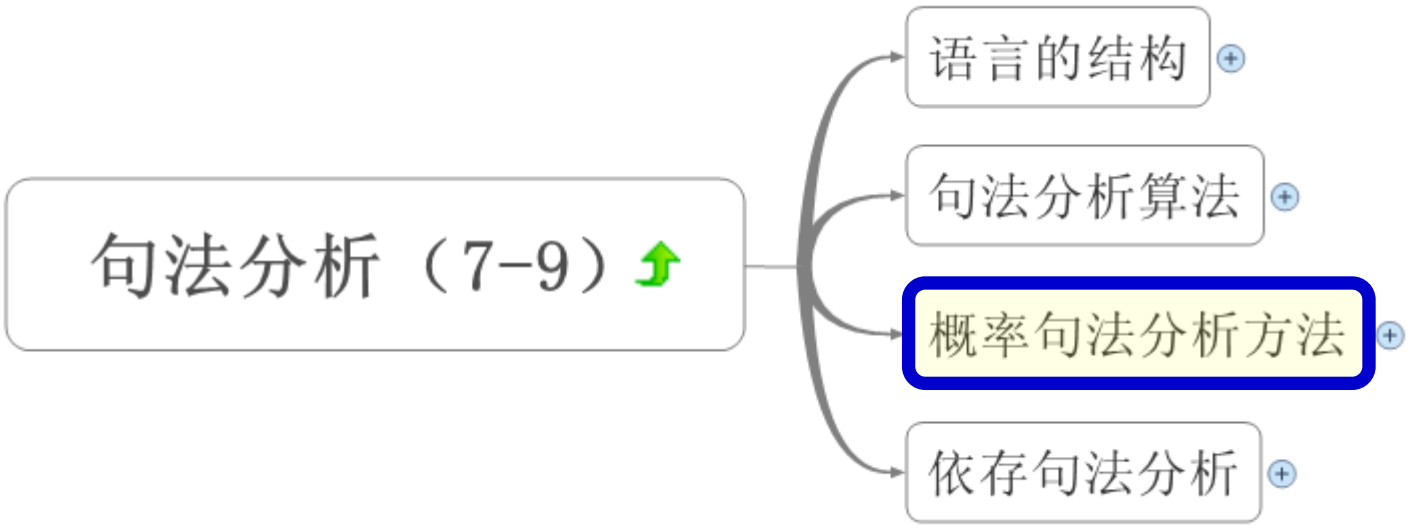
计算语言学

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2010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（Probabilities for determining the sentence），主要用于语音识别。
- 加速分析器（Probabilities for speedier parsing）
- 排歧(Probabilities for choosing between parses）

句法模型与语言模型

- Parsing Model

$$P(t \mid s, G), where \sum_{t} P(t \mid s, G) = 1$$

$$\hat{t} = \arg\max_{t} P(t \mid s, G)$$

- Language Model

$$\sum_{\{t: yield\ (t) \in L\}} P(t) = 1 \qquad P(s) = \sum_{t} P(s,t) = \sum_{\{t: yield\ (t) = s\}} P(t)$$

$$\hat{t} = \arg\max_{t} P(t \mid s) = \arg\max_{t} \frac{P(t,s)}{P(s)} = \arg\max_{t} P(t,s)$$

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

$$\sum_{a} P(A -> \alpha) = 1$$

- 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

$$P\left(\begin{array}{ccc} & {}^1S & \\ & /\backslash & \\ {}^2NP & & {}^3VP \\ /\backslash & & | \\ the\ man & & snores \end{array}\right)$$

$$= P({}^1S_{13} \rightarrow {}^2NP_{12}\ {}^3VP_{33},\ {}^2NP_{12} \rightarrow the_1 man_2,\ {}^3VP_{33} \rightarrow snores_3)$$

$$= P({}^1S_{13} \rightarrow {}^2NP_{12}\ {}^3VP_{33})P({}^2NP_{12} \rightarrow the_1 man_2 | {}^1S_{13} \rightarrow {}^2NP_{12}\ {}^3VP_{33})$$

$$P({}^3VP_{33} \rightarrow snores_3 | {}^1S_{13} \rightarrow {}^2NP_{12}\ {}^3VP_{33},\ {}^2NP_{12} \rightarrow the_1 man_2)$$

$$= P({}^1S_{13} \rightarrow {}^2NP_{12}\ {}^3VP_{33})P({}^2NP_{12} \rightarrow the_1 man_2)P({}^3VP_{33} \rightarrow snores_3)$$

