Probabilistic Parsing and Psychological Plausibility

Paper #739

Abstract

Given the recent evidence for probabilistic mechanisms in models of human ambiguity resolution, this paper investigates the plausibility of exploiting current wide-coverage, probabilistic parsing techniques to model human linguistic performance. In particular, we investigate the performance of standard stochastic parsers when they are revised to operate incrementally, and with reduced memory resources. We present several techniques for ranking and filtering analyses, together with experimental results, varying the type of filtered edges (inactive, active, both) and the beam type (variable, fixed). Our results confirm that stochastic parsers which adhere to these psychologically motivated constraints achieve good performance. Memory can be reduced down to 1% (compared to exhausitve search) without reducing recall and precision. Additionally, these models exhibit substantially faster performance. Finally, we argue that this general result is likely to hold for more sophisticated probabilistic parsing models.

1 Introduction

Language engineering and computational psycholinguistics are often viewed as distinct research programmes: engineering solutions aim at practical methods which can achieve good performance, typically paying little attention to linguistic or cognitive modelling. Computational psycholinguistics, on the other hand, is often focussed on detailed modelling of human behaviour for a relatively small number of well-studied constructions. In this paper, we suggest that, broadly, the human sentence processing mechanism (HSPM) and current statistical parsing technology can be viewed as having similar objectives: to optimally (i.e. rapidly and accurately) understand the text and utterances they encounter.

Our aim is to show that large scale probabilistic parsers, when subjected to basic cognitive constraints, can still achieve high levels of parsing accuracy. If successful, this will provide a plausible model of the fact that people, in general, are also extremely accurate and robust. Such a result would also strengthen existing results showing that related probabilistic mechanisms can explain specific psycholinguistic phenomena.

To investigate this issue, we construct a standard 'baseline' stochastic parser, which mirrors the performance of a similar systems (e.g. (Johnson, 1998)). We then consider an incremental versions of the parser, and evaluate the effects of several probabilistic filtering strategies which are used to prune the parser's search space, and thereby reduce memory load.

We present the results of several parsing performance experiments, showing the accuracy of these systems with respect to both a parsed corpus and the baseline parser. Our experiments suggest that a strictly incremental model, in which memory resources are substantially reduced through filtering, can achieve precision and recall which equals that of 'unrestricted' systems. Furthermore, implementation of these restrictions leads to substantially faster performance. In conclusion, we argue that such broad-coverage probabilistic parsing models provide a valuable framework for explaining the human capacity to rapidly, accurately, and robustly understand "garden variety" language. This lends further support to psycholinguistic accounts which posit probabilistic ambiguity resolution mechanisms to explain "garden path" phenomena.

2 Psycholinguistic Motivation

Theories of human sentence processing have largely been shaped by the study of pathologies in human language processing behaviour. Most psycholinguistic models seek to explain the *difficulty* people have in comprehending structures that are ambiguous or memory-intensive (see (Crocker, 1999) for a recent overview). While often insightful, this approach diverts attention from the fact that people are in fact extremely accurate and effective in understanding the vast majority of their "linguistic experience". This observation, combined with the mounting psycholinguistic evidence for statistically-based mechanisms, leads us to investigate the merit of exploiting robust, broad coverage, probabilistic parsing systems as models of human linguistic performance.

The view that human language processing can be viewed as an optimally adapted system, within a probabilistic framework, is advanced by (Chater et al., 1998), while (Jurafsky, 1996) has proposed a specific probabilistic parsing model of human sentence processing. In work on human lexical category disambiguation, (Crocker and Corley, to appear), have demonstrated that a standard (incremental) HMM-based part-of-speech tagger models the finding from a range of psycholinguistic experiments. In related research, (Crocker and Brants, 1999) present evidence that an incremental stochastic parser based on Cascaded Markov Models (Brants, 1999) can account for a range of experimentally observed local ambiguity preferences. These include NP/S complement ambiguities, reduced relative clauses, noun-verb category ambiguities, and 'that'-ambiguities (where 'that' can be either a complementizer or a determiner).

Crucially, however, there are differences between the classes of mechanisms which are psychologically plausible, and those which prevail in current language technology. We suggest that two of the most important differgences concern *incrementality*, and *memory resources.* There is overwhelming experimental evidence that people constructing connected (i.e. semantically interpretable) analyses for each initial substring of an utterance, as it is encountered. That is, processing takes place incrementally, from left to right, on a word by word basis. In a recent study, (Sturt and Lombardo, 1999) show that processing a corpus incrementally indeed needs a very low number of predictions. They investigate headless projections required during parsing and find that their number is 0 for more than 80% of the constituents.

