

计算语言学

第 8 讲 句法分析（二）

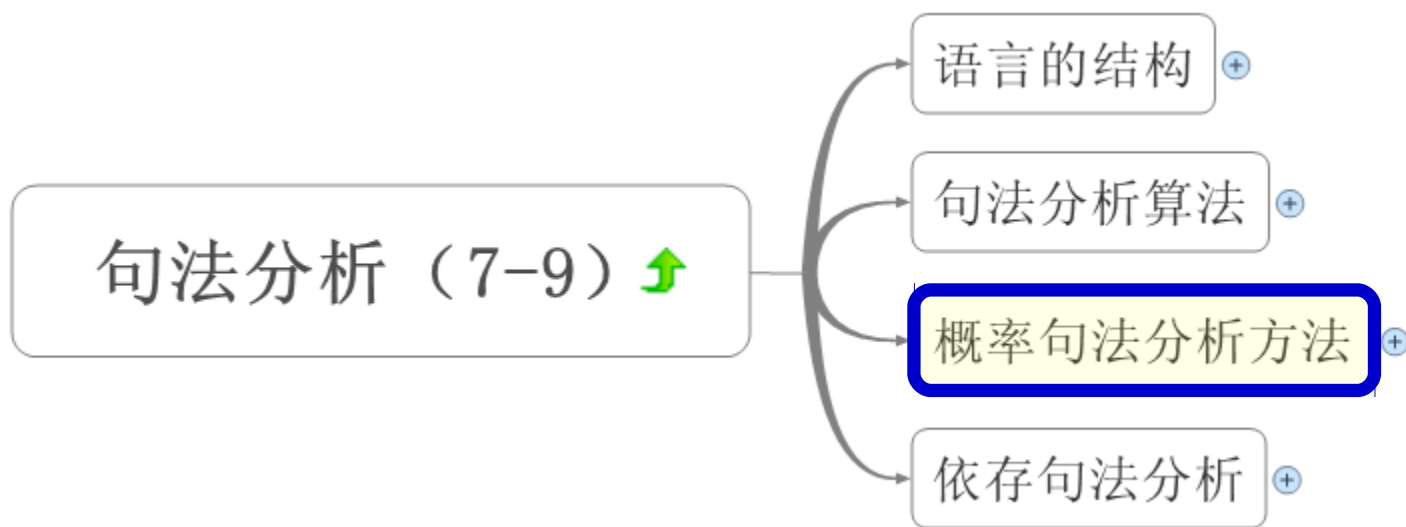
刘群

中国科学院计算技术研究所

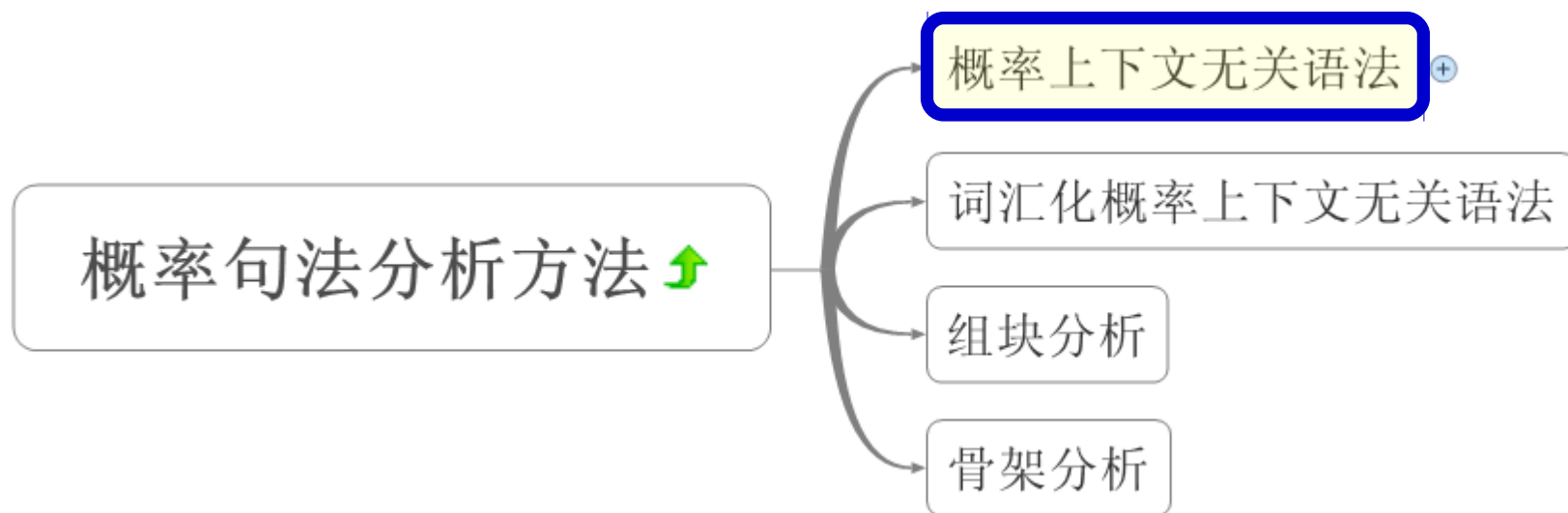
liuqun@ict.ac.cn

中国科学院研究生院 2012 年春季课程讲义

内容提要



内容提要



概率在句法分析中的作用

- 决定最有可能的句子（当作语言模型）
- 加速分析器
- 排歧

句法模型与语言模型

- Parsing Model （条件概率）

$$P(t|s, G), \text{ where } \sum_t P(t|s, G) = 1$$

$$\hat{t} = \operatorname{argmax}_t P(t|s, G)$$

- Language Model （联合概率）

$$\sum_{\{t: \text{yield}(t) \in L\}} P(t) = 1 \quad P(s) = \sum_t P(s, t) = \sum_{\{t: \text{yield}(t) = s\}} p(t)$$

$$\hat{t} = \operatorname{argmax}_t p(t|s) = \operatorname{argmax}_t \frac{P(t, s)}{P(s)} = \operatorname{argmax}_t P(t, s)$$

最简单的概率语法模型

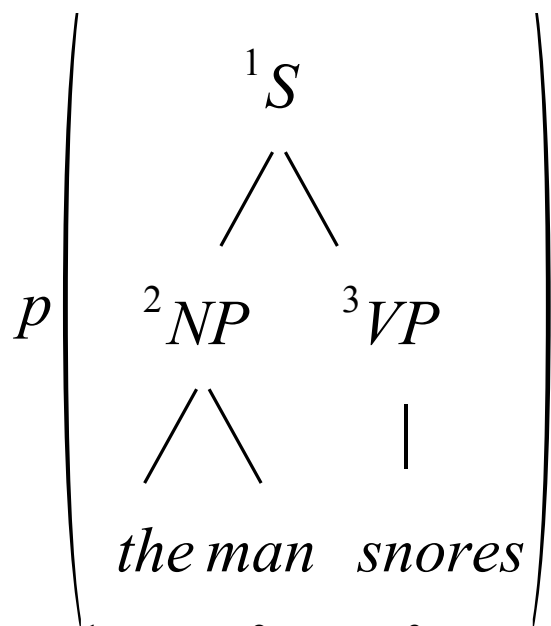
- PCFG : Probabilistic CFG 概率上下文无关语法
SCFG : Stochastic CFG 随机上下文无关语法

- CFG 的简单概率拓广

$$\sum_{\alpha} P(A \rightarrow \alpha) = 1$$

- 基本假设
 - 位置无关 (Place invariance)
 - 上下文无关 (Context-free)
 - 祖先无关 (Ancestor-free)
- 分析树的概率等于所有施用规则概率之积

