Faster Parsing and Supertagging Model Estimation

Jonathan K. Kummerfeld[a] James R. Curran[a] Jessika Roesner[b]

School of Information Technologies[a]
University of Sydney
Australia
{jkum0593,james}@it.usyd.edu.au

Department of Computer Science[b]
University of Texas at Austin
USA
jessi@mail.utexas.edu

ALTW 2009

Motivation – Parsing

Syntactic information is crucial for many tasks in NLP, such as QA and MT, but parsers are slow:

- State-of-the-art, usually $<$ 1 sentence / sec
- Fastest state-of-the-art, $<$ 50 sentences / sec

Far too slow to process the data available:

- $>$ 1,000,000,000,000 words of English online
- More coming

Tagging and Parsing

$$\text{One} \quad \text{claims} \quad \text{he} \quad \text{is} \quad \text{pro} - \text{choice}$$

Part of Speech Tagging

$$\begin{array}{ccccc} \text{One} & \text{claims} & \text{he} & \text{is} & \text{pro} - \text{choice} \\ NN & VBZ & PRP & VBZ & JJ \end{array}$$

Combinatory Categorial Grammar (CCG) – Supertagging

$$
\begin{array}{ccccc}
\text{One} & \text{claims} & \text{he} & \text{is} & \text{pro}-\text{choice} \\
\hline N & (S\backslash NP)/S & NP & (S\backslash NP)/(S\backslash NP) & S\backslash NP
\end{array}
$$

Combinatory Categorial Grammar (CCG) – Parsing

$$
\begin{array}{ccccc}
\text{One} & \text{claims} & \text{he} & \text{is} & \text{pro}-\text{choice} \\
\hline
N & (S\backslash NP)/S & NP & (S\backslash NP)/(S\backslash NP) & S\backslash NP \\
\cline{1-1} \cline{4-5}
NP & & & \multicolumn{2}{c}{S\backslash NP \quad >} \\
\cline{3-5}
& & \multicolumn{3}{c}{S \quad <} \\
\cline{2-5}
& \multicolumn{4}{c}{S\backslash NP \quad >} \\
\hline
\multicolumn{5}{c}{S \quad <}
\end{array}
$$

Supertagging Ambiguity

I ate pizza with cutlery

I ate pizza with anchovies

Supertagging Ambiguity

$$
\begin{array}{ccccc}
\text{I} & \text{ate} & \text{pizza} & \text{with} & \text{cutlery} \\
\hline
NP & (S\backslash NP)/NP & NP & ((S\backslash NP)\backslash(S\backslash NP))/NP & NP \\
 & \multicolumn{2}{c}{\overline{\qquad S\backslash NP \qquad}^{>}} & \multicolumn{2}{c}{\overline{\qquad (S\backslash NP)\backslash(S\backslash NP) \qquad}^{>}} \\
 & \multicolumn{4}{c}{\overline{\qquad\qquad S\backslash NP \qquad\qquad}^{<}} \\
\multicolumn{5}{c}{\overline{\qquad\qquad\qquad S \qquad\qquad\qquad}^{<}}
\end{array}
$$

$$
\begin{array}{ccccc}
\text{I} & \text{ate} & \text{pizza} & \text{with} & \text{anchovies} \\
\hline
NP & (S\backslash NP)/NP & NP & (NP\backslash NP)/NP & NP \\
 & & & \multicolumn{2}{c}{\overline{\qquad NP\backslash NP \qquad}^{>}} \\
 & & \multicolumn{3}{c}{\overline{\qquad\qquad NP \qquad\qquad}^{<}} \\
 & \multicolumn{4}{c}{\overline{\qquad\qquad S\backslash NP \qquad\qquad}^{>}} \\
\multicolumn{5}{c}{\overline{\qquad\qquad\qquad S \qquad\qquad\qquad}^{<}}
\end{array}
$$

Supertagging Ambiguity

I	ate	pizza	with	cutlery
NP	$(S\backslash NP)/NP$	NP	$((S\backslash NP)\backslash(S\backslash NP))/NP$	NP

$$\text{ate pizza} \Rightarrow_{>} S\backslash NP \qquad \text{with cutlery} \Rightarrow_{>} (S\backslash NP)\backslash(S\backslash NP)$$
$$\Rightarrow_{<} S\backslash NP$$
$$\Rightarrow_{<} S$$

