Mathematische Grundlagen III: Statistische Methoden
July 3, 2008

Probabilistic Context-free Grammars:
Inside and Outside Probabilities, and Viterbi Parsing

Matthew Crocker

Computerlinguistik
Universität des Saarlandes

crocker@coli.uni-sb.de

UNIVERSITÄT
DES
SAARLANDES

Overview

- lexicalization of PCFGs
- computing the probability of a string
- inside and outside probabilities
- computing the best parse
- the viterbi algorithm

Again, we will follow Manning and Schuetze (1999), Chapter 11.

Lexicalizing a PCFG

S $\rightarrow$ NP VP	1.0	NP $\rightarrow$ NP PP	0.4
PP $\rightarrow$ P NP	1.0	NP $\rightarrow$ astronomers	0.1
VP $\rightarrow$ V NP	0.7	NP $\rightarrow$ ears	0.18
VP $\rightarrow$ VP PP	0.3	NP $\rightarrow$ saw	0.04
P $\rightarrow$ with	1.0	NP $\rightarrow$ stars	0.18
V $\rightarrow$ saw	1.0	NP $\rightarrow$ telescopes	0.1

Naive lexicalization of only the VP rules:

VP(saw) $\rightarrow$ V(saw) NP(astronomers)	0.1	V(saw) $\rightarrow$ saw	1.0
VP(saw) $\rightarrow$ V(saw) NP(ears)	0.15	NP $\rightarrow$ astronomers	0.1
VP(saw) $\rightarrow$ V(saw) NP(saw)	0.05	NP $\rightarrow$ ears	0.18
VP(saw) $\rightarrow$ V(saw) NP(stars)	0.3	NP $\rightarrow$ saw	0.04
VP(saw) $\rightarrow$ V(saw) NP(telescopes)	0.1	NP $\rightarrow$ stars	0.18
VP(saw) $\rightarrow$ VP(saw) PP(with)	0.3	NP $\rightarrow$ telescopes	0.1

Now, suppose we had 10 verbs ... we would have 60 VP rules.

Lexicalizing a PCFG

A fully lexicalized grammar requires that for all rules:

$N^j \rightarrow N^r \; N^s$

we must estimate:

$P(N^j \rightarrow N^r \; N^s | j, h(N^j), h(N^r), h(N^s))$

However, this requires us to estimate an impossible number of parameters, for which there will be insufficient training data. Thus, typically, we condition on just the lexical head:

$P(N^j \rightarrow N^r \; N^s | j, h(N^j))$

In other words, our grammar now has only two VP rules again:

VP(saw) $\rightarrow$ V(saw) NP	0.7
VP(saw) $\rightarrow$ VP(saw) PP	0.3

Now, suppose we had 10 verbs ... we would have 20 VP rules.

Inside Probability

We can calculate $P(w_{1m}|G)$, the overall probability of a string, by computing all possible parses and summing up their probabilities. However, this naive algorithm is exponential (same problem as with HMMs).

Solution: store partial results (probabilities of substrings), instead of duplicating them.

The overall probability of a string w_{1m} is:

(1)
$$\begin{aligned} P(w_{1m}|G) &= P(N^1 \Rightarrow^* w_{1m}|G) \\ &= P(w_{1m}|N^1_{1m}, G) \\ &= \beta_1(1,m) \end{aligned}$$

This can be generalized to $\beta_j(p,q)$, the probability of the nonterminal N^j spanning the string from word p to q.

Inside probabilities can be calculated efficiently using the inside procedure.

