater than .004. They have been manually grouped into intuitive categories. While we have grouped these manually, work such as [14] suggests how to find such groupings automatically given data such as ours.

To give another example of direct objects, Figure 4 gives all of the heads of noun-phrases following "built" such that their probability given "built" is greater than .002. As with the nouns following "fired" we have mostly substantive nouns with some nouns acting as pronouns (e.g., "lot," and "one") and some heads of time noun-phrases (e.g., "week," "weeks," and "year").

Conditioning on the head of the grandparent constituent is again inspired by prepositional-phrase attachment. Where a pp attaches also depends on the compliment of the pp (consider "Joan bought flowers with Sue/yellow stems"). In Figure 1 this information should help find the attachment for the pp "over rocks" assuming that $p(rocks \mid pulled, over)$ is larger than $p(rocks \mid sleds, over)$.

The distinction between subjects and objects of verbs is the motivation for conditioning on whether the word appears before or after the head of its parent. For example, "She hit her" is more common than "Her hit she." The model captures distinction as $p(she \mid \mathsf{pron}, hit, \mathsf{before})$ is much larger than $p(she \mid \mathsf{pron}, hit, \mathsf{after})$.

aircraft airport apartments assets barrick barrier base batch bed bonds bridge building buses business businesses canal capital car career cars center chain chip church company computers conglomerate dam desk empire expectations facilities factories factory firm fortune fortunes group headquarters home homes hotel house houses housing inc industries industry inventories jose journal laboratory lead levels line lot machine machines malls mansion miles mill million missiles models navy network networks nothing one operation operations organization park part plane planes plant plants position positions power presence products programs project prototype record refineries relationship relationships reputation reserves rocket sales share ships stadium stake statue strategies success support surpluses system systems tank team ties today tower trucks unit units vehicles versions vessel wall way week weeks wpp wright year

Figure 4: Heads of noun-phrases following "built"

3 Smoothing

While conditioning on the aforementioned features is certainly justified, when taken together we have a serious sparse data problem. Our smoothing takes two forms, the first is based upon domain knowledge, the second is a conventional domain-independent technique.

The domain dependent smoothing comes from our observation that pp attachment is a standard case where the grandparent head is a useful conditioner. Here we simply make the reverse point — grandparent heads may not be that useful in other cases. Thus we make the following smoothing assumption:

$$p(h(c) \mid t(c), h\rho(c), h\rho^2(c), a(c)) = \begin{cases} p(h(c) \mid t(c), h\rho(c), h\rho^2(c)) & \text{if } t(\rho(c)) = \mathsf{pp} \\ p(h(c) \mid t(c), h\rho(c), a(c)) & \text{otherwise} \end{cases} \tag{9}$$

This has the effect of reducing our statistics to a bigram level in most situations.[1]

Our domain independent smoothing is of the traditional "back off" variety. We have two equations, one for each case in Equation 9. Since they are quite similar we only give the equation for the first case, where grandparent heads are used. As before $C(a, b, \ldots)$ is the number of times $a, b, \ldots$ occur together.

$$p(h(c) \mid t(c), h\rho(c), h\rho^2(c)) =$$
$$\lambda_1(C(h\rho(c), h\rho^2(c))) \cdot \hat{p}(h(c) \mid t(c), h\rho(c), h\rho^2(c))$$

[1] Since prepositional phrases headed by "to" are currently marked as a to clause, these also allow for the use of $h\rho^2(c)$. Other possible uses, to be explored in the future, are possessive clauses and "wh" clauses modifying noun-phrases.

$$
\begin{aligned}
&+ \lambda_2(C(h\rho(c),\, t(c))) \cdot \hat{p}(h(c) \mid t(c), h\rho(c)) \\
&+ (1 - \lambda_1(C(h\rho(c), h\rho^2(c))) - \lambda_2(C(h\rho(c),\, t(c)))) \\
&\quad \cdot \hat{p}(h(c) \mid t(c))
\end{aligned} \tag{10}
$$

