Non-Independence I
- Independence assumptions are often too strong.
  - All NPs
    - NP: 11%
    - PP: 9%
    - DT: 9%
    - NN: 4%
    - PRP: 6%
  - NPs under S
    - NP: 9%
    - PP: 9%
    - DT: 9%
    - NN: 4%
    - PRP: 6%
  - NPs under VP
    - NP: 21%
    - PP: 9%
    - DT: 7%
    - NN: 4%
    - PRP: 6%
- Example: the expansion of an NP is highly dependent on the parent of the NP (i.e., subjects vs. objects).
- Also: the subject and object expansions are correlated!

Non-Independence II
- Who cares?
  - NB, HMMs, all make false assumptions!
  - For generation, consequences would be obvious.
  - For parsing, does it impact accuracy?
- Symptoms of overly strong assumptions:
  - Rewrites get used where they don’t belong.
  - Rewrites get used too often or too rarely.

Breaking Up the Symbols
- We can relax independence assumptions by encoding dependencies into the PCFG symbols:
  - Parent annotation [Johnson 98]
  - Marking possessive NPs
- What are the most useful “features” to encode?

Lexicalization
- Lexical heads important for certain classes of ambiguities (e.g., PP attachment):
  - Lexicalizing grammar creates a much larger grammar. (cf. next week)
    - Sophisticated smoothing needed
    - Smarter parsing algorithms
    - More data needed
- How necessary is lexicalization?
  - Bilexical vs. monolexical selection
  - Closed vs. open class lexicalization
Typical Experimental Setup

- Corpus: Penn Treebank, WSJ

Training: sections 02-21
Development: section 22 (here, first 20 files)
Test: section 23

- Accuracy – F1: harmonic mean of per-node labeled precision and recall.
- Here: also size – number of symbols in grammar.
  - Passive / complete symbols: NP, NP+S
  - Active / incomplete symbols: NP → NP CC •

Horizontal Markovization

- Order 1
  - NP
  - NP → NP

- Order ∞
  - NP
  - NP → NP

Vertical Markovization

- Vertical Markov order: rewrites depend on past & ancestor nodes.
  (cf. parent annotation)

Vertical and Horizontal

- Examples:
  - Raw treebank: \( v=1, h=\infty \)
  - Johnson 98: \( v=2, h=\infty \)
  - Collins 99: \( v=2, h=2 \)
  - Best F1: \( v=3, h=2v \)

Unary Splits

- Problem: unary rewrites used to transmute categories so a high-probability rule can be used.
- Solution: Mark unary rewrite sites with -U

Tag Splits

- Problem: Treebank tags are too coarse.
- Example: Sentential, PP, and other prepositions are all marked IN.
- Partial Solution: Subdivide the IN tag.
### Other Tag Splits

- **UNARY-DT**: mark demonstratives as DT^U ("the X" vs. "those")
- **UNARY-RB**: mark phrasal adverbs as RB^U ("quickly" vs. "very")
- **TAG-PA**: mark tags with non-canonical parents ("not" is an RB^VP)
- **SPLIT-AUX**: mark auxiliary verbs with –AUX
  [cf. Charniak 97]
- **SPLIT-CC**: separate "but" and "&" from other conjunctions
- **SPLIT%-": "%" gets its own tag.

<table>
<thead>
<tr>
<th></th>
<th>F1</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNARY-DT</td>
<td>80.4</td>
<td>8.1K</td>
</tr>
<tr>
<td>UNARY-RB</td>
<td>80.5</td>
<td>8.1K</td>
</tr>
<tr>
<td>TAG-PA</td>
<td>81.2</td>
<td>8.5K</td>
</tr>
<tr>
<td>SPLIT-AUX</td>
<td>81.6</td>
<td>9.0K</td>
</tr>
<tr>
<td>SPLIT-CC</td>
<td>81.7</td>
<td>9.1K</td>
</tr>
<tr>
<td>SPLIT-%: &quot;%&quot; gets its own tag.</td>
<td>81.8</td>
<td>9.3K</td>
</tr>
</tbody>
</table>

### Some Test Set Results

<table>
<thead>
<tr>
<th>Parser</th>
<th>LP</th>
<th>LR</th>
<th>F1</th>
<th>CB</th>
<th>0 CB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Magerman 95</td>
<td>84.9</td>
<td>84.6</td>
<td>84.7</td>
<td>1.26</td>
<td>56.6</td>
</tr>
<tr>
<td>Collins 96</td>
<td>86.3</td>
<td>85.8</td>
<td>86.0</td>
<td>1.14</td>
<td>59.9</td>
</tr>
<tr>
<td>Unlexicalized</td>
<td>86.9</td>
<td>85.7</td>
<td>86.3</td>
<td>1.10</td>
<td>60.3</td>
</tr>
<tr>
<td>Charniak 97</td>
<td>87.4</td>
<td>87.5</td>
<td>87.4</td>
<td>1.00</td>
<td>62.1</td>
</tr>
<tr>
<td>Collins 99</td>
<td>88.7</td>
<td>88.6</td>
<td>88.6</td>
<td>0.90</td>
<td>67.1</td>
</tr>
</tbody>
</table>

- Beats "first generation" lexicalized parsers.
- Lots of room to improve – more complex models next.

