



统计剖析 (Statistical Parsing)

王小捷
智能科学与技术中心
北京邮电大学

大纲



■引言

■PCFG

■基于PCFG的句法分析

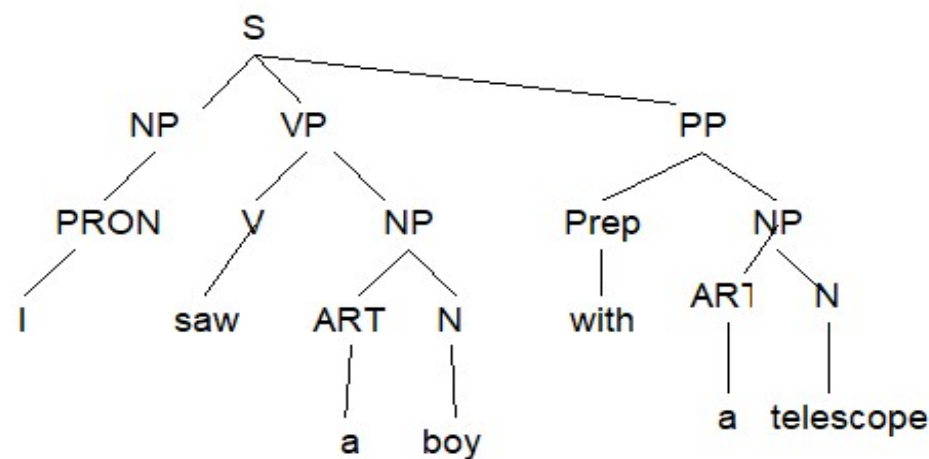
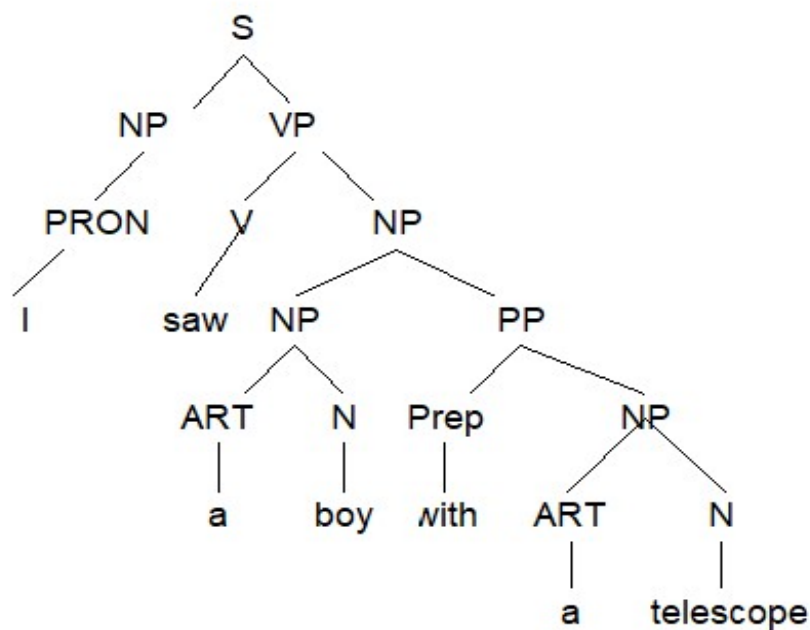
■LPCFG

■合一语法

■总结

引言

- 基于CFG的句法剖析：CKY和Earley能找出所有可能的结构：I saw a boy with a telescope.





■进一步的问题：

■这两棵句法树哪个更适合描述该句子？句法歧义

■CKY\Earley没有更多信息可以帮助回答这个问题

■怎么办？





■进一步的问题：

■这两棵句法树哪个更适合描述该句子？句法歧义

■CKY\Earley没有更多信息可以帮助回答这个问题

■怎么办？

■回忆之前类似的问题

■切分： XJ/Y 还是 X/JY ； XY 还是 X/Y

■解决方法：基于概率的思路



■基于概率的思路：

- 给每棵树赋予一个概率值，比较哪个棵树的概率值大。

■因此，问题转变为：

■如何给每棵树赋予概率？

- 回忆之前：如何给每个句子赋予概率？
- 例如：基于句子的子单元(bigram)的概率，句子的概率是bigram概率的组合。
- 优点：组合性、重用性



- 类似：树的概率由一些子单元的概率组合而成？
- 树的子单元：子树
- 而每棵子树是一个CFG规则
- 因此，子单元概率即为CFG每个规则的概率
- 即赋予每个CFG规则以概率！
- → 概率CFG(PCFG)

大纲



- 引言

- PCFG

- 基于PCFG的句法分析

- LPCFG

- 合一语法

- 总结



概率上下文无关语法(PCFG)

- 定义
- 一致性
- 如何利用PCFG进行句法消歧
 - 消歧
 - 语言模型



PCFG – 定义

- N: 非终止符集合
- T: 终止符集合
- S: 开始符
- R: 产生式规则集，每一个规则形如：
 - $A \rightarrow \beta \quad [\gamma]$
 - $A \in N$, β 是 $(N \cup T)^*$ 的子集
 - $0 \leq \gamma \leq 1$, $P(\beta|A) = \gamma$, 且 $\sum_{\beta} P(A \rightarrow \beta) = 1$
- 与CFG的主要不同就在于为规则引入了概率



■一个简单的PCFG例子

S	→	NP	VP	1.0	
NP	→	DET	N	0.4	} 一个非终止符所有可能的 产生式规则的概率和为1
NP	→	PRON		0.6	
VP	→	V		1	
DET	→	a the		0.4 0.6	
N	→	boy girl		0.5 0.5	
V	→	smile smiles cry cries		0.5 0.5	



■ PCFG – 一致性

■ 基于一个PCFG的一个语言中所有句子的概率和为1

■ 例：

■ $S \rightarrow NP VP$ 1.0

■ $VP \rightarrow V NP$ 0.7

■ $VP \rightarrow V$ 0.3

■ $NP \rightarrow DET N$ 0.4

■ $NP \rightarrow PRON$ 0.6



■PCFG：一致性

■一致性：一个PCFG产生的所有可能句子的概率总和为1。

■一个句子的产生过程及其概率：a boy smiles

■ $S \rightarrow NP VP$ 1

■ $NP \rightarrow DET N$ 0.4

■ $DET \rightarrow a$ 0.4

■ $N \rightarrow boy$ 0.5

■ $VP \rightarrow V$ 1

■ $V \rightarrow smiles$ 0.5

S	\rightarrow NP	VP	1.0
NP	\rightarrow DET	N	0.4
NP	\rightarrow PRON		0.6
VP	\rightarrow V		1
DET	\rightarrow a the		0.4 0.6
N	\rightarrow boy girl		0.5 0.5
V	\rightarrow smile smiles ...		0.5 0.5

■ $P(a\ boy\ smile) = 1 * 0.4 * 0.4 * 0.5 * 1 * 0.5$



■ PCFG – 不具一致性的例子

■ 1) $S \rightarrow w$ $1/3$ 2) $S \rightarrow S S$ $2/3$

■ w $1/3$

■ $w w$ $2/3 * 1/3 * 1/3 = 2/27$

■ $w w w$ $(2/3)^2 * (1/3)^3 * 2 = 8/243$

■ ...

