Head-Driven PCFGs with Latent-Head Statistics

Detlef Prescher
University of Amsterdam
prescher@science.uva.nl

October, 2005
Overview

• Motivation

• Theory
  – Background: Head Lexicalization
  – Latent-Head Models
  – Estimation of Latent-Head Models

• Experiments

• Results

• Discussion

• Conclusion
Motivation

- *state-of-the-art* parsers are lexicalized
- *accurate* parsers can be read off a manually refined tree-bank
- *lexicalized* parsers often suffer from sparse data
- *manual mark-up* is costly

Is it possible to *automatically* induce an *accurate* parser from a tree-bank without resorting to full lexicalization?
Parse tree, and a list of the rules it contains

(Charniak, 1997)
Background: Head Lexicalization (2/7)

\[ p_{CHARNIAK97}( \text{this local tree} ) = p( r \mid C, h, C_p ) \times \prod_{i=1}^{n+m} p( d_i \mid D_i, C, h ) \]

\( r \) is the unlexicalized rule,
\( C_p \) is C’s parent category

**Internal rule, and its probability (Charniak, 1997)**
Background: Head Lexicalization (3/7)

\[
p(r \mid C, h, C_p) = \lambda_1 \cdot \hat{p}(r \mid C, h, C_p) \\
+ \lambda_2 \cdot \hat{p}(r \mid C, h) \\
+ \lambda_3 \cdot \hat{p}(r \mid C, \text{class}(h)) \\
+ \lambda_4 \cdot \hat{p}(r \mid C, C_p) \\
+ \lambda_5 \cdot \hat{p}(r \mid C)
\]

\[
p(d \mid D, C, h) = \lambda_1 \cdot \hat{p}(d \mid D, C, h) \\
+ \lambda_2 \cdot \hat{p}(d \mid D, C, \text{class}(h)) \\
+ \lambda_3 \cdot \hat{p}(d \mid D, C) \\
+ \lambda_4 \cdot \hat{p}(d \mid D)
\]

Smoothing by deleted interpolation (Charniak, 1997)
Background: Head Lexicalization (4/7)

\[
p_{\text{STANDARD-PCFG}}(\text{this sub-tree})
= p(D_1:C:h \ldots D_m:C:h \ H:h \ D_{m+1}:C:h \ldots D_{m+n}:C:h \mid C:h) \times \prod_{i=1}^{m+n} p(D_i:d_i \mid D_i:C:h)
= p(D_1 \ldots D_m H D_{m+1} \ldots D_{m+n} \mid C,h) \times \prod_{i=1}^{m+n} p(d_i \mid D_i, C, h)
= p(r \mid C, h) \times \prod_{i=1}^{m+n} p(d_i \mid D_i, C, h)
\]

(\text{*r is the unlexicalized rule*})

\textbf{Transformed rule, and its probability}
\textbf{(Carroll and Rooth, 1998)}
Transformed parse tree, and a list of the rules it contains

*(Carroll and Rooth, 1998)*
Background: Head Lexicalization (6/7)

Starting Rules

\[ S \rightarrow S:h \]

Lexicalized Rules

\[ C:h \rightarrow D_1:C:h \ldots D_m:C:h \quad H:h \quad D_{m+1}:C:h \ldots D_{m+n}:C:h \]

Dependencies

\[ D:C:h \rightarrow D:d \]

Lexical Rules

\[ C:w \rightarrow w \]

Context-free rule types in the transform

(Carroll and Rooth, 1998)
Background: Head Lexicalization

• Charniak (1997)
  – transforms a given tree bank to the lexicalized format
  – counts and smoothes for parameter estimation
  – exploits head classes in a back-off scheme
  – observes an impressive gain in performance (about 14%)

• Carroll and Rooth (1998)
  – transform a manually written grammar (in the spirit of Charniak)
  – estimate on a corpus of sentences (with the EM algorithm)
  – smooth (but do not use head-classes)
  – observe only a small gain in performance (about 1%)

Summary of both approaches
Latent-Head Models (1/2)

Learning from linguistic principles

(i) all rules have head markers,

(ii) information is projected up a chain of categories marked as heads,

(iii) lexical entries carry latent-head values which can be learned.