Here $\hat{p}(a \mid b)$ is the maximum likelihood estimate of the probability $p(a \mid b)$. The intuitive idea is that we consider three estimates of the probability, each using successively less conditioning information. The weight on each is a function of the number of cases observed of this conditioning situation. Note that the first term is the maximum likelihood estimate of the conditional probability we want, and the last is the most basic estimate of the probability of the head knowing only the probability of the word given its part of speech. The values of $\lambda(C(\mathrm{x}))$ are set so that the final probability is closest to the highly conditioned statistics when the conditioning event x is most common, and the closest to the raw word probabilities (or PCFG probabilities) when x is uncommon. In practice the λ s were set subjectively.

4 The Experiment

A context-free grammar developed from work on context-free grammar induction described in [5,6] is the base grammar for the experiment. This grammar was developed using training data from the Penn tree-bank version of the Brown corpus as provided by the ACL Linguistics Data Collection Initiative (ACL-LDCI). The grammar developed is a slight extension of a dependency grammar. A dependency grammar can be thought of as an $\overline{\mathsf{X}}$ grammar with only one level of bars. Thus the rules are all of the form:

$$\overline{\mathsf{x}} \rightarrow \overline{\mathsf{q}}\ \overline{\mathsf{r}} \ldots \mathsf{x}\ \overline{\mathsf{s}} \ldots$$

The head of an $\overline{\mathsf{x}}$ rule is the x constituent on the right-hand side. This form of rule was used for ease of learning. However it is not a compact rule notation, as it requires many combinations of constituents to be "multiplied out". Thus the number of rules in the grammar is quite large, about 4800.

About 36 million words of WSJ text were parsed with the grammar from which we collected the statistics required by the model. When the statistics are pruned to remove all words that do not occur at least twice and all counts less than .15, the file containing the information is .275 Gig.

To test the model we used sentences from parsed ACL-LDCI WSJ text, none of which was used in any previous training. We restricted consideration to sentences that use the 5400 most common words from this corpus. These were the words which appeared in the corpus 500 or more times. As we noted above in our discussion of smoothing, as words and word-combinations get rare, our smoothing equations rely more and more on the original PCFG probabilities, and less and less on the new lexicalized probabilities we were interested in testing. By restricting consideration to the more common words we avoided attenuating the factors we were most interested in measuring - the effects of our new probabilities.

Furthermore we only consider sentences of length less than or equal to 40 words (counting punctuation as words). This removes fairly few sentences, particularly compared to the 5400 word lexicon restriction, which removes about 90% of the possible sentences. The 5400 word lexicon also biases the sample to shorter sentences, so the average sentence length in our test corpus is 16.8 words.

We take as the principle measurement of parsing success bracket-crossing accuracy (henceforth referred to as simply *accuracy*): the percentage of bracketing found in the model's most probable parse that do not cross the bracketing in the LDCI parses. We like this measure because it is relatively insensitive to disagreements about the structures a grammar should assign. However, we are cognizant of its deficiencies; when used alone it can be made artificially high by certain tricks. Thus to keep ourselves "honest" we also report *recall*: the percentage of bracketing found in the LDCI parse that were also in the model's most probable parse. However, our judgments on the significance of the results will be solely in terms of the accuracy measure.

We compare these measures for 6 models:

1. The full model as described earlier ("complete")
2. The full model minus statistics on secondary heads("No $h\rho^2(c)$")
3. The full model minus the statistics of the probability of a rule given the head of the constituent("No $p(r(c) \mid h(c))$")
4. The model *only* using the statistics of the probability of a rule given the head of the constituent and the PCFG probabilities ("Only $p(r(c) \mid h(c))$")
5. The model only using the PCFG probabilities ("PCFG")
6. Creating uniform right-branching parses ("Right Branching").

For each of the models save "Right Branching" we also give the per-word cross entropy ("CrossEnt per Word") it assigns to the sentences (not meaningful outside this paper because of the restrictions on the sentences, but interesting to contrast with parsing success) and a measure of the degree to which the model hones in on one parse, as opposed to spreading the probability over many parses. This measure, "Ratio," is the probability of the sentence divided by the probability of the most probable parse. Assuming we want the model to pick out the correct parse with a high degree of certainty, this number should ideally be close to one.