The Inside Procedure

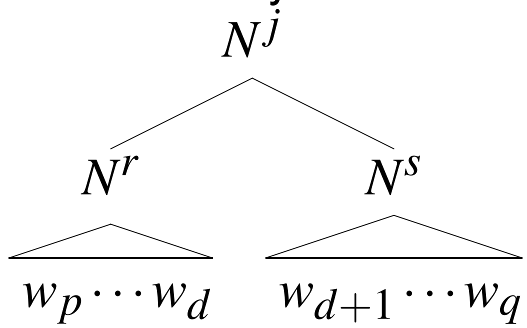
1. Base Case

We compute $\beta_j(k,k)$, the probability of the subtree spanning the word k and headed by the nonterminal N^j (i.e., the probability of the rule $N^j \longrightarrow w_k$):

$$\text{(2)} \qquad \begin{aligned} \beta_j(k,k) &= P(w_k | N^j_{kk}, G) \\ &= P(N^j \longrightarrow w_k | G) \end{aligned}$$

2. Induction: bottom up

We compute $\beta_j(p,q), p < q$, the probability of the nonterminal N^j spanning the string from word p to q. The grammar is in Chomsky Normal Form, so we know that N^j expands to N^r and N^s:



The Inside Procedure

$\forall j, 1 \leq p < q \leq m$

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

Example

Compute the β-table for the following PCFG:

S $\rightarrow$ NP VP	1.0	NP $\rightarrow$ NP PP	0.4
PP $\rightarrow$ P NP	1.0	NP $\rightarrow$ astronomers	0.1
VP $\rightarrow$ V NP	0.7	NP $\rightarrow$ ears	0.18
VP $\rightarrow$ VP PP	0.3	NP $\rightarrow$ saw	0.04
P $\rightarrow$ with	1.0	NP $\rightarrow$ stars	0.18
V $\rightarrow$ saw	1.0	NP $\rightarrow$ telescopes	0.1

And the input string:

$$(\text{astronomers}_1, \text{saw}_2, \text{stars}_3, \text{with}_4, \text{ears}_5)$$

Example

1. Base Case

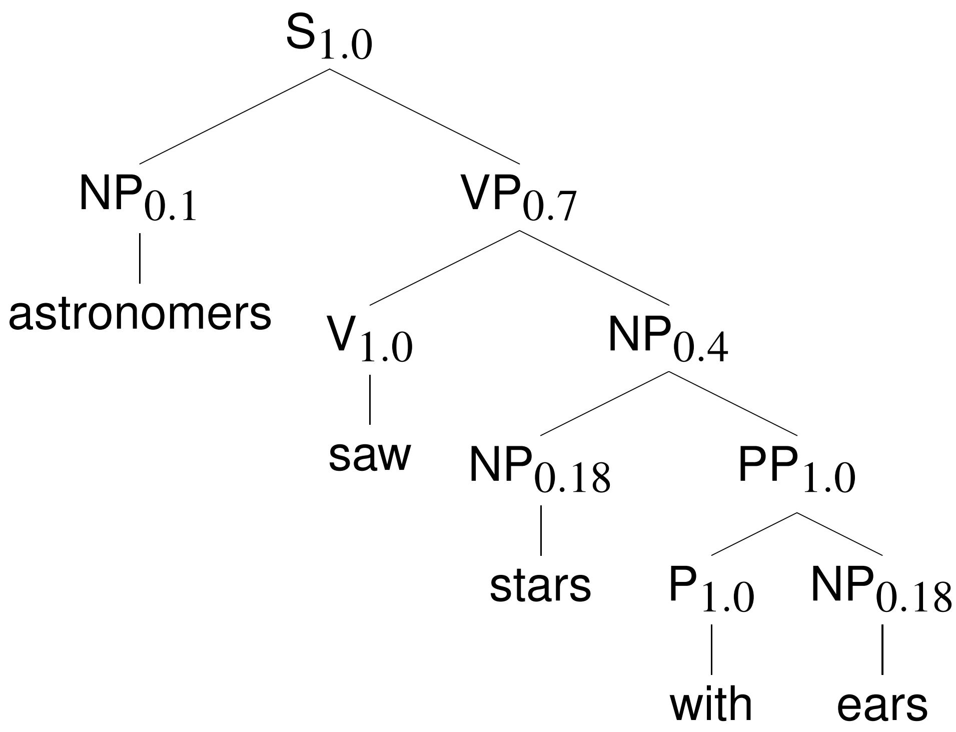
$$
\begin{aligned}
\beta_{\mathrm{NP}}(1,1) &= P(\mathrm{NP} \rightarrow \mathrm{astronomers}) = 0.1 \\
\beta_{\mathrm{V}}(2,2) &= P(\mathrm{V} \rightarrow \mathrm{saw}) = 1.0 \\
\beta_{\mathrm{NP}}(2,2) &= P(\mathrm{NP} \rightarrow \mathrm{saw}) = 0.04 \\
\beta_{\mathrm{NP}}(3,3) &= P(\mathrm{NP} \rightarrow \mathrm{stars}) = 0.18 \\
\beta_{\mathrm{P}}(4,4) &= P(\mathrm{P} \rightarrow \mathrm{with}) = 1.0 \\
\beta_{\mathrm{NP}}(5,5) &= P(\mathrm{NP} \rightarrow \mathrm{ears}) = 0.18
\end{aligned}
$$