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

$$\text{with anchovies} \Rightarrow_{>} NP\backslash NP$$
$$\Rightarrow_{<} NP$$
$$\Rightarrow_{>} S\backslash NP$$
$$\Rightarrow_{<} S$$

Motivation – Parsing

The key idea behind the speed of the fastest parsers today is to shift work from parsing to tagging:
For n words, each with k tags

- Tagging – $O(nk)$
- Parsing – $O(n^3k^2)$

Outline

Core Idea

- Provide fewer tags, but still include the tags the parser would have used anyway

Implementation

- Perceptron Algorithms
- Parallelisation

Results

- Modified rule usage
- Training data type and volume
- Algorithm comparison
- Feature extension

Ideal World

$$\frac{\text{One}}{N} \quad \frac{\text{claims}}{(S\backslash NP)/S} \quad \frac{\text{he}}{NP} \quad \frac{\text{is}}{(S\backslash NP)/(S\backslash NP)} \quad \frac{\text{pro}-\text{choice}}{S\backslash NP}$$

Current World – Problem

$$\begin{array}{ccccc} \text{One} & \text{claims} & \text{he} & \text{is} & \text{pro}-\text{choice} \\ \hline N & \color{red}{(S\backslash NP)/NP} & NP & (S\backslash NP)/(S\backslash NP) & S\backslash NP \end{array}$$

Current World – Solution

One	claims	he	is	pro – choice
N/N	$(S\backslash NP)/NP$	NP	$(S\backslash NP)/(S\backslash NP)$	$S\backslash NP$
N	N		$(S\backslash NP)/NP$	$(S\backslash NP)\backslash(S\backslash NP)$
$(S/S)/(S/S)$			$(S\backslash NP)/(S\backslash NP)$	$(S\backslash NP)/S$
				N
				$(S\backslash NP)/PP$
				$(S\backslash NP)/NP$
				N/N
				$(S\backslash NP)/(S\backslash NP)$

Adaptive Supertagging

One	claims	he	is	pro – choice
N/N	N	NP	$(S\backslash NP)/(S\backslash NP)$	$S\backslash NP$
			$(S\backslash NP)/NP$	$(S\backslash NP)/PP$
			$(S\backslash NP)/(S\backslash NP)$	$(S\backslash NP)/NP$
				N/N

How do we teach the supertagger to produce these tags?

Use the parser!

Outline

Core Idea

- Provide fewer tags, but still include the tags the parser would have used anyway

Implementation

- **Perceptron Algorithms**
- **Parallelisation**

Results

- Modified rule usage
- Training data type and volume
- Algorithm comparison
- Feature extension

Implementation

Component	Initial System	Additions
Statistical Feature Extraction	3 Types	+9 Types
	Single thread	Parallel
Parameter Estimation	BFGS, GIS	AP, MIRA
	Single thread	Parallel

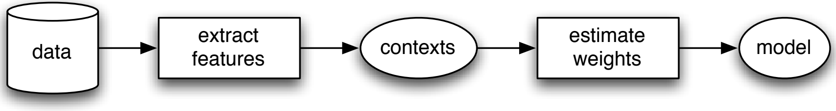
Implementation – Extra Constraint

Added a constraint that only allows Backward Composition to occur if both children are type raised

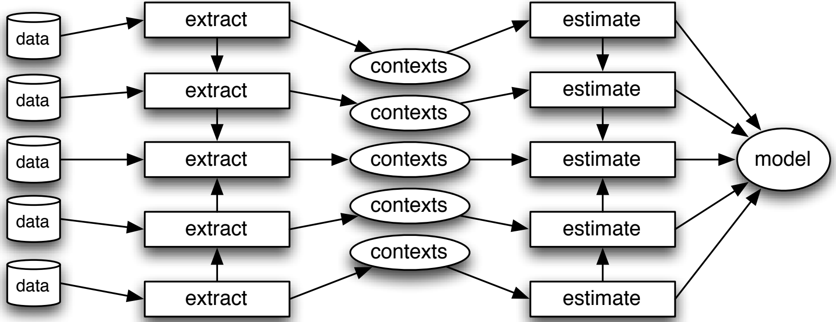
Implementation – AP and MIRA

Algorithm	Training Time (sec)		
	40k	80k	440k
GIS	7,200	14,000	*
BFGS	6,300	13,000	*
AP	76	160	950
MIRA	96	200	1,200

