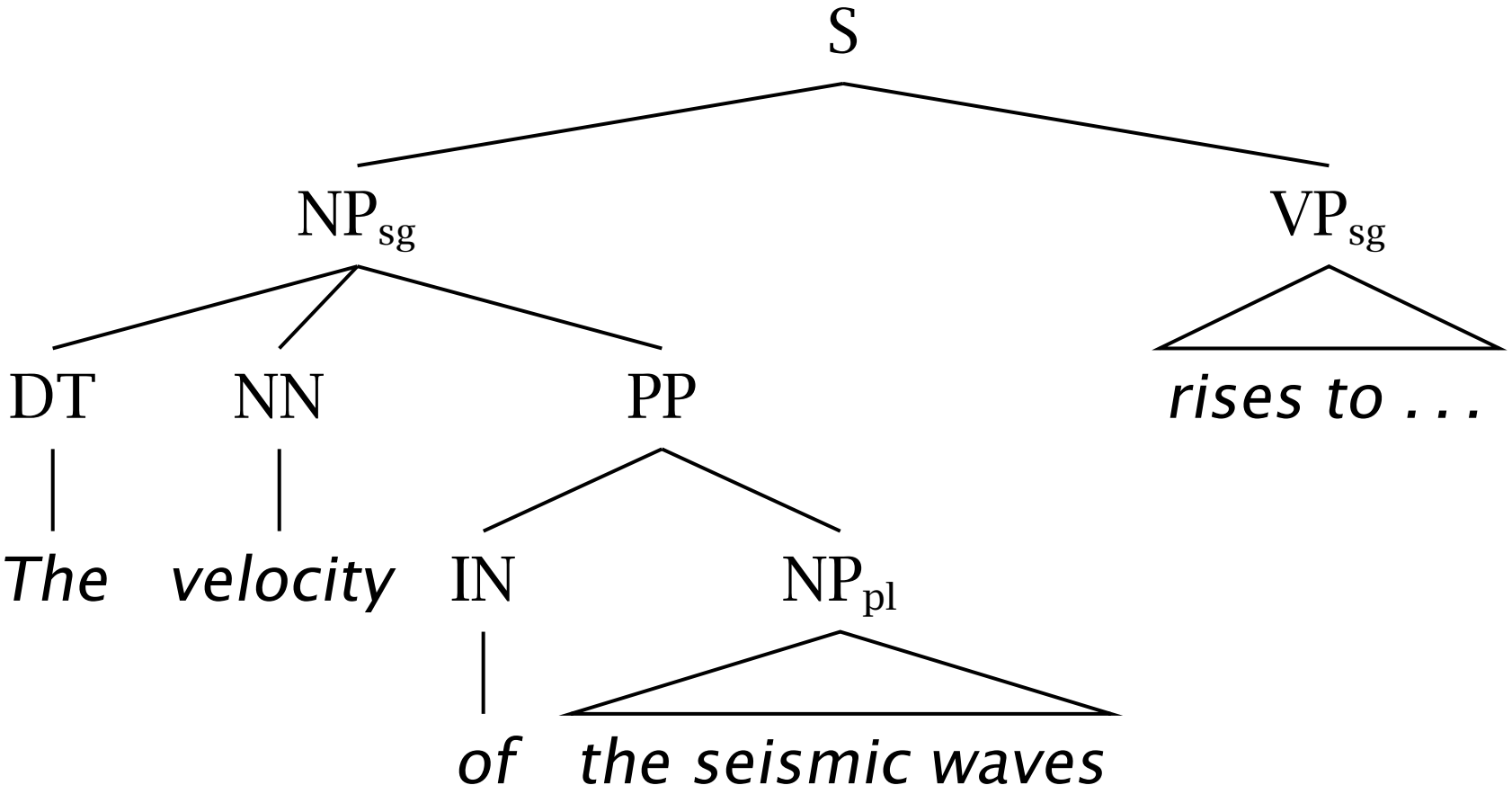
{Probabilistic|Stochastic} Context-Free Grammars (PCFGs)

- The velocity of the seismic waves rises to . . .



PCFGs

A PCFG G consists of:

- A set of terminals, $\{w^k\}, k = 1, \ldots, V$
- A set of nonterminals, $\{N^i\}, i = 1, \ldots, n$
- A designated start symbol, N^1
- A set of rules, $\{N^i \rightarrow \zeta^j\}$, (where ζ^j is a sequence of terminals and nonterminals)
- A corresponding set of probabilities on rules such that:

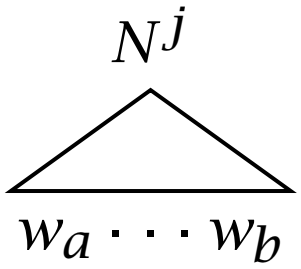
$$\forall i \quad \sum_j P(N^i \rightarrow \zeta^j) = 1$$

