

# Part-of-speech tagging models for parsing

Rebecca Watson

Computer Laboratory, University of Cambridge, Cambridge, CB3 0FD, UK

firstname.lastname@cl.cam.ac.uk

## Abstract

We investigate the accuracy of alternative part-of-speech tag models and their impact on parser performance. In addition to considering single-tag and multiple-tag per word input, tag selection models which draw on information available from the parser are applied. Results indicate that given a ‘good’ PoS tagger, parser-based tag selection models are unable to improve on the low tag error rates of the tagger. Multiple-tag per word input can improve on parser accuracy but at a cost to efficiency. A dynamic tag selection model is also applied, which only increases the number of tags considered for sentences if a full parse could not be found. This achieves the best accuracy and provides a means to overcome the trade-off between tag error rates and increased parse ambiguity introduced by multiple-tag per word input.

## 1 Introduction

Research investigating the use of part-of-speech (PoS) taggers as front ends to parsers has, to date, concentrated on whether or not such a pre-processing stage improves parse accuracy and/or efficiency. We do not investigate whether a PoS tagger should be used (readers are referred to Dalrymple (2004) for a recent survey and discussion of this) but instead focus on the choice of tag model.

Charniak *et al.* (1996) investigates the optimal choice of tag model for a probabilistic context-free grammar (PCFG) parser. By measuring the parser’s tagging accuracy (that is, the tag sequence in the top ranked parse), Charniak concludes that the parser, given multiple-tag per word (tpw) input, can not significantly improve on the tag accuracy of a (single-tpw) PoS tagger. Further, the parser is less efficient given multiple-tpw input.

Similarly, we investigate the optimal tag model with respect to an unlexicalised statistical LR parser. Gold standard tag and dependency relation sets enable comparison of tagging models in terms of both tagging and parser performance. As far as the authors are aware, we are the first to perform such a broad comparison.

Dalrymple (2004) discusses how tagging may help to reduce ambiguity in parser output. She suggests that the tag sequence corresponding to the largest number of parses will be the correct tag sequence and should be selected, to reduce the parse ambiguity by around 50%. While she was unable to test this hypothesis without a gold standard tag set, we found that Dalrymple’s tag selection model performed the worst of all considered. More sophisticated tag disambiguation schemes are also applied. Though these models failed to improve on the accuracy of the tagger.

Clark and Curran (2004) apply a tag selection strategy that uses the grammar to decide whether or not supertags provided by the supertagger are acceptable. That is, whether or not the grammar is able to find a full analysis. They assign a small number of supertags per word initially (1.4 tpw) and continue

to increase the number of supertags until either the parser finds an analysis or the maximal tag set has been parsed. This is shown to improve the efficiency, coverage and accuracy of the parser.

We apply a similar tag selection model as we observe that single-tpw input achieves high accuracy when considering full parses only, while using multiple-tpw input increases the parser's coverage. This model out-performed all others considered herein with respect to parser accuracy.

We find that none of the static tagging models are able to improve on the (tagging) accuracy of the tagger. Given that the (single-tpw) tagger achieves 97.32% accuracy, there is little room for improvement in the tagger itself.

## 2 The RASP System

RASP is a pipelined modular system where text is pre-processed before being passed to the statistical parser using a series of components including: tokenization, tagging and morphological analysis.<sup>1</sup> All experiments use the RASP system, modified in various ways. An overview of this system will be given herein and more detailed explanations will only be provided of specific components required to follow the experiments outlined here. For full details of system components see (Carroll, 1993; Briscoe and Carroll, 1995; Briscoe and Carroll, 2002).

### 2.1 PoS Tagging

RASP incorporates a first order HMM PoS tagger (tagger, henceforth) originally implemented by Elworthy (1994). The tagger can be run in 'single-tag' or 'multiple-tag' modes where either the most probable or the set of all possible tags are retained, respectively. The tagger is trained on 3M words of text from the Susanne, LOB and BNC corpora and applies a subset of the CLAWS tagset (149 PoS tags and 13 punctuation tags) using a lexicon of just over 50K words. Minor modifications were made to the tagger's lexicon recently, based on observed parse failures over Sections from the WSJ (not including Section 23).

