Head-Driven PCFGs with Latent-Head Statistics

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

October, 2005

Overview

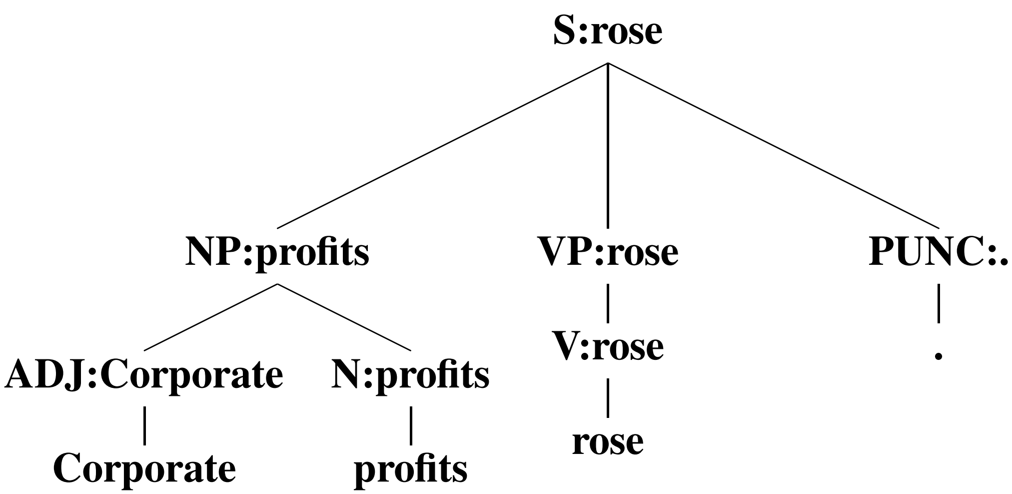
- Motivation
- Theory
  - Background: Head Lexicalization
  - Latent-Head Models
  - Estimation of Latent-Head Models
- Experiments
- Results
- Discussion
- Conclusion

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?

Background: Head Lexicalization (1/7)



Internal Rules:

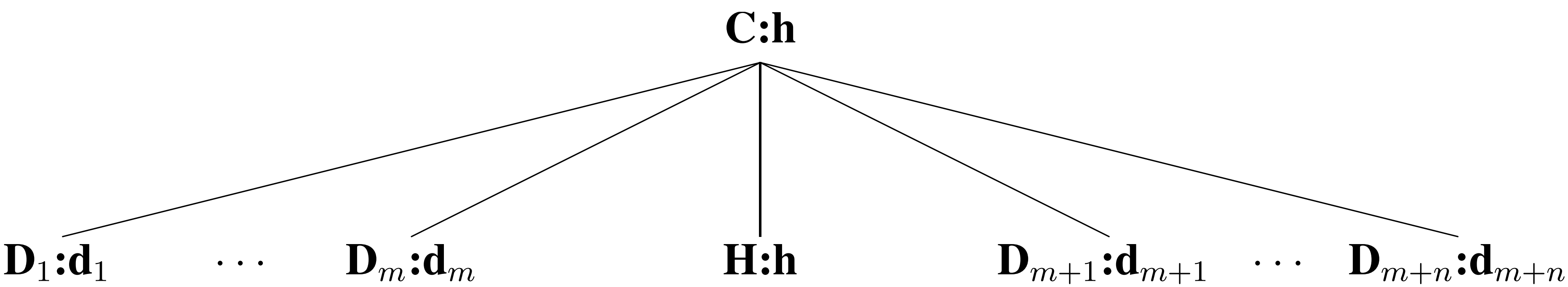
S:rose $\longrightarrow$ NP:profits VP:rose PUNC:.
NP:profits $\longrightarrow$ ADJ:Corporate N:profits
VP:rose $\longrightarrow$ V:rose

Lexical Rules:

ADJ:Corporate $\longrightarrow$ Corporate
N:profits $\longrightarrow$ profits
V:rose $\longrightarrow$ rose
PUNC:. $\longrightarrow$.

Parse tree, and a list of the rules it contains (Charniak, 1997)

Background: Head Lexicalization (2/7)



$$p_{\textbf{CHARNIAK97}}(\ \textbf{this local tree}\ ) \quad = \quad p(\ r \mid \mathbf{C}, h, \mathbf{C}_p\ ) \ \times\ \prod_{i=1}^{n+m} p(\ d_i \mid \mathbf{D}_i, \mathbf{C}, h\ )$$

(r **is the *unlexicalized* rule,**
$\mathbf{C}_p$ **is C's parent category**)

Internal rule, and its probability (Charniak, 1997)

