自然语言处理

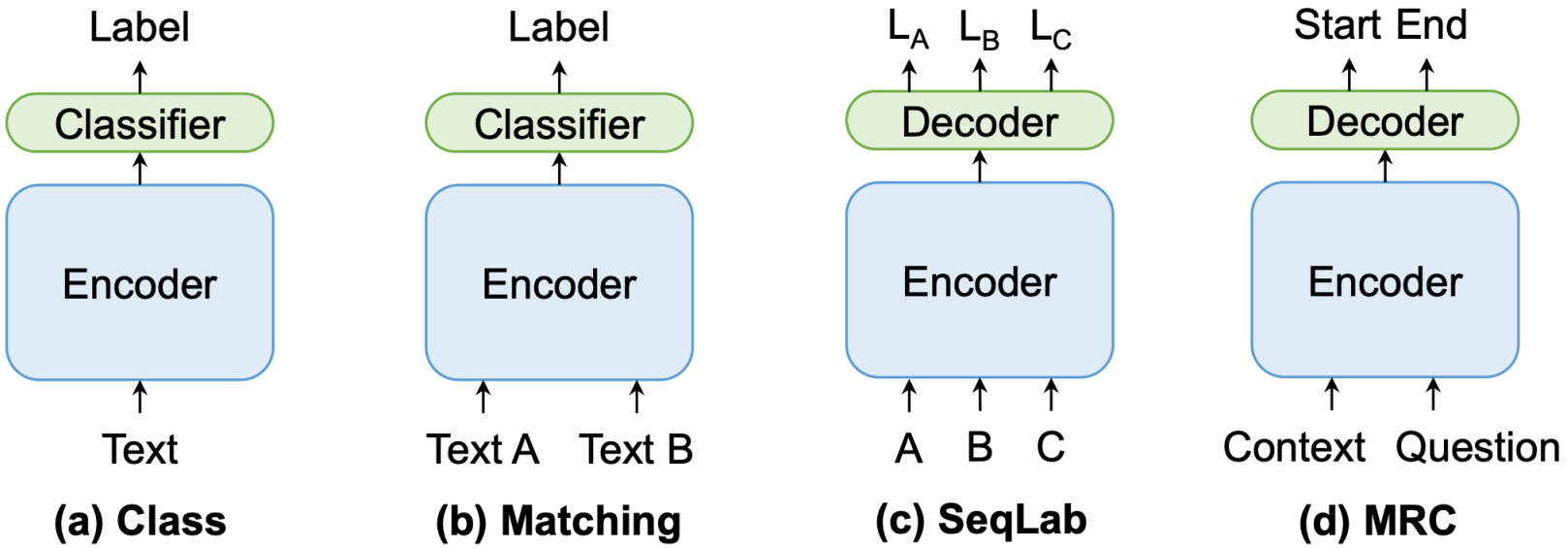
句法分析

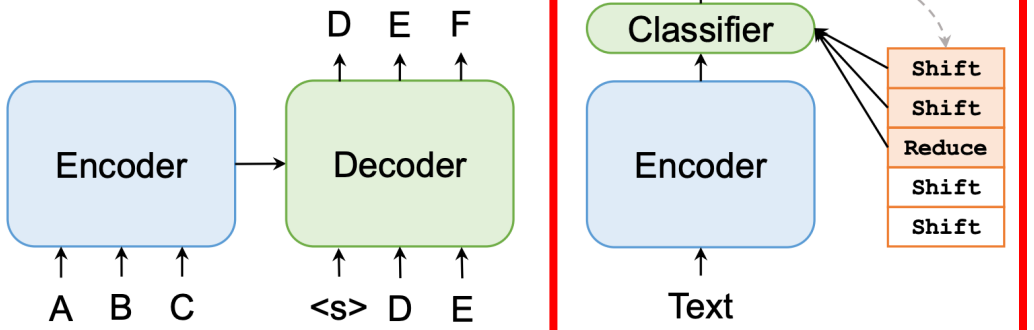
吴震

南京大学人工智能学院
南京大学自然语言处理研究组

2023年6月

自然语言处理中典型的任务形式





目录

01

背景知识

BACKGROUND

语言

- 语言的定义
  - 一种由三部分组成的符号交流系统：记号、意义和两者间的对应关系
  - 由组合语法规则制约、旨在传达语义的记号形式系统。
- 语言背后存在一些固定的结构搭配
  - 动宾结构
  - 主谓结构
  - 介宾短语
  - ……


```c
#include <stdio.h>

int main(int argc, char **argv) {
    printf("hello world !!!");
    return 0;
}
```

