Language and Statistics II

Lecture 11: Modern Parsers

Noah Smith

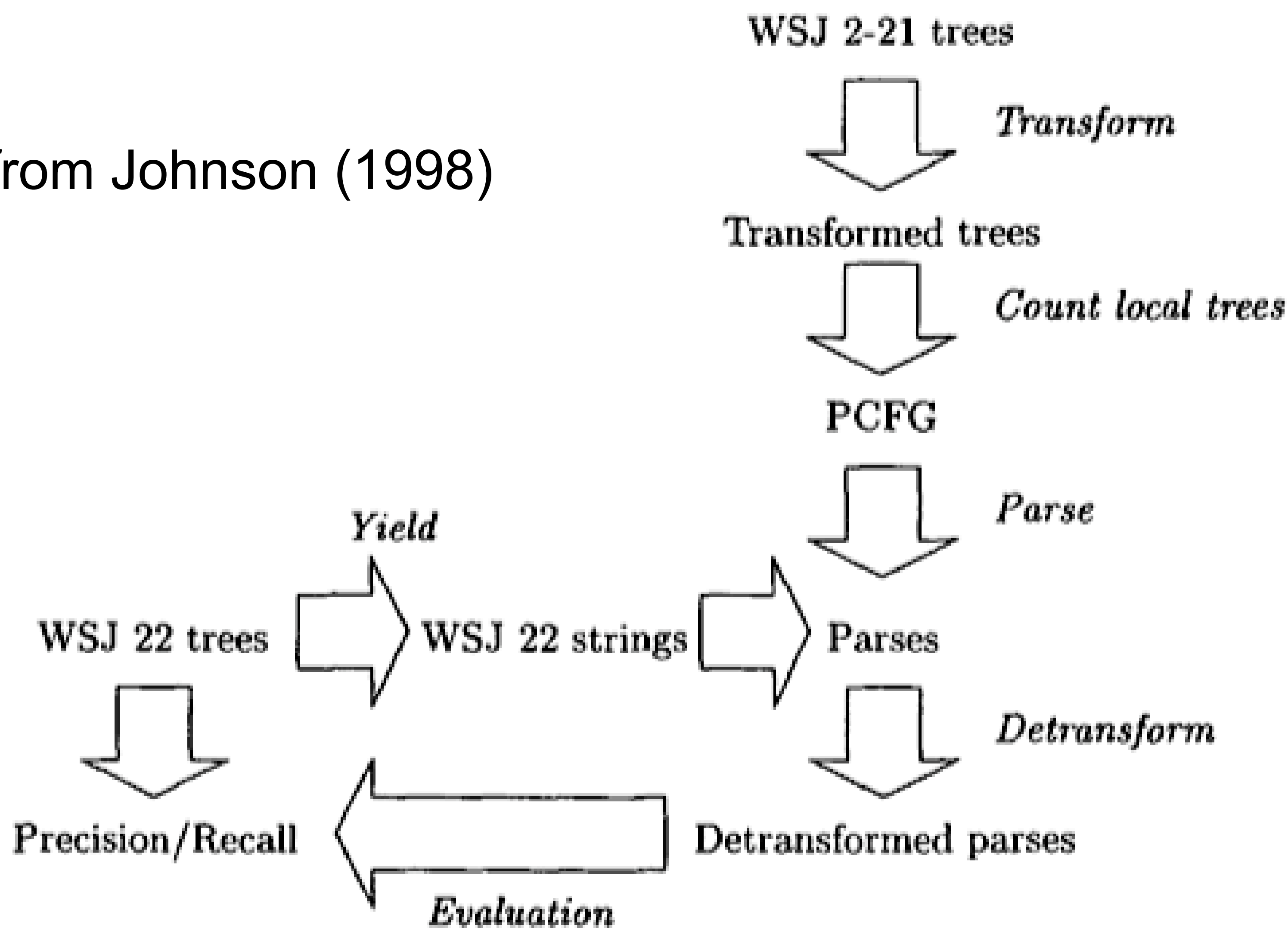
Last Time

- Vanilla PCFGs
- Treebanks
- Parsing Algorithms for PCFGs

Today

- Some useful transformations on trees
- Modern parsing models:
  - Collins (1997; 2003)
  - Charniak (1997; 2000)

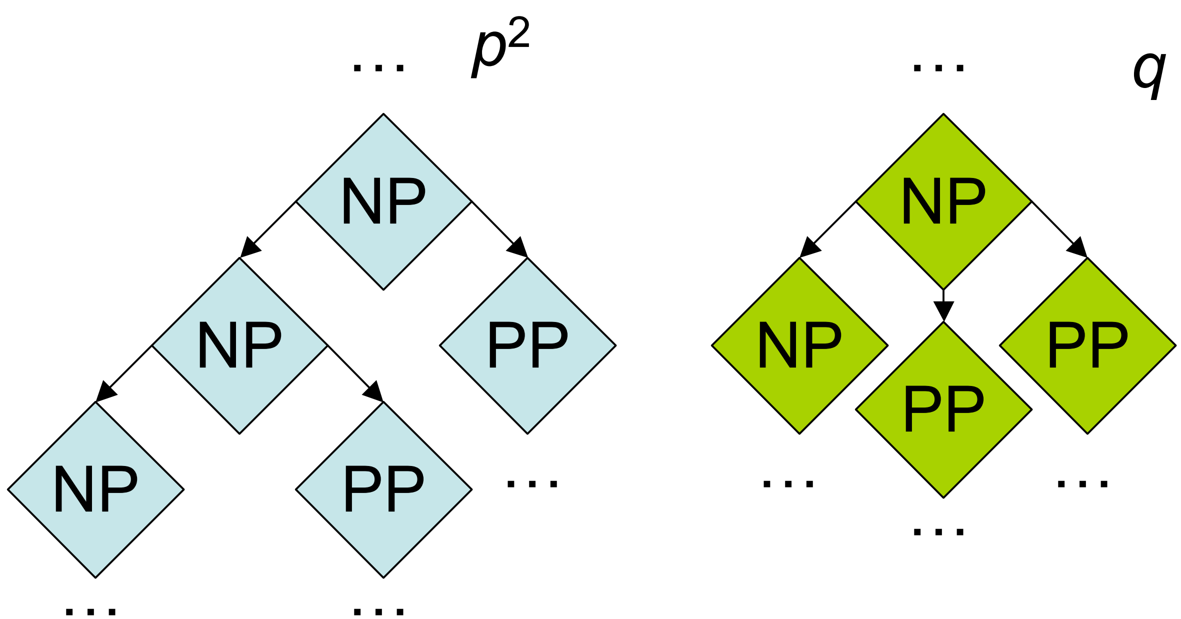
from Johnson (1998)



