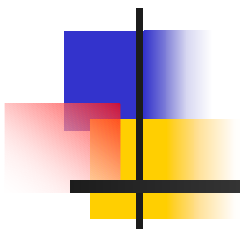


第八章 句法分析（三）

詹卫东

http://ccl.pku.edu.cn/doubtfire/

提纲

- 1 概率上下文无关文法（probabilistic context free grammar）
  - 1.1 向内算法
  - 1.2 Viterbi算法
  - 1.3 向内向外算法
- 2 部分分析方法（partial parsing / shallow parsing / chunking）
  - 2.1 基于HMM的部分分析技术
  - 2.2 基于有限状态自动机的部分分析技术
  - 2.3 基于转换的错误驱动的部分分析技术

1 概率上下文无关文法

PCFG: 为CFG中的每条规则增加一个概率值

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

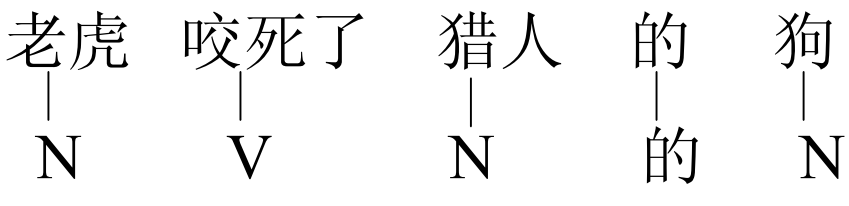
CFG

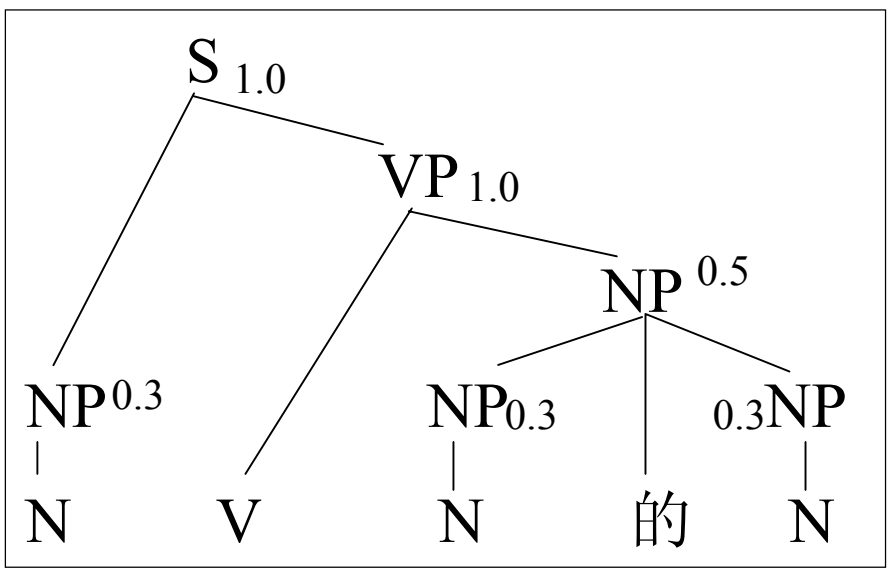
S→NP VP
VP → V NP
NP → N
NP → NP 的 NP
NP → VP 的 NP