例子



$$\begin{aligned}
 &= P({}^1S_{13} \rightarrow {}^2NP_{12} {}^3VP_{33}, {}^2NP_{12} \rightarrow the_1 man_2, {}^3VP_{33} \rightarrow snores_3) \\
 &= P({}^1S_{13} \rightarrow {}^2NP_{12} {}^3VP_{33}) P({}^2NP_{12} \rightarrow the_1 man_2 \mid {}^1S_{13} \rightarrow {}^2NP_{12} {}^3VP_{33}) \\
 &\quad P({}^3VP_{33} \rightarrow snores_3 \mid {}^1S_{13} \rightarrow {}^2NP_{12} {}^3VP_{33}, {}^2NP_{12} \rightarrow the_1 man_2) \\
 &= P({}^1S_{13} \rightarrow {}^2NP_{12} {}^3VP_{33}) P({}^2NP_{12} \rightarrow the_1 man_2) p({}^3VP_{33} \rightarrow snores_3)
 \end{aligned}$$

概率上下文无关语法——示例

CFG

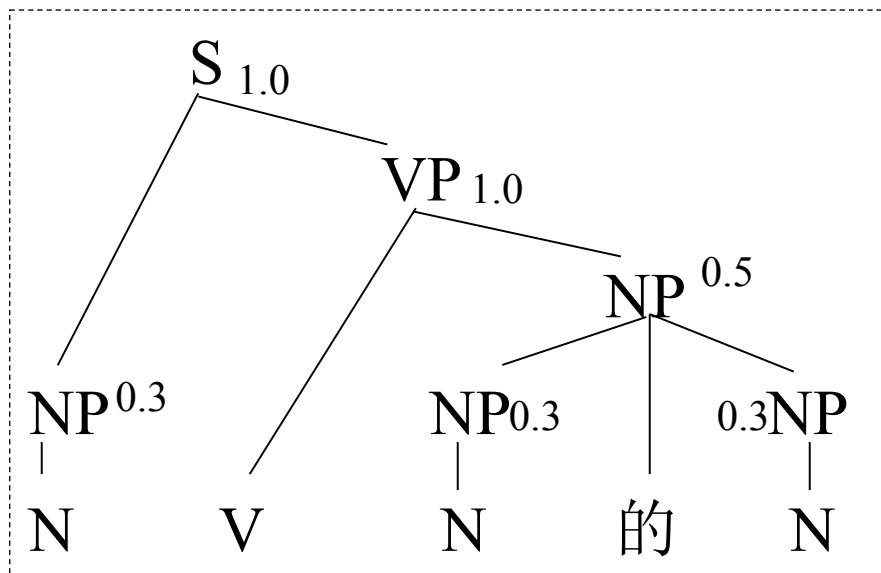
$S \rightarrow NP VP$
 $VP \rightarrow V NP$
 $NP \rightarrow N$
 $NP \rightarrow NP \text{ 的 } NP$
 $NP \rightarrow VP \text{ 的 } NP$

PCFG

$S \rightarrow NP VP$ 1.0
 $VP \rightarrow V NP$ 1.0
 $NP \rightarrow N$ 0.3
 $NP \rightarrow NP \text{ 的 } NP$ 0.5
 $NP \rightarrow VP \text{ 的 } NP$ 0.2

分析树及其概率

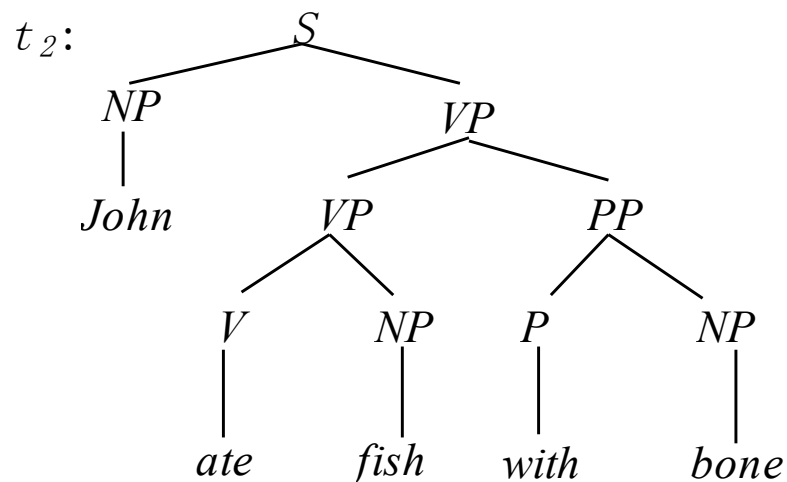
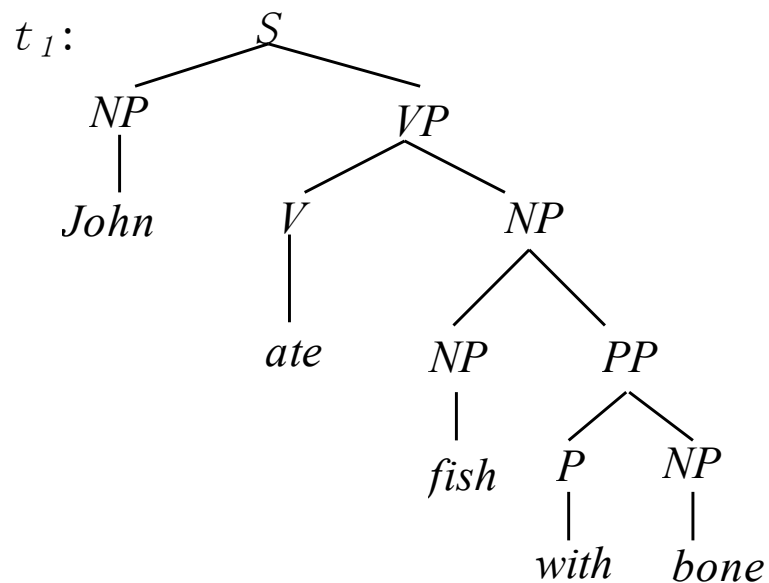
老虎 咬死了 猎人 的 狗
| | | | |
N V N 的 N



$$P(S)=1.0\times 0.3\times 1.0\times 0.3\times 0.5\times 0.3$$
$$=0.0135$$

用概率来帮助判别歧义

sentence = “*John ate fish with bone*”



分析树的概率与句子的概率

$S \rightarrow NP VP$	1.0	$NP \rightarrow NP PP$	0.4
$PP \rightarrow P NP$	1.0	$NP \rightarrow John$	0.1
$VP \rightarrow V NP$	0.7	$NP \rightarrow bone$	0.18
$VP \rightarrow VP PP$	0.3	$NP \rightarrow star$	0.04
$P \rightarrow with$	1.0	$NP \rightarrow fish$	0.18
$V \rightarrow ate$	1.0	$NP \rightarrow telescope$	0.1

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

$$P(t_2) = 1.0 \times 0.1 \times 0.3 \times 0.7 \times 1.0 \times 0.18 \times 1.0 \times 1.0 \times 0.18 = 0.0006804$$

$$P(sentence) = P(t_1) + P(t_2) = 0.0015876$$

PCFG 的三个基本问题

- 一个语句 $W=w_1w_2\dots w_n$ 的 $P(W|G)$, 也就是产生语句 W 的概率? (语言模型)

评估问题

向内算法

- 在语句 W 是歧义的情况下, 如何快速选择最佳的语法树? (句法分析)

解码问题

韦特比算法

- 如何调节 G 的概率参数, 使得 $P(W|G)$ 最大? (学习)

