# Evaluating a Statistical CCG Parser on Wikipedia

**Matthew Honnibal** **Joel Nothman** **James R. Curran**
School of Information Technologies
University of Sydney
NSW 2006, Australia
{mhonn,joel,james}@it.usyd.edu.au

## Abstract

The vast majority of parser evaluation is conducted on the 1984 Wall Street Journal (WSJ). In-domain evaluation of this kind is important for system development, but gives little indication about how the parser will perform on many practical problems.

Wikipedia is an interesting domain for parsing that has so far been underexplored. We present statistical parsing results that for the first time provide information about what sort of performance a user parsing Wikipedia text can expect.

We find that the C&C parser's standard model is 4.3% less accurate on Wikipedia text, but that a simple self-training exercise reduces the gap to 3.8%. The self-training also speeds up the parser on newswire text by 20%.

## 1 Introduction

Modern statistical parsers are able to retrieve accurate syntactic analyses for sentences that closely match the domain of the parser's training data. Breaking this domain dependence is now one of the main challenges for increasing the industrial viability of statistical parsers. Substantial progress has been made in adapting parsers from newswire domains to scientific domains, especially for biomedical literature (Nivre et al., 2007). However, there is also substantial interest in parsing encyclopedia text, particularly Wikipedia.

Wikipedia has become an influential resource for NLP for many reasons. In addition to its variety of interesting metadata, it is massive, constantly updated, and multilingual. Wikipedia is now given its own submission keyword in general CL conferences, and there are workshops largely centred around exploiting it and other collaborative semantic resources.

Despite this interest, there have been few investigations into how accurately existing NLP processing tools work on Wikipedia text. If it is found that Wikipedia text poses new challenges for our processing tools, then our results will constitute a baseline for future development. On the other hand, if we find that models trained on newswire text perform well, we will have discovered another interesting way Wikipedia text can be exploited.

This paper presents the first evaluation of a statistical parser on Wikipedia text. The only previous published results we are aware of were described by Ytrestøl et al. (2009), who ran the LinGo HPSG parser over Wikipedia, and found that the correct parse was in the top 500 returned parses for 60% of sentences. This is an interesting result, but one that gives little indication of how well a user could expect a parser to actually annotate Wikipedia text, or how to go about adjusting one if its performance is inadequate.

To investigate this, we randomly selected 200 sentences from Wikipedia, and hand-labelled them with CCG annotation in order to evaluate the C&C parser (Clark and Curran, 2007). C&C is the fastest deep-grammar parser, making it a likely choice for parsing Wikipedia, given its size.

Even at the parser's WSJ speeds, it would take about 18 days to parse the current English Wikipedia on a single CPU. We find that the parser is 54% slower on Wikipedia text, so parsing a full dump is inconvenient at best. The parser is only 4.3% less accurate, however.

We then examine how these figures might be improved. We try a simple domain adaptation experiment, using self-training. One of our experiments, which involves self-training using the Simple English Wikipedia, improves the accuracy of the parser's standard model on Wikipedia by 0.8%. The bootstrapping also makes the parser faster. Parse speeds on newswire text improve 20%, and speeds on Wikipedia improve by 34%.



*Proceedings of the 2009 Workshop on the People's Web Meets NLP, ACL-IJCNLP 2009*, pages 38–41,
Suntec, Singapore, 7 August 2009. 

| Corpus | Sentences | Mean length |
|---|---|---|
| WSJ 02-21 | 39,607 | 23.5 |
| FEW | 889,027 (586,724) | 22.4 (16.6) |
| SEW | 224,251 (187,321) | 16.5 (14.1) |

Table 1: Sentence lengths before (and after) length filter.

## 2 CCG Parsing

Combinatory Categorial Grammar (CCG) (Steedman, 2000) is a linguistically motivated grammar formalism with several advantages for NLP. Like HPSG, LFG and LTAG, a CCG parse recovers the semantic structure of a sentence, including long-range dependencies and complement/adjunct distinctions, providing substantially more information than skeletal brackets.