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

概率上下文无关语法——示例

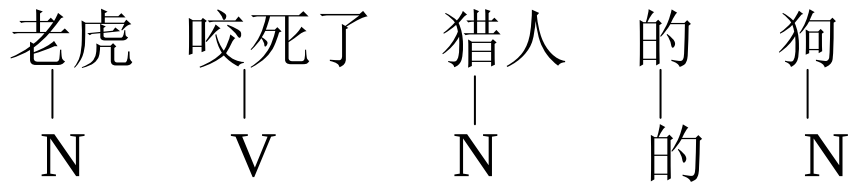
CFG

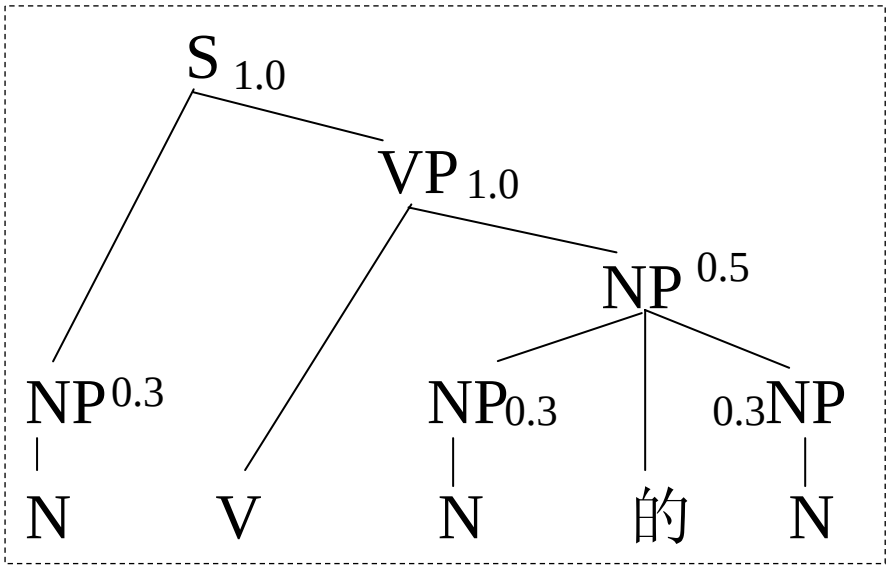
S→NP VP
VP → V NP
NP → N
NP → NP 的 NP
NP → VP 的 NP