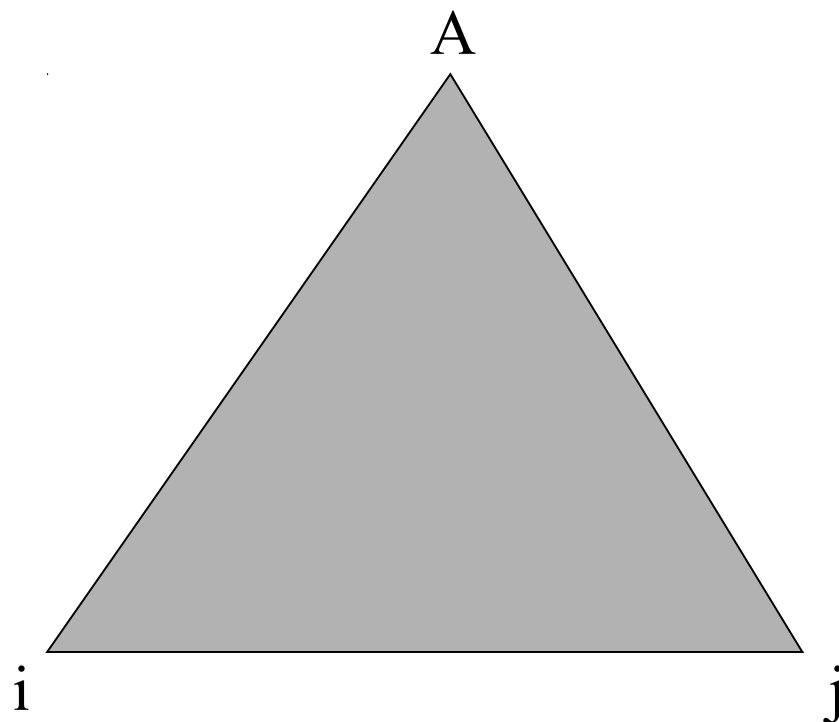
学习问题

向内向外算法

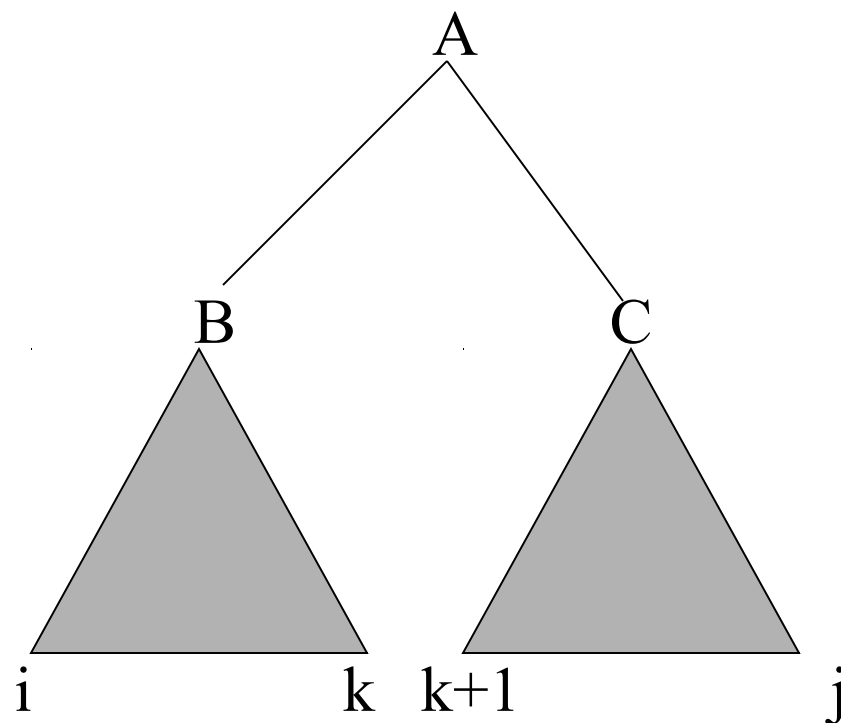
向内变量

- 定义向内变量 $\alpha_{ij}(A) = P(A \Rightarrow w_i w_{i+1} \dots w_j)$

向内变量的递归计算



向内变量的递归计算



向内变量

- 定义向内变量 $\alpha_{ij}(A) = P(A \Rightarrow w_i w_{i+1} \dots w_j)$
 - 动态规划递归公式

$$\alpha_{ii}(A) = P(A \rightarrow w_i)$$

$$\alpha_{ij}(A) = \sum_{B,C} \sum_{i \leq k \leq j} P(A \rightarrow BC) \alpha_{ik}(B) \alpha_{(k+1)j}(C)$$

- 特别地,

$$P(W \mid G) = \alpha_{1n}(S)$$

韦特比算法

- 韦特比变量 $\gamma_{ij}(A)$ 为非终结符 A 经由某一推导而产生 $w_i w_{i+1} \dots w_j$ 的最大概率, $\psi(A)$ 为最佳推导。

– 动态规划公式

$$\gamma_{ii}(A) = \max P(A \Rightarrow w_i)$$

$$\gamma_{ij}(A) = \max_{B, C \in N; i \leq k \leq j} P(A \rightarrow BC) \gamma_{ik}(B) \gamma_{(k+1)j}(C)$$

$$\psi_{ij}(A) = \operatorname{argmax}_{B, C \in N; i \leq k \leq j} P(A \rightarrow BC) \gamma_{ik}(B) \gamma_{(k+1)j}(C)$$