Probability Model	Percent Accuracy	Error Reduction	CrossEnt per Word	Ratio	Percent Recall
Complete	91.1	41.1	6.54	3.08	80.8
No $h\rho^2(c)$	90.8	39.1	6.73	3.4	80.9
No $p(r(c) \mid h(c))$	89.5	30.5	7.21	4.47	78.7
Only $p(r(c) \mid h(c))$	88.9	26.5	8.09	6.63	79.4
PCFG	84.9		8.75	12.5	76.1
Right Branching	57.2				67.2

Figure 5: Results

5 Results

The major results are shown in Figure 5. If we ignore the question of statistical significance, the picture painted by these figures is reasonably clear: increments in conditioning information give increments in performance, ranging from accuracy of 57.2% for right branching, to 84.9% accuracy for a PCFG using none of the word statistics, to 91.1% accuracy for the model using all of the conditioning information. This latter figure represents at 41.1% reduction in the number of bracket-crossing errors when compared to the pure PCFG.

We give the results for the model without conditioning on $h\rho^2(c)$ because the prepositional-phrase attachment model of [9] does not use this information. Furthermore, (Hindle, personal communication) reports that adding this information actual made their performance worse! Our results paint a different picture. While the improvement shown by adding $h\rho^2(c)$ may not be statistically significant, at least things did not worsen.

Some of the differences between models in Figure 5 are quite small and thus we need to address the question of which of these results are statistically significant. Answering this is not completely straight-forward. For those who are willing to take our word for it, the simple answer is shown in Figure 6. Here an arrow between two models indicates that the model at the tail of the arrow is statistically significantly better than the model at the head. To put these results into words, for the top three models, each model is not significantly better than the model just below it, but is better than the models two or more below it. For those interested in such things, Appendix B gives the analysis which leads to Figure 6.

6 Comparison with Previous Results

Our primary claim involves the improvement over pure PCFG parsing. As such it behooves us to show that the PCFG results are typical of what is obtained

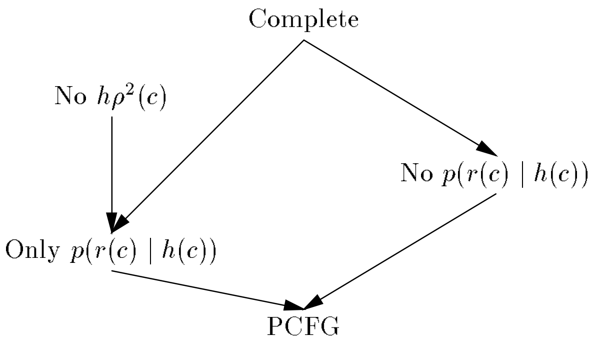

Figure 6: Statistically significant dominance relations between models

by other current non-word-based parsers. In Figure 7 we plot accuracy vs average sentence length for four systems, our PCFG, the transformational parsing systems of [3], the PCFG of [13], and, for completeness, the results for the complete model presented in this paper. It should be clear that the accuracy of our PCFG is in line with these systems. Our PCFG results are also in line with those reported in [15] though their method of reporting results do not allow them to be shown in a graph line that of Figure 7.

The work most similar to our is that of Black et. al. [1] and Schabes and Waters [16]. These papers like ours are concerned with using lexical information together with probabilities on rules to improve parsing performance. Furthermore, these papers also starts by defining a language model in terms of the sum over the probabilities for the parses of a sentence, and selecting the correct parse as the one with the highest probability. Schabes and Waters present a scheme related to Tree-adjoining Grammars, and the burden of their paper is primarily the description of the grammatical formalism and an n-cubed parsing algorithm. No experimental results are given. On the other hand, the the "History-based Grammar" (HBG) model presented by Black et. al. [1] does have related experimental results, so we will concentrate on this paper.

