Integrating Syntactic Priming into an Incremental Probabilistic Parser, with an Application to Psycholinguistic Modeling

Amit Dubey and **Frank Keller** and **Patrick Sturt**
Human Communication Research Centre, University of Edinburgh
2 Buccleuch Place, Edinburgh EH8 9LW, UK
{amit.dubey,patrick.sturt,frank.keller}@ed.ac.uk

Abstract

The psycholinguistic literature provides evidence for syntactic priming, i.e., the tendency to repeat structures. This paper describes a method for incorporating priming into an incremental probabilistic parser. Three models are compared, which involve priming of rules between sentences, within sentences, and within coordinate structures. These models simulate the reading time advantage for parallel structures found in human data, and also yield a small increase in overall parsing accuracy.

1 Introduction

Over the last two decades, the psycholinguistic literature has provided a wealth of experimental evidence for *syntactic priming*, i.e., the tendency to repeat syntactic structures (e.g., Bock, 1986). Most work on syntactic priming has been concerned with sentence production; however, recent studies also demonstrate a preference for structural repetition in human parsing. This includes the so-called *parallelism effect* demonstrated by Frazier et al. (2000): speakers processes coordinated structures more quickly when the second conjunct repeats the syntactic structure of the first conjunct.

Two alternative accounts of the parallelism effect have been proposed. Dubey et al. (2005) argue that the effect is simply an instance of a pervasive syntactic priming mechanism in human parsing. They provide evidence from a series of corpus studies which show that parallelism is not limited to co-ordination, but occurs in a wide range of syntactic structures, both within and between sentences, as predicted if a general priming mechanism is assumed. (They also show this effect is stronger in coordinate structures, which could explain Frazier et al.'s (2000) results.)

Frazier and Clifton (2001) propose an alternative account of the parallelism effect in terms of a *copying mechanism*. Unlike priming, this mechanism is highly specialized and only applies to coordinate structures: if the second conjunct is encountered, then instead of building new structure, the language processor simply copies the structure of the first conjunct; this explains why a speed-up is observed if the two conjuncts are parallel. If the copying account is correct, then we would expect parallelism effects to be restricted to coordinate structures and not to apply in other contexts.

This paper presents a parsing model which implements both the priming mechanism and the copying mechanism, making it possible to compare their predictions on human reading time data. Our model also simulates other important aspects of human parsing: (i) it is broad-coverage, i.e., it yields accurate parses for unrestricted input, and (ii) it processes sentences incrementally, i.e., on a word-by-word basis. This general modeling framework builds on probabilistic accounts of human parsing as proposed by Jurafsky (1996) and Crocker and Brants (2000).

A priming-based parser is also interesting from an engineering point of view. To avoid sparse data problems, probabilistic parsing models make strong independence assumptions; in particular, they generally assume that sentences are independent of each other, in spite of corpus evidence for structural repetition between sentences. We therefore expect a parsing model that includes structural repetition to provide a better fit with real corpus data, resulting in better parsing performance. A simple and principled approach to handling structure re-use would be to use adaptation probabilities for probabilistic grammar rules (Church, 2000), analogous to cache probabilities used in caching language models (Kuhn and de Mori, 1990). This is the approach we will pursue in this paper.

Dubey et al. (2005) present a corpus study that demonstrates the existence of parallelism in corpus data. This is an important precondition for understanding the parallelism effect; however, they

Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, 417–424. Sydney, 2006.

do not develop a parsing model that accounts for the effect, which means they are unable to evaluate their claims against experimental data. The present paper overcomes this limitation. In Section 2, we present a formalization of the priming and copying models of parallelism and integrate them into an incremental probabilistic parser. In Section 3, we evaluate this parser against reading time data taken from Frazier et al.'s (2000) parallelism experiments. In Section 4, we test the engineering aspects of our model by demonstrating that a small increase in parsing accuracy can be obtained with a parallelism-based model. Section 5 provides an analysis of the performance of our model, focusing on the role of the distance between prime and target.