Implementation – Initial System



Implementation – Parallelised



Implementation – Parallelised Weight Estimation

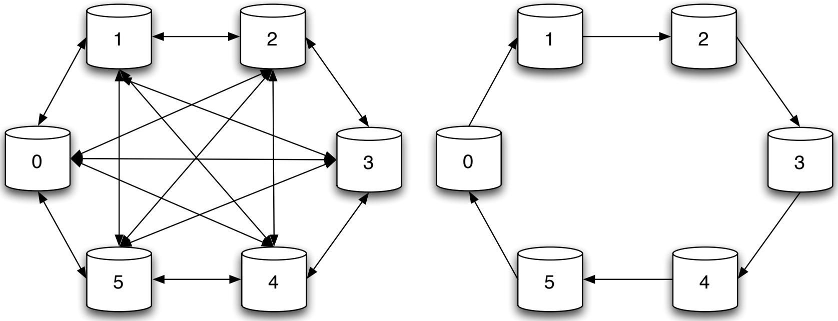


Figure: Information flow for parallel model estimation

Outline

Core Idea

- Provide fewer tags, but still include the tags the parser would have used anyway

Implementation

- Perceptron Algorithms
- Parallelisation

Results

- **Modified rule usage**
- **Training data type and volume**
- **Algorithm comparison**
- **Feature extension**

Extra Constraint on Rule Application

Parser	F-score (%)	Speed (sent / sec)
C&C 1.02	83.22	31.7
Modified	83.41	47.8

Plan

- Acquire a large set of unannotated data – Wikipedia
- Parse the corpus
- Retrain the supertagger, using the parsed sentences