Clark and Curran (2007) describe how a fast and accurate CCG parser can be trained from CCGbank (Hockenmaier and Steedman, 2007). One of the keys to the system's success is *supertagging* (Bangalore and Joshi, 1999). Supertagging is the assignment of lexical categories before parsing. The parser is given only tags assigned a high probability, greatly restricting the search space it must explore. We use this system, referred to as C&C, for our parsing experiments.

## 3 Processing Wikipedia Data

We began by processing all articles from the March 2009 dump of Simple English Wikipedia (SEW) and the matching Full English Wikipedia (FEW) articles. SEW is an online encyclopedia written in basic English. It has stylistic guidelines that instruct contributors to use basic vocabulary and syntax, to improve the articles' readability. This might make SEW text easier to parse, making it useful for our self-training experiments.

`mwlib` (PediaPress, 2007) was used to parse the MediaWiki markup. We did not expand templates, and retained only paragraph text tokenized according to the WSJ, after it was split into sentences using the NLTK (Loper and Bird, 2002) implementation of Punkt (Kiss and Strunk, 2006) parameterised on Wikipedia text. Finally, we discarded incorrectly parsed markup and other noise.

We also introduced a sentence length filter for the domain adaptation data (but not the evaluation data), discarding sentences longer than 25 words or shorter than 3 words. The length filter was used to gather sentences that would be easier to parse. The effect of this filter is shown in Table 1.

## 4 Self-training Methodology

To investigate how the parser could be improved on Wikipedia text, we experimented with semi-supervised learning. We chose a simple method, self-training. Unlabelled data is annotated by the system, and the predictions are taken as truth and integrated into the training system.

Steedman et al. (2003) showed that the selection of sentences for semi-supervised parsing is very important. There are two issues: the *accuracy* with which the data can be parsed, which determines how noisy the new training data will be; and the *utility* of the examples, which determines how informative the examples will be.

We experimented with a novel source of data to balance these two concerns. Simple English Wikipedia imposes editorial guidelines on the length and syntactic style authors can use. This text should be easier to parse, lowering the noise, but the syntactic restrictions might mean its examples have lower utility for adapting the parser to the full English Wikipedia.

We train the C&C supertagger and parser (Clark and Curran, 2007) on sections 02-21 of the Wall Street Journal (WSJ) marked up with CCG annotations (Hockenmaier and Steedman, 2007) in the standard way. We then parse all of the Simple English Wikipedia remaining after our preprocessing. We discard the 826 sentences the parser could not find an analysis for, and set aside 1,486 randomly selected sentences as a future development set, leaving a corpus of 185,000 automatically parsed sentences (2.6 million words).

We retrain the supertagger on a simple concatenation of the 39,607 WSJ training sentences and the Wikipedia sentences, and then use it with the normal-form derivations and hybrid dependencies model distributed with the parser[1].

We repeated our experiments using text from the full English Wikipedia (FEW) for articles whose names match an article in SEW. We randomly selected a sample of 185,000 sentences from these, to match the size of the SEW corpus.

We also performed a set of experiments where we re-parsed the corpus using the updated supertagger and retrained on output, the logic being that the updated model might make fewer errors, producing higher quality training data. This iterative retraining was found to have no effect.

[1] `http://svn.ask.it.usyd.edu.au/trac/candc`

| Model | WSJ Section 23 | | | | | Wiki 200 | | | | | Wiki 90k | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $P$ | $R$ | $F$ | speed | cov | $P$ | $R$ | $F$ | speed | cov | speed | cov |
| WSJ derivs | 85.51 | 84.62 | 85.06 | 545 | 99.58 | 81.20 | 80.51 | 80.86 | 394 | 99.00 | 239 | 98.81 |
| SEW derivs | 85.06 | 84.11 | 84.59 | 634 | 99.75 | 81.96 | 81.34 | 81.65 | 739 | 99.50 | 264 | 99.11 |
| FEW derivs | 85.24 | 84.32 | 84.78 | 653 | 99.79 | 81.94 | 81.36 | 81.65 | 776 | 99.50 | 296 | 99.15 |
| WSJ hybrid | 86.20 | 84.80 | 85.50 | 481 | 99.58 | 81.93 | 80.51 | 81.22 | 372 | 99.00 | 221 | 98.81 |
| SEW hybrid | 85.80 | 84.30 | 85.05 | 571 | 99.75 | 82.16 | 80.49 | 81.32 | 643 | 99.50 | 257 | 99.11 |
| FEW hybrid | 85.94 | 84.46 | 85.19 | 577 | 99.79 | 82.49 | 81.03 | 81.75 | 665 | 99.50 | 275 | 99.15 |

Table 2: Parsing results with automatic POS tags. SEW and FEW models incorporate self-training.

