CMPT-413

Computational Linguistics

Anoop Sarkar

`http://www.cs.sfu.ca/~anoop`

Learning to Parse: History-based models

- Parsing can be framed as a supervised learning task
- Induce function $f : \mathcal{S} \rightarrow \mathcal{T}$ given $S_i \in \mathcal{S}$, pick best T_i from $\mathcal{T}(S)$
- Statistical parser builds model $P(T, S)$ for each (T, S)
- The best parse is then $\underset{T \in \mathcal{T}(S)}{\text{arg max}} \; P(T, S)$

History-based models and PCFGs

- History-based approaches maps (T, S) into a decision sequence $d_1, \ldots, d_n$

- Probability of tree T for sentence S is:

$$P(T, S) = \prod_{i=1 \ldots n} P(d_i \mid \phi(d_1, \ldots, d_{i-1}))$$

- ϕ is a function that groups histories into equivalence classes

History-based models and PCFGs

- PCFGs can be viewed as a history-based model using leftmost derivations

- A tree with rules $\langle \gamma_i \rightarrow \beta_i \rangle$ is assigned a probability $\prod_{i=1}^{n} P(\beta_i \mid \gamma_i)$ for a derivation with n rule applications

Generative models and PCFGs

$$
\begin{aligned}
T_{best} &= \underset{T}{\text{arg max}}\; P(T \mid S) \\
&= \underset{T}{\text{arg max}}\; \frac{P(T, S)}{P(S)} \\
&= \underset{T}{\text{arg max}}\; P(T, S) \\
&= \prod_{i=1 \ldots n} P(RHS_i \mid LHS_i)
\end{aligned}
$$

Evaluation of Statistical Parsers: EVALB

$$\text{Bracketing recall} = \frac{\text{num of correct constituents}}{\text{num of constituents in the goldfile}}$$

$$\text{Bracketing precision} = \frac{\text{num of correct constituents}}{\text{num of constituents in the parsed file}}$$

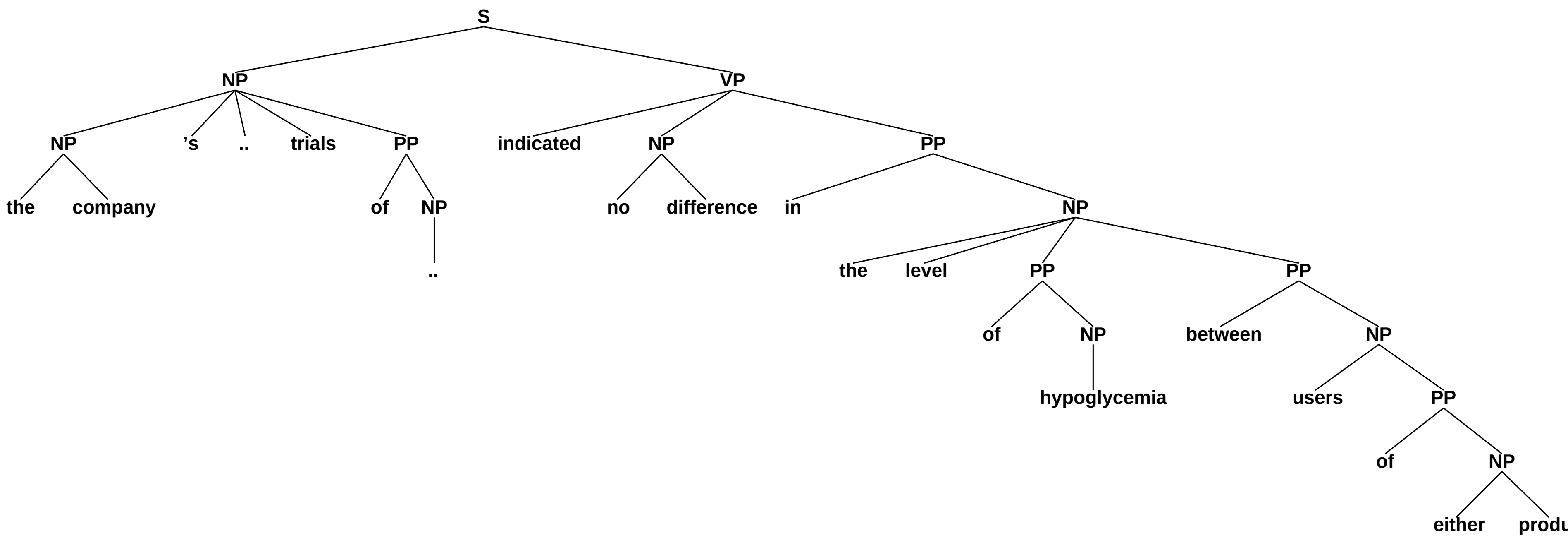
$$\text{Complete match} = \text{\% of sents where recall \& precision are both 100\%}$$

