Adaptive Supertagging for Faster Parsing

Author: Jonathan Kay Kummerfeld *jkum0593@it.usyd.edu.au*

Supervisor: Dr. James R. Curran

School of Information Technologies

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], ie. 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
NP	$(S\backslash NP)/NP$	NP	$((S\backslash NP)\backslash(S\backslash NP))/NP$	NP
	$S\backslash NP$ $>$		$(S\backslash NP)\backslash(S\backslash NP)$ $>$	
	$S\backslash NP$ $<$			
S $<$				

I	ate	pizza	with	anchovies
NP	$(S\backslash NP)/NP$	NP	$(NP\backslash NP)/NP$	NP
			$NP\backslash NP$ $>$	
		NP $<$		
	$S\backslash NP$ $>$			
S $<$				

Figure 1: *Two CCG derivations with PP ambiguity.*

4. Approach

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

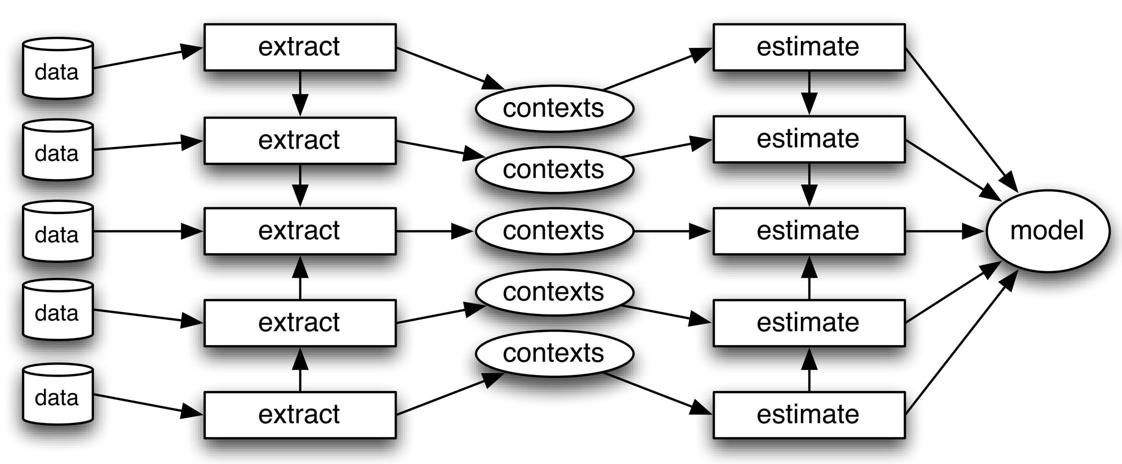

Figure 2: *Parallel Feature Extraction*

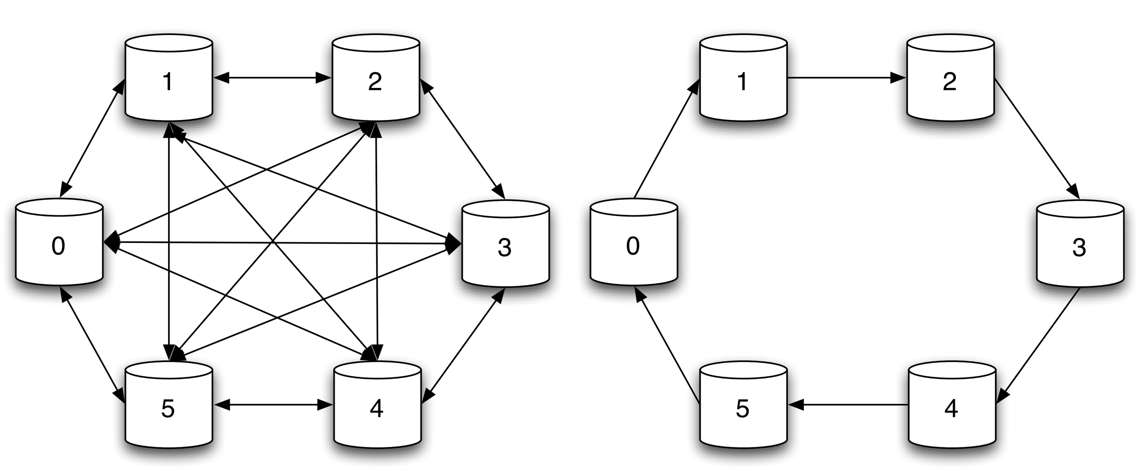

Figure 3: *Parallel Model Estimation, Maximum Entropy Models and Perceptrons*