Parent Annotation

$NP \xrightarrow{p} NP\ PP$

$NP \xrightarrow{q} NP\ PP\ PP$

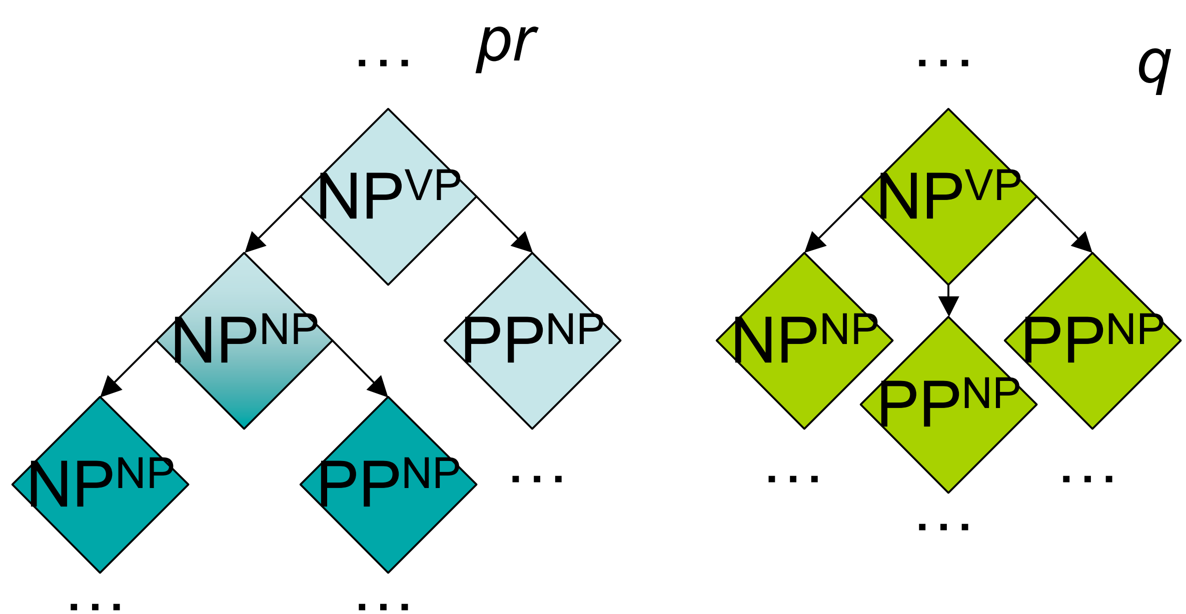


Parent Annotation

$NP^{VP} \rightarrow_p NP^{NP}\ PP^{NP}$

$NP^{NP} \rightarrow_r NP^{NP}\ PP^{NP}$

$NP^{VP} \rightarrow_q NP^{NP}\ PP^{NP}\ PP^{NP}$



Parent Annotation

- Another way to think about it …

Before: $$p(\text{tree}) = \prod_{n \in \text{tree's nonterminal tokens}} \rho(n\text{'s children} \mid n)$$

Now: $$p(\text{tree}) = \prod_{n \in \text{tree's nonterminal tokens}} \rho(n\text{'s children} \mid n, n\text{'s parent})$$

- This could conceivably **help** performance (weaker independence assumptions)
- This could conceivably **hurt** performance (data sparseness)

Parent Annotation

- From Johnson (1998):

  PCFG from WSJ Treebank: 14,962 rules
    - Of those, 1,327 would **always** be subsumed!

  After parent annotation: 22,773 rules
    - Only 965 would always be subsumed!

  Recall 69.7% → 79.2%; precision 73.5% → 80.0%

- Trick: check for subsumed rules, remove them from the grammar → faster parsing.

Head Annotation

"I love all my children, but one of them is **special**."

S $\rightarrow$ NP VP

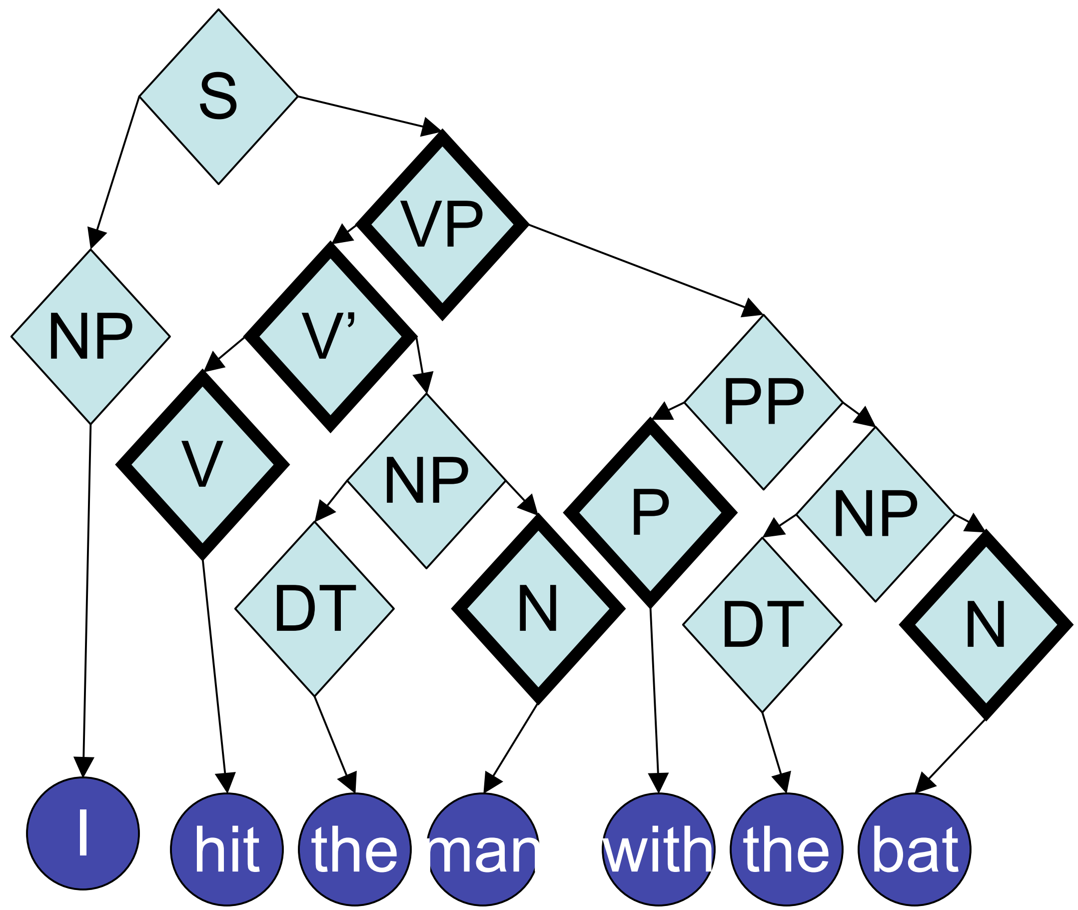
VP $\rightarrow$ VBD NP

NP $\rightarrow$ DT NNS PP

Heads not in the Treebank.

Usually people use **deterministic head rules** (Magerman, 1995).