2 Priming Models

We propose three models designed to capture the different theories of structural repetition discussed above. To keep our model as simple as possible, each formulation is based on an unlexicalized probabilistic context free grammar (PCFG). In this section, we introduce the models and discuss the novel techniques used to model structural similarity. We also discuss the design of the probabilistic parser used to evaluate the models.

2.1 Baseline Model

The unmodified PCFG model serves as the Baseline. A PCFG assigns trees probabilities by treating each rule expansion as conditionally independent given the parent node. The probability of a rule $LHS \rightarrow RHS$ is estimated as:

$$P(RHS|LHS) = \frac{c(LHS \rightarrow RHS)}{c(LHS)}$$

2.2 Copy Model

The first model we introduce is a probabilistic variant of Frazier and Clifton's (2001) copying mechanism: it models parallelism in coordination and nothing else. This is achieved by assuming that the default operation upon observing a coordinator (assumed to be anything with a *CC* tag, e.g., 'and') is to copy the full subtree of the preceding coordinate sister. Copying impacts on how the parser works (see Section 2.5), and in a probabilistic setting, it also changes the probability of trees with parallel coordinated structures. If coordination is present, the structure of the second item is either identical to the first, or it is not.[1] Let us call the probability of having a copied tree as p_{ident}. This value may be estimated directly from a corpus using the formula

$$\hat{p}_{ident} = \frac{c_{ident}}{c_{total}}$$

Here, c_{ident} is the number of coordinate structures in which the two conjuncts have the same internal structure and c_{total} is the total number of coordinate structures. Note we assume there is only one parameter p_{ident} applicable everywhere (i.e., it has the same value for all rules).

How is this used in a PCFG parser? Let t_1 and t_2 represent, respectively, the first and second coordinate sisters and let $P_{PCFG}(t)$ be the PCFG probability of an arbitrary subtree t.

Because of the independence assumptions of the PCFG, we know that $p_{ident} \gg P_{PCFG}(t)$. One way to proceed would be to assign a probability of p_{ident} when structures match, and $(1 - p_{ident}) \cdot P_{PCFG}(t_2)$ when structures do not match. However, some probability mass is lost this way: there is a nonzero PCFG probability (namely, $P_{PCFG}(t_1)$) that the structures match.

In other words, we may have identical subtrees in two different ways: either due to a copy operation, or due to a PCFG derivation. If p_{copy} is the probability of a copy operation, we can write this fact more formally as: $p_{ident} = P_{PCFG}(t_1) + p_{copy}$.

Thus, if the structures do match, we assign the second sister a probability of:

$$p_{copy} + P_{PCFG}(t_1)$$

If they do not match, we assign the second conjunct the following probability:

$$\frac{1 - P_{PCFG}(t_1) - p_{copy}}{1 - P_{PCFG}(t_1)} \cdot P_{PCFG}(t_2)$$

This accounts for both a copy mismatch and a PCFG derivation mismatch, and assures the probabilities still sum to one. These probabilities for parallel and non-parallel coordinate sisters, therefore, gives us the basis of the Copy model.

This leaves us with the problem of finding an estimate for p_{copy}. This value is approximated as:

$$\hat{p}_{copy} = \hat{p}_{ident} - \frac{1}{|T_2|} \sum_{t \in T_2} P_{PCFG}(t)$$

In this equation, T_2 is the set of all second conjuncts.

2.3 Between Model

While the Copy model limits itself to parallelism in coordination, the next two models simulate structural priming in general. Both are similar in design, and are based on a simple insight: we may

[1] The model only considers two-item coordination or the last two sisters of multiple-item coordination.

condition a PCFG rule expansion on whether the rule occurred in some previous context. If *Prime* is a binary-valued random variable denoting if a rule occurred in the context, then we define:

$$P(RHS|LHS, Prime) = \frac{c(LHS \rightarrow RHS, Prime)}{c(LHS, Prime)}$$

This is essentially an instantiation of Church's (2000) adaptation probability, albeit with PCFG rules instead of words. For our first model, this context is the previous sentence. Thus, the model can be said to capture the degree to which rule use is primed between sentences. We henceforth refer to this as the Between model. Following the convention in the psycholinguistic literature, we refer to a rule use in the previous sentence as a 'prime', and a rule use in the current sentence as the 'target'. Each rule acts once as a target (i.e., the event of interest) and once as a prime. We may classify such adapted probabilities into 'positive adaptation', i.e., the probability of a rule given the rule occurred in the preceding sentence, and 'negative adaptation', i.e., the probability of a rule given that the rule did not occur in the preceding sentence.