Secondly, it is universally accecpted that people can at most consider a relatively small number of competing analyses (indeed, some would argue that number is one, i.e. processing is strictly serial). In contrast, many existing stochastic parsers are "unrestricted", in that they are optimised for accuracy, and ignore such psychologically motivated constraints. Thus the appropriateness of using broad-coverage probabilistic parsers to model the high level of human performance is contingent upon being able to maintain these levels of accuracy when the constraints of incrementality and resource limitations are imposed.

3 Incremental Stochastic Context-Free Parsing

The following assumes that the reader is familiar with stochastic context-free grammars (SCFG) and stochastic chart-parsing techniques. A good introduction can be found, e.g., in (Manning and Schütze, 1999). We use standard abbreviations for terminial nodes, nonterminal nodes, rules and probabilities.

This paper investigates stochastic context-free parsing based on a grammar that is derived from a treebank, starting with part-of-speech tags as terminals. The grammar is derived by collecting all rules $X \to \alpha$ that occur in the treebank and their frequencies f. The probability of a rule is set to

$$P(X \to \alpha) = \frac{f(X \to \alpha)}{\sum\limits_{\beta} f(X \to \alpha)} \qquad (1)$$

For a description of treebank grammars see (Charniak, 1996). The grammar does not contain ϵ-rules, otherwise there is no restriction on the rules. In particular, we do not require Chomsky-Normal-Form.

In addition to the rules that correspond to structures in the corpus, we add a new start symbol ROOT to the grammar and rules $\mathsf{ROOT} \to X$ for all non-terminals X togehter with probabilities derived from the root nodes in the corpus[1].

For parsing these grammars, we rely upon a standard bottom-up chart-parsing technique with a modification for incremental parsing, i.e., for each words, all edges are processed and possibly pruned before proceeding to the next word. The outline of the algorithm is as follows.

A chart entry E consists of a start and endposition i and j, a dotted rule $X \rightarrow \alpha.\gamma$, the inside probability $\beta(X_{i,j})$ that X generates the terminal string from position i to j, and information about the most-probable inside structure. If the dot of the dotted rule is at the rightmost position, the corresponding edge is an *inactive* edge. If the dot is at any other position, it is an *active* edge. Inactive edges represent recognized hypothetical constituents, while active edges represent prefixes of hypothetical constituents.

The ith terminal node t_i that enters the chart generates an inactive edge for the span $(i-1, i)$. Based on this, new active and inactive edges are generated according to the standard algorithm. Since we are interested in the most-probable parse, the chart can be minimized in the following way while still performing an exhaustive search. If there is more than one edge that covers a span (i, j) having the same non-terminal symbol on the left-hand side of the dotted rule, only the one with the highest inside probability is kept in the chart. The others cannot contribute to the most-probable parse.

For an inactive edge spanning i to j and representing the rule $X \rightarrow Y^1 \ldots Y^k$, the inside probability β_I is set to

$$\beta_I(X_{i,j}) = P(X \rightarrow Y_1 \ldots Y_k) \prod_{l=1}^{k} \beta_I(Y^l_{i_l,j_l}) \quad (2)$$

where i_l and j_l mark the start and end postition of Y^l, having $i = i_1$ and $j = j_l$. The inside probability for an active edge β_A with the dot after the kth symbol of the right-hand side is set to

$$\beta_A(X_{i,j}) = \prod_{l=1}^{k} \beta_I(Y^k_{i_l,j_l}) \quad (3)$$

We do not use the probability of the rule at this point. This allows us to combine all edges with the same span and the dot at the same position but with different symbols on the left-hand side. Introducing a distinguished left-hand side only for inactive edges significantly reduces the number of active edges in the chart. This goes one step further than implicitly right-binarizing the grammar; not only suffixes of right-hand sides are joined, but also the corresponding left-hand sides.

[1]The ROOT node is used internally for parsing; it is neither emitted nor counted for recall and precision.

4 Memory Restrictions

We investigate the elimination (pruning) of edges from the chart in our incremental parsing scheme. After processing a word and before proceeding to the next word during incremental parsing, low ranked edges are removed. This is equivalent to imposing memory restrictions on the processing system.