PCFG notation

Sentence: sequence of words $w_1 \cdots w_m$

w_{ab}: the subsequence $w_a \cdots w_b$

N^i_{ab}: nonterminal N^i dominates $w_a \cdots w_b$



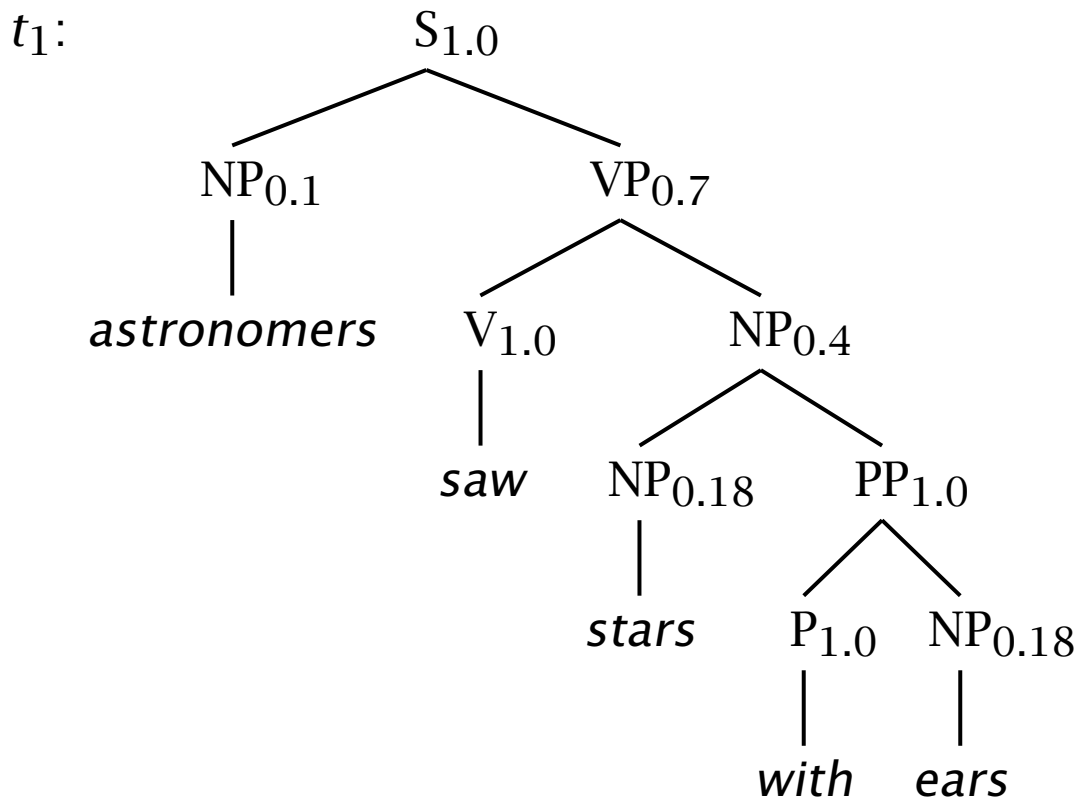
$N^i \overset{*}{\Longrightarrow} \zeta$: Repeated derivation from N^i gives ζ.

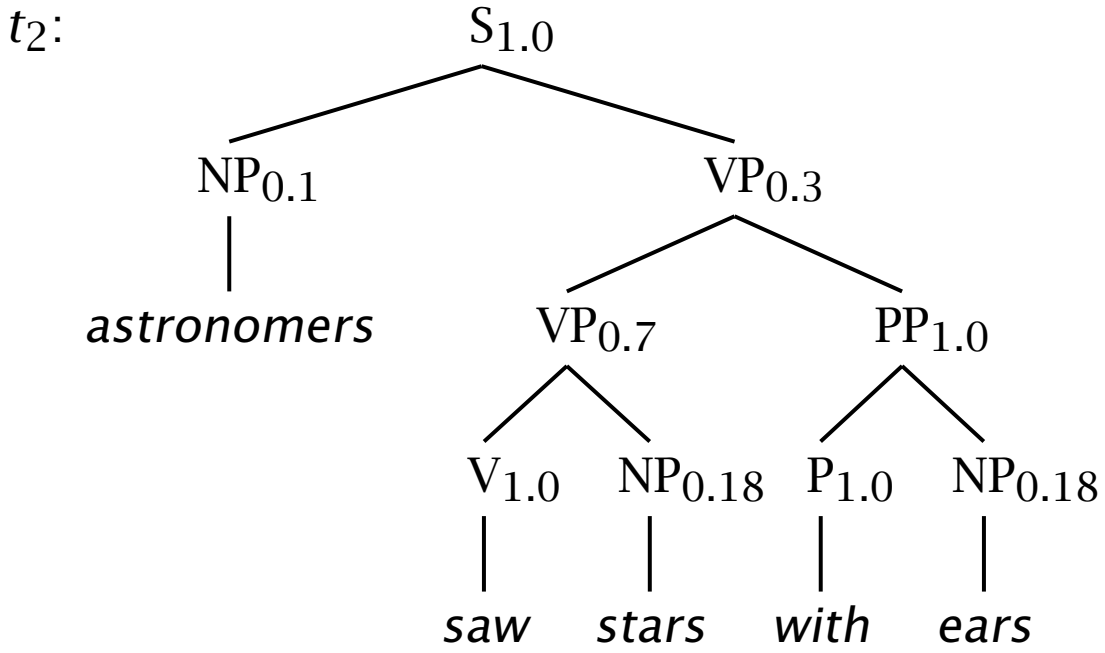
PCFG probability of a string

$$
\begin{aligned}
P(w_{1n}) &= \sum_{t} P(w_{1n}, t) \quad t \text{ a parse of } w_{1n} \\
&= \sum_{\{t:\text{yield(t)}=\text{w}_{1\text{n}}\}} P(t)
\end{aligned}
$$

A simple PCFG (in CNF)

S → NP VP	1.0	NP → NP PP	0.4
PP → P NP	1.0	NP → *astronomers*	0.1
VP → V NP	0.7	NP → *ears*	0.18
VP → VP PP	0.3	NP → *saw*	0.04
P → *with*	1.0	NP → *stars*	0.18
V → *saw*	1.0	NP → *telescopes*	0.1


t_1:
$S_{1.0}$
$NP_{0.1}$
$VP_{0.7}$
astronomers
$V_{1.0}$
$NP_{0.4}$
saw
$NP_{0.18}$
$PP_{1.0}$
stars
$P_{1.0}$
$NP_{0.18}$
with
ears


t_2:
$S_{1.0}$
$NP_{0.1}$
$VP_{0.3}$
astronomers
$VP_{0.7}$
$PP_{1.0}$
$V_{1.0}$
$NP_{0.18}$
$P_{1.0}$
$NP_{0.18}$
saw
stars
with
ears

