CCG Supertagging with a Recurrent Neural Network

Wenduan Xu 1  Michael Auli 2  Stephen Clark 1

1 Cambridge University
2 Facebook AI Research

July, 2015
Combinatory Categorial Grammar (CCG)

- A lexicalized formalism [Steedman, 2000]
Combinatory Categorial Grammar (CCG)

\[
\begin{array}{c}
\text{the} \quad \text{books} \quad \text{which} \quad \text{John} \quad \text{likes} \\
NP/N \quad N \quad (NP/NP)/(S/NP) \quad NP \quad (S/NP)/NP \\
\text{NP} \quad > \\
\frac{S/(S/NP)}{S/NP} \quad > \quad B \\
\frac{NP\backslash NP}{NP\backslash NP} \quad < \\
\end{array}
\]
Combinatory Categorial Grammar (CCG)

- A lexicalized formalism [Steedman, 2000]

[Clark & Curran] compared to about 50 POS tags in PTB

[Petkov and Klein, 2007] only a dozen combinatory rules compared to over 500K for a PTB parser
Combinatory Categorial Grammar (CCG)

- A lexicalized formalism [Steedman, 2000]
- 425 lexical categories in the standard models [Clark & Curran]
Combinatory Categorial Grammar (CCG)

- A lexicalized formalism [Steedman, 2000]
- 425 lexical categories in the standard models [Clark & Curran]
  - compared to about 50 POS tags in PTB
Combinatory Categorial Grammar (CCG)

- A lexicalized formalism [Steedman, 2000]
- 425 lexical categories in the standard models [Clark & Curran]
  - compared to about 50 POS tags in PTB
- Only a dozen combinatory rules
Combinatory Categorial Grammar (CCG)

- A lexicalized formalism [Steedman, 2000]

- 425 lexical categories in the standard models [Clark & Curran]
  - compared to about 50 POS tags in PTB

- Only a dozen combinatory rules
  - compared to over 500K for a PTB parser [Petrov and Klein, 2007]
CCG Parsing

- Method 1: leave all lexical ambiguity to the parser
  - was done in the first CCG parser [Hockenmaier, 2003]
CCG Parsing

- Method 1: leave all lexical ambiguity to the parser
  - was done in the first CCG parser [Hockenmaier, 2003]

- Method 2: use only 1-best supertags
  - accuracy not good enough (C&C supertagger accuracy is about 92%)
CCG Parsing

- Method 1: leave all lexical ambiguity to the parser
  - was done in the first CCG parser [Hockenmaier, 2003]

- Method 2: use only 1-best supertags
  - accuracy not good enough (C&C supertagger accuracy is about 92%)

- Current optimal solution, two stage process [Clark & Curran, 2007]:
  - multi-supertagging + parsing
CCG Parsing

- Method 1: leave all lexical ambiguity to the parser
  - was done in the first CCG parser [Hockenmaier, 2003]

- Method 2: use only 1-best supertags
  - accuracy not good enough (C&C supertagger accuracy is about 92%)

- Current optimal solution, two stage process [Clark & Curran, 2007]:
  - multi-supertagging + parsing

- In [Auli and Lopez, 2011]
  - called integrated supertagging and parsing
  - still, the same supertagging model and two-stage process as in C&C
Adaptive Supertagging [Clark & Curran, 2007]

Start with an initial prob. cutoff $\beta$

<table>
<thead>
<tr>
<th>He</th>
<th>reads</th>
<th>the</th>
<th>book</th>
</tr>
</thead>
<tbody>
<tr>
<td>$NP$</td>
<td>$(S[pss] \backslash NP) / NP$</td>
<td>$NP / N$</td>
<td>$N$</td>
</tr>
</tbody>
</table>
Adaptive Supertagging [Clark & Curran, 2007]

Prune a category, if its probability is below $\beta$ times the prob. of the best category

$$\begin{align*}
\text{He} & \quad \text{reads} & \quad \text{the} & \quad \text{book} \\
NP & \quad (S[pss]\NP)/NP & \quad NP/N & \quad N
\end{align*}$$
Decrease $\beta$ if no spanning analysis
Adaptive Supertagging [Clark & Curran, 2007]