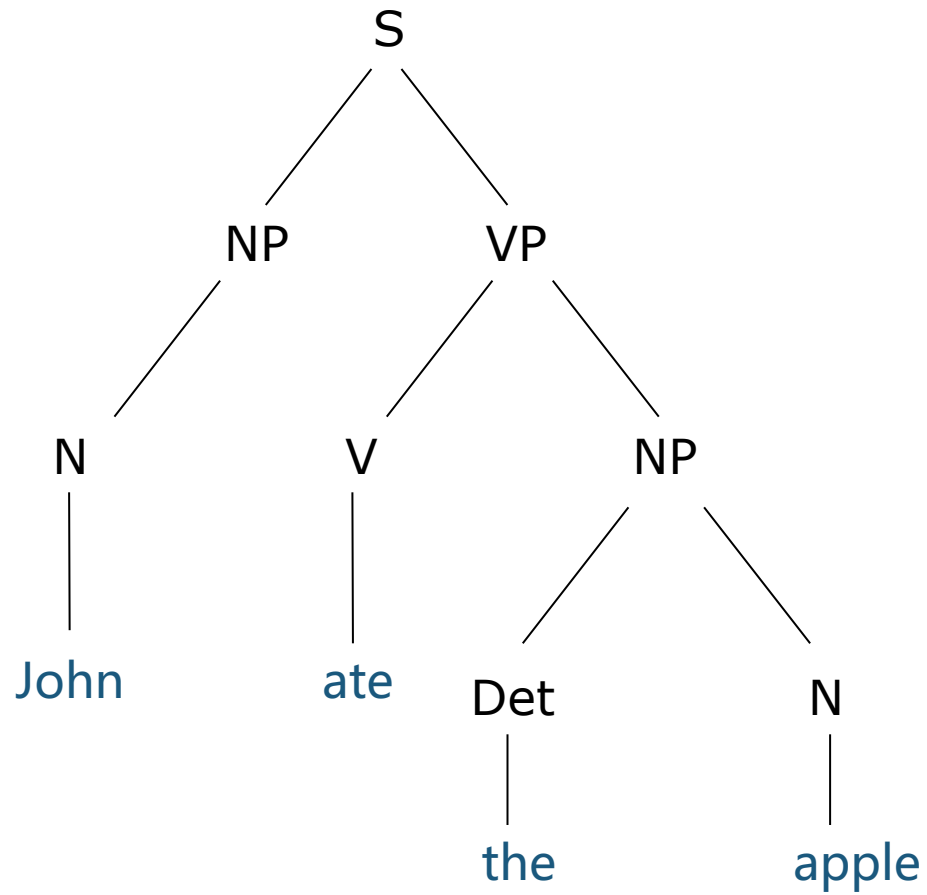
这些规则使得语言生成遵循某些约束，降低了语言习得和使用的难度

句法 (SYNTAX)

- 句法：一门语言里支配句子结构，决定词、短语、从句等句子成分如何组成其上级成分，直到组成句子的规则或过程。

John ate the apple

成分句法

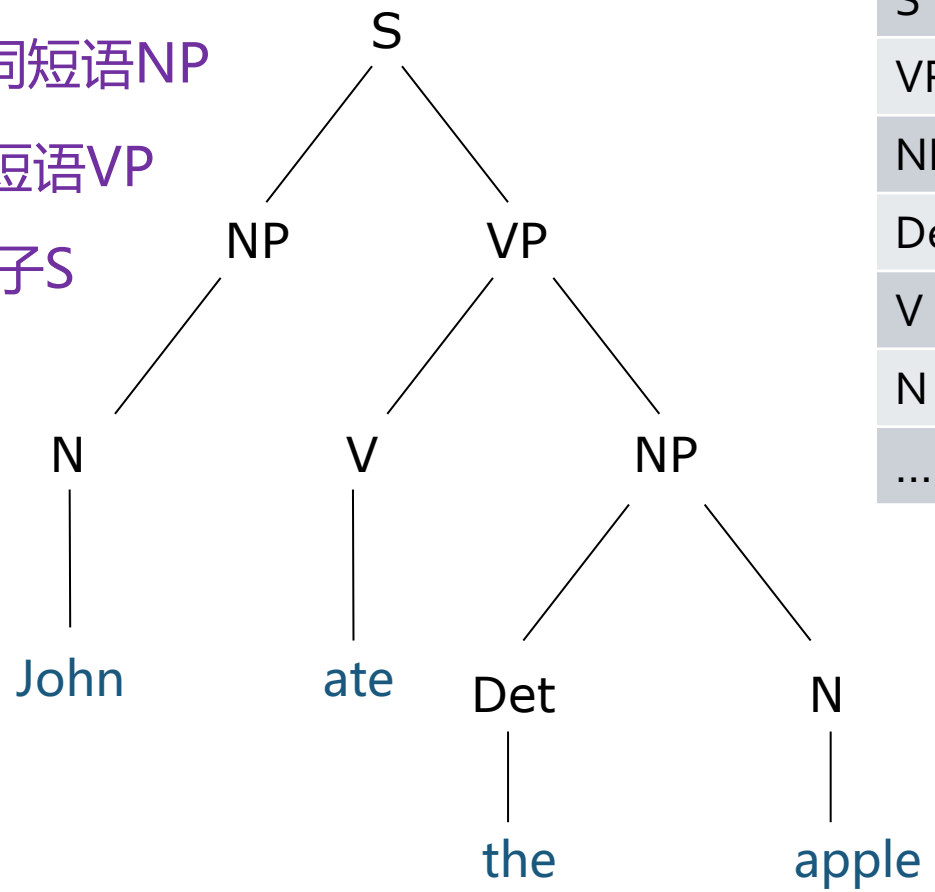


句法分析 (PARSING)

- 确定句子的组成
  - 词、短语以及它们之间的关系
- 句法分析类型
  - 成分句法分析（ Constituency Parsing ）
    - 研究词如何构成短语、短语如何构成句子
  - 依存句法分析（Dependency Parsing ）
    - 研究词之间的依赖（或支配）关系