The two parse trees' probabilities and the sentence probability

$$
\begin{aligned}
P(t_1) &= 1.0 \times 0.1 \times 0.7 \times 1.0 \times 0.4 \\
&\quad \times 0.18 \times 1.0 \times 1.0 \times 0.18 \\
&= 0.0009072 \\
P(t_2) &= 1.0 \times 0.1 \times 0.3 \times 0.7 \times 1.0 \\
&\quad \times 0.18 \times 1.0 \times 1.0 \times 0.18 \\
&= 0.0006804 \\
P(w_{15}) &= P(t_1) + P(t_2) = 0.0015876
\end{aligned}
$$

Assumptions of PCFGs

1. Place invariance (like time invariance in HMM):

$$\forall k \quad P(N^j_{k(k+c)} \rightarrow \zeta) \text{is the same}$$

2. Context-free:

$$P(N^j_{kl} \rightarrow \zeta | \text{words outside } w_k \ldots w_l) = P(N^j_{kl} \rightarrow \zeta)$$

3. Ancestor-free:

$$P(N^j_{kl} \rightarrow \zeta | \text{ancestor nodes of } N^j_{kl}) = P(N^j_{kl} \rightarrow \zeta)$$

Let the upper left index in ${}^{i}N^{j}$ be an arbitrary identifying index for a particular token of a nonterminal.

Then,

$$P\left(\begin{array}{c} {}^{1}\mathrm{S} \\ {}^{2}\mathrm{NP} \quad {}^{3}\mathrm{VP} \\ \textit{the} \quad \textit{man} \quad \textit{snores} \end{array}\right) = P({}^{1}S_{13} \rightarrow {}^{2}NP_{12}\ {}^{3}VP_{33}, {}^{2}NP_{12} \rightarrow the_{1}\ man_{2}, {}^{3}VP_{33} \rightarrow sno$$

$$= \ldots$$

$$= P(S \rightarrow NP\ VP)P(NP \rightarrow the\ man)P(VP \rightarrow snores)$$

Some features of PCFGs

Reasons to use a PCFG, and some idea of their limitations:

- Partial solution for grammar ambiguity: a PCFG gives some idea of the plausibility of a sentence.
- But not a very good idea, as not lexicalized.
- Better for grammar induction (Gold 1967)
- Robustness. (Admit everything with low probability.)

Some features of PCFGs

- Gives a probabilistic language model for English.
- In practice, a PCFG is a worse language model for English than a trigram model.
- Can hope to combine the strengths of a PCFG and a trigram model.
- PCFG encodes certain biases, e.g., that smaller trees are normally more probable.

Improper (inconsistent) distributions

- S → *rhubarb* $P = \frac{1}{3}$

  S → S S $P = \frac{2}{3}$

- *rhubarb* $\frac{1}{3}$

  rhubarb rhubarb $\frac{2}{3} \times \frac{1}{3} \times \frac{1}{3} = \frac{2}{27}$

  rhubarb rhubarb rhubarb $\left(\frac{2}{3}\right)^2 \times \left(\frac{1}{3}\right)^3 \times 2 = \frac{8}{243}$

  ...

- $P(\mathcal{L}) = \frac{1}{3} + \frac{2}{27} + \frac{8}{243} + \ldots = \frac{1}{2}$
- Improper/inconsistent distribution
- Not a problem if you estimate from parsed treebank: Chi and Geman 1998).

Questions for PCFGs

Just as for HMMs, there are three basic questions we wish to answer:

- $P(w_{1m}|G)$
- $\arg\max_t P(t|w_{1m}, G)$
- Learning algorithm. Find G such that $P(w_{1m}|G)$ is maximized.

Chomsky Normal Form grammars

We'll do the case of Chomsky Normal Form grammars, which only have rules of the form:

$$N^i \rightarrow N^j N^k$$

$$N^i \rightarrow w^j$$

Any CFG can be represented by a weakly equivalent CFG in Chomsky Normal Form. It's straightforward to generalize the algorithm (recall chart parsing).

PCFG parameters

We'll do the case of Chomsky Normal Form grammars, which only have rules of the form:

$$N^i \rightarrow N^j N^k$$

$$N^i \rightarrow w^j$$

The parameters of a CNF PCFG are:

$P(N^j \rightarrow N^r N^s | G)$ A n^3 matrix of parameters

$P(N^j \rightarrow w^k | G)$ An nt matrix of parameters

For $j = 1, \ldots, n$,

$$\sum_{r,s} P(N^j \rightarrow N^r N^s) + \sum_k P(N^j \rightarrow w^k) = 1$$

Probabilistic Regular Grammar:

$N^i \rightarrow w^j N^k$

$N^i \rightarrow w^j$

Start state, N^1

HMM:

$$\sum_{w_{1n}} P(w_{1n}) = 1 \quad \forall n$$

whereas in a PCFG or a PRG:

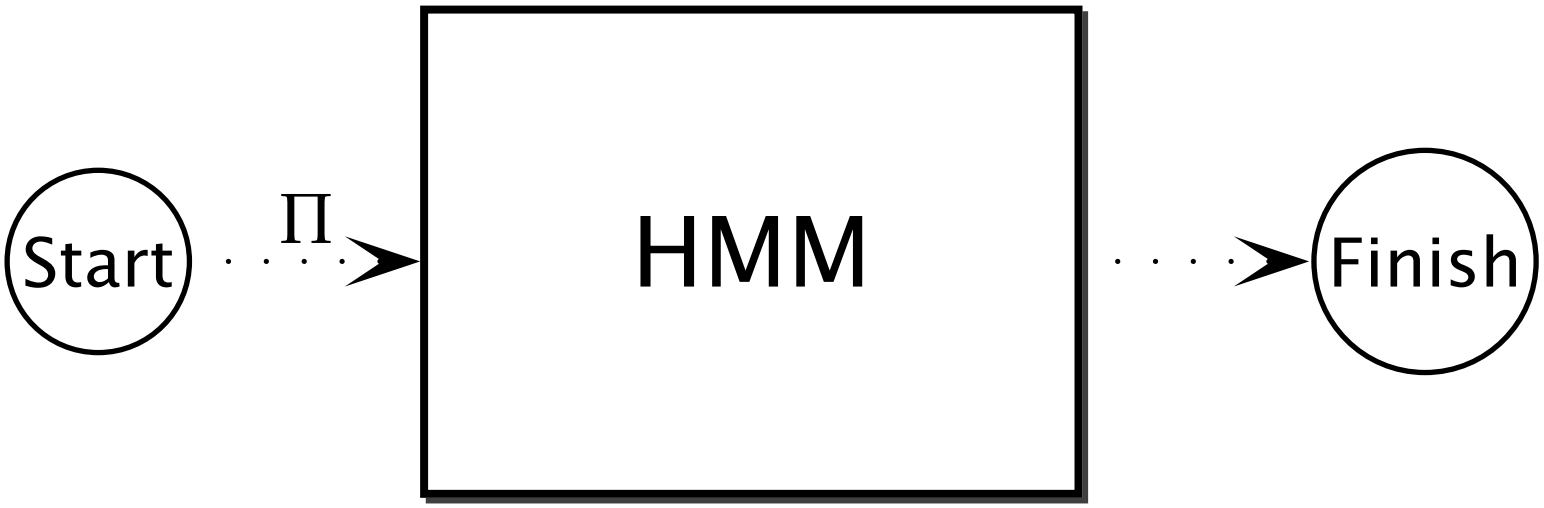
$$\sum_{w \in L} P(w) = 1$$

Consider:

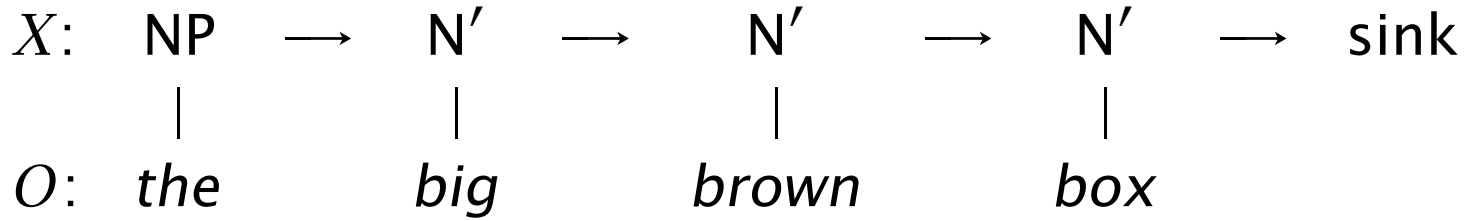
$$P(\text{John decided to bake a})$$

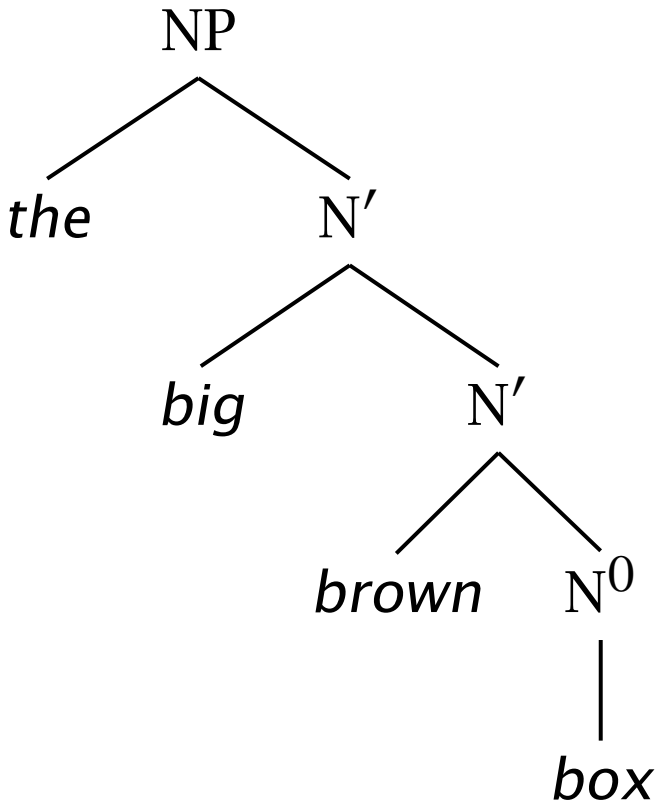
High probability in HMM, low probability in a PRG or a PCFG. Implement via sink state.

A PRG



Comparison of HMMs (PRGs) and PCFGs





Inside and outside probabilities

This suggests: whereas for an HMM we have:

Forwards = $\alpha_i(t) = P(w_{1(t-1)}, X_t = i)$

Backwards = $\beta_i(t) = P(w_{tT}|X_t = i)$

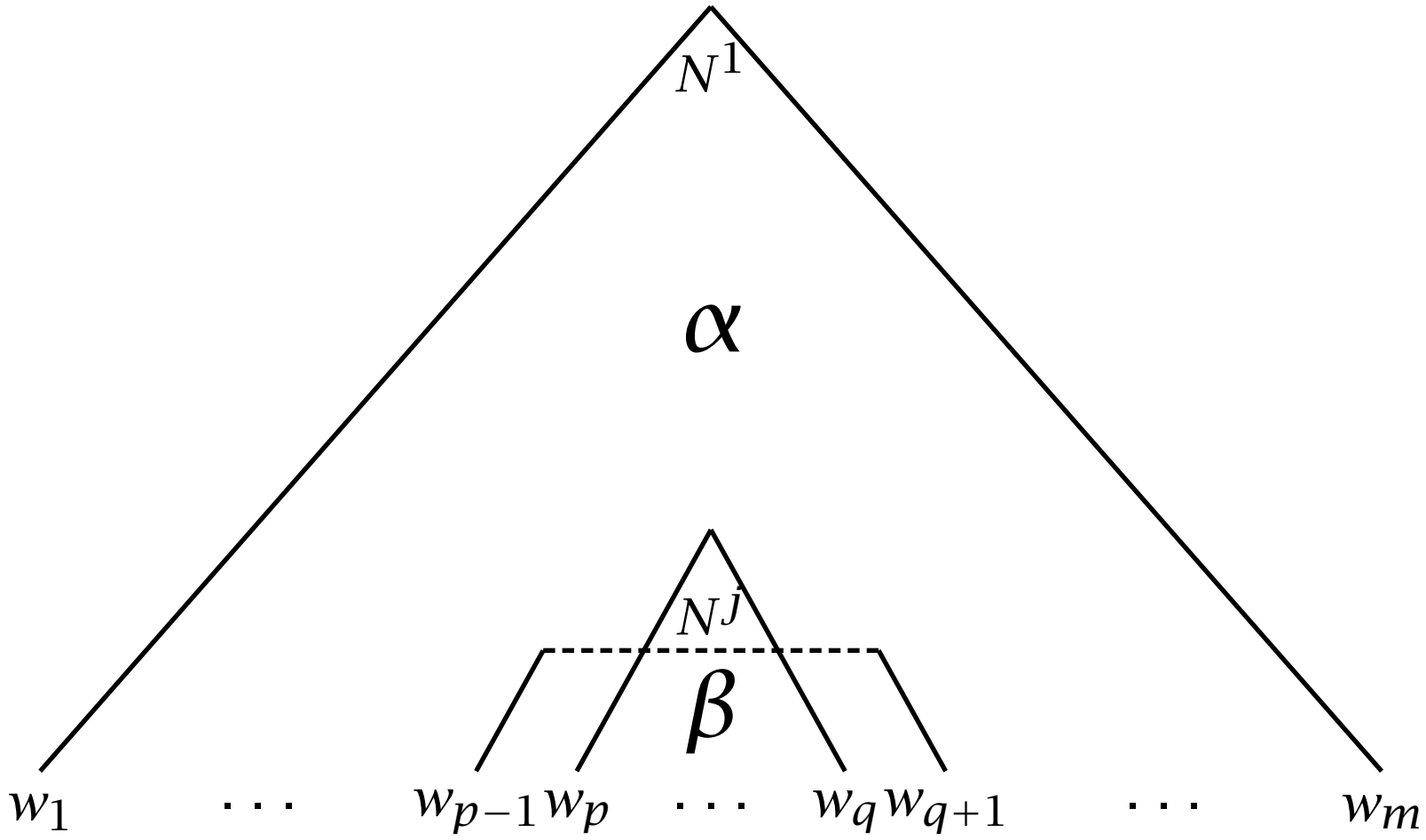
for a PCFG we make use of Inside and Outside probabilities, defined as follows:

Outside = $\alpha_j(p,q) = P(w_{1(p-1)}, N^j_{pq}, w_{(q+1)m}|G)$

Inside = $\beta_j(p,q) = P(w_{pq}|N^j_{pq}, G)$

A slight generalization of dynamic Bayes Nets covers PCFG inference by the inside-outside algorithm (and-or tree of conjunctive daughters disjunctively chosen)

Inside and outside probabilities in PCFGs.



Probability of a string

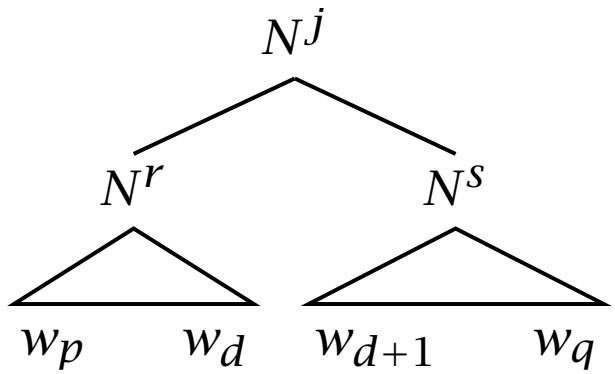
Inside probability

$$P(w_{1m}|G) = P(N^1 \Rightarrow w_{1m}|G)$$
$$= P(w_{1m}, N^1_{1m}, G) = \beta_1(1,m)$$

Base case: We want to find $\beta_j(k,k)$ (the probability of a rule $N^j \rightarrow w_k$):

$$\beta_j(k,k) = P(w_k|N^j_{kk}, G)$$
$$= P(N^j \rightarrow w_k|G)$$

Induction: We want to find $\beta_j(p,q)$, for $p < q$. As this is the inductive step using a Chomsky Normal Form grammar, the first rule must be of the form $N^j \rightarrow N^r \quad N^s$, so we can proceed by induction, dividing the string in two in various places and summing the result:



These inside probabilities can be calculated bottom up.

For all j,

$$
\begin{aligned}
\beta_j(p,q) &= P(w_{pq}|N^j_{pq},G) \\
&= \sum_{r,s}\sum_{d=p}^{q-1} P(w_{pd},N^r_{pd},w_{(d+1)q},N^s_{(d+1)q}|N^j_{pq},G) \\
&= \sum_{r,s}\sum_{d=p}^{q-1} P(N^r_{pd},N^s_{(d+1)q}|N^j_{pq},G) \\
&\qquad P(w_{pd}|N^j_{pq},N^r_{pd},N^s_{(d+1)q},G) \\
&\qquad P(w_{(d+1)q}|N^j_{pq},N^r_{pd},N^s_{(d+1)q},w_{pd},G) \\
&= \sum_{r,s}\sum_{d=p}^{q-1} P(N^r_{pd},N^s_{(d+1)q}|N^j_{pq},G) \\
&\qquad P(w_{pd}|N^r_{pd},G)P(w_{(d+1)q}|N^s_{(d+1)q},G) \\
&= \sum_{r,s}\sum_{d=p}^{q-1} P(N^j \rightarrow N^r N^s)\beta_r(p,d)\beta_s(d+1,q)
\end{aligned}
$$