$$\text{Average crossing} = \frac{\text{num of constituents crossing a goldfile constituent}}{\text{num of sents}}$$

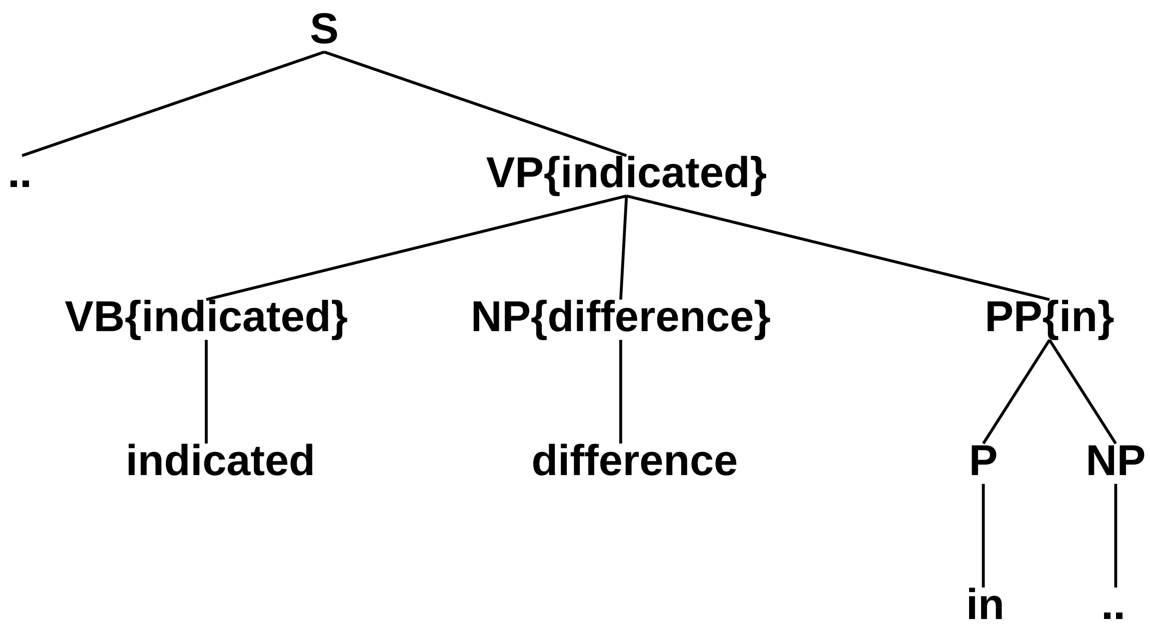
$$\text{No crossing} = \text{\% of sents which have } 0 \text{ crossing brackets}$$

$$\text{2 or less crossing} = \text{\% of sents which have } \leq 2 \text{ crossing brackets}$$