Head Annotation

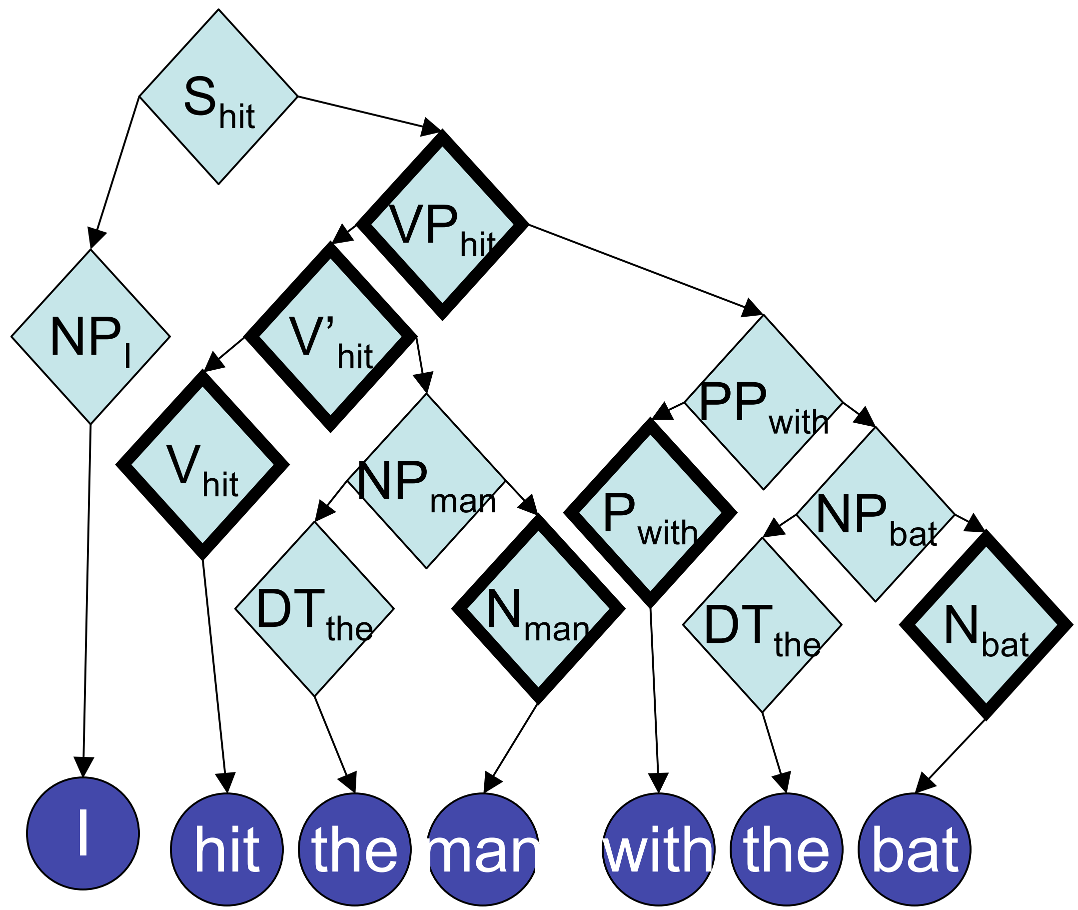


Lexicalization

- Every nonterminal node is annotated with a word from its yield; such that

$$lex(n) = lex(head(n))$$

Lexical Head Annotation



Lexicalization

- Every nonterminal node is annotated with a word from its yield; such that

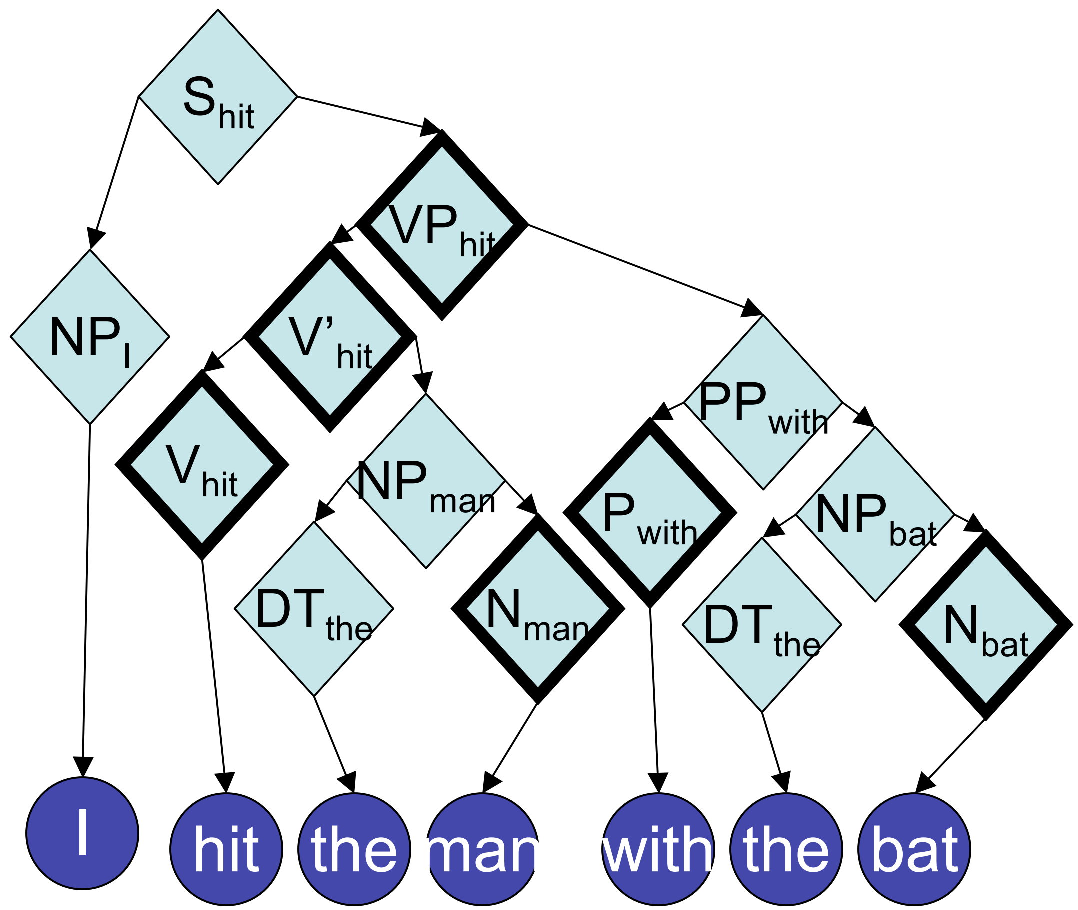
$$\text{lex}(n) = \text{lex}(\text{head}(n))$$

- What might this allow?
- What might we worry about?

Currently, this is controversial (we'll see why)!