2. Induction

$$
\begin{aligned}
\beta_{\mathrm{VP}}(2,3) &= P(\mathrm{VP} \rightarrow \mathrm{V\ NP})\beta_{\mathrm{V}}(2,2)\beta_{\mathrm{NP}}(3,3) \\
&= 0.7 \times 1.0 \times 0.18 = 0.126 \\
\beta_{\mathrm{S}}(1,3) &= P(\mathrm{S} \rightarrow \mathrm{NP\ VP})\beta_{\mathrm{NP}}(1,1)\beta_{\mathrm{VP}}(2,3) = 0.0126 \\
\beta_{\mathrm{PP}}(4,5) &= P(\mathrm{PP} \rightarrow \mathrm{P\ NP})\beta_{\mathrm{P}}(4,4)\beta_{\mathrm{NP}}(5,5) = 0.18 \\
\beta_{\mathrm{NP}}(3,5) &= P(\mathrm{NP} \rightarrow \mathrm{NP\ PP})\beta_{\mathrm{NP}}(3,3)\beta_{\mathrm{PP}}(4,5) \\
&= 0.4 \times 0.18 \times 0.18 = 0.01296 \\
\beta_{\mathrm{VP}}(2,5) &= P(\mathrm{VP} \rightarrow \mathrm{V\ NP})\beta_{\mathrm{V}}(2,2)\beta_{\mathrm{NP}}(3,5) \\
&+ P(\mathrm{VP} \rightarrow \mathrm{VP\ PP})\beta_{\mathrm{VP}}(2,3)\beta_{\mathrm{PP}}(4,5) = 0.015876 \\
\beta_{\mathrm{S}}(1,5) &= P(\mathrm{S} \rightarrow \mathrm{NP\ VP})\beta_{\mathrm{NP}}(1,1)\beta_{\mathrm{VP}}(2,5) = 0.0015876
\end{aligned}
$$

Example Parse Tree

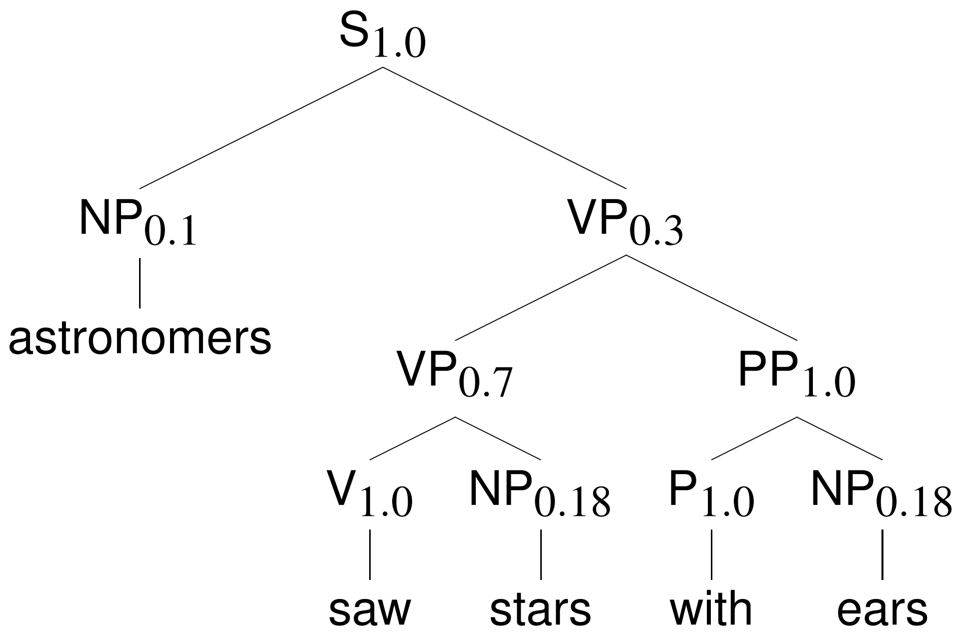
t_1:



$$P(t_1) = 1.0 \cdot 0.1 \cdot 0.7 \cdot 1.0 \cdot 0.4 \cdot 0.18 \cdot 1.0 \cdot 1.0 \cdot 0.18 = 0.0009072$$

Example Parse Tree

t_2:



$$P(t_1) = 1.0 \cdot 0.1 \cdot 0.3 \cdot 0.7 \cdot 1.0 \cdot 0.18 \cdot 1.0 \cdot 1.0 \cdot 0.18 = 0.0006804$$

Overall probability of the sentence:

$$P(w_{15}) = P(t_1) + P(t_2) = 0.0015876$$

Computing the Most Probable Parse

We want to compute $\hat{t}$, the most probable parse for a string w_{1m}:

(4) $\hat{t} = \arg\max_{t} P(t|w_{1m}, G)$

An efficient way of doing this is the Viterbi Algorithm. In analogy with HMMs, it uses a substring table to store subparses:

(5) $\delta_i(p,q) =$ the highest inside probability parse of a subtree N^i_{pq}

The backtrace $\psi_i(p,q)$ stores the rule whose application leads to the highest inside probability for the parse N^i_{pq}.

$\psi_i(p,q) = (j,k,r)$ stores an application of rule $N^i \rightarrow N^j\ N^k$, spanning the string w_{pq} and splitting it at point r.

Viterbi Algorithm

1. Initialization

(6) $$\delta_i(p,p) = P(N^i \to w_p)$$

2. Induction

(7) $$\delta_i(p,q) = \max_{\substack{1 \le j,k \le n \\ p \le r < q}} P(N^i \to N^j\, N^k)\delta_j(p,r)\delta_k(r+1,q)$$

Store Backtrace

(8) $$\psi_i(p,q) = \arg\max_{(j,k,r)} P(N^i \to N^j\, N^k)\delta_j(p,r)\delta_k(r+1,q)$$

3. Termination and Path Readout

The probability of the most probable parse rooted in the start symbol is:

(9) $$P(\hat{t}) = \delta_1(1,m)$$

Viterbi Algorithm

The most probable tree $\hat{t}$ can be reconstructed as follows:

1. The root node of the most probable parse is N^1_{1m}.
2. Let N^i_{pq} be the most probable parse and $\psi_i(p,q) = (j,k,r)$ the corresponding backtrace. Then the left and right daughters of N^i_{pq} are:

$$\text{left}(N^i_{pq}) = N^j_{pr} \quad (10)$$

$$\text{right}(N^i_{pq}) = N^k_{(r+1)q} \quad (11)$$

If there is no unique maximum in the computation of δ and ψ, then we simply chose one maximum at random

Example

Compute the δ- and ψ-table for the following PCFG:

S $\rightarrow$ NP VP	1.0	NP $\rightarrow$ NP PP	0.4
PP $\rightarrow$ P NP	1.0	NP $\rightarrow$ astronomers	0.1
VP $\rightarrow$ V NP	0.7	NP $\rightarrow$ ears	0.18
VP $\rightarrow$ VP PP	0.3	NP $\rightarrow$ saw	0.04
P $\rightarrow$ with	1.0	NP $\rightarrow$ stars	0.18
V $\rightarrow$ saw	1.0	NP $\rightarrow$ telescopes	0.1

And the input string:

$$(\text{astronomers}_1, \text{saw}_2, \text{stars}_3, \text{with}_4, \text{ears}_5)$$