Variations

- Amount of data
- Estimation algorithms
- Feature set

Training Data Type and Volume

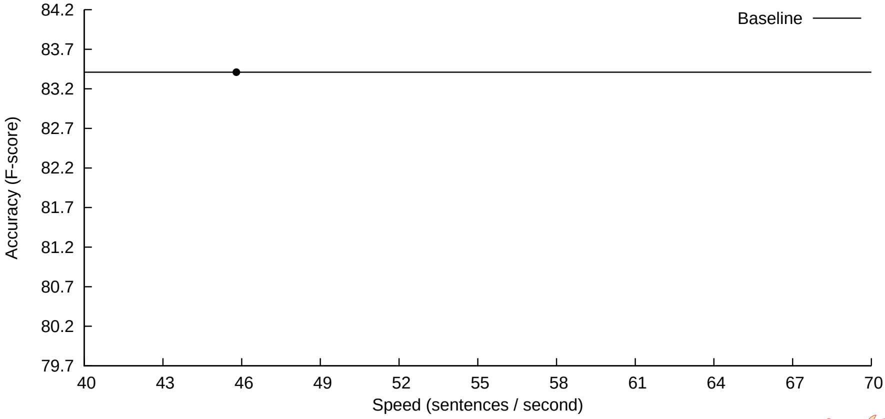


Figure: Evaluation on the Wall Street Journal

Training Data Type and Volume

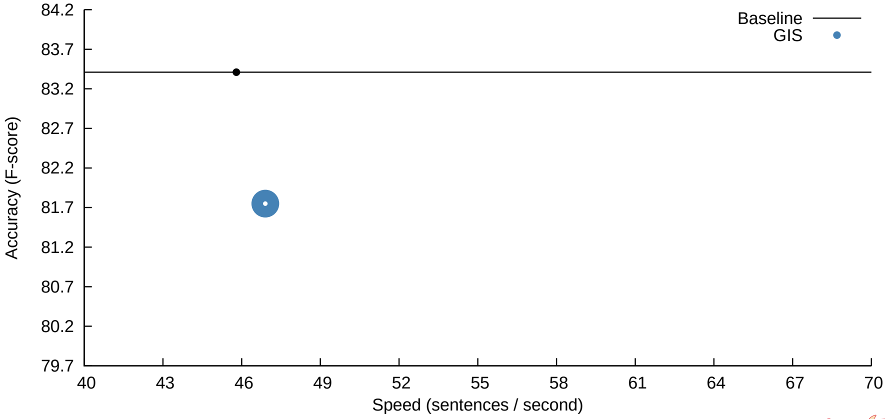


Figure: Evaluation on the Wall Street Journal

Training Data Type and Volume

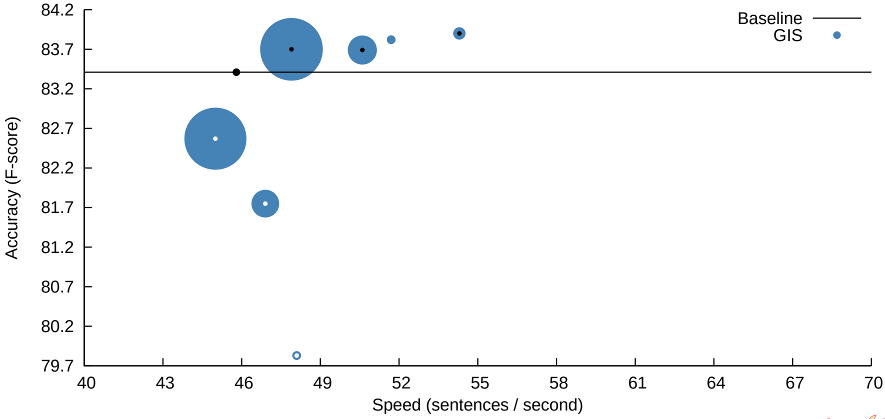


Figure: Evaluation on the Wall Street Journal

Training Data Type and Volume

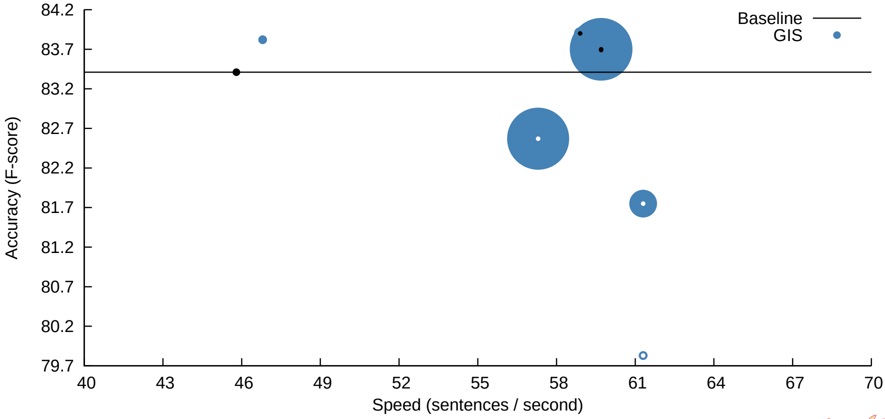


Figure: Evaluation on Wikipedia

Algorithm Comparison

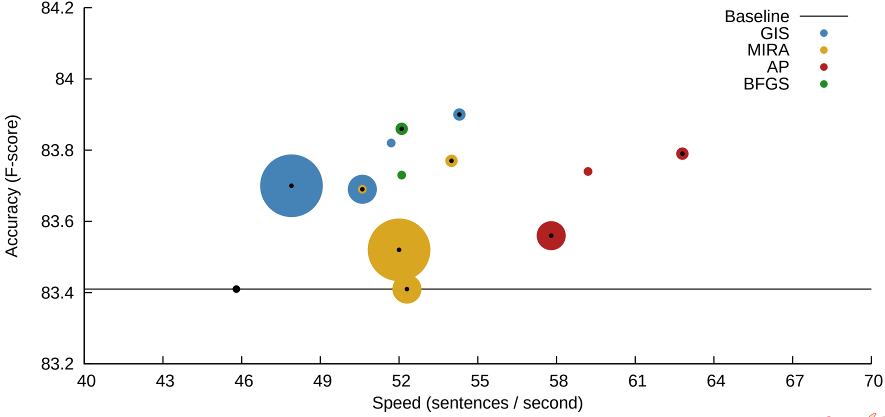


Figure: Evaluation on the Wall Street Journal

Algorithm Comparison

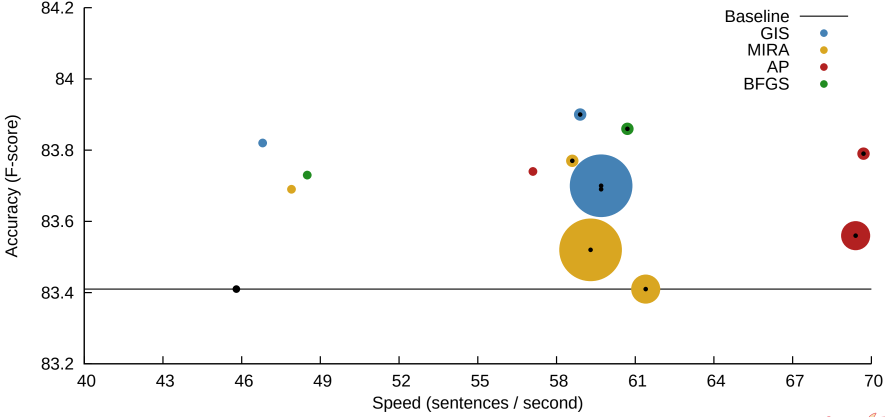


Figure: Evaluation on Wikipedia

Feature Extension

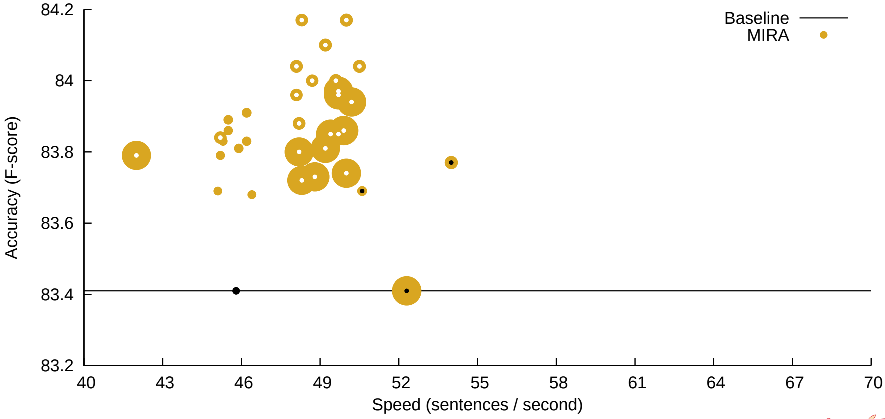


Figure: Evaluation on the Wall Street Journal