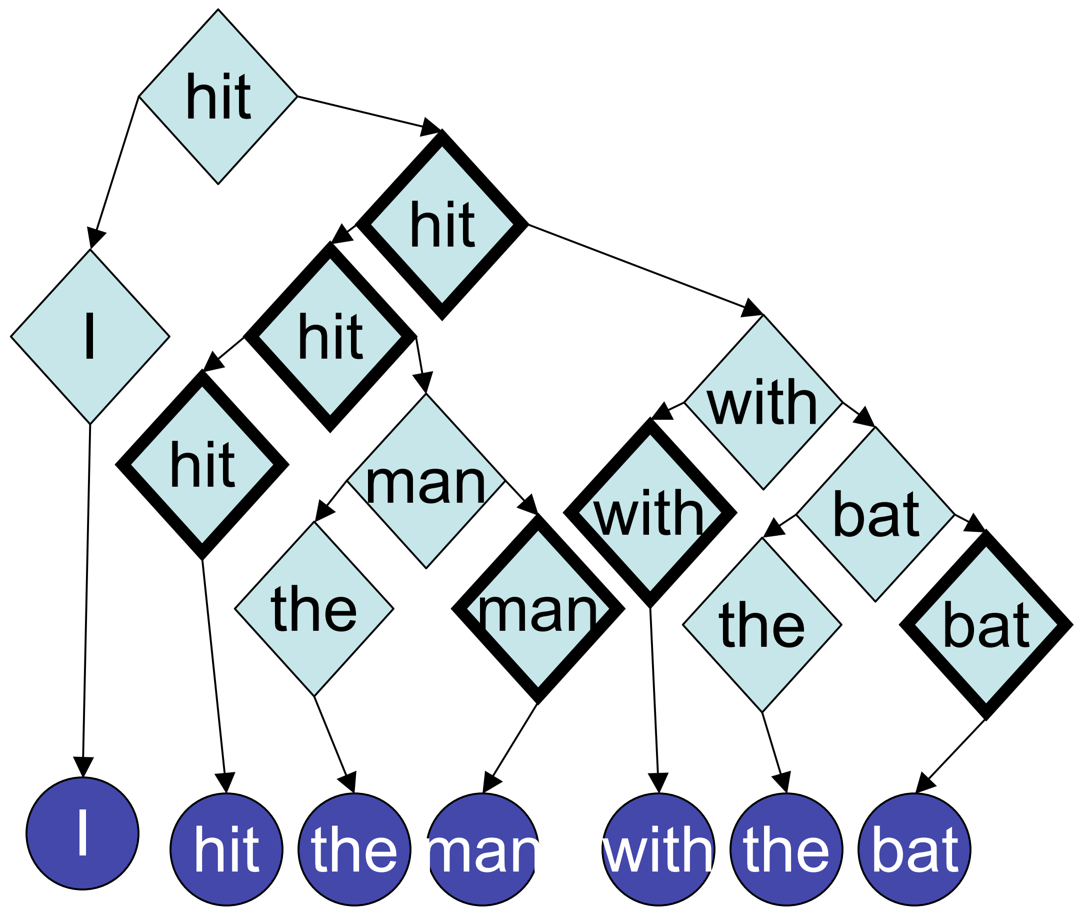
Dependencies

- Take away the nonlexical parts.



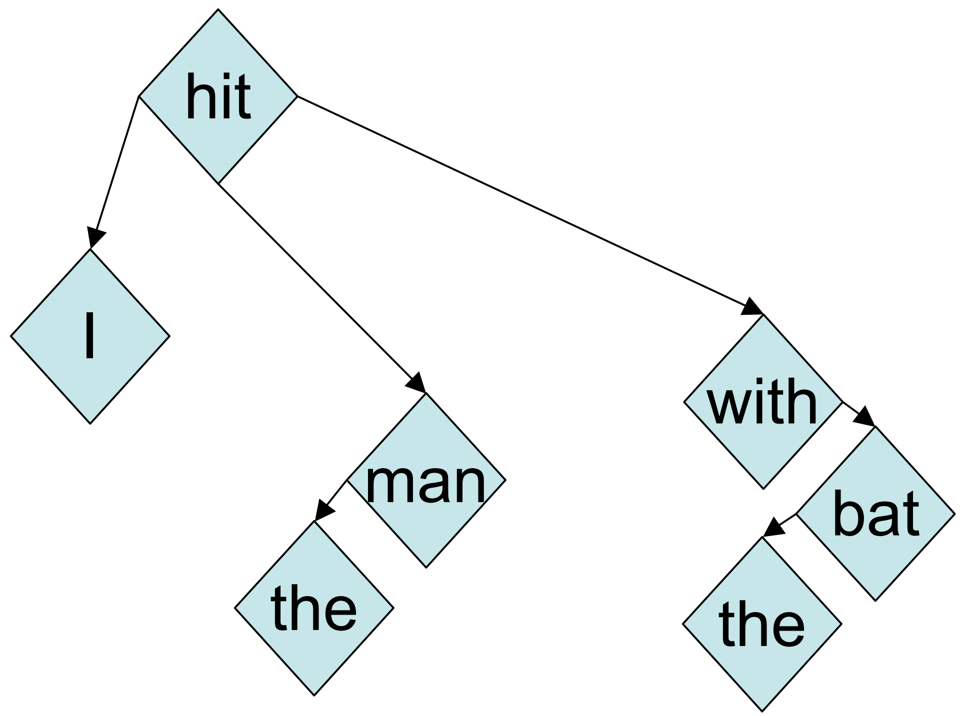
Dependencies

- Take away the nonlexical parts.



Dependencies

- Merge redundant nodes upward.