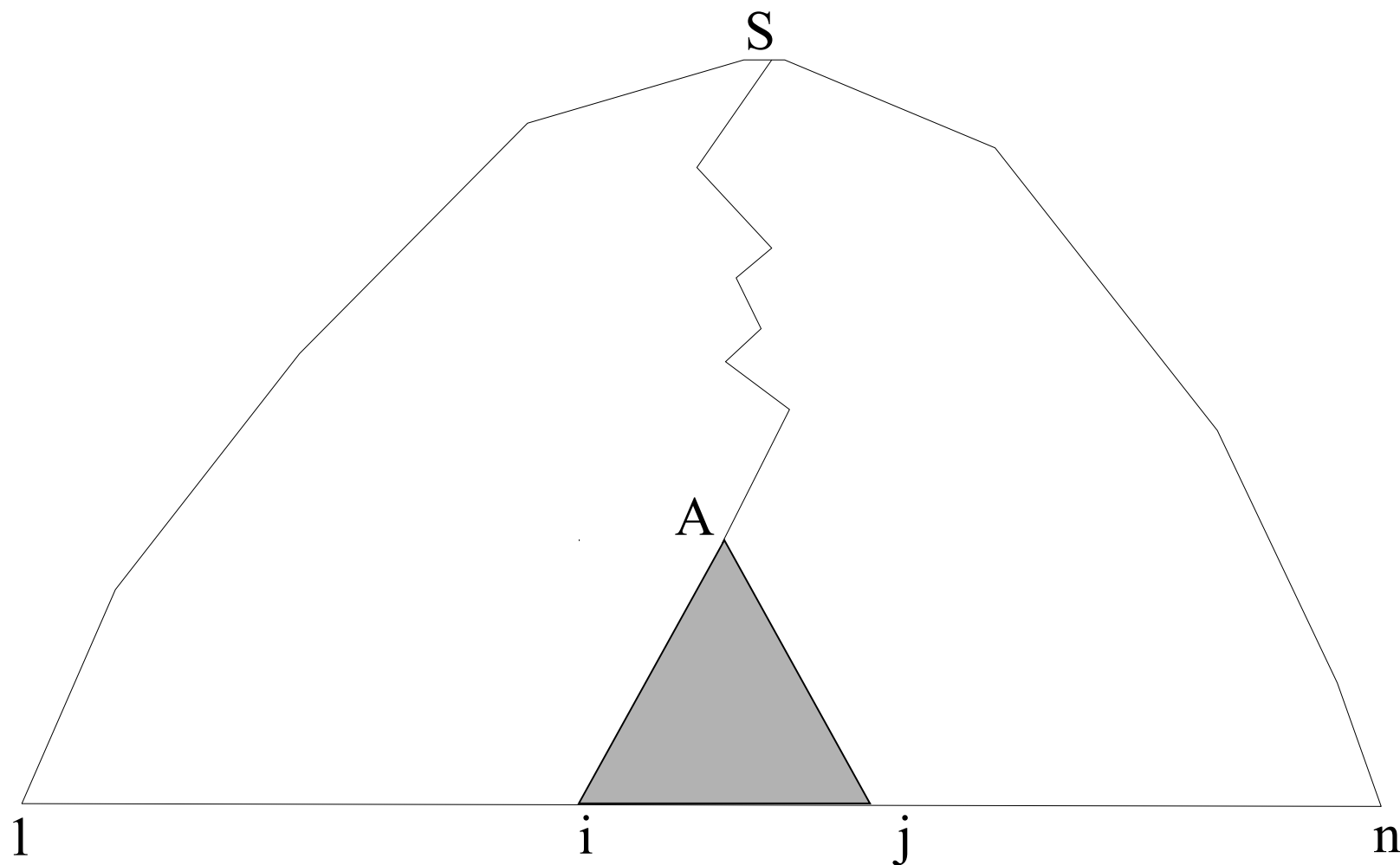
– 特别地,

$$\gamma_{1n}(S)$$

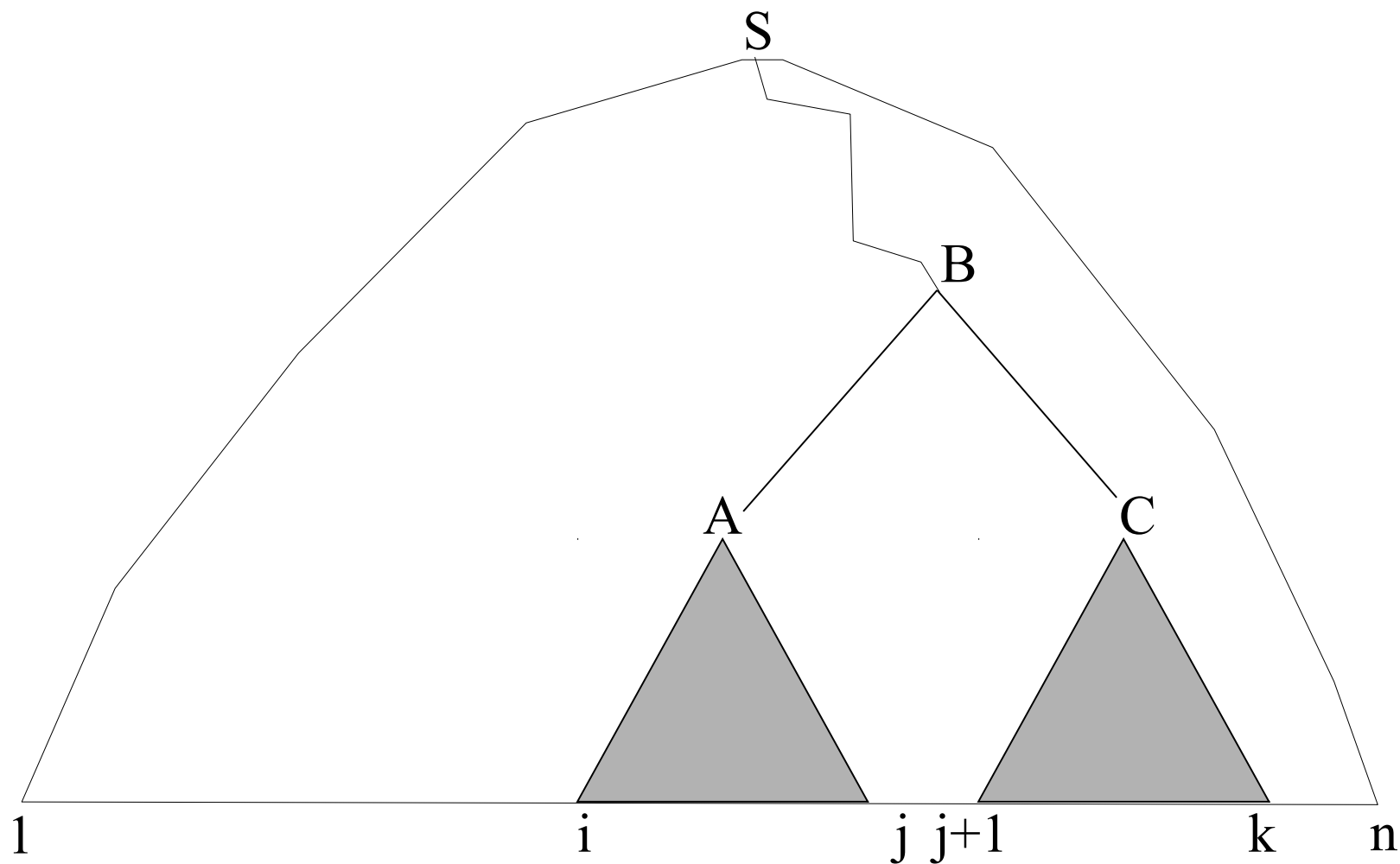
向外变量

- 向外变量 $\beta_{ij}(A) = P(S \Rightarrow w_1 \dots w_{i-1} A w_{j+1} \dots w_n)$

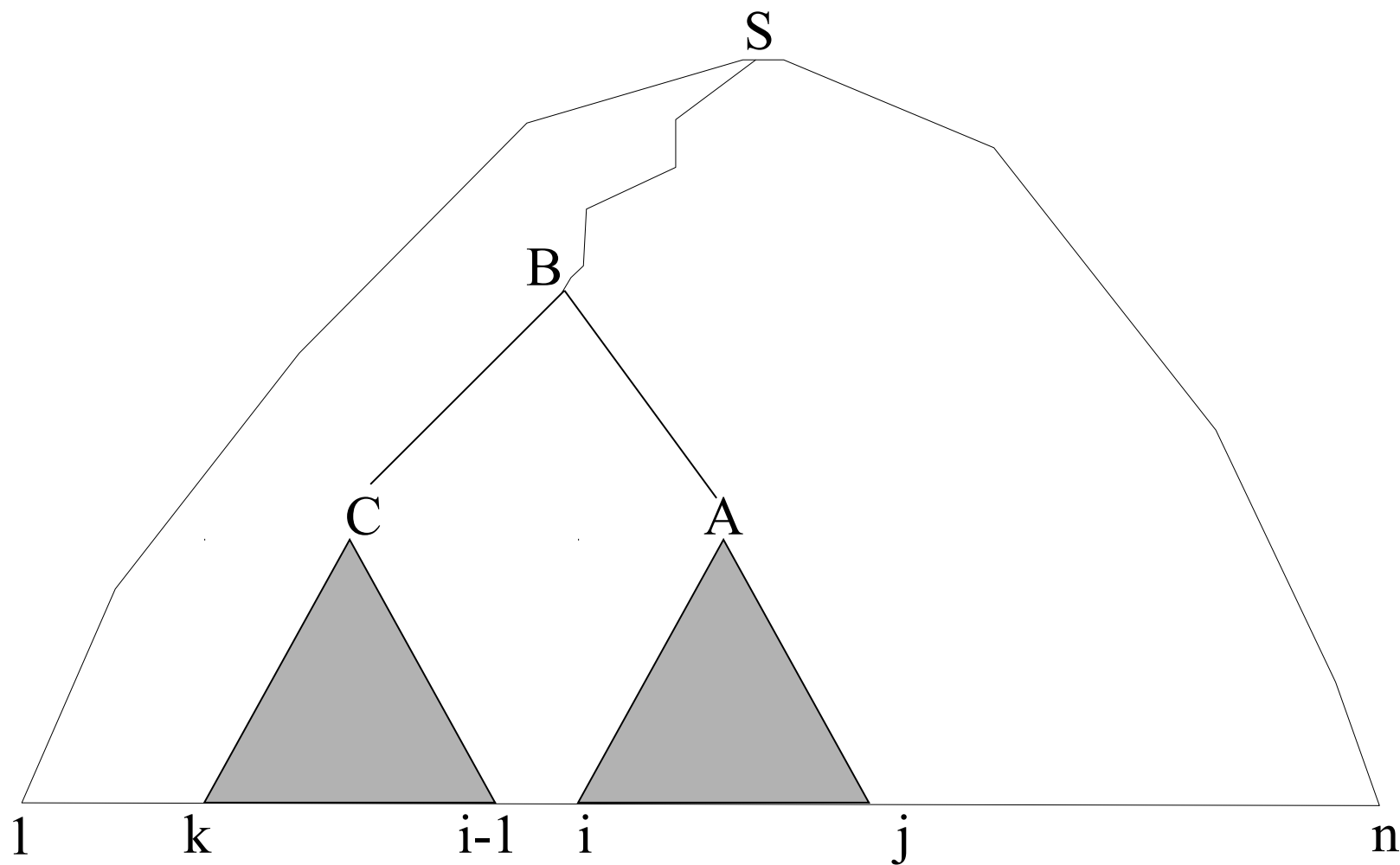
向外变量的递归计算



向外变量的递归计算



向外变量的递归计算



向外变量

- 向外变量 $\beta_{ij}(A) = P(S \Rightarrow w_1 \dots w_{i-1} A w_{j+1} \dots w_n)$
— 动态规划递归公式

$$\beta_{1n}(A) = \delta(A, S)$$

$$\begin{aligned} \beta_{ij}(A) = & \sum_{B, C} \sum_{k > j} P(B \rightarrow AC) \alpha_{j+1, k}(C) \beta_{ik}(B) \\ & + \sum_{B, C} \sum_{k < j} P(B \rightarrow CA) \alpha_{k, i-1}(C) \beta_{kj}(B) \end{aligned}$$

向内向外算法

- EM 算法运用于 PCFG 的参数估计的具体算法。
 - 初始化：随机地给 $P(A \rightarrow \mu)$ 赋值，使得 $\sum_{\mu} P(A \rightarrow \mu) = 1$. 由此得到语法 G_0 . $i \leftarrow 0$.
 - EM 步骤：
 - E 步骤：计算期望值 $C(A \rightarrow BC)$ 和 $C(A \rightarrow a)$