PCFG

S→NP VP	1.0
VP → V NP	1.0
NP → N	0.3
NP → NP 的 NP	0.5
NP → VP 的 NP	0.2

分析树及其概率

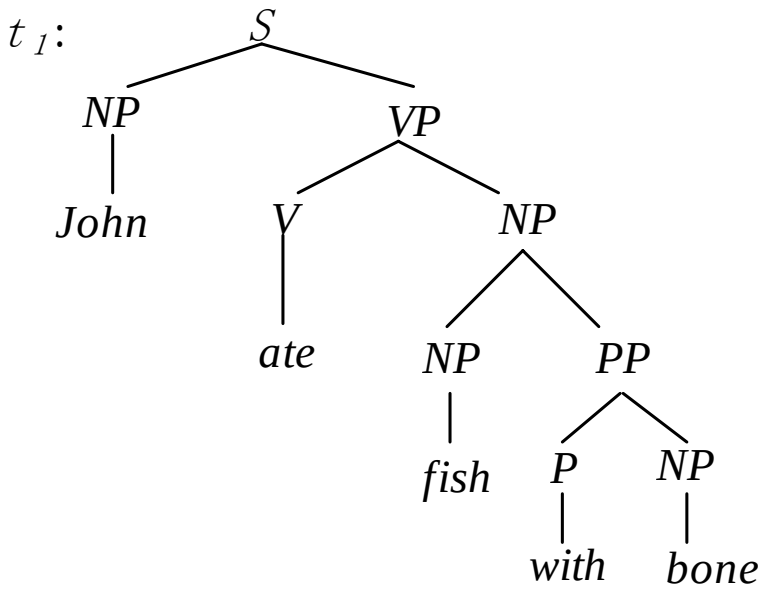


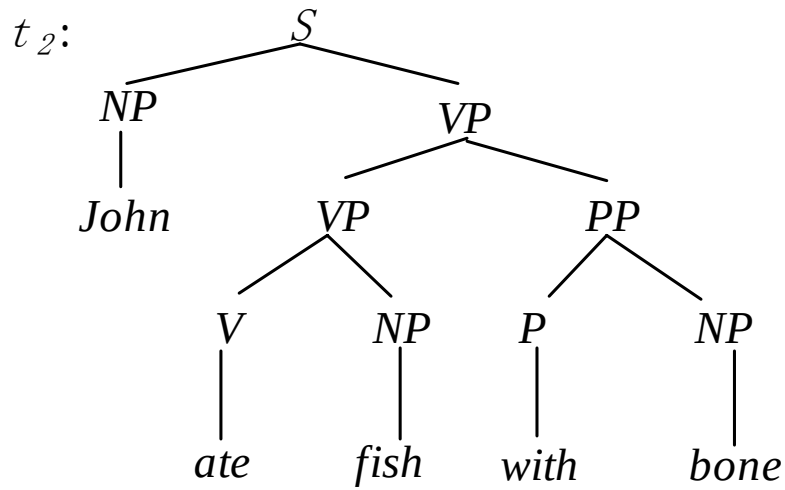


$$P(S)=1.0\times0.3\times1.0\times0.3\times0.5\times0.3$$
$$=0.0135$$

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$W=w_1w_2....w_n$的$P(W|G)$,也就是产生语句W的概率？
  - 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分析(parse)？
  - 韦特比算法
- 如何调节G的概率参数,使得$P(W|G)$最大?
  - 向内向外算法

