

# 统计句法分析

常宝宝

北京大学计算语言学研究所

chbb@pku.edu.cn

# 基于规则的句法分析

- ◆ 基于规则的句法分析  
上下文无关语法、特征结构、合一运算
- ◆ 基于规则的句法分析的缺陷
  - (1) 很难穷尽所有的句法规则
  - (2) 很难保证规则之间的一致性
  - (3) 通常只能处理有限的比较规范的句子
  - (4) 对真实语料的处理能力不够
  - (5) 分析效率通常不高
- ◆ 引入概率，建立基于统计的句法分析
- ◆ 统计句法分析 近年来引起了较多的关注，并取得了较好的研究成果。

# 统计句法分析的基本思路

- ◆ 对给定的句子 $S$ ，该句子的统计句法分析结果为：

$$\hat{T} = \arg \max_T P(T | S)$$

根据贝叶斯公式，有：

$$\hat{T} = \arg \max_T P(T, S)$$

- ◆ 如何计算句法树的概率？  
概率上下文无关文法 ( PCFG )  
词汇化 PCFG

# 什么是概率上下文无关文法？

- ◆ PCFG是CFG的一种扩展。
- ◆ 一个PCFG  $G$  是一个四元组

$$G = (V_N, V_T, S, P)$$

其中：

$V_N$  有限个非终结符号组成的集合

$V_T$  有限个终结符号组成的集合

$S$  文法的开始符号

$P$  是一组带有概率信息的重写规则组成的集合，每条规则形式如下：

$$A \rightarrow \alpha [P(A \rightarrow \alpha)]$$

$\alpha \in (V_N \cup V_T)^*$ ， $P(A \rightarrow \alpha)$  是重写规则的概率。且：

$$\sum_j P(A \rightarrow \alpha_j) = 1$$

# 概率上下文无关文法举例

$S \rightarrow NP VP$  1.0

$PP \rightarrow P NP$  1.0

$VP \rightarrow V NP$  0.7

$VP \rightarrow VP PP$  0.3

$P \rightarrow with$  1.0

$V \rightarrow saw$  1.0

$NP \rightarrow NP PP$  0.4

$NP \rightarrow astronomers$  0.1

$NP \rightarrow ears$  0.18

$NP \rightarrow saw$  0.04

$NP \rightarrow stars$  0.18

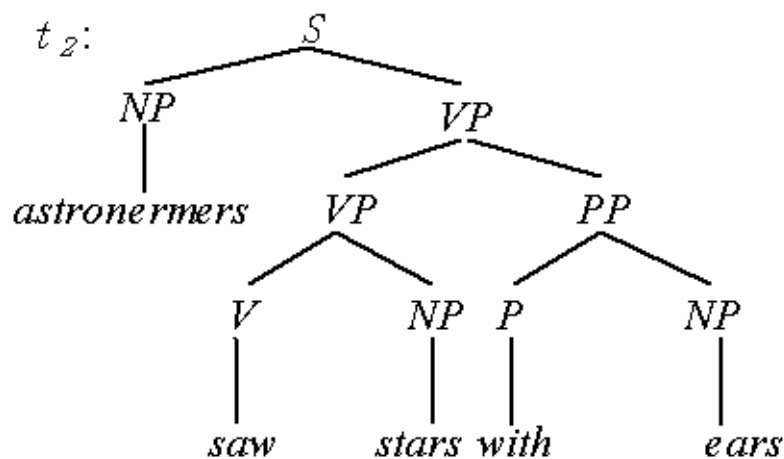
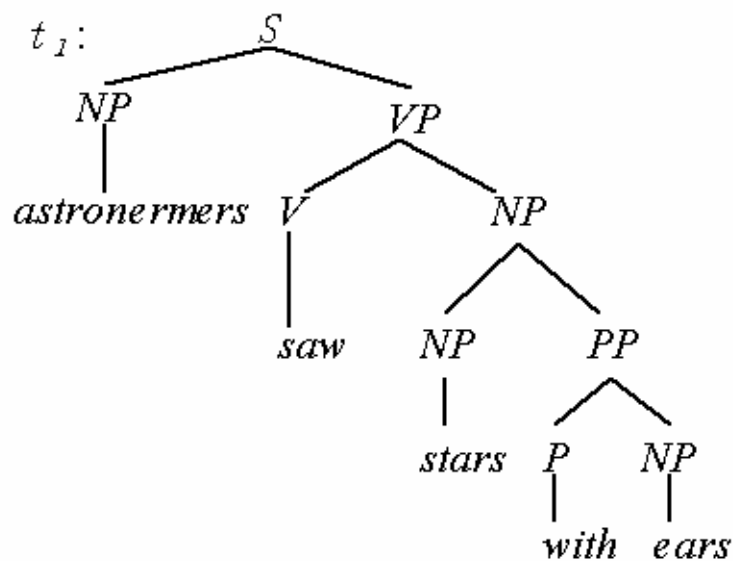
$NP \rightarrow telescopes$  0.1