 - M 步骤：用 E- 步骤所得的期望值，利用：
$$\bar{P}(A \rightarrow \mu) = \frac{C(A \rightarrow \mu)}{\sum_{\mu} C(A \rightarrow \mu)}$$
重新估计 $P(A \rightarrow \mu)$, 得到语法 G_{i+1}
 - 循环计算 : $i++$, 重复 EM 步骤，直至 $P(A \rightarrow \mu)$ 收敛 .

向内向外算法—— 语法规则使用次数的期望值

$$C(A \rightarrow BC)$$

$$= \sum_{1 \leq i \leq k \leq j \leq n} P(A_{ij}, B_{ik}, C_{k+1,j} \mid w_1 \dots w_n, G)$$

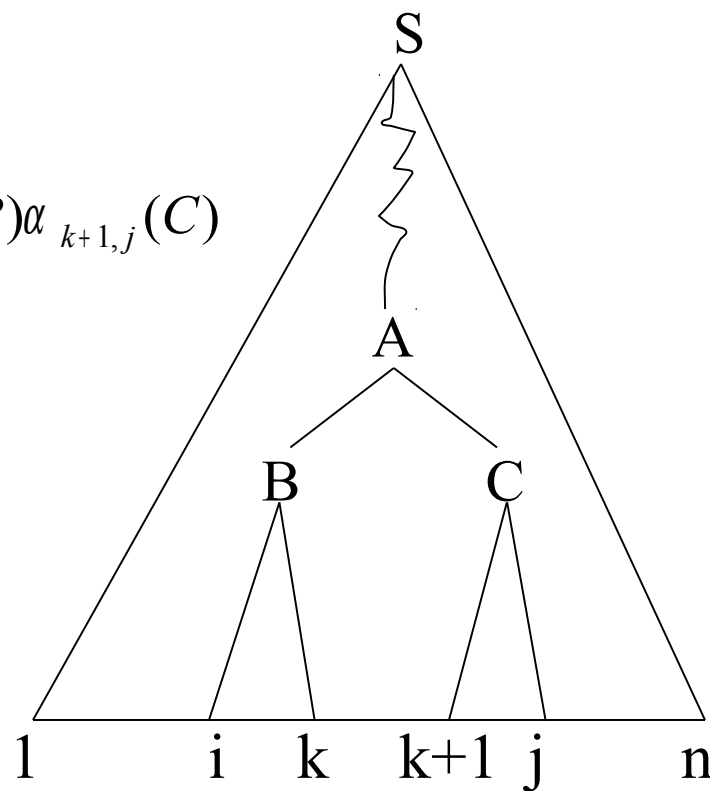
$$= \frac{1}{P(w_1 \dots w_n \mid G)} \sum_{1 \leq i \leq k \leq j \leq n} \beta_{ij}(A) P(A \rightarrow BC) \alpha_{ik}(B) \alpha_{k+1,j}(C)$$

$$C(A \rightarrow a)$$

$$= \sum_{1 \leq i \leq n} P(A_{ii} \mid w_1 \dots w_n, G)$$

$$= \frac{1}{P(w_1 \dots w_n \mid G)} \sum_{1 \leq i \leq n} P(A_{ii}, w_1 \dots w_n \mid G)$$

$$= \frac{1}{P(w_1 \dots w_n \mid G)} \sum_{1 \leq i \leq n} \beta_{ii}(A) P(A \rightarrow a) \delta(a, w_i)$$