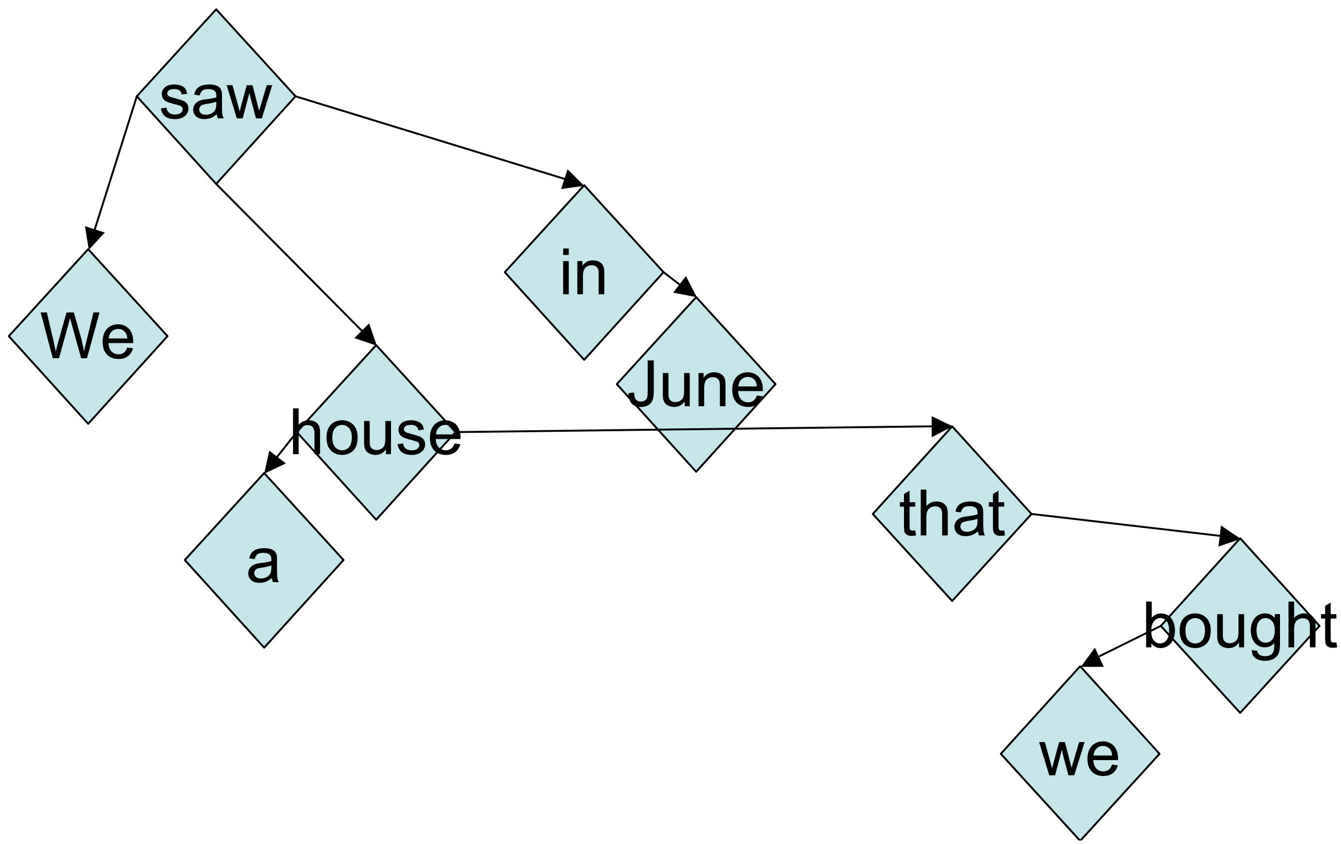


Crucial Point

- By "decorating" the treebank, we have been carrying additional information around the trees.
- The **hope** is to improve the ability of a PCFG to predict syntactic structure correctly.
- The **worry** is that our grammar will get really big and the probabilities too hard to estimate.
  - Also, speed. More rules → bigger grammar → slower parsing.

Dependencies

- Can represent some things that are hard for CFGs (but then it's not a PCFG anymore):



Dependencies

- Don’t have to be lexicalized
- Often faster to parse
- Closer to semantics?
- We’ll come back to this representation.

Collins Model 1 (1997)

- Trees are headed & lexicalized.
- Many, many rules!

$VP_{saw} \rightarrow \underline{V}_{saw}\ NP_{man}\ PP_{through}$

$VP_{saw} \rightarrow \underline{V}_{saw}\ NP_{man}\ PP_{with}$

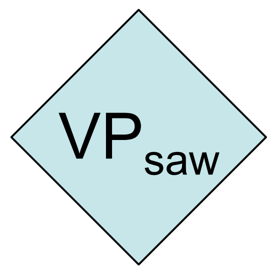
$VP_{saw} \rightarrow \underline{V}_{saw}\ NP_{woman}\ PP_{through}$

$VP_{saw} \rightarrow \underline{V}_{saw}\ NP_{man}$

…

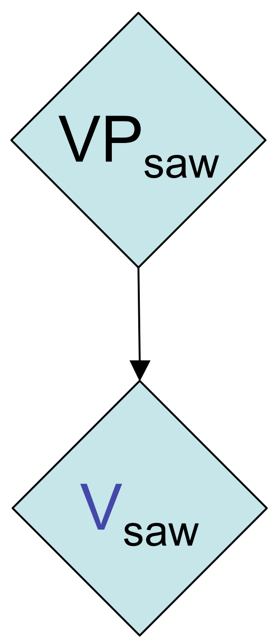
Collins Model 1 (1997)

- We are given the parent and its lexeme.



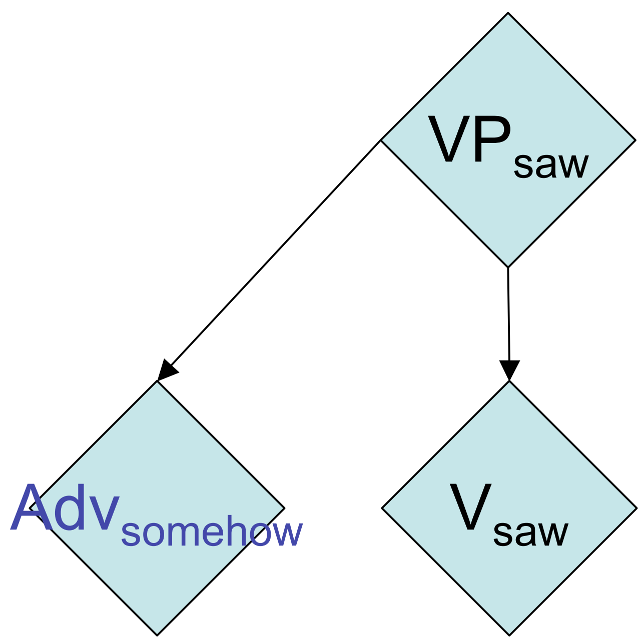
Collins Model 1 (1997)

- We are given the parent and its lexeme.
- Randomly generate the head nonterminal.



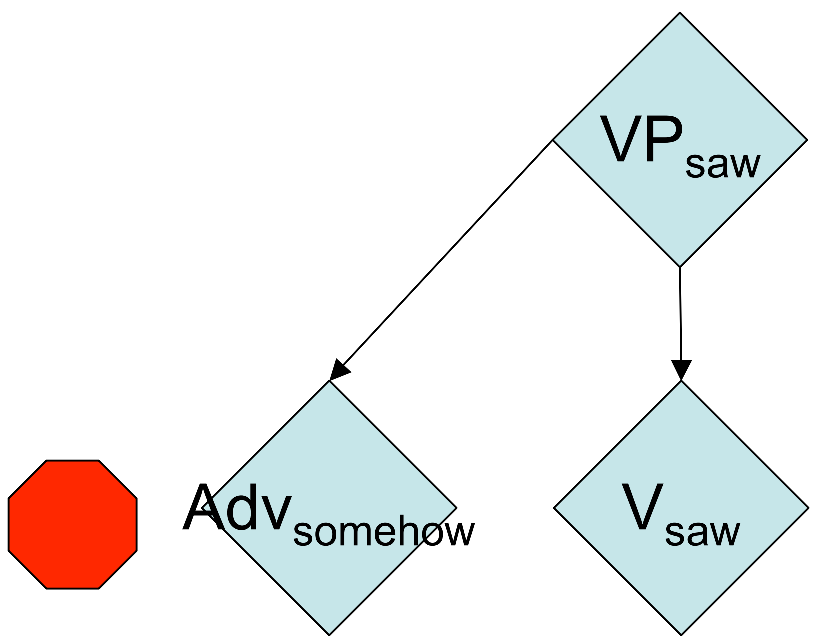
Collins Model 1 (1997)

- We are given the parent and its lexeme.
- Randomly generate the head nonterminal.
- Generate a sequence of left children.