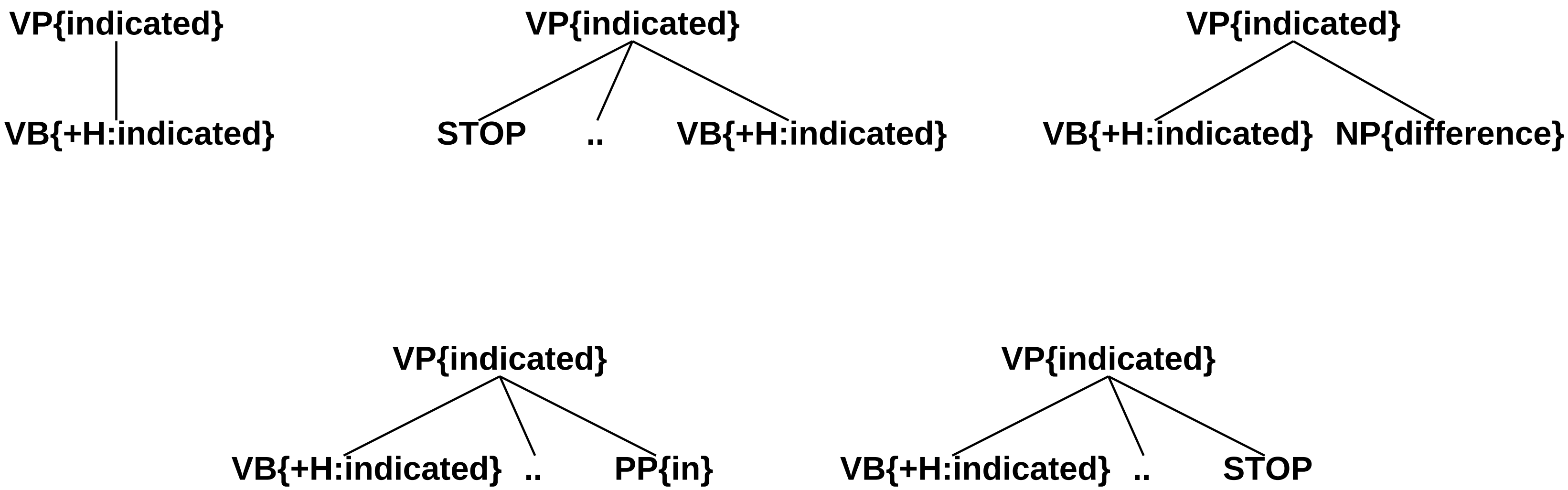
Statistical Parsing and PCFGs



Bilexical CFG: (Collins 1997)



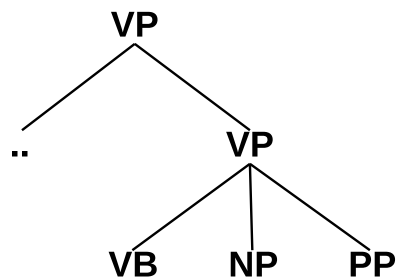
Bilexical CFG: VP{indicate} $\rightarrow$ VB{+H:indicate} NP{difference} PP{in}



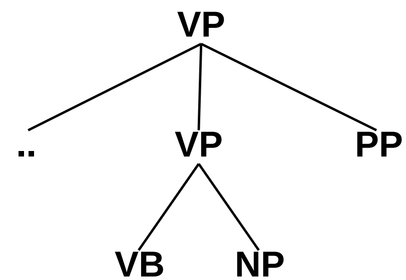
$$P_h(\texttt{VB} \mid \texttt{VP}, \texttt{indicated}) \times P_l(\texttt{STOP} \mid \texttt{VP}, \texttt{VB}, \texttt{indicated}) \times$$
$$P_r(\texttt{NP(difference)} \mid \texttt{VP}, \texttt{VB}, \texttt{indicated}) \times$$
$$P_r(\texttt{PP(in)} \mid \texttt{VP}, \texttt{VB}, \texttt{indicated}) \times$$
$$P_r(\texttt{STOP} \mid \texttt{VP}, \texttt{VB}, \texttt{indicated})$$

Independence Assumptions

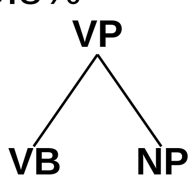
2.23%



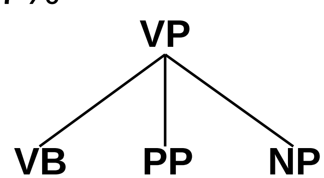
0.06%



60.8%



0.7%



Independence Assumptions

- Also violated in cases of coordination.

