Last Time

- Vanilla PCFGs
- Treebanks
- Parsing Algorithms for PCFGs
Today

• Some useful transformations on trees
• Modern parsing models:
  – Collins (1997; 2003)
  – Charniak (1997; 2000)
from Johnson (1998)
Parent Annotation

NP $\rightarrow^p$ NP PP

NP $\rightarrow^q$ NP PP PP

\[ ... \]

\[ \cdots \]

\[ p^2 \quad q \]

\[ ... \]

\[ \cdots \]

\[ \cdots \]

\[ \cdots \]

\[ \cdots \]

\[ \cdots \]
Parent Annotation

NP^{VP} \rightarrow^\rho NP^{NP} PP^{NP}

NP^{NP} \rightarrow^r NP^{NP} PP^{NP}

NP^{VP} \rightarrow^q NP^{NP} PP^{NP} PP^{NP}
Parent Annotation

• Another way to think about it …

Before: \[ p(\text{tree}) = \prod_{n \in \text{tree's nonterminal tokens}} \rho(\text{n's children}|n) \]

Now: \[ p(\text{tree}) = \prod_{n \in \text{tree's nonterminal tokens}} \rho(\text{n's children}|n, \text{n's parent}) \]

• This could conceivably help performance (weaker independence assumptions)

• This could conceivably hurt performance (data sparseness)
Parent Annotation

• From Johnson (1998):

PCFG from WSJ Treebank: 14,962 rules
  • Of those, 1,327 would *always* be subsumed!

After parent annotation: 22,773 rules
  • Only 965 would always be subsumed!

Recall 69.7% → 79.2%; precision 73.5% → 80.0%

• Trick: check for subsumed rules, remove them from the grammar → faster parsing.
“I love all my children, but one of them is special.”

S → NP VP

VP → VBD NP

NP → DT NNS PP

Heads not in the Treebank.

Usually people use deterministic head rules (Magerman, 1995).
Head Annotation

I hit the man with the bat
Lexicalization

• Every nonterminal node is annotated with a word from its yield; such that
  \[ \text{lex}(n) = \text{lex}(\text{head}(n)) \]
Lexical Head Annotation

S

hit

NP

I

V

‘hit

V

hit

NP

man

VP

hit

P

with

PP

with

NP

bat

NP

man

DT

the

N

man

P

with

NP

bat

DT

the

N

bat

VP

hit

V

hit

NP

man

DT

the

N

man

P

with

NP

bat

NP

man

DT

the

N

man

P

with

NP

bat

I

hit

the

man

with

the

bat
Lexicalization

• Every nonterminal node is annotated with a word from its yield; such that
  \[ \text{lex}(n) = \text{lex}(\text{head}(n)) \]

• What might this allow?
• What might we worry about?

Currently, this is controversial (we’ll see why)!
• Take away the nonlexical parts.
Dependencies

• Take away the nonlexical parts.
Dependencies

• Merge redundant nodes upward.
Crucial Point

• By “decorating” the treebank, we have been carrying additional information around the trees.

• The hope is to improve the ability of a PCFG to predict syntactic structure correctly.

• The worry is that our grammar will get really big and the probabilities too hard to estimate.
  – Also, speed. More rules $\rightarrow$ bigger grammar $\rightarrow$ slower parsing.
Dependencies

• Can represent some things that are hard for CFGs (but then it’s not a PCFG anymore):

We saw a house in June that we bought.
Dependencies

• Don’t have to be lexicalized
• Often faster to parse
• Closer to semantics?
• We’ll come back to this representation.
Collins Model 1 (1997)

- Trees are headed & lexicalized.
- Many, many rules!

\[
\begin{align*}
\text{VP}_{\text{saw}} & \to \underline{\text{V}}_{\text{saw}} \text{ NP}_{\text{man}} \text{ PP}_{\text{through}} \\
\text{VP}_{\text{saw}} & \to \underline{\text{V}}_{\text{saw}} \text{ NP}_{\text{man}} \text{ PP}_{\text{with}} \\
\text{VP}_{\text{saw}} & \to \underline{\text{V}}_{\text{saw}} \text{ NP}_{\text{woman}} \text{ PP}_{\text{through}} \\
\text{VP}_{\text{saw}} & \to \underline{\text{V}}_{\text{saw}} \text{ NP}_{\text{man}} \\
\end{align*}
\]
Collins Model 1 (1997)

• We are given the parent and its lexeme.
Collins Model 1 (1997)