2.4 Within Model

Just as the Between model conditions on rules from the previous sentence, the Within sentence model conditions on rules from earlier in the current sentence. Each rule acts once as a target, and possibly several times as a prime (for each subsequent rule in the sentence). A rule is considered 'used' once the parser passes the word on the leftmost corner of the rule. Because the Within model is finer grained than the Between model, it can be used to capture the parallelism effect in coordination. In other words, this model could explain parallelism in coordination as an instance of a more general priming effect.

2.5 Parser

As our main purpose is to build a psycholinguistic model of structure repetition, the most important feature of the parsing model is to build structures incrementally.[2]

Reading time experiments, including the parallelism studies of Frazier et al. (2000), make word-by-word measurements of the time taken to read sentences. Slower reading times are known to be correlated with processing difficulty, and faster reading times (as is the case with parallel structures) are correlated with processing ease. A probabilistic parser may be considered to be a sentence processing model via a 'linking hypothesis', which links the parser's word-by-word behavior to human reading behavior. We discuss this topic in more detail in Section 3. At this point, it suffices to say that we require a parser which has the prefix property, i.e., which parses incrementally, from left to right.

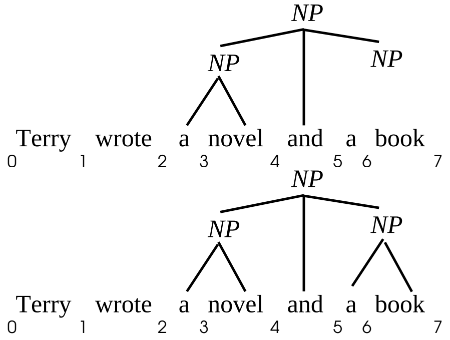
NP
NP
NP
Terry wrote a novel and a book
0 1 2 3 4 5 6 7
NP
NP
NP
Terry wrote a novel and a book
0 1 2 3 4 5 6 7

Figure 1: Upon encountering a coordinator, the copy model copies the most likely first conjunct.

Therefore, we use an Earley-style probabilistic parser, which outputs Viterbi parses (Stolcke, 1995). We have two versions of the parser: one which parses exhaustively, and a second which uses a variable width beam, pruning any edges whose merit is $\frac{1}{2000}$ of the best edge. The merit of an edge is its inside probability times a prior $P(LHS)$ times a lookahead probability (Roark and Johnson, 1999). To speed up parsing time, we right binarize the grammar,[3] remove empty nodes, coindexation and grammatical functions. As our goal is to create the simplest possible model which can nonetheless model experimental data, we do not make any tree modification designed to improve accuracy (as, e.g., Klein and Manning 2003).

The approach used to implement the Copy model is to have the parser copy the subtree of the first conjunct whenever it comes across a *CC* tag. Before copying, though, the parser looks ahead to check if the part-of-speech tags after the *CC* are equivalent to those inside the first conjunct. The copying model is visualized in Figure 1: the top panel depicts a partially completed edge upon seeing a *CC* tag, and the second panel shows the completed copying operation. It should be clear that

[2] In addition to incremental parsing, a characteristic some of psycholinguistic models of sentence comprehension is to parse deterministically. While we can compute the best incremental analysis at any point, ours models do not parse deterministically. However, following the principles of rational analysis (Anderson, 1991), our goal is not to mimic the human parsing *mechanism*, but rather to create a model of human parsing *behavior*.