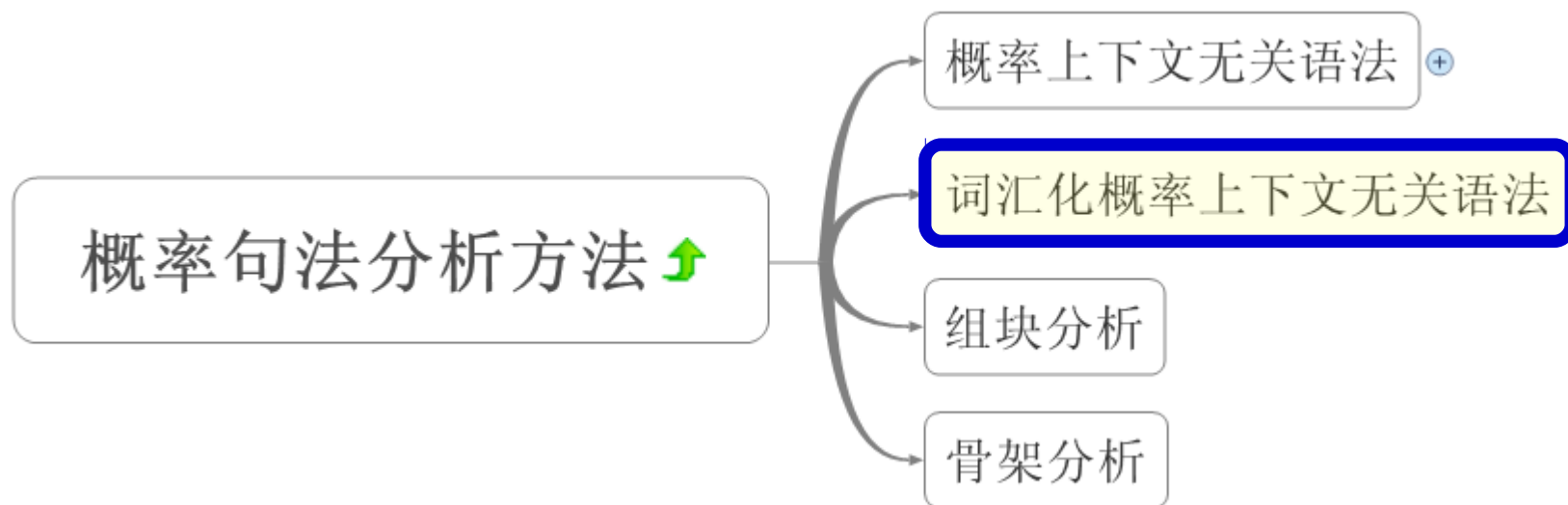
PCFG 的优点

- 化解结构歧义 (structurally different parses)
- 加速语法分析 (尽早删除小概率子结构)
- 增强分析器鲁棒性 (use of low probabilities)
- 定量比较语法 (language model)
- 便于语法归纳 (grammar induction)

PCFG 的缺点

- 合理性差 (单纯依据结构给出概率估计)
- 不如 n 元语法 (importance of lexical context)
- 明显的偏向性 (smaller tree, small number of expansions will be favored)

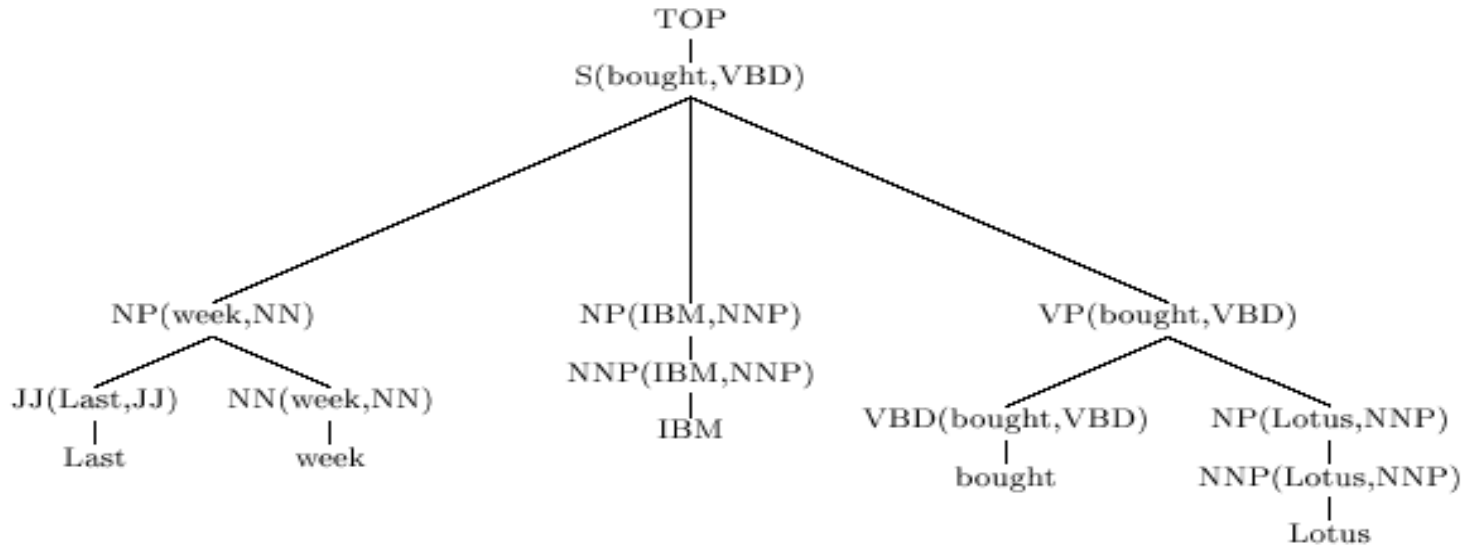
内容提要



Lexicalized PCFG

- 词汇化上下文概率上下文无关语法
Lexicalized PCFG
- 每一个非终结符被关联到一个中心词 w 和一个中心词形 t

Lexicalized PCFG



Internal Rules:

TOP	→	S(bought, VBD)		
S(bought, VBD)	→	NP(week, NN)	NP(IBM, NNP)	VP(bought, VBD)
NP(week, NN)	→	JJ>Last, JJ)	NN(week, NN)	
NP(IBM, NNP)	→	NNP(IBM, NNP)		
VP(bought, VBD)	→	VBD(bought, VBD)	NP(Lotus, NNP)	
NP(Lotus, NNP)	→	NNP(Lotus, NNP)		

Lexical Rules:

JJ>Last, JJ)	→	Last
NN(week, NN)	→	week
NNP(IBM, NNP)	→	IBM
VBD(bought, VBD)	→	bought
NNP(Lotus, NNP)	→	Lotus

Collins Model (1)

- Michael Collins. Head-Driven Statistical Models for Natural Language Parsing. PhD thesis, University of Pennsylvania, 1999.
- 复杂程度递增的三个模型
 - Model 1 : lexical dependency
 - Model 2 : complement/adjunct distinction, subcat frame
 - Model 3: Wh-movement

Collins Model (2) 中心成分的生成

- 两个组成部分：词汇中心和结构中心
- 词汇中心：中心词 (*hw*) 和中心词词性标记 (*ht*)
- 结构中心：中心成分的短语标记 (*H*)
- 中心成分的生成
 - 首先生成词汇中心
 - 其次生成结构中心

Collins Model (3) 修饰成分的扩展

- 基本规则形式

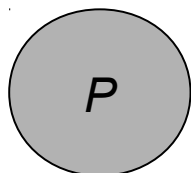
$$P(h) \rightarrow L_m(l_m) \dots L_1(l_1) H(h) R_1(r_1) \dots R_n(r_n)$$

- 修饰成分 $M_i(m_i)$ 扩展

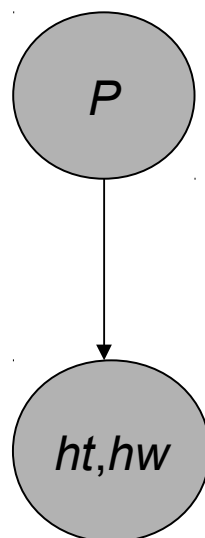
- 修饰成分词汇中心的扩展：根据 $P(h)$ 和 $H(h)$ 估计 M_i 的词汇中心
- 结构扩展：估计 M_i 的标记