When run in multiple-tag mode, the tagger will output the posterior tag probabilities of each tag

---

<sup>1</sup>Processing overheads quoted throughout do not include these pre-processing stages.

which occurs as a possible tag for the word in the lexicon. RASP incorporates these probabilities directly as the probability of the word (with given tag) during parsing.

Prior to parsing however two thresholds are applied.<sup>2</sup> All but the most probable tag are removed from consideration if this tag has a posterior probability higher than 0.90 and all tags which are less than one in fifty times as probable as the top tag are removed.

### 2.2 Parser

Carroll (1993) gives full details of the probabilistic LALR(1) parser which is trained on the Susanne treebank. The LALR(1) table defines shift and reduce actions (and associated probabilities) that drives the parser based on a 'context free backbone' of the underlying unification grammar. Unifications are then performed on each reduce action and sub-analyses related via subsumption are packed resulting in the 'parse forest' representation.<sup>3</sup>

### 2.3 Parser Output

Briscoe and Carroll (2002) outline the different output formats available, including syntactic tree, grammatical relations (GRs) and robust minimal recursion semantics (RMRS). As well as the top or n-best GR outputs, another output format available is weighted GRs (Carroll and Briscoe, 2002): the unique set of GRs output from all parses weighted according to the frequency and probabilities of the parses in which the particular GR occurs.

If the parser is unable to find a full analysis (that is, one rooted in the start category) then the system outputs a 'fragmentary' (Frag) analysis that is a connected sequence of partial analysis spanning the input by applying a modified shortest paths algorithm (Briscoe and Carroll, 2005). Therefore, given sufficient memory and time, the system is able to produce an analysis for most sentences which lie outside the grammar.

However, time and memory limitations self-imposed on the system will sometimes result in parse time outs, we therefore remove these limitations. When non-determinism in the input and

---

<sup>2</sup>These thresholds can be specified by the user.

<sup>3</sup>We modified the packing algorithm, details to appear in (Watson et al., 2005).

output are increased, in some cases overall parser throughput decreases significantly.

### 3 Data

King *et al.* (2003) outline the development of the PARC 700 Dependency Bank (henceforth DepBank), a gold-standard set of relational dependencies for 700 sentences from Section 23 of the WSJ.

Briscoe and Carroll (2005) extended DepBank with a set of gold-standard GRs and (manually corrected) PoS tags. We used the gold-standard GRs to measure parser performance, and used the tag set as a gold standard (after we applied further manual corrections). We use the same 560 sentence subset from the DepBank utilised in Kaplan *et al.* (2004).

A gold standard named-entity (NE) markup for the test data was provided by Stephan Riezler, co-author of Kaplan *et al.* (2004). The text version of the 560 sentences was created by removing all NE tags from this file.

### 4 Tagging Experimentation

Charniak *et al.* (1996) concludes that single-tpw taggers should be used (instead of multiple-tpw) as front ends to parsers. These findings are based on the minimal increase in tagging accuracy achieved by the parser when using multiple-tpw input (which requires increased parse time). Results illustrate that the coverage of the parser is only marginally affected by incorrect tags, where the single-tpw input results in 99.2% of sentences parsed compared to 100% given multiple-tpw input (all tags in the lexicon). A coarse tag set consisting of only 19 PoS tags is applied during experimentation. Therefore these results may not translate for more fine grained tag sets.

Dalrymple (2004) recently investigated the impact of PoS tags on parse ambiguity; represented by the number of parses licensed by the grammar. By grouping parses via their tag sequences, Dalrymple found that the majority of parses could in fact be differentiated in terms of their tag sequence as only 30% of sentences had parses which all contained the same tag sequence. If the correct tag sequence could be determined in these cases, parse ambiguity is shown to decrease by around 46%. Dalrymple hypothesised that the correct tag sequence could be

selected by choosing the tag sequence corresponding to the largest number of parses.