Feature Extension

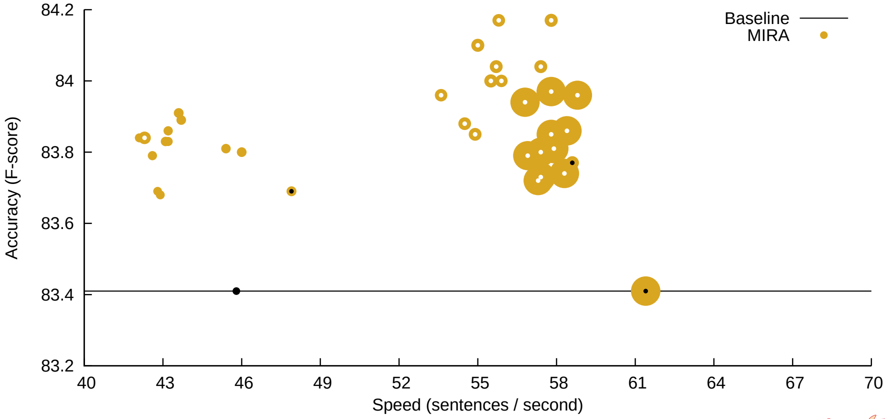


Figure: Evaluation on Wikipedia

Future Work

- Other domains
- Expanded training sets
- Co-training
- Online learning

Conclusion

Improved training:

- Enabled access to more text
- Constructed an effective source of more text

Improved parsing speed:

- Added an extra constraint on rule usage
- Trained models that are adapted to the parser

Improved parsing accuracy:

- Constructed statistical models using more evidence
- Expanded the set of statistical features

Conclusion

Metric	Initial	Final	Ratio
Training			
Sentences	40k	80k	2
Time (secs)	6,300	160	1/40
Accuracy			
F-score (%)	83.22	83.79	n/a
Speed			
WSJ (sents / sec)	31.7	62.8	2.0
Wikipedia (sents / sec)	30.8	69.7	2.3

Acknowledgements

- Johns Hopkins University, CLSP Summer Workshop
- Capital Markets Cooperative Research Centre Limited