概率句法分析

哈工大信息检索研究室

2004年春

PCFG
(Probabilistic Context Free Grammars)

Chomsky hierarchy

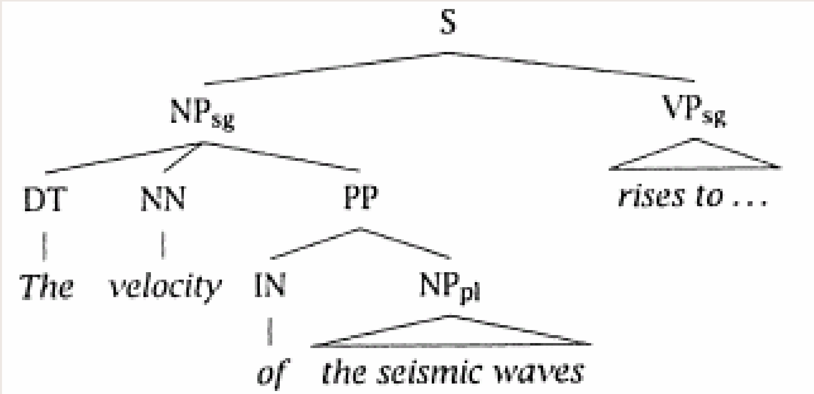
- 0-型（无约束文法）
  - 无限制
- 1-型（上下文相关文法）
  - $\alpha A \beta \rightarrow \alpha \gamma \beta$
- 2-型（上下文无关文法）
  - $A \rightarrow \gamma$
- 3-型（正规文法）
  - $A \rightarrow aB$
  - $A \rightarrow a$

Motivation

- N-gram和HMM只能处理线性序列
- 用这些方法对句子进行分析时，面临这一些问题
- The velocity of the seismic waves rises to
- 如何解决这种“矛盾”?

Motivation

- The velocity of the seismic waves rises to



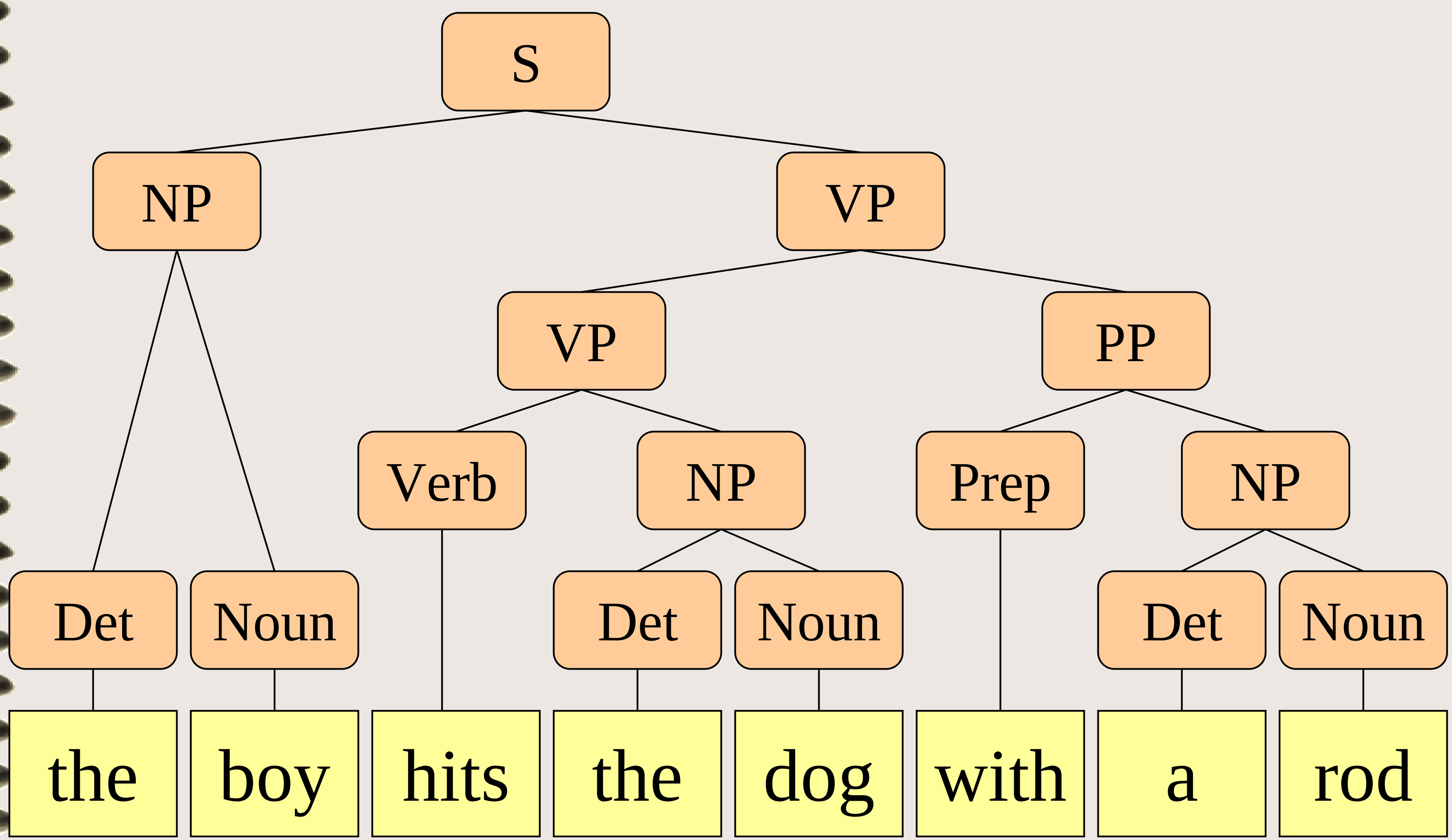
- 自然语言是一种非线性的符号序列
- 句子结构表现为复杂的嵌套性

Context Free Grammar

- (a) S→ NP, VP.
- (b) NP → Det, Noun.
- (c) VP → Verb, NP.
- (d) VP → VP, PP.
- (e) PP → Prep, NP.
- (f) Det → [the].

(g) Det→ [a].
(h) Noun → [boy].
(i) Noun → [dog].
(j) Noun → [rod].
(k) Verb → [hits].
(l) Prep → [with].

CFG



S
NP
VP
VP
PP
Verb
NP
Prep
NP
Det
Noun
Det
Noun
Det
Noun
the
boy
hits
the
dog
with
a
rod

PCFG

- 将CFG进行扩展，给每条规则一个概率值，就得到了PCFG

S → NP VP	1.0	NP → NP PP	0.4
PP → P NP	1.0	NP → *astronomers*	0.1
VP → V NP	0.7	NP → *ears*	0.18
VP → VP PP	0.3	NP → *saw*	0.04
P → *with*	1.0	NP → *stars*	0.18
V → *saw*	1.0	NP → *telescopes*	0.1

- 对于循环嵌套的树型结构，PCFG是最简单的一种概率模型

Notation

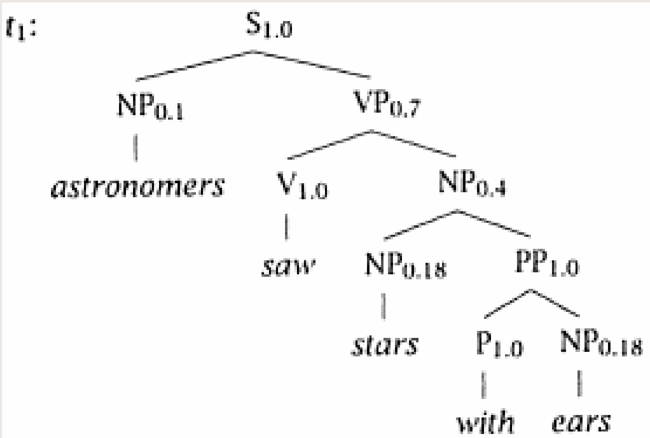
- G：语法
- L：由语法G生成的语言
- t：句法分析树
- $\{N^1, \ldots, N^n\}$：非终结点集合，N^1是开始符号
- $\{w^1, \ldots, w^V\}$：终结点集合
- $w^1 \ldots w^m$：句子序列
- N^j_{pq}：w^p到w^q的非终结点

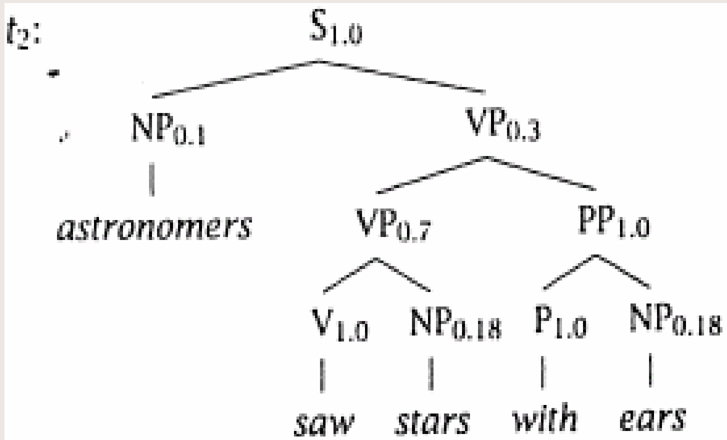
Formal Definition of a PCFG

- 终结点集合, $\{w^k\}$, $k= 1,\ldots,V$
- 非终结点集合，N^i, $i= 1,\ldots, n$
- 指定的开始符号，N^1
- 规则集$\{N^i \to \xi^j\}$, (ξ^j 是终结点和非终结点序列)
  - 并且满足$\forall i \ \Sigma_j P(N^i \to \xi^j) = 1$

A example

- astronomers saw stars with ears





$$
\begin{aligned}
P(t_1) &= 1.0 \times 0.1 \times 0.7 \times 1.0 \times 0.4 \times 0.18 \times 1.0 \times 1.0 \times 0.18 \\
&= 0.0009072 \\
P(t_2) &= 1.0 \times 0.1 \times 0.3 \times 0.7 \times 1.0 \times 0.18 \times 1.0 \times 1.0 \times 0.18 \\
&= 0.0006804 \\
P(w_{15}) &= P(t_1) + P(t_2) = 0.0015876
\end{aligned}
$$

Assumptions

- 位置无关
  - 子树的概率与构成子树所在的位置无关
  - 类似于HMM中的时间无关

$$\forall k \quad P(N^j_{k(k+c)} \to \zeta) \text{ is the same}$$

- 上下文无关
  - 子树的概率与子树以外的词无关

$$P(N^j_{kl} \to \zeta \mid \text{anything outside } k \ldots l) = P(N^j_{kl} \to \zeta)$$