向内变量

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

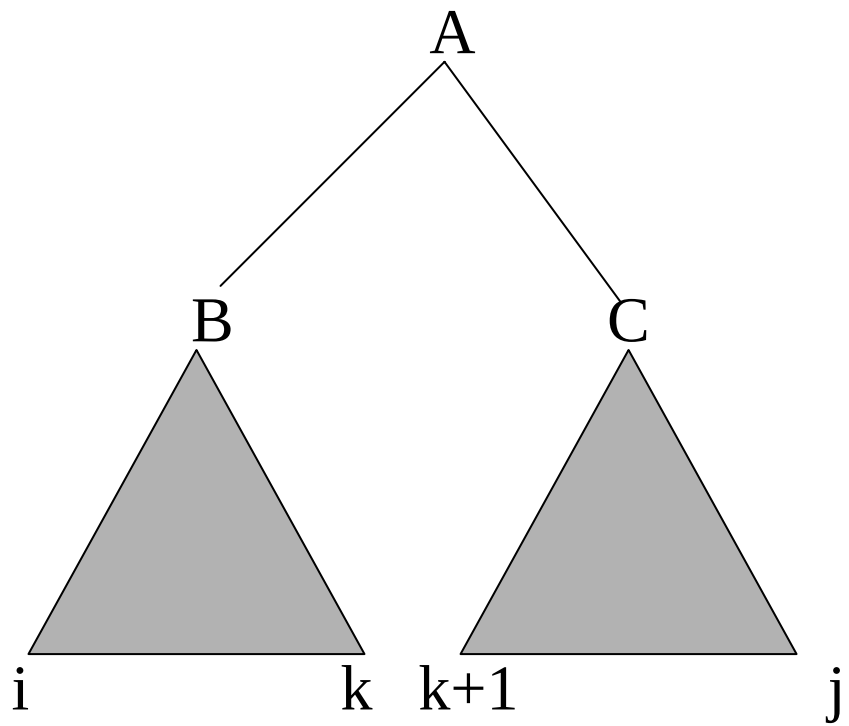
$$\alpha_{ii}(A) = P(A \to w_i)$$

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

  - 特别地，

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

向内算法示图



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$w_i w_{i+1} \ldots w_j$的最大概率。$\psi(A)$为最佳推导。
  - 动态规划公式

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

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

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

  - 特别地，

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

向外变量

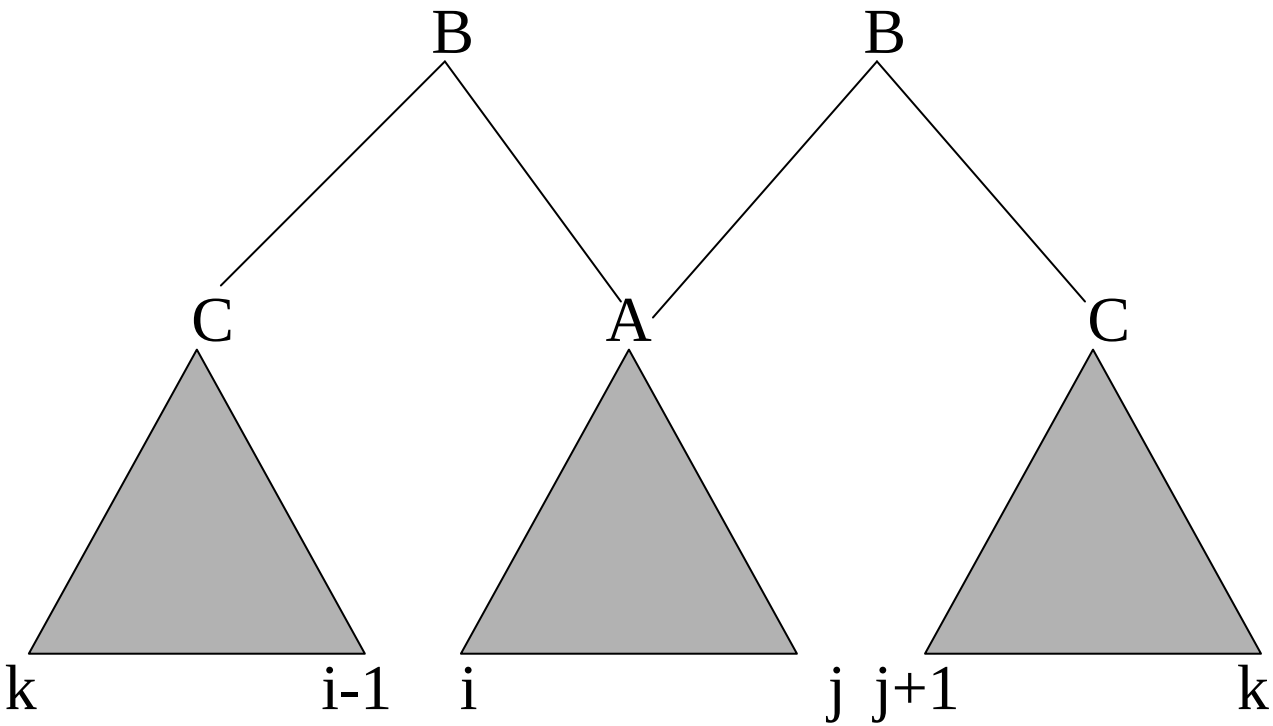
- 向外变量$\beta_{ij}(A)=P(S\Rightarrow w_1\ldots w_{i-1}A\,w_{j+1}\ldots w_n)$
  - 动态规划递归公式