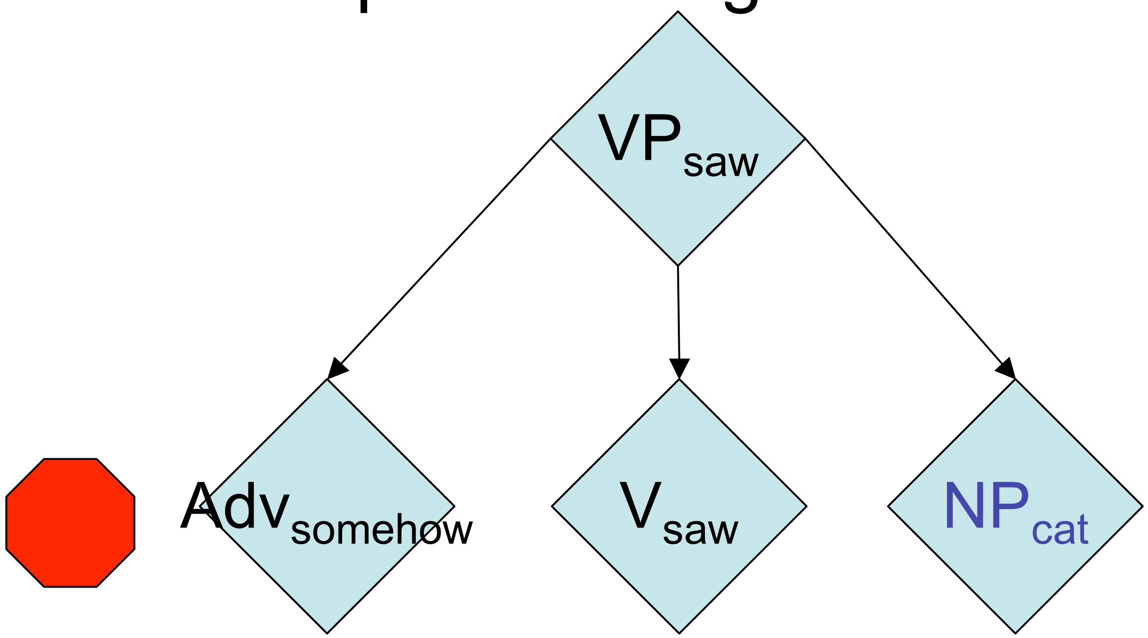
Collins Model 1 (1997)

- We are given the parent and its lexeme.
- Randomly generate the head nonterminal.
- Generate a sequence of left children.



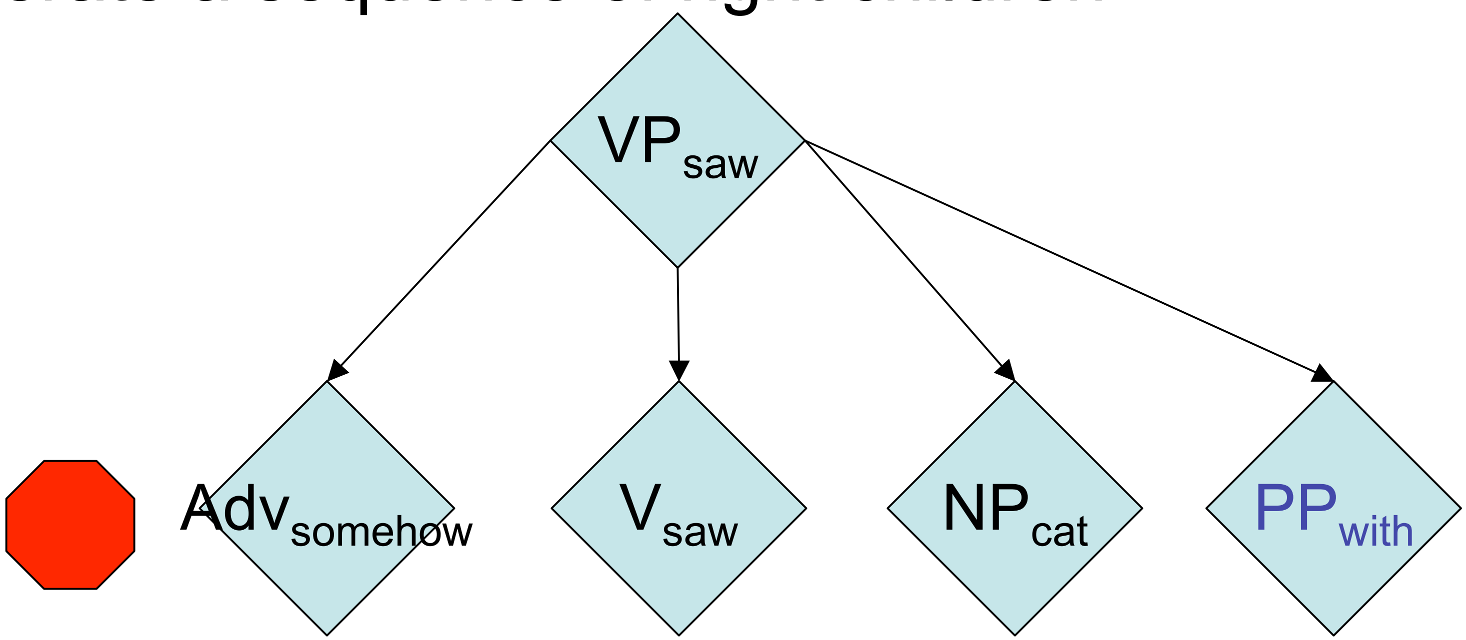
Collins Model 1 (1997)

- We are given the parent and its lexeme.
- Randomly generate the head nonterminal.
- Generate a sequence of left children.
- Generate a sequence of right children.



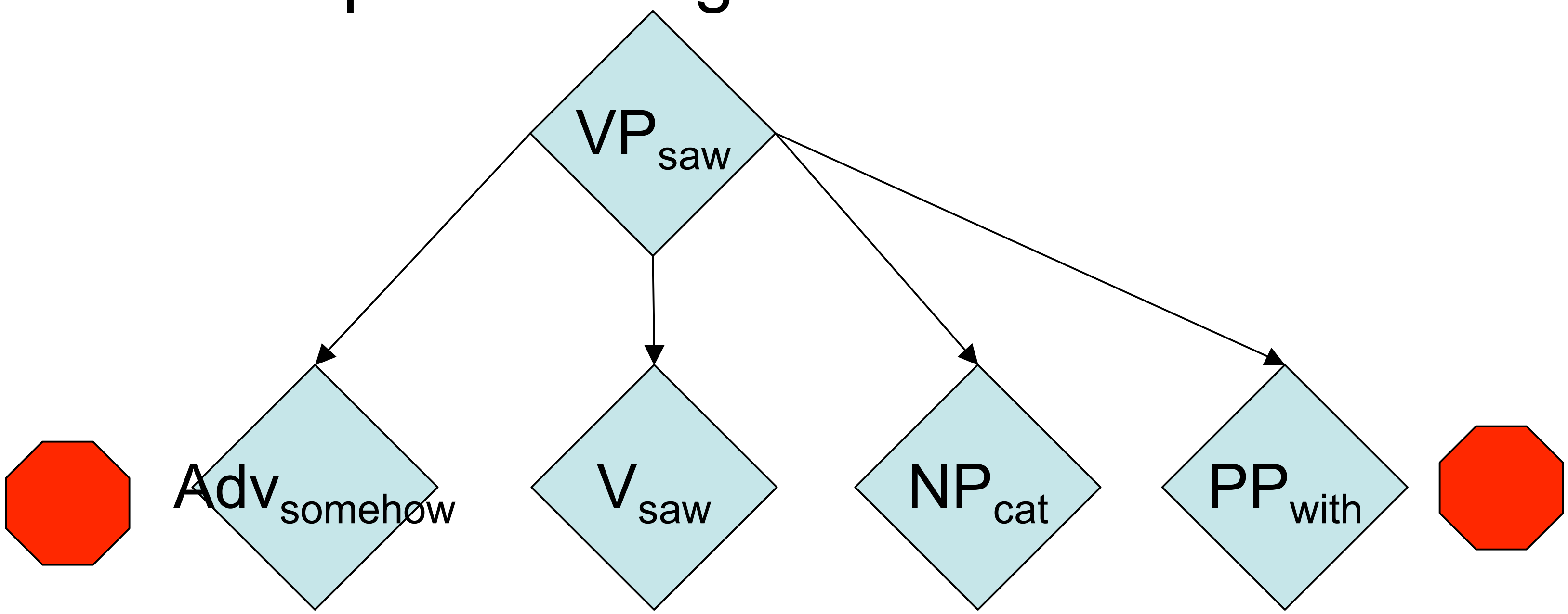
Collins Model 1 (1997)