The original algorithm keeps one edge in the chart for each combination of span (start and end position) and non-terminal symbol (for inactive edges) or right-hand side prefixes of dotted rules (for active edges). With pruning, we restrict the number of edges allowed per span. The limitation can be expressed in two ways:

1. *Variable beam.* Select a threshold $\theta \geq 1$. Edge e is removed, if its probability is p_e, the best probability for the span is p_1, and
$$p_e < \frac{p_1}{\theta}. \quad (4)$$
2. *Fixed beam.* Select a maximum number of edges per span m. An edge e is removed, if its probability is not in the first m highest probabilities for edges with the same span.

We compare and rank edges covering the same span only, and we rank active and inactive edges separately. This is in contrast to (Charniak et al., 1998) who rank all edges. They use normalization in order to account for different spans since in general, edges for longer spans involve more multiplications of probabilities, yielding lower probabilities. Charniak *et al.*'s normalization value is calculated by a different probability model than the inside probabilities of the edges. So, in addition to the normalization for different span lengths, they need a normalization constant that accounts for the different probability models.

This investigation is based on a much simpler ranking formula. We use what informally can be described as the unigram probability of a non-terminal node, i.e., the *a priori* probability of the corresponding non-terminal symbol(s) times the inside probability.

For an inactive edge $\langle i, j, X \rightarrow \alpha, \beta_I(X_{i,j})\rangle$, we use the probability

$$\begin{aligned} P_{RI}(X_{i,j}) &= P(X) \cdot P(t_i \ldots t_{j-1}|X) & (5) \\ &= P(X) \cdot \beta_I(X_{i,j}) & (6) \end{aligned}$$

for ranking. β_I is the inside probability for inactive edges as given in equation 2, $P(X)$ is the *a priori* probability for non-terminal X, (as estimated from the frequency in the training corpus) and P_{RI} is the probability of the edge for the non-terminal X spanning positions i to j that is used for ranking.

For an active edge $\langle i, j, X \rightarrow Y^1 \cdots Y^k . Y^{k+1} \cdots Y^m, \beta_A(Y^1 \cdots Y^k)\rangle$ (the dot is after the kth symbol of the right-hand side) we use:

$$\begin{aligned} & P_{RA}(Y^1_{i_1,j_1} \ldots Y^k_{i_k,j_k}) & (7) \\ &= P(Y^1 \cdots Y^k) \cdot P(t_i \ldots t_{j-1}|Y^1 \cdots Y^k) & (8) \\ &= P(Y^1 \cdots Y^k) \cdot \beta_A(Y^1_{i_1,j_1} \ldots Y^k_{i_k,j_k}) & (9) \end{aligned}$$

$P(Y^1 \cdots Y^k)$ can be read off the corpus. It is the a-priori probability that the right-hand side of a production has the prefix $Y^1_{i_1,j_1} \ldots Y^k_{i_k,j_k}$, which is estimated by

$$\frac{f(Y^1 \cdots Y^k \text{ is prefix})}{N} \qquad (10)$$

where N is the total number of productions in the corpus[2] and β_A is the inside probability.

5 Experiments

We use sections 2 – 21 of the Wall Street Journal part of the Penn Treebank (Marcus et al., 1993) to generate a treebank grammar. Traces, functional tags and other tag extensions that do not mark syntactic category are removed before training[3]. No other modifications are made. For testing, we use the 1578 sentences of length 40 or less of section 22. The input to the parser is the sequence of part-of-speech tags.

For accuracy evaluation, we report labeled recall (LR), labeled precision (LP), and labeled F-Score (LF; the harmonic mean of LR and LP). As a measure of the amount of work done by the parser, we report the size of the chart (SIZE), the average number of parallel hypotheses (PAR), and parsing speed in tokens per second (T/S). SIZE is the number of active and inactive edges that enter the chart[4], not counting those hypothetical edges that are replaced or rejected because there is an alternative edge with higher probability, and not counting those that are immediately pruned before generating any other edge. PAR gives the average number of parallel hypotheses per span for all those spans that have at least one analysis. T/S is given as a relative measure. No serious efforts of optimization have gone into the parser. Speed is measured on a Pentium II 500 running Linux.

5.1 Variable Beam

For the first set of experiments, we define the beam by a threshold θ on the quotient of the highest probability and the probability in question (cf. equation 4). First, beams for active and inactive edges are investigated separately. Then, both types of edges are pruned. The beams runs on a logarithmic scale from 2 to 2000. For pruning both types of edges, we choose the sub-range around the first value, that achieves at least an F-score equivalent to exhaustive parsing. No attempt to find the global optimum is made.