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



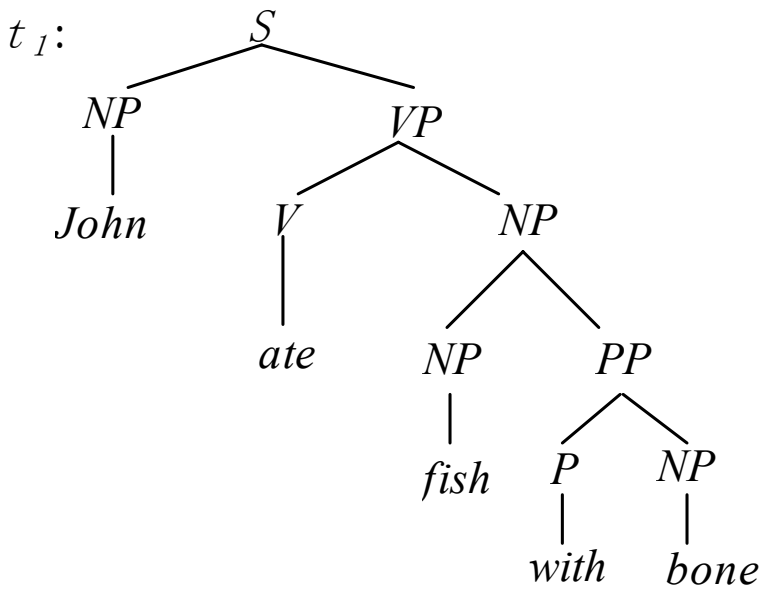


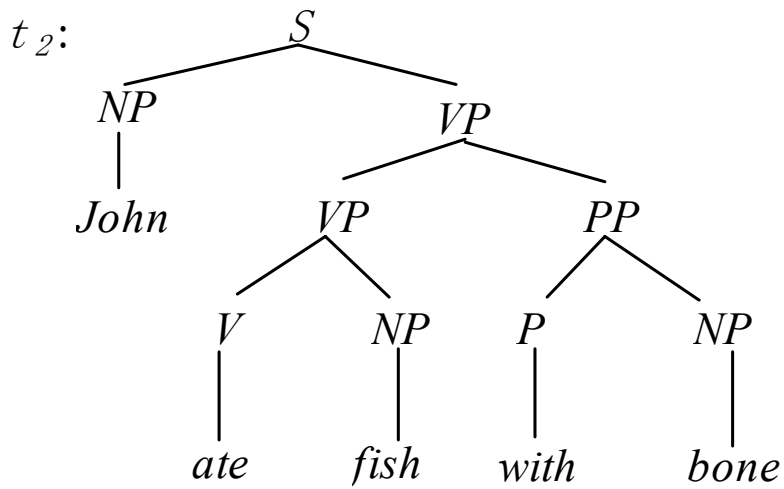
$$P(S)=1.0\times0.3\times1.0\times0.3\times0.5\times0.3$$

$$=0.0135$$

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的三个问题

给定一个符号串W=$w_1 w_2 \ldots w_n$和一部概率上下文无关文法G,

(1) 如何快速计算由G产生符号串W的概率 P(W|G)?

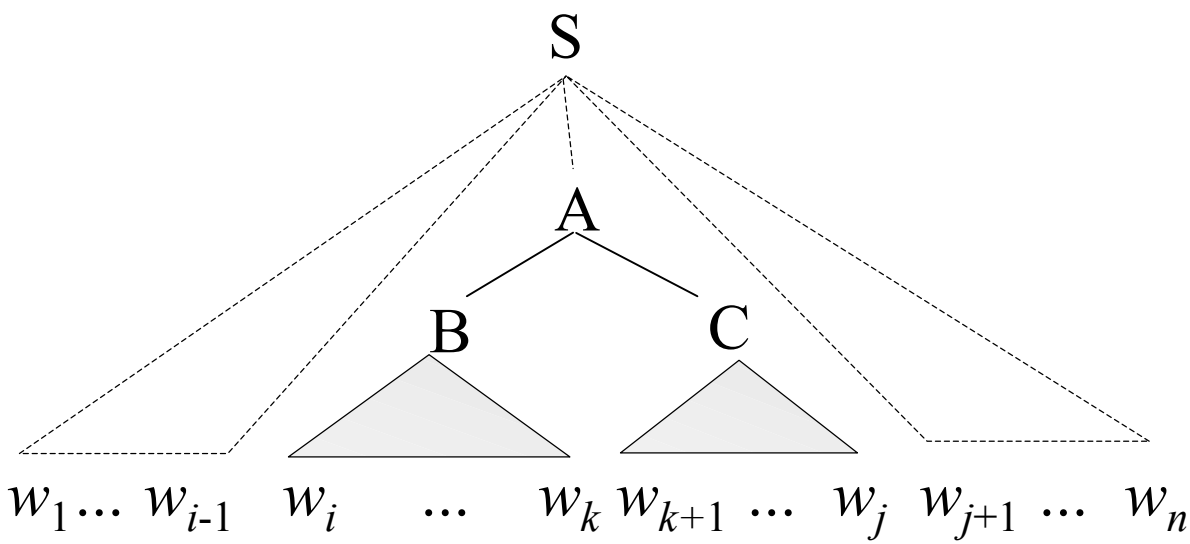
(2) 如果W有多种可能的语法树，如何选择一棵最好的树？

$$\arg\max_{tree} P(tree \mid W, G)$$

(3) 如何调整G的规则概率参数，使得P(W|G)最大？

$$\arg\max_{G} P(W \mid G)$$

1.1 向内算法



非终结符A的内部概率（inside probability）定义为：

根据文法G从A推出词串$w_i \ldots w_j$的概率，记做 $\alpha_{i,j}(A)$ $i \le j$

向内变量

内部概率公式

$$\alpha_{i,j}(A) = P(w_i \ldots w_j \mid A) \quad i < j$$

$$= \sum_{B,C,k} P(w_i \ldots w_k, B, w_{k+1} \ldots w_j, C \mid A)$$

$$= \sum_{B,C,k} P(B,C \mid A) P(w_i \ldots w_k \mid A,B,C) P(w_{k+1} \ldots w_j \mid w_i \ldots w_k, A, B, C)$$

独立性假设

$$= \sum_{B,C,k} P(B,C \mid A) P(w_i \ldots w_k \mid B) P(w_{k+1} \ldots w_j \mid C)$$

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

$$\alpha_{i,j}(A) = P(A \to w_i) \quad i = j$$

向内算法过程描述（自底向上）

输入：G=(S,V_N,V_T, P), 字符串W=$w_1w_2...w_n$

输出：P($w_1...w_n$|G)= $\alpha_{1,n}$(S)

1) 初始化： $\alpha_{i,i}$(A) =P(A→w_i), A∈ V_N, 1 ≤i ≤n

2) 归纳计算：j从1到n，i从1到n-j，重复下面的计算

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

3) 结束： P(S→$w_1...w_n$|G)= $\alpha_{1,n}$(S)

向内算法计算过程示例

5					α(NP)=0.18
4				α(P)=1.0	α(PP)=0.18
3			α(NP)=0.18		α(NP)=0.01296
2		α(V)=1.0	α(VP)=0.126		α(VP)=0.015876
1	α(NP)=0.1		α(S)=0.0126		α(S)=0.0015876
	1	2	3	4	5
	John	ate	fish	with	bone