$M_i(m_i)$ 为 $L_i(l_i)$ 和 $R_i(r_i)$ 的统称

Collins Model (3a) 图示

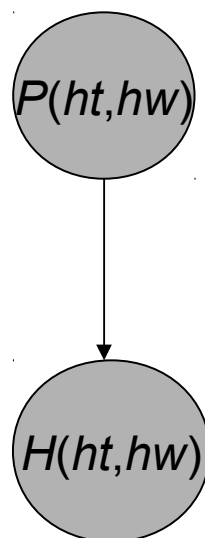


Collins Model (3b) 图示



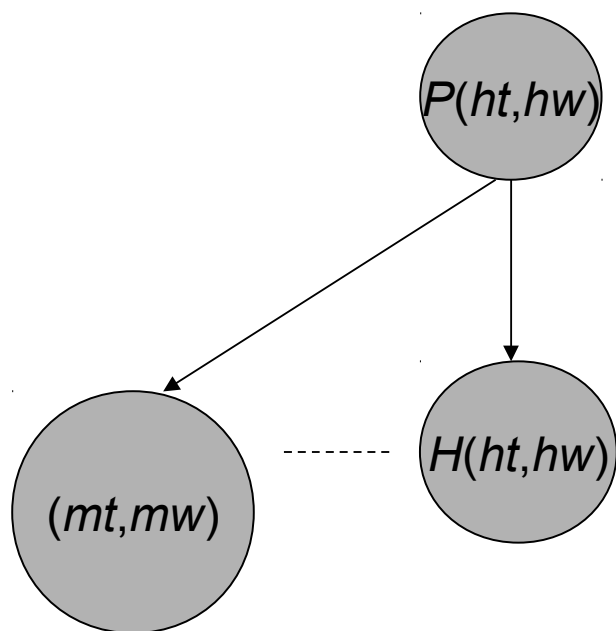
1. generate (ht,hw)

Collins Model (3c) 图示



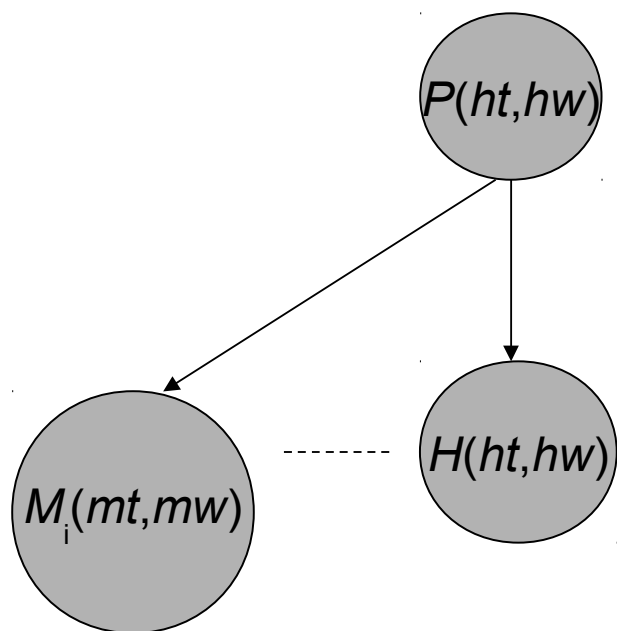
1. generate (ht, hw)
2. generate H

Collins Model (3d) 图示



1. generate (ht, hw)
2. generate H
3. expand (mt, mw)

Collins Model (3e) 图示



1. generate (ht, hw)
2. generate H
3. expand (mt, mw)
4. expand M_i

Collins Model (4) 模型的集成

- 根据当前短语标记生成中心词汇信息:

$$p(ht, hw|P)$$

- 根据结构信息和词汇信息生成中心短语标记:

$$p(H|P, ht, hw)$$

- 根据词汇依赖信息和结构信息生成修饰成分的词汇信息:

$$p(mt_{dist, dir}, mw_{dist, dir}|P, H, ht, hw, dist, dir)$$

- 根据词汇信息和结构依赖信息生成修饰成分的短语标记:

$$p(M_{dist, dir}|P, H, ht, hw, dist, dir, mt_{dist, dir}, mw_{dist, dir})$$

上面 ***dist*** 是修饰成分到中心成分的距离, ***dir*** 是修饰成分对于中心成分的方向 (左或右)

Collins Model (5) 模型的集成

$$\begin{aligned} & p(P(h) \rightarrow L_m(l_m) \dots L_1(l_1) H(h) R_1(r_1) \dots R_n(r_n)) \\ &= p(ht, hw | P) \times p(H | P, ht, hw) \\ & \times \prod_{dist, dir} p(mt_{dist, dir}, mw_{dist, dir} | P, H, ht, hw, dist, dir) \\ & \times p(M_{dist, dir} | P, H, ht, hw, dist, dir, mt_{dist, dir}, mw_{dist, dir}) \end{aligned}$$

其中, $h = (ht, hw)$

$$L_{dist}(l_{dist}) = M_{dist, left}(mt_{dist, left}, mw_{dist, left})$$

$$R_{dist}(r_{dist}) = M_{dist, right}(mt_{dist, right}, mw_{dist, right})$$

Collins Model (6) 参数估计

- Collins Model 数据稀疏问题极为严重
- 采用回退法进行参数平滑
 - 对概率的条件部分逐级回退
 - 最坏情况下回退到基本的 PCFG
- 可以看出， Collins Model 是典型的生成模型

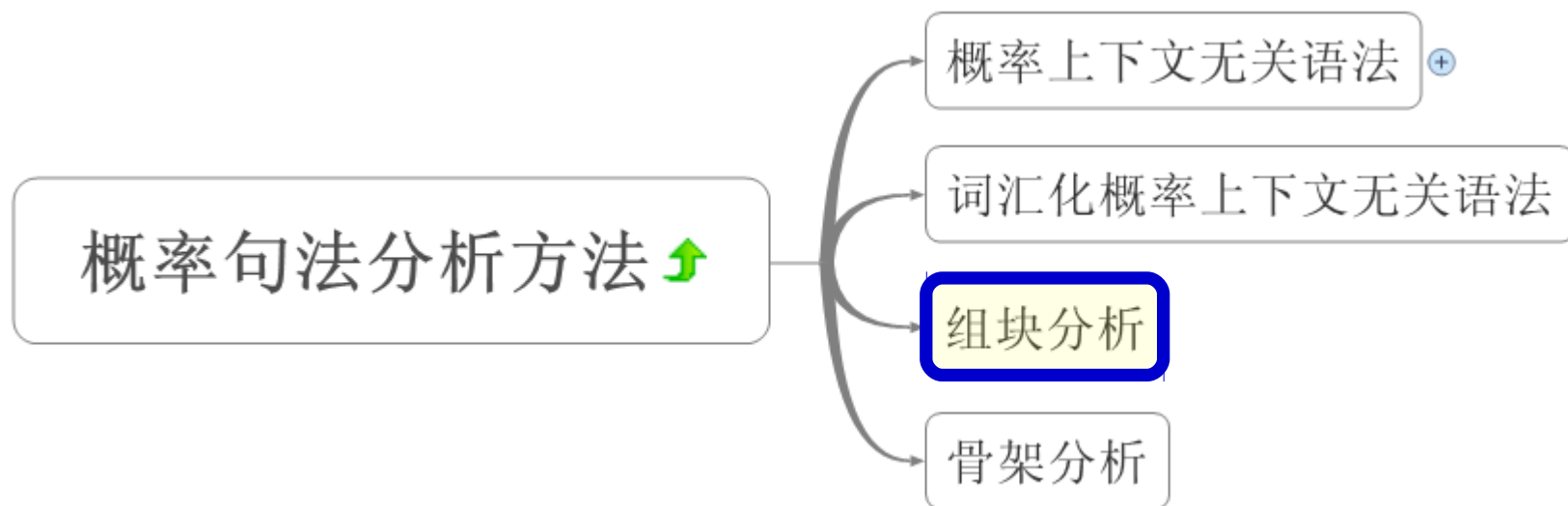
Collins Model (7) 中心成分确定

- 中心成分映射规则示例
 - 规则： IP right { IP VP }
 - 意义：对于句法树中标识为 IP 的节点，自右向左扫描该节点的所有孩子，第一个出现在列表 { IP VP } 中的孩子即为中心孩子节点。父节点的中心词和中心词词性等价于其中心孩子结点的中心词和中心词词性