# 利用PCFG计算分析树的概率

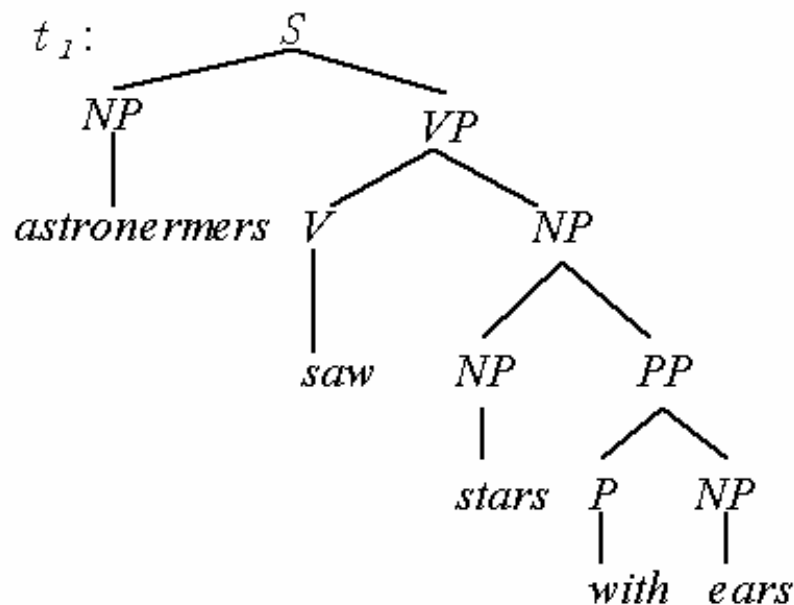
$$P(t, S) = \prod_{i=1..n} P(r_i)$$

$$P(t, S) = P(t)P(S / t) = P(t)$$

◆ 句子“astronomers saw stars with ears”的分析树



# 利用PCFG计算分析树的概率

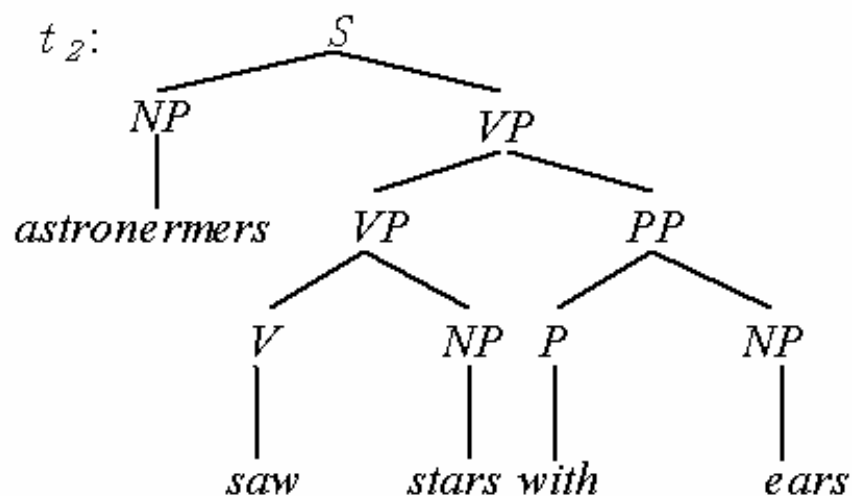


$$P(t_1) = 1.0 \times 0.1 \times 0.7 \times 1.0 \times 0.4 \times 0.18 \times 1.0 \times 1.0 \times 0.18$$

$$= 0.0009072$$

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

# 利用PCFG计算分析树的概率



$$P(t_2) = 1.0 \times 0.1 \times 0.3 \times 0.7 \times 1.0 \times 0.18 \times 1.0 \times 1.0 \times 0.18$$

$$= 0.0006804$$

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



# PCFG 用于句法分析

- ◆ 基于PCFG可以计算分析树的概率值。
- ◆ 若一个句子有多个分析树，可以依据概率值对所有的分析树进行排序。
- ◆ PCFG可以用来进行句法排歧。面对多个分析结果，选择概率最大者为最终分析结果。

# PCFG 用作 语言模型

- ◆ 基于概率上下文无关文法，一个句子 $w_{1m}$ 的概率为：

$$P(S) = \sum_t P(S, t)$$

- ◆ 句子“astronomers saw stars with ears”的概率

$$P(S) = P(t_1) + P(t_2) = 0.0009072 + 0.0006804 = 0.0015876$$

- ◆ PCFG提供了一种统计语言模型，同  $n$ -gram 模型以及 HMM 模型相比，基于PCFG的语言模型考虑了句子的结构信息，而 $n$ -gram模型以及HMM模型则认为句子是线性结构。

# PCFG的基本问题

- ① 给定一部概率上下文无关文法 $G$ ，如何计算句子 $S$ 的概率？即计算 $P(S|G)$ 的问题。（语言模型）
  - ② 给定一部概率上下文无关文法 $G$ 以及句子 $S$ ，最为可能的分析树是什么，即计算  $\operatorname{argmax}_t P(t/S, G)$  的问题。（句法分析）
  - ③ 如何为文法规则选择概率，使得训练句子的概率最大？即计算  $\operatorname{argmax}_G P(S/G)$  的问题。（模型训练）
- ◆ 可以通过计算每个分析树的概率，然后以求和或求最大值的方式解决上述第①、②个问题，缺点是效率不高。

# Chomsky范式

- ◆ 若上下文无关文法的重写规则形式如下：

$$A \rightarrow BC$$

$$A \rightarrow w$$

则该上下文无关文法为*Chomsky*范式(*CNF*)。

(即产生式右端或者是两个非终结符号，或者是一个终结符号)