■ $P(L) = 1/3 + 2/27 + 8/243 + \dots = 1/2$

■ 递归带来的

■ 不影响相对大小，所以对于剖析应用不是问题

■ Not a problem if you estimate from parsed treebank (Chin and Geman 1998, from SNLP)



PCFG中的上下文无关假设

■ 概率上下文无关

■ 规则是上下文无关的规则

■ 同CFG $A \rightarrow \beta$

■ 规则的概率值也是上下文无关的

■ 三个方面的要求

■ 位置无关性

■ 上下文无关性

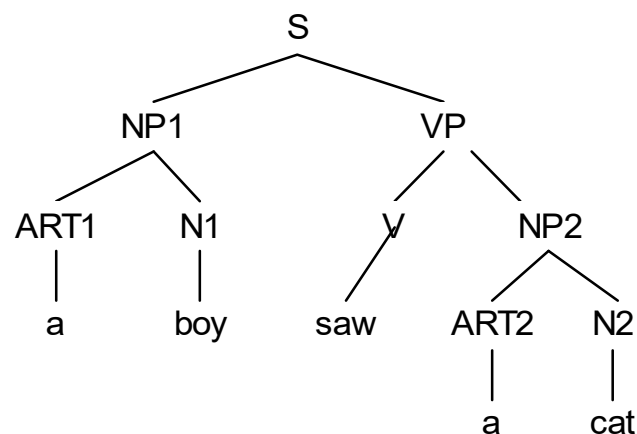
■ 祖先节点无关性



■位置无关性假设

■子节点概率与子节点所管辖的字符串在句子中的位置无关，即

\forall 位置 m , $P(N_{m(m+c)} \rightarrow \zeta)$ 相同



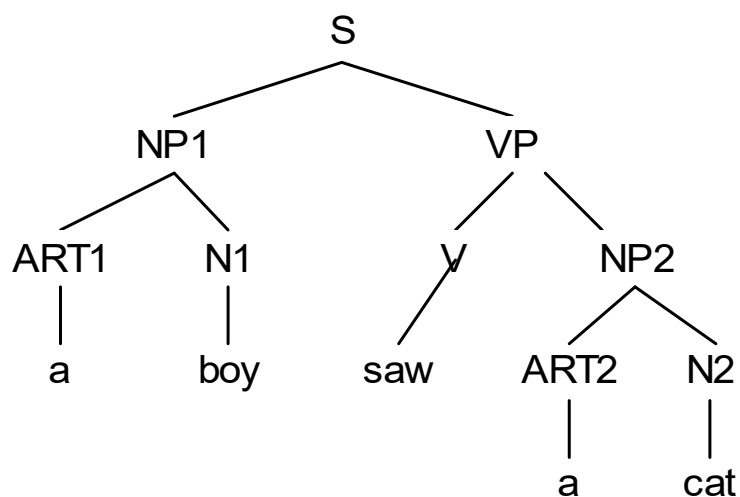
两个位置的ART \rightarrow a 概率相同



■前后节点无关性

■子节点概率与不受子节点管辖的其他符号串无关，即：

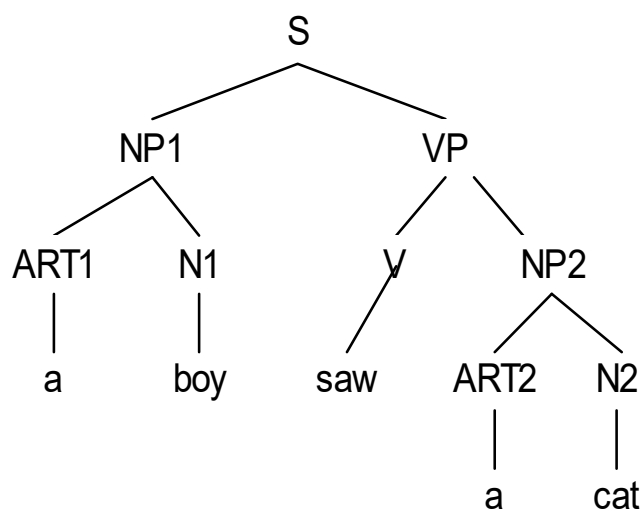
$$P(N_{mn} \rightarrow \zeta \mid \text{从m 到n 之外的其他 词}) = P(N_{mn} \rightarrow \zeta)$$



■祖先节点无关性

■子节点概率与导出该节点的所有祖先节点无关，即：

$$P(N_{mn} \rightarrow \zeta \mid \text{节点的任何祖先节点}) = P(N_{mn} \rightarrow \zeta)$$

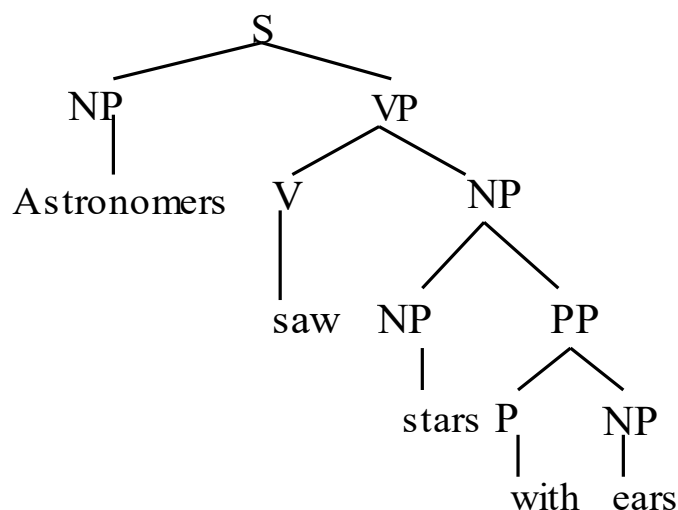




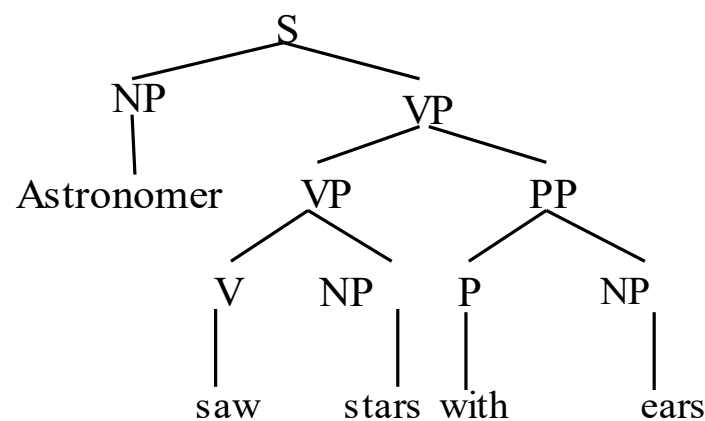
- 有了PCFG，则有了每个规则的概率，也即有了句法树中每个子树的概率，基于此进行句法树选择的一种直观方法是：
- 1)先基于CFG按CKY等算法得到所有可能的句法树：
- 2)为句法树种的每棵子树从PCFG中继承概率
- 3)依据子树概率计算树的概率，选择具有最大概率的树为最可能的树。

■ 例如：Astronomers saw stars with ears.

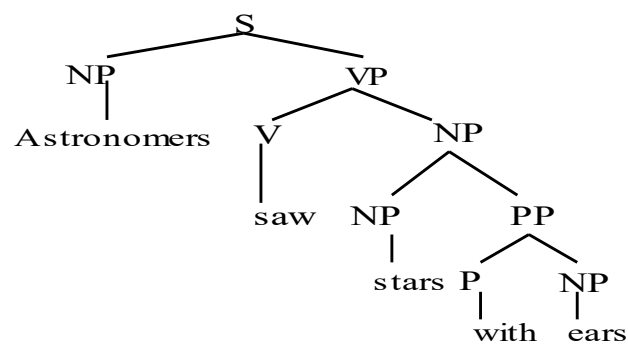
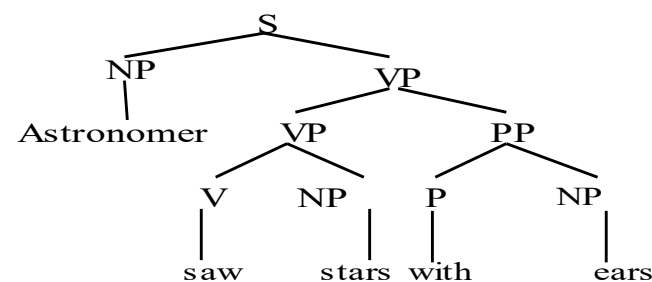
■ 1) 先基于CFG按CKY等算法得到所有可能的句法树：