概率句法分析的训练与评价

- 采用 **Treebank** 作为训练和测试语料库
- 目前研究界普遍采用 **Penn Treebank**（宾州树库）作为训练和测试语料库（汉语和英语），并且有公认的语料库划分标准
- 句法分析的评价通常采用标记正确率（**label precision**）和标记召回率（**label recall**）
- 在宾州树库上目前已发表论文中的英语句法分析的标记正确率和标记召回率是 **93%**，汉语句法分析的标记正确率和标记召回率大约是 **86%**

内容提要



组块分析

- 组块分析（ **Chunking** ），又称为部分分析（ **Partial Parsing** ）或浅层分析（ **Shallow Parsing** ）
- 基本思想：由于完全句法分析（ **Full Parsing** ）非常困难，于是人们考虑采用分而治之的策略：首先从句子中识别出组块（ **Chunk** ，也有人译为语块），然后再由组块结合成句子。组块分析就是指上述的第一个步骤。

英语组块的定义 (1)

- 组块：非递归的结构
 - [Abney] a chunk is the non-recursive core of an intra-clausal constituent, extending from the beginning of the constituent to its head, but not including post-head dependents.

英语组块的定义 (2)

- 非递归
- 不重叠
- 严格按照语法规则，而不考虑语义

[_{NX} former fire chief] [_{NX} Marvin Dirtwater]

apart from [_{NX} my good friend] and [_{NX} colleague]

in [_{NX} spite] of [_{NX} his objections]

组块的其他定义

- 最小名词短语 minNP
- 最大名词短语 maxNP
- 汉语的实语块
 - 孙宏林，现代汉语非受限文本的实语块分析，北京大学博士论文，2001
 - 实语块（**content chunk**）是由实词序列组成的短语，这里实词包括：名词、动词（助动词和系动词除外）、形容词、状态词、区别词、时间词、处所词、实义副词。
 - 我国 的 铁路建设 发展 得 很快。

组块分析的方法

- 有限状态机
- 转换为标记问题
 - 隐马尔科夫模型
 - 最大熵模型
 -

Abney 的重叠有限状态机 (1)

- Finite State Cascade: 重叠有限状态机

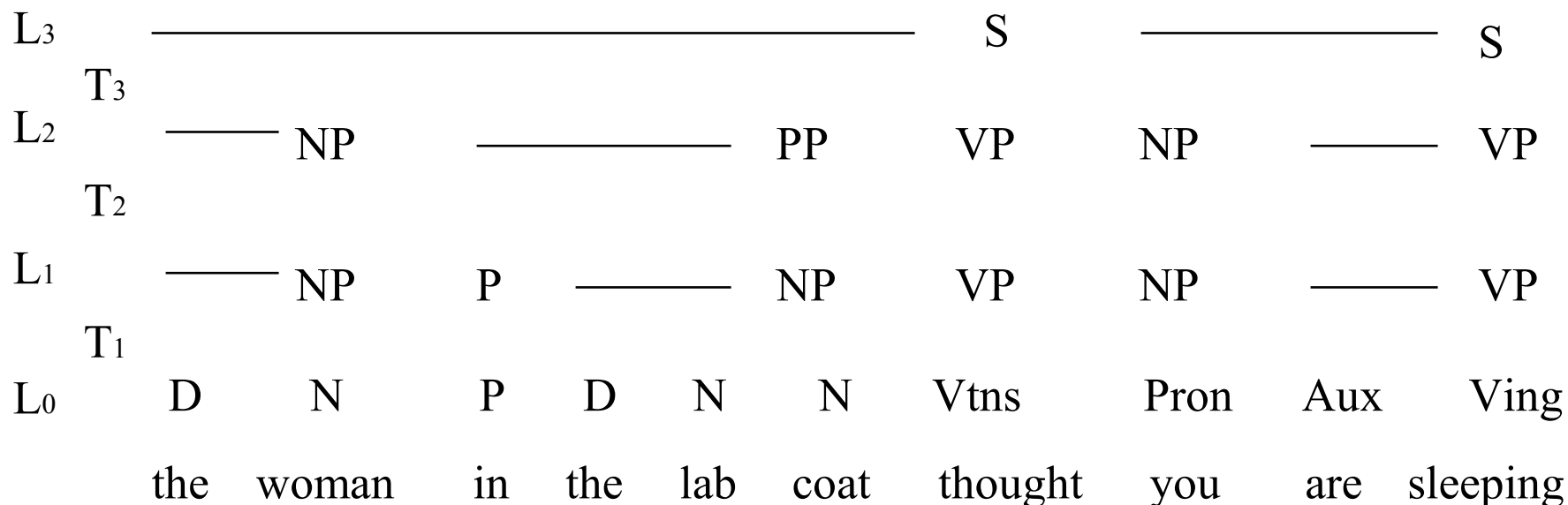
$$T_1 : \left\{ \begin{array}{l} \text{NP} \rightarrow (\text{D}) \text{ A} * \text{N}^+ \\ \text{VP} \rightarrow \text{Vtns} \mid \text{Aux} \text{Ving} \\ \text{NP} \rightarrow \text{Pron} \end{array} \right\}$$

$$T_2 : \{ \text{PP} \rightarrow \text{P} \text{ NP} \}$$

$$T_3 : \{ \text{S} \rightarrow \text{PP} * \text{NP} \text{ PP} * \text{VP} \text{ PP} * \}$$

Abney 的重叠有限状态机 (2)

例句分析:



基于隐马尔科夫模型的组块分析 (1)

已知:

$$W = w_1 w_2 \dots w_n$$

$$T = t_1 t_2 \dots t_n$$

求解:

$$C = c_1 c_2 \dots c_n$$

$$\hat{C} = \underset{C}{\operatorname{argmax}} P(C|W, T)$$

$$c_i \in 0, 1, 2, 3, 4$$

$$\text{where } 0 = (\quad), 1 = (\quad), 2 = (\quad \quad), 3 = (\text{I}), 4 = (0)$$

基于隐马尔科夫模型的组块分析 (2)

为词语之间的每一个间隔赋予一个标记：

[句法 /n 分析 /v][是 /v][自然 /n 语言 /n 处理 /v] 中 /f 的 /u [重点 /n] ./w									
< Φ ,n>	<v,v>	<n,v>	<v,n>	<n,n>	<n,v>	<v,f>	<f,u>	<u,n>	<n,w>
0	3	2	2	3	3	1	4	0	1

根据标记即可划分出组块。

CoNLL2002 的公共任务

- 自然语言理解会议 CoNLL2000 上面，以英语 Penn Treebank 为训练和测试语料，定义了一个组块分析的 Shared Task，结果如下：

	Precision	Recall	F β 1
[KM00]	93.45%	93.51%	93.48
[Hal00]	93.13%	93.51%	93.32
[TKS00]	94.04%	91.00%	92.50
[ZST00]	91.99%	92.25%	92.12
[Dej00]	91.87%	91.31%	92.09
[Koe00]	92.08%	91.86%	91.97
[Osb00]	91.65%	92.23%	91.94
[VB00]	91.05%	92.03%	91.54
[PMP00]	90.63%	89.65%	90.14
[Joh00]	86.24%	88.25%	87.23
[VD00]	88.82%	82.91%	85.76
Baseline	72.58%	82.14%	77.07%

内容提要