成分句法分析

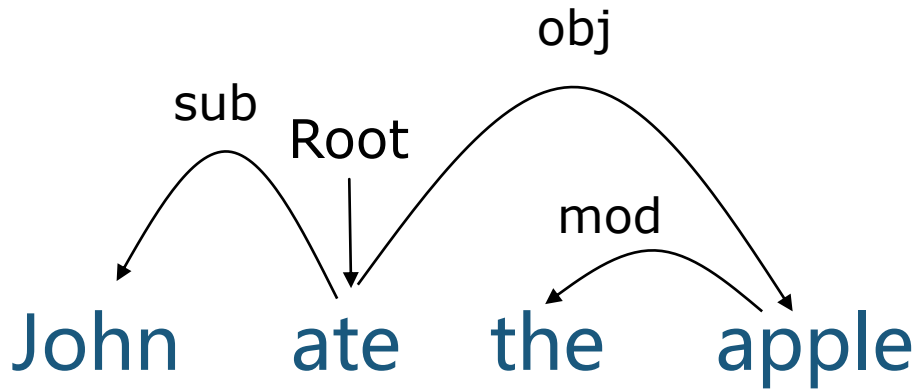
- 研究词如何构成短语、短语如何构成句子
- “John ate the apple”
  - 限定词 the 和名词 apple 构成名词短语NP
  - 动词 ate 和名词短语NP构成动词短语VP
  - 名词短语NP和动词短语VP构成句子S



词性	含义
S	sentence
VP	verb phrase
NP	noun phrase
Det	determiner
V	verb
N	noun
……	……

依存句法分析

- 识别词之间的依赖（或支配）关系
  - 依存是有向的：词与词之间的依赖关系是二元不对称的（“箭头”），“箭头”头部指向的词依赖“箭头”尾部指向的词（称为依存头）
  - 依存边是有类型的：表明两个词之间的依赖关系类型，如主语 (sub)、宾语 (obj)等
  - 每个词只有一个依存头：没有环，依存关系是树结构
- “John ate the apple”
  - John依赖于ate，是ate的主语
  - ate依赖于虚拟根节点 (ROOT)
  - the依赖于apple，是apple的修饰词
  - apple依赖于ate，是ate的宾语