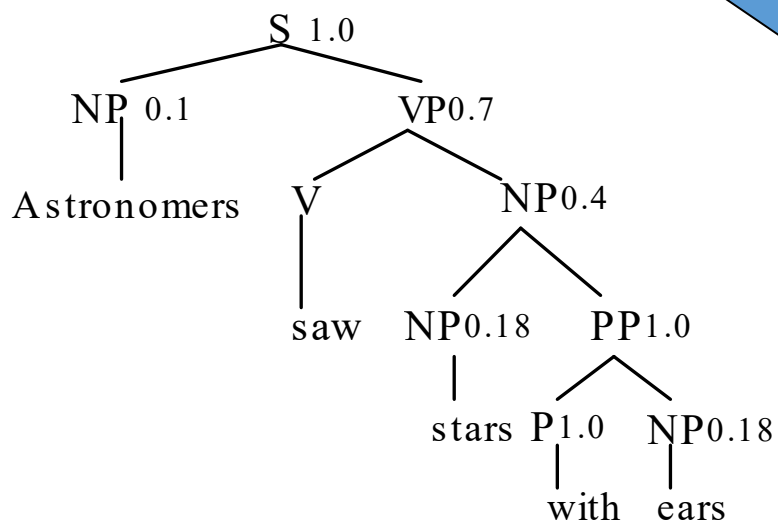
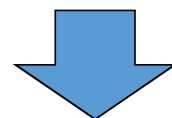
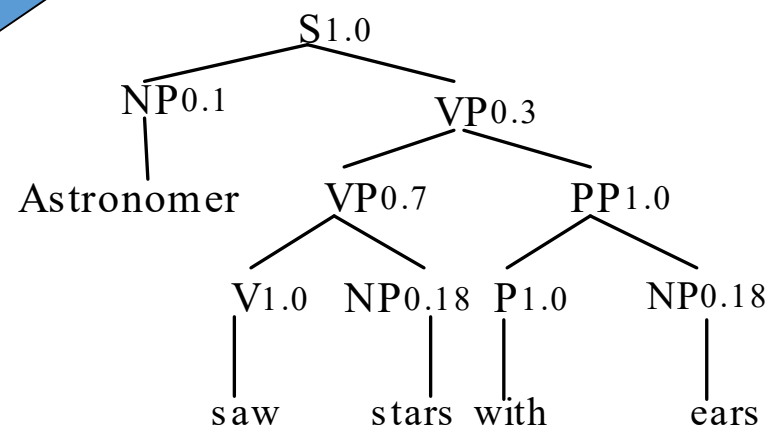
(a) T_1



(b) T_2

(a) T_1 (b) T_2

■2)依据PCFG规则的概率给每个子树加上概率

(a) T_1 (b) T_2



■ 3) 依据子树概率计算每棵树的概率，具有最大概率的树即为最合适的句法分析树

■ 依据子树概率计算树概率的方法？

- $P(T_i, S) = \prod P(\text{RHS}_j / \text{LHS}_j)$

- 句法分析树中所用到的规则的概率乘积

- 则上图两棵句法分析树，可分别计算其概率值为

$$P(T_1, S) = 1.0 \times 0.1 \times 0.7 \times 1.0 \times 0.4 \times 0.18 \times 1.0 \times 1.0 \times 0.18 = 0.0009072$$

$$P(T_2, S) = 1.0 \times 0.1 \times 0.3 \times 0.7 \times 1.0 \times 0.18 \times 1.0 \times 1.0 \times 0.18 = 0.0006804$$

■ 依据概率大小选择合适的树

- $\hat{T}(S) = \operatorname{argmax} P(T|S) = \operatorname{argmax} (P(T, S) / P(S))$

- $= \operatorname{argmax} P(T, S)$

- 上例中：

- $T_1 > T_2$ ，因此 T_1 更可能是正确的分析树。

■ 3) 依据子树概率计算每棵树的概率，具有最大概率的树即为最合适的句法分析树

■ 依据子树概率计算树概率的方法？

■ $P(T_i, S) = \prod P(\text{RHS}_j / \text{LHS}_j)$

■ 句法分析树中所用到的规则的概率

■ 则上图两棵句法分析树可分别计算

$$P(T_1, S) = 1.0 \times 0.1 \times 0.7 \times 1.0$$

$$P(T_2, S) = 1.0 \times 0.1 \times 0.3 \times 0.7$$

■ 依据概率大小选择合适句法分析树

■ $\hat{T}(S) = \operatorname{argmax} P(T|S) = \operatorname{argmax}_T P(T, S)$

■ $= \operatorname{argmax} P(T, S)$

■ 上例中：

■ $T_1 > T_2$ ，因此 T_1 更可能是正确句法分析树

$$P(T_1), P(T_1|S), P(T_1, S)$$

$$P(T_1) = \sum_s P(T_1, S)$$

$$P(T_1, S) = P(T_1|S)P(S)$$

$$P(T_1|S) = \frac{P(T_1, S)}{P(S)} = \frac{P(T_1, S)}{\sum_T P(T, S)}$$

9072

804



■顺便得到一种基于句法树的概率来计算句子概率的方法：

■
$$P(S) = \sum_T P(T, S)$$

$$P(T_1, S) = 1.0 \times 0.1 \times 0.7 \times 1.0 \times 0.4 \times 0.18 \times 1.0 \times 1.0 \times 0.18 = 0.0009072$$

$$P(T_2, S) = 1.0 \times 0.1 \times 0.3 \times 0.7 \times 1.0 \times 0.18 \times 1.0 \times 1.0 \times 0.18 = 0.0006804$$

■ $P(s)=0.0015876$

■回忆之前计算句子概率的方法：

■基于n-gram语言模型的句子概率计算



■ PCFG vs. N-gram

- N-gram侧重考察词汇搭配

- 概率上下文无关侧重考虑句法结构

■ PCFG优势:

- 语法线索:

 - I boy a am

- 长距相依:

 - Fred watered his father's small garden:



■ PCFG vs. N-gram

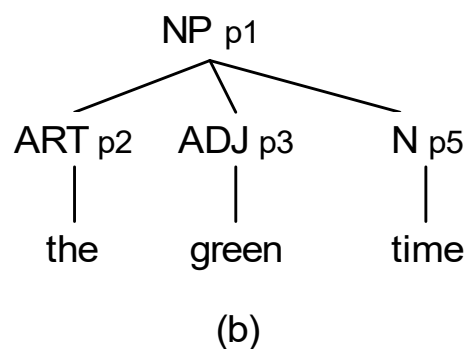
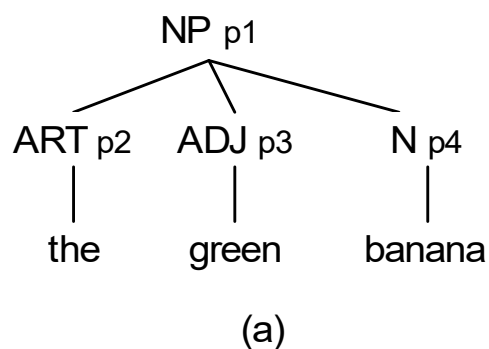
- N-gram侧重考察词汇搭配

- 概率上下文无关侧重考虑句法结构

■ N-Gram优势

- 词汇信息：

- the green banana vs. the green time





- 前述方法先基于CFG剖析出所有可能句法树，再用PCFG的概率来判决，间接。
- 是否可以基于PCFG进行句法分析，直接得到具有最大概率的句法树？
- → PCKY 剖析算法(CKY的概率扩展)

大纲

- 引言
- PCFG
- 基于PCFG的句法分析
- LPCFG
- 合一语法
- 总结



基于PCFG的句法分析

■从CKY(基于CFG)到PCKY (基于PCFG)

0	Book	1	the	2	flight	3	through	4	Huston	5
S, VP, Verb, Nominal, Noun (0,1)	φ		S, VP, X2 (0,3)	φ		S1, VP1, S2, VP2, S3 (0,5)				
	Det (0,2)		NP (0,3)	φ		NP (0,4)				
		Det (1,2)		NP (1,3)	φ					
			Nominal, Noun (2,3)	φ		Nominal (2,4)				
				Prep (2,4)		PP (2,5)				
					Prep (3,4)					
						NP, Proper- Noun (4,5)				

- $[0,1,S] = p(S \rightarrow \text{book})$
- $[0,1,VP] = p(VP \rightarrow \text{book})$
- $[0,1,Verb] = p(Verb \rightarrow \text{book})$
- $[0,1,Nom] = p(Nom \rightarrow \text{book})$
- $[0,1,Nou] = p(Nou \rightarrow \text{book})$