- 祖先无关
  - 子树的概率与子树以外的节点无关

$$P(N^j_{kl} \to \zeta \mid \text{any ancestor nodes not in } N^j_{kl}) = P(N^j_{kl} \to \zeta)$$

计算句子及分析树的概率

- 句子的分析树T的概率
- $P(T) = \Pi_{i=1..k}\, p(r(i))$
  - $r(1), \ldots, r(k)$是CFG的规则
- 由语法G生成的句子w_{1m}的概率

$$P(w_{1m}) = \Sigma_t P(w_{1m},t) = \Sigma_{\{t:\ yield(t)=w1m\}} P(t)$$

  - t 是句子的分析树

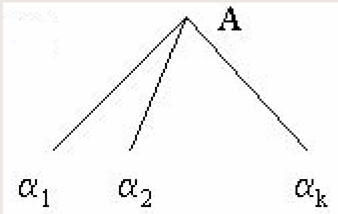
规则的概率

- 规则 r: $A \rightarrow \alpha$
- R_A： 左边为非终结点N^j 的所有规则的集合
- 则R_A 的概率分布

  $\Sigma_{r\in R}\, p(r) = 1,\ 0 \leqslant p(r) \leqslant 1$
- 从另一个角度
  - $p(\alpha \mid A)=p(r)$,
  - 其中 $r = A \rightarrow \alpha$， $\alpha \in (N \cup T)$

规则概率估计

- 根据树库应用极大似然估计(MLE)
- 规则r： $A \rightarrow \alpha_1 \alpha_2 \ldots \alpha_k$，其概率



- $p(r) = c(r) / c(A)$
  - c(r)：规则r在树库中出现的次数
  - c(A)：非终结点N^j在树库中出现的次数
  - 即$c(A)= \Sigma\gamma\ c(A \rightarrow \alpha)$

An example of PCFG

Rule	Count for LHS	Count for Rule	PROB
1. S → NP VP	300	300	1
2. VP → V	300	116	.386
3. VP → V NP	300	118	.393
4. VP → V NP PP	300	66	.22
5. NP → NP PP	1032	241	.23
6. NP → N N	1032	92	.09
7. NP → N	1032	141	.14
8. NP → ART N	1032	558	.54
9. PP → P NP	307	307	1

PCFG的三个问题

- 同HMM类似，PCFG也有三个基本问题
  - 已有语法G，计算由该语法生成的句子w_{1m}的概率$P(w_{1m}|G)$
  - 寻找句子w_{1m}的最可能的分析树$P(t|w_{1m},G)$
  - 已知句子w_{1m}，如何确定语法G，即计算规则的概率，使句子的概率最大

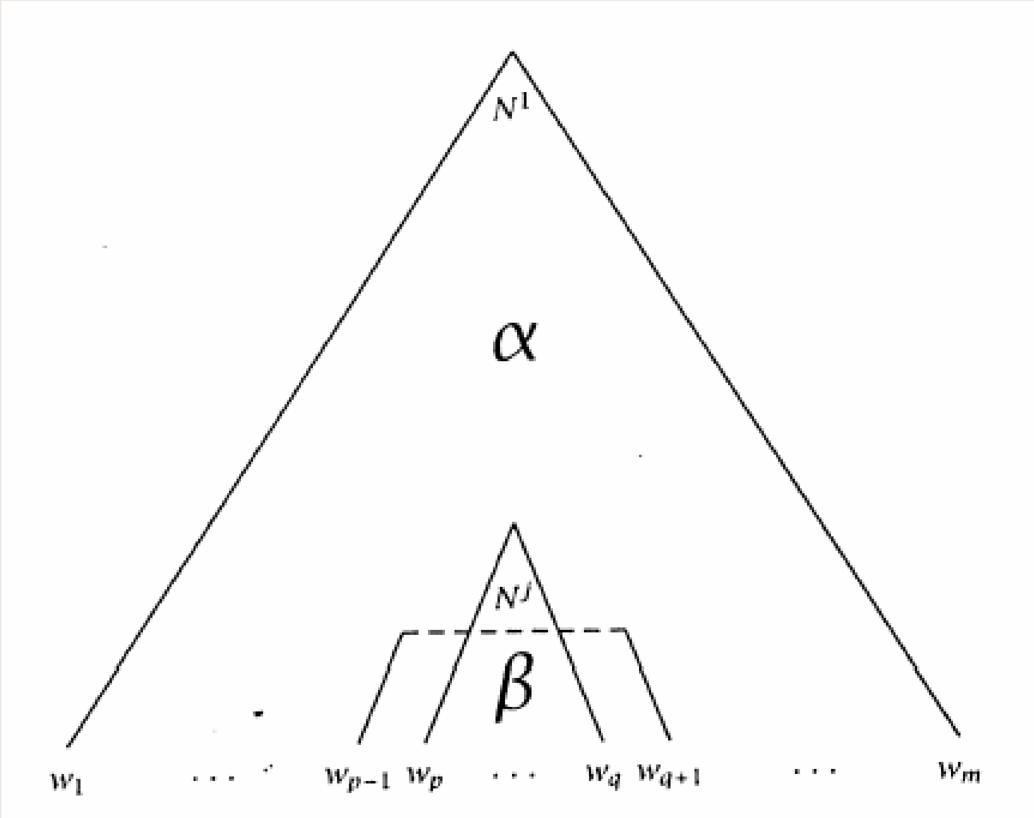
  $\operatorname{argmax}_G P(w_{1m}|G)$

HMMs和PCFGs

- HMM中
- 用前向概率(forward probability)和后向概率(backward probability)计算结点概率
  - $\alpha_i(t) = P(w_{1(t-1)}, X_t = i)$
  - $\beta_i(t) = P(w_{tT} | X_t = i)$

- PCFG中
- 前向概率对应外部概率(outside probability)
- 后向概率对应内部概率(inside probability)

Inside and Outside Probabilities



Inside and Outside Probabilities

- Inside probability $\beta_i(p,q)$是以N^j开始,生成序列w_{pq}的概率

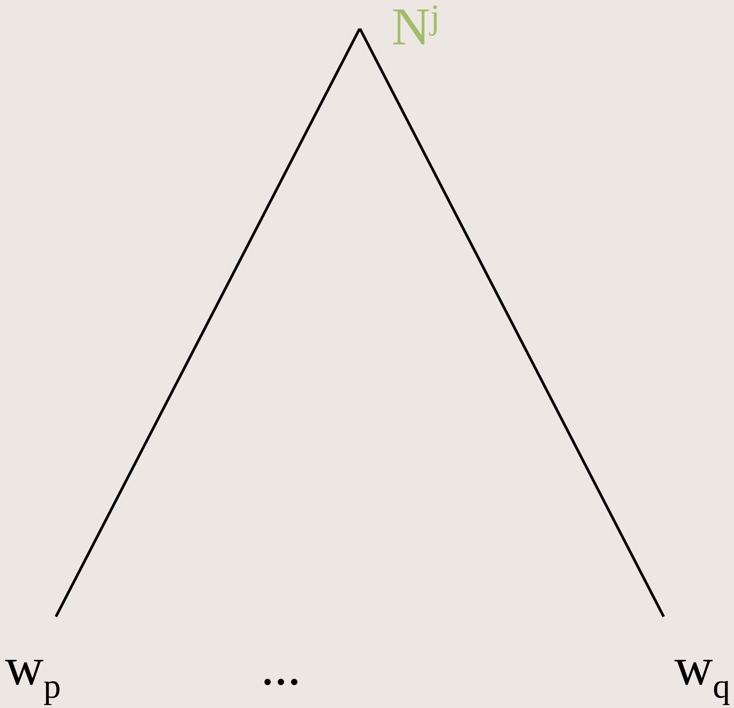
$$\beta_j(p,q) = P(w_{pq} \mid N^j_{pq}, G)$$

- Outside probability $\alpha_i(p,q)$是 以N^1开始，生成N^j及序列w_{pq}以外的所有节点的概率

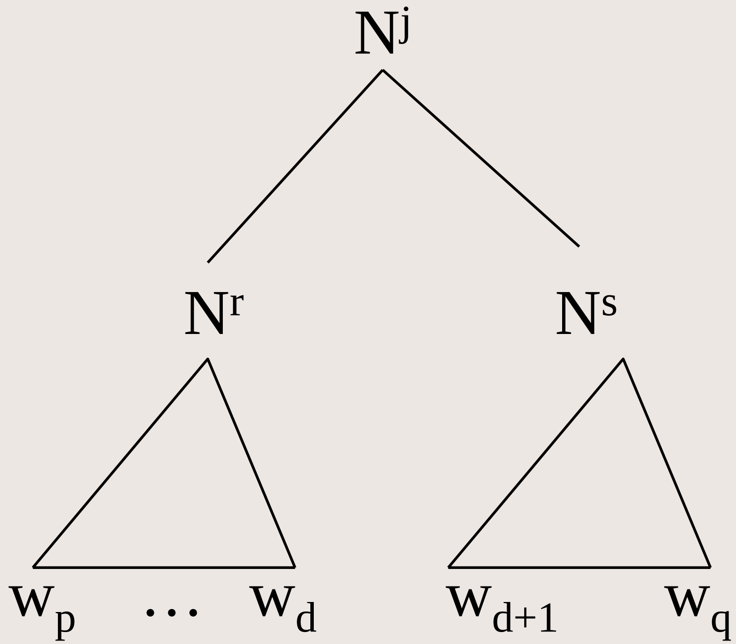
$$\alpha_j(p,q) = P(w_{1(p-1)}, N^j_{pq}, w_{(q+1)m} \mid G)$$

Inside Probability

- $\beta_j(p,q) = P(N^j \Rightarrow^* w_{pq})$



Induction



Induction

- Base Case: $\forall j$

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

- Induction

$$\begin{aligned}
\beta_j(p,q) &= P(w_{pq} \mid N_{pq}^j, G) \\
&= \sum_{r,s} \sum_{d=p}^{q-1} P(w_{pd}, N_{pd}^r, w_{(d+1)q}, N_{(d+1)q}^s \mid N_{pq}^j, G) \\
&= \sum_{r,s} \sum_{d=p}^{q-1} P(N_{pd}^r, N_{(d+1)q}^s \mid N_{pq}^j, G) \times P(w_{pd} \mid N_{pq}^j, N_{pd}^r, N_{(d+1)q}^s, G) \times \\
&\qquad P(w_{(d+1)q} \mid N_{pq}^j, N_{pd}^r, N_{(d+1)q}^s, w_{pd}, G) \\
&= \sum_{r,s} \sum_{d=p}^{q-1} P(N_{pd}^r, N_{(d+1)q}^s \mid N_{pq}^j, G) \times P(w_{pd} \mid N_{pd}^r, G) \times P(w_{(d+1)q} \mid N_{(d+1)q}^s, G) \\
&= \sum_{r,s} \sum_{d=p}^{q-1} P(N^j \to N^r N^s) \beta_r(p,d) \beta_s(d+1,q)
\end{aligned}$$