The main difference between our model and HBG is relative complexity. In contrast to the 4 quantities used to condition in our model (with a maximum of three being used at any one time) HBG has 16 with a maximum of 8. Smoothing over this number of conditioning events is done with a combination of decision trees and back-off methods. One extra complexity in HBG that we hope to include in subsequent versions of our model is word clustering.

The figure of merit used in [1] is the percentage of parse trees identical to those in the test corpus except for pre-terminal labels. As such it is hard to directly compare their results and ours. Furthermore the technical manual corpus they used (restricted to the most common 3000 words) is different. Their sentences ranged in length from 7 to 17, rather than our maximum of 40. They

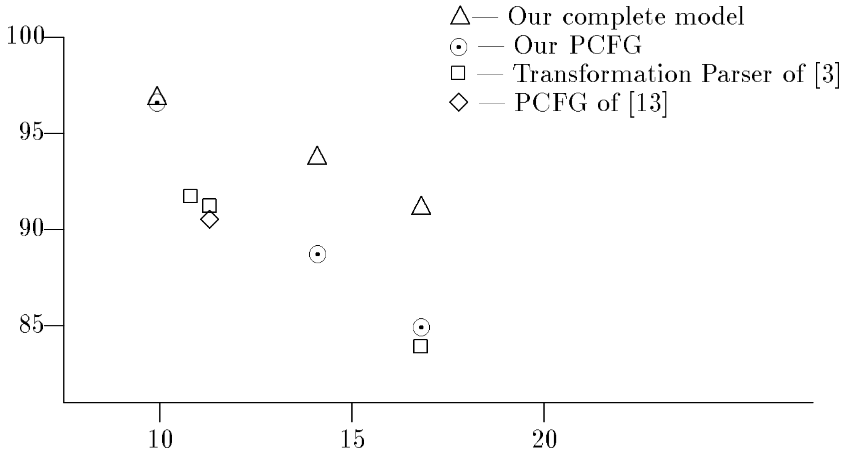

Figure 7: Accuracy vs. average sentence length for four parsers

report that their pure PCFG gave a success rate of 59.8% and their complete model a success rate of 74.6%. Given their much stricter definition of success their results, both for the PCFG and for the enriched model, are better than ours. That the PCFG result is better suggests that some of this might be due to a simpler set of sentences.

Perhaps the biggest difference in their results and ours is in the area of how the performance improve as one increases the number of conditioning events. Besides the main model whose results we have just mentioned Black et. al. describe a simpler model (which is actually close to our model) but report that this model does not do much better than their basic PCFG model, suggesting that there is some threshold of complexity before which significant improvements are not observed. This contrast with our results. Note that the simpler model "No $h\rho^2(c)$" (i.e., the head of the grand-parent is not used) performs almost as well as the full model, and in general the degree of performance improvement decreases as we add more conditioning information, seemingly opposite to the results reported in [1]. It seems evident that this area needs more models evaluated using a wider variety of metrics before clear patterns will appear.

7 Conclusion

We have seen how the use of word-statistics from parsing 38 million words of the Wall-Street Journal helps a parser determine the correct parse for a sentence. The main results of the paper show that the statistics give us a 41.1% reduction in the bracket-crossing error rate.

At the same time the program developed to test the ideas herein needs many improvements. For one thing, the current algorithm is not usable in day-to-day situations because its statistical data-base is so large — about 250 Megabytes. We can not read this into memory, and the scheme we constructed for reading the necessary data off disk does not work nearly well enough. Currently about 90% of parsing time is spent reading in statistics. We also need to better smooth our word data. We intend to cluster words by their statistics and smooth the probability estimates for words by using the statistics for the classes into which they fall [4,7],

Appendices

A Equations for Inside Probabilities

In this appendix we derive the equations for determining in polynomial time the probability of a sentence according to a slightly simplified version of our probabilistic model. That is, we want to calculate the following:

$$p(s) = p(w_{1,n}) = \sum_{\pi \in \mathcal{P}(s)} p(w_{1,n}, \pi) \tag{11}$$

To keep things relatively simple we show how to compute this probability according to the model:

$$p(\pi, s) = \prod_{c \in \pi} p(h(c) \mid t(c), h\rho(c), a(c)) \cdot p(r(c) \mid h(c)) \tag{12}$$

This differs from Equation 4 in that it does not condition on the head of the grandparent of constituent c, $h\rho^2(c)$.