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

$$\beta_{ij}(A)=\sum_{B,C}\sum_{k>j}P(B\rightarrow AC)\alpha_{j+1,k}(C)\beta_{ik}(B)$$

$$+\sum_{B,C}\sum_{k<j}P(B\rightarrow CA)\alpha_{k,i-1}(C)\beta_{kj}(B)$$

向外算法示图



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 $\Sigma_{\mu} P(A \rightarrow \mu) = 1$.由此得到语法$G_0$. i<-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

$$C(A \to BC)$$

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

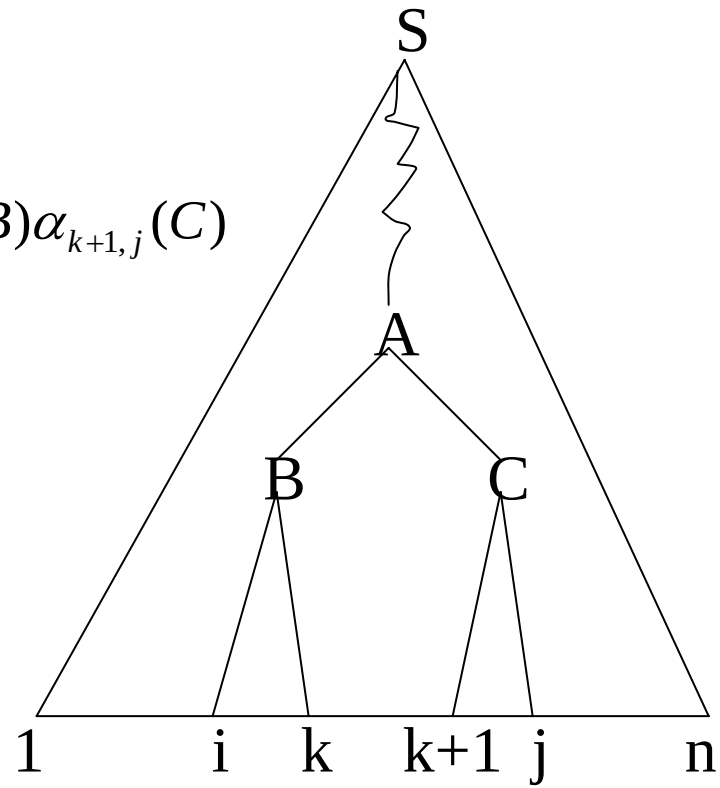
$$= \frac{1}{P(w_1 \ldots w_n \mid G)} \sum_{1 \le i \le k \le j \le n} \beta_{ij}(A) P(A \to BC) \alpha_{ik}(B) \alpha_{k+1,j}(C)$$