Calculation of inside Probabilities

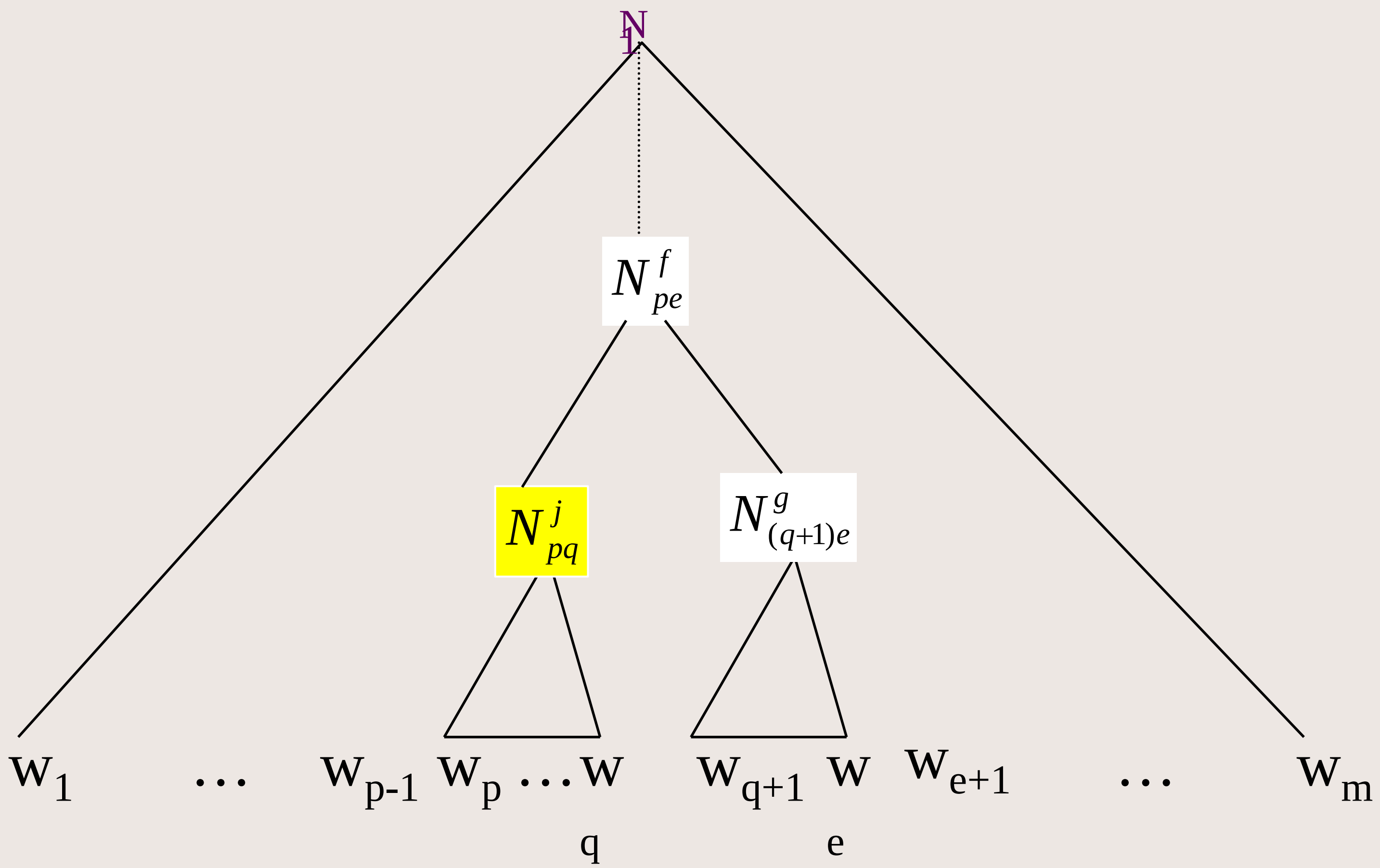
	1	2	3	4	5
1	$\beta_{NP} = 0.1$		$\beta_S = 0.0126$		$\beta_S = 0.0015876$
2		$\beta_{NP} = 0.04$ $\beta_V = 1.0$	$\beta_{VP} = 0.126$		$\beta_{VP} = 0.015876$
3			$\beta_{NP} = 0.18$		$\beta_{NP} = 0.01296$
4				$\beta_P = 1.0$	$\beta_{PP} = 0.18$
5					$\beta_{NP} = 0.18$
	astronomers	*saw*	*stars*	*with*	*ears*

计算(2,5)的值

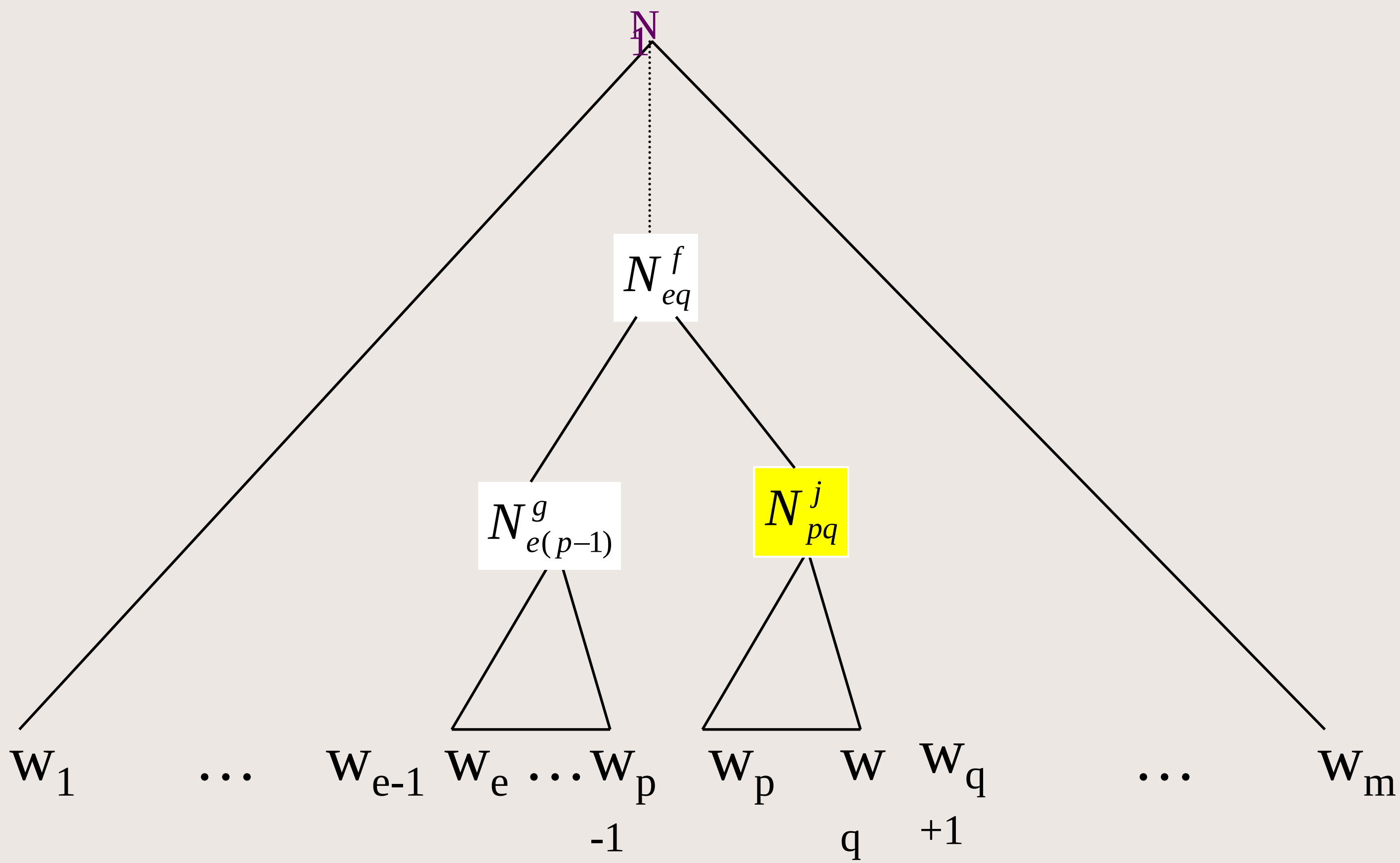
$$P(VP \rightarrow V\ NP)\ \beta_V(2,2)\ \beta_{NP}(3,5)$$

$$+\ P(VP \rightarrow VP\ PP)\ \beta_{VP}(2,3)\ \beta_{PP}(4,5)$$

Outside Probabilities



Outside Probabilities



Induction

Base Case:

$\alpha_1(1,m)= 1$; $\alpha_j(1,m)=0$ for $j\neq 1$

Induction:

$$\alpha_j(p,q) = [\sum_{f,g\neq j}\sum_{e=q+1}^{m} P(w_{1(p-1)}, w_{(q+1)m}, N^f_{pe}, N^j_{pq}, N^g_{(q+1)e}) +$$

$$\sum_{f,g}\sum_{e=1}^{p-1} P(w_{1(p-1)}, w_{(q+1)m}, N^f_{eq}, N^g_{e(p-1)}, N^j_{pq})]$$

$$= [\sum_{f,g\neq j}\sum_{e=q+1}^{m} P(w_{1(p-1)}, w_{(e+1)m}, N^f_{pe}) \times P(N^j_{pq}, N^g_{(q+1)e} \mid N^f_{pe}) \times P(w_{(q+1)e} \mid N^g_{(q+1)e}) +$$

$$\sum_{f,g}\sum_{e=1}^{p-1} P(w_{1(e-1)}, w_{(q+1)m}, N^f_{eq}) \times P(N^g_{e(p-1)}, N^j_{pq} \mid N^f_{eq}) \times P(w_{e(p-1)} \mid N^g_{e(p-1)})]$$

$$= [\sum_{f,g\neq j}\sum_{e=q+1}^{m} \alpha_f(p,e) P(N^f \to N^j N^g) \beta_g(q+1,e) +$$

$$\sum_{f,g}\sum_{e=1}^{p-1} \alpha_f(e,q) P(N^f \to N^g N^j) \beta_g(e,p-1)]$$

计算序列的概率

- Inside Algorithm

  用Inside Probability计算整个序列的概率

$$P(w_{1m} \mid G) = P(N^1 \overset{*}{\Rightarrow} w_{1m} \mid G)$$
$$= P(w_{1m} \mid N^1_{1m}, G) = \beta_1(1, m)$$