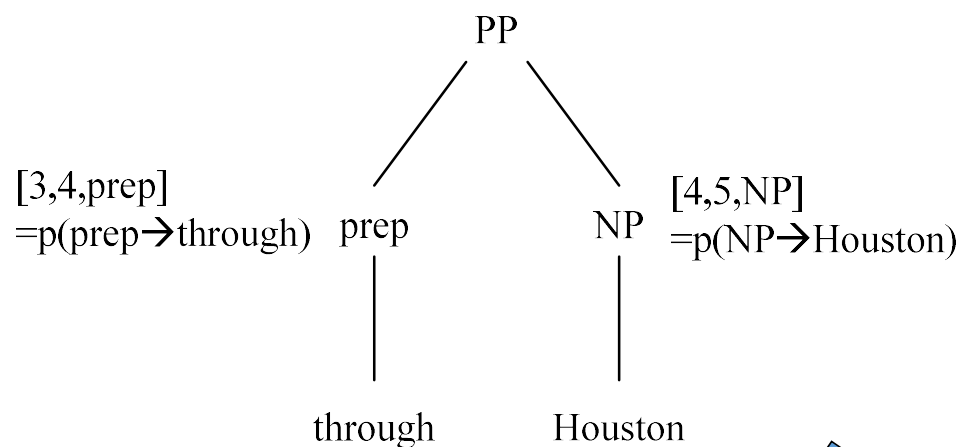
基于PCFG的句法分析

■从CKY(基于CFG)到PCKY (基于PCFG)

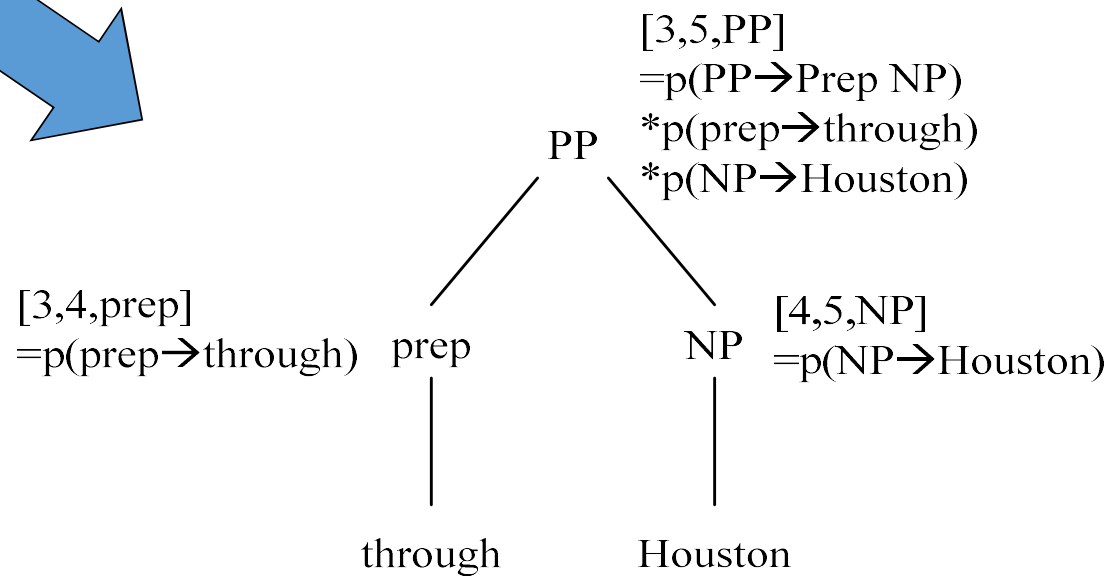
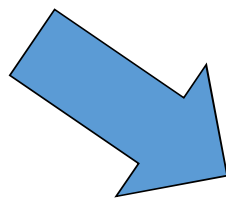
0	Book	1	the	2	flight	3	through	4	Huston	5
S, VP, Verb, Nominal, Noun (0,1)	φ	S, VP, X2 (0,3)	φ	S1, VP1, S2, VP2, S3 (0,5)						
	Det (1,2)	NP (1,3)	φ	NP (1,5)						
		Nominal, Noun (2,3)	φ	Nominal (2,5)						
			Prep (3,4)	PP (3,5)						
				NP, Proper- Noun (4,5)						

- $[0,1,S] = p(S \rightarrow \text{book})$
- $[0,1,VP] = p(VP \rightarrow \text{book})$
- $[0,1,Verb] = p(Verb \rightarrow \text{book})$
- $[0,1,Nom] = p(Nom \rightarrow \text{book})$
- $[0,1,Nou] = p(Nou \rightarrow \text{book})$

$$\begin{aligned}
 \blacksquare [3,5,PP] = & \\
 & p(PP \rightarrow \text{Prep NP}) \\
 & * [3,4,Prep] \\
 & * [4,5,NP]
 \end{aligned}$$



■ 子树累积概率





基于PCFG的句法分析

■从CKY(基于CFG)到PCKY (基于PCFG)

0	Book	1	the	2	flight	3	through	4	Houston	5
S, VP, Verb, Nominal, Noun	ϕ	S, VP, X2	ϕ	S1, VP1, S2, VP2, S3						
(0,1)	(0,2)	(0,3)	(0,4)	(0,5)						
	Det	NP	ϕ	NP						
	(1,2)	(1,3)	(1,4)	(1,5)						
		Nominal, Noun	ϕ	Nominal						
		(2,3)	(2,4)	(2,5)						
			Prep	PP						
			(3,4)	(3,5)						
				NP, Proper- Noun						
				(4,5)						

$$\blacksquare \hat{S} = \operatorname{argmax} \{ [0, 5, Si] \}$$

$$Si \in \{s1, s2, s3\}$$

$$[0,5,S_1]=P(T_1,S)$$

$$[0,5,S_2]=P(T_2,S)$$

$$[0,5,S_3]=P(T_3,S)$$

■ $[3,5,PP]=$

■ $p(\text{PP} \rightarrow \text{Prep NP})$

- $*[3,4,\text{Prep}]$

- ***[4,5,NP]**

- $[0,1,S]=p(S \rightarrow \text{book})$
- $[0,1,VP]=p(VP \rightarrow \text{book})$
- $[0,1,Verb]=p(Verb \rightarrow \text{book})$
- $[0,1,Nom]=p(Nom \rightarrow \text{book})$
- $[0,1,Nou]=p(Nou \rightarrow \text{book})$



■ PCFG vs. CFG

- PCFG的优势：消歧

- PCFG的代价：？每个规则附加一个概率值

■ 具有概率的规则 vs 无概率的规则

- 如果在语法规则完全由人工总结获得时，规则的概率会更难获得。

- 但是通过大规模语料自动学习语法规则(语法归纳)时，则可以在获得规则的同时也获得概率，两种情况：



学习PCFG中的概率

- 1) 当有大规模剖析语料Treebank时: MLE
 - $p(A \rightarrow \gamma) = \frac{\#(A \rightarrow \gamma)}{\sum_x \#(A \rightarrow x)}$
- 2) 当没有Treebank时(有确定性CFG):
 - Inside-outside算法
 - 初始化:
 - 随机初始化CFG, 构建出概率为 p_0 的PCFG: PCFG(p_0)
 - 选择语料C, 用基于PCFG(p_0)的剖析器剖析语料C得到树库T
 - 循环:
 - E-step: 用树库T估计新的概率PCFG(p_i)
 - M-step: 用基于PCFG(p_i)的剖析器剖析语料C得到树库T
 - $i++$
 - 当收敛时结束循环

PCFG的问题与改进



■ 概率独立性假设导致的问题：



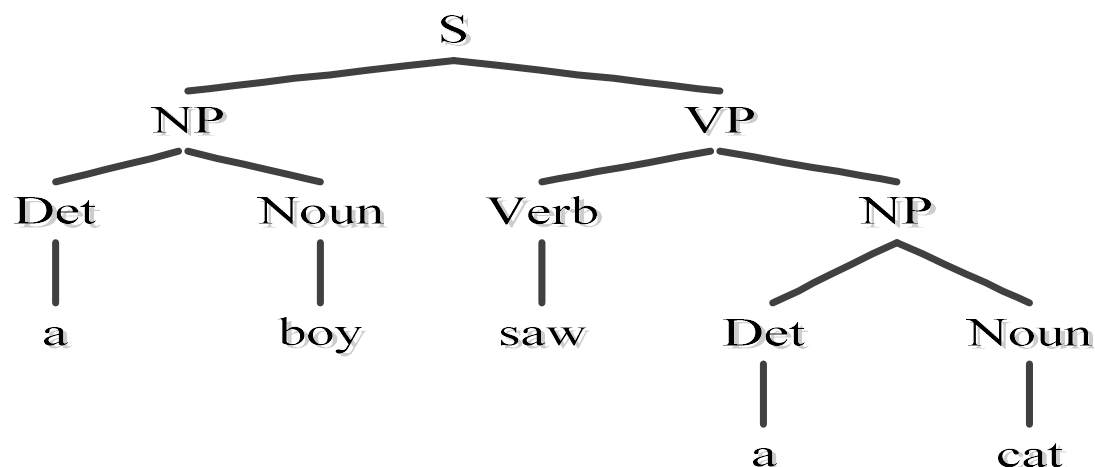
■上下文独立性假设：

- PCFG中规则(如 $NP \rightarrow Det \ Noun$; $NP \rightarrow Pron$)的概率独立于其在树中的使用位置。
- 实际语言现象：处于句法主语位置和句法宾语位置时的不同NPs概率有差异 (Francis al., 1999), 也即：规则的概率与上下文有关。

