Large-Scale Syntactic Processing: Parsing the Web

JHU 2009 Summer Research Workshop
The Team

- Stephen Clark (Cambridge, UK)
- Ann Copestake (Cambridge, UK)
- James Curran (Sydney, Australia)
- Byung-Gyu Ahn (Johns Hopkins, US)
- James Haggerty (Sydney, Australia)
- Aurelie Herbelot (Cambridge, UK)
- Yue Zhang (Oxford, UK)
- Jessi Roesner (Texas, US)
- Curt Van Wyk (Northwestern, US)
Parsing the Web

- Intelligent information access using natural language queries
  - question answering/semantic search, e.g. Powerset
- Data for large-scale knowledge acquisition
- Focus on (English) Wikipedia
Better Parsing Technology

- More *adaptable* parsers
- More *efficient* parsers
- More *accurate* parsers
Which Parser?

- Parser based on Combinatory Categorial Grammar (Clark and Curran, 2007)
- Robust parser using a wide-coverage *lexicalised* grammar

Research Questions

- How does a newspaper-trained lexicalised-grammar parser perform on Wikipedia text?
Research Questions

- How does a newspaper-trained lexicalised-grammar parser perform on Wikipedia text?
- Can we improve the accuracy of the parser on Wikipedia text?
Research Questions

- How does a newspaper-trained lexicalised-grammar parser perform on Wikipedia text?
- Can we improve the accuracy of the parser on Wikipedia text?
- Can we improve the efficiency of the parser?
Research Questions

• How does a newspaper-trained lexicalised-grammar parser perform on Wikipedia text?
• Can we improve the accuracy of the parser on Wikipedia text?
• Can we improve the efficiency of the parser?
• Can we use large amounts of parsed data to improve coordination disambiguation?
Research Questions

• How does a newspaper-trained lexicalised-grammar parser perform on Wikipedia text?
• Can we improve the accuracy of the parser on Wikipedia text?
• Can we improve the efficiency of the parser?
• Can we use large amounts of parsed data to improve coordination disambiguation?
• Can we relieve the annotation bottleneck by training on a novel form of bootstrapped Wikipedia data?
Aims and Goals

- Produce the parser of choice for the NLP community (and any other community interested in parsing text – semantic web, bioinformatics, . . .)
  - including an end-to-end pipeline starting with HTML files
- Produce a parsed version of Wikipedia, in multiple output formats
- Investigate a number of research questions w.r.t. improving the accuracy and efficiency of a lexicalised-grammar chart parser applied to non-newspaper text
Combinatory Categorial Grammar

interleukin – 10 inhibits production

NP (S\NP)/NP NP
Combinatory Categorial Grammar

\[
\text{interleukin} - 10 \quad \text{inhibits} \quad \text{production}
\]

\[
\begin{align*}
\text{NP} & \quad (S\backslash NP)/NP \\
& \quad S\backslash NP
\end{align*}
\]

> forward application
Combinatory Categorial Grammar

\[
\begin{align*}
\text{interleukin} - 10 & \quad \text{inhibits} \quad \text{production} \\
NP & \quad (S \backslash NP)/NP \quad NP \\
& \quad S \backslash NP > \\
& \quad S <
\end{align*}
\]

> forward application
< backward application
**CCG vs. CFG**

```
S
\[ S\backslash NP \]
  \[ NP \]
  interleukin-10
  inhibits
  production
  \[ (S\backslash NP)/NP \]
  \[ NP \]
```
C CG vs. CFG

interleukin-10 inhibits production

CCG has some additional combinatory rules (e.g. type-raising and composition) motivated by the desire to recover unbounded dependencies in text.
CCG vs. CFG

- CCG has some additional combinatory rules (e.g. type-raising and composition) motivated by the desire to recover unbounded dependencies in text.
Why Combinatory Categorial Grammar?