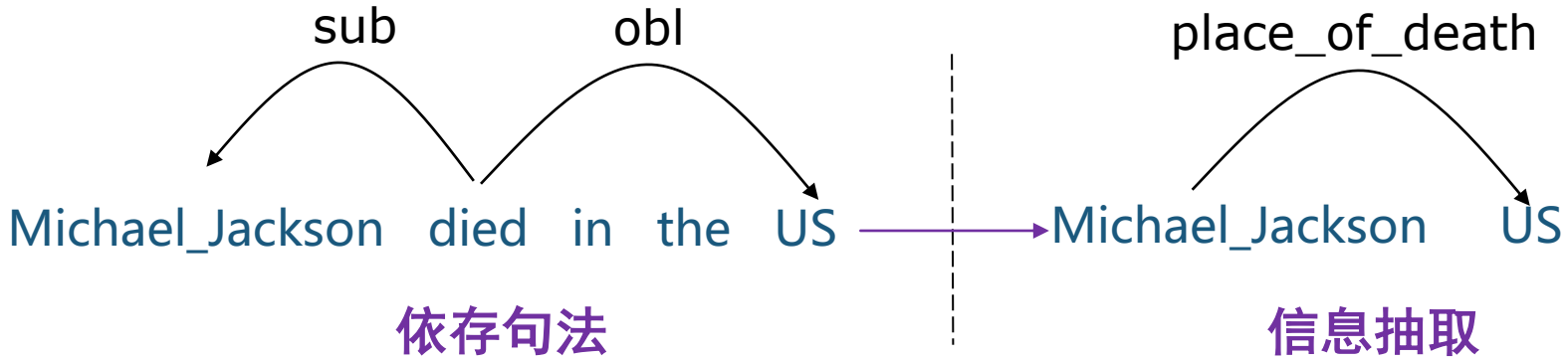


句法分析的应用

- 语法检查
- 信息抽取
- 问答系统
- 机器翻译
- ……

如何（形式化）表示语言背后的句法规则？

语法检查



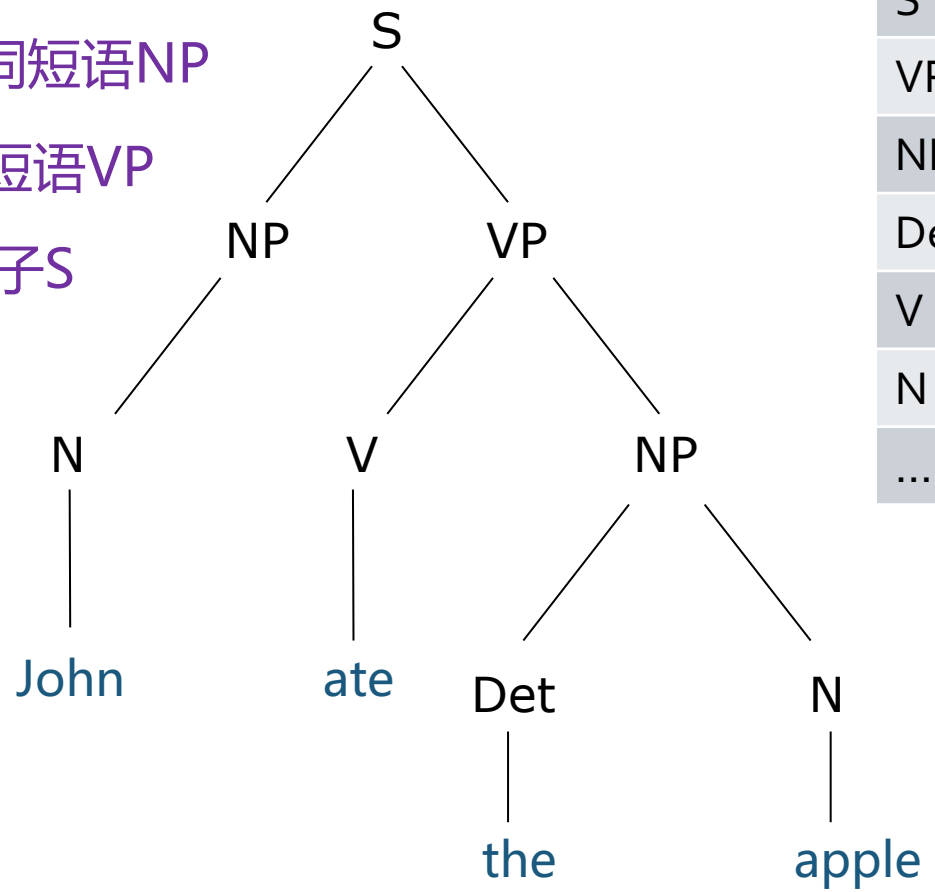
02

成分句法分析

CONSTITUENCY PARSING

成分句法分析

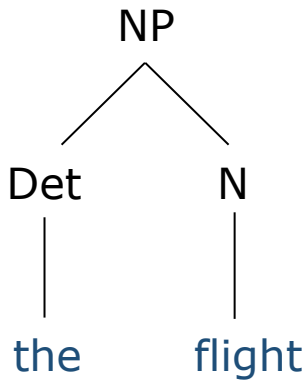
- 研究词如何构成短语、短语如何构成句子
- “John ate the apple”
  - 限定词 the 和名词 apple 构成名词短语NP
  - 动词 ate 和名词短语NP构成动词短语VP
  - 名词短语NP和动词短语VP构成句子S



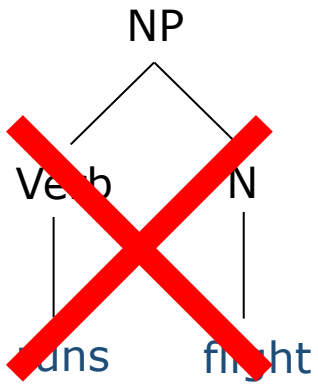
词性	含义
S	sentence
VP	verb phrase
NP	noun phrase
Det	determiner
V	verb
N	noun
……	……

上下文无关文法

- Context-free Grammar (CFG)
  - CFG定义语言中的有效结构，它通过形式化方法定义句子中有意义的成分，以及一个成分是如何由其他成分（单词或短语）构成的。



NP → Det N



NP → V N

上下文无关文法

- CFG一般由四元组构成$G=(N, T, S, R)$
  - N：非终结符集合
  - T：终结符集合
  - S：开始符号
  - R：产生式规则集合
    - 形式一般为 $X \rightarrow Y_1 Y_2 \ldots Y_n$, $n>0$, $X \in N$, $Y_i \in (N \cup T)$

上下文无关文法示例

- N = {S, NP, VP, PP, Det, N, V, P}
- T = {mom, caviar, spoon, ate, the, a, with}
- R =

S → N VP S → NP VP	N → mom N → caviar N → spoon
NP → Det N NP → NP PP	V → ate
VP → V NP VP → VP PP	Det → the Det → a
PP → P NP	P → with