This section considers the tagging performance of various tagging models including those applied by Charniak *et al.* (1996) and suggested by Dalrymple (2004). Consideration of the parser as a tagging model assumes that a feed-back loop would enable the parser to first select the PoS tag sequence and then select a parse from the group of parses which contain that tag sequence.

#### 4.1 Tag Data

Table 1 outlines the different tag models considered and the corresponding tag setup name for each. The tag setup name will be referred to henceforth for brevity.

For this experiment, we had the parser compute all parses licensed by the grammar,<sup>4</sup> enabling calculation of the weighted GR output, the number of parses containing each tag, and the weighted sum of parse probabilities containing each tag.

#### 4.2 Evaluation Measures

The measures described here are designed to represent the tagging error rates only and not the impact of alternative tagging models on parsing performance which will be discussed in Section 5.

Standard precision, recall and  $F_1$  measures are determined, along with the mean reciprocal rank (MRR) score of tags. Calculation of the MRR is performed using the equation:

$$MRR = \frac{1}{\#tags} \sum_{i=1}^{\#tags} \frac{1}{correct - tag - rank_i}$$

The MRR score is considered here as the MSD-WEIGHT-ALL and MSD-NUM-ALL re-rank tags from the MTAG-SYS-DEF setup and therefore tag rankings can be compared. Also considered is the proportion of sentences affected by tagging errors, calculated as the percentage of tagged sentences containing at least one tagging error (Sent).

Lastly, the average tag cost (ATC) is designed to illustrate the average distance between a tag and the gold-standard tag, thereby representing the predicted impact on parsing accuracy. In the CLAWS

<sup>4</sup>Due to parse time required to unpack all parses from the parse forest, a dynamic programming approach is applied over the parse forest ((Watson *et al.*, 2005) forthcoming).

Tag Setup	Description
STAG	Tagger in single-tag mode.
MTAG	Tagger in multiple-tag mode.
MTAG-SYS-DEF	Default thresholds (0.90, 50) applied to MTAG.
MTAG-SYS	Thresholds (0.99, 200) applied to MTAG.
MSD-TOP-PARSE	Tag sequence in top parse when parsing MTAG-SYS-DEF.
MS-TOP-PARSE	Tag sequence in top parse when parsing MTAG-SYS.
M-TOP-PARSE	Tag sequence in top parse when parsing MTAG.
MSD-NUM-TOP	Most frequently used tag by all parses when parsing MTAG-SYS-DEF.
MSD-NUM-ALL	Normalised counts of tags used by all parses when parsing MTAG-SYS-DEF.
MSD-WEIGHT-TOP	Highest scoring tag based upon the (normalised) sum of probabilities of parses in which tags occur when parsing MTAG-SYS-DEF.
MSD-WEIGHT-ALL	Tag score determined as for MSD-WEIGHT-TOP but all tags' scores recorded.
GOLD	The gold standard tag set.

Table 1: Tag Setup Descriptions. MSD-NUM and MSD-WEIGHT tag setups are normalised based upon the number of parses and sum of all parse probabilities respectively.

tag set generally the first letter codes major PoS category and subsequent letters/numbers more minor differences. Thus, ATC is determined using the average position in which the tag names disagree. If the first letters disagree then this is assumed to be more detrimental than if the last letters or numbers disagree. Therefore, NN is closer to NP1 than to VBZ.

The distance between two tags is calculated as the reciprocal of the position in which tag letters disagree. In the previous example, the distance would therefore be 0.5 and 1, respectively.<sup>5</sup>

### 4.3 Results

Table 2 illustrates the tagging performance of all eleven tag setups measured against the GOLD tag setup.