Example

1. Initialization

$$
\begin{aligned}
\delta_{\mathrm{NP}}(1,1) &= P(\mathrm{NP} \rightarrow \text{astronomers}) = 0.1 \\
\delta_{\mathrm{V}}(2,2) &= P(\mathrm{V} \rightarrow \text{saw}) = 1.0 \\
\delta_{\mathrm{NP}}(2,2) &= P(\mathrm{NP} \rightarrow \text{saw}) = 0.04 \\
\delta_{\mathrm{NP}}(3,3) &= P(\mathrm{NP} \rightarrow \text{stars}) = 0.18 \\
\delta_{\mathrm{P}}(4,4) &= P(\mathrm{P} \rightarrow \text{with}) = 1.0 \\
\delta_{\mathrm{NP}}(5,5) &= P(\mathrm{NP} \rightarrow \text{ears}) = 0.18
\end{aligned}
$$

2. Induction

$$
\begin{aligned}
\delta_{\mathrm{VP}}(2,3) &= \max_{\substack{1 \leq j,k \leq n \\ 2 \leq r < 3}} P(\mathrm{VP} \rightarrow N^j\, N^k)\delta_j(2,r)\delta_k(r+1,3) \\
&= P(\mathrm{VP} \rightarrow \mathrm{V}\ \mathrm{NP})\delta_{\mathrm{V}}(2,2)\delta_{\mathrm{NP}}(3,3) = 0.126 \\
\psi_{\mathrm{VP}}(2,3) &= \arg\max_{(j,k,r)} P(\mathrm{VP} \rightarrow \mathrm{V}\ \mathrm{NP})\delta_{\mathrm{V}}(2,2)\delta_{\mathrm{NP}}(3,3) = (\mathrm{V}, \mathrm{NP}, 2)
\end{aligned}
$$

Example continued

$$
\begin{aligned}
\delta_{\mathrm{S}}(1,3) &= \max_{\substack{1 \le j,k \le n \\ 1 \le r < 3}} P(\mathrm{S} \to N^j\, N^k)\delta_j(1,r)\delta_k(r+1,3) \\
&= P(\mathrm{S} \to \mathrm{NP\ VP})\delta_{\mathrm{NP}}(1,1)\delta_{\mathrm{VP}}(2,3) = 0.0126 \\
\psi_{\mathrm{S}}(1,3) &= \arg\max_{(j,k,r)} P(\mathrm{S} \to \mathrm{NP\ VP})\delta_{\mathrm{NP}}(1,1)\delta_{\mathrm{VP}}(2,3) = (\mathrm{NP}, \mathrm{VP}, 1)
\end{aligned}
$$

$$
\begin{aligned}
\delta_{\mathrm{PP}}(4,5) &= \max_{\substack{1 \le j,k \le n \\ 4 \le r < 5}} P(\mathrm{PP} \to N^j\, N^k)\delta_j(4,r)\delta_k(r+1,5) \\
&= P(\mathrm{PP} \to \mathrm{P\ NP})\delta_{\mathrm{P}}(4,5)\delta_{\mathrm{NP}}(5,5) = 0.18 \\
\psi_{\mathrm{PP}}(4,5) &= \arg\max_{(j,k,r)} P(\mathrm{PP} \to \mathrm{P\ NP})\delta_{\mathrm{P}}(4,4)\delta_{\mathrm{NP}}(5,5) = (\mathrm{P}, \mathrm{NP}, 4)
\end{aligned}
$$

$$
\begin{aligned}
\delta_{\mathrm{NP}}(3,5) &= \max_{\substack{1 \le j,k \le n \\ 3 \le r < 5}} P(\mathrm{NP} \to N^j\, N^k)\delta_j(3,r)\delta_k(r+1,5) \\
&= P(\mathrm{NP} \to \mathrm{NP\ PP})\delta_{\mathrm{NP}}(3,3)\delta_{\mathrm{PP}}(4,5) = 0.01296 \\
\psi_{\mathrm{NP}}(3,5) &= \arg\max_{(j,k,r)} P(\mathrm{NP} \to \mathrm{NP\ PP})\delta_{\mathrm{NP}}(3,3)\delta_{\mathrm{PP}}(4,5) = (\mathrm{NP}, \mathrm{PP}, 3)
\end{aligned}
$$