- We are given the parent and its lexeme.
- Randomly generate the head nonterminal.
- Generate a sequence of left children.
- Generate a sequence of right children.



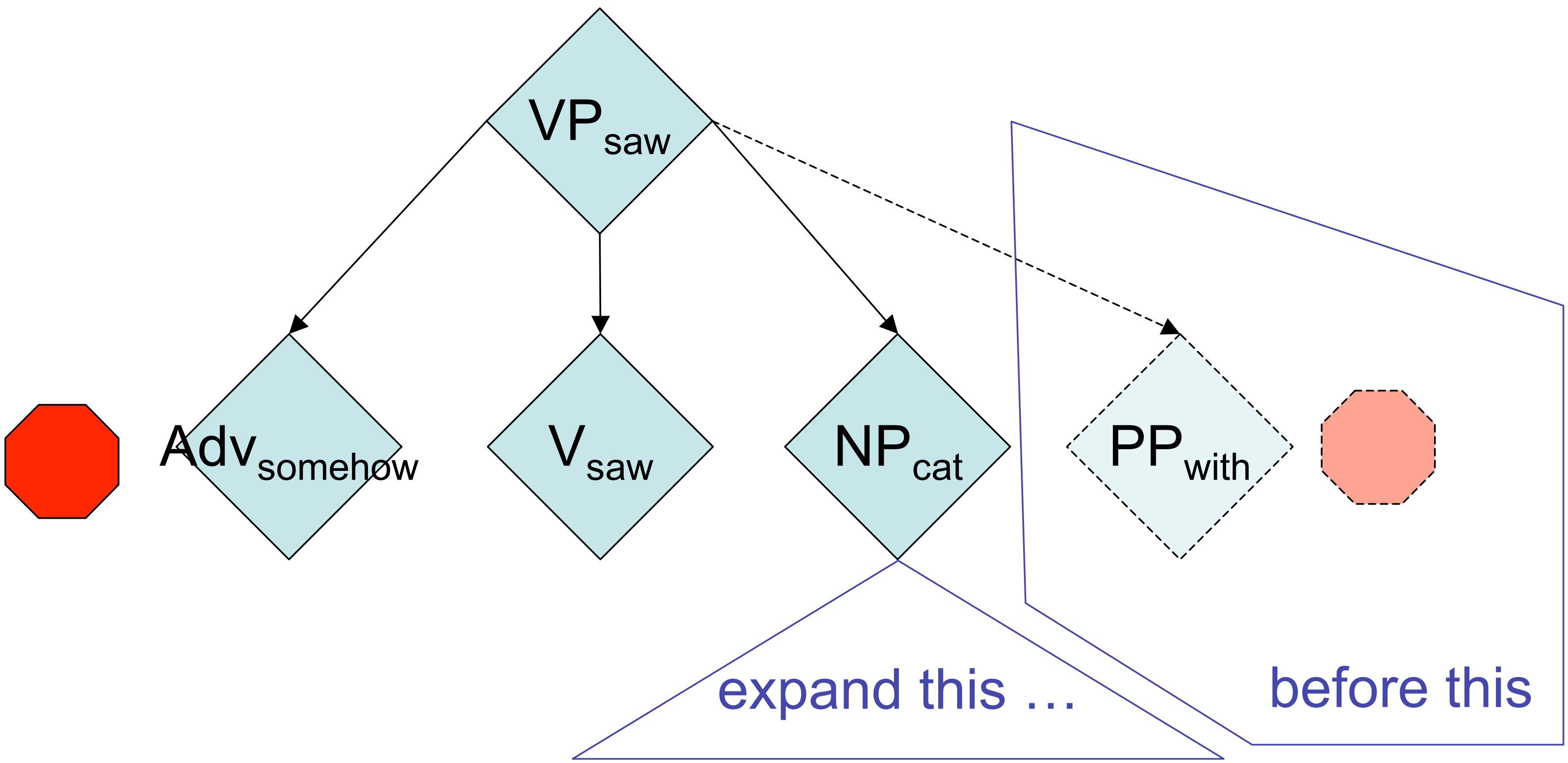
Collins Model 1 (1997)

- We are given the parent and its lexeme.
- Randomly generate the head nonterminal.
- Generate a sequence of left children.
- Generate a sequence of right children.



Collins Model 1 (1997)

- Wanted to model **distance**. How?
- Assume depth-first recursion.



Collins Model 1 (1997)

- Wanted to model **distance**. How?
- Assume depth-first recursion.
- Can then condition the next child on (features of) the yield between it and the head:

$$p(PP_{with} \mid VP_{saw}, \text{right}, \text{“the cat who liked milk”})$$
$$\approx p(PP_{with} \mid VP_{saw}, \text{right}, \text{length}>0, \text{+verb})$$

- 1997 version looked for commas, too; later this was removed.