$X \rightarrow Y_1\ Y_2 \ldots Y_n$

CFG中产生式右侧项数不定，不利于进行句法分析

乔姆斯基范式

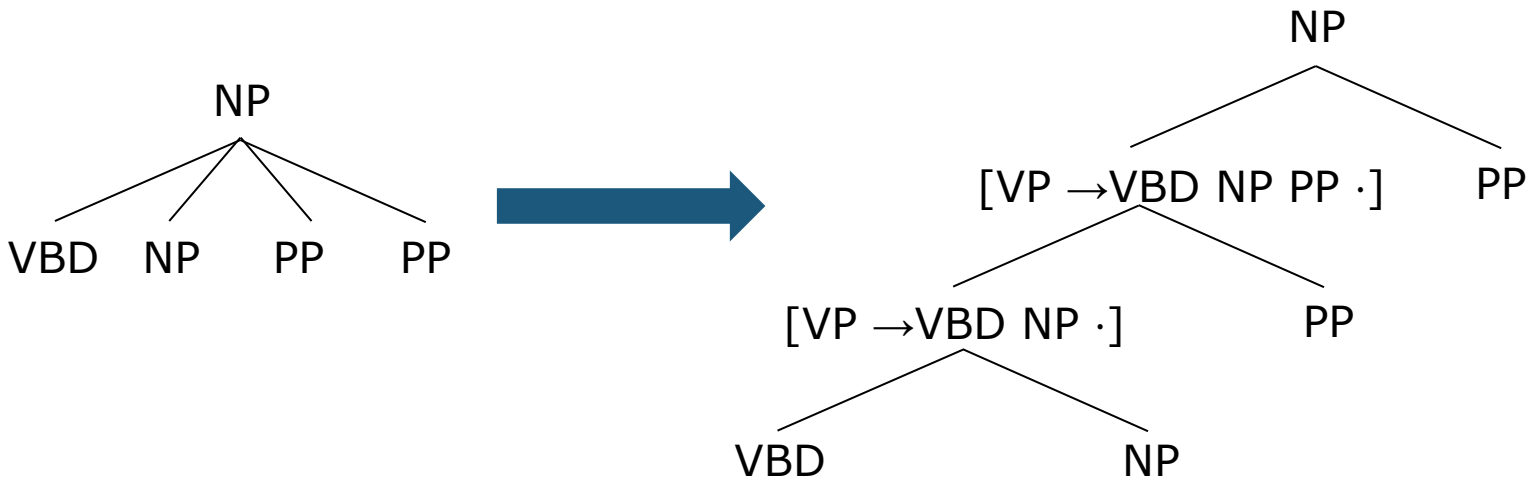
南京大学 NANJING UNIVERSITY

- 一种特殊的上下文无关文法
- 乔姆斯基范式 (Chomsky Normal Form, CNF)中的产生式满足以下两种形式：
  - $X \rightarrow Y_1\ Y_2$, *for* $X \in N$, and $Y_1, Y_2 \in N$
  - $X \rightarrow Y$, *for* $X \in N$, and $Y \in T$

- 目的：使句法分析的分析算法变得简单

乔姆斯基范式

- 如何将普通的上下文无关文法转化为乔姆斯基范式？
  - 将多叉树转化为多层二叉树



如何得到文法中的句法产生式规则？

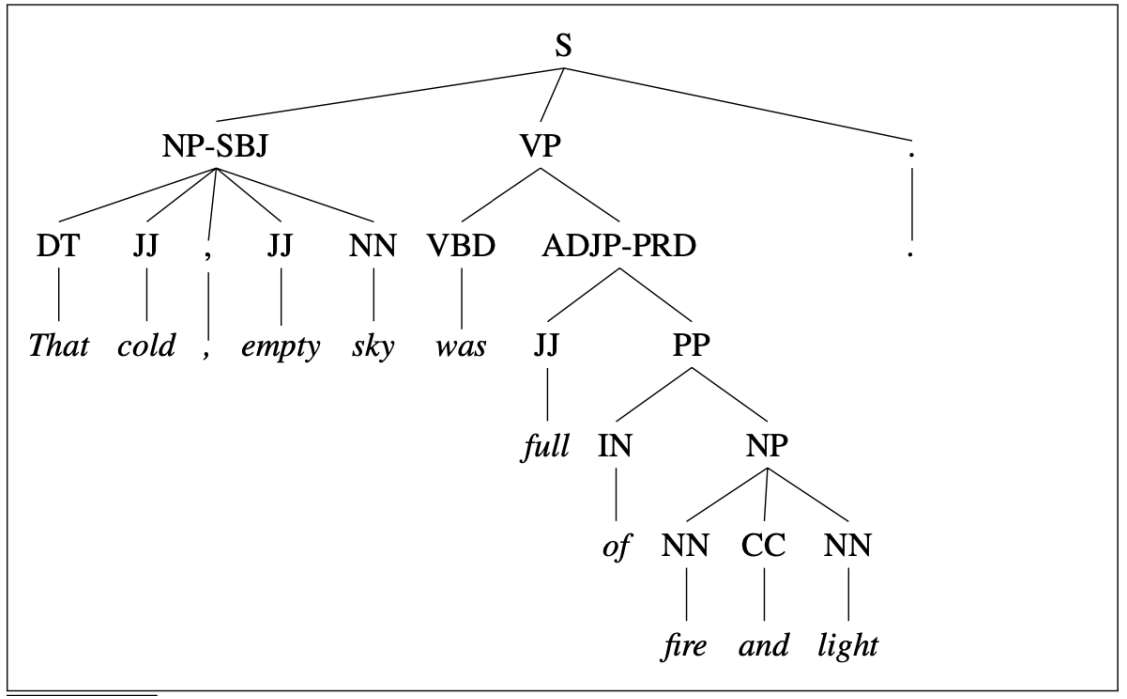
树库 (TREEBANK)

- 带用句法结构注释的句子集合
  - 人工对句子进行句法标注，将标注结果收集起来形成树库
  - Brown corpus (1967)
  - Lancaster-IBM Treebank (1980s)
  - The Penn Treebank (1993)
- 构建树库的好处
  - 构建句法分析器，可重复使用
  - 用于评价句法分析器的性能

树库 (TREEBANK)