1.2 Viterbi算法

- Viterbi变量$\delta_{i,j}(A)$：根据文法G，从非终结符A推导出词串$w_i \ldots w_j$，每个推导都有对应的概率值，其中概率最大的记做$\delta_{i,j}(A)$。

- 比较：

向内变量$\alpha_{i,j}(A)$是从A产生出词串$w_i \ldots w_j$的所有推导的概率之和。

Viterbi算法过程描述

输入：G=(S,V_N,V_T, P), 字符串W=$w_1w_2...w_n$

输出：t* （W在G下最可能的分析树）

1) 初始化： $\delta_{i,i}(A) = P(A \rightarrow w_i)$ $A \in V_N$, $1 \le i \le n$

2) 归纳计算：j从1到n，i从1到n-j，重复下面的计算

$$\delta_{i,i+j}(A) = \max_{B,C \in V_N; i \le k \le i+j} P(A \rightarrow BC)\delta_{i,k}(B)\delta_{k+1,i+j}(C)$$

$$\Delta_{i,i+j}(A) = \arg\max_{B,C \in V_N; i \le k \le i+j} P(A \rightarrow BC)\delta_{i,k}(B)\delta_{k+1,i+j}(C)$$

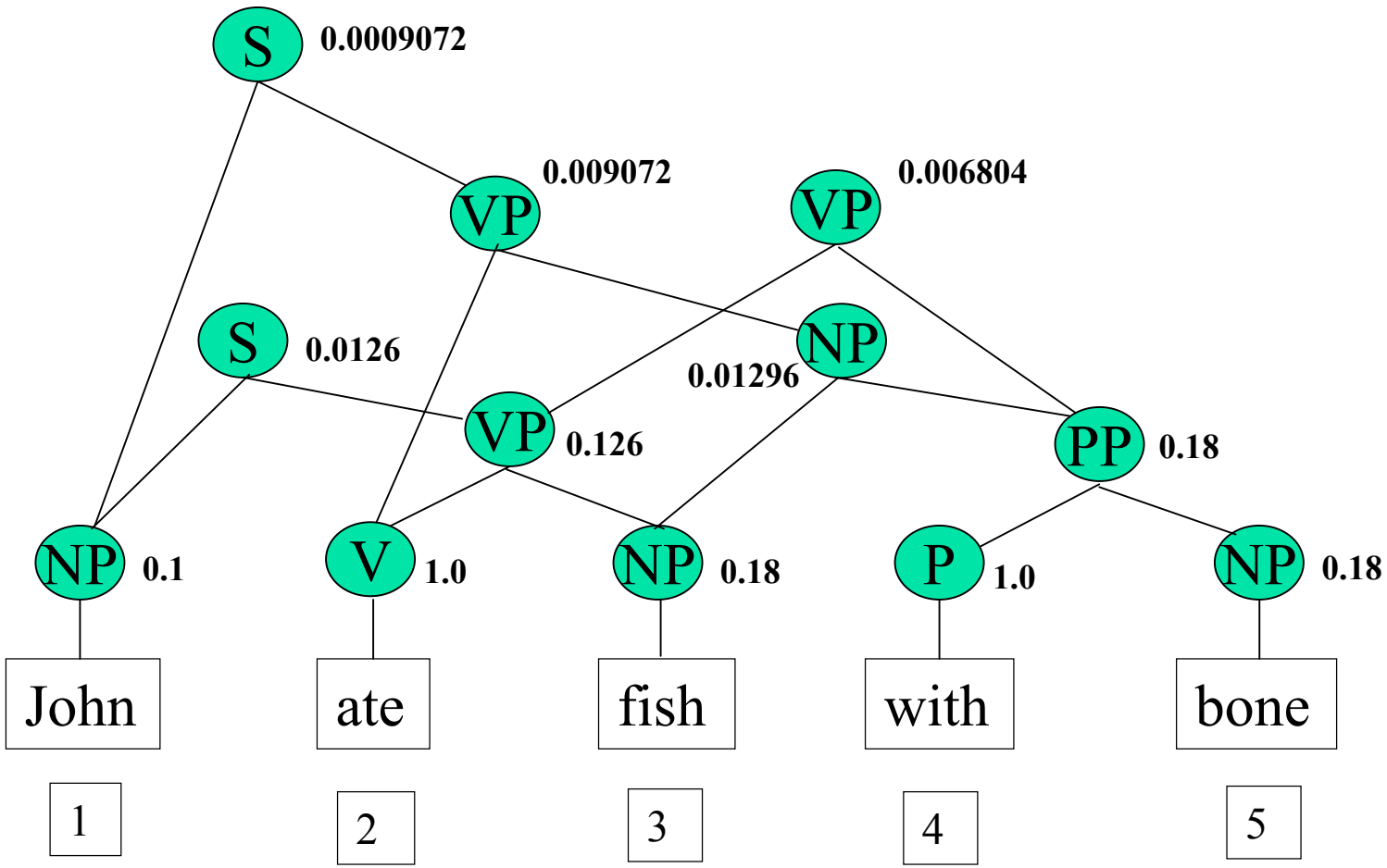
Δ 变量用于记录分析过程的历史

3) 结束：$P(t^*) = \delta_{1,n}(S)$

t*的根节点为S（文法开始符号）；

从$\Delta_{1,n}(S)$开始回溯，得到S的最优树结构

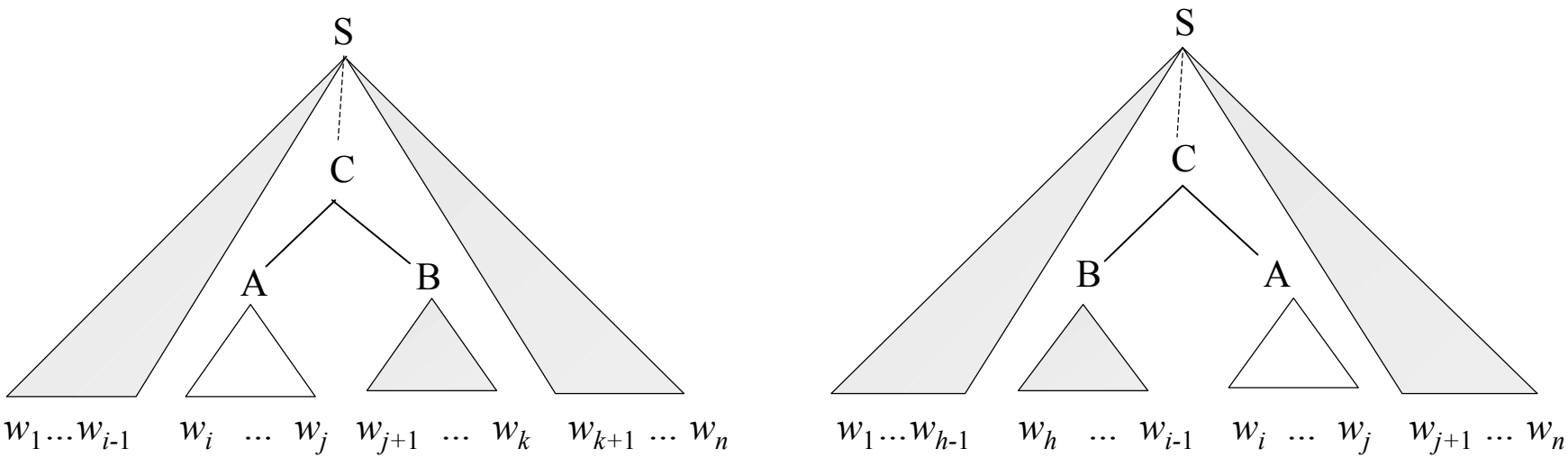
Viterbi算法过程示例



Viterbi算法过程示例（续）

节点号	节点	起点	终点	δ	Δ
1	NP	1	1	0.1	
2	V	2	2	1.0	
3	NP	3	3	0.18	
4	VP	2	3	0.126	(2,3)
5	S	1	3	0.0126	(1,4)
6	P	4	4	1.0	
7	NP	5	5	0.18	
8	PP	4	5	0.18	(6,7)
9	NP	3	5	0.01296	(3,8)
10	VP	2	5	0.006804	(4,8)
11	VP	2	5	0.009072	(2,9)
12	S	1	5	0.0009072	(1,11)