[3] We found that using an unbinarized grammar did not alter the results, at least in the exhaustive parsing case.

the copy operation gives the most probable subtree in a given span. To illustrate this, consider Figure 1. If the most likely *NP* between spans 2 and 7 does not involve copying (i.e. only standard PCFG rule derivations), the parser will find it using normal rule derivations. If it does involve copying, for this particular rule, it must involve the most likely *NP* subtree from spans 2 to 3. As we parse incrementally, we are guaranteed to have found this edge, and can use it to construct the copied conjunct over spans 5 to 7 and therefore the whole co-ordinated *NP* from spans 2 to 7.

To simplify the implementation of the copying operation, we turn off right binarization so that the constituent before and after a coordinator are part of the same rule, and therefore accessible from the same edge. This makes it simple to calculate the new probability: construct the copied subtree, and decide where to place the resulting edge on the chart.

The Between and Within models require a cache of recently used rules. This raises two dilemmas. First, in the Within model, keeping track of full contextual history is incompatible with chart parsing. Second, whenever a parsing error occurs, the accuracy of the contextual history is compromised. As we are using a simple unlexicalized parser, such parsing errors are probably quite frequent.

We handle the first problem by using one single parse as an approximation of the history. The more realistic choice for this single parse is the best parse so far according to the parser. Indeed, this is the approach we use for our main results in Section 3. However, because of the second problem noted above, in Section 4, we simulated the context by filling the cache with rules from the correct tree. In the Between model, these are the rules of the correct parse of the previous tree; in the Within model, these are the rules used in the correct parse at points up to (but not including) the current word.

3 Human Reading Time Experiment

In this section, we test our models by applying them to experimental reading time data. Frazier et al. (2000) reported a series of experiments that examined the parallelism preference in reading. In one of their experiments, they monitored subjects' eye-movements while they read sentences like (1):

(1) a. Hilda noticed a strange man and a tall woman when she entered the house.
b. Hilda noticed a man and a tall woman when she entered the house.

They found that total reading times were faster on the phrase *tall woman* in (1a), where the coordinated noun phrases are parallel in structure, compared with in (1b), where they are not.

There are various approaches to modeling processing difficulty using a probabilistic approach. One possibility is to use an incremental parser with a beam search or an *n*-best approach. Processing difficulty is predicted at points in the input string where the current best parse is replaced by an alternative derivation (Jurafsky, 1996; Crocker and Brants, 2000). An alternative is to keep track of all derivations, and predict difficulty at points where there is a large change in the shape of the probability distribution across adjacent parsing states (Hale, 2001). A third approach is to calculate the forward probability (Stolcke, 1995) of the sentence using a PCFG. Low probabilities are then predicted to correspond to high processing difficulty. A variant of this third approach is to assume that processing difficulty is correlated with the (log) probability of the best parse (Keller, 2003). This final formulation is the one used for the experiments presented in this paper.

3.1 Method

The item set was adapted from that of Frazier et al. (2000). The original two relevant conditions of their experiment (1a,b) differ in terms of length. This results in a confound in the PCFG framework, because longer sentences tend to result in lower probabilities (as the parses tend to involve more rules). To control for such length differences, we adapted the materials by adding two extra conditions in which the relation between syntactic parallelism and length was reversed. This resulted in the following four conditions:

(2) a. DT JJ NN and DT JJ NN (parallel)
Hilda noticed a tall man and a strange woman when she entered the house.
b. DT NN and DT JJ NN (non-parallel)
Hilda noticed a man and a strange woman when she entered the house.
c. DT JJ NN and DT NN (non-parallel)
Hilda noticed a tall man and a woman when she entered the house.
d. DT NN and DT NN (parallel)
Hilda noticed a man and a woman when she entered the house.

In order to account for Frazier et al.'s parallelism effect a probabilistic model should predict a greater difference in probability between (2a) and (2b) than between (2c) and (2d) (i.e., (2a)−(2b) > (2c)−(2d)). This effect will not be confounded with length, because the relation between length and parallelism is reversed between (2a,b) and (2c,d). We added 8 items to the original Frazier et al. materials, resulting in a new set of 24 items similar to (2).