$$C(A \to a)$$

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

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

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



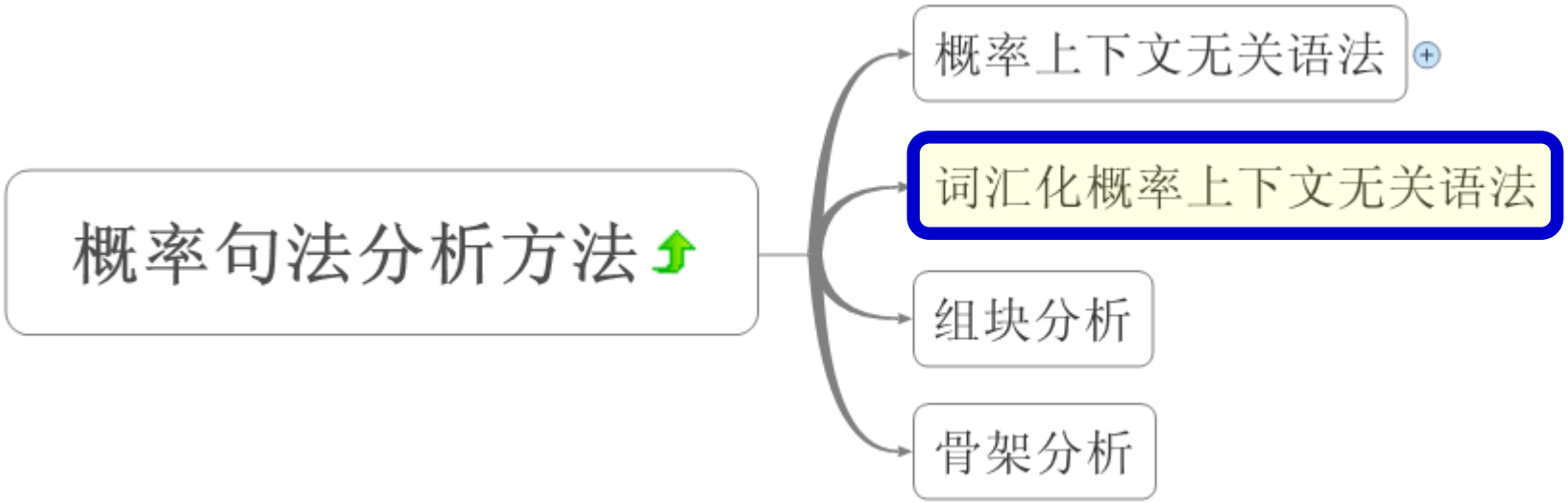
PCFG的优点

- 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)

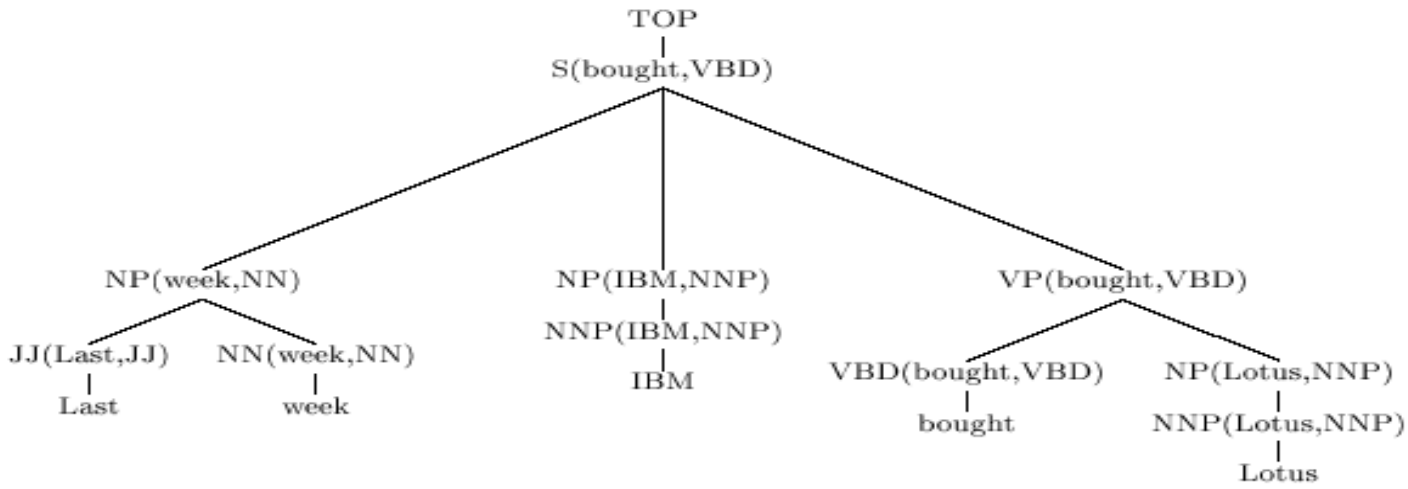
内容提要



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,NN)	→	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