- Outside Algorithm

  用Outside Probability计算整个序列的概率

$$P(w_{1m} \mid G) = \sum_j P(w_{1(k-1)}, w_k, w_{(k+1)m}, N^j_{kk} \mid G)$$
$$= \sum_j \alpha_j(k, k) P(N^j \rightarrow w_k)$$

- Inside-Outside Algorithm

$P(w_{1m}, N_{pq}|G) = \Sigma_j\ \alpha_j(p,q)\ \beta_j(p,q)$

最优路径

- A Viterbi-style algorithm
- HMM中，用$\delta_i(t)$记录时刻t通过状态j的路径的最大值
- PCFG中，用$\delta_i(p,q)$记录子树N_{pq}^{j} 的最大概率
- $\psi_i(p,q) = (j, k, r)$记录当前的最佳路径

最优路径

- 1. 初始化

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

- 2. 循环
- 计算子树最大概率
- $\delta_i(p,q) = \max_{1\leq j,k\leq n,p\leq r<q} P(N^i \rightarrow N^j\ N^k)\ \delta_j(p,r)\ \delta_k(r+1,q)$
- 记录路径
- $\psi_i(p,q) = \mathrm{argmax}_{(j,k,r)} P(N^i \rightarrow N^j\ N^k)\ \delta_j(p,r)\ \delta_k(r+1,q)$
- 3. 终止回溯

  $P(t') = \delta_1(1,m)$

Training a PCFG

- 目标
  - 对已有的语法规则，赋给其最佳的概率，推导出真实合理的语法
- 限制
  - 预先提供语法规则，包括终结点，非终结点，开始节点，并给每条规则赋予一个初始概率值（可以是随机概率）
- 方法
  - 在未标注的语料上，应用EM的方法训练，也叫Inside-Outside Algorithm
- 基本假设
  - 符合真实语法的句子更可能在训练语料中出现

EM

- 计算规则的概率

$$\hat{P}(N^j \to \zeta) = \frac{C(N^j \to \zeta)}{\sum_\gamma C(N^j \to \gamma)}$$

- 根据初始语法分析句子，在所有的分析结果中，统计每个规则出现的次数，然后计算出规则的期望值

$$P(N^j \to N^r N^s) = \frac{E(N^j \to N^r N^s, N^j \text{ used})}{E(N^j \text{ used})}$$

Problems with the Inside-Outside Algorithm

- 速度非常慢
  - $O(m^3n^3)$, m 是句子的长度，n 是非终结点的数量
  - 需多次迭代
- 容易陷入局部最大值
- 学习的规则同语言学知识可能相距甚远
- 从未标注的语料中的学习的语法并不理想

句法分析

Why to parse?

- 自然语言处理
- 分析句子的语法结构
- 解决语言中的句法歧义
  - saw (a cat with a telescope)
  - (saw a cat) with a telescope
- PCFG中的第二、三个问题
  - 已知句子w_{1m}，如何确定语法G，即计算规则的概率，使句子的概率最大 $\boldsymbol{argmax_G\ P(w_{1m}|G)}$
  - 寻找句子w_{1m}的最可能的分析树$P(t|\ w_{1m},G)$

How to parse?

- 制定句法分析的规范
- 规范是指导性的，对象是人
- 人能够根据语义等知识标注出符合规范的句子
  - (saw a cat) with a telescope
- 机器不具备人的能力，要让机器代替人分析
  - 人先教给机器足够的知识
  - 利用所学知识自动标注新的句子

Rules or Statistics

- 传授知识的方法
  - 基于规则的方法
    - 制定出面向机器的若干规则
  - 基于统计的方法
    - 人工标注语料，机器从语料库中自动获取知识
- 统计方法表现得更有优势
  - 大规模的标注语料（Treebanks）
  - 机器学习算法
- 获取知识的过程也是语法推导的过程
  - grammar induction
  - 每种语言对应一种语法

Treebanks

- 利用树库生成语法，树库越大越好
- 最常用的树库—Penn Treebank
- Penn Treebank的一些特点：
  - Flat structure for NPs
    - Arizona real estate loans
  - 一些语法、 语义功能标记
    - -SBJ, -LOC, etc.
  - Empty nodes for gaps (“understood” subject)
    - Marked as *

A Penn Treebank tree

```
( (S (NP-SBJ The move)
     (VP followed
         (NP (NP a round)
             (PP of
                 (NP (NP similar increases)
                     (PP by
                         (NP other lenders))
                     (PP against
                         (NP Arizona real estate loans)))))
         ,
         (S-ADV (NP-SBJ *)
                (VP reflecting
                    (NP (NP a continuing decline)
                        (PP-LOC in
                                (NP that market))))))
     .))
```

Penn Treebank POS Tag

CC	Coordinating conj.	TO	infinitival *to*
CD	Cardinal number	UH	Interjection
DT	Determiner	VB	Verb, base form
EX	Existential there	VBD	Verb, past tense
FW	Foreign word	VBG	Verb, gerund/present pple
IN	Preposition	VBN	Verb, past participle
JJ	Adjective	VBP	Verb, non-3rd ps. sg. present
JJR	Adjective, comparative	VBZ	Verb, 3rd ps. sg. present
JJS	Adjective, superlative	WDT	Wh-determiner
LS	List item marker	WP	Wh-pronoun
MD	Modal	WP$	Possessive *wh*-pronoun
NN	Noun, singular or mass	WRB	Wh-adverb
NNS	Noun, plural	#	Pound sign
NNP	Proper noun, singular	$	Dollar sign
NNPS	Proper noun, plural	.	Sentence-final punctuation
PDT	Predeterminer	,	Comma
POS	Possessive ending	:	Colon, semi-colon
PRP	Personal pronoun	(	Left bracket character
PP$	Possessive pronoun	)	Right bracket character
RB	Adverb	"	Straight double quote
RBR	Adverb, comparative	‘	Left open single quote
RBS	Adverb, superlative	“	Left open double quote
RP	Particle	’	Right close single quote
SYM	Symbol	”	Right close double quote

Penn Treebank Phrasal Categories

- S: simple clause (sentence)
- SBAR: S' clause with complementizer
- SBARQ: Wh S' clause
- SQ: Yes/No question
- SINV: Declarative inverted sentence
- RRC: reduced relative clause
- ADJP: adjective phrase
- ADVP: adverbial phrase
- NP: noun phrase
- PP: prepositional phrase
- QP: quantifier phrase (in NP)
- VP: verb phrase
- CONJP: multiword conjunction phrases
- WHNP: WH noun phrase
- WHPP: WH prepositional phrase
- WHADJP
- WHADVP
- UCP: unlike coordinated conjunction
- PRT: particle
- FRAG: fragment
- INTJ: interjection
- LST: list marker
- X: who knows?
- NAC: not a constituent grouping

Functional Tags

Text Categories	
-HLN	headlines and datelines
-LST	list markers
-TTL	titles
Grammatical Functions	
-CLF	true clefts
-NOM	non NPs that function as NPs
-ADV	clausal and NP adverbials
-LGS	logical subjects in passives
-PRD	non VP predicates
-SBJ	surface subject
-TPC	topicalized and fronted constituents
-CLR	closely related - see text
Semantic Roles	
-VOC	vocatives
-DIR	direction & trajectory
-LOC	location
-MNR	manner
-PRP	purpose and reason
-TMP	temporal phrases

Parsing Models

- 句法分析的任务是：输入一个句子，输出一个符合给定语法的最可能分析结果
- 数学意义为一个概率评价函数，评价某个句法分析结果（通常表示为语法树形式）是当前句子的正确语法解释的概率
- 对于任意分析树 $t \in T$,统计句法分析模型能够计算出 t 的概率 $p(t \mid S, G)$
- 同时满足： $\sum_t p(t \mid S, G) = 1$
- 最佳分析结果: *t'= arg max t P(t|S,G)*
  - G为语法
  - *S*为句子
  - *T*为根据语法G对*S*的全部分析结果所组成的集合

Building Models

- 上下文无关文法的独立假设过于严格
- 上下文是很重要的信息，人在分析句子的时候借助于上下文信息
  - 上下文的信息很多，结构信息更容易应用
- 词汇包含着更为丰富的信息，只有用词汇信息才可能解决一些语法歧义
  - 词汇信息太多，可以部分考虑词汇，比如考虑头结点的词汇

Structural Context

- 主语和宾语位置的名词短语有着不同的概率分布
  - NP→PRP subj: 13.7% obj: 2.1%
  - NP→NNP subj: 3.5% obj: 0.9%
  - NP→DT NN subj: 5.6% obj: 4.6%
  - NP→NN subj: 1.4% obj: 2.4%
  - NP→NP SBAR subj: 0.5% obj: 2.6%
  - NP→NP PP subj: 5.6% obj: 14.1%