Calculation of inside probabilities (CKY algorithm)

	1	2	3	4	5
1	$\beta_{NP} = 0.1$		$\beta_{S} = 0.0126$		$\beta_{S} = 0.0015876$
2		$\beta_{NP} = 0.04$ $\beta_{V} = 1.0$	$\beta_{VP} = 0.126$		$\beta_{VP} = 0.015876$
3			$\beta_{NP} = 0.18$		$\beta_{NP} = 0.01296$
4				$\beta_{P} = 1.0$	$\beta_{PP} = 0.18$
5					$\beta_{NP} = 0.18$
	astronomers	*saw*	*stars*	*with*	*ears*

Outside probabilities

Probability of a string: For any k, $1 \leq k \leq m$,

$$\begin{aligned} P(w_{1m}|G) &= \sum_j P(w_{1(k-1)}, w_k, w_{(k+1)m}, N^j_{kk}|G) \\ &= \sum_j P(w_{1(k-1)}, N^j_{kk}, w_{(k+1)m}|G) \\ &\qquad \times P(w_k|w_{1(k-1)}, N^j_{kk}, w_{(k+1)n}, G) \\ &= \sum_j \alpha_j(k,k) P(N^j \rightarrow w_k) \end{aligned}$$

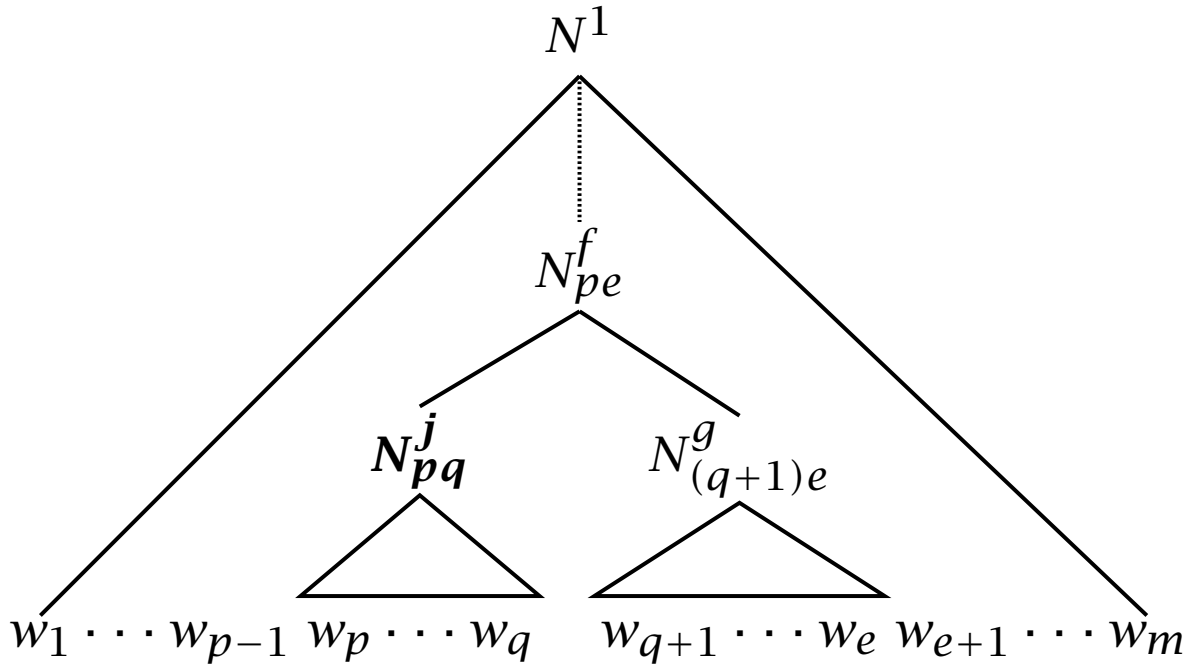
Inductive (DP) calculation: One calculates the outside probabilities top down (after determining the inside probabilities).

Outside probabilities

Base Case:

$$\alpha_1(1, m) = 1$$
$$\alpha_j(1, m) = 0, \text{ for } j \neq 1$$

Inductive Case:

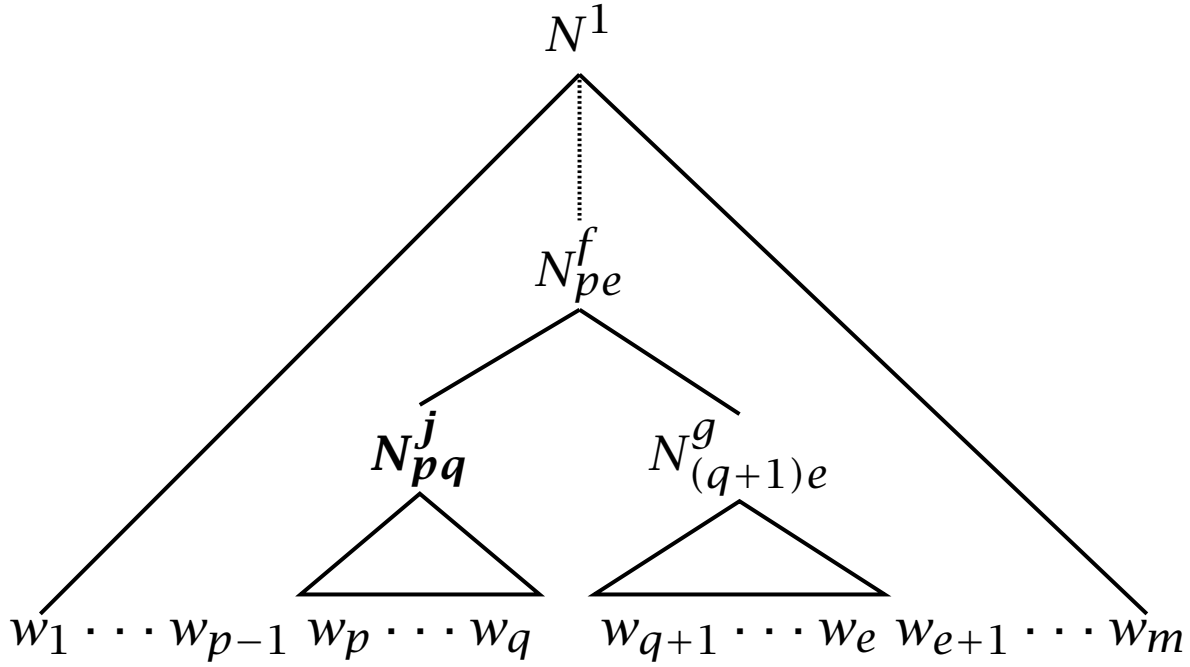


Outside probabilities

Base Case:

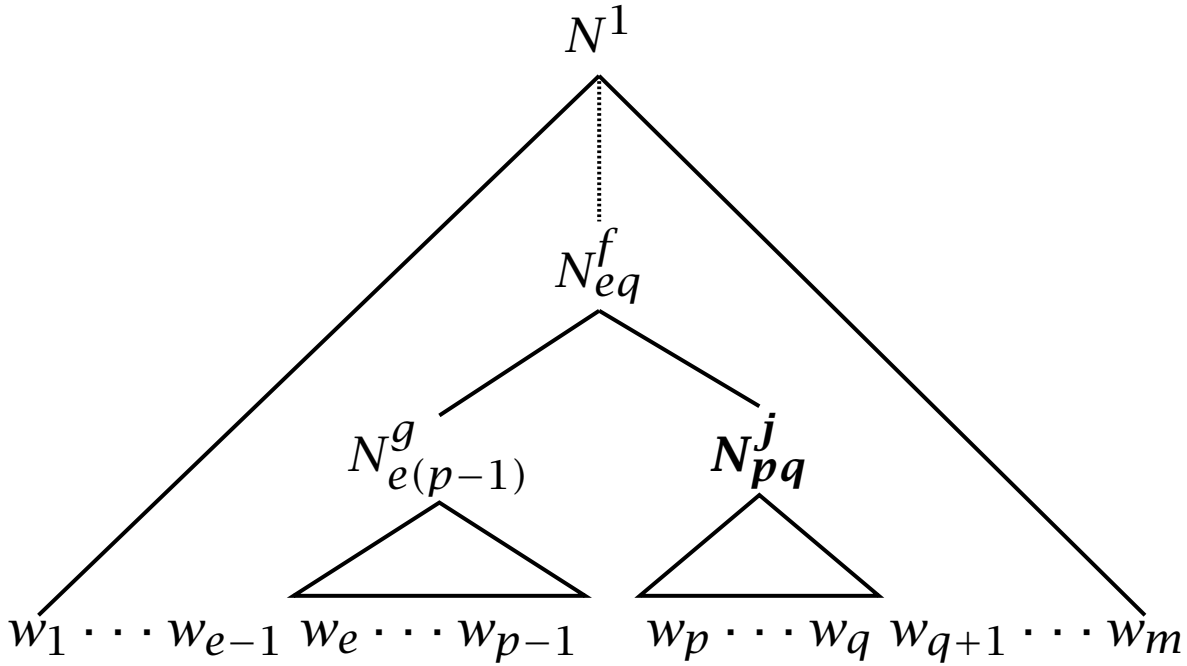
$$\alpha_1(1, m) = 1$$
$$\alpha_j(1, m) = 0, \text{ for } j \neq 1$$

Inductive Case: it's either a left or right branch – we will some over both possibilities and calculate using outside *and* inside probabilities



Outside probabilities – inductive case

A node N^j_{pq} might be the left or right branch of the parent node. We sum over both possibilities.



Inductive Case:

$$
\begin{aligned}
\alpha_j(p,q) &= [\sum_{f,g}\sum_{e=q+1}^{m} P(w_{1(p-1)}, w_{(q+1)m}, N^f_{pe}, N^j_{pq}, N^g_{(q+1)e})] \\
&\quad + [\sum_{f,g}\sum_{e=1}^{p-1} P(w_{1(p-1)}, w_{(q+1)m}, N^f_{eq}, N^g_{e(p-1)}, N^j_{pq})] \\
&= [\sum_{f,g ne j}\sum_{e=q+1}^{m} P(w_{1(p-1)}, w_{(e+1)m}, N^f_{pe}) P(N^j_{pq}, N^g_{(q+1)e} | N^f_{pe}) \\
&\quad \times P(w_{(q+1)e} | N^g_{(q+1)e})] + [\sum_{f,g}\sum_{e=1}^{p-1} P(w_{1(e-1)}, w_{(q+1)m}, N^f_{eq}) \\
&\quad \times P(N^g_{e(p-1)}, N^j_{pq} | N^f_{eq}) P(w_{e(p-1)} | N^g_{e(p-1)})] \\
&= [\sum_{f,g}\sum_{e=q+1}^{m} \alpha_f(p,e) P(N^f \rightarrow N^j\ N^g) \beta_g(q+1,e)] \\
&\quad + [\sum_{f,g}\sum_{e=1}^{p-1} \alpha_f(e,q) P(N^f \rightarrow N^g\ N^j) \beta_g(e,p-1)]
\end{aligned}
$$

Overall probability of a node existing

As with a HMM, we can form a product of the inside and outside probabilities. This time:

$$\begin{aligned}
&\alpha_j(p,q)\beta_j(p,q) \\
&= P(w_{1(p-1)}, N^j_{pq}, w_{(q+1)m}|G)P(w_{pq}|N^j_{pq}, G) \\
&= P(w_{1m}, N^j_{pq}|G)
\end{aligned}$$

Therefore,

$$p(w_{1m}, N_{pq}|G) = \sum_j \alpha_j(p,q)\beta_j(p,q)$$

Just in the cases of the root node and the preterminals, we know there will always be some such constituent.