Probabilistic Head-Lexicalized Context-Free Grammars with Latent Heads

The grammar transformation of Carroll and Rooth (1998) is modified to satisfy principle (iii); The latents heads are estimated on the original treebank.
Latent-Head Models (2/2)

Starting Rule:
\[ S \rightarrow S:h_V \]

Lexicalized Rules:
\[ S:h_V \rightarrow NP:S:h_V \ VP:h_V \ PUNC:S:h_V \]
\[ NP:h_N \rightarrow ADJ:NP:h_N \ N:h_N \]
\[ VP:h_V \rightarrow V:h_V \]

Dependencies:
\[ NP:S:h_V \rightarrow NP:h_N \]
\[ PUNC:S:h_V \rightarrow PUNC:h_{PUNC} \]
\[ ADJ:NP:h_N \rightarrow ADJ:h_{ADJ} \]

Lexical Rules:
\[ ADJ:h_{ADJ} \rightarrow Corporate \]
\[ N:h_N \rightarrow profits \]
\[ V:h_V \rightarrow rose \]
\[ PUNC:h_{PUNC} \rightarrow . \]

Model 1 (Completely Latent Heads):
\[ h_{ADJ}, h_N, h_V, \text{ and } h_{PUNC} \in \{1, \ldots, L\} \]

Number of Latent-Head Types = \[
\begin{cases} 
L & \text{for Model 1} \\
|POS| \times L & \text{for Model 2} 
\end{cases}
\]

\[(L \text{ is a free parameter)}\]

Model 2 (Latent Heads Based on POS Tags):
\[ h_{ADJ} \in \{ADJ\} \times \{1, \ldots, L\} \]
\[ h_N \in \{N\} \times \{1, \ldots, L\} \]
\[ h_V \in \{V\} \times \{1, \ldots, L\} \]
\[ h_{PUNC} \in \{PUNC\} \times \{1, \ldots, L\} \]
Estimation

...from latent-head distributions (EM scheme)

- **E-step.** Generate a lexicalized tree-bank $T_{\text{LEX}}$, by running over the unlexicalized trees $t$ in the tree-bank, generating all their transforms $t'$, and allocating the frequency $\text{count}(t') := \text{count}(t) \cdot p(t'|t)$

- **M-step.** Read the tree-bank grammar off $T_{\text{LEX}}$, by calculating relative frequencies for all rules with the same left-hand side

...from most probable heads (More traditional scheme)

- **Annotate.** Take the best EM model to generate the most probable head-lexicalized version of each tree in the original tree-bank

- **Count.** Read the tree-bank grammar off this Viterbi-lexicalized tree-bank
Experiments

Training on WSJ sections 2-21 of the Penn Tree-Bank

- insert top nodes, delete empty nodes and non-syntactic information, introduce word-suffix based unknown-word symbols
- different runs of the EM algorithm with varying starting parameters and iteration numbers (from 50 to 200) for both models; Different settings, however, affected results on a held-out corpus only up to 0.5%

Evaluation on WSJ Section 22

- unknown-words were mapped to unknown-word symbols; Viterbi parses were calculated from the unpruned parse forests, and de-transformed to the original format (by stripping away the latent heads)
## Results

### Most Probable Heads

<table>
<thead>
<tr>
<th></th>
<th>Model 1 (latent)</th>
<th>Model 2 (POS+latent)</th>
<th>Δ =</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>15 400</td>
<td>25 000</td>
<td>73.5</td>
</tr>
<tr>
<td>L=2</td>
<td>17 900</td>
<td>32 300</td>
<td>76.3</td>
</tr>
<tr>
<td>L=5</td>
<td>22 800</td>
<td>46 200</td>
<td>80.7</td>
</tr>
<tr>
<td>L=10</td>
<td>28 100</td>
<td>58 900</td>
<td>83.3</td>
</tr>
</tbody>
</table>

### Head Distributions

<table>
<thead>
<tr>
<th></th>
<th>Model 1 (latent)</th>
<th>Model 2 (POS+latent)</th>
<th>Δ =</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>15 400</td>
<td>25 000</td>
<td>73.5</td>
</tr>
<tr>
<td>L=2</td>
<td>25 900</td>
<td>49 500</td>
<td>76.9</td>
</tr>
<tr>
<td>L=5</td>
<td>49 200</td>
<td>116 300</td>
<td>82.0</td>
</tr>
<tr>
<td>L=10</td>
<td>79 200</td>
<td>224 300</td>
<td>84.6</td>
</tr>
</tbody>
</table>