1.3 向内向外算法



非终结符A的外部概率（outside probability）定义为：

根据文法G从A推出词串$w_i \ldots w_j$的上下文的概率，记做 $\beta_{i,j}(A)$ $i \le j$

向外变量

外部概率公式

$$\beta_{1,n}(A) = \begin{cases} 1, & A = S \\ 0, & A \neq S \end{cases}$$

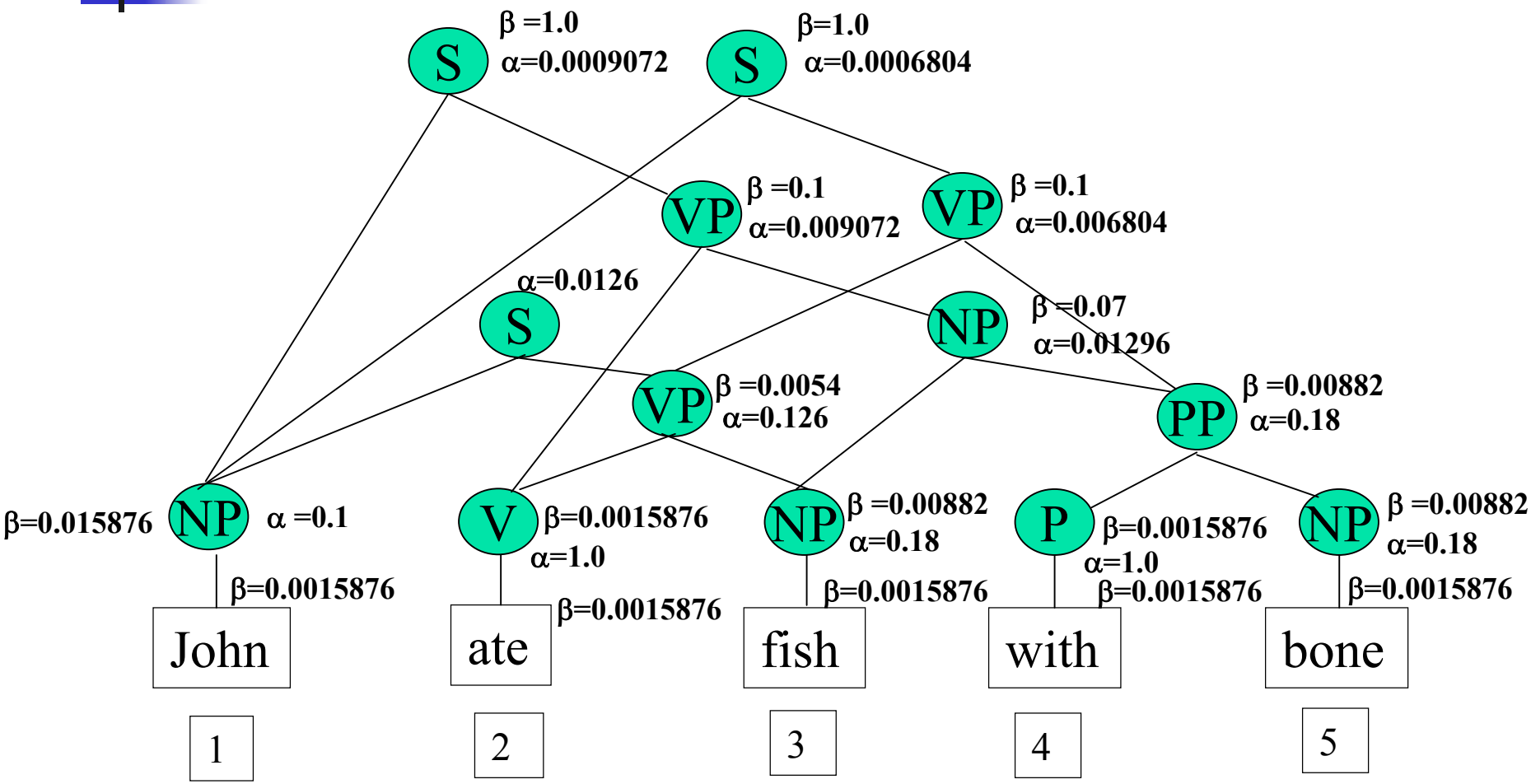
$$\beta_{i,j}(A) = P(w_1 \ldots w_{i-1}, A, w_{j+1} \ldots w_n \mid G)$$

$$= \sum_{B,C,j<k} P(w_1 \ldots w_{i-1}, C, w_{k+1} \ldots w_n) P(C \to AB) P(B \to w_{j+1} \ldots w_k)$$

$$+ \sum_{B,C,h<i} P(w_1 \ldots w_{h-1}, C, w_{j+1} \ldots w_n) P(C \to BA) P(B \to w_h \ldots w_{i-1})$$

$$= \sum_{B,C,j<k} \beta_{i,k}(C) P(C \to AB) \alpha_{j+1,k}(B) + \sum_{B,C,h<i} \beta_{h,j}(C) P(C \to BA) \alpha_{h,i-1}(B)$$

计算外部概率示例（自顶向下）



规则的概率

文法中每条规则的概率，可以用下面公式估计：

$$P(A \to \xi) = \frac{Number\ (A \to \xi)}{\sum_{\gamma} Number\ (A \to \gamma)}$$

S→NP VP	S→NP VP　　P(S→NP VP)
VP → V NP	VP → V NP　　P(VP→V NP)
NP → N	NP → N　　P(NP→N)
NP → NP 的 NP	NP → NP 的 NP　　P(NP→NP 的 NP)
NP → VP 的 NP	NP → VP 的 NP　　P(NP→VP 的 NP)

$$P(NP \to N) = \frac{Number(NP \to N)}{Number(NP \to N) + Number(NP \to NP\text{的}\ NP) + Number(NP \to VP\text{的}\ NP)}$$

规则使用次数的数学期望

规则“A→B C”使用次数的数学期望是给定语句条件下，A, B, C 这三个非终结符同时出现的概率

$$\overline{Number}(A \to B\ C) = \sum_{1 \le i \le k \le j \le n} P(A_{i,j}, B_{i,k}, C_{k+1,j} \mid w_1 \ldots w_n, G)$$

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

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

规则的概率（续）

$$P(A \to B\ C) = \frac{\overline{Number}(A \to B\ C)}{\sum_{\gamma} \overline{Number}(A \to \gamma)}$$

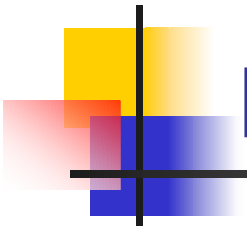
$$= \frac{\sum_{1 \le i \le k < j \le n} \beta_{i,j}(A)\ \ P(A \to BC)\, \alpha_{i,k}(B)\ \ \alpha_{k+1,j}(C)}{\sum_{1 \le i \le j \le n} \beta_{i,j}(A)\alpha_{i,j}(A)}$$

公式②

向内向外算法过程描述

- 初始化：随机地给P(A→μ) 赋值，使得Σ_{μ}P(A→ μ) =1，
  得到语法G_i，$i=0$
- 循环执行下面步骤，直至P(A→μ)收敛：
  - 1) 根据公式①计算每条规则使用次数的期望值
  - 2) 根据上一步所得期望值,即根据公式②，重新估计
    P(A→μ) , 得到语法G_{i+1}