We tested three of our PCFG-based models on all 24 sets of 4 conditions. The models were the Baseline, the Within and the Copy models, trained exactly as described above. The Between model was not tested as the experimental stimuli were presented without context. Each experimental sentence was input as a sequence of correct POS tags, and the log probability estimate of the best parse was recorded.

3.2 Results and Discussion

Table 1 shows the mean log probabilities estimated by the models for the four conditions, along with the relevant differences between parallel and non-parallel conditions.

Both the Within and the Copy models show a parallelism advantage, with this effect being much more pronounced for the Copy model than the Within model. To evaluate statistical significance, the two differences for each item were compared using a Wilcoxon signed ranks test. Significant results were obtained both for the Within model ($N = 24$, $Z = 1.67$, $p < .05$, one-tailed) and for the Copy model ($N = 24$, $Z = 4.27$, $p < .001$, one-tailed). However, the effect was much larger for the Copy model, a conclusion which is confirmed by comparing the differences of differences between the two models ($N = 24$, $Z = 4.27$, $p < .001$, one-tailed). The Baseline model was not evaluated statistically, because by definition it predicts a constant value for (2a)−(2b) and (2c)−(2d) across all items. This is simply a consequence of the PCFG independence assumption, coupled with the fact that the four conditions of each experimental item differ only in the occurrences of two NP rules.

The results show that the approach taken here can be successfully applied to the modeling of experimental data. In particular, both the Within and the Copy models show statistically reliable parallelism effects. It is not surprising that the copy model shows a large parallelism effect for the Frazier et al. (2000) items, as it was explicitly designed to prefer structurally parallel conjuncts. The more interesting result is the parallelism effect found for the Within model, which shows that such an effect can arise from a more general probabilistic priming mechanism.

4 Parsing Experiment

In the previous section, we were able to show that the Copy and Within models are able to account for human reading-time performance for parallel coordinate structures. While this result alone is sufficient to claim success as a psycholinguistic model, it has been argued that more realistic psycholinguistic models ought to also exhibit high accuracy and broad-coverage, both crucial properties of the human parsing mechanism (e.g., Crocker and Brants, 2000).

This should not be difficult: our starting point was a PCFG, which already has broad coverage behavior (albeit with only moderate accuracy). However, in this section we explore what effects our modifications have to overall coverage, and, perhaps more interestingly, to parsing accuracy.

4.1 Method

The models used here were the ones introduced in Section 2 (which also contains a detailed description of the parser that we used to apply the models). The corpus used for both training and evaluation is the Wall Street Journal part of the Penn Treebank. We use sections 1–22 for training, section 0 for development and section 23 for testing. Because the Copy model posits coordinated structures whenever POS tags match, parsing efficiency decreases if POS tags are not predetermined. Therefore, we assume POS tags as input, using the gold-standard tags from the treebank (following, e.g., Roark and Johnson 1999).

4.2 Results and Discussion

Table 2 lists the results in terms of F-score on the test set.[4] Using exhaustive search, the baseline model achieves an F-score of 73.3, which is comparable to results reported for unlexicalized incremental parsers in the literature (e.g. the RB1 model of Roark and Johnson, 1999). All models exhibit a small decline in performance when beam search is used. For the Within model we observe a slight improvement in performance over the baseline, both for the exhaustive search and the beam

[4] Based on a χ^2 test on precision and recall, all results are statistically different from each other. The Copy model actually performs slightly better than the Baseline in the exhaustive case.