- We are given the parent and its lexeme.
- Randomly generate the head nonterminal.
Collins Model 1 (1997)

- We are given the parent and its lexeme.
- Randomly generate the head nonterminal.
- Generate a sequence of left children.
Collins Model 1 (1997)

- We are given the parent and its lexeme.
- Randomly generate the head nonterminal.
- Generate a sequence of left children.
Collins Model 1 (1997)

- We are given the parent and its lexeme.
- Randomly generate the head nonterminal.
- Generate a sequence of left children.
- Generate a sequence of right children.
• We are given the parent and its lexeme.
• Randomly generate the head nonterminal.
• Generate a sequence of left children.
• Generate a sequence of right children.

Collins Model 1 (1997)
Collins Model 1 (1997)

- We are given the parent and its lexeme.
- Randomly generate the head nonterminal.
- Generate a sequence of left children.
- Generate a sequence of right children.
Collins Model 1 (1997)

- Wanted to model **distance**. How?
- Assume depth-first recursion.
Collins Model 1 (1997)

- Wanted to model **distance**. How?
- Assume depth-first recursion.
- Can then condition the next child on (features of) the yield between it and the head:

\[
p(PP_{\text{with}} \mid VP_{\text{saw}}, \text{right, “the cat who liked milk”}) \\
\approx p(PP_{\text{with}} \mid VP_{\text{saw}}, \text{right, length}>0, +\text{verb})
\]

- 1997 version looked for commas, too; later this was removed.
Collins Model 1 (1997)

\[ p\left(\langle L,u \rangle_1^n, \langle H,w \rangle, \langle R,v \rangle_1^m | \langle P,w \rangle \right) = \]

\[ p(H | \langle P,w \rangle) \cdot \left( \prod_{i=1}^{n} p(\langle L,u \rangle_i | \langle P,w \rangle, H, \text{left}, \Delta_i) \right) p(\text{stop} | \langle P,w \rangle, H, \text{left}, \Delta_{n+1}) \]

\[ \cdot \left( \prod_{i=1}^{m} p(\langle R,v \rangle_i | \langle P,w \rangle, H, \text{right}, \Delta_i) \right) p(\text{stop} | \langle P,w \rangle, H, \text{right}, \Delta_{m+1}) \]
Collins Models 2 & 3 (1997)

(blackboard)
Other Details

• Smoothing: deleted interpolation.
• Unknown words: every type with count $\leq 5$ became UNK
• Tagging is not a separate stage; it is just part of the parse.
Further Refinements

• Base noun phrases
  – Labeled “NPB”
  – First-order Markov model for children of head!

• Coordinators (“and”) predicted together with the later argument.

• Punctuation treated similarly (see the 2003 paper)
Charniak (1997)

• Similar setup.
  – Lexicalized PCFG, factored model for rules
  – Tags don’t travel up the tree as in Collins
  – Tagging part of parsing
  – Deleted interpolation for smoothing

• Used an additional 30 million words of unannotated data.
Charniak (1997)

- $p(\text{Adv } \text{saw} \ \text{NP} \ \text{PP} | \ \text{VP}_{\text{saw}}, S)$
- $p(\text{somehow} | \ \text{VP}_{\text{saw}}, \ \text{Adv})$
- $p(\text{cat} | \ \text{VP}_{\text{saw}}, \ \text{NP})$
- $p(\text{with} | \ \text{VP}_{\text{saw}}, \ \text{PP})$
Charniak (2000)

• The 2000 parser is “maximum entropy inspired.”

• It is closer to Collins’ model (Markovized children), but the estimation is bizarre.
  – Smoothed, backed-off probabilities are multiplied together - almost like a product of experts.
## Comparison

<table>
<thead>
<tr>
<th></th>
<th>labeled recall</th>
<th>labeled precision</th>
<th>average crossing brackets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collins</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Model 1</td>
<td>87.5</td>
<td>87.7</td>
<td>1.09</td>
</tr>
<tr>
<td>Model 2</td>
<td>88.1</td>
<td>88.3</td>
<td>1.06</td>
</tr>
<tr>
<td>Model 3</td>
<td>88.0</td>
<td>88.3</td>
<td>1.05</td>
</tr>
<tr>
<td>Charniak</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1997</td>
<td>86.7</td>
<td>86.6</td>
<td>1.20</td>
</tr>
<tr>
<td>2000</td>
<td>89.6</td>
<td>89.5</td>
<td>0.88</td>
</tr>
</tbody>
</table>