## 5 Annotating the Wikipedia Data

We manually annotated a Full English Wikipedia evaluation set of 200 sentences. The sentences were sampled at random from the 5000 articles that were linked to most often by Wikipedia pages. Articles used for self-training were excluded.

The annotation was conducted by one annotator. First, we parsed the sentences using the C&C parser. We then manually corrected the supertags, supplied them back to the parser, and corrected the parses using a GUI. The interface allowed the annotator to specify bracket constraints until the parser selected the correct analysis. The annotation took about 20 hours in total.

We used the CCGbank manual (Hockenmaier and Steedman, 2005) as the guidelines for our annotation. There were, however, some systematic differences from CCGbank, due to the faulty noun phrase bracketing and complement/adjunct distinctions inherited from the Penn Treebank.

## 6 Results

The results in this section refer to precision, recall and $F$-Score over labelled CCG dependencies, which are 5-tuples (head, child, category, slot, range). Speed is reported as words per second, using a single core 2.6 GHz Pentium 4 Xeon.

### 6.1 Out-of-the-Box Performance

Our experiments were performed using two models provided with v1.02 of the C&C parser. The *derivs* model is calculated using features from the Eisner (1996) normal form derivation. This is the model C&C recommend for general use, because it is simpler and faster to train. The *hybrid* model achieves the best published results for CCG parsing (Clark and Curran, 2007), so we also experimented with this model. The models' performance is shown in the WSJ rows of Table 2. We report accuracy using automatic POS tags, since we did not correct the POS tags in the Wikipedia data.

The derivs and hybrid models show a similar drop in performance on Wikipedia, of about 4.3%. Since this is the first accuracy evaluation conducted on Wikipedia, it is possible that Wikipedia data is simply harder to parse, possibly due to its wider vocabulary. It is also possible that our manual annotation made the task slightly harder, because we did not reproduce the CCGbank noun phrase bracketing and complement/adjunct distinction errors.

We also report the parser's speed and coverage on Wikipedia. Since these results do not require labelled data, we used a sample of 90,000 sentences to obtain more reliable figures. Speeds varied enormously between this sample and the 200 annotated sentences. A length comparison reveals that our manually annotated sentences are slightly shorter, with a mean of 20 tokens per sentence. Shorter sentences are often easier to parse, so this issue may have affected our accuracy results, too.

The 54% drop in speed on Wikipedia text is explained by the way the supertagger and parser are integrated. The supertagger supplies the parser with a beam of categories. If parsing fails, the chart is reinitialised with a wider beam and it tries again. These failures occur more often when the supertagger cannot produce a high quality tag sequence, particularly if the problem is in the tag dictionary, which constrains the supertagger's selections for frequent words. This is why we focused on the supertagger in our domain adaptation experiments.

### 6.2 Domain Adaptation Experiments

The inclusion of parsed data from Wikipedia articles in the supertagger's training data improves its accuracy on Wikipedia data, with the FEW enhanced model achieving 89.86% accuracy, compared with the original accuracy of 88.77%. The SEW enhanced supertagger achieved 89.45% accuracy. The *derivs* model parser improves in accuracy by 0.8%, the *hybrid* model by 0.5%.

The out-of-domain training data had little impact on the models' accuracy on the WSJ, but did improve parse speed by 20%, as it did on Wikipedia. The speed increases because the supertagger's beam width is decided by its confidence scores, which are more narrowly distributed after the model has been trained with more data.

After self-training, the *derivs* and *hybrid* models performed equally accurately. With no reason to use the hybrid model, the total speed increase is 34%. With our pre-processing, the full Wikipedia dump had close to 1 billion words, so speed is an important factor.

Overall, our simple self-training experiment was quite successful. This result may seem surprising given that the CoNLL 2007 participants generally failed to use similar resources to adapt dependency parsers to biomedical text (Dredze et al., 2007). However, our results confirm Rimell and Clark's (2009) finding that the C&C parser's division of labour between the supertagger and parser make it easier to adapt to new domains.