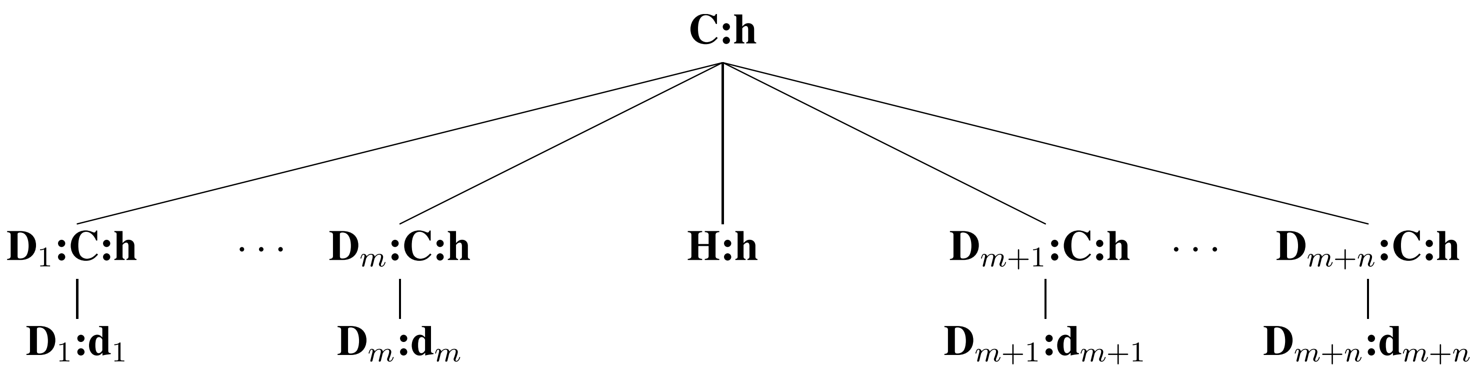
Background: Head Lexicalization (3/7)

$$
\begin{aligned}
p(\ r \mid \mathsf{C},\ h,\ \mathsf{C}_p\ ) \ = \ & \ \ \lambda_1 \cdot \hat{p}(\ r \mid \mathsf{C},\ h,\ \mathsf{C}_p\ ) \\
& + \lambda_2 \cdot \hat{p}(\ r \mid \mathsf{C},\ h\ ) \\
& + \lambda_3 \cdot \hat{p}(\ r \mid \mathsf{C},\ \mathsf{class}(h)\ ) \\
& + \lambda_4 \cdot \hat{p}(\ r \mid \mathsf{C},\ \mathsf{C}_p\ ) \\
& + \lambda_5 \cdot \hat{p}(\ r \mid \mathsf{C}\ )
\end{aligned}
$$

$$
\begin{aligned}
p(\ d \mid \mathsf{D},\ \mathsf{C},\ h\ ) \ = \ & \ \ \lambda_1 \cdot \hat{p}(\ d \mid \mathsf{D},\ \mathsf{C},\ h\ ) \\
& + \lambda_2 \cdot \hat{p}(\ d \mid \mathsf{D},\ \mathsf{C},\ \mathsf{class}(h)\ ) \\
& + \lambda_3 \cdot \hat{p}(\ d \mid \mathsf{D},\ \mathsf{C}\ ) \\
& + \lambda_4 \cdot \hat{p}(\ d \mid \mathsf{D}\ )
\end{aligned}
$$

Smoothing by deleted interpolation (Charniak, 1997)

Background: Head Lexicalization (4/7)



$p_{\textbf{STANDARD-PCFG}}(\ \textbf{this sub-tree}\ )$

$$= \ p(\ \mathbf{D}_1 : \mathbf{C} : h \ldots \mathbf{D}_m : \mathbf{C} : h \ \ \mathbf{H} : h \ \ \mathbf{D}_{m+1} : \mathbf{C} : h \ldots \mathbf{D}_{m+n} : \mathbf{C} : h \ \mid \ \mathbf{C} : h \ ) \ \times \ \prod_{i=1}^{m+n} \ p(\ \mathbf{D}_i : d_i \ \mid \ \mathbf{D}_i : \mathbf{C} : h \ )$$