- ◆ 可以证明对于任何一部上下文无关文法，都存在一个*Chomsky*范式的文法，二者弱等价。(证明以及构造方式可参见“A.V.Aho, J.D. Ullman, *the theory of parsing, translation, and compiling: volume I: parsing*, prentice hall, Inc., 1972”)
- ◆ 以下的讨论假定上下文无关文法为*Chomsky*范式。

# 符号说明

$N^i$

非终结符号

$w^i$

终结符号, 即词

$t$

分析树

$W_{1m} = w_1 w_2 \dots w_m$

待分析的句子,  $w_i$  是组成句子的词(终结符号)

$w_{ab} = w_a w_{a+1} \dots w_b$

句子的一个子串

$N^j \xRightarrow{*} w_a w_{a+1} \dots w_b$

$N^j$  推导出子串  $w_{ab}$ , 或  $N^j$  支配子串  $w_{ab}$ 。

$N_{ab}^j$

表示  $N^j$  支配句子中从位置  $a$  开始到位置  $b$  结束的子串。

# 向内变量和向外变量

◆ 为了有效解答PCFG的三个问题，定义向外变量(outside variable)和向内变量(inside variable)。

- 向外变量

$$\alpha_j(p, q) = P(w_{1(p-1)}, N_{pq}^j, w_{(q+1)m} \mid G)$$

$$N_{pq}^j \xRightarrow{*} w_{1(p-1)} N_{pq}^j w_{(q+1)m}$$

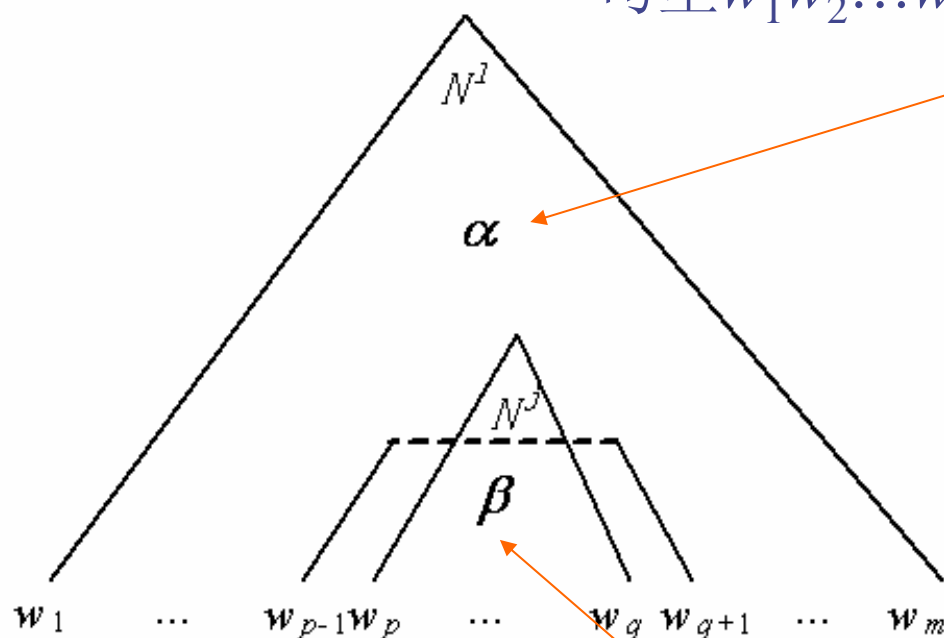
- 向内变量

$$\beta_j(p, q) = P(w_{pq} \mid N_{pq}^j, G)$$

$$N_{pq}^j \xRightarrow{*} w_{pq}$$

# 向内变量和向外变量

向外变量  $\alpha_j(p, q)$  是开始符号  $N^i$  推导出句型  $w_1 w_2 \dots w_{p-1} N^j_{pq} w_{q+1} \dots w_m$  的概率



向内变量  $\beta_j(p, q)$  为非终结符号  $N^j$  推导出词串  $w_{pq} = w_p w_{p+1} \dots w_q$  的概率。



# 计算句子 $w_{1m}$ 的概率

◆ 向内变量和 $P(w_{1m}/G)$ 的关系

$$P(w_{1m} / G) = P(N^l \overset{*}{\Rightarrow} w_{1m} / G) = P(w_{1m} / N_{1m}^l, G) = \beta_l(1, m)$$

◆ 另有

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

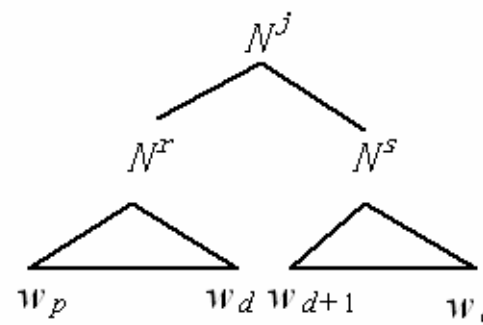


# 向内算法(Inside Algorithm)

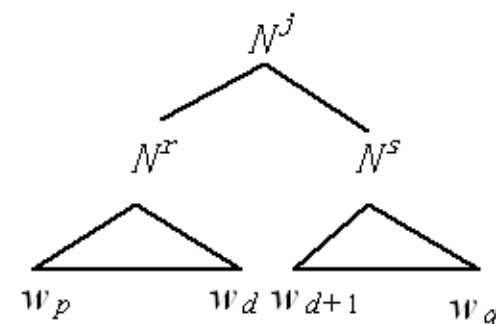
- ◆ 如何计算  $\beta_j(p, q)$ , 其中  $p < q$
- ◆ 因为限制文法为 *Chomsky* 范式, 因此第一条使用的重写规则必为