The first four rows of this table illustrate the tagging performance of the system's tagger. The following three rows illustrate the performance of the parser's top parse tag selection for the three alternative multiple-tag setups and the remaining four rows

<sup>5</sup>A few exceptions apply to the distance measure: tags identified as equivalent in the grammar have a distance of 0 (for example, &FO is treated by the grammar as a name so is equivalent to NP1 tags); other confusions considered less detrimental (for example, confusion of nouns and adjectives) have a distance equal to the reciprocal of the position plus one or two. Note that tags identified as equivalent in the grammar are not considered tag errors when calculating any of the tag evaluation measures.

illustrate the top tag and tag ranking based on the sum of parses and weighted sum of parses respectively.

Upper bounds on tagging performance are illustrated by the MTAG results, where the only tagging errors are made by the unknown word handling module.

As shown in Table 2, none of the alternative parser-based tagging models are able to improve on the accuracy of the single-tpw output of the tagger (or the ranking of multiple-tpw as shown by the MRR score). As the average number of tpw increases the performance of the parsers' top parse declines (reflected in all evaluation measures).

Further, the tagging model MSD-NUM-TOP, as suggested by Dalrymple (2004) is the poorest performing tagging model. The more sophisticated weighted model (MSD-WEIGHT-TOP) also performed poorly and further, failed to improve on the ranking of tags (MSD-WEIGHT-ALL).

If we consider MSD-TOP-PARSE the gold standard, and measure the precision of tags in MSD-NUM-TOP and MSD-WEIGHT-TOP we find that 99.72% and 94.86% of tags agree respectively. This suggests that the top ranked parse tags (MSD-TOP-PARSE) tend to occur frequently in the higher ranked parses but less frequently across all parses. These findings suggest that it is the statistical model more than the grammar causing incorrect tag-

Tag Setup	Avg tpw <sup>†</sup>	Precision	Recall	MRR	ATC	Sent
STAG	1	97.23	97.23	97.18	0.5757	40.71
MTAG-SYS-DEF	1.12	88.50	98.79	97.94	-	21.79
MTAG-SYS	1.23	80.86	99.42	98.26	-	11.25
MTAG	1.51	65.89	99.78	98.42	-	4.64
MSD-TOP-PARSE	1	95.38	95.38	95.38	0.6086	59.11
MS-TOP-PARSE	1	94.47	94.47	94.41	0.6286	64.46
M-TOP-PARSE	1	93.77	93.77	93.71	0.6496	69.29
MSD-NUM-TOP	1	92.72	93.86	93.68	0.6325	65.71
MSD-NUM-ALL	1.12	89.23	98.65	95.99	-	24.11
MSD-WEIGHT-TOP	1	94.67	95.84	95.66	0.6127	54.82
MSD-WEIGHT-ALL	1.12	89.23	98.65	97.05	-	24.11

Table 2: Tagging Performance.<sup>†</sup>The average tag per word.

selection.

Accuracy of the tagger in single-tpw mode (STAG) on DepBank is good, achieving precision of 97.23%. This precision is higher than might be expected on arbitrary text, as the tagger lexicon has been adapted to the WSJ corpus. Therefore, there is limited room for improvement by the alternative tagging models.

In order to emulate performance of the tagging models over data with higher levels of unseen words, we determined tagging performance of each model using an initial PoS tagger with artificially reduced performance over the 560 test sentences (reduced accuracy by around 2%). Similar results were observed for the tagging models and none of the alternative parser-based tagging models were able to improve on the accuracy of the PoS tagger. Further, we tested the performance of the tagging models over the *grammar development corpus* (GDC), a list of 1825 simple sentences that RASP’s grammar is designed to cover. Whilst RASP is not trained on the GDC, this data represents sentences for which the grammar will have a correct parse. Therefore, this data provides an approximate upper bound on how well the parser could correct over a PoS tagger. However, similar results were also observed over this data set and the initial PoS tagger out-performed all other tagging models.