Structural Context

- 做第一个宾语和第二个宾语的概率也不同

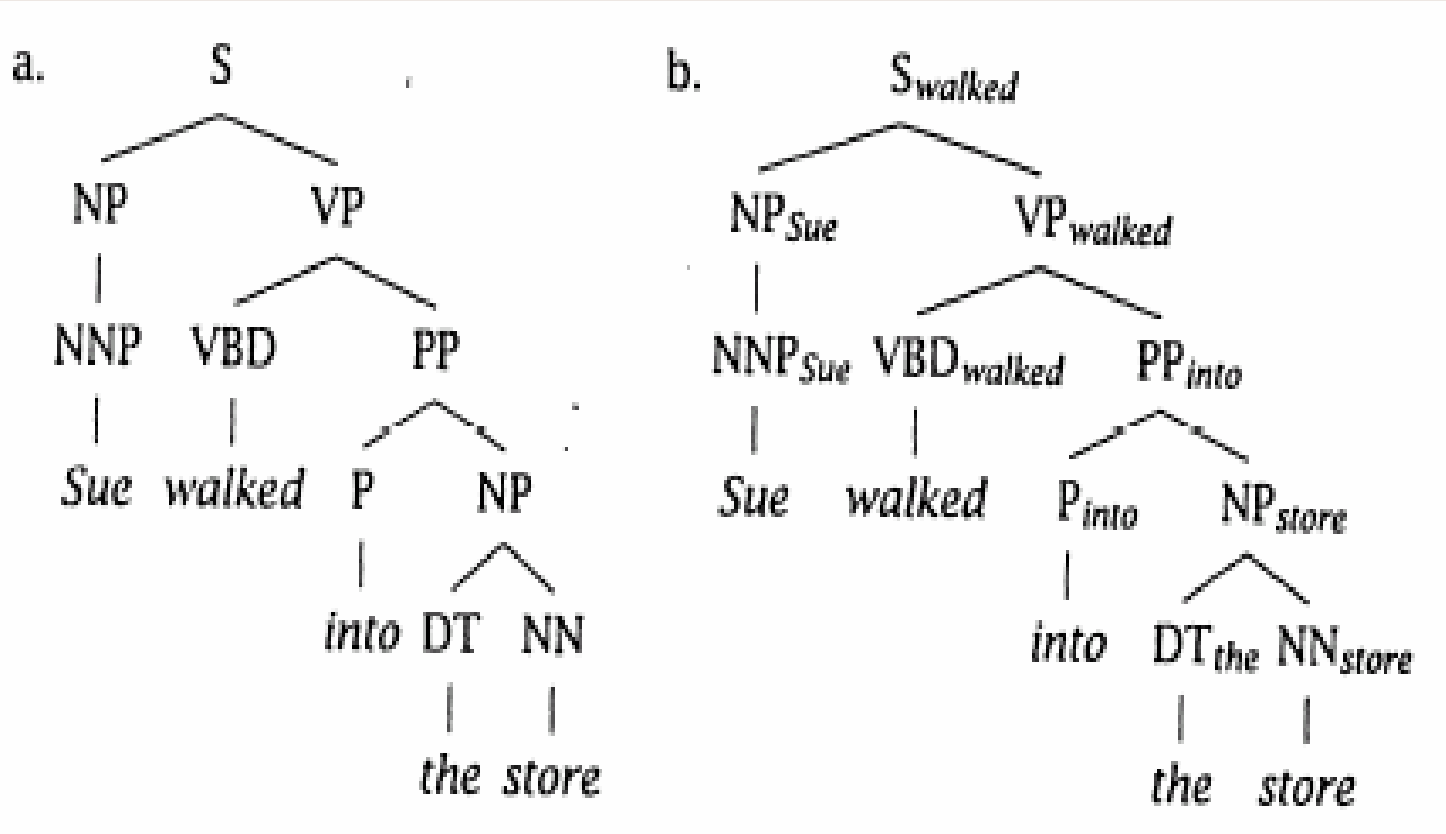
Expansion	% as 1st Obj	% as 2nd Obj
NP → NNS	7.5%	0.2%
NP → PRP	13.4%	0.9%
NP → NP PP	12.2%	14.4%
NP → DT NN	10.4%	13.3%
NP → NNP	4.5%	5.9%
NP → NN	3.9%	9.2%
NP → JJ NN	1.1%	10.4%
NP → NP SBAR	0.3%	5.1%

Lexicalization

- PCFG中，规则的左部与词无关
- 对Penn Treebank中扩展VP的一些统计

Local tree	*come*	*take*	*think*	*want*
VP → V	9.5%	2.6%	4.6%	5.7%
VP → V NP	1.1%	32.1%	0.2%	13.9%
VP → V PP	34.5%	3.1%	7.1%	0.3%
VP → V SBAR	6.6%	0.3%	73.0%	0.2%
VP → V S	2.2%	1.3%	4.8%	70.8%
VP → V NP S	0.1%	5.7%	0.0%	0.3%
VP → V PRT NP	0.3%	5.8%	0.0%	0.0%
VP → V PRT PP	6.1%	1.5%	0.2%	0.0%

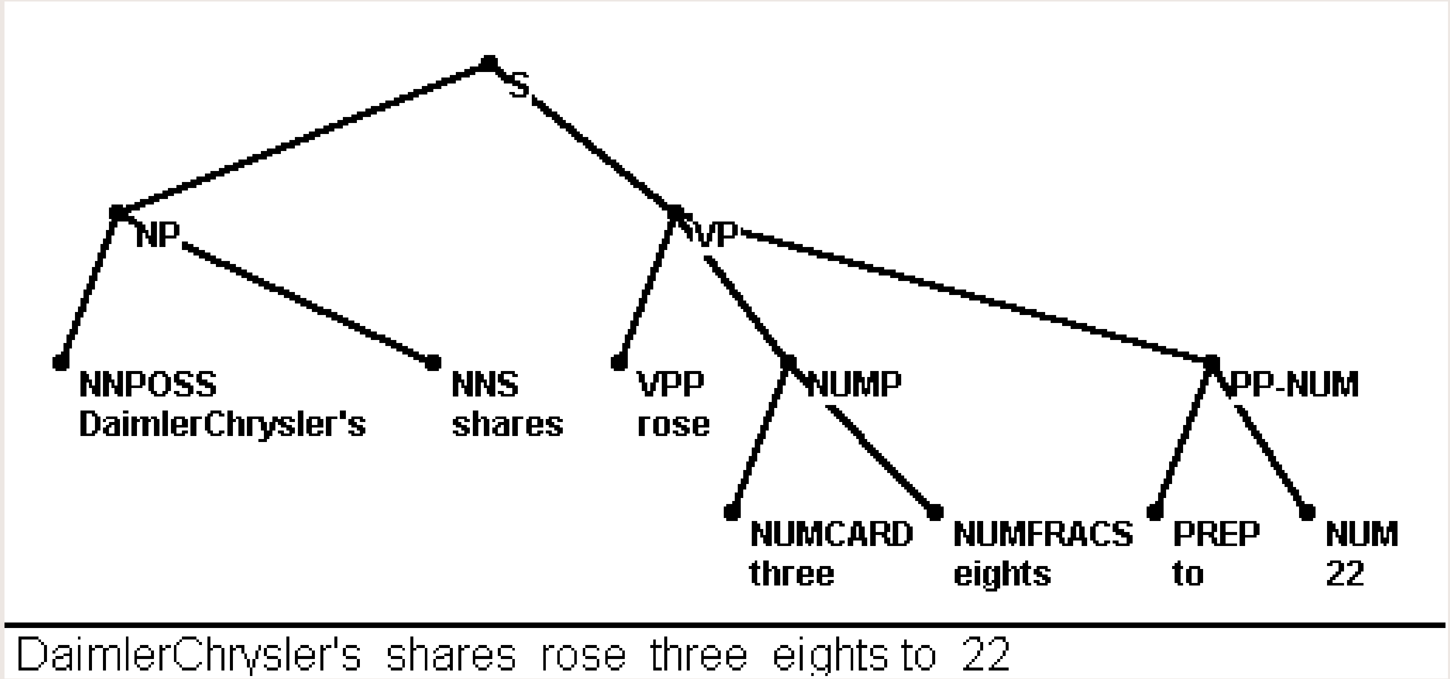
Lexicalized Tree Example



Lexicalized Tree Example

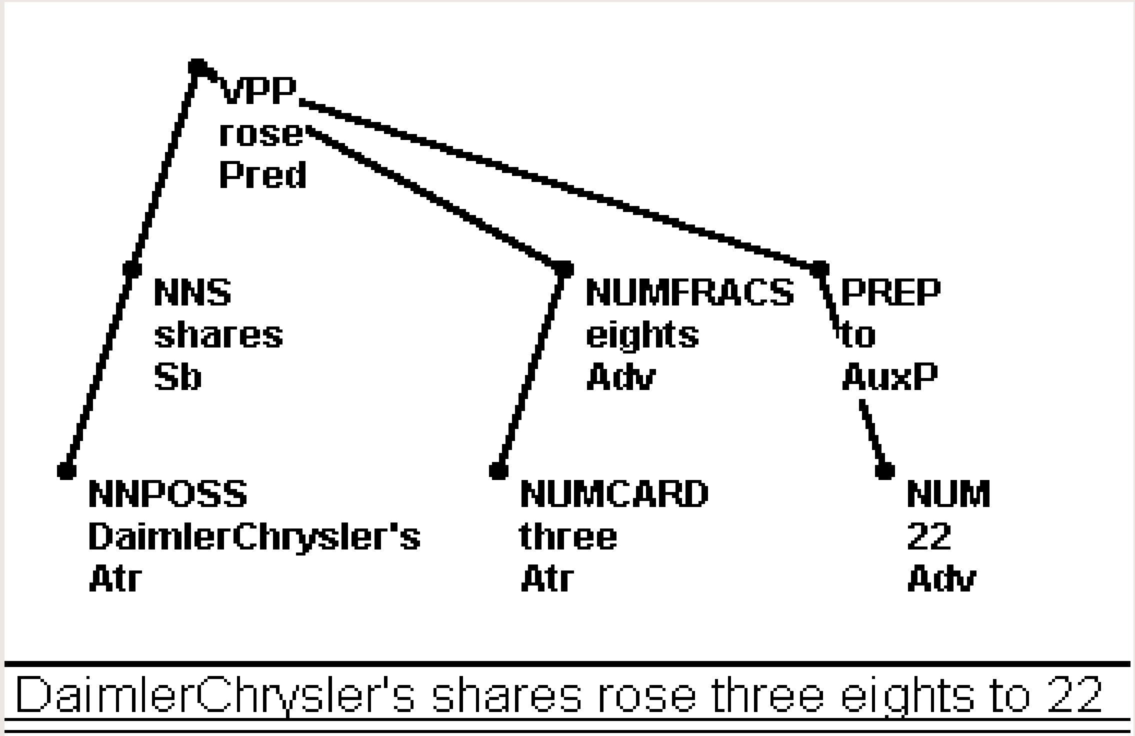
```
(S{represents} (NP-SBJ{Blair}(NNP{John} John) (NNP{Blair} Blair))
               (VP{represents} (VBZ{represents} represents)
                        (NP {stations}
                             (NP{stations} (QP{130} (RB{about} about)
                                                     (CD{130} 130))
                                            (JJ{local} local)
                                            (NN{television} television)
                                            (NNS{stations} stations))
                             (PP-LOC{in} (IN{in} in)
                                          (NP{placement}
                                              (NP{placement} (DT{the} the)
                                                   (NN{placement} placement))
                                              (PP{of} (IN{of} of)
                                                 (NP{advertising}
                                                     (ADJP{other}
                                                         (JJ{national} national)
                                                         (CC{and} and)
                                                         (JJ{other} other))
                                                     (NN{advertising}
                                                         advertising))))))
              (. .))
```

Phrase Structure Grammars



(S (NP DaimlerChrysler's shares)
(VP rose (NUMP three eighths) (PP-NUM to 22)))