	$NP \rightarrow Pron$	$NP \rightarrow Det \ Noun$等非Pronoun的NP
Subject	91%	9%
Object	34%	66%



- 使得PCFG更好地吻合语言现象的一种改进
 - 直观思路：为不同上下文的相同句法单元(NP等) 指派不同的概率。
 - 问题：如何区分不同上下文
 - 一个单元在不同位置时，其父节点可能不同，父节点对其在句子中的作用有很好的指示作用：例如 对于 NP：
 - NP的父节点是S时，该NP常常是做主语的(例如 a boy)
 - NP的父节点是VP时，该NP常常是做宾语的 (例如 a cat)

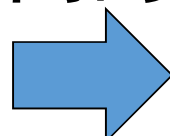


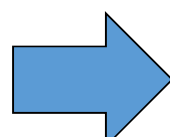


■因此，标记一个单元的父亲节点可以区分该单元的位置和句法作用，并依据实际语言现象指派不同的概率。

■例如 对于 NP:

■可以分裂为两个不同的单元，分别得到规则:

■ $NP \rightarrow Det \ Noun \ p_1$  $NP^S \rightarrow Det \ Noun \ p_{11}=0.09$
 $NP^VP \rightarrow Det \ Noun \ p_{21}=0.66$

■ $NP \rightarrow Pron \ p_2$  $NP^S \rightarrow Pron \ p_{12}=0.91$
 $NP^VP \rightarrow Pron \ p_{22}=0.34$



■依据父节点标记进行节点分裂带来的新问题：

- 分裂得越细，可以区分得情况就越多

- 但是分裂得越细，增加的规则越多，在数据确定时减小了对每个规则的训练样例。

■因此，重要的是对于特定的训练集分到正确的粒度层次。

- Klein2003用手工规则来找最优规则数

- Petrov2006自动找最优分

-

PCFG的问题和改进



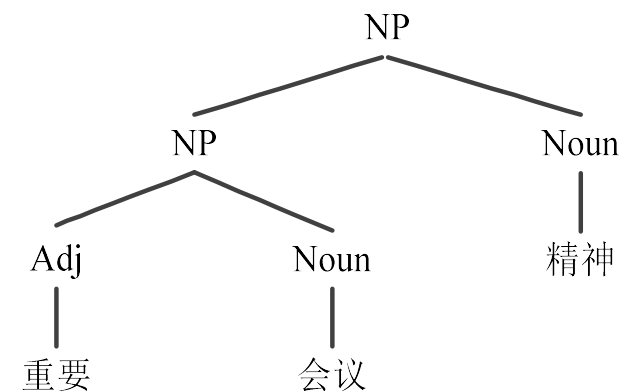
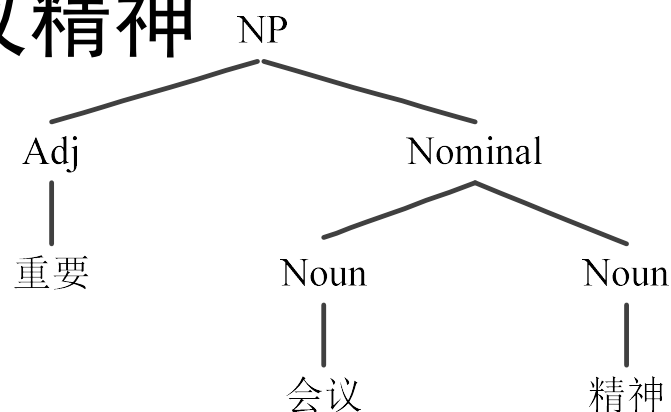
■ 概率独立性假设导致的问题：

■ 缺乏词汇化条件导致的问题：



■缺乏词汇化条件

■例:重要会议精神



■ $P(T1,s) = P(NP \rightarrow Adj \text{ Nominal}) P(Adj \rightarrow \text{重要}) P(Nominal \rightarrow Noun \text{ Noun}) P(Noun \rightarrow \text{会议}) P(Noun \rightarrow \text{精神})$

■ $P(T2,s) = P(NP \rightarrow NP \text{ Noun}) P(NP \rightarrow Adj \text{ Noun}) P(Adj \rightarrow \text{重要}) P(Noun \rightarrow \text{会议}) P(Noun \rightarrow \text{精神})$



■即意味着，选择哪颗树，取决于

■ $P(\text{NP} \rightarrow \text{Adj Nominal}) P(\text{Nominal} \rightarrow \text{Noun Noun})$

■ $P(\text{NP} \rightarrow \text{NP Noun}) P(\text{NP} \rightarrow \text{Adj Noun})$

■这两个乘积的大小，与具体词汇无关！

■更明确地，如果这几个规则的概率给定后，具有如下POS序列的任何句子：

■Adj Noun Noun

■都总是会剖析为其中相同的一种结构，不会依据词不同而发生变化

■这与实际语言现象是不太符的！



■例子

■重要会议精神 第? 种

■圆形 会议 桌 第一种

■新 会议 大楼 第一种

■重大 会议 期间 第二种

■全体 会议 公告 第二种

■结论: 是那种结构应该会依据具体的Adj、Noun是什么词(词汇化)而有变化, 而PCFG不能建模此现象!



- 更多类似情形：
- 英语PP附着结构的情况：
- Astronomers saw stars with ears.
- Astronomers saw stars with telescopes.
- Astronomers saw stars with moons.

- 结构选择与词有关



- 更多类似情形:

- 并列结构的情况

- Dogs in house and cats

- [Dogs in house] and [cats]

- Dogs in house and yard

- Dogs in [house and yard]

结构选择与词有关



■词汇对规则应用应该是有影响的！

大纲



- 引言
- PCFG
- 基于PCFG的句法分析
- LPCFG
- 合一语法
- 总结



■关于词汇化的改进 (LPCFG)

■语法的词汇化：以规则 $VP \rightarrow VBD \ NP \ PP$ 为例

■首先把各个非终止符的头词加入规则→

■ $VP(dumped) \rightarrow VBD(dumped) \ NP(sacks) \ PP(into)$

■进一步把各个头的POS 加入规则→

■ $VP(dumped, VBD) \rightarrow VBD(dumped, VBD)$
 $NP(sacks, NNS) \ PP(into, P)$

■这条规则使用时不仅POS对上，词也要对上。



■规则的概率估计

■ $R1: VP(dumped, VBD) \rightarrow VBD(dumped, VBD)$

$NP(sacks, NNS) \quad PP(into, P)$

$$p(R1) = \frac{count(R1)}{count(VP(dumped, VBD))}$$

■这样的规则非常特定化

- 前面在讲依据父节点标记来缓解概率上下文假设的问题时讲到过：进行节点分裂带来的问题是分裂得越细，增加的规则越多，在数据确定时减小了对每个规则的训练样例。
- 现在这么特定化，更难以找到较大规模的语料进行估计。
- 数据太稀疏，联合分布不好估计，引入独立假设以进行合理的估计是一种方案

The Collins Parser

■ 规则 $LHS \rightarrow L_n L_{n-1} \dots L_1 H R_1 \dots R_{n-1} R_n$ 的生成假设

■ 准备:

■ $LHS \rightarrow \textcolor{red}{STOP} L_n L_{n-1} \dots L_1 H R_1 \dots R_{n-1} R_n \textcolor{red}{STOP}$