## 5 Parser Experimentation

While the alternative tagging models outlined in Section 4 are unable to improve on the accuracy

of the tagger, this will not necessarily translate to equally detrimental parsing performance. That is, the tagging evaluation measures may not accurately reflect the impact of these models on the parser’s performance given that the parser can recover from certain tag confusions and not others. Therefore, this section discusses the optimal tagging model in terms of parser evaluation measures.

### 5.1 Evaluation Measures

The parser’s output is evaluated using a relational dependency evaluation scheme (Carroll et al., 1998; Lin, 1998). The standard evaluation measures of precision (prec), recall (rec) and  $F_1$  measure are considered for both the ‘macroaverage’ and ‘microaverage’ over the parser’s GRs. The microaverage is determined over the counts for all relations while the macroaverage is the average over each individual relation’s performance. Unless otherwise stated,  $F_1$  will refer to microaverage  $F_1$ .

The proportion of sentences which result in a Frag analysis is also reported, along with the time taken to parse the sentences using an Intel Pentium 4 3.2GHz CPU with 1GB of Ram on a 32 bit version of Linux.

### 5.2 Results

Table 3 illustrates the performance of the parser using alternative tag models as front-ends. The ten tag selection models outlined in Section 4.1 are shown in the first 20 rows of the table, where each model has two corresponding lines of results. The first line of each pair illustrates the parser’s performance over

all 560 test sentences. The second line will be discussed below.

If we first compare the performance of the parser for the alternative tag models over all 560 test sentences, it is clear that the most efficient tagging model is the tagger in single-tpw mode (STAG). However, the coverage of the parser is increased (the percentage of Frag parses is halved) by using MTAG-SYS-DEF tag setup and an increase of 0.66%  $F_1$  measure results (a relative error reduction of 2.27%). Though there is also a similar decrease in macroaveraged  $F_1$ . The increase in parse time required to process MTAG-SYS-DEF illustrates that there is, as is often the case, a trade-off between accuracy and efficiency.

Parsing over the correctly tagged test suite (GOLD) illustrates that a 2.02% increase in  $F_1$  (6.97% relative reduction in error) results from removing the 2.77% of tag errors. Comparing this to the 0.53% increase in  $F_1$  (1.83% relative reduction in error) which results from using gold standard NE mark-up (STAG-NE), suggests there is more to be gained by concentrating on tag selection than NE recognition. However, this may not be the case over data sets for which a high number of unknown words occur that could be marked as NE, for example, in biological texts.

In order to compare the impact of the alternative tagging models on parser accuracy, it is also necessary to consider the accuracy for those sentences for which Frag parses did not occur. To compare all tagging models across a consistent set of sentences, the sentences for which Frag parses occurred in the STAG set (21.25%) were effectively removed.<sup>6</sup> The accuracy over this set is illustrated in the second row for each tag setup in Table 3.

The 3.29% increase in  $F_1$  resulting for the STAG tag setup illustrates that a large proportion of the errors are introduced by the Frag parse output. Further, the margin between the STAG and GOLD tag setups has narrowed to only 0.84%  $F_1$ , illustrating that tag errors in STAG account for a large proportion of the 21.25% resulting Frag parses. This raises an interesting question: can we rely on the grammar

---

<sup>6</sup>The proportion of Frag parses is around 5% worse than reported by Briscoe and Carroll (2005) as STAG does not include NE mark-up and contains different tokenisation (for example, quotation marks varied significantly).

to find parses if and only if the correct tag sequence is input?

Clark and Curran (2004) apply a tag selection strategy whereby they assign a small number of supertags per word initially (1.4 tpw) and increase the number of supertags if the parser fails to find an analysis. This is shown to improve the efficiency, coverage and accuracy of the parser. We implement a similar tag selection strategy here. However, their approach utilises supertags rather than PoS tags and the supertagger performs with lower single tpw accuracy (around 91% compared to 97.23%) and lower multiple tpw accuracy (99.1% compared to 99.78%).

