### Lecture 14: PCFGs

**Statistical NLP**  
**Spring 2010**  
**UC Berkeley**  
**Dan Klein – UC Berkeley**

---

#### Treebank PCFGs

- Use PCFGs for broad coverage parsing
- Can take a grammar right off the trees (doesn’t work well):

```
ROOT → S
S → NP VP
NP → PRP
VP → VBD ADJP
```

<table>
<thead>
<tr>
<th>Model</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>72.0</td>
</tr>
</tbody>
</table>

#### Conditional Independence?

- Not every NP expansion can fill every NP slot
  - A grammar with symbols like “NP” won’t be context-free
  - Statistically, conditional independence too strong

#### Non-Independence

- Independence assumptions are often too strong.
  - 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!

#### Grammar Refinement

- Example: PP attachment

#### Grammar Refinement

- Structure Annotation [Johnson ’98, Klein&Manning ’03]
- Lexicalization [Collins ’99, Charniak ’00]
- Latent Variables [Matsuzaki et al. 05, Petrov et al. ’06]
The Game of Designing a Grammar

- Annotation refines base treebank symbols to improve statistical fit of the grammar
  - Structural annotation

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

Vertical Markovization

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

<table>
<thead>
<tr>
<th>Vertical Markov Order</th>
<th>Order 1</th>
<th>Order 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symbols</td>
<td>75%</td>
<td>77%</td>
</tr>
<tr>
<td>1</td>
<td>78%</td>
<td>78%</td>
</tr>
<tr>
<td>2</td>
<td>79%</td>
<td>79%</td>
</tr>
<tr>
<td>3</td>
<td>79%</td>
<td>79%</td>
</tr>
<tr>
<td>4</td>
<td>79%</td>
<td>79%</td>
</tr>
<tr>
<td>5</td>
<td>79%</td>
<td>79%</td>
</tr>
<tr>
<td>6</td>
<td>79%</td>
<td>79%</td>
</tr>
<tr>
<td>7</td>
<td>79%</td>
<td>79%</td>
</tr>
<tr>
<td>8</td>
<td>79%</td>
<td>79%</td>
</tr>
<tr>
<td>9</td>
<td>79%</td>
<td>79%</td>
</tr>
<tr>
<td>10</td>
<td>79%</td>
<td>79%</td>
</tr>
</tbody>
</table>

Horizontal Markovization

<table>
<thead>
<tr>
<th>Horizontal Markov Order</th>
<th>Order 1</th>
<th>Order ∞</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symbols</td>
<td>73%</td>
<td>73%</td>
</tr>
<tr>
<td>0</td>
<td>75%</td>
<td>75%</td>
</tr>
<tr>
<td>1</td>
<td>76%</td>
<td>76%</td>
</tr>
<tr>
<td>2</td>
<td>77%</td>
<td>77%</td>
</tr>
<tr>
<td>3</td>
<td>78%</td>
<td>78%</td>
</tr>
<tr>
<td>4</td>
<td>79%</td>
<td>79%</td>
</tr>
<tr>
<td>5</td>
<td>79%</td>
<td>79%</td>
</tr>
<tr>
<td>6</td>
<td>79%</td>
<td>79%</td>
</tr>
<tr>
<td>7</td>
<td>79%</td>
<td>79%</td>
</tr>
<tr>
<td>8</td>
<td>79%</td>
<td>79%</td>
</tr>
<tr>
<td>9</td>
<td>79%</td>
<td>79%</td>
</tr>
<tr>
<td>10</td>
<td>79%</td>
<td>79%</td>
</tr>
</tbody>
</table>

 Unary Splits

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

<table>
<thead>
<tr>
<th>Annotation</th>
<th>F1</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base</td>
<td>77.8</td>
<td>7.5K</td>
</tr>
<tr>
<td>UNARY</td>
<td>78.3</td>
<td>8.0K</td>
</tr>
</tbody>
</table>

Tag Splits

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

<table>
<thead>
<tr>
<th>Annotation</th>
<th>F1</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Previous</td>
<td>78.3</td>
<td>8.0K</td>
</tr>
<tr>
<td>SPLIT-IN</td>
<td>80.3</td>
<td>8.1K</td>
</tr>
</tbody>
</table>
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>Tag Type</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-%</td>
<td>81.8</td>
<td>9.3K</td>
</tr>
</tbody>
</table>

A Fully Annotated (Unlex) Tree

```
S
  NP
    NP-B
    VP-B
    NP

This is buying
```

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>Magemann 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

```
She heard the noise
```

- Annotation refines base treebank symbols to improve statistical fit of the grammar
- Structural annotation [Johnson '98, Klein and Manning '03]
- Head lexicalization [Collins '99, Charniak '00]

Problems with PCFGs

```
NP: Take leftmost NP
   Take rightmost NP
   Take rightmost JJ
   Take right child

VP: Take leftmost VB
    Take left child
```

- What’s different between basic PCFG scores here?
- What (lexical) correlations need to be scored?