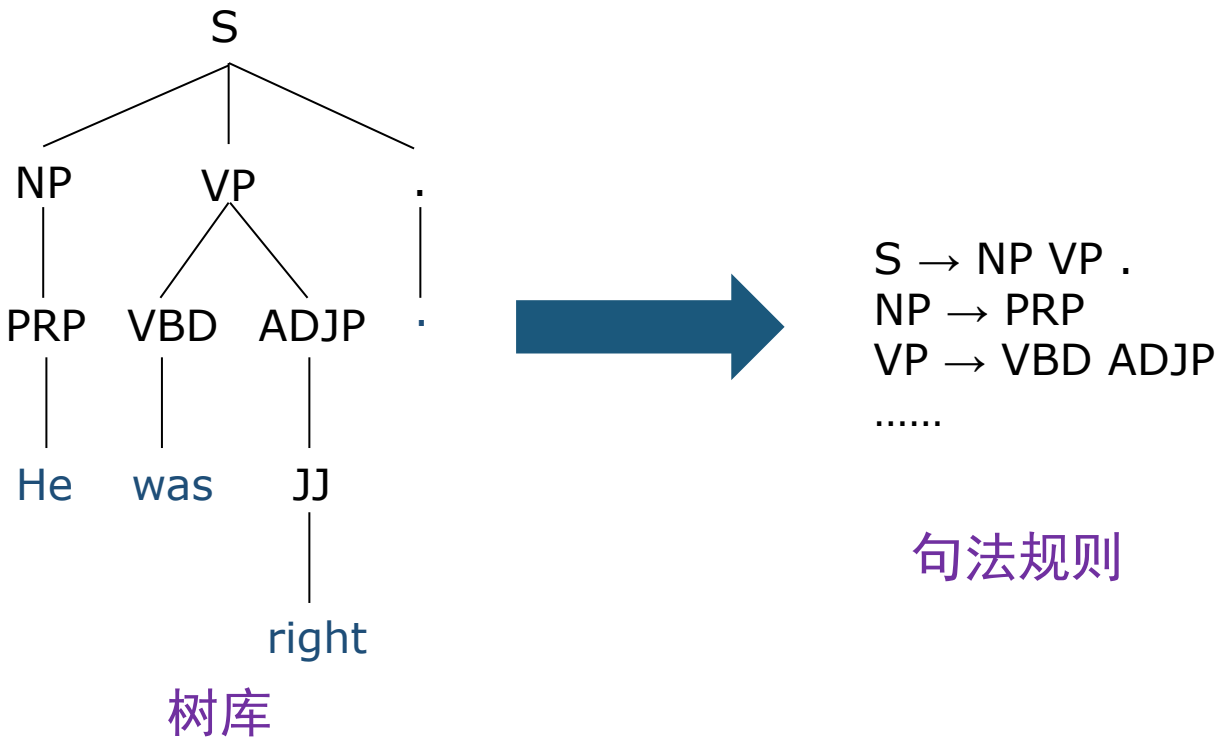
- The Penn Treebank样例

```
((S
   (NP-SBJ (DT That)
     (JJ cold) (, ,)
     (JJ empty) (NN sky) )
   (VP (VBD was)
     (ADJP-PRD (JJ full)
       (PP (IN of)
         (NP (NN fire)
           (CC and)
           (NN light) ))))
   (. .) ))
```



树库 (TREEBANK)

- 从树库中抽取句法规则



如何根据句法规则（文法G）对句子进行句法分析？

句法分析-自顶向下

- 输入：待分析的句子，文法G
- 输出：句子对应的句法树
- 算法流程：
  1. 取 ((S) 1)作为当前状态（初始状态），后备状态为空。
  2. 若当前状态为空，则失败，算法结束，
  3. 否则，若当前状态的符号表为空，
     ① 位置计数器值处于句子末尾，则成功，算法结束
     ② 位置计数器值处于句子中间，转5
  4. 否则，进行状态转换，若转换成功，则转2
  5. 否则，回溯，转2。

S → NP VP
NP → Det N NP → Det ADJ N
VP → V VP → VP NP

N → cat N → house
V → caught
Det → the Det → a

句法分析-自顶向下

- “$_1$ The $_2$ cat $_3$ caught $_4$ a $_5$ mouse $_6$”的分析过程

1. S→NP VP 2. NP→Det N 3. NP→Det ADJ N 4. VP→V 5. VP→V NP

步骤	*当前状态*	*后备状态*	*备注*
1	((S) 1)		初始状态
2	((NP VP) 1)		规则1改写
3	((Det N VP) 1)	((Det ADJ N VP) 1)	规则2、3改写
4	((N VP) 2)	((Det ADJ N VP) 1)	Det匹配the
5	((VP) 3)	((Det ADJ N VP) 1)	N匹配cat
6	((V) 3)	((V NP) 3) ((Det ADJ N VP) 1)	规则4、5改写
7	(() 4)	((V NP) 3) ((Det ADJ N VP) 1)	V匹配caught