5.2 Fixed Number of Analyses

For the second set of experiments, we define the beam by a maximum number of edges per span. Again, beams for active and inactive edges are first investigated separately, then in combination. The beams run from 2 to 10. For pruning both types of edges, we choose the sub-range around the first value, that achieves at least an F-score equivalent to that of exhaustive parsing.

5.3 Experimental Results

The results of our experiments are given in figure 1. The main finding is that for all types of pruning (inactive/active/both, variable/fixed beam) accuracy results equivalent to exhaustive search can be achieved. Only very small beams

[2] Here, we use proper prefixes, i.e., all prefixes not including the last element.

[3] As an example, PP-TMP=3 is replaced by PP.

[4] SIZE is comparable to the "number of edges popped" as given in (Charniak et al., 1998)

VARIABLE BEAM

PRUNING INACTIVE EDGES

θ_i	SIZE	PAR	T/S	LR	LP	LF
2	3.86%	1.2	282	67.00	72.39	69.59
5	4.95%	1.4	226	67.07	72.51	69.68
10	8.25%	1.9	133	68.84	73.83	71.25
20	10.89%	2.4	95	68.76	73.69	71.14
50	13.88%	2.7	72	68.82	73.80	71.22
100	16.96%	2.9	55	68.81	73.78	71.21
200	20.57%	3.3	41	68.82	73.79	71.22
500	28.29%	4.0	26	68.86	73.84	71.26
1000	36.59%	4.8	18	68.85	73.82	71.25
2000	42.09%	5.6	13	68.95	73.76	71.28
exh.	141650	18.7	1.8	68.82	73.77	71.21

PRUNING ACTIVE EDGES

θ_a	SIZE	PAR	T/S	LR	LP	LF
2	3.61%	16.7	184	64.99	69.22	67.04
5	3.77%	16.8	146	66.46	71.19	68.74
10	3.94%	17.0	122	67.62	72.45	69.95
20	4.22%	17.1	92	68.32	73.26	70.71
50	4.52%	17.2	78	68.48	73.45	70.88
100	4.82%	17.3	67	68.63	73.62	71.04
200	5.22%	17.4	56	68.79	73.78	71.20
500	5.88%	17.6	46	68.79	73.79	71.20
1000	6.52%	17.7	39	68.80	73.80	71.21
2000	7.31%	17.9	33	68.83	73.81	71.24
exh.	141650	18.7	1.8	68.82	73.77	71.21

PRUNING BOTH TYPES

θ_i	θ_a	SIZE	PAR	T/S	LR	LP	LF
5	500	1.44%	1.4	352	67.02	72.44	69.63
5	1000	1.61%	1.4	338	67.03	72.46	69.64
5	2000	1.83%	1.4	320	67.07	72.50	69.68
10	500	1.95%	1.8	258	68.81	73.80	71.22
10	1000	2.21%	1.8	245	68.82	73.82	71.23
10	2000	2.54%	1.9	230	68.84	73.83	71.25
20	500	2.29%	2.3	213	68.74	73.67	71.12
20	1000	2.62%	2.3	199	68.75	73.69	71.13
20	2000	3.00%	2.3	185	68.76	73.69	71.14
	exh.	141650	18.7	1.8	68.82	73.77	71.21

FIXED BEAM

PRUNING INACTIVE EDGES

m_i	SIZE	PAR	T/S	LR	LP	LF
1	1.49%	1.0	439	29.90	56.44	39.09
2	9.39%	2.0	112	68.78	73.67	71.14
3	17.34%	3.0	51	68.83	73.79	71.22
4	27.68%	3.9	26	68.84	73.81	71.24
5	36.45%	4.9	16	68.81	73.77	71.21
6	45.16%	5.9	11	68.82	73.79	71.22
7	52.84%	6.9	8	68.82	73.79	71.22
8	59.32%	7.9	6	68.82	73.78	71.21
9	64.83%	8.9	5	68.82	73.78	71.21
10	71.03%	9.8	4	68.82	73.78	71.21
exh.	141650	18.7	1.8	68.82	73.77	71.21

