Accurate Unlexicalized Parsing
by Dan Klein and Christopher D. Manning (ACL 2003)

Presented by Ulrich Germann
Background

*Naïve PCFGs tend to perform poorly, because their assumptions of context-freeness are too strong.*


<table>
<thead>
<tr>
<th>Performance on sentences of up to 40 words</th>
</tr>
</thead>
<tbody>
<tr>
<td>system</td>
</tr>
<tr>
<td>baseline naïve PCFG</td>
</tr>
<tr>
<td>Magerman (1995)</td>
</tr>
<tr>
<td>Collins (1996)</td>
</tr>
<tr>
<td>Charniak (1997)</td>
</tr>
<tr>
<td>Collins (1999)</td>
</tr>
<tr>
<td>Charniak (2001)</td>
</tr>
</tbody>
</table>
Beyond and Apart from Lexicalization

- Johnson (1998): Annotating each node with the category of its parent category boosts performance from 73.5% to 80.0% on sequences of POS tags.
- Charniak (2001) also considers parent annotation in a ME framework.
- Gildea (2001) shows that removing *bilexical* probabilities from Collins’s model 1 has only a very small negative effect on parsing quality.
Collins’s Model 1: 

\[ P(w_i, T_i, t_i | T_p, T_h, w_h, t_h, \Delta) = P(w_i | T_i, t_i, T_p, T_h, w_h, t_h, \Delta) \times P(T_i, t_i | T_p, T_h, w_h, t_h, \Delta) \]
Daniel Gildea’s Experiment (cont’d)

\[ P(w_i|T_i, t_i, T_p, T_h, w_h, t_h, \Delta) \approx \]

\[ \lambda_1 \bar{P}(w_i|T_i, t_i, T_p, T_h, w_h, t_h, \Delta) \]

\[ + (1 - \lambda_1) \left( \lambda_2 \bar{P}(w_i|T_i, t_i, T_p, T_h, t_h, \Delta) + (1 - \lambda_2) \bar{P}(w_i|t_i) \right) \]
Daniel Gildea’s Experiment (cont’d)

\[
P(w_i|T_i, t_i, T_p, T_h, w_h, t_h, \Delta) \approx \lambda_1 \tilde{P}(w_i|T_i, t_i, T_p, T_h, w_h, t_h, \Delta) + \left(1 - \lambda_1\right) \left(\lambda_2 \tilde{P}(w_i|T_i, t_i, T_p, T_h, t_h, \Delta) + (1 - \lambda_2) \tilde{P}(w_i|t_i)\right)
\]

<table>
<thead>
<tr>
<th>training set</th>
<th>test set</th>
<th>recall</th>
<th>prec.</th>
<th>recall</th>
<th>prec.</th>
</tr>
</thead>
<tbody>
<tr>
<td>WSJ</td>
<td>WJS</td>
<td>86.1</td>
<td>86.6</td>
<td>85.6</td>
<td>86.2</td>
</tr>
<tr>
<td>WSJ</td>
<td>Brown</td>
<td>80.3</td>
<td>81.0</td>
<td>80.3</td>
<td>81.0</td>
</tr>
<tr>
<td>Brown</td>
<td>Brown</td>
<td>83.6</td>
<td>84.6</td>
<td>83.5</td>
<td>84.4</td>
</tr>
<tr>
<td>WSJ+Brown</td>
<td>Brown</td>
<td>83.9</td>
<td>84.8</td>
<td>83.4</td>
<td>84.3</td>
</tr>
<tr>
<td>WSJ+Brown</td>
<td>WSJ</td>
<td>86.3</td>
<td>86.9</td>
<td>85.7</td>
<td>86.4</td>
</tr>
</tbody>
</table>

WSJ: \(\sim\) 40k sentences/950k words; Brown: \(\sim\) 22k sentences/413k words
What the Paper is About ...

How far can we get without lexicalization?

Why bother?

- improved baseline for unlexicalized probabilistic parsing
- insights
- smaller grammars that are easier to reason about
- faster parsing $O(n^3)$ with lower grammar constant
What’s Wrong with Naïve PCFGs?

- Category symbols are too coarse; the probability distribution within the categories is not accounted for well.

  **Example:** A subject-NP is 8.7 times more likely than an object-NP to expand just as a pronoun.

- Training data is too sparse for accurate occurrence counts of rare rules.
  - probability of seen rare events is overestimated
  - probability of unseen rare events is underestimated
Klein & Manning’s Approach

- Vertical and horizontal “Markovization” of probabilistic estimates.
- Additional annotation of tags with information available from the trees.
- Linguistically (and empirically) motivated splitting of POS-level categories into subcategories.
- Selective splitting of categories based on information obtainable from the trees in the treebank.
- Expressly no smoothing except for POS tagging.
Except for the root node, every node in a parse tree has

- a **vertical** history/context (parent, grandparent, etc.)
- a **horizontal** history/context

Traditional PCFGs use the full horizontal context and a
vertical context of 1.
Horizontal Markovization

Also used by Collins (1997, 1999).