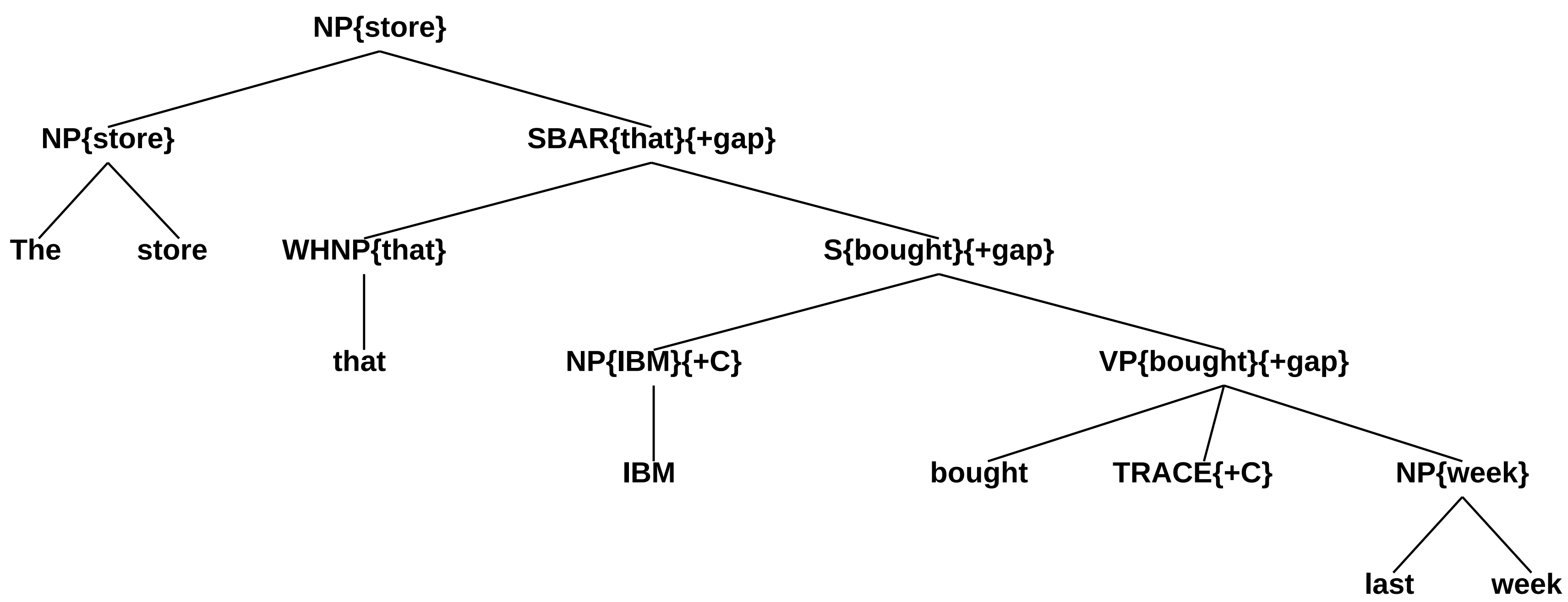
  e.g. `NP and NP`; `VP and VP`

- Processing facts like attach low in general.

- Also, English parse trees are generally right branching due to SVO structure.

- Language specific features are used heavily in the statistical model for parsing: cf. (Haruno et al. 1999)

Bilexical CFG with attributes (Collins 1997)



NP	$\longrightarrow$	[	NP{+H}	SBAR{+gap}		]
SBAR{+gap}	$\longrightarrow$	[	WHNP	S{+H}{+C}{+gap}		]
S{+gap}	$\longrightarrow$	[	NP{+C}	SBAR{+H}{+gap}		]
VP{+gap}	$\longrightarrow$	[	VB{+H}	TRACE{+C}	NP	]

Parsing Results using Lexicalized PCFGs

System	$\leq 40wds$ LP	$\leq 40wds$ LR	$\leq 100wds$ LP	$\leq 100wds$ LR
(Magerman 95)	84.9	84.6	84.3	84.0
(Collins 99)	88.5	88.7	88.1	88.3
(Charniak 97)	87.5	87.4	86.7	86.6
(Ratnaparkhi 97)			86.3	87.5
(Charniak 99)	90.1	90.1	89.6	89.5
(Collins 00)	90.1	90.4	89.6	89.9
(Shen, Sarkar, Joshi 03)	90.2	90.5	89.7	90.0

Tree Adjoining Grammars

- Locality and independence assumptions are captured elegantly.