Collins Model 1 (1997)

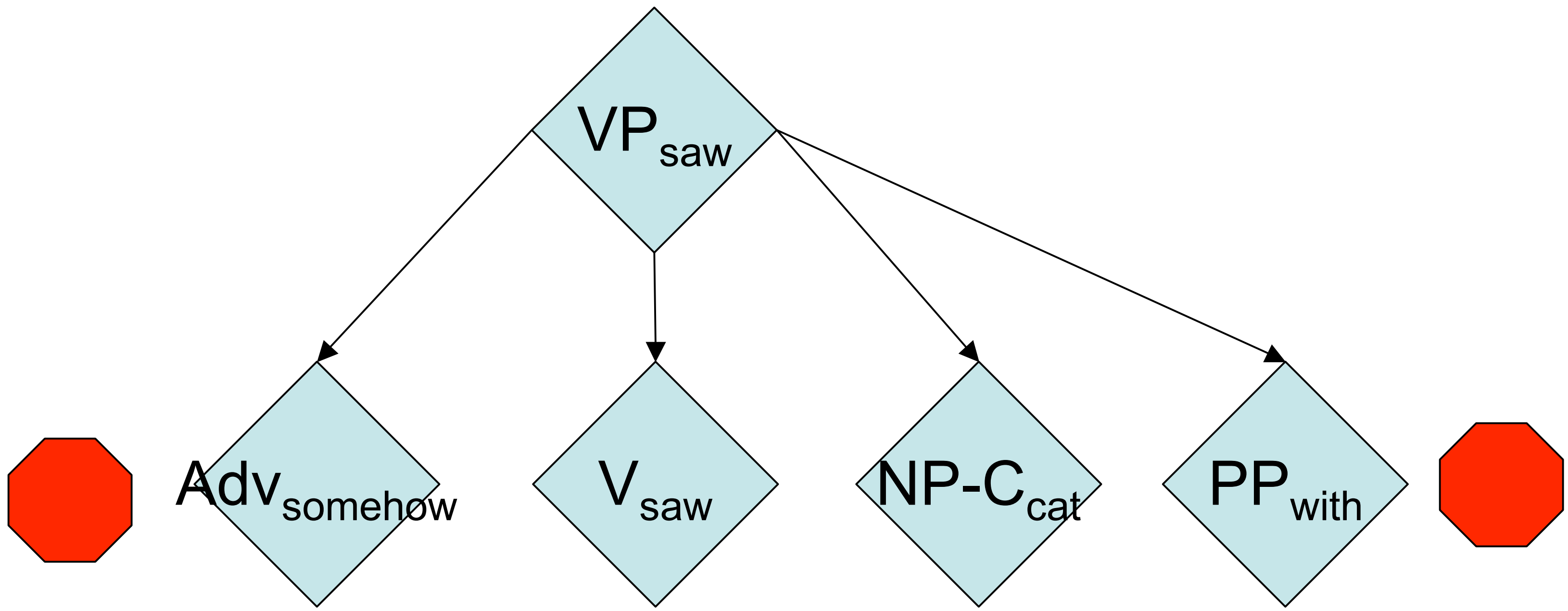
$$p\left(\langle L,u\rangle_1^n,\langle H,w\rangle,\langle R,v\rangle_1^m \middle| \langle P,w\rangle\right) =$$

$$p\left(H \middle| \langle P,w\rangle\right)\cdot\left(\prod_{i=1}^{n} p\left(\langle L,u\rangle_i \middle| \langle P,w\rangle,H,\text{left},\Delta_i\right)\right) p\left(\text{stop} \middle| \langle P,w\rangle,H,\text{left},\Delta_{n+1}\right)$$

$$\cdot\left(\prod_{i=1}^{m} p\left(\langle R,v\rangle_i \middle| \langle P,w\rangle,H,\text{right},\Delta_i\right)\right) p\left(\text{stop} \middle| \langle P,w\rangle,H,\text{right},\Delta_{m+1}\right)$$

Collins Models 2 & 3 (1997)

(blackboard)



Other Details

- Smoothing: deleted interpolation.
- Unknown words: every type with count ≤ 5 became UNK
- Tagging is not a separate stage; it is just part of the parse.

Further Refinements

- Base noun phrases
  - Labeled “NPB”
  - First-order Markov model for children of head!
- Coordinators (“and”) predicted **together** with the later argument.
- Punctuation treated similarly (see the 2003 paper)

Charniak (1997)

- Similar setup.
  - Lexicalized PCFG, factored model for rules
  - Tags don’t travel up the tree as in Collins
  - Tagging part of parsing
  - Deleted interpolation for smoothing
- Used an additional 30 million words of unannotated data.

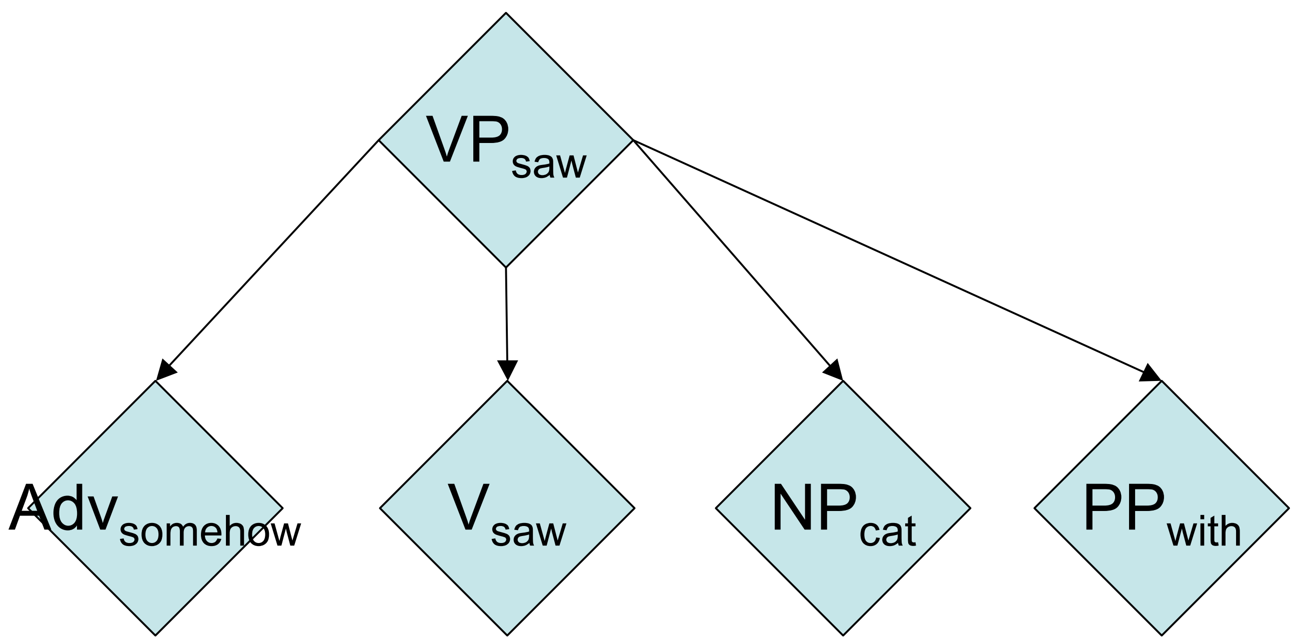
Charniak (1997)

p(Adv $\underline{V}_{saw}$ NP PP | VP_{saw}, S)

p(somehow | VP_{saw}, Adv)

p(cat | VP_{saw}, NP)

p(with | VP_{saw}, PP)



Charniak (2000)

- The 2000 parser is “maximum entropy inspired.”
- It is closer to Collins’ model (Markovized children), but the estimation is bizarre.
  - Smoothed, backed-off probabilities are multiplied together - almost like a **product of experts**.

Comparison

		labeled recall	labeled precision	average crossing brackets
Collins	Model 1	87.5	87.7	1.09
	Model 2	88.1	88.3	1.06
	Model 3	88.0	88.3	1.05
Charniak	1997	86.7	86.6	1.20
	2000	89.6	89.5	0.88