PRUNING ACTIVE EDGES

m_a	SIZE	PAR	T/S	LR	LP	LF
1	3.49%	16.5	217	60.60	64.44	62.46
2	3.85%	17.1	135	68.02	72.68	70.27
3	4.16%	17.6	102	68.78	73.55	71.08
4	4.39%	17.8	83	68.89	73.75	71.24
5	4.64%	18.0	70	68.81	73.75	71.19
6	4.85%	18.1	61	68.84	73.82	71.24
7	5.09%	18.3	54	68.91	73.90	71.32
8	5.30%	18.3	49	68.88	73.87	71.29
9	5.51%	18.4	45	68.87	73.83	71.26
10	5.71%	18.4	41	68.86	73.83	71.26
exh.	141650	18.7	1.8	68.82	73.77	71.21

PRUNING BOTH TYPES

m_i	m_a	SIZE	PAR	T/S	LR	LP	LF
2	3	0.97%	2.0	331	68.62	73.42	70.94
2	4	1.15%	2.0	301	68.82	73.66	71.16
2	5	1.32%	2.0	278	68.81	73.72	71.18
3	3	1.19%	3.0	263	68.82	73.60	71.13
3	4	1.37%	3.0	231	68.87	73.73	71.22
3	5	1.56%	3.0	208	68.81	73.73	71.19
4	3	1.39%	3.9	225	68.78	73.57	71.10
4	4	1.59%	3.9	191	68.87	73.75	71.23
4	5	1.77%	3.9	170	68.80	73.75	71.19
	exh.	141650	18.7	1.8	68.82	73.77	71.21

Figure 1: Experimental results using a variable beam (left column) and a fixed beam (right column), for pruning inactive and active separately and pruning both. *exh.* gives the results for exhaustive search. θ_i and θ_a are the threshold on the quotient for the beam for inactive and active edges, m_i and m_a are the max. number of parallel analyses for the fixed beam. SIZE is the size of the chart in percent of edges used for exhaustive search (for which the averaged absolute chart size per sentence is given), PAR is the average number of parallel analyses per span, T/S is the parsing speed in tokens per second, LR, LP, and LF are labeled recall, precision, and F-Score.

degrade performance[5]. The effect is very robust despite the simple ranking formula. This significantly reduces memory requirements (given as size of the chart) and increases parsing speed.

When comparing SIZE vs. F-Score, pruning active edges yields a more efficient parser (active: 4.82% chart size for 71.04% F-Score; inactive: 8.25% chart size for 71.25% F-Score; similar results for a fixed beam), although somewhat slower since more edges are tested (and immediately pruned). Best efficiency is achieved by pruning both types of edges. Using a variable beam, 1.95% of the chart size are used for an F-Score of 71.22%. The result is even better for a fixed beam: 1.15% chart size for 71.16% F-Score[6]. When comparing VARIABLE vs. FIXED, the fixed beams yield equivalent accuracies but a much more efficient parser when pruning both types of edges (1.15% vs. 1.95% chart size; 301 vs. 258 tokens/second). This is a surprising result and we do not have an explanation for it.

6 Related Work

Probably mostly related to the work reported here are (Charniak et al., 1998) and (Roark and Johnson, 1999). Both report on significantly improved parsing efficiency by selecting only a subset of edges for processing. There are three main differences to our approach. One is that they use a ranking for best-first search while we immediately prune hypotheses. They need to store a large number edges because it is not known in advance how many of the edges will be used until a parse is found. The second difference is that we proceed strictly incrementally without look-ahead. (Charniak et al., 1998) use a non-incremental procedure, (Roark and Johnson, 1999) use a look-ahead of one word. Thirdly, we use a much simpler ranking formula.

Additionally, (Charniak et al., 1998) and (Roark and Johnson, 1999) do not use the original Penntree encoding for the context-free structures. Before training and parsing, they change/remove some of the productions and introduce new part-of-speech tags for auxiliaries. The exact effect of these modifications is unknown, and it is unclear if these affect comparability to our results.

The heavy restrictions in our method (immediate pruning, no look-ahead, very simple ranking formula) have consequences on the accuracy. Using right context and sorting instead of pruning yields roughly 2% higher results[7]. But our work shows that even with these massive restrictions, the chart size can be reduced to only slightly over 1% without a decrease in accuracy when compared to exhaustive search.

We use a pure stochastic context-free grammar as the base model. This can be improved upon. As (Charniak, 1997) shows, adding word statistics significantly improves accuracy. Furthermore, (Johnson, 1998) shows that changes in the structural encoding also significantly improves results. In order to test our hypothesis that the effects of incremental processing and massive pruning found here are also valid for improved models, we performed the series of experiments as reported in section 5 with the "parent" encoding proposed by Johnson. The relative results were equivalent, but recall and precision were shifted up by around 8%, which supports our hypothesis.