■ 生成

■ 生成头

■ 生成头左边: 由头到左边STOP

■ 生成头右边: 由头到右边STOP

例：

$VP(dumped, VBD) \rightarrow VBD(dumped, VBD) NP(sacks, NNS) PP(into P)$

■准备：

■确认头

■ $VP(dumped, VBD) \rightarrow VBD(dumped, VBD) NP(sacks, NNS) PP(into P)$

■加STOP

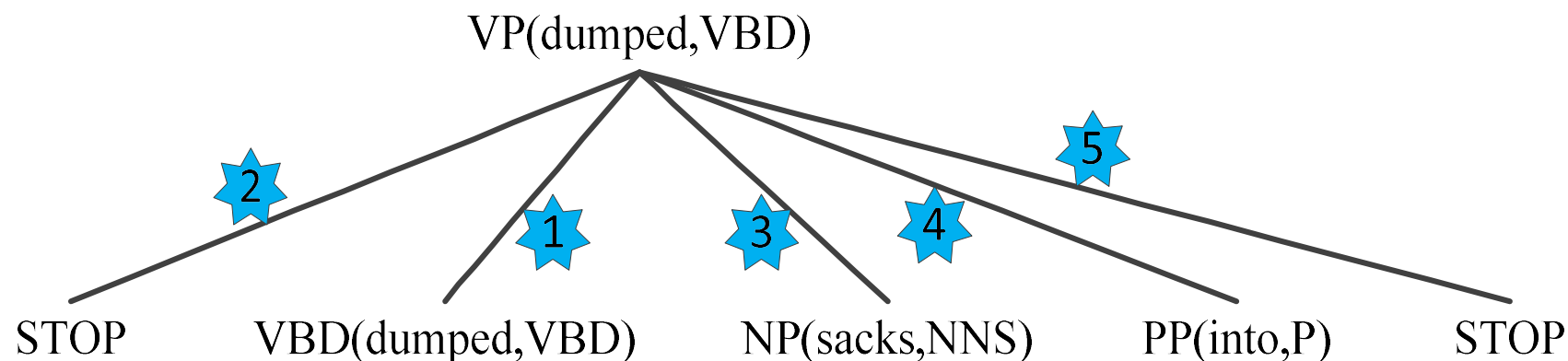
■ $VP(dumped, VBD) \rightarrow STOP VBD(dumped, VBD)$

$NP(sacks, NNS) PP(into P) STOP$

■生成： next page



头词



- 1) 生成头, 概率为 $p(H|LHS) = P(VBD(dumped, VBD) | VP(dumped, VBD))$
- 2) 生成左边第一个(当前是STOP), 概率为 $p(STOP | VP(dumped, VBD) VBD(dumped, VBD))$
- 3) 生成右边第一个(NP(sacks, NNS)), 概率为 $p(NP(sacks, NNS) | VP(dumped, VBD) VBD(dumped, VBD))$
- 4) 生成右边第二个(PP(into, P)), 概率为 $p(PP(into, P) | VP(dumped, VBD) VBD(dumped, VBD))$
- 5) 生成右边第三个(STOP), 概率为 $p(STOP | VP(dumped, VBD) VBD(dumped, VBD))$

■在这样的生成假设下:

$$\blacksquare P(VP(dumped, VBD)$$

$$\rightarrow VBD(dumped, VBD) NP(sacks, NNS) PP(into P))$$

$$= PH(VBD|VP, dumped)$$

$$\times PL(STOP|VP, VBD, dumped)$$

$$\times PR(NP(sacks, NNS)|VP, VBD, dumped)$$

$$\times PR(PP(into, P)|VP, VBD, dumped)$$

$$\times PR(STOP|VP, VBD, dumped)$$

■数据稀疏比原来要缓解



■ Stanford parser (LPCFG)

■ <https://nlp.stanford.edu/software/lex-parser.html#About>

■ Berkeley Parser(PCFG-LA)

■ <https://github.com/slavpetrov/berkeleyparser>

■ Zong书 205-207：性能比较

大纲



- 引言
- PCFG
- 基于PCFG的句法分析
- LPCFG
- 合一语法
- 总结



■从CFG→PCFG

- CFG不能进行句法结构消歧歧义

- CKY→PCKY

■从PCFG → LPCFG

- PCFG和词汇无关导致不能利用上下文消歧

- 词汇化: LPCFG→Collins Parser

■CFG的其他问题



CFG的其他问题 --- 1

■关于动词的CFG规则:

■ $VP \rightarrow \text{Verb}$

1)

■ $VP \rightarrow \text{Verb NP}$

2)

■ $VP \rightarrow \text{Verb NP NP}$

3)

■ ...

- $VP \rightarrow \text{Verb}$ 1)
- $VP \rightarrow \text{Verb NP}$ 2)
- $VP \rightarrow \text{Verb NP NP}$ 3)



■观察如下两个动词(V):

- denied
- disappeared

■利用上述上下文无关语法可以判定如下两个结构是合法的:

- He **denied the accusation.** \checkmark $S(\text{NP VP}(\text{Verb NP}))$
- The problem **disappeared.** \checkmark $S(\text{NP VP}(\text{Verb}))$

■但是, 如下两个不合语法的句子在上述CFG下同样也被认为是合法:

- He denied. $? S(\text{NP VP}(\text{Verb}))$
- He disappeared the problem. $? S(\text{NP VP}(\text{Verb NP}))$

■这是个问题!

■存在这个问题的原因：不同的动词只能用于某些特定的CFG中

■解决办法，如下两步：

■1)区分不同动词类别，引入动词次范畴

■denied: V → denied: TV

■disappeared: V → disappeared: IV

■2)为不同类分别设计重写规则：

■VP → V → VP → IV

■VP → V NP → VP → TV NP

■VP → V NP NP → VP → DTV NP NP

■则此时就可以正确判定如下的不合法结构

■He **denied**. S(NP VP(TV)) ×

■He **disappeared the problem**. S(NP VP(IV NP)) ×



CFG的其他问题 --- 2

- 观测CFG规则: $S \rightarrow NP \ VP$
- 基于上述规则, 下面句子合法:
 - A bird sings. (NP VP)
 - Birds sing. (NP VP)
- 但规则不能判定下面句子非法:
 - A bird sing. (NP VP)
 - Birds sings. (NP VP)
- 原因: 该规则不能区分单复数



■解决办法:

■1)引入新范畴

■NP-SING、VP-SING

■NP-PLU、VP-PLU

■2)将 $S \rightarrow NP \ VP$ 改为如下两个规则:

■ $S \rightarrow NP-SING \ VP-SING$

■ $S \rightarrow NP-PLU \ VP-PLU$

■则可认定下面句子非法

■A bird sing. (NP-SING VP-PLU) ×

■Birds sings. (NP-PLU VP-SING) ×



进一步：问题1和问题2同时出现

- 例如：He **deny**.
- 一方面： VP(TV) 不合法，需要改为：
 - He **deny** +NP. VP(TV NP)
- 另一方面：S(NP-SING **VP-PLU**)不合法，需要进一步改为：
 - He **denies** +NP. S(NP-SING VP-SING(TV NP))
 - 或改为：
 - They **deny** +NP. S(NP-PLU VP-PLU(TV NP))
- 可以将这些改变整合在一起：
 - 需要考虑引入更多范畴，例如：
 - VP: (SING, PLU)*(IV,TV,DTV)



■则有：

■ $S \rightarrow NP-SING \quad VP-SING$

■ $S \rightarrow NP-PLU \quad VP-PLU$

■ $VP-SING \rightarrow IV-SING$

■ $VP-PLU \rightarrow IV-PLU$

■ $VP-SING \rightarrow TV-SING \quad NP$

■ $VP-PLU \rightarrow TV-PLU \quad NP$

■ $VP-SING \rightarrow DTV-SING \quad NP \quad NP$

■ $VP-PLU \rightarrow DTV-PLU \quad NP \quad NP$



■上述策略可以解决相应的问题，但是带来了新的问题：

■导致引入更多的新范畴，导致规则数目的增长。更多的此类语言现象，导致组合增长！

■新增范畴数*新增范畴数*新增范畴数...

■从揭示语言结构的目的来看这种解决方案缺乏深度：

■例如：引入IV-SING 与 IV-PLU，从符号层来看，其关系不明，但是二者关系紧密！

■规则冗余度大，利用这样的语法系统进行句法分析从实现上来看不经济

■ $S \rightarrow NP-SING \quad VP-SING$

■ $S \rightarrow NP-PLU \quad VP-PLU$



■控制范畴数增长！

■控制规则数量！

■揭示结构的内部关联！

■ ? ? ?



特征结构的引入

■观察：IV-SING, IV-PLU...