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

子串  $w_{pq}$  一定在某个位置  $d$  被分成两个部分, 使得  $N^r$  支配子串  $w_{pd}$ , 而  $N^s$  支配子串  $w_{(d+1)q}$ ,



# 向内算法



$$\beta_j(p, q) = P(w_{pq} \mid N_{pq}^j, G)$$

$$= \sum_{r,s} \sum_{d=p}^{q-1} P(w_{pd}, N_{pd}^r, w_{(d+1)q}, N_{(d+1)q}^s \mid N_{pq}^j, G)$$

$$= \sum_{r,s} \sum_{d=p}^{q-1} P(N_{pd}^r, N_{(d+1)q}^s \mid N_{pq}^j, G) \times P(w_{pd} \mid N_{pd}^r, N_{(d+1)q}^s, N_{pq}^j, G) \\ \times P(w_{(d+1)q} \mid N_{pd}^r, N_{(d+1)q}^s, N_{pq}^j, w_{pd}, G)$$

$$= \sum_{r,s} \sum_{d=p}^{q-1} P(N_{pd}^r, N_{(d+1)q}^s \mid N_{pq}^j, G) \times P(w_{pd} \mid N_{pd}^r, G) \\ \times P(w_{(d+1)q} \mid N_{(d+1)q}^s, G)$$

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

# 向内算法

① 初始化

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

② 归纳计算  $\beta_j(p, q)$  , 其中  $p < q$

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

③  $P(w_{1m}) = \beta_l(1, m)$

向内算法自底而上的递归计算句子概率。

# 向内算法计算实例

	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					$\beta_{NP} = 0.18$
	<i>Astronomers</i>	<i>saw</i>	<i>stars</i>	<i>with</i>	<i>ears</i>

$S \rightarrow NP VP$  1.0

$PP \rightarrow P NP$  1.0

$VP \rightarrow V NP$  0.7

$VP \rightarrow VP PP$  0.3

$P \rightarrow with$  1.0

$V \rightarrow saw$  1.0

$NP \rightarrow NP PP$  0.4

$NP \rightarrow astronomers$  0.1

$NP \rightarrow ears$  0.18

$NP \rightarrow saw$  0.04

$NP \rightarrow stars$  0.18

$NP \rightarrow telescopes$  0.1

单元格( $p, q$ )内为  
向内概率  $\beta_i(p, q)$

# 向外算法(outside a

◆ 如何计算  $\alpha_j(p,q)$ ?

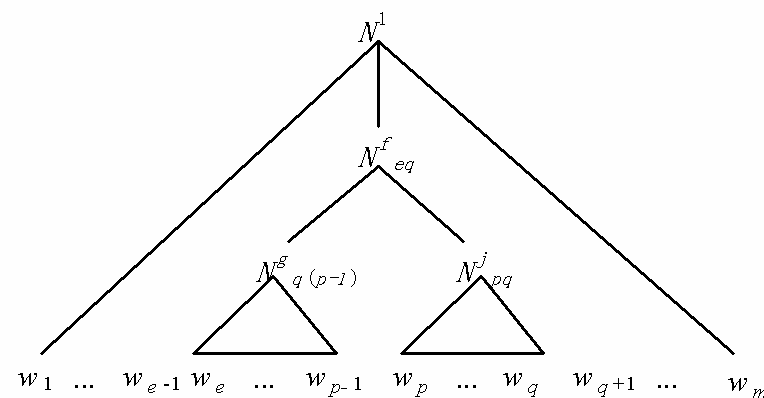
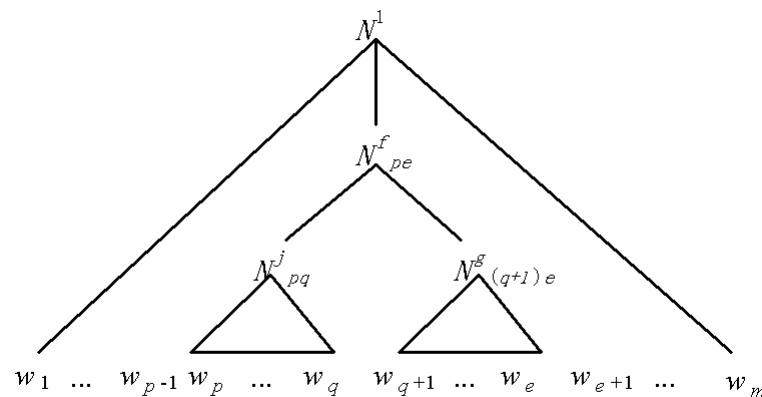
◆ 两种情况