7 Conclusions

A central challenge in computational psycholinguistics is to explain how it is that people are so accurate and robust in processing language. Given the substantial psycholinguistic evidence for statistical cognitive mechanisms, our objective in this paper was to assess the plausibility of using wide-coverage probabilistic parsers to model human linguistic performance. In particular, we set out to investigate the effects of imposing incremental processing and significant memory limitations on such parsers.

The central finding of our experiments is that incremental parsing with massive (almost 99%) pruning of the search space does not impair the accuracy of stochastic context-free parsers. This basic finding was robust across both the pruning strategy (active edges, inac-

[5] Given the amount of test data (26,322 non-terminal nodes), results within a range of around 1% are equivalent with a confidence degree of $\alpha = 99\%$.

[6] Or even 0.97% chart size for 70.94% F-Score.

[7] Comparison of results is not straight-forward since (Roark and Johnson, 1999) report accuracies only for those sensentences for which a parse tree was generated (between 93 and 98% of the sentences), while our parser (except for very small beams) generates parses for virtually all sentences, hence we report accuracies for all sentences.

tive, or both) and a range of relative and absolute beam widths. We did however, observe significantly reduced memory and time complexity when using combined active/inactive edge filtering, and a fixed beam size. To our knowledge, this is the first investigation on tree-bank grammars that systematically varies the edge type and the beam type for pruning. Especially the better performance of the fixed beam compared to the variable beam is surprising.

Our aim in this paper is not to challenge state-of-the-art parsing accuracy results. For our experiments we used a purely context-free stochastic parser combined with a very simple pruning scheme based on simple "unigram" probabilities, and no use of right context. We do, however suggest that our result should apply to richer, more sophistacted probabilistic models. As mentioned in section 6, for example, using Johnson's technique of encoding the parent node to introduce additional context information, resulted in the same performance gains in our incremental, resource limited, parser.

We therefore conclude that wide-coverage, probabilistic parsers do not suffer impaired accuracy when subject to strict cognitive memory limitations and incremental processing. Furthermore, parse times are substantially reduced. This suggests that it may be fruitful to pursue the use of these models within computational psycholinguistics, where it is necessary to explain not only the relatively rare 'pathologies' of the human parser, but also its more frequently observed accuracy and robustness.

References

Thorsten Brants. 1999. Cascaded markov models. In *Proceedings of 9th Conference of the European Chapter of the Association for Computational Linguistics EACL-99*, Bergen, Norway.

Eugene Charniak, Sharon Goldwater, and Mark Johnson. 1998. Edge-based best-first chart parsing. In *Proceedings of the Sixth Workshop on Very Large Corpora (WVLC-98)*, Montreal, Kanada.

Eugene Charniak. 1996. Tree-bank grammars. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 1031–1036, Menlo Park: AAAI Press/MIT Press.

Eugene Charniak. 1997. Statistical parsing with a context-free grammar and word statistics. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, pages 1031–1036, Menlo Park: AAAI Press/MIT Press.

Nicholas Chater, Matthew Crocker, and Martin Pickering. 1998. The rational analysis of inquiry: The case for parsign. In Chater and Oaksford, editors, *Rational Models of Cognition*. Oxford University Press.

Matthew Crocker and Thorsten Brants. 1999. Incremental probabilistic models of human linguistic performance. In *The 5th Conference on Architectures and Mechanisms for Language Processing*, Edinburgh, U.K.

Matthew Crocker and Steffan Corley. to appear. Modular architectures and statistical mechanisms: The case from lexical category disambiguation. In Merlo and Stevenson, editors, *The Lexical Basis of Sentence Processing*. John Benjamins.

Matthew Crocker. 1999. Mechanisms for sentence processing. In Garrod and Pickering, editors, *Language Processing*. Psychology Press, London, UK.

Mark Johnson. 1998. PCFG models of linguistic tree representations. *Computational Linguistics*, 24(4):613–632.

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

Christopher Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, Massachusetts.

Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Brian Roark and Mark Johnson. 1999. Efficient probabilistic top-down and left-corner parsing. In *Proceedings of the 37th Annual Meeting of the Association for Computation Linguistics ACL-99*, ...

Patrick Sturt and Vincenzo Lombardo. 1999. Incrementality and lexicalism: a treebank study. In *Proceedings of Architectures and Mechanisms in Language Processing AMLaP-99*, Edinburgh.