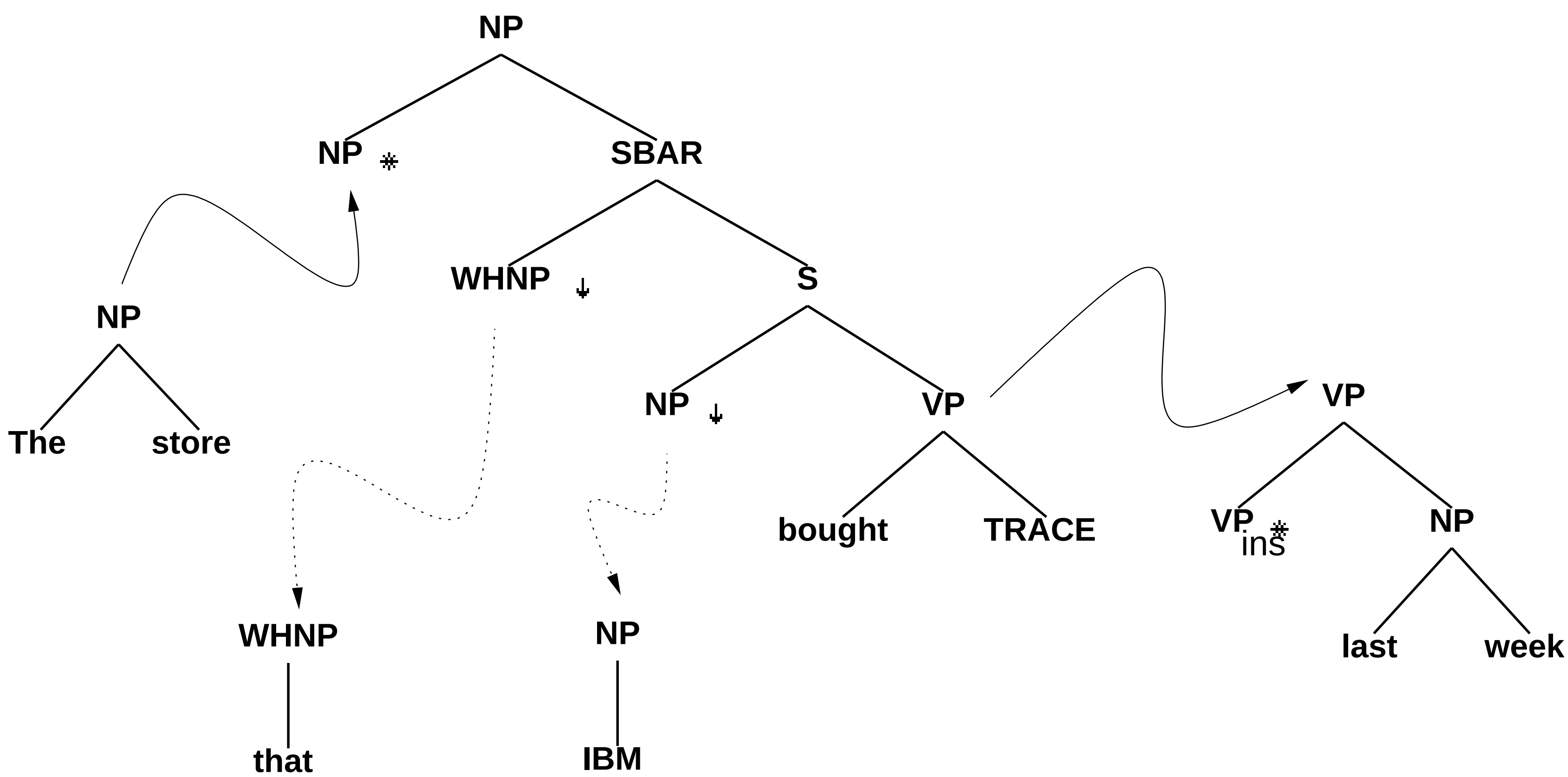
- Simple and well-defined probability model.