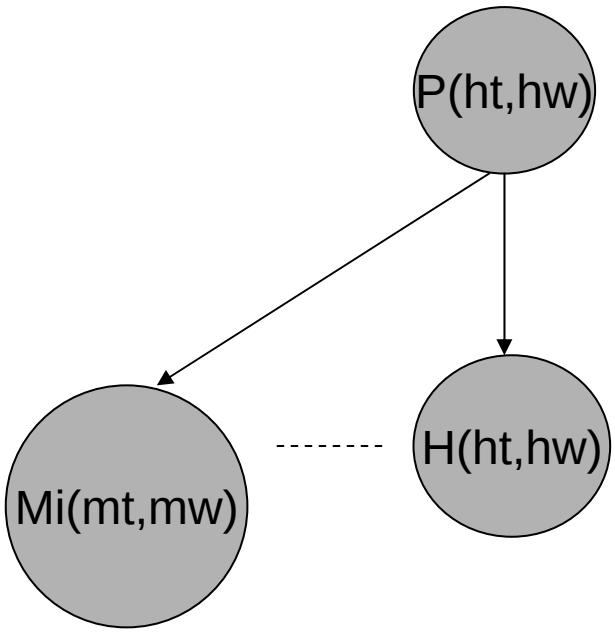
- 两个组成部分：词汇中心和结构中心
- 词汇中心：中心词(hw)和中心词词性标记(ht)
- 结构中心：中心成分的短语标记(hn)
- 中心成分的生成
  - 首先生成词汇中心
  - 其次生成结构中心

Collins Model (3) 修饰成分的扩展

- 基本规则形式
  $P(h) \rightarrow L_m(l_m)\ldots L_1(l_1)H(h)R_1(r_1)\ldots R_n(r_n)$
- 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 (3) 图示



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

Collins Model (4) 模型的集成

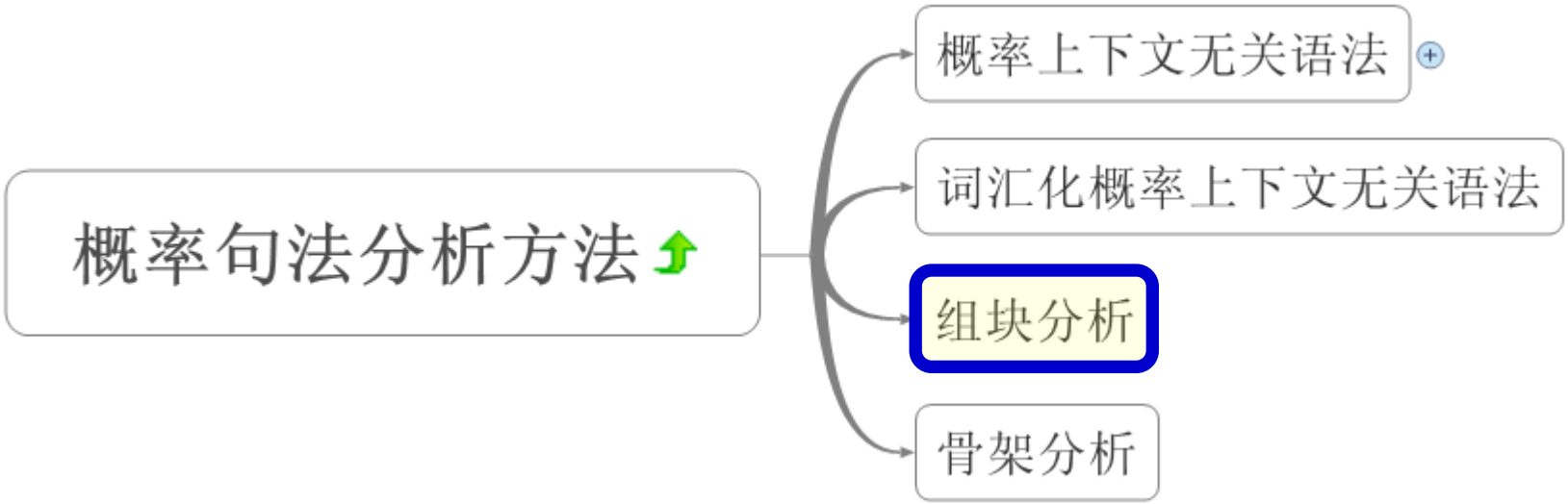
- 根据当前短语标记生成中心词汇信息：
  P(ht,hw|P)
- 根据结构信息和词汇信息生成中心短语标记：
  P(H|P,ht,hw).
- 根据词汇依赖信息和结构信息生成修饰成分的词汇信息：
  P(mt,mw|P,H,ht,hw,dist,dir)
- 根据词汇信息和结构依赖信息生成修饰成分的短语标记：
  P(Mi|mt,mw,ht,hw,P,H,dis,dir)