**Parsing results in LP/LR F₁ (the baseline is L = 1)**
Discussion (1/4)

- **useful head-classes have been learned**: both original grammar and POS-lexicalized grammar are outperformed,
- **the granularity of the head classes is not yet fine enough**: increasing $L$ increases $F_1$; further refinements may lead to even better results,
- **learning from head-class distributions** outperforms more traditional learning from most probable annotations,
- **POS information benefits coarse-grained models** ($L = 1, 2, 5$), but the best models with and without POS are almost on a par,
- **the best latent-head model** ($\approx 225\,000$ rules, $F_1 = 85.7\%$ with POS) is smaller than fully-lexicalized models,
- **the latent-head model which learned the most** ($\Delta = 11.1\%$) is even smaller ($\approx 80\,000$ rules, $F_1 = 84.6\%$ without POS)
## Discussion (2/4)

<table>
<thead>
<tr>
<th></th>
<th>LP</th>
<th>LR</th>
<th>F₁</th>
<th>Exact</th>
<th>CB</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Model 1 (this paper)</strong></td>
<td>84.8</td>
<td>84.4</td>
<td>84.6</td>
<td>26.4</td>
<td>1.37</td>
</tr>
<tr>
<td>Magerman (1995)</td>
<td>84.9</td>
<td>84.6</td>
<td>1.26</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Model 2 (this paper)</strong></td>
<td>85.7</td>
<td>85.7</td>
<td>85.7</td>
<td>29.3</td>
<td>1.29</td>
</tr>
<tr>
<td>Collins (1996)</td>
<td>86.3</td>
<td>85.8</td>
<td></td>
<td>1.14</td>
<td></td>
</tr>
<tr>
<td>Matsuzaki et al. (2005)</td>
<td>86.6</td>
<td>86.7</td>
<td></td>
<td>1.19</td>
<td></td>
</tr>
<tr>
<td>Klein and Manning (2003)</td>
<td>86.9</td>
<td>85.7</td>
<td>86.3</td>
<td>30.9</td>
<td>1.10</td>
</tr>
<tr>
<td>Charniak (1997)</td>
<td>87.4</td>
<td>87.5</td>
<td></td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>Collins (1997)</td>
<td>88.6</td>
<td>88.1</td>
<td></td>
<td>0.91</td>
<td></td>
</tr>
</tbody>
</table>

**Comparison with other parsers (sentences of length ≤ 40)**
• in contrast to fully-lexicalized models, latent-head models bundle lexical and contextual information from the whole tree-bank into abstract higher-order heads; They do not suffer from sparse data and can do without pruning and smoothing.

• in contrast to manual linguistic mark-up (Klein and Manning, 2003), automatic linguistic mark-up by latent-head models is not cost and time intensive, and moreover, it is not based on individual intuition,

• latent-head models complement and extend the approach of discovering latent head-markers in tree-banks to improve manually written head-percolation rules (Chiang and Bikel, 2002)
Discussion (4/4)

- compared to the approach of learning general latent annotations for PCFGs (Matsuzaki et al., 2005), latent-head models
  - are based on an explicit linguistic grammar,
  - are constrained by the linguistic principle of headedness,
  - are three orders of magnitude more space efficient,
  - do without smoothing and pruning,
  - are on a par with the so-called 'Viterbi complete tree parsing' regime ($F_1=85.5\%$), suggesting that both models have learned a comparable degree of information (being a surprise!),
  - should also incorporate a routine to bag all complete parses with the same incomplete skeleton to gain a crucial 1% final improvement.
Conclusion

• we introduced a method for inducing a head-driven PCFG with latent-head statistics from a tree-bank

• the automatically trained parser is time and space efficient and achieves a performance already better than early lexicalized ones

• this suggests that our grammar induction method can be successfully applied across domains, languages, and tree-bank annotations
Thank you!
Latent-Head Models

• transform the *tree-bank grammar* instead of the tree-bank
• use *latent heads* instead of full word-forms as heads
• learn the latent heads via *unsupervised estimation on the tree-bank*
• reduce the *parameter space* drastically
• select from the *full parse forest* for parsing (without pruning)
• observe a performance better than those of early lexicalized parsers

*Our approach: a blend of previous ones*
Latent-Head Models

• latent-head models perform unambiguous transformations for $L = 1$: no relevant changes for Model 1; lexicalization with POS tags for Model 2

• in contrast to full lexicalization, latent-head models map an unlexicalized tree to multiple transforms (for $L \geq 2$)

• although information is freely introduced at the lexical level, it is not freely distributed over the nodes of the tree. Rather, the space of latent information for a tree is constrained according the linguistic principle of headedness

Some features