向内向外算法示例，可参见陈小荷2000，第12章



PCFG的优缺点

- 优点
  - 可以对句法分析的歧义结果进行概率排序
  - 提高文法的容错能力（robustness）
- 缺点
  - 没有考虑词对结构分析的影响
  - 没有考虑上下文对结构分析的影响

2 部分分析方法

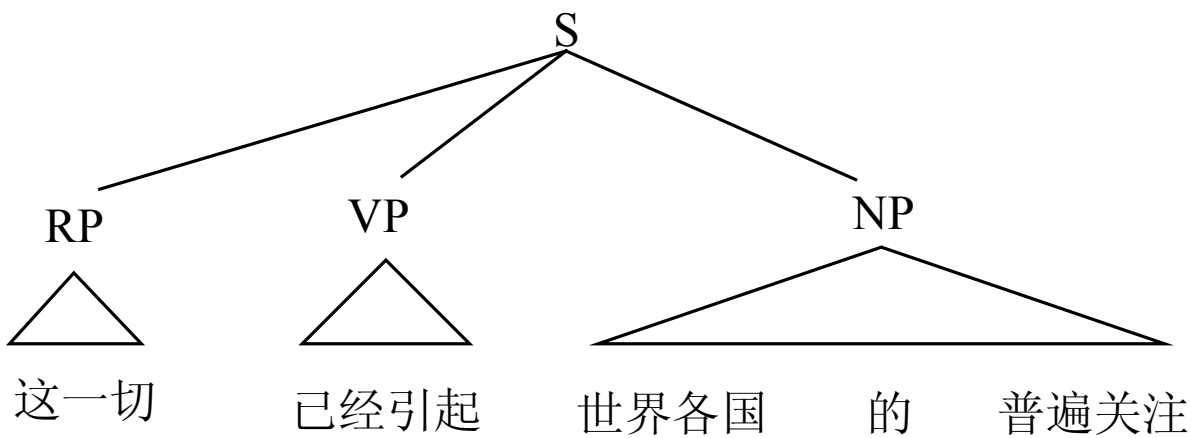
- 从完全句法分析（complete parsing）到部分句法分析（partial parsing）

  - 真实语料的复杂性
  - 语言知识的不足
  - 提高分析的效率
  - 应用目标驱动　e.g. 命名实体识别（Named Entity Recognition）