  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）

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）
- 目前论文中已报道的英语句法分析的标记正确率和标记召回率是90%，汉语句法分析的标记正确率和标记召回率大约是80%（不到）

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
  - 因马尔科夫模型
  - 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

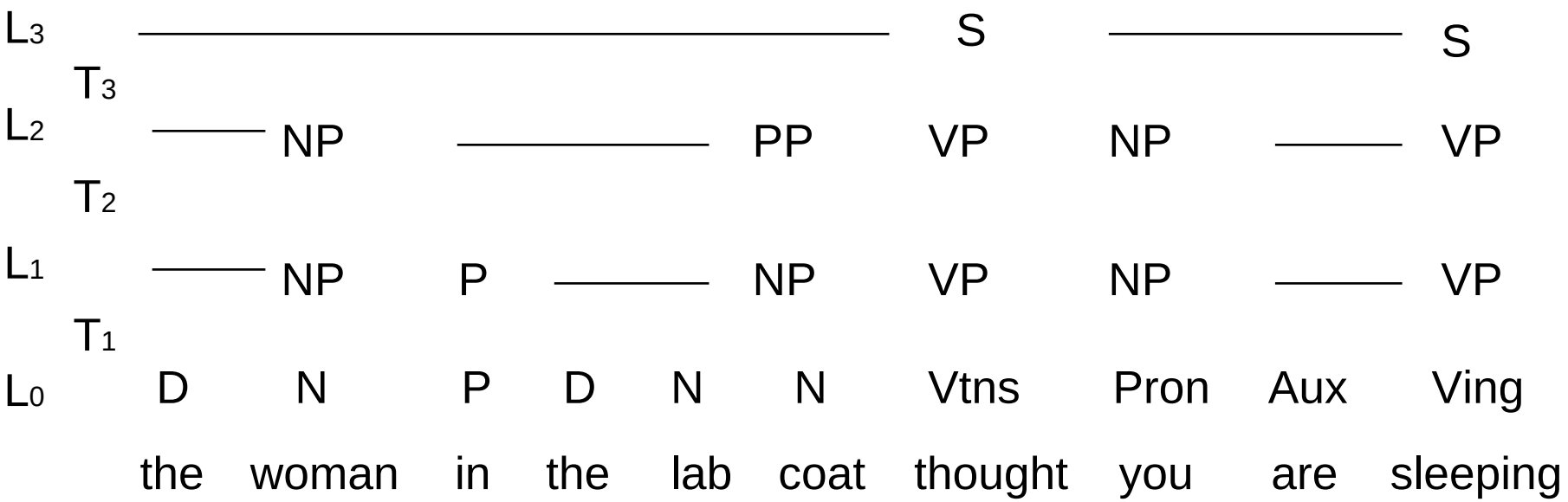
$$T_1 : \left\{ \begin{array}{l} \text{NP} \rightarrow \text{(D)} \quad \text{A} * \quad \text{N}^+ \\ \text{VP} \rightarrow \text{Vtns} \quad | \quad \text{Aux} \quad \text{Ving} \\ \text{NP} \rightarrow \text{Pron} \end{array} \right\}$$

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

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

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：

$$W = w_1 w_2 \ldots w_n$$

$$T = t_1 t_2 \ldots t_n$$

求解：

$$C = c_1 c_2 \ldots c_n$$

$$C' = \arg \max_{c} P(C \mid W, T)$$

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

0（**[**）、**1**（**]**）、**2**（**][**）、**3**（**I**）、**4**（**O**）

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

<Φn>	<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	Fβ1
[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