句法分析-自顶向下

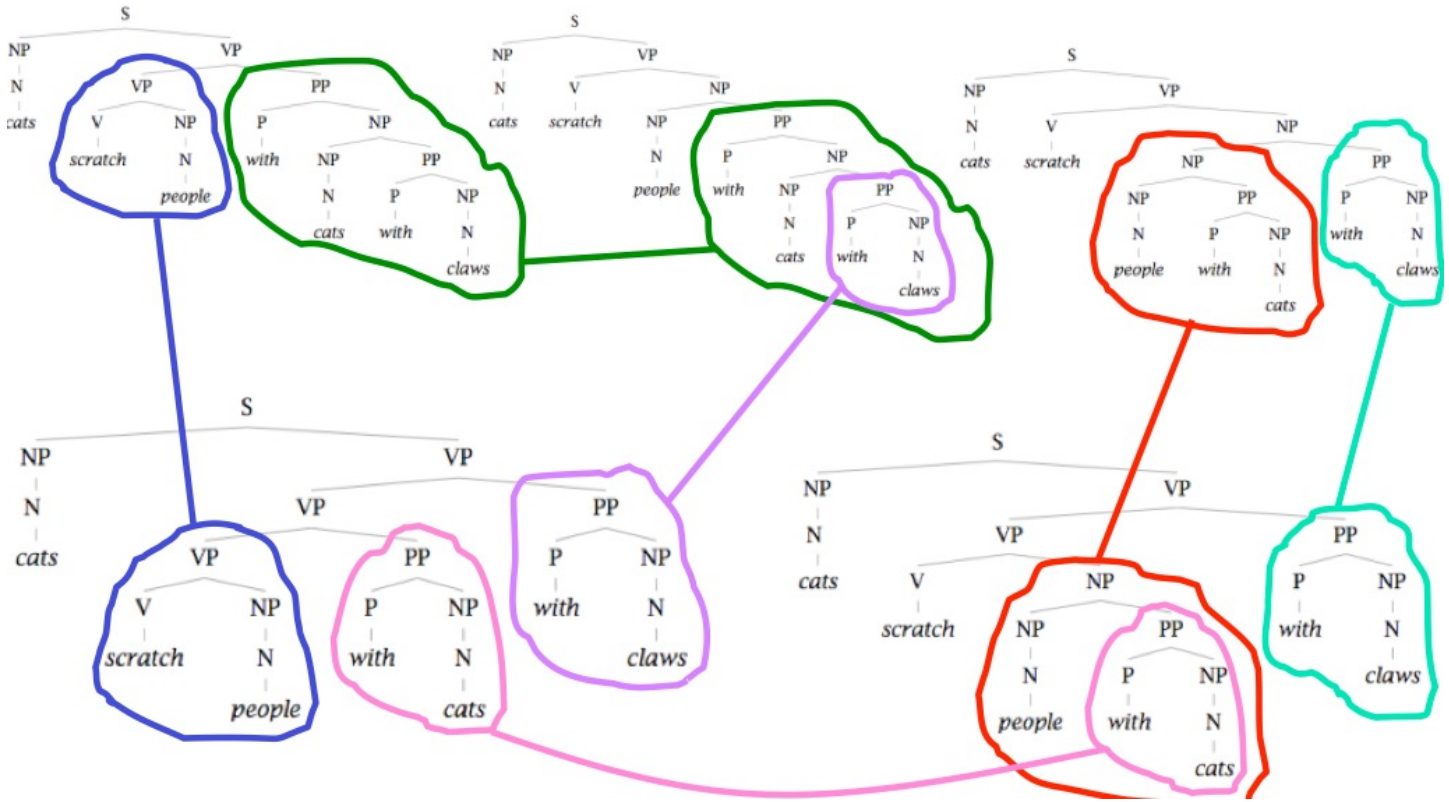
- “$_1$ The $_2$ cat $_3$ caught $_4$ a $_5$ mouse $_6$”的分析过程（续）

1. S→NP VP 2. NP→Det N 3. NP→Det ADJ N 4. VP→V 5. VP→V NP

步骤	*当前状态*	*后备状态*	*备注*
8	((V NP) 3)	((Det ADJ N VP) 1)	回溯
9	((NP) 4)	((Det ADJ N VP) 1)	V匹配caught
10	((Det N) 4)	((Det ADJ N) 4) ((Det ADJ N VP) 1)	规则2、3改写
11	((N) 5)	((Det ADJ N) 4) ((Det ADJ N VP) 1)	Det匹配a
12	(() 6)	((Det ADJ N) 4) ((Det ADJ N VP) 1)	N匹配mouse
13			结束

自顶向下分析方法的问题

- 需要搜索的树数量达到指数级别
- 许多子树是相同的，存在重复解析



自底向上
对已分析的子树进行存储（记忆化），避免重复解析

CKY算法

- Cocke-Kasami-Younger Algorithm
- 自底向上的动态规划算法

Mom	ate	the	caviar	with	a	spoon
N	V	Det	N	P	Det	N

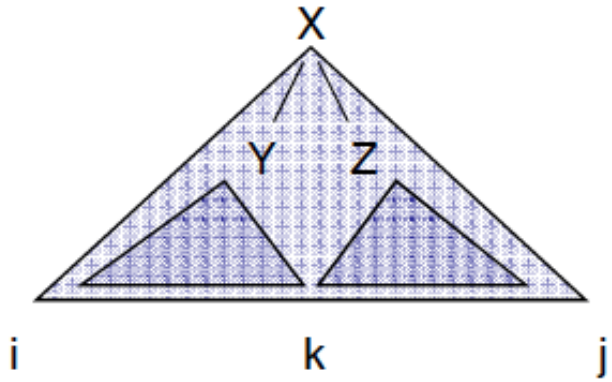
待分析句子

S → N VP S → NP VP	N → mom N → caviar N → spoon
NP → Det N NP → NP PP	V → ate
VP → V NP VP → VP PP	Det → the Det → a
PP → P NP	P → with