### The Game of Designing a Grammar

- Annotation refines base treebank symbols to improve statistical fit of the grammar
- Parent annotation [Johnson 98]

```
S
   NP she
   PRP VBD NP-noise
   She heard DT NN the noise
```

- Head lexicalization [Collins 99, Charniak 00]
Manual Annotation

- Manually split categories
  - NP: subject vs object
  - DT: determiners vs demonstratives
  - IN: sentential vs prepositional
- Advantages:
  - Fairly compact grammar
  - Linguistic motivations
- Disadvantages:
  - Performance leveled out
  - Manually annotated manual annotation

<table>
<thead>
<tr>
<th>Model</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naive Treebank Grammar</td>
<td>72.6</td>
</tr>
<tr>
<td>Klein &amp; Manning '03</td>
<td>86.3</td>
</tr>
</tbody>
</table>

Automatic Annotation Induction

- Advantages:
  - Automatically learned:
    - Label all nodes with latent variables.
    - Same number $k$ of subcategories for all categories.
- Disadvantages:
  - Grammar gets too large
  - Most categories are oversplit while others are undersplit.

<table>
<thead>
<tr>
<th>Model</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Klein &amp; Manning '03</td>
<td>86.3</td>
</tr>
<tr>
<td>Matsuzaki et al. '05</td>
<td>86.7</td>
</tr>
</tbody>
</table>

Learning Latent Annotations

EM algorithm:
- Brackets are known
- Base categories are known
- Only induce subcategories

Just like Forward-Backward for HMMs.

Refinement of the DT tag

Adaptive Splitting

- Want to split complex categories more
- Idea: split everything, roll back splits which were least useful
Adaptive Splitting

- Evaluate loss in likelihood from removing each split =
  
  \[ \text{Data likelihood with split reversed} - \text{Data likelihood with split} \]

- No loss in accuracy when 50% of the splits are reversed.

Adaptive Splitting Results

<table>
<thead>
<tr>
<th>Model</th>
<th>F1 ≤ 40 words</th>
<th>F1 all words</th>
</tr>
</thead>
<tbody>
<tr>
<td>Previous</td>
<td>88.4</td>
<td></td>
</tr>
<tr>
<td>With 50% Merging</td>
<td>89.5</td>
<td></td>
</tr>
</tbody>
</table>

Final Results (Accuracy)

<table>
<thead>
<tr>
<th></th>
<th>F1 ≤ 40 words</th>
<th>F1 all words</th>
</tr>
</thead>
<tbody>
<tr>
<td>Charniak &amp; Johnson '05 (generative)</td>
<td>90.1</td>
<td>89.6</td>
</tr>
<tr>
<td>Petrov and Klein '07</td>
<td>90.6</td>
<td>90.1</td>
</tr>
<tr>
<td>Dubey '05</td>
<td>76.3</td>
<td></td>
</tr>
<tr>
<td>Petrov and Klein '07</td>
<td>80.8</td>
<td>80.1</td>
</tr>
<tr>
<td>Chiang et al. '02</td>
<td>80.0</td>
<td>76.6</td>
</tr>
<tr>
<td>Petrov and Klein '07</td>
<td>86.3</td>
<td>83.4</td>
</tr>
</tbody>
</table>

Final Results

<table>
<thead>
<tr>
<th>Model</th>
<th>F1 ≤ 40 words</th>
<th>F1 all words</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parser</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Petrov et. al. '06</td>
<td>90.2</td>
<td>89.7</td>
</tr>
<tr>
<td>Klein &amp; Manning '03</td>
<td>86.3</td>
<td>85.7</td>
</tr>
<tr>
<td>Matsuzaki et al. '05</td>
<td>86.7</td>
<td>86.1</td>
</tr>
<tr>
<td>Collins '99</td>
<td>88.6</td>
<td>88.2</td>
</tr>
<tr>
<td>Charniak &amp; Johnson '05</td>
<td>90.1</td>
<td>89.6</td>
</tr>
</tbody>
</table>

Number of Phrasal Subcategories

Number of Lexical Subcategories
Learned Splits

- Proper Nouns (NNP):
  - NNP-12: John, Robert, James
  - NNP-2: J. E. L.
  - NNP-1: Bush, Noriega, Peters
  - NNP-15: New, San, Wall
  - NNP-3: York, Francisco, Street

- Personal Pronouns (PRP):
  - PRP-0: It, He, I
  - PRP-1: It, he, they
  - PRP-2: It, them, him

Learned Splits

- Relative Adverbs (RBR):
  - RBR-0: further, lower, higher
  - RBR-1: more, less, More
  - RBR-2: earlier, Earlier, later

- Cardinal Numbers (CD):
  - CD-7: 1
  - CD-4: 1989
  - CD-11: million, billion, trillion

Coarse-to-Fine Parsing

[Goodman 97, Charniak & Johnson '05]

Coarse grammar

NP VP ... QP ...
NP-17 NP-12 VP-6 VP-31

Refined grammar

NP VP ... QP ...
NP-17 NP-12 VP-6 VP-31

Prune?

For each chart item $X(i,j)$, compute posterior probability:

$$P_{in}(X(i,j)) P_{node}(X(i,j)) P_{end}(root, 0, n)$$

E.g. consider the span 5 to 12:

coarse:

refined:

Hierarchical Pruning

coarse:

split in two:

split in four:

split in eight:

Bracket Posteriors (after $G_0$)
Bracket Posteriors (after $G_1$)

Bracket Posteriors (Final)

Bracket Posteriors (Best Tree)

Parsing times

1621 min
111 min
35 min
15 min
(no search error)