In what follows we recursively compute the two probabilities, both related to the inside probability of a constituent:

$$\beta(N^i_{j,k}) \equiv p(w_{i,j} \mid N^i) \tag{13}$$

The two quantities are

$$\begin{aligned} \beta(N^i_{j,k} \mid \rho) &\equiv p(w_{j,k} \mid N^i, h\rho(N^i_{j,k}), a(N^i_{j,k})) && (14) \\ \beta(N^i_{j,k} \mid h) &\equiv p(w_{j,k} \mid N^i, h(N^i_{j,k})) && (15) \end{aligned}$$

We care about the first of these because if we created a "dummy" head constituent (call it "@") for the non-existent constituent above our s node, and arbitrarily saying that s is "after" the head of its (non-existent) parent, then

$$\begin{aligned} p(w_{1,n}) &= p(w_{1,n} \mid \mathsf{s}) && (16) \\ &= p(w_{1,n} \mid \mathsf{s}, h\rho(\mathsf{s}) = @, a(\mathsf{s}) = 1) && (17) \\ &= p(w_{1,n} \mid \mathsf{s}, h\rho(\mathsf{s}) = @, a(\mathsf{s}) = 1) && (18) \\ &= \beta(\mathsf{s}_{1,n} \mid \rho) && (19) \end{aligned}$$

In what follows let $H(c)$ be all possible heads that constituent c can take under all possible parses. (We discuss later how to compute this set.) Given this set, the relation between $\beta(N^i_{j,k} \mid \rho)$ and $\beta(N^i_{j,k} \mid h)$ is as follows:

$$
\begin{aligned}
\beta(N^i_{j,k} \mid \rho) &\equiv p(w_{j,k} \mid N^i, h\rho(N^i_{j,k}), a(N^i_{j,k})) && (20)\\
&= \sum_{h \in H(N^i_{j,k})} p(w_{j,k}, h(N^i_{j,k}) = h \mid N^i, h\rho(N^i_{j,k}), a(N^i_{j,k})) && (21)\\
&= \sum_{h \in H(N^i_{j,k})} p(h(N^i_{j,k}) = h \mid N^i, h\rho(N^i_{j,k}), a(N^i_{j,k})) \\
&\qquad p(w_{j,k} \mid N^i, h(N^i_{j,k}) = h, h\rho(N^i_{j,k}), a(N^i_{j,k})) && (22)\\
&= \sum_{h \in H(N^i_{j,k})} p(h(N^i_{j,k}) = h \mid N^i, h\rho(N^i_{j,k}), a(N^i_{j,k})) \\
&\qquad p(w_{j,k} \mid N^i, h(N^i_{j,k}) = h) && (23)\\
&= \sum_{h \in H(N^i_{j,k})} p(h(N^i_{j,k}) = h \mid N^i, h\rho(N^i_{j,k}), a(N^i_{j,k})) \\
&\qquad \beta(N^i_{j,k} \mid h) && (24)
\end{aligned}
$$

The only part of this other than substitution obvious equals for equals is Equation 23, where we use the fact that that the probability of a constituent is independent of everything above it, once we know the type of the constituent and its head. (This follows from our probability model as expressed in Equation 12.) Equation 24 then expresses $\beta(N^i_{j,k} \mid \rho)$ in terms of $\beta(N^i_{j,k} \mid h)$ plus probabilities which need to be collected. In our case, as we have already described, these were obtained from parsing 36 million words of the Wall Street Journal.

We now show how to compute $\beta(N^i_{j,k} \mid h)$. As is typical in such cases, we initially assume that our grammar is in Chomsky-normal form. After we have derived the equations for this case, we discuss the generalization to arbitrary grammars.