Dependency Tree



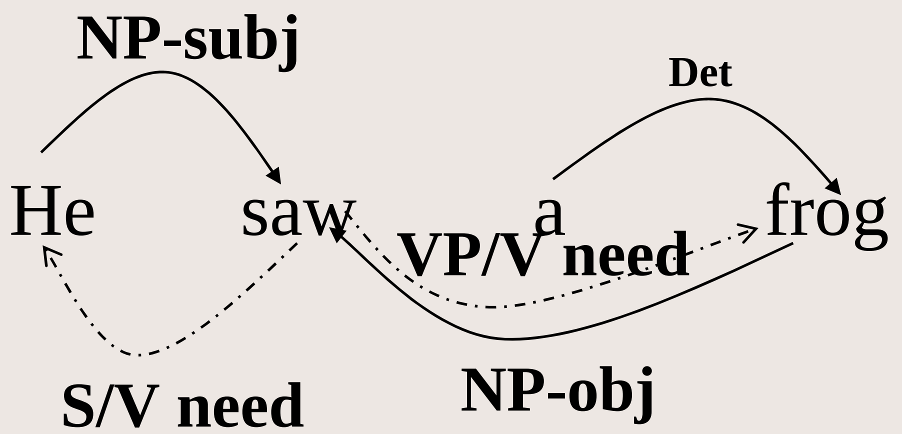
$\text{rose}_{\text{Pred}}(\text{shares}_{\text{Sb}}(\text{DaimlerChrysler's}_{\text{Atr}}),\text{eights}_{\text{Adv}}(\text{three}_{\text{Atr}}),\text{to}_{\text{Aux}}\text{P}(22_{\text{Adv}}))$

The Mapping: Incomplete?

(S (NP (Pronoun He)

(VP (V saw)

(NP (Det a) (Noun frog))))



Comparison

共同点：

- 两种语法本质上是一致的(isomorphism)
- 依存文法等价于词汇化的短语结构文法
- 短语结构文法等价于无向弧的依存文法

Comparison

- 不同点：
  - 短语结构
    - 生成结构时不考虑结构的意义
    - 要词汇化需在规则中增加词汇
    - Flat structure 引起数据稀疏
    - CFG
  - 依存
    - 对结构的意义有影响
    - 依存本身就是词汇化的结构
    - 依存结构不易稀疏，有助于解决flat structure 问题
    - Beyond CFG is possible

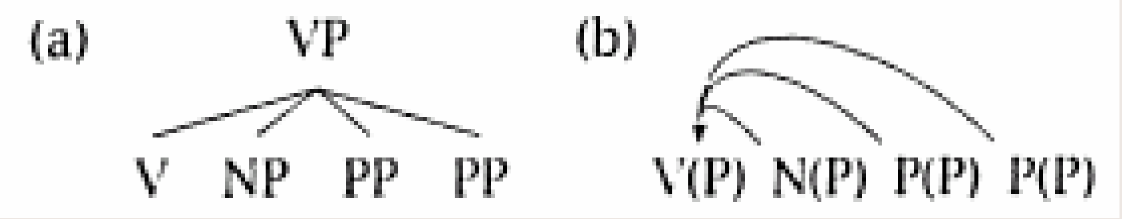
Two Advantages(1)

- 依存文法表示为两个词之间的依存关系，歧义结构直接在词汇依存关系中解决，不必考虑短语结构中的词汇化问题

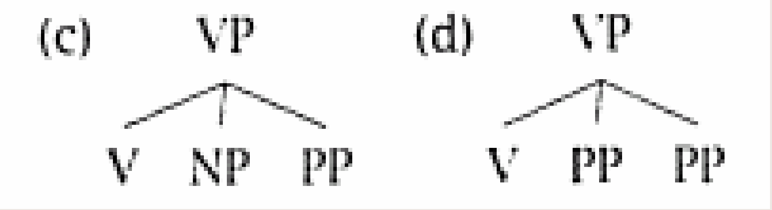
- 没有短语结构中的上层结构superstructure)，短语结构中许多这样的结构是多余的

Two Advantages(2)

- 短语结构的flat structure易引起数据稀疏



- 依存文法可以将这种结构分解



- 实际上是做了进一步的独立假设

Parser Evaluation

- 语言模型评价
  - 用held out数据计算模型交叉熵(cross entropy)
  - 更适合数据预测
  - 弱等价语法有相同的交叉熵
- 基于任务的评价(task-based)
  - 将Parser嵌入到任务中，比较parser对该任务的影响
  - 评价方法最为合理

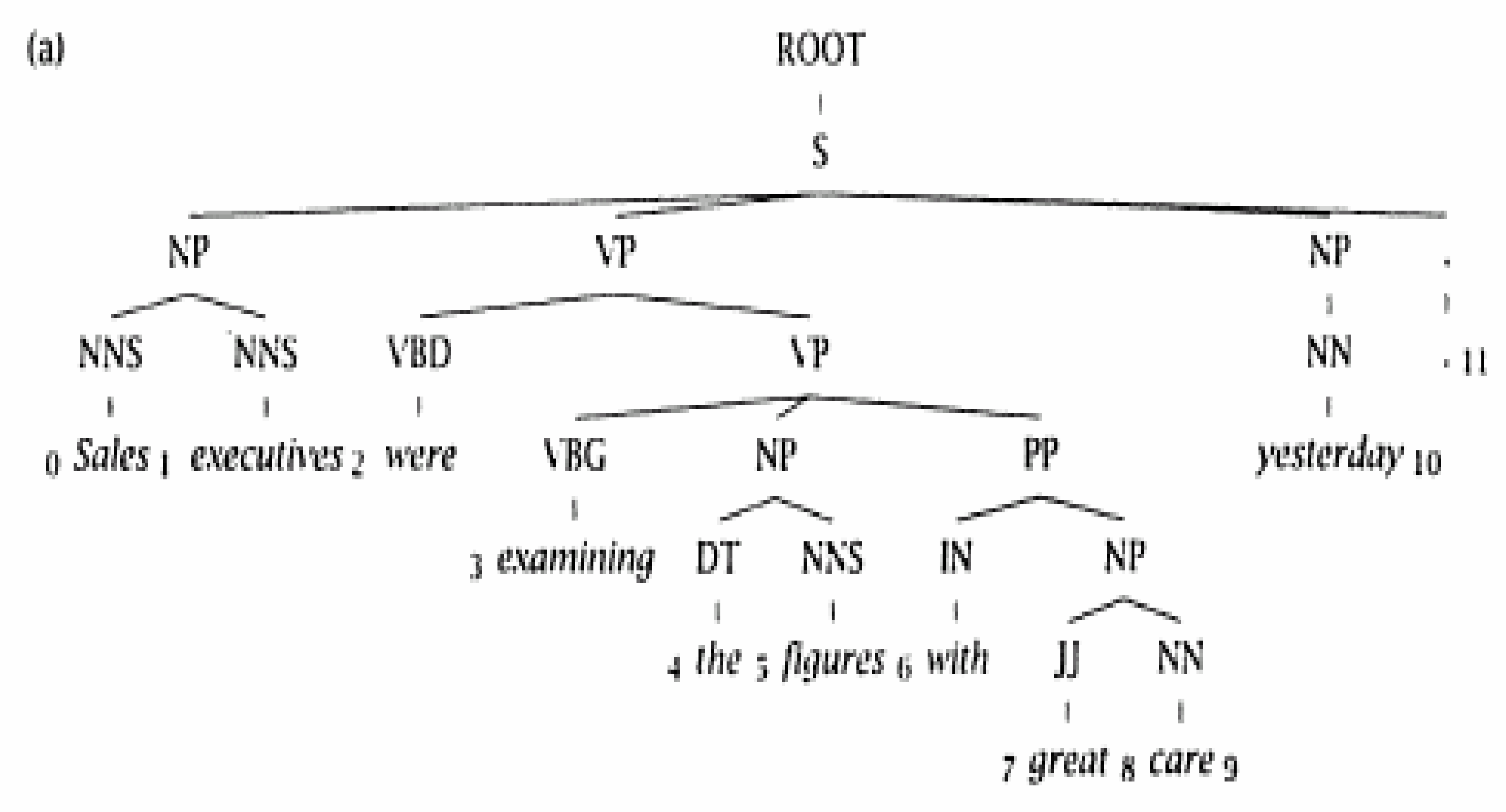
Parser Evaluation

- 简化评价过程
- 评价过程模块化
- 目标原则(objective criterion)
  - 最严格的目标原则：
    - 整个树分析正确 1，有任何错误 0
    - 也叫树准确率(tree accuracy)或完全匹配原则(exact match criterion)，仍然是一个评价标准
  - 合理的目标原则：
    - 部分正确的分析对实际应有有价值

