Reranking the Berkeley and Brown Parsers

Mark Johnson and Engin Ural
Brown University
Macquarie University

June 2010
The Brown and the Berkeley parsers

- Both state-of-the-art, PCFG-based, generative parsers
- **Brown parser:**
  - conditions on a wide variety of manually-chosen information
  - simple training procedure, hand-designed smoothing
- **Berkeley parser:**
  - split-merge procedure learns refined non-terminals
  - complex but fully automatic training procedure

⇒ The parsers are *very different from each other*

Reranking the $n$-best parser output

- Reranking rescores the $n$-best trees produced by a parser
  - incorporates features difficult to use in generative models
  - discriminatively trained MaxEnt model with L2 regularisation
- Research questions:
  - will reranking work with the Berkeley parser?
  - if it does work, will the same features be most useful?
  - can we rerank the combined $n$-best trees of both the Brown and Berkeley parsers?
- Relevant previous work: Zhang et al (2009)
  - also combine $n$-best lists from Brown and Berkeley parsers
  - only use a small set of reranker features
  - their results are consistent with results reported here
  - also describe experiments using self-trained reranking parser

Experimental setup

- Brown parser run “out of the box”
- Berkeley trained with 6 splits, parsing in “accurate” mode
- Reranker training data consisted of PTB sections 2–21
  - 50-best parses produced using 20-fold cross-validation procedure
- Sections 22, 23 and 24 parsed using “out of the box” 50-best parser
- In order to avoid overtraining on section 23:
  - Folds 1–18 used as main training data
  - Folds 19 and 20 used as development data
  - PTB section 22 used as test data

See: Collins and Koo (2005)
Reranker features

- “Standard” features come “out of the box” with reranker
  - are probably tuned to Brown parser
- “Extended” features include more features that might help Berkeley parser
  - e.g., features that include heads, governors, head-to-head dependencies, etc.

<table>
<thead>
<tr>
<th></th>
<th>Standard</th>
<th>Extended</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of feature super-classes</td>
<td>14</td>
<td>20</td>
</tr>
<tr>
<td>Number of feature classes</td>
<td>90</td>
<td>162</td>
</tr>
<tr>
<td>Number of features</td>
<td>1,333,950</td>
<td>4,256,553</td>
</tr>
</tbody>
</table>
Super-classes in extended feature set (1)

**Parser:** an indicator feature indicating which parsers generated this parse,

**RelLogP:** the log probability of this parse according to each parser,

**InterpLogCondP:** an indicator feature based on the binned log conditional probability according to each parser,

**RightBranch:** an indicator function of each node that lies on the right-most branch of the parse tree,

**Heavy:** an indicator function based on the size and location of each nonterminal (designed to identify the locations of “heavy” phrases),

**LeftBranchLength:** an indicator function of the binned length of each left-branching chain,

**RightBranchLength:** an indicator function of the binned length of each right-branching chain,
Super-classes in extended feature set (2)

**Rule:** an indicator function of parent and children categories, optionally with head POS annotations,

**NNGram:** and indicator function of parent and $n$-gram sequences of children categories, optionally head annotated, inspired by the $n$-gram rule features described by Collins and Koo

**Heads:** an indicator function of “head-to-head” dependencies,

**SynSemHeads:** an indicator function of the pair of syntactic (i.e., functional) and semantic (i.e., lexical) heads of each non-terminal,

**RBContext:** an indicator function of how much each subtree deviates from right-branching,

**SubjVerbAgr:** an indicator function of whether subject-verb agreement is violated,
Super-classes in extended feature set (3)

CoPar: an indicator function that fires when conjoined phrases in a coordinate structure have approximately parallel syntactic structure,

CoLenPar: an indicator function that fires when conjoined phrases in a coordinate structure have approximately the same length,

Word: an indicator function that identifies words and their preterminals,

WProj: an indicator function that identifies words and their phrasal projections up to their maximal projection,
Super-classes in extended feature set (4)

**WEdges:** an indicator function that identifies the words and POS tags appearing at the edges of each nonterminal,

**NGramTree:** an indicator function of the subtree consisting of nodes connecting each pair of adjacent words in the parse tree, and

**HeadTree:** a tree fragment consisting of a head word and its projection up to its maximal projection, plus all of the siblings of each node in this sequence (this is like an auxiliary tree in a TAG).
## Parsing accuracy (f-score) on section 22

<table>
<thead>
<tr>
<th></th>
<th>No reranker</th>
<th>Reranker features</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>standard</td>
</tr>
<tr>
<td>Berkeley trees</td>
<td>89.5</td>
<td>91.6</td>
</tr>
<tr>
<td>Brown trees</td>
<td>89.5</td>
<td><strong>91.8</strong></td>
</tr>
<tr>
<td>Combined trees</td>
<td></td>
<td><strong>91.8</strong></td>
</tr>
</tbody>
</table>

- Feature weights estimated by minimising EM-based log-loss with L2 regularisation using L-BFGS

Oracle f-score on section 22

![Graph showing the oracle f-score for Berkeley and Brown parsers across different beam widths. The x-axis represents the n-best beam width, ranging from 0 to 50, and the y-axis represents the oracle f-score, ranging from 0.88 to 1.0. The graph compares the performance of the Berkeley parser (red line) and the Brown parser (green dotted line), highlighting their convergence as the beam width increases.](image)
Feature super-class ablation experiment

- **Average f-score change** on folds 19–20 and section 22
- Rerankers used *extended feature set* trained with *averaged perceptron algorithm*
  - **91.2%** f-scores on both *Berkeley and Brown trees*, and
  - **91.6%** f-scores on *combined trees*. 
Conclusions from feature super-class ablation experiment

- Linguistically-informed features (e.g., Heads, SynSemHeads, HeadTree) are more important when reranking combined trees than single parser output
  - perhaps log prob scores from individual parsers are effective when used on their own trees, but need recalibration on combined trees?
- Log prob scores from parsers also supply important information
- *Edge features* are particularly useful for Berkeley parser

Conclusions

• Reranker on section 23 combined trees achieves 91.49% f-score
  ▶ only 0.1% higher than standard reranker on Brown trees

• Reranking the output of the Berkeley parser or a combination of Berkeley and Brown trees is not significantly more accurate than reranking the Brown trees alone, even with the extended feature set
  ▶ perhaps the reranker features are still too oriented around Brown trees?

• There is still room for improvement in parsing!

See: Huang (2008)
Interested in parsing?

Macquarie University (Sydney, Australia) is recruiting PhD students and post-docs.

Contact Mark.Johnson@mq.edu.au for more information.