We first consider the base case. We have some word w_j, and a rule $N^i \rightarrow w_j$. The head of N^i must be w_j, and thus

$$
\begin{aligned}
\beta(N^i_{j,j} \mid h) &\equiv p(w_j \mid N^i, h(N^i) = w_j) && (25)\\
&= 1 && (26)
\end{aligned}
$$

Now we turn to finding $\beta(N^i_{j,k} \mid h)$ where $k > j$. In such cases the immediate expansion of the constituent must involve a rule of the form $N^i \rightarrow N^a\ N^b$. In any such expansion of $N^i_{j,k}$ the head of one of the subconstituents is also the head of $N^i_{j,k}$, and as stated in the body of the paper, we assume that all PCFG rules indicate which of their subconstituents provide the head for the parent constituent. Without loss of generality, let the head providing constituent be

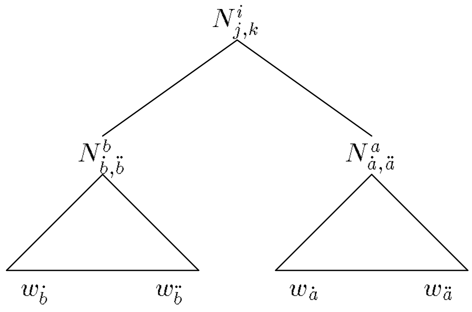

Figure 8: Expansion of $N^i_{j,k}$ into N^a and N^b

N^a (if it is the second constituent we simply exchange the indicators a and b). We denote the start and end locations of a as $\dot{a}$ and $\ddot{a}$, and similarly for b. The situation is illustrated in Figure 8. There we have put the b constituent first to emphasize the fact that a and b denote head relationships, not ordering. Thus the PCFG rule used in the expansion of Figure 8 is $N^i \to N^b\ N^a$. So in what follows a always varies over the possible sub-constituents that are heads, and b over non-heads.

$$\begin{aligned}
\beta(N^i_{j,k} \mid h) &\equiv p(w_{j,k} \mid N^i, h(N^i_{j,k})) && (27)\\
&= \sum_{a,b,\ddot{a},\ddot{b}} p(w_{\dot{b},\ddot{b}}, w_{\dot{a},\ddot{a}}, N^b_{\dot{b},\ddot{b}}, N^a_{\dot{a},\ddot{a}} \mid N^i, h(N^i_{j,k})) && (28)\\
&= \sum_{a,b,\ddot{a},\ddot{b}} p(N^b_{\dot{b},\ddot{b}}, N^a_{\dot{a},\ddot{a}} \mid N^i, h(N^i_{j,k})) \\
&\qquad p(w_{\dot{b},\ddot{b}} \mid N^b_{\dot{b},\ddot{b}}, N^a_{\dot{a},\ddot{a}}, N^i, h(N^i_{j,k})) \\
&\qquad p(w_{\dot{a},\ddot{a}} \mid w_{\dot{b},\ddot{b}}, N^b_{\dot{b},\ddot{b}}, N^a_{\dot{a},\ddot{a}}, N^i, h(N^i_{j,k})) && (29)\\
&= \sum_{a,b,\ddot{a},\ddot{b}} p(N^b_{\dot{b},\ddot{b}}, N^a_{\dot{a},\ddot{a}} \mid N^i, h(N^i_{j,k})) \\
&\qquad p(w_{\dot{b},\ddot{b}} \mid N^b_{\dot{b},\ddot{b}}, h\rho(N^b_{\dot{b},\ddot{b}}) = h(N^i_{j,k}), a(N^b, N^i, N^a)) \\
&\qquad p(w_{\dot{a},\ddot{a}} \mid N^a_{\dot{a},\ddot{a}}, h(N^a_{\dot{a},\ddot{a}}) = h(N^i_{j,k})) && (30)\\
&= \sum_{a,b,\ddot{a},\ddot{b}} p(N^i \to N^b\ N^a \mid h(N^i_{j,k})) \\
&\qquad \beta(N^b_{\dot{b},\ddot{b}} \mid \rho) \cdot \beta(N^a_{\dot{a},\ddot{a}} \mid h) && (31)
\end{aligned}$$

In this derivation Equation 29 breaks up the probability into successive conditional probabilities, Equation 30 uses independence assumptions to simplify the probabilities, and Equation 31 substitutes definitions back into the formula. (In Equation 30 we briefly introduce a function $a(N^b, N^i, N^a)$ to compute if constituent N^b occurs before or after N^a in N^i.)