Example continued

$$
\begin{aligned}
\delta_{\mathrm{VP}}(2,5) &= \max_{\substack{1\le j,k\le n\\ 2\le r<5}} P(\mathrm{VP} \to N^j\, N^k)\delta_j(2,r)\delta_k(r+1,5) \\
&= P(\mathrm{VP} \to \mathrm{V\ NP})\delta_{\mathrm{V}}(2,2)\delta_{\mathrm{NP}}(3,5) = 0.009072 \\
(&= P(\mathrm{VP} \to \mathrm{VP\ PP})\delta_{\mathrm{VP}}(2,3)\delta_{\mathrm{PP}}(4,5) = 0.006804) \\
\psi_{\mathrm{VP}}(2,5) &= \arg\max_{(j,k,r)} P(\mathrm{VP} \to \mathrm{V\ NP})\delta_{\mathrm{V}}(2,2)\delta_{\mathrm{NP}}(3,5) = (\mathrm{V}, \mathrm{NP}, 2)
\end{aligned}
$$

$$
\begin{aligned}
\delta_{\mathrm{S}}(1,5) &= \max_{\substack{1\le j,k\le n\\ 1\le r<5}} P(\mathrm{S} \to N^j\, N^k)\delta_j(1,r)\delta_k(r+1,5) \\
&= P(\mathrm{S} \to \mathrm{NP\ VP})\delta_{\mathrm{NP}}(1,1)\delta_{\mathrm{VP}}(2,5) = 0.0009072 \\
\psi_{\mathrm{S}}(1,5) &= \arg\max_{(j,k,r)} P(\mathrm{S} \to \mathrm{NP\ VP})\delta_{\mathrm{NP}}(1,1)\delta_{\mathrm{VP}}(2,5) = (\mathrm{NP}, \mathrm{VP}, 1)
\end{aligned}
$$