Training a PCFG

We construct an EM training algorithm, as for HMMs. We would like to calculate how often each rule is used:

$$\hat{P}(N^j \to \zeta) = \frac{C(N^j \to \zeta)}{\sum_\gamma C(N^j \to \gamma)}$$

Have data $\Rightarrow$ count; else work iteratively from expectations of current model.

Consider:

$$\begin{aligned} \alpha_j(p,q)\beta_j(p,q) &= P(N^1 \overset{*}{\Longrightarrow} w_{1m}, N^j \overset{*}{\Longrightarrow} w_{pq}|G) \\ &= P(N^1 \overset{*}{\Longrightarrow} w_{1m}|G)P(N^j \overset{*}{\Longrightarrow} w_{pq}|N^1 \overset{*}{\Longrightarrow} w_{1m}, G) \end{aligned}$$

We have already solved how to calculate $P(N^1 \Rightarrow w_{1m})$; let us call this probability π. Then:

$$P(N^j \overset{*}{\Longrightarrow} w_{pq}|N^1 \overset{*}{\Longrightarrow} w_{1m}, G) = \frac{\alpha_j(p,q)\beta_j(p,q)}{\pi}$$

and

$$E(N^j \text{ is used in the derivation}) = \sum_{p=1}^{m}\sum_{q=p}^{m}\frac{\alpha_j(p,q)\beta_j(p,q)}{\pi}$$

In the case where we are not dealing with a preterminal, we substitute the inductive definition of β, and $\forall r, s, p > q$:

$$P(N^j \rightarrow N^r\ N^s \Rightarrow w_{pq} | N^1 \Rightarrow w_{1n}, G) =$$

$$\frac{\sum_{d=p}^{q-1} \alpha_j(p,q) P(N^j \rightarrow N^r\ N^s) \beta_r(p,d) \beta_s(d+1,q)}{\pi}$$

Therefore the expectation is:

$E(N^j \rightarrow N^r\ N^s, N^j \text{ used})$

$$\frac{\sum_{p=1}^{m-1} \sum_{q=p+1}^{m} \sum_{d=p}^{q-1} \alpha_j(p,q) P(N^j \rightarrow N^r\ N^s) \beta_r(p,d) \beta_s(d+1,q)}{\pi}$$

Now for the maximization step, we want:

$$P(N^j \rightarrow N^r\ N^s) = \frac{E(N^j \rightarrow N^r\ N^s, N^j \text{ used})}{E(N^j \text{ used})}$$

Therefore, the reestimation formula, $\hat{P}(N^j \rightarrow N^r\ N^s)$ is the quotient:

$$\hat{P}(N^j \rightarrow N^r\ N^s) =$$

$$\frac{\sum_{p=1}^{m-1}\sum_{q=p+1}^{m}\sum_{d=p}^{q-1}\alpha_j(p,q)P(N^j \rightarrow N^r\ N^s)\beta_r(p,d)\beta_s(d+1,q)}{\sum_{p=1}^{m}\sum_{q=1}^{m}\alpha_j(p,q)\beta_j(p,q)}$$

Similarly,

$$E(N^j \rightarrow w^k | N^1 \Rightarrow w_{1m}, G) =$$

$$\frac{\sum_{h=1}^{m}\alpha_j(h,h)P(N^j \rightarrow w_h, w_h = w^k)}{\pi}$$

Therefore,

$$\hat{P}(N^j \rightarrow w^k) = \frac{\sum_{h=1}^{m}\alpha_j(h,h)P(N^j \rightarrow w_h, w_h = w^k)}{\sum_{p=1}^{m}\sum_{q=1}^{m}\alpha_j(p,q)\beta_j(p,q)}$$

Inside-Outside algorithm: repeat this process until the estimated probability change is small.

Multiple training instances: if we have training sentences $W = (W_1, \ldots W_\omega)$, with $W_i = (w_1, \ldots, w_{m_i})$ and we let u and v bet the common subterms from before:

$$u_i(p, q, j, r, s) =$$

$$\frac{\sum_{d=p}^{q-1} \alpha_j(p, q) P(N^j \rightarrow N^r N^s) \beta_r(p, d) \beta_s(d+1, q)}{P(N^1 \Rightarrow W_i | G)}$$

and

$$v_i(p, q, j) = \frac{\alpha_j(p, q) \beta_j(p, q)}{P(N^1 \Rightarrow W_i | G)}$$

Assuming the observations are independent, we can sum contributions:

$$\hat{P}(N^j \rightarrow N^r \; N^s) = \frac{\sum_{i=1}^{\omega} \sum_{p=1}^{m_i - 1} \sum_{q=p+1}^{m_i} u_i(p, q, j, r, s)}{\sum_{i=1}^{\omega} \sum_{p=1}^{m_i} \sum_{q=p}^{m_i} v_i(p, q, j)}$$

and

$$\hat{P}(N^j \rightarrow w^k) = \frac{\sum_{i=1}^{\omega} \sum_{\{h: w_h = w^k\}} v_i(h, h, j)}{\sum_{i=1}^{\omega} \sum_{p=1}^{m_i} \sum_{q=p}^{m_i} v_i(p, q, j)}$$

Problems with the Inside-Outside algorithm

1. Slow. Each iteration is $O(m^3 n^3)$, where $m = \sum_{i=1}^{\omega} m_i$, and n is the number of nonterminals in the grammar.
2. Local maxima are much more of a problem. Charniak reports that on each trial a different local maximum was found. Use simulated annealing? Restrict rules by initializing some parameters to zero? Or HMM initialization? Reallocate nonterminals away from "greedy" terminals?
3. Lari and Young suggest that you need many more nonterminals available than are theoretically necessary to get good grammar learning (about a threefold increase?). This compounds the first problem.
4. There is no guarantee that the nonterminals that the algorithm learns will have any satisfactory resemblance to the kinds of nonterminals normally motivated in linguistic analysis (NP, VP, etc.).