■  $N^j$  是某个结点的左子女

◆  $N^f \rightarrow N^j N^g$

■  $N^j$  是某个结点的右子女

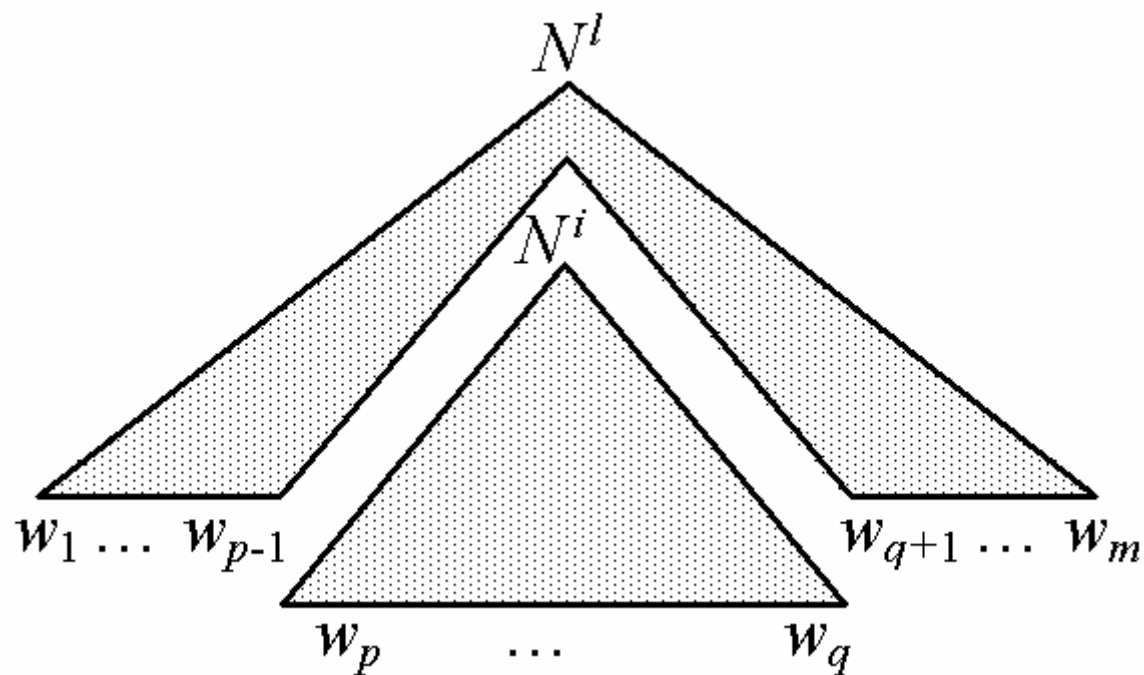
◆  $N^f \rightarrow N^g N^j$



# 向外算法

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

# 向外算法



$$\begin{aligned}
 P(w_{1:m} / G) &= \sum_j P(w_{1:(p-1)}, w_{pq}, w_{(q+1):m}, N_{pq}^j / G) \\
 &= \sum_j P(w_{1:(p-1)}, N_{pq}^j, w_{(q+1):m} / G) \times P(w_{pq} / P(w_{1:(p-1)}, N_{pq}^j, w_{(q+1):m}, G)) \\
 &= \sum_j \alpha_j(p, q) \beta_j(p, q)
 \end{aligned}$$

# 向外算法

向外算法自顶向下  
递归计算句子的概率

## ① 初始化

$$\begin{aligned}\alpha_l(1, m) &= 1 \\ \alpha_j(1, m) &= 0, \quad \text{当 } j \neq l \text{ 时}\end{aligned}$$

## ② 递归计算

$$\begin{aligned}\alpha_j(p, q) &= \left[ \sum_{f, g} \sum_{e=q+1}^m \alpha_f(p, e) P(N^f \rightarrow N^j N^g) \beta_g(q+1, e) \right] \\ &\quad + \left[ \sum_{f, g} \sum_{e=1}^{p-1} \alpha_f(e, q) P(N^f \rightarrow N^g N^j) \beta_g(e, p-1) \right]\end{aligned}$$

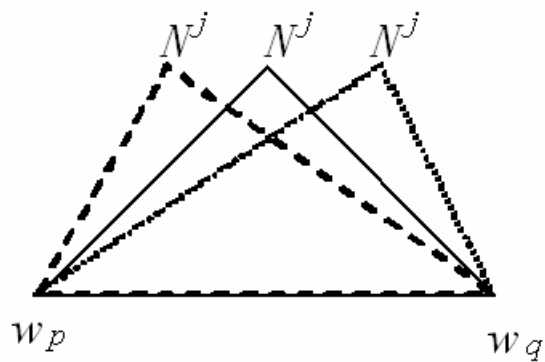
## ③

$$P(w_{1m} / G) = \sum_j \alpha_j(p, q) \beta_j(p, q)$$



# 寻找最佳的分析树

- ◆ PCFG的第二个基本问题是在给定文法 $G$ 和句子 $w_{1m}$ 的前提下，如何有效找出最为可能的分析树，这可以通过韦特比算法求得。
- ◆ 韦特比变量  $\delta_i(p, q)$   
定义该变量为子树  $N_{pq}^i$  的最大概率，即所有  $N^i$  支配  $w_{pq}$  的子树中概率最大的子树。



