Parsing with PCFGs

- As with HMMs, Viterbi-style search can be used to find the highest probability tree for a given string

- Computing probabilities bottom up (inside-style) gives us a probabilistic CYK parser

- Computing probabilities top down (outside-style) gives us a probabilistic Earley parser (Stolke 1995)

- Probabilistic non-deterministic (best first) parsers also work quite well
Estimating PCFGs

- Given a treebank (a corpus of parsed sentences), estimating the parameters of PCFG is easy:

\[ P(N^i \rightarrow \zeta | N^i) = \frac{C(N^i \rightarrow \zeta)}{C(N^i)} \]

- If we have a CFG but do not have a parsed corpus, we can use a version of the EM algorithm (the Inside-Outside algorithm)

- Inside-Outside estimation suffers from the same problems as Forward-Backward estimation, only worse

- The grammar induction problem, constructing a PCFG directly from an unannotated corpus, is an active research area
Limitations of PCFGs

- Ordinary PCFGs don’t work very well
- Position-independence assumption isn’t true:

<table>
<thead>
<tr>
<th></th>
<th>Pronoun</th>
<th>Lexical</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject</td>
<td>91%</td>
<td>9%</td>
</tr>
<tr>
<td>Object</td>
<td>34%</td>
<td>66%</td>
</tr>
</tbody>
</table>

- PCFGs fail to capture lexical relationships:
  - subcategorization frames
  - attachment ambiguities: *The astronomer saw the moon with a telescope.*
  - coordination: *dogs in houses and cats*

- Various alternate models in §12.2
Lexicalization

- One strategy for improving PCFGs is to augment non-terminals with their head word:

- Captures subcat preferences and lexical dependencies

- Lexicalization is expensive (e.g., parsing goes from $O(n^3)$ to $O(n^5)$)
Lexicalization

- The effect of this is to condition the probability of a rule application on the head-word of the phrase:

  \[ P(N^i \rightarrow \zeta^j | N^i, \text{Head}(N^i) = w^k) \]

- We can then model the probability of a phrase having a particular head:

  \[ P(\text{Head}(N^i) = w^k | N^i, \text{Head}(\text{Mother}(N^i)) = w^m) \]

- Multiplying head/rule probabilities and head/head probabilities gives us the tree probability.
Other improvements

- Parent annotation adds information about external context:

\[
S \rightarrow NP^S \ VP^S \\
VP^S \rightarrow V^VP \ NP^VP
\]

- Subjects are \(NP^S\), objects are \(NP^VP\)

- This has the effect of weakening the *history-free* assumption, and greatly improves accuracy of treebank grammars (Johnson 1998)
Other improvements

- Treebank grammars have lots of very specific rules:

  \[ VP \rightarrow VBZ \ NP \ PP \ PP \]

- We can reduce sparse data problems via head-driven Markovization (Collins 1999):
Other improvements

- Klein and Manning (2003) combine horizontal and vertical markovization

- Standard PCFGs correspond to $v = 1$ and $h = \infty$

- Setting $v = 3$ and $h \leq 2$ improves F score from 72.62 to 79.74

- The same strategies can be used to add appropriate internal and external annotation to non-terminals

- Annotating nodes weakens independence assumptions without changing the way probabilities are calculated
<table>
<thead>
<tr>
<th>Feature</th>
<th>$F$</th>
<th>change</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline ($v \leq 2, h \leq 2$)</td>
<td>77.77</td>
<td>—</td>
</tr>
<tr>
<td>UNARY-INTERNAL</td>
<td>78.32</td>
<td>0.55</td>
</tr>
<tr>
<td>UNARY-DT</td>
<td>78.48</td>
<td>0.17</td>
</tr>
<tr>
<td>UNARY-RB</td>
<td>78.86</td>
<td>0.43</td>
</tr>
<tr>
<td>TAG-PA</td>
<td>80.62</td>
<td>2.52</td>
</tr>
<tr>
<td>SPLIT-IN</td>
<td>81.19</td>
<td>2.12</td>
</tr>
<tr>
<td>SPLIT-AUX</td>
<td>81.66</td>
<td>0.57</td>
</tr>
<tr>
<td>SPLIT-CC</td>
<td>81.69</td>
<td>0.12</td>
</tr>
<tr>
<td>SPLIT-%</td>
<td>81.81</td>
<td>0.15</td>
</tr>
<tr>
<td>TMP-NP</td>
<td>82.25</td>
<td>1.07</td>
</tr>
<tr>
<td>GAPPED-S</td>
<td>82.28</td>
<td>0.17</td>
</tr>
<tr>
<td>POSS-NP</td>
<td>83.06</td>
<td>0.28</td>
</tr>
<tr>
<td>SPLIT-VP</td>
<td>85.72</td>
<td>1.36</td>
</tr>
<tr>
<td>BASE-NP</td>
<td>86.04</td>
<td>0.73</td>
</tr>
<tr>
<td>DOMINATES-V</td>
<td>86.91</td>
<td>1.42</td>
</tr>
<tr>
<td>RIGHT-REC-NP</td>
<td>87.04</td>
<td>1.94</td>
</tr>
</tbody>
</table>
Probabilistic parsing

- Parsers are (often) evaluated by labeled precision and recall (perhaps dependency precision and recall would more useful)

- Klein and Manning's unlexicalized parser yields $F = 87.04$

- The very best lexicalized PCFG parsers get $F \approx 89$, and don’t seem to be improving much

- Future directions:
  - Better models (MaxEnt)
  - More sophisticated grammar formalisms (LTAGs, DCGs, HPSGs)
  - Richer representations (grammatical relations, semantics)