Lexicalized Trees

```
NP
  NP
    NP-B
    VP-B
    NP

houses cats
```

- Add “headwords” to each phrasal node
  - Syntactic vs. semantic heads
  - Headship not in (most) treebanks
  - Usually use head rules, e.g.:
    - NP:
      - Take leftmost NP
      - Take rightmost N
      - Take rightmost JJ
      - Take right child
    - VP:
      - Take leftmost VB
      - Take left child

Lexicalized PCFGs?
- Problem: we now have to estimate probabilities like
  \[ VP_{(\text{aw})} \rightarrow \text{VBP}_{(\text{aw})} \text{NP}_{(\text{aw})} \text{RB}_{(\text{aw})} \text{NP}_{(\text{aw})} \text{today} \]
- Never going to get these atomically off of a treebank
- Solution: break up derivation into smaller steps

Lexical Derivation Steps
- A derivation of a local tree [Collins 99]
  - Choose a head tag and word
  - Choose a complement bag
  - Generate children (incl. adjuncts)
  - Recursively derive children

Lexicalized CKY

Pruning with Beams
- The Collins parser prunes with per-cell beams [Collins 99]
  - Essentially, run the O(n^3) CKY
  - Remember only a few hypotheses for each span <i,j>.
    - If we keep K hypotheses at each span, then we do at most O(nK) work per span (why?)
    - Keeps things more or less cubic
  - Also: certain spans are forbidden entirely on the basis of punctuation (crucial for speed)

Pruning with a PCFG
- The Charniak parser prunes using a two-pass approach [Charniak 97+]
  - First, parse with the base grammar
  - For each X[i,j] calculate \( P(X[i,j]) \)
    - This isn’t trivial, and there are clever speed ups
  - Second, do the full O(n^3) CKY
    - Skip any X[i,j] which had low (say, < 0.0001) posterior
    - Avoids almost all work in the second phase!
- Charniak et al 06: can use more passes
- Petrov et al 07: can use many more passes

Pruning with A*
- You can also speed up the search without sacrificing optimality
- For agenda-based parsers:
  - Can select which items to process first
  - Can do with any “figure of merit” [Charniak 98]
  - If your figure-of-merit is a valid A* heuristic, no loss of optimality [Klein and Manning 03]
Results

- **Some results**
  - Collins 99 – 88.6 F1 (generative lexical)
  - Charniak and Johnson 05 – 89.7 / 91.3 F1 (generative lexical / reranked)
  - Petrov et al 06 – 90.7 F1 (generative unlexical)
  - McClosky et al 06 – 92.1 F1 (gen + rerank + self-train)

- **However**
  - Bilexical counts rarely make a difference (why?)
  - Gildea 01 – Removing bilexical counts costs < 0.5 F1

The Game of Designing a Grammar

- Annotation refines base treebank symbols to improve statistical fit of the grammar
  - Structural annotation
  - Head lexicalization
  - Automatic clustering?

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

Hierarchical refinement

Hierarchical Estimation Results

<table>
<thead>
<tr>
<th>Total Number of grammars</th>
<th>Parsing accuracy (F1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>74</td>
</tr>
<tr>
<td>300</td>
<td>76</td>
</tr>
<tr>
<td>500</td>
<td>78</td>
</tr>
<tr>
<td>700</td>
<td>80</td>
</tr>
<tr>
<td>900</td>
<td>82</td>
</tr>
<tr>
<td>1100</td>
<td>84</td>
</tr>
<tr>
<td>1300</td>
<td>86</td>
</tr>
<tr>
<td>1500</td>
<td>88</td>
</tr>
<tr>
<td>1700</td>
<td>88</td>
</tr>
</tbody>
</table>

Flat Training | 87.3
Hierarchical Training | 88.4

Adaptive Splitting

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

Refinement of the , tag

Splitting all categories equally is wasteful:

Adaptive Splitting Results

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

Model | F1   |
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

Number of Phrasal Subcategories

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: one two Three
  - CD-11: million billion trillion
  - CD-0: 1 50 100
  - CD-3: 1 30 31
  - CD-9: 78 58 34

Learned Splits

Coarse-to-Fine Inference

- **Example: PP attachment**

Prune?

For each chart item $X[i,j]$, compute posterior probability:

$$P_{\text{coarse}}(X[i,j]) \cdot P_{\text{root}}(X[i,j])$$

$$P_{\text{root}}(\text{root}, 0, n)$$

E.g. consider the span 5 to 12:
Bracket Posteriors

Hierarchical Pruning

Final Results (Accuracy)

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

Still higher numbers from reranking / self-training methods