The generalization of Equation 31 to general PCFGs is quite straight-forward. In such grammars there must still be a constituent which provides the head of the parent constituent (corresponding to constituent a in our derivation), but there may be zero or more constituents which do not (corresponding to b). Thus for such grammars we take the product of $\beta(N^b_{b,\ddot{b}} \mid \rho)$ over all non-head-producing constituents b.

Finally, we comment briefly on how Equations 24 and 31 plays out algorithmically. We determine the set of possible heads of a constituent c, H(c), in a bottom up fashion. The head of a rule which introduces a terminal is that terminal. Then for each higher level constituent we look at all of the rules which could build it, find the lower-level constituent from which that rule finds its head, and take the union of head constituents over all possible building rules.

The actually computation of our probabilities can be done in either a top-down or bottom up fashion. The top-down version is perhaps more intuitive, since $\beta(c \mid \rho)$ requires knowing the head of the parent constituent. However, the bottom-up version is also straight-forward, and perhaps ultimately more useful since if we want to build a "best-first" parser using this model it must build up the constituents, and their probabilities, incrementally.

The critical point for the bottom-up algorithm is that $\beta(c \mid h)$ can be computed using Equation 31 without knowing the parent of the constituent. Thus upon completing c we compute $\beta(c \mid h)$ for all possible heads of c . Then, as we create higher constituents in which c appears we compute the $\beta(c \mid \rho)$ as required by these higher constituents. Note that because this algorithm depends on $\beta(c \mid h)$ not requiring information from above c this algorithm does not generalize well to the full model of Equation 4, where grand-parent constituents are also required. For the full model we have only implemented the top-down algorithm, though a bottom up version is still possible, albeit considerably more complicated.

B Statistical Significance Calculations

We give here the analysis of statistical significance of the data in Figure 5 as summarized in Figure 6.

Since all models have been run on exactly the same data and the models are deterministic, this suggests using a paired sample test for significance. That is, for each sentence we note the values of our random variable (in our case the accuracy for the sentence) for each model. Let $a_i(k)$ be the performance of the i'th model on sentence k. Then let $d_{i,j}(k) = a_i(k)$ - $a_j(k)$, the difference in accuracy between models i and j on sentence k. Let $\overline{d_{i,j}}$ be the mean over all k of $d_{i,j}(k)$. We then compute the Z test on $\overline{d_{i,j}}$ to see if model i is signif-

Superior Model	Mean	No $h\rho^2(c)$	No $p(r(c) \mid h(c))$	Only $p(r(c) \mid h(c))$	PCFG
Complete	93.5	.752	1.66	2.42	5.08
No $h\rho^2(c)$	93.2		.93	2.34	4.67
No $p(r(c) \mid h(c))$	92.5			.94	4.46
Only $p(r(c) \mid h(c))$	91.8				3.36
PCFG	89.1				

Figure 9: Statistical significance of accuracy differences

icantly better than model j. We assume that $d_{i,j}(k) - \overline{d_{i,j}}$ are normally and independently distributed with population mean zero. In Figure 9 we given the Z test results. The left-hand column specifies what we believe to be the superior model. Ignoring the second from left column, the other columns give the Z test values for the improvement over the presumed inferior models as specified across the top of the table. These values show the statistical significance summarized in Figure 6, assuming a one-tailed test at a level of at least 95%.