- Parsing can be treated in two steps:

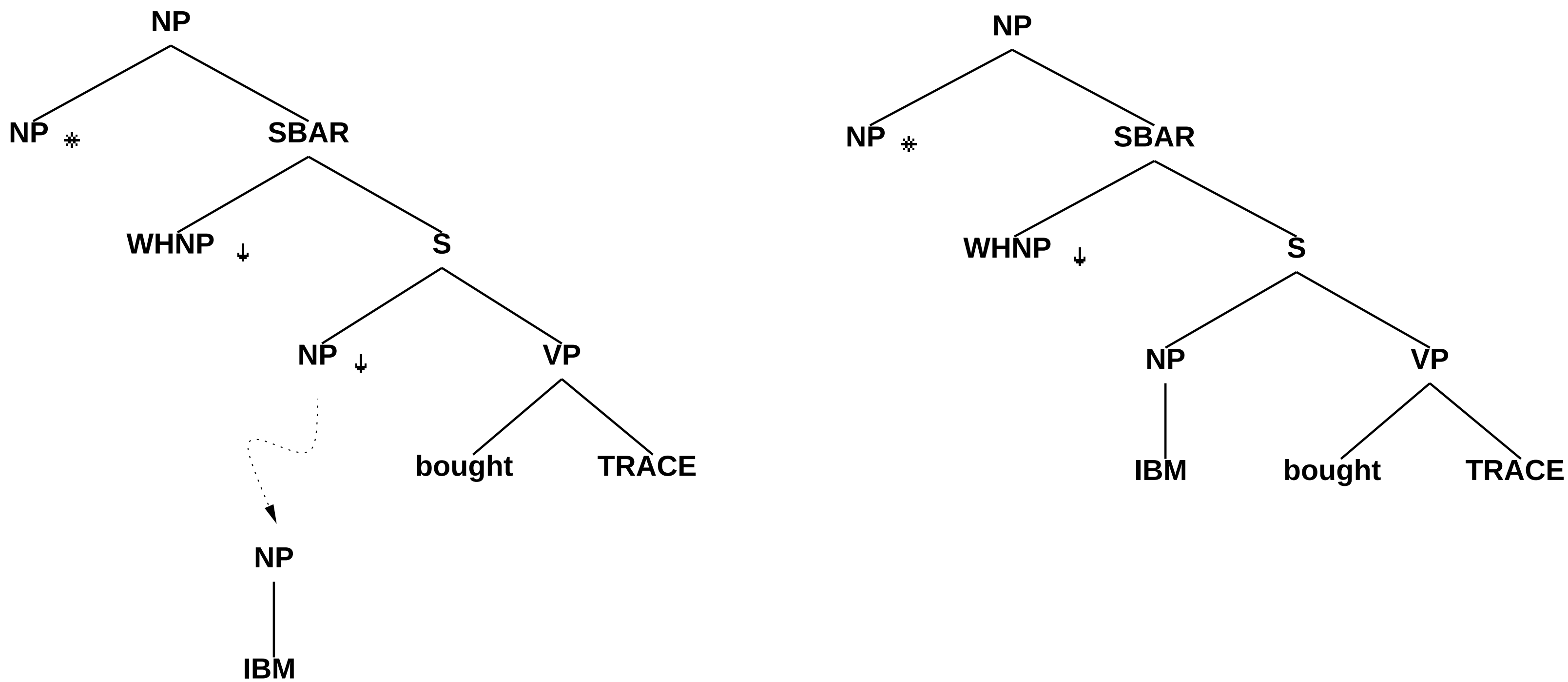
  1. Classification: structured labels (elementary trees) are assigned to each word in the sentence.