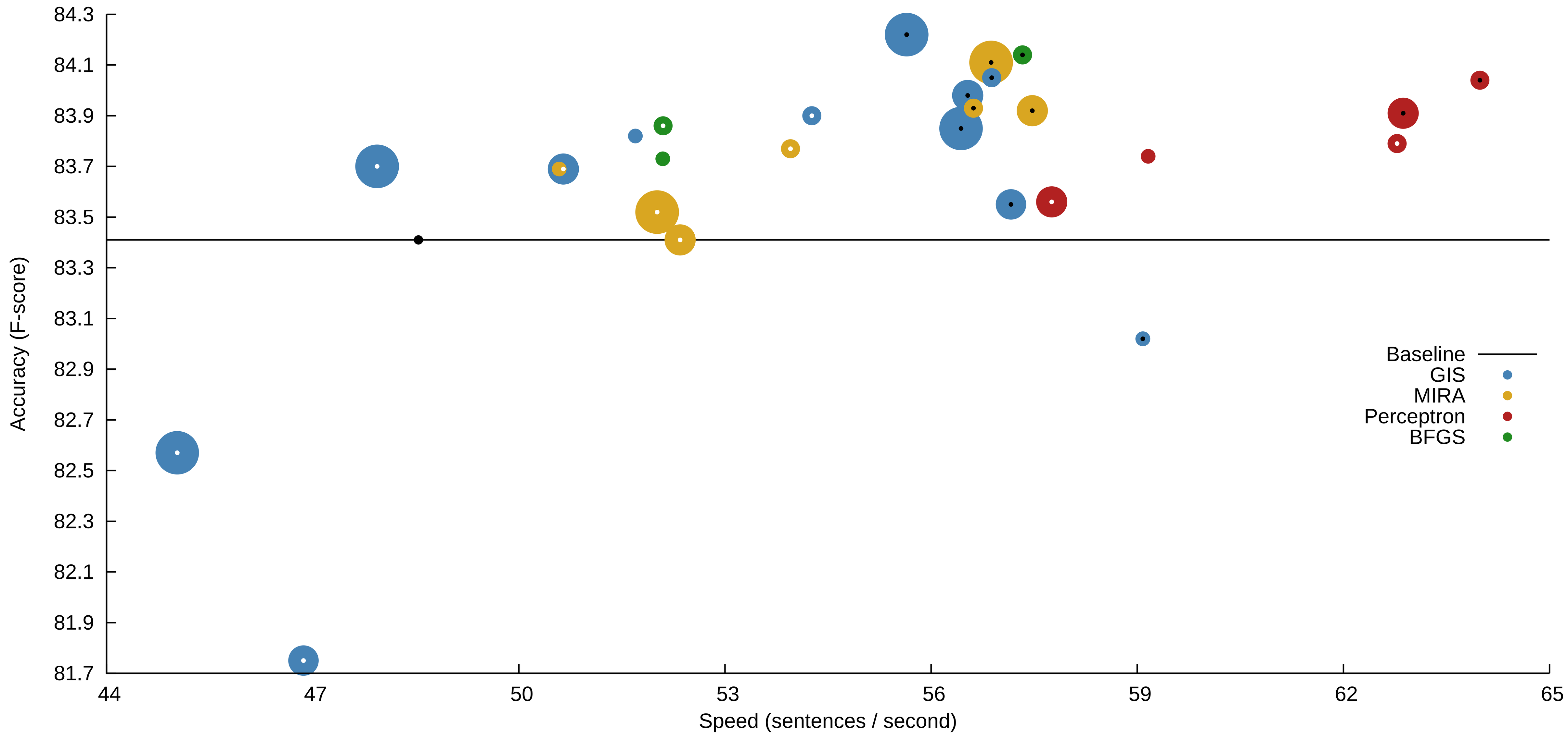

Figure 4: *Performance for various algorithms and training sets The size of each circle represents the amount of training data, the centre indicates the type of data – black for WSJ, white for Wikipedia, and the origin is the baseline model.*

Averaged Perceptron - As for the standard Perceptron, except that the final matrix of weights returned by training contains the average of each weight over training, rather than the final value. [4]

Margin Infused Relaxed Algorithm (MIRA) - The same as the Averaged Perceptron, but when updating make as small a change as possible. The change must satisfy:

$$\min_{\bar{\tau}} \tfrac{1}{2}\textstyle\sum_r \|\ \bar{M}_r + \tau_r \bar{x}^t\ \|_2^2$$
$$\text{subject to: (1) } \tau_r \leq \delta_{r,y^t} \text{ for } r = 1, ..., k$$
$$\text{(2) } \textstyle\sum_{r=1}^{k} \tau_r = 0$$

Figure 5: *Equations for the MIRA update scheme [3]*

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.