Always takes the head into account (not by definition, but as used by Collins and K&M).

Markov assumption:

\[ P(L_i | P, H, L_1, \ldots, L_{i-n+1}, \ldots, L_{i-1}) = P(L_i | P, H, L_{i-n+1}, \ldots, L_{i-1}) \]
\[ P(R_i | P, H, R_1, \ldots, R_{i-n+1}, \ldots, R_{i-1}) = P(R_i | P, H, R_{i-n+1}, \ldots, R_{i-1}) \]

Amounts to tree binarization:

\[ VP \rightarrow VBZ \, NP \, PP \, PP \]
\[ \Rightarrow \langle VP:[VBZ]\rangle \rightarrow VBZ \]
\[ \langle VP:[VBZ] \ldots NP \rangle \rightarrow \langle VP:[VBZ]\rangle \, NP \]
\[ \langle VP:[VBZ] \ldots PP \rangle \rightarrow \langle VP:[VBZ] \ldots NP \rangle \, PP \]
Vertical Markovization

generalization of parent annotation

\[
\begin{align*}
S & \rightarrow \text{NP VP} & S & \rightarrow \text{NP}^S \text{ VP}^S \\
\text{NP} & \rightarrow \text{NN} & \Rightarrow & \text{NP}^S & \rightarrow \text{NN} \\
\text{VP} & \rightarrow \text{VBZ NP} & \text{VP}^S & \rightarrow \text{VBZ NP}^\text{VP} \\
\end{align*}
\]

\[\cdots\]

On a marginal note: K&M treat POS tags as terminals and discuss parent-annotation of POS-tags separately.
## Markovization: Results

<table>
<thead>
<tr>
<th>Vertical Order</th>
<th>Horizontal Markov Order</th>
<th>h = 0</th>
<th>h = 1</th>
<th>h ≤ 2</th>
<th>h = 2</th>
<th>h = ∞</th>
</tr>
</thead>
<tbody>
<tr>
<td>v = 1 No annotation</td>
<td></td>
<td>71.27</td>
<td>72.5</td>
<td>73.46</td>
<td>72.96</td>
<td>72.62</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(854)</td>
<td>(3119)</td>
<td>(3863)</td>
<td>(6207)</td>
<td>(9657)</td>
</tr>
<tr>
<td>v ≤ 2 Sel. Parents</td>
<td></td>
<td>74.75</td>
<td>77.42</td>
<td>77.77</td>
<td>77.50</td>
<td>76.91</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(2285)</td>
<td>(6564)</td>
<td>(7619)</td>
<td>(11398)</td>
<td>(14247)</td>
</tr>
<tr>
<td>v = 2 All Parents</td>
<td></td>
<td>74.68</td>
<td>77.42</td>
<td>77.81</td>
<td>77.50</td>
<td>76.81</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(2984)</td>
<td>(7312)</td>
<td>(8367)</td>
<td>(12132)</td>
<td>(14666)</td>
</tr>
<tr>
<td>v ≤ 3 Sel. GParents</td>
<td></td>
<td>76.50</td>
<td>78.59</td>
<td>79.07</td>
<td>78.97</td>
<td>78.54</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(4943)</td>
<td>(12374)</td>
<td>(13627)</td>
<td>(19545)</td>
<td>(20123)</td>
</tr>
<tr>
<td>v = 3 All GParents</td>
<td></td>
<td>76.74</td>
<td>79.18</td>
<td>79.74</td>
<td>79.07</td>
<td>78.72</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(7797)</td>
<td>(15740)</td>
<td>(16994)</td>
<td>(22886)</td>
<td>(22002)</td>
</tr>
</tbody>
</table>

Figure 2: Markovizations: $F_1$ and grammar size.

Figure from Klein & Manning (2003)
^
U (external unary) “I am the only child.”

-U (internal unary) “I have only one child.”

- Roughly the same performance in isolation; in combination with other features “internal unary” is better.

- On the preterminal level (POS → word), external unary mark-up helps with
  - demonstratives (that, this) vs. articles (a, the) — both labeled as DT in Penn TreeBank
  - adverbs (e.g., also vs. as well).