Model	para: (2a)	non-para: (2b)	non-para: (2c)	para: (2d)	(2a)−(2b)	(2c)−(2d)
Baseline	−33.47	−32.37	−32.37	−31.27	−1.10	−1.10
Within	−33.28	−31.67	−31.70	−29.92	−1.61	−1.78
Copy	−16.18	−27.22	−26.91	−15.87	11.04	−11.04

Table 1: Mean log probability estimates for Frazier et al (2000) items

	Exhaustive Search		Beam Search		Beam + Coord		Fixed Coverage	
Model	F-score	Coverage	F-score	Coverage	F-score	Coverage	F-score	Coverage
Baseline	73.3	100	73.0	98.0	73.1	98.1	73.0	97.5
Within	73.6	100	73.4	98.4	73.0	98.5	73.4	97.5
Between	71.6	100	71.7	98.7	71.5	99.0	71.8	97.5
Copy	73.3	100	–	–	73.0	98.1	73.1	97.5

Table 2: Parsing results for the Within, Between, and Copy model compared to a PCFG baseline.

search conditions. The Between model, however, resulted in a decrease in performance.

We also find that the Copy model performs at the baseline level. Recall that in order to simplify the implementation of the copying, we had to disable binarization for coordinate constituents. This means that quaternary rules were used for coordination ($X \rightarrow X_1\ CC\ X_2\ X'$), while normal binary rules ($X \rightarrow Y\ X'$) were used everywhere else. It is conceivable that this difference in binarization explains the difference in performance between the Between and Within models and the Copy model when beam search was used. We therefore also state the performance for Between and Within models with binarization limited to non-coordinate structures in the column labeled 'Beam + Coord' in Table 2. The pattern of results, however, remains the same.

The fact that coverage differs between models poses a problem in that it makes it difficult to compare the F-scores directly. We therefore compute separate F-scores for just those sentences that were covered by all four models. The results are reported in the 'Fixed Coverage' column of Table 2. Again, we observe that the copy model performs at baseline level, while the Within model slightly outperforms the baseline, and the Between model performs worse than the baseline. In Section 5 below we will present an error analysis that tries to investigate why the adaptation models do not perform as well as expected.

Overall, we find that the modifications we introduced to model the parallelism effect in humans have a positive, but small, effect on parsing accuracy. Nonetheless, the results also indicate the success of both the Copy and Within approaches to parallelism as psycholinguistic models: a modification primarily useful for modeling human behavior has no negative effects on computational measures of coverage or accuracy.

5 Distance Between Rule Uses

Although both the Within and Copy models succeed at the main task of modeling the parallelism effect, the parsing experiments in Section 4 showed mixed results with respect to F-scores: a slight increase in F-score was observed for the Within model, but the Between model performed below the baseline. We therefore turn to an error analysis, focusing on these two models.

Recall that the Within and Between models estimate two probabilities for a rule, which we have been calling the positive adaptation (the probability of a rule when the rule is also in the history), and the negative adaptation (the probability of a rule when the rule is *not* in the history). While the effect is not always strong, we expect positive adaptation to be higher than negative adaptation (Dubey et al., 2005). However, this is not always the case.

In the Within model, for example, the rule $NP \rightarrow DT\ JJ\ NN$ has a higher negative than positive adaptation (we will refer to such rules as 'negatively adapted'). The more common rule $NP \rightarrow DT\ NN$ has a higher positive adaptation ('positively adapted'). Since the latter is three times more common, this raises a concern: what if adaptation is an artifact of frequency? This 'frequency' hypothesis posits that a rule recurring in a sentence is simply an artifact of the its higher frequency. The frequency hypothesis could explain an interesting fact: while the majority of rules tokens have positive adaptation, the majority of rule types have negative adaptation. An important corollary of the frequency hypothesis is that we would not expect to find a bias towards local rule re-uses.

```
Iterate through the treebank
  Remember how many words each constituent spans
Iterate through the treebank
  Iterate through each tree
    Upon finding a constituent spanning 1-4 words
      Swap it with a randomly chosen constituent
      of 1-4 words
      Update the remembered size of the swapped
      constituents and their subtrees
Iterate through the treebank 4 more times
  Swap constituents of size 5-9, 10-19, 20-35
  and 35+ words, respectively
```