The problem here is that strictly speaking the Z test results do not apply to the data in Figure 5. Note that in Figure 9 we are considering the average sentence performance. This value is given in column two of the figure, and is defined as:

$$\frac{\sum_{j=1}^{s} a_i(j)}{s} \tag{32}$$

Here s is the number of sentences in the corpus, and $a_i(s)$ is the accuracy for model i on sentence s. Column 2 of the table in Figure 9 gives this performance measure for our models. However, in the body of the paper Figure 5 gives an average accuracy as defined by

$$\frac{\sum_{j=1}^{n_i} ac_i(j)}{n_i} \tag{33}$$

where n_i is the number of attachments made by model i over the entire corpus and $ac_i(j)$ is 1 if for model i attachment j is correct and 0 if it is incorrect. As you can see, the values using $a_i(j)$ from Figure 9 are higher than those using $ac_i(j)$ from Figure 5.. This is because longer sentences tend to have more mistakes per attachment than do shorter ones, and the $a_i(j)$ measure gives equal weight to each sentence, rather than to each attachment. In the body of the paper we report the $ac_i(j)$ number for two reasons: first, it is the figure used in related studies [3,13] and second, it seems more reasonable to us to give equal weight to each attachment decision. However, it is harder to do a statistical analysis on this number, and in particular one cannot do a paired sample test. There are two reasons. First, while it is reasonable to assume that performance on one sentence is independent from the performance on another, it is not reasonable to assume that getting one attachment correct is independent of a second attachment decision in the same sentence. Second, since different models produce different trees, there is no obvious way to pair the decisions on one tree with decision on the second. Thus in this appendix we did the analysis on $a_i(j)$ rather than on $ac_i(j)$. However, there seems no reasons to believe that the statistical significance we find for $a_i(j)$ do not translate to $ac_i(s)$.

References

1. 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.
2. Brent, M. R. *Automatic acquisition of subcategorization frames from untagged text.* In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics.* 1991, 209–214.
3. Brill, E. *Automatic grammar induction and parsing free text: a transformation-based approach.* In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics.* 1993, 259–265.
4. Brown, P. F., Pietra, V. J. D., DeSouza, P. V., Lai, J. C. and Mercer, R. L. Class-based n-gram models of natural language. *Computational Linguistics 18* 4 (1992), 467–479.
5. Carroll, G. and Charniak, E. *Two experiments on learning probabilistic dependency grammars from corpora.* In *Workshop Notes, Statistically-Based NLP Techniques.* AAAI, 1992, 1–13.
6. Charniak, E. and Carroll, G. *Context-Sensitive Statistics for Improved Grammatical Language Models.* In *Proceedings of the Twelfth National Conference on Artificial Intelligence.* AAAI Press/MIT Press, Menlo Park, 1994.
7. Dagan, I., Pereira, F. and Lee, L. *Similarity-based estimation of word cooccurence probabilities.* In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics.* 1994, 272–278.
8. Dunning, T. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics 121* (1993), 61–74.
9. Hindle, D. and Rooth, M. *Structural ambiguity and lexical relations.* In *Proceedings of the Association for Computational Linguistics.* ACL, 1991, 229 – 236.
10. Jones, M. A. and Eisner, J. M. *A probabilistic parser and its applictions.* In *AAAI-92 Worshop on Statistically-Based NLP Techniques.* AAAI, 1992, 20–27.

11. 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.

12. Manning, C. D. *Automatic acquisition of a large subcategorization dictionary from corpora.* In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*. 1993, 235–242.

13. Pereira, F. and Schabes, Y. *Inside-outside reestimation from partially bracketed corpora.* In *27th Annual Meeting of the Association for Computaitonal Linguistics*. ACL, 1992, 128–135.

14. Pereira, F., Tishby, N. and Lee, L. *Distributional clustering of English words.* In *Proceedings of the Association for Computational Linguistics*. ACL, 1993.

15. Schabes, Y., Roth, M. and Osborne, R. *Parsing the Wall Street Journal with the Inside-Outside Algorithm.* In *Proceedings of the Sixth Meeting of the European Chaapter of the Association for Computational Linguistics*. 1993, 341–347.

16. Schabes, Y. and Waters, R. C. *Stochastic Lexicalized Context-Free Grammar.* In *Proceedings of the Third International Workshop on Parsing Technologies*. 1993, 257–266.