Decrease $\beta$ if no spanning analysis

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>He</td>
<td>reads</td>
<td>the</td>
<td>book</td>
<td></td>
</tr>
<tr>
<td>(NP)</td>
<td>((S[pss]\ NP) / NP)</td>
<td>(NP / N)</td>
<td>(N)</td>
<td></td>
</tr>
<tr>
<td>(N)</td>
<td>((S \ NP) / NP)</td>
<td>(NP / NP)</td>
<td>((S \ NP) / NP)</td>
<td></td>
</tr>
<tr>
<td>(N / N)</td>
<td>(S \ NP)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(NP / NP)</td>
<td>((S[pt] \ NP) / NP)</td>
<td>((S[dcl] \ NP) / NP)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
• Affects search for training and testing [Clark & Curran, 2007]

• Also affects parsing efficiency [Curran et al., 2006]

• The standard MaxEnt model
  – relies heavily on POS tags
  – uses only sparse indicator features
  – considers only local contexts

• The recent feed-forward neural supertagger [Lewis & Steedman, 2014]
  – no POS tag features
  – all dense features
  – still considers only local contexts
Supertagging with a RNN

- Using only dense features
  - word embedding
  - suffix embedding
  - capitalization

- The input layer is a concatenation of all embeddings of all words in a context window
Supertagging with a RNN

... bought some books and ...

...
Supertagging with a RNN

... bought some books and ...
Supertagging with a RNN

... bought some books and ...
Supertagging with a RNN

... bought some books and ...
Supertagging with a RNN

... bought some books and ...

...
Training & Experiments

- Mini-batched BPTT [Rumelhart et al., 1988; Mikolov, 2012]
- A context window-size of 7, a BPTT step size of 9
- 50-dim scaled Turian embeddings [Turian et al., 2010]
- Other two look-up tables randomly initialized
- Embedding fine-tuning during training
- Dropout regularization
- Parsing experiments: use the same supertagger prob. cutoff values as C&C
## 1-best Supertagging Results: dev

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>C&amp;C (gold POS)</td>
<td>92.60</td>
<td>-</td>
</tr>
<tr>
<td>C&amp;C (auto POS)</td>
<td>91.50</td>
<td>0.57</td>
</tr>
<tr>
<td>NN</td>
<td>91.10</td>
<td>21.00</td>
</tr>
<tr>
<td>RNN</td>
<td>92.63</td>
<td>-</td>
</tr>
<tr>
<td>RNN+dropout</td>
<td>93.07</td>
<td>2.02</td>
</tr>
</tbody>
</table>

Table 1: 1-best tagging accuracy and speed comparison on CCGBank Section 00 with a single CPU core (1,913 sentences), tagging time in secs.
Table 2: 1-best tagging accuracy comparison on CCGBank Section 23 (2,407 sentences), Wikipedia (200 sentences) and Bio-GENIA (1,000 sentences).
Multi-tagging Results: dev
Multi-tagging Results: test
### Final Parsing Results

<table>
<thead>
<tr>
<th></th>
<th>LP</th>
<th>LR</th>
<th>LF</th>
<th>cov.</th>
<th></th>
<th>LP</th>
<th>LR</th>
<th>LF</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CCGBank Section 23</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>Wikipedia</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C&amp;C</td>
<td>86.24</td>
<td>84.85</td>
<td>85.54</td>
<td>99.42</td>
<td>81.58</td>
<td>80.08</td>
<td>80.83</td>
<td>99.50</td>
<td></td>
</tr>
<tr>
<td>(NN)</td>
<td>86.71</td>
<td>85.56</td>
<td>86.13</td>
<td>99.92</td>
<td>82.65</td>
<td>81.36</td>
<td>82.00</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>(RNN)</td>
<td><strong>87.68</strong></td>
<td><strong>86.47</strong></td>
<td><strong>87.07</strong></td>
<td><strong>99.96</strong></td>
<td><strong>83.22</strong></td>
<td><strong>81.78</strong></td>
<td><strong>82.49</strong></td>
<td><strong>100</strong></td>
<td></td>
</tr>
<tr>
<td>C&amp;C</td>
<td>86.24</td>
<td>84.17</td>
<td>85.19</td>
<td>100</td>
<td>81.58</td>
<td>79.48</td>
<td>80.52</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>(NN)</td>
<td>86.71</td>
<td>85.40</td>
<td>86.05</td>
<td>100</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>(RNN)</td>
<td><strong>87.68</strong></td>
<td><strong>86.41</strong></td>
<td><strong>87.04</strong></td>
<td><strong>100</strong></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Parsing test results (auto POS). We evaluate on all sentences (100% coverage) as well as on only those sentences that returned spanning analyses (% cov.). RNN and NN both have 100% coverage on the Wikipedia data.
The End

Thank You!