- “Beyond these cases, unary tag marking was detrimental.”
Figure 4: An error which can be resolved with the UNARY-INTERNAL annotation (incorrect baseline parse shown).
Tag Splitting

- Parent annotation also for preterminal tags.
- Splitting of IN tags into 6 linguistically motivated groups (prepositions vs. conjunctions vs. complementizers; noun-modifying vs. primarily verb-modifying prepositions (of vs. as)).
- Distinction between auxiliaries have and be.
- Special conjunction class containing but/But and &.
- % gets its own tag.
Benefits of TAG-PA/SPLIT-IN

Figure 5: An error resolved with the TAG-PA annotation (of the IN tag): (a) the incorrect baseline parse and (b) the correct TAG-PA parse. SPLIT-IN also resolves this error.
Annotations already in the treebank

- generally hurt, with two exceptions
  - mark-up of temporal NPs (NP-TMP)
  - mark-up of sentences with a gap (GAPPED-S)

Figure 6: An error resolved with the TMP-NP annotation: (a) the incorrect baseline parse and (b) the correct TMP-NP parse.

Figure from Klein & Manning (2003)
Head Annotation

- propagates information from the head to the parent

2 mark-ups found particularly useful:
- Mark-up of possessive NPs (POSS-NP).
- Distinction between finite and non-finite VPs (SPLIT-VP).
Tackling Attachment Ambiguities

Three features found useful:

- mark-up of plain base NPs (NP → NN)
- mark-up of nodes that dominate a verb
- mark-up of NPs that contain another NP in their right periphery
## Results

<table>
<thead>
<tr>
<th>Annotation</th>
<th>Size</th>
<th>$F_1$</th>
<th>$\Delta F_1$</th>
<th>$\Delta F_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline ($v \leq 2, h \leq 2$)</td>
<td>7619</td>
<td>77.77</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>UNARY-INTERNAL</td>
<td>8065</td>
<td>78.32</td>
<td>0.55</td>
<td>0.55</td>
</tr>
<tr>
<td>UNARY-DT</td>
<td>8066</td>
<td>78.48</td>
<td>0.71</td>
<td>0.17</td>
</tr>
<tr>
<td>UNARY-RB</td>
<td>8069</td>
<td>78.86</td>
<td>1.09</td>
<td>0.43</td>
</tr>
<tr>
<td>TAG-PA</td>
<td>8520</td>
<td>80.62</td>
<td>2.85</td>
<td>2.52</td>
</tr>
<tr>
<td>SPLIT-IN</td>
<td>8541</td>
<td>81.19</td>
<td>3.42</td>
<td>2.12</td>
</tr>
<tr>
<td>SPLIT-AUX</td>
<td>9034</td>
<td>81.66</td>
<td>3.89</td>
<td>0.57</td>
</tr>
<tr>
<td>SPLIT-CC</td>
<td>9190</td>
<td>81.69</td>
<td>3.92</td>
<td>0.12</td>
</tr>
<tr>
<td>SPLIT-%</td>
<td>9255</td>
<td>81.81</td>
<td>4.04</td>
<td>0.15</td>
</tr>
<tr>
<td>TMP-NP</td>
<td>9594</td>
<td>82.25</td>
<td>4.48</td>
<td>1.07</td>
</tr>
<tr>
<td>GAPPED-S</td>
<td>9741</td>
<td>82.28</td>
<td>4.51</td>
<td>0.17</td>
</tr>
<tr>
<td>POSS-NP</td>
<td>9820</td>
<td>83.06</td>
<td>5.29</td>
<td>0.28</td>
</tr>
<tr>
<td>SPLIT-VP</td>
<td>10499</td>
<td>85.72</td>
<td>7.95</td>
<td>1.36</td>
</tr>
<tr>
<td>BASE-NP</td>
<td>11660</td>
<td>86.04</td>
<td>8.27</td>
<td>0.73</td>
</tr>
<tr>
<td>DOMINATES-V</td>
<td>14097</td>
<td>86.91</td>
<td>9.14</td>
<td>1.42</td>
</tr>
<tr>
<td>RIGHT-REC-NP</td>
<td>15276</td>
<td>87.04</td>
<td>9.27</td>
<td>1.94</td>
</tr>
</tbody>
</table>

Figure from Klein & Manning (2003)
Conclusions

- K&M significantly raise the baseline on unlexicalized parsing.
- Their work shows that one can recover from over-generalizations in the treebank ...
- ... and that it’s worth the effort.
- Better modeling is based on linguistic analysis.
- Raises some interesting questions ...
Questions

- What do the learning curves for unlexicalized vs. lexicalized parsing look like?
- How do the different parsers perform on out-of-domain data?
- What are the confidence intervals for the results?
- What do the parsers still struggle with? (According to Collins (2003), coordination structures are a big problem.)