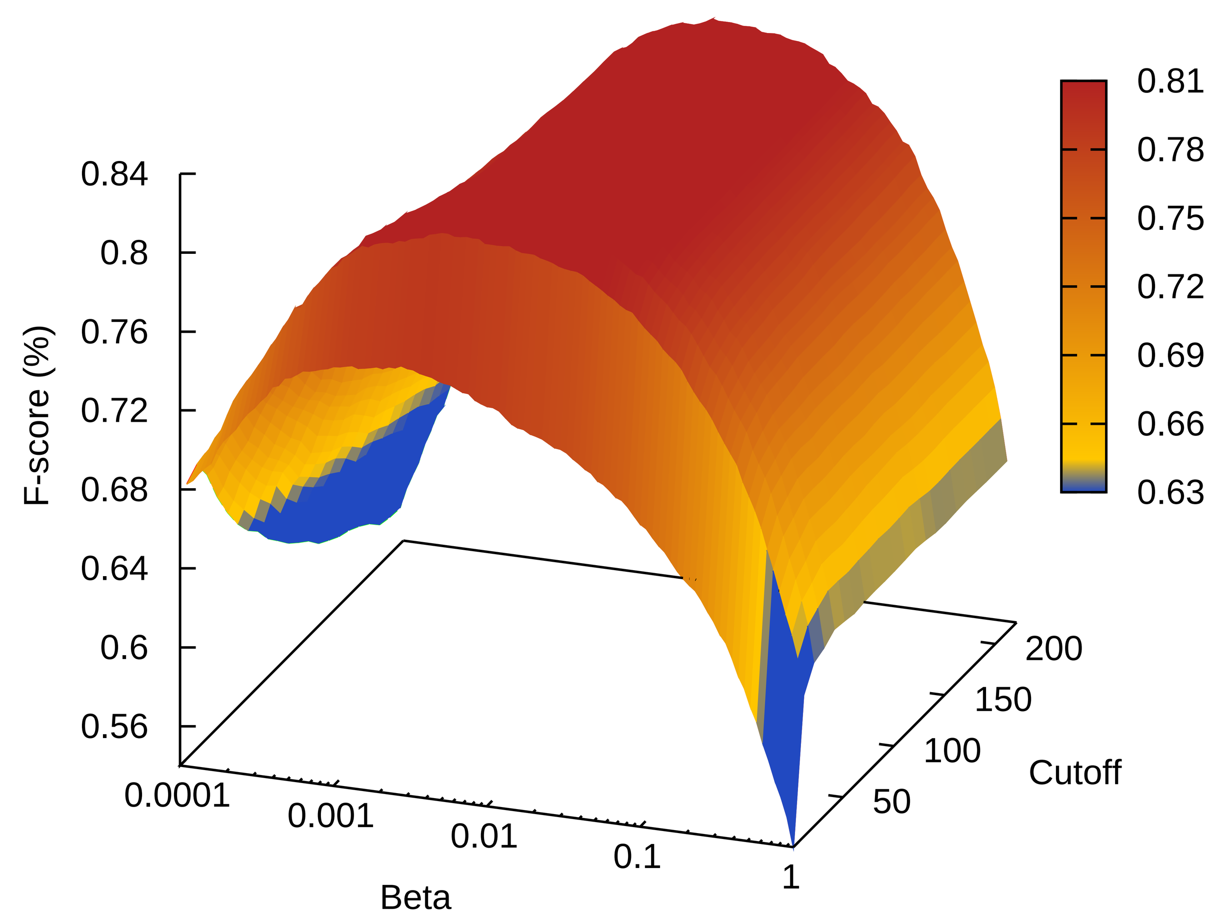

Figure 6: *Parser accuracy for a range of parameter settings.*

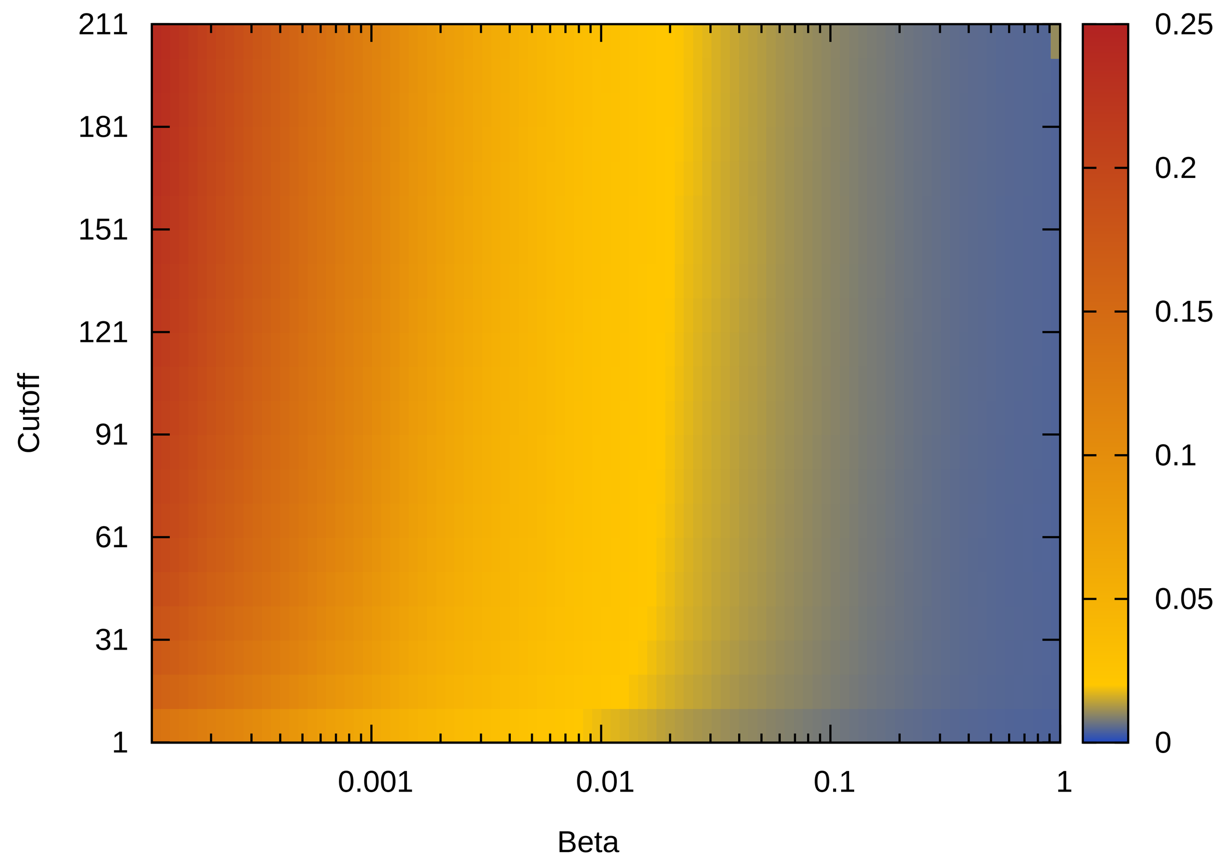

Figure 7: *Parsing time for a range of parameter settings.*

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

[1] Stephen Clark and James R. Curran. The importance of supertagging for wide-coverage ccg parsing. In *Proceedings of COLING 2004*, pages 282–288.

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

[3] Koby Crammer and Yoram Singer. Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research*, 3:951–991, 2003.

[4] Yoav Freund and Robert E. Schapire. Large margin classification using the perceptron algorithm. *Machine Learning*, 37(3):277–296, 1999.

[5] Aravind K. Joshi and Srinivas Bangalore. Disambiguation of super parts of speech (or supertags): almost parsing. In *Proceedings of the 15th conference on Computational linguistics*, pages 154–160, 1994.

[6] Jonathan K. Kummerfeld and James R. Curran. Faster parsing and supertagging model estimation. In *the proceedings of ALTW 2009*.

[7] Mark Steedman. *The Syntactic Process*. MIT Press, Cambridge, MA, USA, 2000.

The University of Sydney

University of Sydney, Camperdown, NSW, 2006, Australia