PARSEVAL

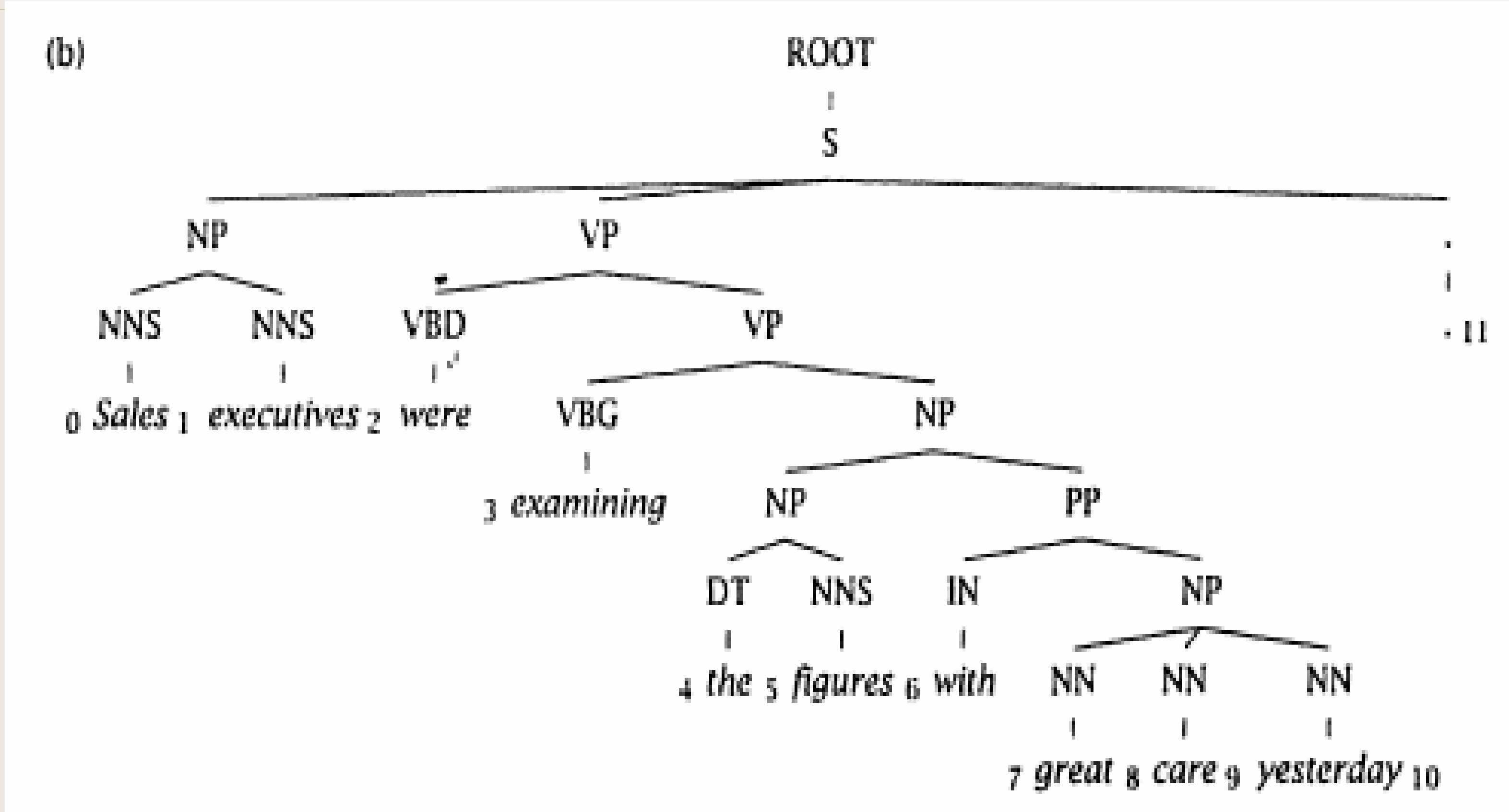
- 短语结构的评价标准
- 比较分析结果的成分
  - Labeled Precision:
    - 分析结果中的括号及标记/标准结果中的括号及标记
  - Labeled Recall:
    - 分析结果中正确的括号及标记/分析结果中所有的括号及标记
  - Crossing brackets:
    - 分析结果中每个句子的的成分边界同标准结果中的成分边界发生交叉的数量平均值
  - Non-crossing Accuracy: non-crossing brackets的百分比
  - F-measure: F = 1 / (α/P + (1-α)/R); **for α=.5, F = 2PR/(R+P)**

An Example of the PARSEVAL Measures



- 标准结果
- S-(0:11),NP-(0:2), VP-(2:9), VP-(3:9),NP-(4:6),PP-(6,9),NP-(7,9),NP-(9,10)

An Example of the PARSEVAL Measures



- 分析结果
- S-(0:11),NP-(0:2), VP-(2:10), VP-(3:10),NP-(4:10),NP-(4:6),PP-(6,10),NP-(7,10)

An Example of the PARSEVAL Measures

- Labeled Precision: 3/8 = 37.5%
- Labeled Recall: 3/8 = 37.5%
- Crossing Brackets: 0
- Crossing Accuracy: 100%
- Tagging Accuracy: 10/11 = 90.9%

Dependency Parser Evaluation

- 依存召回率R_D
  - R_D= Correct(D) / |S|
    - Correct(D): 分析结果中正确的依存关系总数
    - |S|： 标准结果中的词数(|dependencies| = |words|)
- 依存准确率P_D（不完全输出）
  - P_D = Correct(D) / Generated(D)
    - Generated(D)：分析结果中的依存关系总数
  - 有些词没有父结点
  - 有些词有多个父结点

Parsing as a Search

- 一些语法能够应用Viterbi这样的多项式时间搜索算法
  - 通过一个表（chart）记录中间派生过程

- 更多的复杂一些的语法不支持这种功能，无法应用Viterbi算法

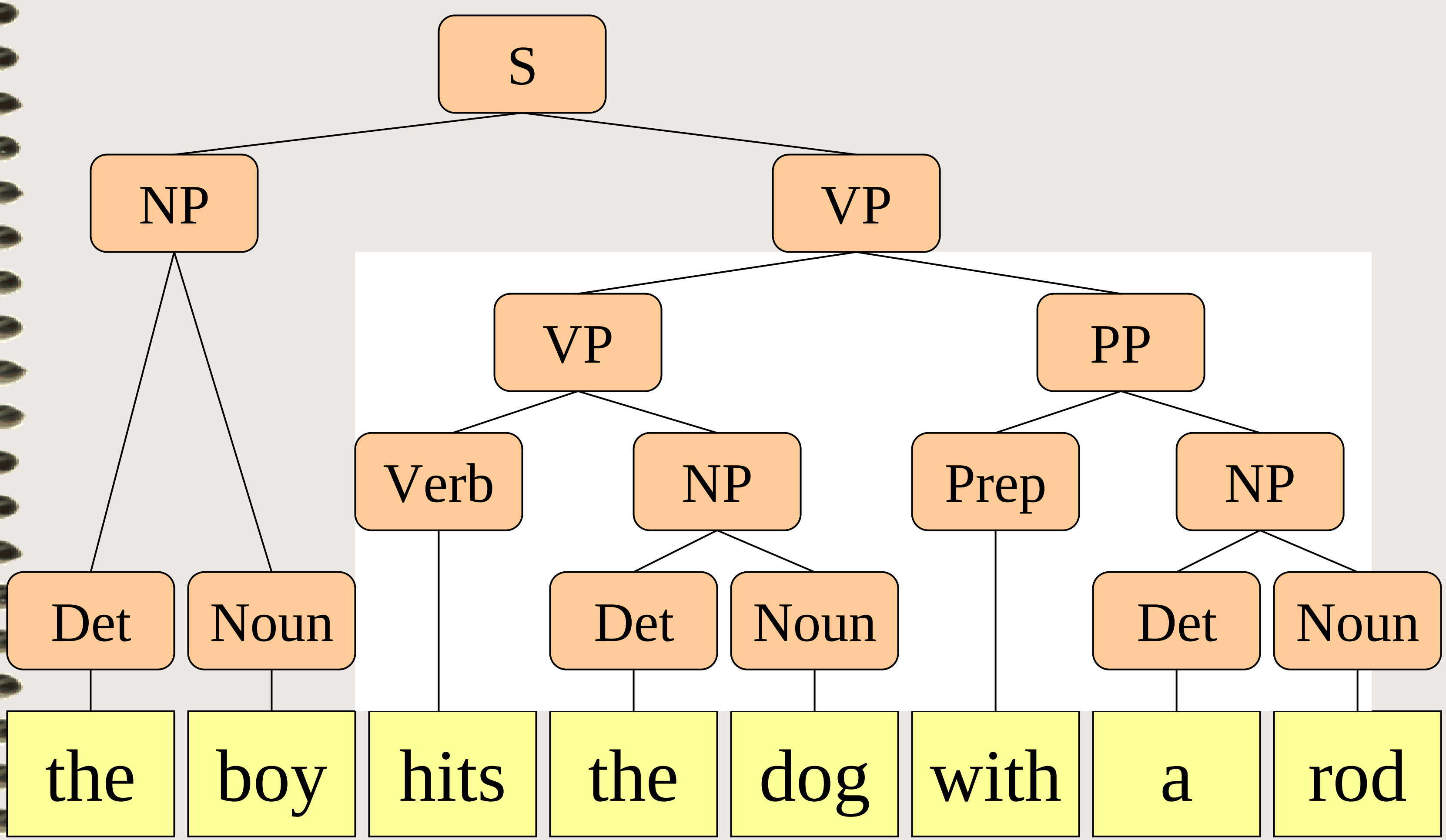
Search Methods

- Stack Decoding
  - 相同代价搜索算法(uniform-cost search)
  - 从开始结点开始，顺着代价等高点向外扩展
  - 可靠的算法
- Beam Search
  - 限制结点的扩展数量
  - 不可靠算法
- A*
  - 根据启发式策略扩展最可能的节点
  - 小于当前路径的节点不进行扩展
  - 可靠的算法

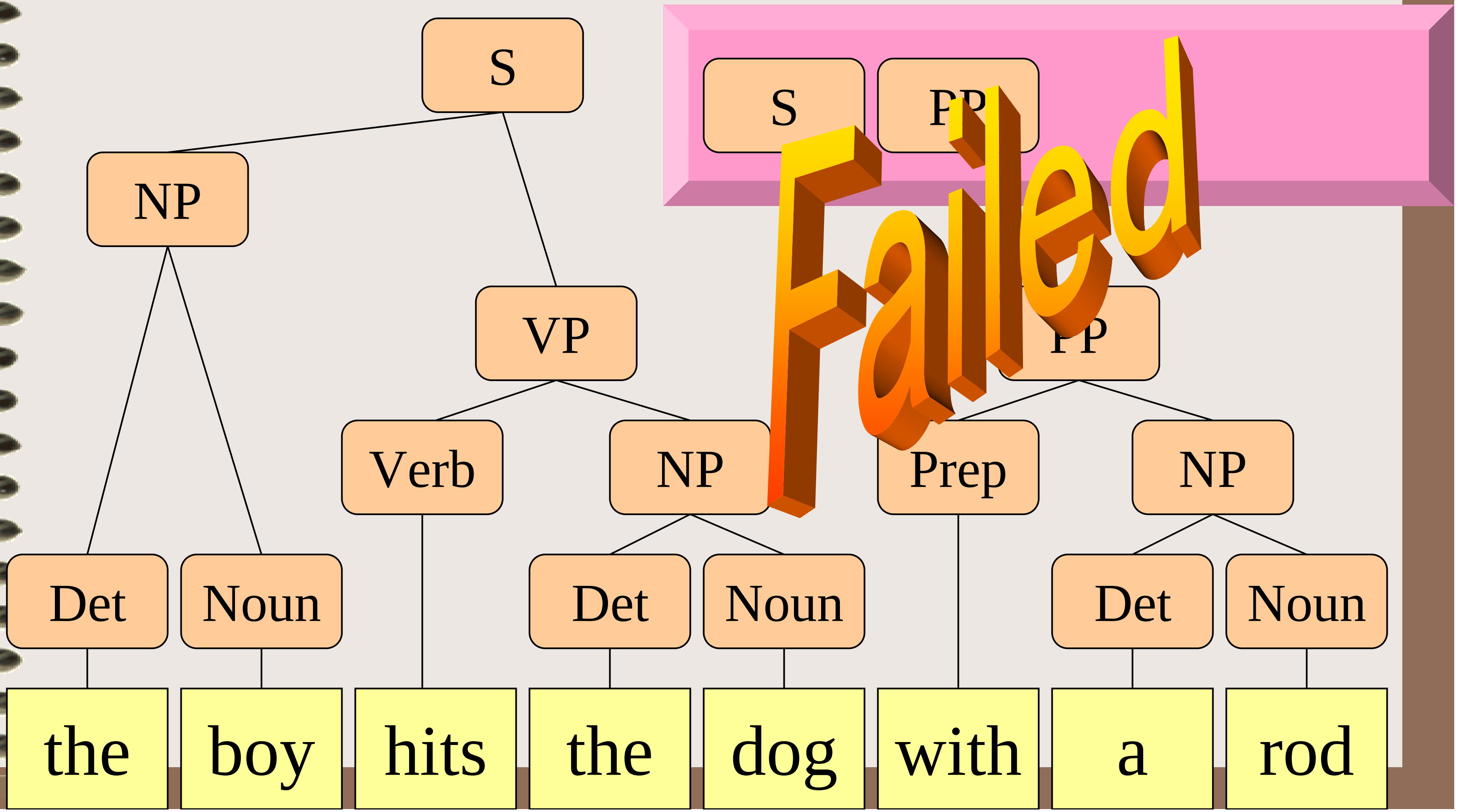
Search Methods

- Top-down Parsing
  - 自顶向下
- Bottom-up Parsing
  - 自底向上
- Left-corner Parsing
  - 自顶向下结合自底向上