Figure 2: The treebank randomization algorithm

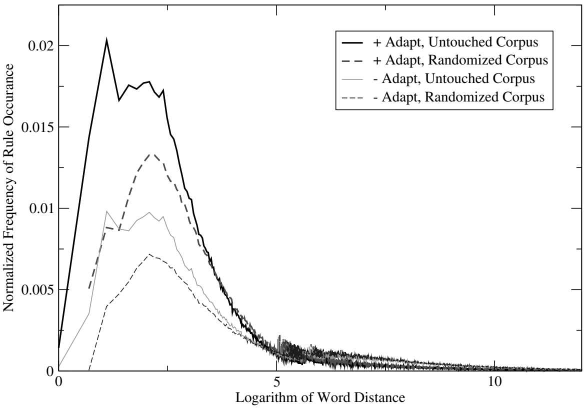

Figure 3: Log of number of words between rule invocations

Nevertheless, the $NP \rightarrow DT\ \ JJ\ \ NN$ rule is an exception: most negatively adapted rules have very low frequencies. This raises the possibility that sparse data is the cause of the negatively adapted rules. This makes intuitive sense: we need many rule occurrences to accurately estimate positive or negative adaptation.

We measure the distribution of rule use to explore if negatively adapted rules owe more to frequency effects or to sparse data. This distributional analysis also serves to measure 'decay' effects in structural repetition. The decay effect in priming has been observed elsewhere (Szmrecsanyi, 2005), and suggests that positive adaptation is higher the closer together two rules are.

5.1 Method

We investigate the dispersion of rules by plotting histograms of the distance between subsequent rule uses. The basic premise is to look for evidence of an early peak or skew, which suggests rule re-use. To ensure that the histogram itself is not sensitive to sparse data problems, we group all rules into two categories: those which are positively adapted, and those which are negatively adapted.

If adaptation is not due to frequency alone, we would expect the histograms for both positively and negatively adapted rules to be skewed towards local rule repetition. Detecting a skew requires a baseline without repetition. We propose the concept of 'randomizing' the treebank to create such a baseline. The randomization algorithm is described in Figure 2. The algorithm entails swapping subtrees, taking care that small subtrees are swapped first (otherwise large chunks would be swapped at once, preserving a great deal of context). This removes local effects, giving a distribution due frequency alone.

After applying the randomization algorithm to the treebank, we may construct the distance histogram for both the non-randomized and randomized treebanks. The distance between two occurrences of a rule is calculated as the number of words between the first word on the left corner of each rule. A special case occurs if a rule expansion invokes another use of the same rule. When this happens, we do not count the distance between the first and second expansion. However, the second expansion is still remembered as the most recent.

We group rules into those that have a higher positive adaptation and those that have a higher negative adaptation. We then plot a histogram of rule re-occurrence distance for both groups, in both the non-randomized and randomized corpora.

5.2 Results and Discussion

The resulting plot for the Within model is shown in Figure 3. For both the positive and negatively adapted rules, we find that randomization results in a lower, less skewed peak, and a longer tail. We conclude that rules tend to be repeated close to one another more than we expect by chance, even for negatively adapted rules. This is evidence against the frequency hypothesis, and in favor of the sparse data hypothesis. This means that the small size of the increase in F-score we found in Section 4 is not due to the fact that the adaption is just an artifact of rule frequency. Rather, it can probably be attributed to data sparseness.

Note also that the shape of the histogram provides a decay curve. Speculatively, we suggest that this shape could be used to parameterize the decay effect and therefore provide an estimate for adaptation which is more robust to sparse data. However, we leave the development of such a smoothing function to future research.

6 Conclusions and Future Work

The main contribution of this paper has been to show that an incremental parser can simulate syntactic priming effects in human parsing by incorporating probability models that take account of previous rule use. Frazier et al. (2000) argued that the best account of their observed parallelism advantage was a model in which structure is copied from one coordinate sister to another. Here, we explored a probabilistic variant of the copy mechanism, along with two more general models based on within- and between-sentence priming. Although the copy mechanism provided the strongest parallelism effect in simulating the human reading time data, the effect was also successfully simulated by a general within-sentence priming model. On the basis of simplicity, we therefore argue that it is preferable to assume a simpler and more general mechanism, and that the copy mechanism is not needed. This conclusion is strengthened when we turn to consider the performance of the parser on the standard Penn Treebank test set: the Within model showed a small increase in F-score over the PCFG baseline, while the copy model showed no such advantage.[5]

All the models we proposed offer a broad-coverage account of human parsing, not just a limited model on a hand-selected set of examples, such as the models proposed by Jurafsky (1996) and Hale (2001) (but see Crocker and Brants 2000).