# 韦特比算法(Viterbi Algorithm)

## ① 初始化

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

## ② 归纳计算

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

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

## ③ 归纳终止

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

## ④ 按照下面的步骤构造最为可能的分析树，即 $\hat{t}$ 。

(1)  $\hat{t}$  的根结点为  $N_{1m}^l$ ，因为  $N^l$  是文法的开始符号。

(2) 若  $N_{pq}^i$  是  $\hat{t}$  的一个内部结点且  $\psi_i(p, q) = (j, k, r)$ ，则  $N_{pq}^i$  的左儿子结点是  $N_{pr}^j$ ，右儿子结点是  $N_{(r+1)q}^k$ 。

# 模型训练

- ◆ 如何为语法规则选择概率，使得训练句子的概率最大？也就是如何得到语法规则的概率的问题。
- ◆ 有指导训练
- ◆ 无指导训练， 向内向外算法

# 模型训练

- ◆ 有指导的训练

- ◆ 树库(Treebank), 是标记了句法树结构的语料库。

$$P(RHS | LHS) = \frac{Count(LHS \rightarrow RHS)}{Count(LHS)}$$

- ◆ 树库的构建的工作量巨大, 耗时耗力, 但在没有可靠的无指导训练技术的前提下, 树库的构造必须进行
- ◆ 美国宾夕法尼亚大学一直致力于树库的构建工作, 其构建的树库被称作*Penn Treebank*。其中英文树库规模较大、汉语树库的规模较小
- ◆ *Penn treebank* 尽管规模很小, 但为统计句法分析研究提供了一个很好的基础。

# 向内向外算法(inside-outside算法)

- ◆ 无指导训练算法是IO算法
- ◆ 同Baum-Welch 算法类似，IO算法也是一个反复迭代、逐步求精的算法。
- ◆ 通常要首先给定一组不准确的参数，以反复迭代计算的方式调整模型参数，最终使参数稳定在一个可以接受的精度。
- ◆ IO算法不能保证求得最优模型，一般能得到一个局部最优模型。

# 基于PCFG的句法分析

- ◆ PCFG把概率引入上下文无关文法，将统计方法和规则方法进行了有效的融合，具有十分重要的意义，但是PCFG缺陷也是十分明显的。
- ◆ 作为一种统计句法分析方法，基于PCFG的句法分析效果有限。
  - PCFG没有考虑结构之间的依存关系。
  - PCFG没有考虑词汇对句法结构的影响。
- ◆ 针对基于PCFG的句法分析表现出来的缺陷，目前已提出几个的统计句法分析模型。统计句法分析近年来取得了显著的进展。分析精度、速度都不比基于纯粹规则的句法分析逊色。（尤其是对于英语的分析）



# 结构依存关系

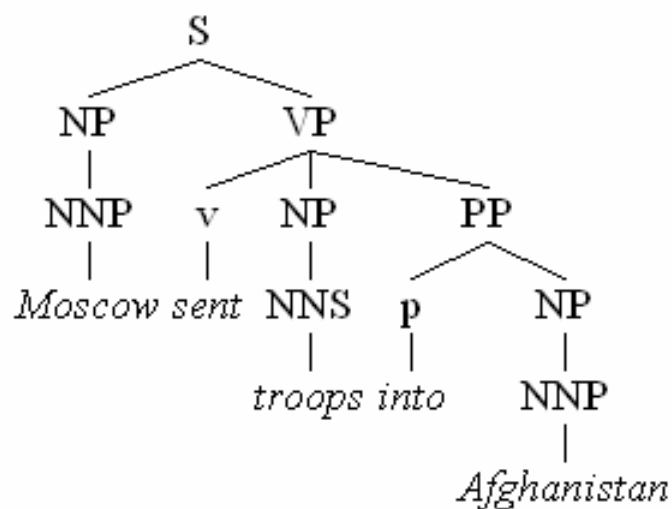
## ◆ 代词作为主语的可能性高于作为宾语的可能性

- 主题与述题
- 主题通常是已知信息，在句子中做主语
- 代词用来指代已经陈述过的已知信息
- 述题引入新信息
- 统计数据，来自Switchboard corpus:
  - ◆ 陈述句中91%的主语是代词
  - ◆ 陈述句中66% 的直接宾语不是代词

*NP* → *Pron*   *NP* → *Det Noun* 的概率应和其在句中所处的位置有关，处在*VP*前后概率应该不同。

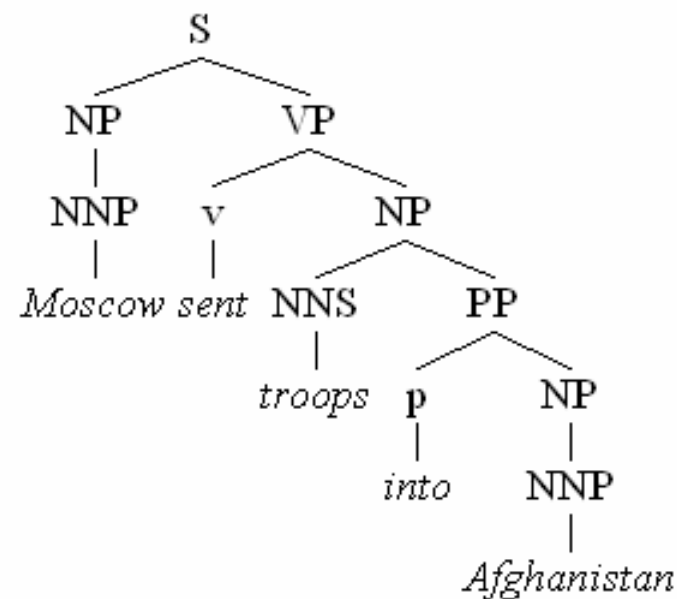
# 词汇依存关系

*Moscow sent troops into Afghanistan*



正确的句法树

*PP* 可以修饰 *VP*, *PP* 也可以修饰 *NP*, 但不同的动词 *PP* 修饰 *NP* 和 *VP* 的可能性并不相同。



错误的句法树



# 基于PCFG的统计语言模型

- ◆ 作为一种统计语言模型，其效果甚至还不如n-gram模型以及HMM模型(原因同上)