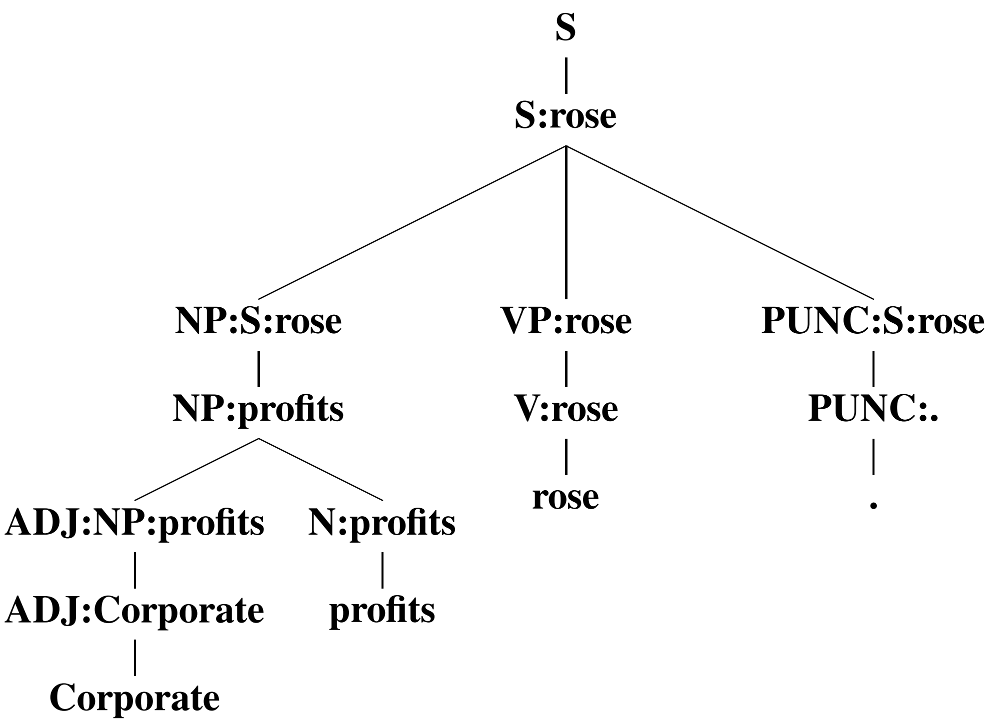
$$= \ p(\ \mathbf{D}_1 \ldots \mathbf{D}_m \ \mathbf{H} \ \mathbf{D}_{m+1} \ldots \mathbf{D}_{m+n} \ \mid \ \mathbf{C}, h \ ) \ \times \ \prod_{i=1}^{m+n} \ p(\ d_i \ \mid \ \mathbf{D}_i, \mathbf{C}, h \ )$$

$$= \ p(\ r \ \mid \ \mathbf{C}, h \ ) \ \times \ \prod_{i=1}^{m+n} \ p(\ d_i \ \mid \ \mathbf{D}_i, \mathbf{C}, h \ )$$

(r **is the *unlexicalized* rule**)

Transformed rule, and its probability
(Carroll and Rooth, 1998)

Background: Head Lexicalization (5/7)



Starting Rule:

S $\longrightarrow$ S:rose

Lexicalized Rules:

S:rose $\longrightarrow$ NP:S:rose VP:rose PUNC:S:rose
NP:profits $\longrightarrow$ ADJ:NP:profits N:profits
VP:rose $\longrightarrow$ V:rose

Dependencies:

NP:S:rose $\longrightarrow$ NP:profits
PUNC:S:rose $\longrightarrow$ PUNC:.
ADJ:NP:profits $\longrightarrow$ ADJ:Corporate

Lexical Rules:

ADJ:Corporate $\longrightarrow$ Corporate
N:profits $\longrightarrow$ profits
V:rose $\longrightarrow$ rose
PUNC:. $\longrightarrow$.

Transformed parse tree, and a list of the rules it contains (Carroll and Rooth, 1998)

Background: Head Lexicalization (6/7)

Starting Rules

$$\mathsf{S} \longrightarrow \quad \mathsf{S}{:}h$$

Lexicalized Rules

$$\mathsf{C}{:}h \longrightarrow \quad \mathsf{D}_1{:}\mathsf{C}{:}h \ \ldots \ \mathsf{D}_m{:}\mathsf{C}{:}h \ \mathsf{H}{:}h \ \mathsf{D}_{m+1}{:}\mathsf{C}{:}h \ \ldots \ \mathsf{D}_{m+n}{:}\mathsf{C}{:}h$$

Dependencies

$$\mathsf{D}{:}\mathsf{C}{:}h \longrightarrow \quad \mathsf{D}{:}d$$

Lexical Rules

$$\mathsf{C}{:}w \longrightarrow \quad w$$

Context-free rule types in the transform
(Carroll and Rooth, 1998)

Background: Head Lexicalization (7/7)