部分分析的另外两个名称：
- shallow parsing / 浅层分析
- chunking / 组块分析

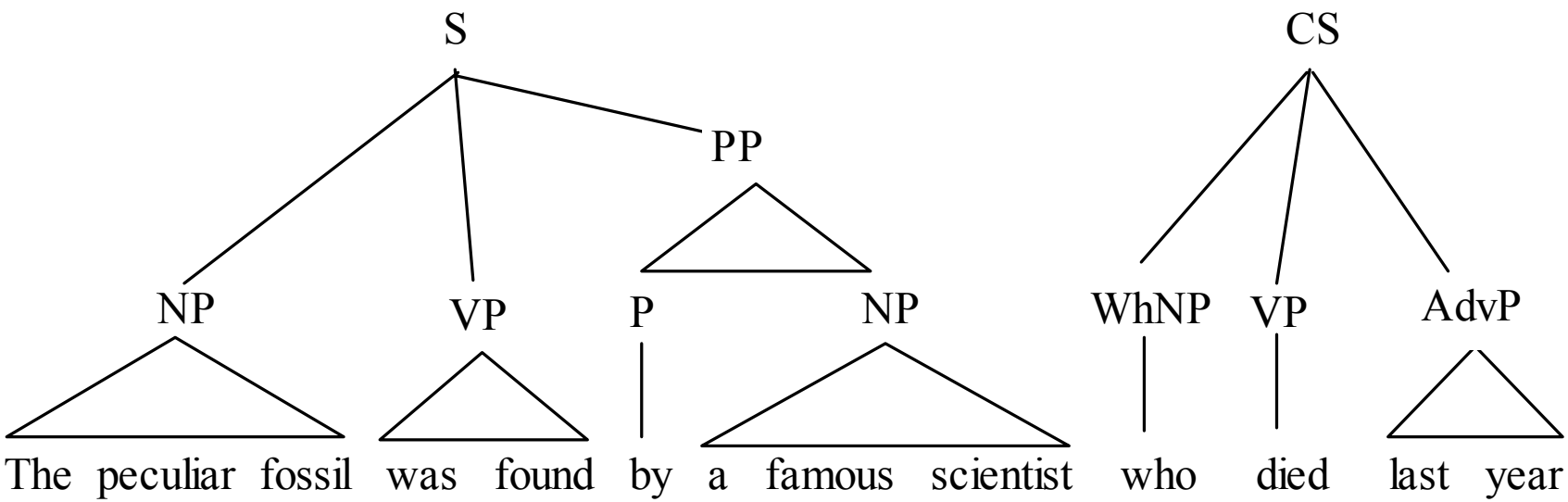
部分分析示例

这一切已经引起世界各国的普遍关注



部分分析示例（续）

The peculiar fossil was found by a famous scientist who died last year



2.1 基于HMM的部分分析技术

识别目标：非递归NP

组块分析：在词性序列中插入括号，来标示组块边界

[The/ DT prosecutor/ NN] said / VB in/ IN [closing / NN] that / CS …

短语边界

一对词性标记<α, β>之间可能插入的标记：

（1） [　　表示一个NP组块的开始

（2）]　　表示一个NP组块的结束

（3）][　　表示两个NP组块相邻

（4） I　　表示不是NP组块边界，且处在NP内部

（5） O　　表示不是NP组块边界，且处在NP外部

基于HMM的NP组块边界标注

（1）带有词性标记、组块边界标记的语料库Corpus

（2）可观察符号序列：词性标记对序列<α, β>

（3）隐状态：5个可能的NP组块边界标记（chunk_tag）

（4）通过对Corpus的统计，得到：

（I）状态转移矩阵；

（II）每个状态输出不同词性标记对的概率；

$	The	prosecutor	said	in	closing	that	…
<$, DT>	<DT, NN>	<NN, VB>	<VB, IN>	<IN, NN>	<NN, CS>		
[	I	]	O	[	]		