## 7 Conclusion

We have presented the first investigation into statistical parsing on Wikipedia data. The parser's accuracy dropped 4.3%, suggesting that the system is still useable out-of-the-box. The parser is also 54% slower on Wikipedia text. Parsing a full Wikipedia dump would therefore take about 52 days of CPU time using our 5-year-old architecture, which is inconvenient, but manageable over multiple processors.

Using simple domain adaptation techniques, we are able to increase the parser's accuracy on Wikipedia, with the fastest model improving in accuracy by 0.8%. This closed the gap in accuracy between the two parser models, removing the need to use the slower *hybrid* model. This allowed us to achieve an overall speed improvement of 34%.

Our results reflect the general trend that NLP systems perform worse on foreign domains (Gildea, 2001). Our results also support Rimell and Clark's (2009) conclusion that because C&C is highly lexicalised, domain adaptation is largely a process of adapting the supertagger.

A particularly promising aspect of these results is that the parse speeds on the Wall Street Journal improved, by 15%. This improvement came with no loss in accuracy, and suggests that further bootstrapping experiments are likely to be successful.

## 8 Acknowledgements

We would like to thank Stephen Clark and the anonymous reviewers for their helpful feedback. Joel was supported by a Capital Markets CRC PhD scholarship and a University of Sydney Vice-Chancellor's Research Scholarship.

## References

Srinivas Bangalore and Aravind Joshi. 1999. Supertagging: An approach to almost parsing. *Computational Linguistics*, 25(2):237–265.

Stephen Clark and James R. Curran. 2007. Wide-coverage efficient statistical parsing with CCG and log-linear models. *Computational Linguistics*, 33(4):493–552.

Mark Dredze, John Blitzer, Partha Pratim Talukdar, Kuzman Ganchev, João Graca, and Fernando Pereira. 2007. Frustratingly hard domain adaptation for dependency parsing. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 1051–1055. ACL, Prague, Czech Republic.

Jason Eisner. 1996. Efficient normal-form parsing for Combinatory Categorial Grammar. In *Proceedings of the Association for Computational Linguistics*, pages 79–86. Santa Cruz, CA, USA.

Daniel Gildea. 2001. Corpus variation and parser performance. In *Proceedings of the EMNLP Conference*, pages 167–202. Pittsburgh, PA.

Julia Hockenmaier and Mark Steedman. 2005. CCGbank manual. Technical Report MS-CIS-05-09, Department of Computer Science, University of Pennsylvania.

Julia Hockenmaier and Mark Steedman. 2007. CCGbank: a corpus of CCG derivations and dependency structures extracted from the Penn Treebank. *Computational Linguistics*, 33(3):355–396.

Tibor Kiss and Jan Strunk. 2006. Unsupervised multilingual sentence boundary detection. *Computational Linguistics*, 32(4):485–525.

Edward Loper and Steven Bird. 2002. NLTK: The natural language toolkit.

Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. 2007. The CoNLL 2007 shared task on dependency parsing. In *Proceedings of the CoNLL Shared Task Session*, pages 915–932. Prague, Czech Republic.

PediaPress. 2007. `mwlib` MediaWiki parsing library. `http://code.pediapress.com`.

Laura Rimell and Stephen Clark. 2009. Porting a lexicalized-grammar parser to the biomedical domain. *Journal of Biomedical Informatics*. (in press).

Mark Steedman. 2000. *The Syntactic Process*. The MIT Press, Cambridge, MA.

Mark Steedman, Rebecca Hwa, Stephen Clark, Miles Osborne, Anoop Sarkar, Julia Hockenmaier, Paul Ruhlen, Steven Baker, and Jeremiah Crim. 2003. Example selection for bootstrapping statistical parsers. In *Proceedings of HLT-NAACL 2003*. Edmonton, Alberta.

Gisle Ytrestøl, Stephan Oepen, and Daniel Flickinger. 2009. Extracting and annotating Wikipedia sub-domains. In *Proceedings of the 7th International Workshop on Treebanks and Linguistic Theories*, pages 185–197. Groningen, Netherlands.