- 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,

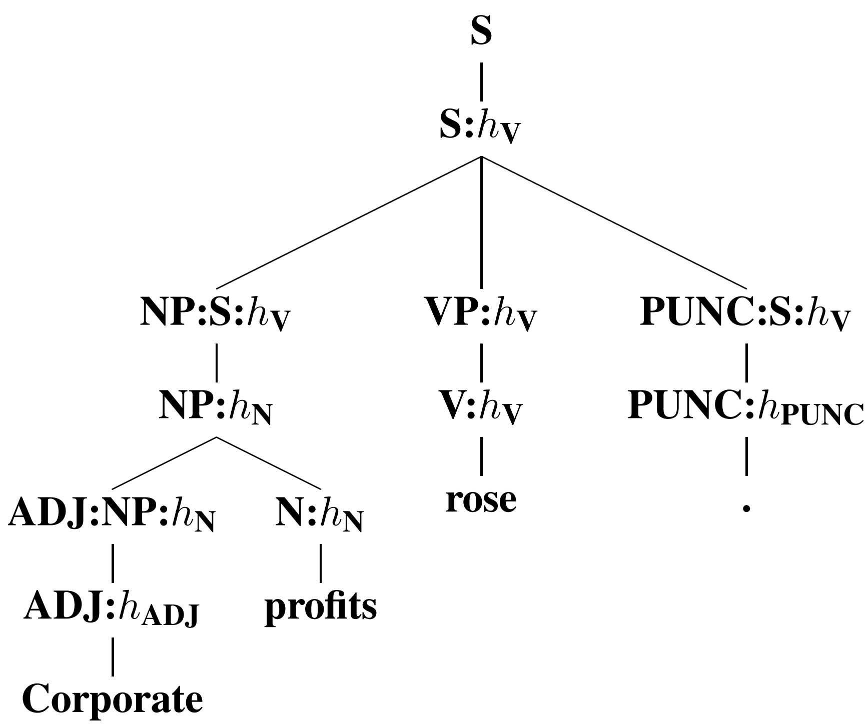
(ii) information is projected up a chain of categories marked as heads,

(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 $\longrightarrow$ S:h_V

Lexicalized Rules:

S:h_V $\longrightarrow$ NP:S:h_V VP:h_V PUNC:S:h_V

NP:h_N $\longrightarrow$ ADJ:NP:h_N N:h_N

VP:h_V $\longrightarrow$ V:h_V

Dependencies:

NP:S:h_V $\longrightarrow$ NP:h_N

PUNC:S:h_V $\longrightarrow$ PUNC:h_{PUNC}

ADJ:NP:h_N $\longrightarrow$ ADJ:h_{ADJ}

Lexical Rules:

ADJ:h_{ADJ} $\longrightarrow$ Corporate

N:h_N $\longrightarrow$ profits

V:h_V $\longrightarrow$ rose

PUNC:h_{PUNC} $\longrightarrow$.

Model 1 (Completely Latent Heads):

h_{ADJ}, h_N, h_V, and $h_{PUNC} \in \{1, \ldots, L\}$

Model 2 (Latent Heads Based on POS Tags):

$$\begin{aligned} h_{ADJ} &\in \{\text{ADJ}\} \times \{1, \ldots, L\} \\ h_N &\in \{\text{N}\} \times \{1, \ldots, L\} \\ h_V &\in \{\text{V}\} \times \{1, \ldots, L\} \\ h_{PUNC} &\in \{\text{PUNC}\} \times \{1, \ldots, L\} \end{aligned}$$

$$\textbf{Number of Latent-Head Types} = \begin{cases} L & \text{for Model 1} \\ |POS| \times L & \text{for Model 2} \end{cases} \qquad (L \text{ is a free parameter})$$

Estimation

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

- **E-step.** Generate a lexicalized tree-bank T_{LEX}, by running over the unlexicalized trees t in the tree-bank, generating all their transforms t', and allocating the frequency $\texttt{count}(t') := \texttt{count}(t) \cdot p(t'|t)$
- **M-step.** Read the tree-bank grammar off T_{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		Head Distributions	
	Model 1 (latent)	**Model 2** (POS+latent)	**Model 1** (latent)	**Model 2** (POS+latent)
baseline	(15 400) 73.5	(25 000) 78.9	(15 400) 73.5	(25 000) 78.9
L=2	(17 900) 76.3	(32 300) 81.1	(25 900) 76.9	(49 500) 81.6
L=5	(22 800) 80.7	(46 200) *83.3*	(49 200) 82.0	(116 300) 84.9
L=10	(28 100) *83.3*	(58 900) 82.6	(79 200) *84.6*	(224 300) **85.7**
	Δ=9.8	Δ=4.4	Δ=**11.1**	Δ=6.8

Parsing results in LP/LR F_1 (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)

	LP	LR	F_1	Exact	CB
Model 1 (this paper)	84.8	84.4	84.6	26.4	1.37
Magerman (1995)	84.9	84.6			1.26
Model 2 (this paper)	85.7	85.7	85.7	29.3	1.29
Collins (1996)	86.3	85.8			1.14
Matsuzaki etal. (2005)	86.6	86.7			1.19
Klein and Manning (2003)	86.9	85.7	86.3	30.9	1.10
Charniak (1997)	87.4	87.5			1.00
Collins (1997)	88.6	88.1			0.91

Comparison with other parsers (sentences of length ≤ 40)

Discussion (3/4)

- 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