2.2 基于有限状态自动机的部分分析

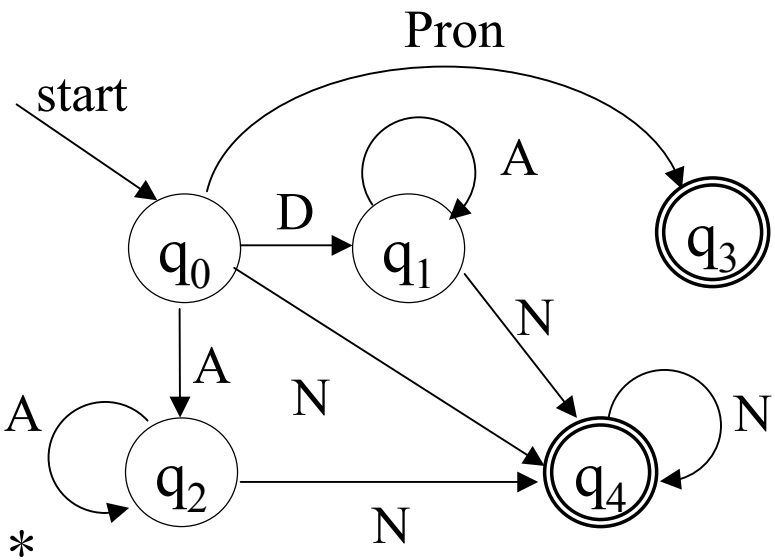
分层的有限状态自动机（finite state cascades）

Level 1: NP→D? A* N+ | Pron
VP→Vz | …

Level 2: PP→P NP

Level 3: SV→NP VP

Level 4: S→(Adv|PP)? SV NP? (Adv|PP)*



合法的NP：D N；D A N；Pron；A N；N N

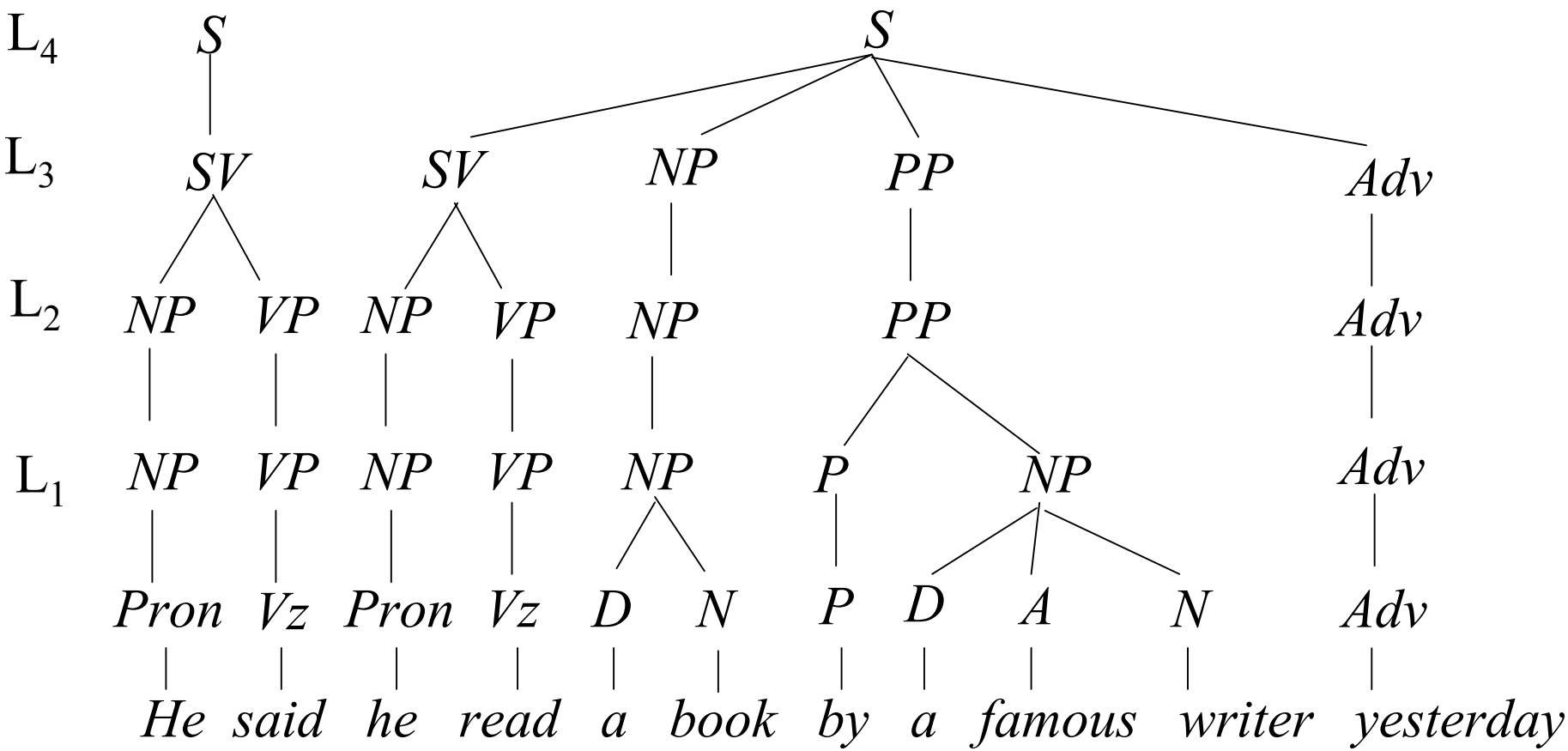
非法的NP：A D N；D D N；N A N

? 不出现或出现1次
* 出现0次，1次或多次
+ 出现1次或多次
| 逻辑“或”

基于FSA的部分分析过程描述

1) 从左向右扫描输入字符串，按照 *Li* 层级上的正则表达式模式进行归约，得到新的模式序列，对于输入串中无法归约的符号，直接输出；
2) *i*=*i*+1，在新的 *Li* 层级上，用正则表达式模式进行归约；
3) 不断进行上述步骤，直至无法归约为止；
4) 如果归约过程中有多种选择，以覆盖范围最大的归约子串为输出结果。

基于FSA的部分分析过程示例



基于FSA的部分分析结果

[S
[SV [NP He] [VP said]]
]
[S
[SV [NP he] [VP read]]
[NP a book]
[PP [P by] [NP a famous writer]]
[Adv yesterday]
]

3.3 基于转换的错误驱动的部分分析

Eric Brill (1995)

Ramshaw & Marcus (1995)

将识别NP组块边界的问题等同于词性标注问题

利用经人工标注的语料库学习转换规则

基于转换的错误驱动的部分分析示例

1) O 表示它所对应单词处在名词组块的外部。

2) L 表示它所对应的单词是名词组块的左边界。

3) I 表示它所对应单词处在名词组块的内部。

4) R表示它所对应的单词是名词组块的右边界。

5) S 表示它所对应的单词单独构成一个名词组块。

He	*puts*	*his*	*dirty*	*hand*	*in*	*the*	*bag*	.
PRP	***VBZ***	***PRP$***	***JJ***	***NN***	***IN***	***ART***	***NN***	**.**
S	O	L	I	R	O	L	R	O

转换规则示例

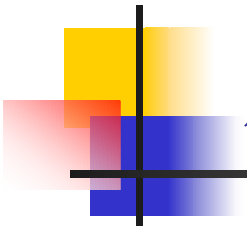
