State-of-the-art Dependency Parsing

State-of-the-art Parsers

- Unlabeled attachment score: fraction of words with correct parent
- Labeled attachment score: have to label each edge correctly (but this isn’t that hard — noun before verb -> nsubj in most contexts)
- 2005: Eisner algorithm graph-based parser was SOTA (~91 UAS)
- 2010: Better graph-based parsers using “parent annotation” (~93 UAS)
- 2012: Transition-based Maltparser achieved good results (~90 UAS)
- 2014: Stanford neural dependency parser (Chen and Manning) got 92 UAS with transition-based neural model
- 2016: Improvements to Chen and Manning

Stanford Dependency Parser

- Feedforward neural network on top of feature vector extracted from stack and buffer

  1st in stack  2nd in stack  1st in buf  ...  POS of leftmost child of 1st in stack  ...

Chen and Manning (2014)

Stanford Dependency Parser

- Softmax layer:
  \[ p = \text{softmax}(W_2h) \]
- Hidden layer:
  \[ h = (W_1^1x^w + W_1^2x^x + W_1^3x^l + b_1)^3 \]
- Input layer: \( [x^w, x^x, x^l] \)

Chen and Manning (2014)
### Stanford Dependency Parser

<table>
<thead>
<tr>
<th>Parser</th>
<th>Dev UAS</th>
<th>Dev LAS</th>
<th>Test UAS</th>
<th>Test LAS</th>
<th>Speed (sent/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>standard</td>
<td>90.2</td>
<td>87.8</td>
<td>89.4</td>
<td>87.3</td>
<td>26</td>
</tr>
<tr>
<td>eager</td>
<td>89.8</td>
<td>87.4</td>
<td>89.6</td>
<td>87.4</td>
<td>34</td>
</tr>
<tr>
<td>Malt:sp</td>
<td>89.8</td>
<td>87.2</td>
<td>89.3</td>
<td>86.9</td>
<td>469</td>
</tr>
<tr>
<td>Malt:eager</td>
<td>89.6</td>
<td>86.9</td>
<td>89.4</td>
<td>86.8</td>
<td>448</td>
</tr>
<tr>
<td>MSTParser</td>
<td>91.4</td>
<td>88.1</td>
<td>90.7</td>
<td>87.6</td>
<td>10</td>
</tr>
<tr>
<td>Our parser</td>
<td><strong>92.0</strong></td>
<td><strong>89.7</strong></td>
<td><strong>91.8</strong></td>
<td><strong>89.6</strong></td>
<td><strong>654</strong></td>
</tr>
</tbody>
</table>

- MSTParser: “graph-based” parser (like CKY) from 2005 — so Chen+Manning’s parser isn’t much better but is much faster!

Chen and Manning (2014)

### Parsey McParseFace (a.k.a. SyntaxNet)

- Close to state-of-the-art, released by Google publicly
- 94.61 UAS on the Penn Treebank using a transition-based system
- Additional data harvested via “tri-training”, form of self-training
- Same feature set as Chen and Manning (2014), Google fine-tuned it

https://github.com/tensorflow/models/tree/master/research/syntaxnet

Andor et al. (2016)

### Other languages

- Annotate dependencies with the same representation in many languages

http://universaldependencies.org/