1. Introduction
An enormous amount of the world’s data is in the form of natural language. With access to the meaning of natural language computers could perform tasks such as question answering and sentiment analysis. Parsers are a crucial part of extracting meaning from natural language, but currently they are too slow to be applied effectively.

The first step in parsing, attaching lexical roles to the words in a sentence, or supertagging [5], is particularly important for parsers of lexicalised grammars such as Combinatory Categorial Grammar (CCG) [7].

If we can reduce the number of supertags assigned to each word by constructing models based on more data, the parser will have less work to do. But how can we get more labelled data without great expense?

Here I investigate self-training of the supertagger in the C&C parser [1, 2], i.e. using the parser to generate training data, which is then used to retrain its supertagger.

2. Aims
Increase parsing speed without decreasing accuracy
- Parallelise the training process
- Implement perceptron algorithms
- Construct models using much larger training sets
- Explore more complex features

3. CCG Supertag Ambiguity
These sentences show one form of ambiguity that the parser must handle. Note how the change of supertag for ‘with’ leads to a completely different derivation.

```
I ate pizza with cutlery
```

4. Approach
Parallelisation - Using the Message Passing Interface (MPI) I implemented parallel versions of the feature extraction and model estimation processes.

5. Self-Training on WSJ
By training on extra data from the Wall Street Journal, with labels provided by the baseline system, we can improve speed without losing accuracy. See Figure 4 for the results of these experiments.

6. Domain Adaptation
The same self-training technique was applied to Wikipedia, which also led to increased parsing speeds, without loss of accuracy.

7. Algorithm for Parameter Optimisation
I created an algorithm to optimise parsing speed while maintaining full coverage, and explored a more sophisticated version that optimises for accuracy as well. See Figures 6 and 7 for some of this exploration.

8. Further work
- Larger self-training experiments
- Adaptation to other domains, e.g. Biomedical
- More advanced features
- Co-training using multiple estimation algorithms
- Perceptron multitagging
- Online learning
- Less restricted parsing for automatic annotation
- Global features for whole sentence tagging

9. Conclusion
My work has made model estimation orders of magnitude faster. By adapting models to specific domains I increased parsing speed on either newspaper text or Wikipedia by 30%, while maintaining accuracy. For further information see [6].

10. Acknowledgements
This work was supported by a University of Sydney Merit Scholarship.

References