触发条件：

(POS[0]=v) & (POS[-1]=q) & (POS[1]=n) & (CT[0]=O)

转换动作：

CT[0] = O → CT[0] = L

一/m	辆/q	出租/v	汽车/n	起火/v	了/u
O	O	O	S	O	O
		↓			
O	O	L	S	O	O



小结

对分析算法进行评价的几个指标

（1）效率

（2）容错能力

（3）维护性

（4）应用目标

进一步阅读文献

- 陈小荷，2000，《现代汉语自动分析》，第12章“概率语法”，北京语言文化大学出版社。
- 翁富良、王野翊（1998）《计算语言学导论》，第8章，中国社会科学出版社
- 赵铁军 等，2000，《机器翻译原理》，哈尔滨工业大学出版社，第5.2，5.3节
- Christopher D. Manning & Hinrich Schutze, 1999, Foundations of Statistical Natural Language Processing, The MIT Press. Chapter 11, 12
- 孙宏林、俞士汶，2000，《浅层句法分析方法概述》，载《当代语言学》2000年第2期
- Abney, Steven, 1991, Parsing by chunks, in Robert Berwick, et al, eds. Principle-based parsing, Dordercht: Kluwer Academic Publishers.
- Church, K. 1988, A stochastic parts program and noun phrase parser for unrestricted text. In Proceedings of the Second Conference on Applied Natural Language Processing, pp.136-143.
- Ramshaw L. & M. Marcus, 1995, Text chunking using transformation-based learning, In Proceedings of the3rd Workshop on Very Large Corpora.

复习思考题

1) 构造带概率的CFG规则，用向内算法计算句子“老虎咬死了猎人的狗”的概率
2) 构造带概率的CFG规则，用Viterbi算法计算句子“他对学校的意见很大”的最佳分析树
3) 构造合适的正则表达式，找出句子“一只大老虎咬死了老猎人的小花狗”中的基本名词短语（不含“的”字的NP）