A further contribution of the present paper has been to develop a methodology for analyzing the (re-)use of syntactic rules over time in a corpus. In particular, we have defined an algorithm for randomizing the constituents of a treebank, yielding a baseline estimate of chance repetition.

In the research reported in this paper, we have adopted a very simple model based on an unlexicalized PCFG. In the future, we intend to explore the consequences of introducing lexicalization into the parser. This is particularly interesting from the point of view of psycholinguistic modeling, because there are well known interactions between lexical repetition and syntactic priming, which require lexicalization for a proper treatment. Future work will also involve the use of smoothing to increase the benefit of priming for parsing accuracy. The investigations reported in Section 5 provide a basis for estimating the smoothing parameters.

[5] The broad-coverage parsing experiment speaks against a 'facilitation' hypothesis, i.e., that the copying and priming mechanisms work together. However, a full test of this (e.g., by combining the two models) is left to future research.

References

Anderson, John. 1991. Cognitive architectures in a rational analysis. In K. VanLehn, editor, *Architectures for Intelligence*, Lawrence Erlbaum Associates, Hillsdale, N.J., pages 1–24.

Bock, J. Kathryn. 1986. Syntactic persistence in language production. *Cognitive Psychology* 18:355–387.

Church, Kenneth W. 2000. Empirical estimates of adaptation: the chance of two Noriegas is closer to $p/2$ than p^2. In *Proceedings of the 17th Conference on Computational Linguistics*. Saarbrücken, Germany, pages 180–186.

Crocker, Matthew W. and Thorsten Brants. 2000. Wide-coverage probabilistic sentence processing. *Journal of Psycholinguistic Research* 29(6):647–669.

Dubey, Amit, Patrick Sturt, and Frank Keller. 2005. Parallelism in coordination as an instance of syntactic priming: Evidence from corpus-based modeling. In *Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing*. Vancouver, pages 827–834.

Frazier, Lyn, Alan Munn, and Chuck Clifton. 2000. Processing coordinate structures. *Journal of Psycholinguistic Research* 29(4):343–370.

Frazier, Lynn and Charles Clifton. 2001. Parsing coordinates and ellipsis: Copy α. *Syntax* 4(1):1–22.

Hale, John. 2001. A probabilistic Earley parser as a psycholinguistic model. In *Proceedings of the 2nd Conference of the North American Chapter of the Association for Computational Linguistics*. Pittsburgh, PA.

Jurafsky, Daniel. 1996. A probabilistic model of lexical and syntactic access and disambiguation. *Cognitive Science* 20(2):137–194.

Keller, Frank. 2003. A probabilistic parser as a model of global processing difficulty. In R. Alterman and D. Kirsh, editors, *Proceedings of the 25th Annual Conference of the Cognitive Science Society*. Boston, pages 646–651.

Klein, Dan and Christopher D. Manning. 2003. Accurate Unlexicalized Parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*. Sapporo, Japan, pages 423–430.

Kuhn, Roland and Renate de Mori. 1990. A cache-based natural language model for speech recognition. *IEEE Transanctions on Pattern Analysis and Machine Intelligence* 12(6):570–583.

Roark, Brian and Mark Johnson. 1999. Efficient probabilistic top-down and left-corner parsing. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*. pages 421–428.

Stolcke, Andreas. 1995. An efficient probabilistic context-free parsing algorithm that computes prefix probabilities. *Computational Linguistics* 21(2):165–201.

Szmrecsanyi, Benedikt. 2005. Creatures of habit: A corpus-linguistic analysis of persistence in spoken English. *Corpus Linguistics and Linguistic Theory* 1(1):113–149.