■有内部特征结构：

$$IV - SING = \begin{bmatrix} POS & V \\ AGR & 3s \\ VAL & itr \end{bmatrix} \quad IV - PLU = \begin{bmatrix} POS & V \\ AGR & p \\ VAL & itr \end{bmatrix}$$

其中 AGR :

$3s$: 第三人称+单数

p : 复数



■特征结构的一般形式

$$F = \begin{bmatrix} FEATURE_1 & \dots & VALUE_1 \\ FEATURE_i & \dots & VALUE_i \\ FEATURE_n & \dots & VALUE_n \end{bmatrix}$$

F 为结构名称
 $FEATURE_i \quad (i = 1, \dots, n)$ 为特征名称
 $VALUE_i \quad (i = 1, \dots, n)$ 为特征值

例如: $IV - SING = \begin{bmatrix} POS & V \\ AGR & 3s \\ VAL & itr \end{bmatrix}$



■特征结构的函数表示：

$$F(FEATURE_i) = VALUE_i \quad (i = 1, \dots, n)$$

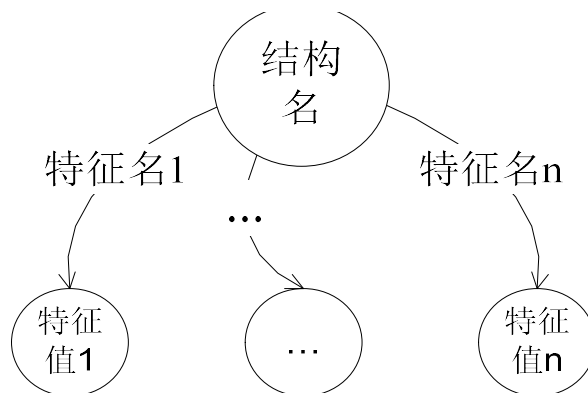
■书写方便

■任意矩阵表示都存在等价函数表示

■例如：

$$IV - SING = \begin{bmatrix} POS & V \\ AGR & 3s \\ VAL & itr \end{bmatrix} = \begin{array}{l} IV - SING(POS) = V \\ IV - SING(AGR) = 3s \\ IV - SING(VAL) = itr \end{array}$$

■特征结构的有向无循环图 (DAG: Directed Acyclic Graph)表示:

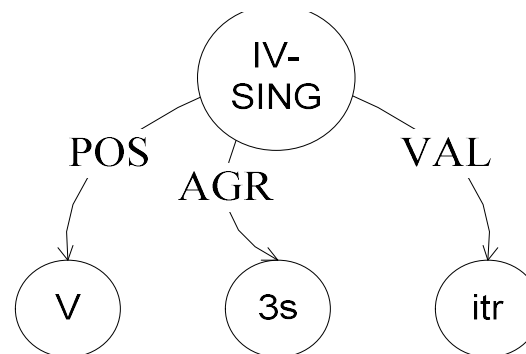


■图表式, 直观

■任意矩阵表示都存在等价DAG表示

■例如:

$$IV - SING = \begin{bmatrix} POS & V \\ AGR & 3s \\ VAL & itr \end{bmatrix} =$$





- 特征结构的括号(Bracket)表示:
- 结构名(特征名1=特征值1; ...; 特征名n=特征值n)
- 线形表示, 易于书写, 常用
- 任意矩阵表示都存在等价DAG表示
- 例如:

$$IV - SING = \begin{bmatrix} POS & V \\ AGR & 3s \\ VAL & itr \end{bmatrix} = IV-SING(POS=V; AGR=3s; VAL=itr)$$



原子特征、复杂特征

■ 原子特征

- 单一特征 $FEATURE_i$ ，当其值 $VALUE_i$ 无内嵌的特征结构。

■ 例如

- $POS=V$

- 而AGR可进一步分解为两个NUM、PER特征组成的结构

■ 复杂特征：包含多个原子特征



特征结构的基本性质

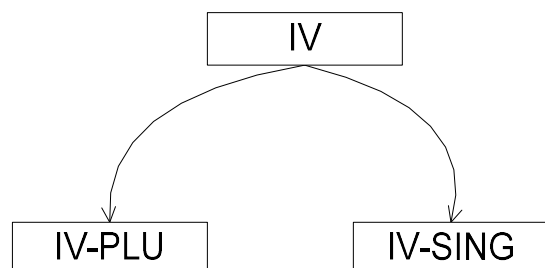
- 特征结构中各个特征间的次序是不重要，无需区分的。

$$IV-SING = \begin{bmatrix} POS & V \\ AGR & 3s \\ VAL & itr \end{bmatrix} = IV-SING = \begin{bmatrix} POS & V \\ VAL & itr \\ AGR & 3s \end{bmatrix}$$

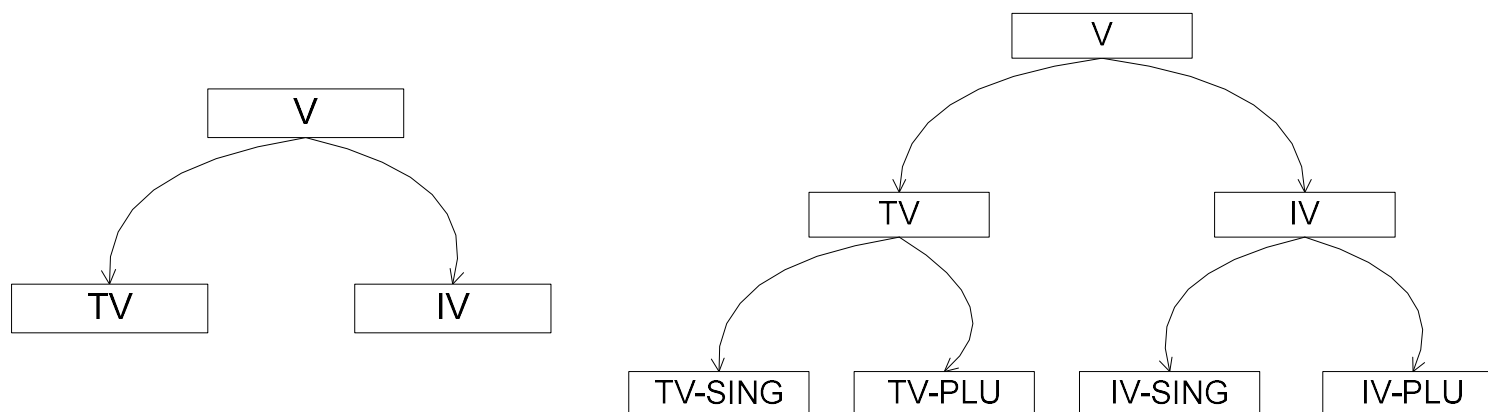
■特征结构使得被描述对象具有组织上的层次性

$$IV = \begin{bmatrix} POS & V \\ VAL & itr \end{bmatrix} \quad IV - SING = \begin{bmatrix} POS & V \\ VAL & itr \\ AGR & 3s \end{bmatrix} \quad IV - PLU = \begin{bmatrix} POS & V \\ VAL & itr \\ AGR & p \end{bmatrix}$$

■可得到:



- 通过利用少数几个特征来控制范畴数量的增加





- 特征结构中特征的选择是由被描述的对象以及面向的任务所决定的
 - 对于合句法判定任务
 - AGR: 重要
 - VAL: 重要
 - 对于论元结构识别任务
 - AGR: 不重要
 - VAL: 重要
 -

基于特征结构表示CFG

■ 例如: $S \rightarrow NP \ VP$

$$S \rightarrow \begin{bmatrix} POS & NP \\ AGR & a \end{bmatrix} \begin{bmatrix} POS & VP \\ AGR & a \end{bmatrix}$$



$$S \rightarrow (POS = NP \quad AGR ? a)(POS = VP \quad AGR ? a)$$

简写: $S \rightarrow NP(AGR?a) \ VP(AGR?a)$



揭示语言结构的关系：

■IV-SING与IV-PLU的关系

$$IV-SING = \begin{bmatrix} POS & V \\ AGR & 3s \\ VAL & itr \end{bmatrix} \quad IV-PLU = \begin{bmatrix} POS & V \\ AGR & p \\ VAL & itr \end{bmatrix}$$

IV-SING与IV-PLU的关系： 差异只在AGR特征



控制规则数目的增加：

$$S \rightarrow \begin{bmatrix} NP \\ ARG & a \end{bmatrix} \begin{bmatrix} VP \\ ARG & a \end{bmatrix}$$