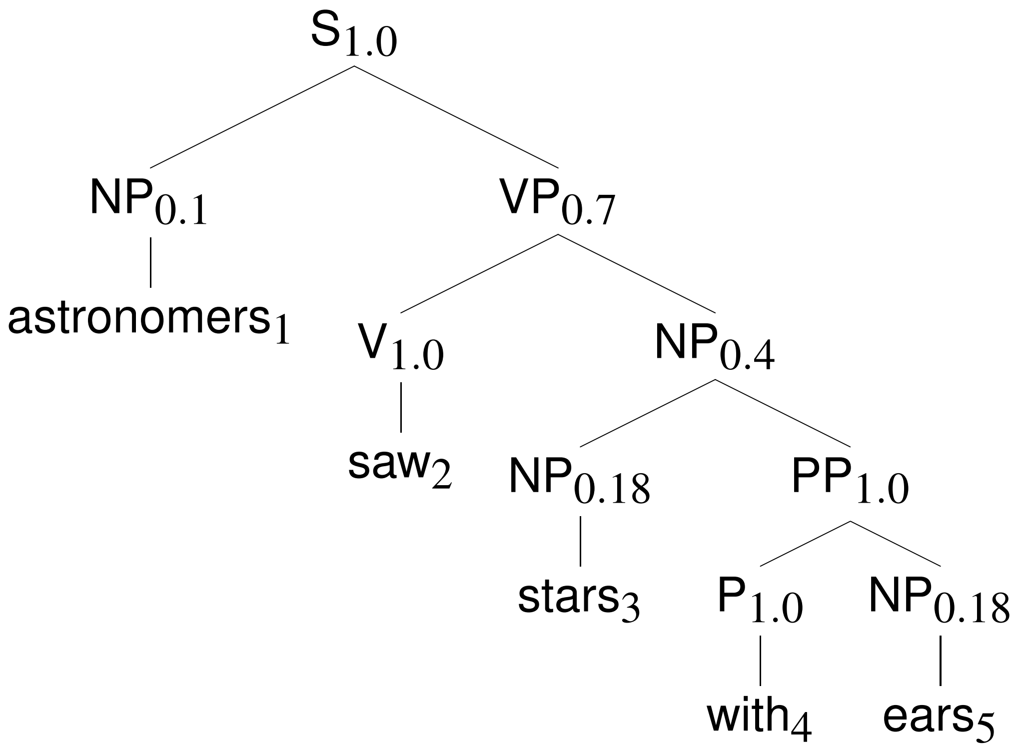
Example

3. Termination and Path Readout - the root node is $N^1_{1m} = S_{15}$:

$$
\begin{aligned}
\psi_{\mathrm{S}}(1,5) &= (j,k,r) = (\mathrm{NP},\mathrm{VP},1) \\
\mathrm{left(S)} &= N^j_{1r} = \mathrm{NP}_{11} \\
\mathrm{right(S)} &= N^k_{(r+1)5} = \mathrm{VP}_{25} \\
\psi_{\mathrm{VP}}(2,5) &= (j,k,r) = (\mathrm{V},\mathrm{NP},5) \\
\mathrm{left(VP)} &= N^j_{2r} = \mathrm{V}_{22} \\
\mathrm{right(VP)} &= N^k_{(r+1)5} = \mathrm{NP}_{35} \\
\psi_{\mathrm{NP}}(3,5) &= (j,k,r) = (\mathrm{NP},\mathrm{PP},3) \\
\mathrm{left(NP)} &= N^j_{3r} = \mathrm{NP}_{33} \\
\mathrm{right(NP)} &= N^k_{(r+1)5} = \mathrm{PP}_{45} \\
\psi_{\mathrm{PP}}(4,5) &= (j,k,r) = (\mathrm{P},\mathrm{NP},4) \\
\mathrm{left(PP)} &= N^j_{4r} = \mathrm{P}_{44} \\
\mathrm{right(PP)} &= N^k_{(r+1)5} = \mathrm{NP}_{55}
\end{aligned}
$$

The Highest Probability Tree

t_1:



$$P(t_1) = 1.0 \cdot 0.1 \cdot 0.7 \cdot 1.0 \cdot 0.4 \cdot 0.18 \cdot 1.0 \cdot 1.0 \cdot 0.18 = 0.0009072$$

Training PCFGs

Training a PCFG is grammar induction in a limited sense:

- the structure of the grammar is given (i.e., a set of terminals and non-terminals);
- we can also assume that a set of grammar rules is given; then we train probabilities for these rules;
- alternatively, we can assume all possible grammar rules for a given set of terminals and non-terminals (e.g., all binary rules);
- during training, some rules will turn out to have zero probability: we learn which grammar rules are possible.

Formally, training amounts to choosing the grammar G that maximizes the probability of the training sentence, $\arg\max_G P(w_{1m}|G)$. (In the general case, we maximize the probability of a training corpus.)

Training PCFGs

The Inside-Outside Algorithm is an efficient way of maximizing the probability of the training corpus. It is an instance of the Expectation Maximization Algorithm.

1. Start with an arbitrary set of rule probabilities. Compute $P(w_{1m}|G)$ for these rule probabilities.
2. Figure out which rules were used most in calculating $P(w_{1m}|G)$.
3. Increase their probabilities, which will yield a new set of rule probabilities with a higher $P(w_{1m}|G)$.
4. Iterate until a (local) maximum is reached.

Note that this is an unsupervised learning algorithm; it doesn't required any annotated training data.

Summary of PCFGs

A straightforward way to use probabilistic information to improve ambiguity resolution for wide-coverage parsers.

1. Simple augmentation of standard CFG formalism
2. Estimation of parameters using Inside-Outside Algorithm, or from a Treebank
3. Can be used as both a *language model* (probability of a sequence of words) and a *parsing model* (probability of a specific parse tree)
4. Efficient algorithms for computing string probabilities (using Inside or Outside probabilities), and finding the best parse (Viterbi)
5. Limited performance due to context and lexical “freeness”. Can be improved through lexicaliztion, parentization, etc.