the green banana

the green time

# 句法分析的评价

## ◆ 标记准确率

$$\text{Labeled Precision} = \frac{\text{number of correct constituents in proposed parse}}{\text{number of constituents in proposed parse}}$$

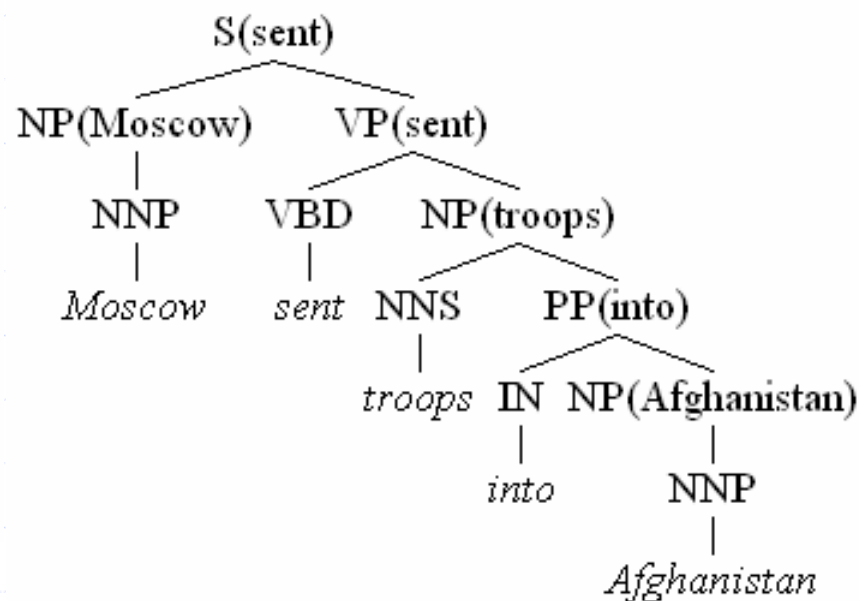
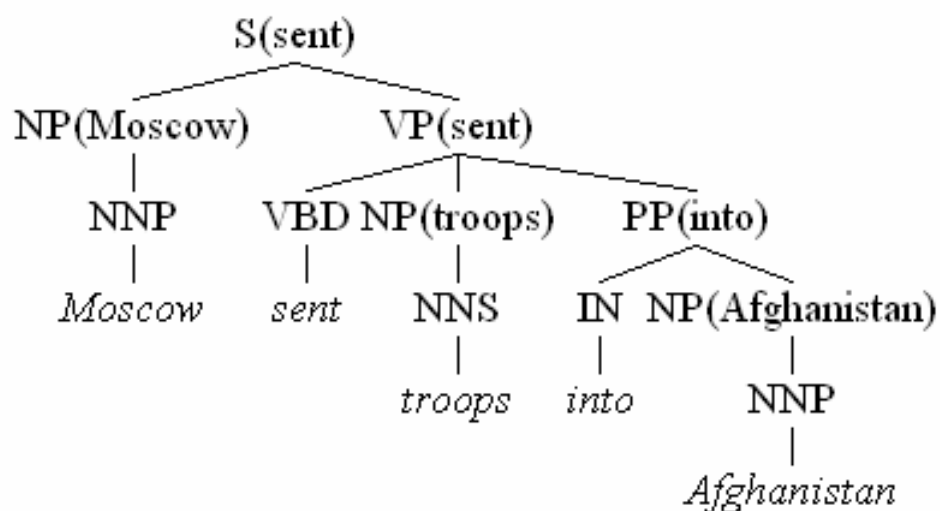
## ◆ 标记召回率

$$\text{Labeled Recall} = \frac{\text{number of correct constituents in proposed parse}}{\text{number of constituents in treebank parse}}$$

## ◆ 括号交叉数

$$\text{Crossing Brackets} = \text{number of constituents which violate constituent boundaries with a constituent in the treebank parse}$$