■可以描述原来的两个规则

■ $S \rightarrow NP-SING \quad VP-SING$

■ $S \rightarrow NP-PLU \quad VP-PLU$



特征结构的运算：合一运算

■ 两个特征结构的合一运算定义为： $A \bar{\cup} B$

■ 若A、B均为原子，则：

- 如果 $A \cap B$ 非空，则 $A \bar{\cup} B = A \cap B$ ；
- 如果 $A \cap B$ 为空，则 $A \bar{\cup} B$ 为空

■ 若A、B为两个特征结构，则：

- 如果A中的特征 f ，有 $A(f)=w$ (w 为原子)，而该特征在B中没有定义，那么有 $A \bar{\cup} B(f)=w$ ；
- 如果B中的特征 f ，有 $B(f)=w$ (w 为原子)，而该特征在A中没有定义，那么有 $A \bar{\cup} B(f)=w$ ；
- 如果A中的特征 f ，有 $A(f)=w$ (w 为原子)，且该特征在B中有 $B(f)=w'$ (w' 为原子)，那么有 $A \bar{\cup} B(f)=w \cap w'$



例:

■特征结构

■N1(POS=N; ROOT=fish; AGR={3s,3p})

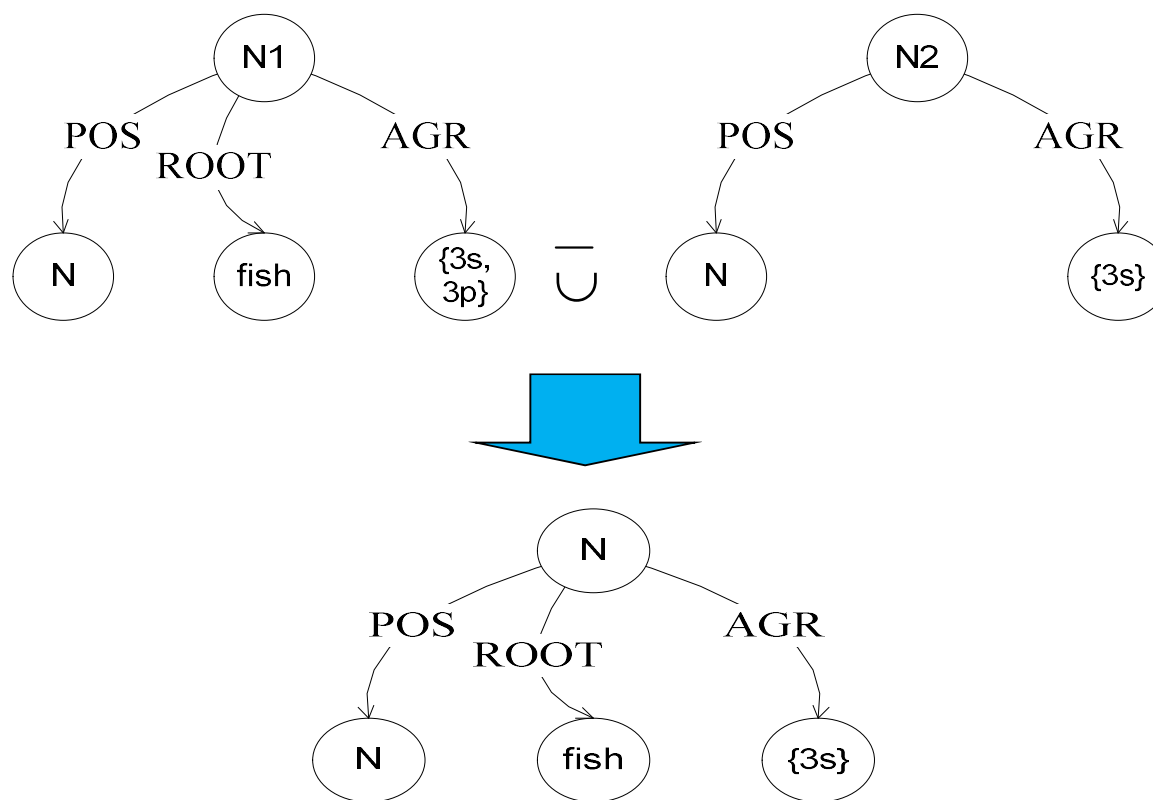
■与特征结构

■N2(POS=N; AGR={3s})

■的合一运算结果为: $N3 = N1 \bar{\cup} N2$:

■N3(POS=N; ROOT=fish; AGR={3s})

■用DAG表示:





■基于CFG规则的合一运算

■例如：两个特征结构 $NP(AGR=a1)$ 和 $VP(AGR=a2)$ 依据规则

$S \rightarrow NP(AGR?a) \ VP(AGR?a)$ 进行合一运算，则输出的S结构为：

■ $S(AGR=AGR_{NP} \cap AGR_{VP})$

■所以，如果 $a1=\{3s\}$ $a2=\{p\}$ 时， $S(AGR=\phi)$,即不能产生合法句子



基于特征结构的句法分析

■一般上下文无关语法 → 基于特征结构的语法

■ $NP \rightarrow DET\ N$

■ →

■ $NP(AGR?a) \rightarrow DET(AGR?a)\ N(AGR?a)$

■词的POS → 词的特征结构

■ w , 除了要求其POS信息之外, 还要获取其他需要用到的特征信息, 比如AGR等



基于特征结构的句法分析

- 在进行规则匹配时，除了结构名称的一致，还要匹配内部结构中每一个特征的一致性。
- 以Earley算法中的两个操作为例：
 - 在遇到状态 $NP \rightarrow Det \cdot Noun [0,1]$ 时
 - 执行扫描操作：读取 w_1 ，检查其是否有POS为Noun，若有，扫描操作成功，产生新状态：
 - $Noun \rightarrow w_1 \cdot [1, 2]$
 - 该状态激活完成操作，更新前述状态 $NP \rightarrow Det \cdot Noun [0,0]$ 为：
 - $\rightarrow NP \rightarrow Det Noun \cdot [0,2]$



■基于复杂特征结构的scan 操作

■在遇到状态 $NP(AGR=3s) \rightarrow Det(AGR=3s) \cdot Noun(AGR=3s) [0,1]$ 时

■执行扫描操作：读取 w_1 ，除了要求其POS有Noun之外，还要获取其AGR信息，与 $AGR=3s$ 进行合一。

■如果读取的词为： $w_1(POS=Det, AGR=3s)$ ，则产生新状态：

■ $Noun(AGR=3s) \rightarrow w_1 \cdot [1, 2]$

■该状态激活完成操作，更新前述状态 $NP(AGR=3s) \rightarrow Det(AGR=3s) \cdot Noun(AGR=3s) [0,1]$ 为：

■ $\rightarrow NP(AGR=3s) \rightarrow Det(AGR=3s) Noun(AGR=3s) [0,1] \cdot [0,2]$ 即得到一个第三人称单数的NP

■如果读取的词为 $w_1(POS=Det, AGR=p)$ ，则不能产生产生新状态，在形成NP的过程出现了人称数的一致！



- 可以看到：基于复杂特征的规则在通过特征的取值来传递上下文相关的信息，规则已不是完全的上下文无关
- ➔增强的上下文无关语法
- 基于复杂特征的句法分析算法可以在一般方法上进行扩展(匹配时考虑特征约束)。



合一语法：进一步抽象

- 语法系统也可以看成是不同特征结构之间的约束的集合，在这样一种观点下形成的语法系统通常称为合一语法



■ 例如：

■ $S(AGR?a) \rightarrow NP(AGR?a) \ VP(AGR?a)$

■ $NP(AGR?a) \rightarrow DET(AGR?a) \ N(AGR?a)$

■ 用合一语法来表述：

■ 规则部分：

■ $X_0 \rightarrow X_1 \ X_2$

■ 约束部分：

■ $POS_0 = S \quad POS_1 = NP \quad POS_2 = VP$

■ $AGR_0 = AGR_1 = AGR_2$



合一表示的优点

- 合一表示具有更强的表达能力
- 合一表示具有更好的灵活性
- 合一表示更易于规范分析

大纲

- 引言
- PCFG
- 基于PCFG的句法分析
- LPCFG
- 合一语法
- 总结





- 针对CFG的问题展开(并行发展相应的剖析器)
 - 只能发现多个结构，但不能选择→PCFG
 - PCFG与词汇无关导致的问题→LPCFG
 - CFG描述能力弱→基于特征结构
 - 进一步精简且强大的描述→合一语法



Thank you !