  2. Attachment: the elementary trees are connected to each other to form the parse.

Tree Adjoining Grammars: Different Bilexical Grammar

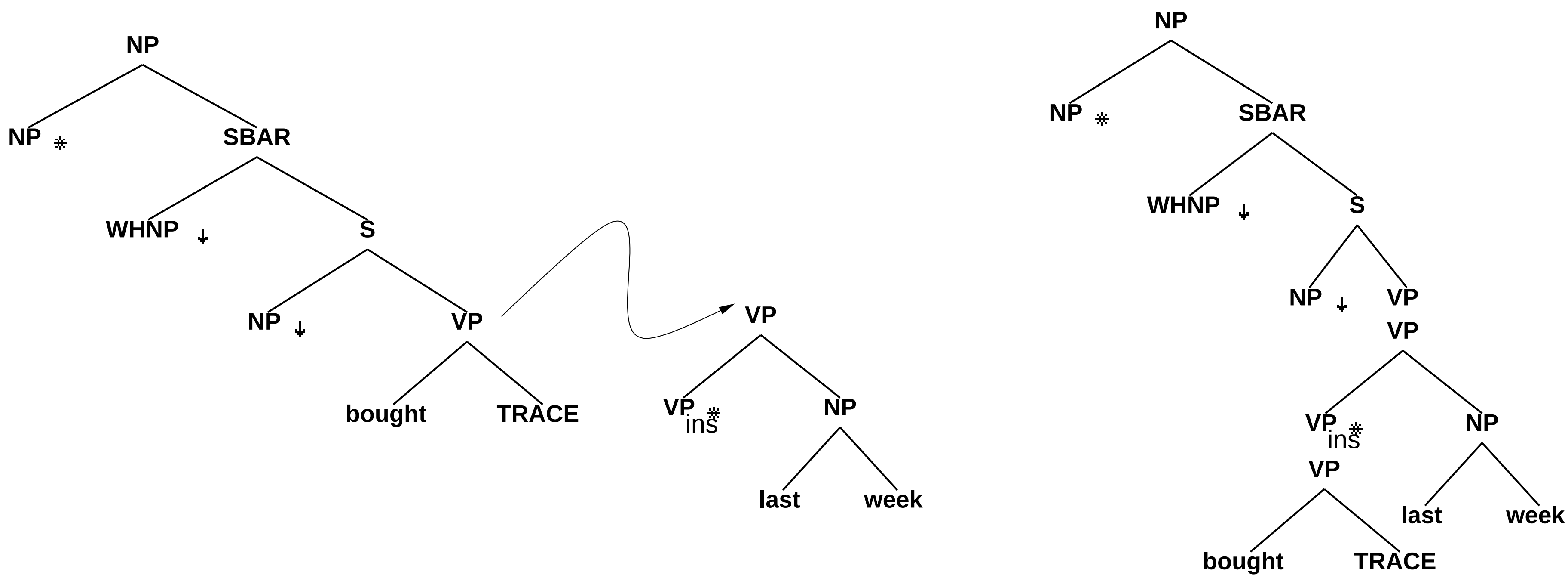


Probabilistic TAGs: Substitution



$$\sum_{t'} \mathcal{P}(t, \eta \rightarrow t') = 1$$

Probabilistic TAGs: Adjunction



$$\mathcal{P}(t, \eta \rightarrow NA) + \sum_{t'} \mathcal{P}(t, \eta \rightarrow t') = 1$$

Tree Adjoining Grammars

- Simpler model for parsing.
  Performance(Chiang 2000): 86.9% LR 86.6% LP ($\leq$ 40 words)
  Latest results: $\approx$ 88% average LP/LR

- Parsing can be treated in two steps:

  1. Classification: structured labels (elementary trees) are assigned to each word in the sentence.

  2. Attachment: Apply substitution or adjunction to combine the elementary trees to form the parse.

Practical Issues: Beam Thresholding and Priors

- Probability of nonterminal X spanning $j \ldots k$: $N[X, j, k]$
- Beam Thresholding compares $N[X, j, k]$ with every other Y where $N[Y, j, k]$
- But what should be compared?
- Just the *inside probability*: $P(X \stackrel{*}{\Rightarrow} t_j \ldots t_k)$?
  written as $\beta(X, j, k)$
- Perhaps $\beta(\texttt{FRAG}, 0, 3) > \beta(\texttt{NP}, 0, 3)$, but NPs are much more likely than FRAGs in general

Practical Issues: Beam Thresholding and Priors

- The correct estimate is the *outside probability*:

$$P(S \stackrel{*}{\Rightarrow} t_1 \ldots t_{j-1}\ X\ t_{k+1} \ldots t_n)$$

  written as $\alpha(X, j, k)$

- Unfortunately, you can only compute $\alpha(X, j, k)$ efficiently after you finish parsing and reach $(S, 0, n)$

Practical Issues: Beam Thresholding and Priors

- To make things easier we multiply the prior probability $P(X)$ with the inside probability

- In beam Thresholding we compare every new insertion of X for span j, k as follows:
  Compare $P(X) \cdot \beta(X, j, k)$ with every Y $P(Y) \cdot \beta(Y, j, k)$