# 词汇化的句法分析简介



# 词汇化PCFG

$S(\text{sent}) \rightarrow NP(\text{Moscow}) VP(\text{sent})$

$NP(\text{Moscow}) \rightarrow NNP(\text{Moscow})$

$VP(\text{sent}) \rightarrow VBD(\text{sent}) NP(\text{troops}) PP(\text{into})$

$NP(\text{troops}) \rightarrow NNS(\text{troops})$

$PP(\text{into}) \rightarrow IN(\text{into}) NP(\text{Afghansitan})$

$NP(\text{Afghanistan}) \rightarrow NNP(\text{Afghanistan})$

$NNP(\text{Moscow}) \rightarrow \text{Moscow}$

$NNP(\text{Afghanistan}) \rightarrow \text{Afghanistan}$

$IN(\text{into}) \rightarrow \text{into}$

$VBD(\text{sent}) \rightarrow \text{sent}$

$NNS(\text{troops}) \rightarrow \text{troops}$

非终结符号的数量激烈膨胀。

参数估计时，数据稀疏问题十分严重！

# 词汇化PCFG

- ◆ 为避免数据稀疏问题，概率计算需要分解
- ◆ 在词汇化PCFG中

$$P(h) \rightarrow L_n(l_n) \dots L_1(l_1) H(h) R_1(r_1) \dots R_m(r_m)$$

其中， $H$ 是短语的中心成分

$h$ 是成分的中心词

$L_i$ 和 $R_i$ 分别是中心成分左右的修饰性成分

$l_i$ 和 $r_i$ 分别是左右修饰成分的中心词

- ◆ 在左右两端增加 $STOP$   
令  $L_{n+1} = STOP$   $R_{n+1} = STOP$

# 词汇化PCFG

$$\begin{aligned} &P(L_{n+1}(l_{n+1})...L_1(l_1)H(h)R_1(r_1)...R_{m+1}(r_{m+1}) \mid P(h)) = \\ &P_h(H \mid P(h)) \times \\ &\prod_{i=1...n+1} P_l(L_i(l_i) \mid L_1(l_1)...L_{i-1}(l_{i-1}), P(h), H) \times \\ &\prod_{j=1...m+1} P_r(R_j(r_j) \mid L_1(l_1)...L_{n+1}(l_{n+1}), R_1(r_1)...R_{j-1}(r_{j-1}), P(h), H) \end{aligned}$$

◆ 作如下的独立性假设

$$P_l(L_i(l_i) \mid L_1(l_1)...L_{i-1}(l_{i-1}), P(h), H) = P_l(L_i(l_i) \mid P(h), H)$$

$$P_r(R_j(r_j) \mid L_1(l_1)...L_{n+1}(l_{n+1}), R_1(r_1)...R_{j-1}(r_{j-1}), P(h), H) = P_r(R_j(r_j) \mid P(h), H)$$

# 词汇化PCFG

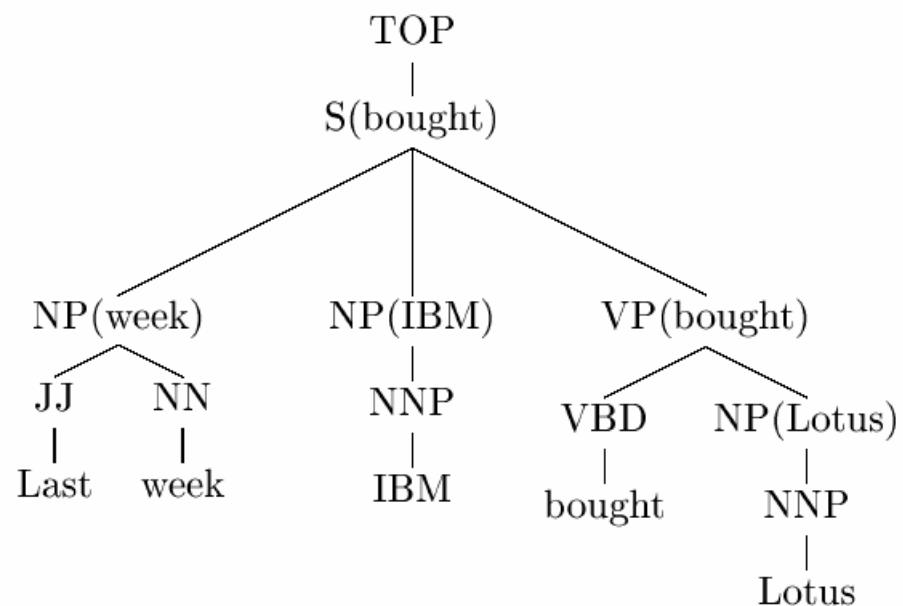
$$P(L_{n+1}(l_{n+1})...L_1(l_1)H(h)R_1(r_1)...R_{m+1}(r_{m+1}) | P(h)) = P(H | P(h)) \times \prod_{i=1...n+1} P_L(L_i(l_i) | P(h), H) \times \prod_{j=1...m+1} P_R(R_j(r_j) | P(h), H)$$

◆ 句法树可以视为自顶向下按照三个步骤产生：

1. 以概率  $P_H(H | P, h)$  生成短语的中心成分标签；
2. 以概率  $\prod_{i=1...n+1} P_L(L_i(l_i) | P, h, H)$  生成中心词左面的修饰性成分，其中  $L_{n+1}(l_{n+1}) = STOP$ 。  $STOP$  可视作一个特殊的非终结符号，模型在生成  $STOP$  标记时结束生成。
3. 以概率  $\prod_{i=1...m+1} P_R(R_i(r_i) | P, h, H)$  生成中心词右面的修饰性成分，其中  $R_{m+1}(r_{m+1}) = STOP$ 。

# 词汇化PCFG

◆ 例子



$S(bought) \rightarrow NP(week)NP(IBM)VP(bought)$

$P_h(VP \mid S, bought) \times P_i(NP(IBM) \mid S, VP, bought) \times$   
 $P_i(NP(week) \mid S, VP, bought) \times P_r(STOP \mid S, VP, bought) \times$   
 $P_r(STOP \mid S, VP, bought)$