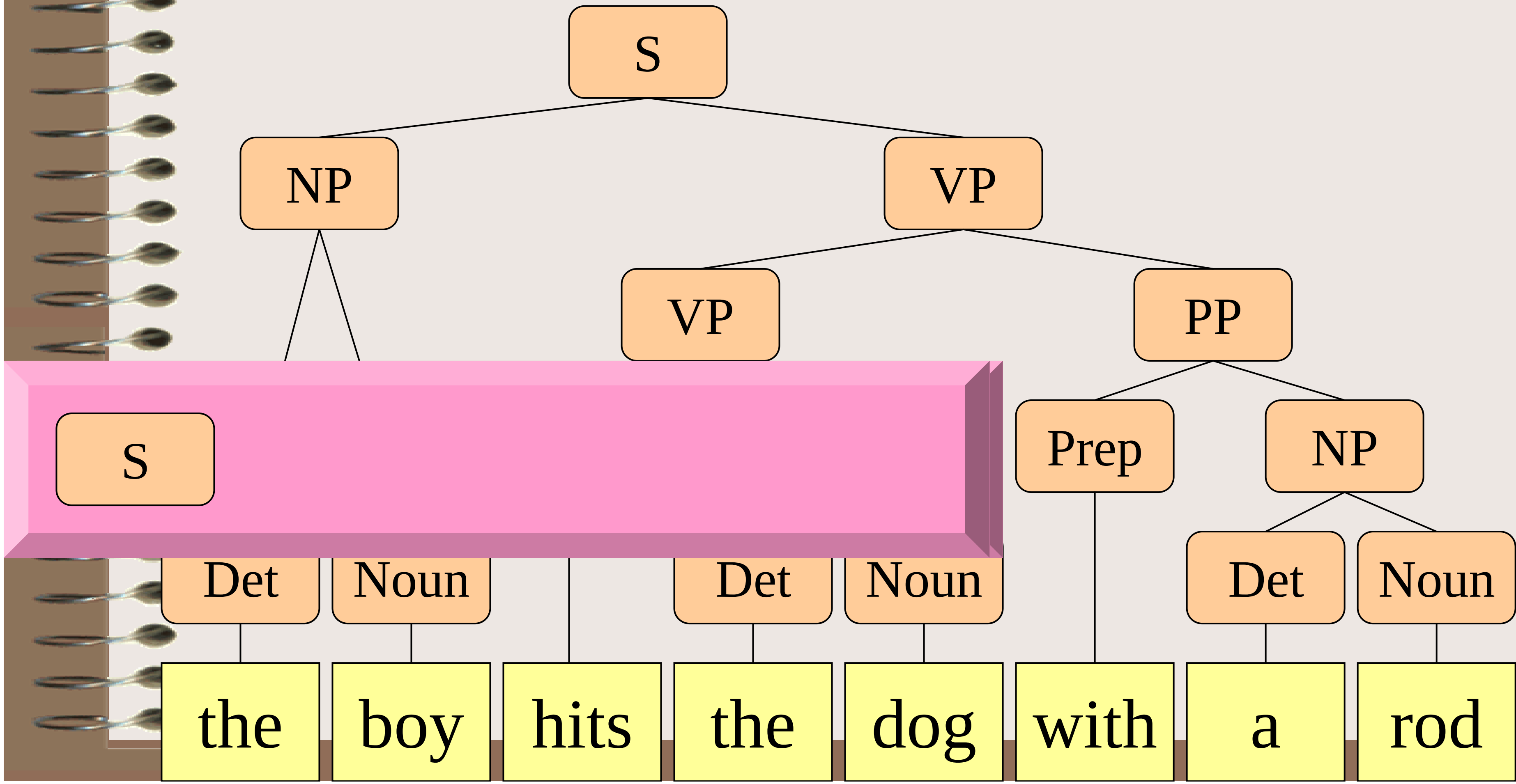
Top-down Parsing



Bottom-up Parsing



Bottom-up Parsing



Left-Corner Parsing

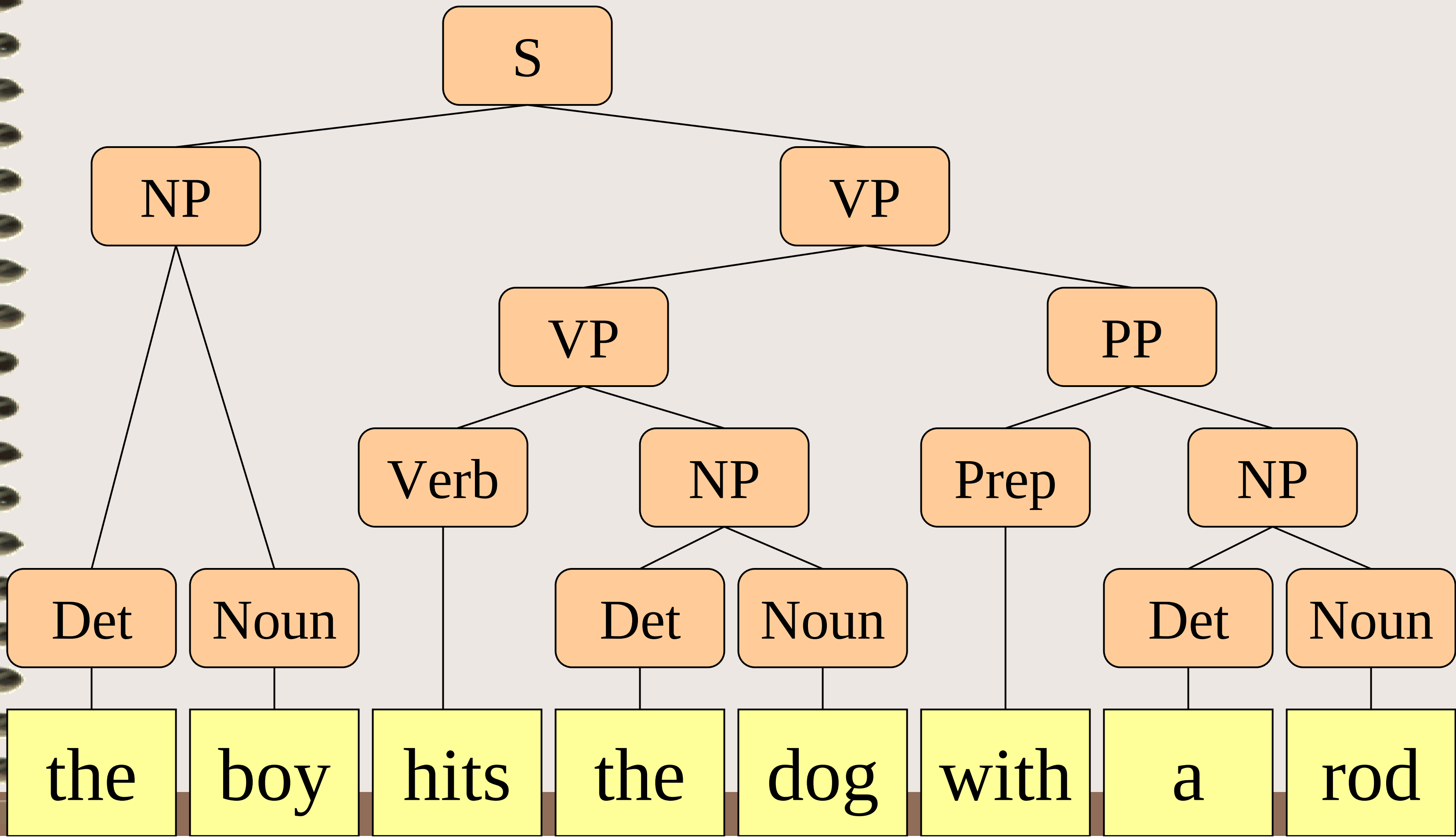
S

with a rod

Left-Corner Parsing(cont'd)

S

Parsed Sentence



Some Approaches

- Non-Lexicalized Treebank Grammars
  - Charniak (1996)
  - 输入时一系列词类标记(word categories)，没有使用词汇信息
  - 参数少，不必考虑数据稀疏
- History-Based Grammars (HBGs)
  - Black(1992)
  - 应用语法派生过程的历史信息
  - 用决策树学则历史信息的特征(Magerman, 1994)

Some Approaches

- Dependency Model Approach
  - Collins (1996)
  - 基本名词短语(baseNP)作为一个单元
  - 句子看成是有baseNP和其他词(表示为B)，以及它们之间的依存关系(表示为D)构成
  - $P(t \mid s) = P(B,D \mid s) = P(B \mid s) \times P(D \mid s, B)$

[He] saw [a frog]

- Head-Driven
  - VP(saw) –> V NP

Comparison of Systems

System	% LR	% LP	CB	% 0 CB
Charniak (1996) PCFG	80.4	78.8	NA	NA
Spatter (1995)	84.6	84.9	1.26	56.6
Collins (1996)	85.8	86.3	1.14	59.9
Charniak (1997)	87.5	87.4	1	62.1
Collins (1997)	88.1	88.6	0.91	66.5
Charniak (2000)	90.1	90.1	0.74	70.1

Thanks