- Transparent interface between syntax and semantics
- Handles unbounded dependencies particularly well
- Lexicalised formalism leads to a surprisingly efficient parser
- There exists a CCG parser which is ideally suited to this workshop
- Other parsers based on similar formalisms can learn from our results
CCGbank

- Parser is based on CCGbank, a CCG version of the Penn Treebank (Hockenmaier and Steedman)
- CCGbank provides the wide-coverage grammar (essentially the lexicon, in addition to the small number of manually defined combinatory rules)
- CCGbank provides the training data for the statistical models
Parsing with CCG

- **Stage 1**
  - Assign lexical categories to words in the sentence
    - *supertagging* (almost parsing)
  - Use finite-state supertagger
    - based on Maximum Entropy tagging techniques
Parsing with CCG

- **Stage 1**
  - Assign lexical categories to words in the sentence
    - *supertagging* (almost parsing)
  - Use finite-state supertagger
    - based on Maximum Entropy tagging techniques

- **Stage 2a**
  - Combine the categories using the CCG combinatory rules
  - Use bottom-up CKY chart-parsing algorithm
  - Currently *no pruning* at this stage
Parsing with **CCG**

- **Stage 1**
  - Assign lexical categories to words in the sentence
    - *supertagging* (almost parsing)
  - Use finite-state supertagger
    - based on Maximum Entropy tagging techniques

- **Stage 2a**
  - Combine the categories using the **CCG** combinatory rules
  - Use bottom-up **CKY** chart-parsing algorithm
  - Currently *no pruning* at this stage

- **Stage 2b**
  - Find the highest scoring derivation according to some model
    - e.g. generative model, **CRF**, perceptron
  - Viterbi algorithm finds this efficiently
Multiple Output Formats

- CCG derivations (including N-best)
- CCG dependency structures (including weighted dependencies)
- Grammatical relations
- First-order logical forms
Speed Demo

• Use of supertagger and highly engineered C++ leads to a surprisingly fast parser
• We have used the parser to analyse 1 billion words of text in less than 5 days using only 18 CPUs
Accuracy

- Statistically no different to the Berkeley parser on a representative subset of the Penn Treebank standard test set
- State-of-the-art results on the DepBank grammatical relations test suite (83.4 F)
- Substantially outperforms Stanford, DCU, RASP on an unbounded dependency test suite
Adapting the Parser to Wikipedia Text

- Two approaches:
  - manually annotate and re-train at lower levels of representation
  - self-training on parsed Wikipedia data

- Create training and test data for Wikipedia:
  - 1,000 sentences of CCG pos-tagged and supertagged training data
  - 200 sentences of CCG dependencies for evaluation
  - 300 sentences of grammatical relations for evaluation

- People:
  - Stephen Clark, Ann Copestake
  - Laura Rimell (Cambridge associate)
  - Matthew Honnibal (Sydney associate)
Pre-Processing

• Important but neglected task in NLP pipelines
• Good pre-processor of HTML, plus tokenisation and sentence boundary detection, crucial for user-friendly and accurate web parser
• People:
  • Curt Van Wyk
  • Tim Dawborn (Sydney associate)
  • James Curran
Self-Training

- McClosky et al. apply self-training to a two-stage parsing system
- The CCG parser has two stages: supertagger and chart parser
- We will investigate the use of self-training to improve:
  - accuracy on Wikipedia data
  - efficiency: train the parser to select the lexical categories that the parser will eventually select anyway
  - advantage of self-training is the unlimited amount of data
- People:
  - Jessi Roesner
  - Jonathan Kummerfeld (Sydney associate)
  - Stephen Clark, James Curran
Chart Classification

- Various ways of pruning the chart:
  - predict in advance which cells are likely (or not) to be constituents
  - rerank the constituents in a cell and prune
  - use standard beam search

- *All of these models trained on parser output*

- People:
  - Byung-Gyu Ahn
  - Yue Zhang
  - Stephen Clark, James Curran
Coordination Disambiguation

- Parsers typically perform poorly on coordination constructions
- CCG offers elegant analyses of various coordination phenomena
- Apply coordination reranker to N-best parser output
- Knowledge sources: distributional semantic models induced from parsed data, google n-gram counts, ...
- People:
  - Aurelie Herbelot
  - Ann Copestake
Bootstrapping

- Various facts from Wikipedia can be used to obtain high precision training data:
  - e.g. *Mozart was born January 27, 1756*
  - short factual snippets can be reliably parsed
  - any other sentence containing any of these keywords highly likely to have same dependency relations

- Research questions:
  - how to identify the facts
  - how to train the parser model on partial data

- People:
  - James Haggerty
  - James Curran
Summary

- Parser of choice for the **NLP** community (especially for researchers doing large-scale web processing)
  - adaptable, efficient, accurate, easy-to-use, multiple outputs, ...
- Moving on from the standard Penn Treebank parsing task
  - larger scale, new domains, different representations