In order to test this tag selection strategy, we combine the output from the set of sentences resulting in full parses when parsing STAG and the output resulting from parsing MTAG-SYS-DEF for the remaining sentences.<sup>7</sup> The accuracy achieved is illustrated in the row with HYBRID tag setup. This model is the only tag selection model which improves on the accuracy of the parser in terms of both macro- and micro-averaged  $F_1$  (compared over all test sentences). Further, as multiple-tpw input is only considered for the fraction of sentences for which a Frag parse occurs, the time taken to parse the sentences should also improve; ranging between time taken to parse the STAG and MTAG-SYS-DEF tag setups.

The last two rows of the table illustrate the upper bounds on precision and recall (for all test sentences) when parsing the MTAG-DEF-SYS tag setup. Carroll and Briscoe (2002) give full details of high-precision GR determination. Future work will address the issue of boosting the recall measure based on the high precision weighted GR output.

## 6 Conclusions

Contrasting the alternative tag models' performance both in terms of tagging and parser performance has supported Charniak's previous findings (Charniak et al., 1996): that single-tpw input to a parser is sensible given large speed improvements with small decrease in accuracy. However, if accuracy is a higher priority than efficiency, then multiple-tpw should be used as accuracy gains are available (0.66%  $F_1$ , relative error reduction of 2.27%). Further, significant

---

<sup>7</sup>Note that this could easily be implemented in the system.

Tag Setup	Microaverage			Macroaverage			Frag <sup>†</sup>	Time <sup>‡</sup>
	Prec	Rec	F <sub>1</sub>	Prec	Rec	F <sub>1</sub>		
STAG	71.06	70.96	71.01	58.08	58.98	58.53	21.25	0:03:50
	73.66	74.94	74.30	62.51	63.45	62.98		
MTAG-SYS-DEF	71.14	72.21	71.67	58.02	57.64	57.83	12.85	0:05:23
	73.09	74.71	73.89	61.44	60.81	61.12		
MTAG-SYS	70.10	71.39	70.74	56.26	57.59	56.92	10.00	0:18:27
	72.07	73.49	72.77	59.66	59.72	59.69		
MTAG	68.42	70.14	69.27	54.61	56.90	55.73	6.96	13:40:32
	70.48	72.24	71.35	60.39	59.00	59.69		
STAG-NE	73.53	69.66	71.54	59.10	58.16	58.63	25.00	0:03:13
	75.66	73.33	74.48	62.88	62.50	62.69		
MTAG-SYS-NE	72.54	70.49	71.50	56.96	56.92	56.94	12.68	0:10:57
	74.09	72.62	73.34	62.90	59.16	60.97		
MTAG-NE	71.32	69.30	70.30	55.21	56.13	55.67	9.28	0:45:51
	73.01	71.29	72.14	61.17	58.29	59.70		
MSD-WEIGHT-TOP	71.08	72.21	71.64	58.20	57.82	58.01	12.85	0:03:42
	72.99	74.69	73.83	61.68	61.04	61.38		
MSD-NUM-TOP	67.95	69.11	68.52	53.21	54.85	54.02	12.85	0:03:13
	69.59	71.26	70.41	54.90	57.57	56.20		
GOLD	72.94	73.12	73.03	60.53	61.04	60.78	14.46	0:04:39
	74.58	75.70	75.14	64.94	63.91	64.42		
HYBRID	71.59	72.39	71.99	59.02	60.36	59.68	-	-
Upper Prec	82.25	31.34	45.39	70.22	21.87	33.36	-	4:02:49
Upper Rec	17.81	87.74	29.60	20.11	82.26	32.32		

Table 3: Parser Performance. <sup>†</sup>Frag represents the percentage of Frag parses (the percentage of sentences for which full parses could not be found). <sup>‡</sup>Time is shown in format hours:minutes:seconds.

gains can be made in parser coverage by using multiple tags.

Interestingly, the system's default thresholds (optimised on Susanne data) for consideration of multiple tags appear near optimal (MTAG-SYS-DEF) for DepBank, achieving a trade-off between increased parse ambiguity and incorrect PoS tag errors.

None of the tag selection models outlined in Section 4.1 are able to improve on the system tagger's accuracy. The tag selection model outlined by Dalrymple (2004) proved to be the worst performing model. The parser's inability to improve on the tagger's accuracy may reflect a problem with the integration of the tag probabilities into the statistical model of the parser and in future work we will investigate this issue.

The dynamic tag selection model is the only model to improve both macro- and micro-averaged  $F_1$ . Here, the known trade-off between increased parse ambiguity and PoS tag error provides a means to gauge PoS tag error based on parser output. Therefore, when no parses are found the model assumes that a PoS error is the cause and increases the number of tags considered.

## 7 Acknowledgements

The author would like to thank Ted Briscoe and John Carroll for many helpful discussions throughout the course of this work and for comments on previous drafts of this paper. This work was in most part funded by the Overseas Research Students Awards Scheme and the Poynton Scholarship appointed by the Cambridge Australia Trust in collaboration with the Cambridge Commonwealth Trust.

## References

- Ted Briscoe and John Carroll. 1995. Developing and evaluating a probabilistic lr parser of part-of-speech and punctuation labels. In *ACL/SIGPARSE 4th International Workshop on Parsing Technologies*, pages 48–58, Prague / Karlovy Vary, Czech Republic.
- Ted Briscoe and John Carroll. 2002. Robust accurate statistical annotation of general text. In *Conference on Language Resources and Evaluation*, pages 1499–1504, Palmas, Canary Islands, May.
- Ted Briscoe and John Carroll. 2005. Evaluating the speed and accuracy of a domain-independent statistical parser on the parc depbank. Submitted.
- John Carroll and Ted Briscoe. 2002. High precision extraction of grammatical relations. In *19th International Conference on Computational Linguistics*, Taipei, Taiwan.
- John Carroll, Ted Briscoe, and Antonio Sanfilippo. 1998. Parser evaluation: a survey and a new proposal. In *1st International Conference on Language Resources and Evaluation*, pages 447–454, Granada.
- John Carroll. 1993. *Practical unification-based parsing of natural language*. Ph.D. thesis, Computer Laboratory, University of Cambridge. Technical Report No. 314.
- Eugene Charniak, Glenn Carroll, John Adcock, Antony Cassandra, Yoshihiko Gotoh, Jeremy Katz, Michael Littman, and John McCann. 1996. Taggers for parsers. *Artificial Intelligence*.
- Stephen Clark and James Curran. 2004. The importance of supertagging for wide-coverage CCG parsing. In *20th International Conference on Computational Linguistics*, Geneva, Switzerland.
- Mary Dalrymple. 2004. How much can part-of-speech tagging help parsing? Available: <http://users.ox.ac.uk/~cpgl0015/bib.html> [2005, 3 June].
- David Elworthy. 1994. Does baum-welch re-estimation help taggers? In *Fourth Conference on Applied Natural Language Processing*, pages 53–58.
- Ronald Kaplan, Stephen Riezler, Tracy King, John Maxwell, Alexander Vasserman, and Richard Crouch. 2004. Speed and accuracy in shallow and deep stochastic parsing. In *Human Language Technology conference / North American chapter of the Association for Computational Linguistics annual meeting*, pages 97–113, Boston, Massachusetts, May.
- Tracy King, Richard Crouch, Stephen Riezler, Mary Dalrymple, and Ronald Kaplan. 2003. The parc700 dependency bank. In *4th International Workshop on Linguistically Interpreted Corpora*.
- Dekang Lin. 1998. Dependency-based evaluation of minipar. In *Workshop at Language Resources and Evaluation on The Evaluation of Parsing Systems*, Granada, Spain.
- Rebecca Watson, John Carroll, and Ted Briscoe. 2005. Efficient extraction of grammatical relations. To be published.