句法规则

CKY算法流程

- 输入：待分析的句子，文法G
- 输出：句子对应的句法树
- 算法流程
  - for i := 1 to n
    - add to [i-1, i] all POS tags for the i^{th} word　设置主对角线词性
  - for width := 2 to n　枚举成分覆盖的长度范围
    - for i := 0 to n-width　枚举开始位置
      - j := start + width　计算结束位置
      - for k := i+1 to j　枚举左右分割节点的位置
        - for every rule X →Y Z in the CNF G　枚举CNF中的句法规则
          - if Y in [i, k] and Z in [k, j]　判断[i, k]和[k, j]两个区域的成分是否分别存在Y和Z
            - then add X to [i, j]　将成分X添加到[i, j]对应的区域中



CKY算法示例

0	1	2	3	4	5	6	7
	Mom	ate	the	caviar	with	a	spoon
	N	V	Det	N	P	Det	N

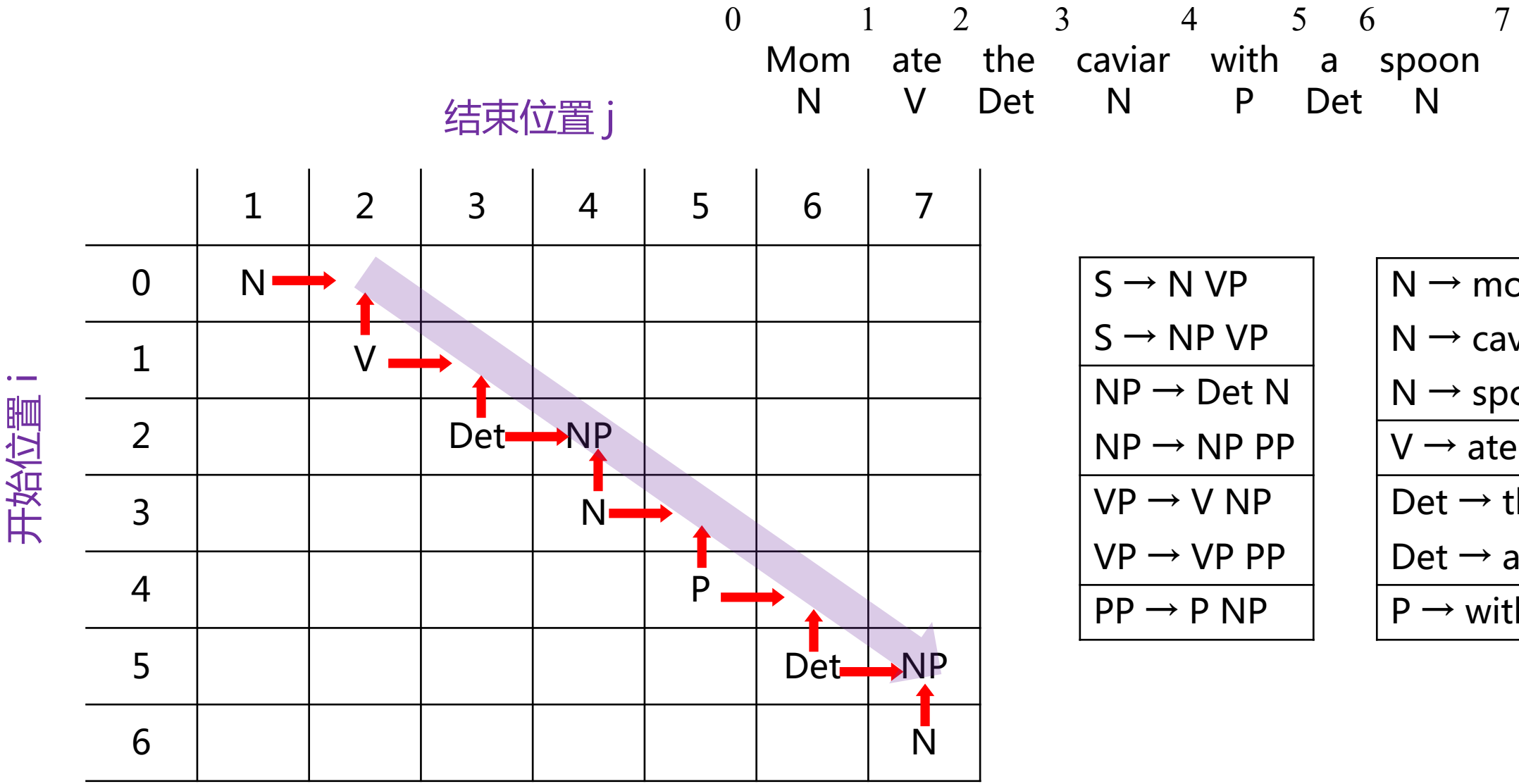
结束位置 j

开始位置 i

	1	2	3	4	5	6	7
0	N						
1		V					
2			Det				
3				N			
4					P		
5						Det	
6							N

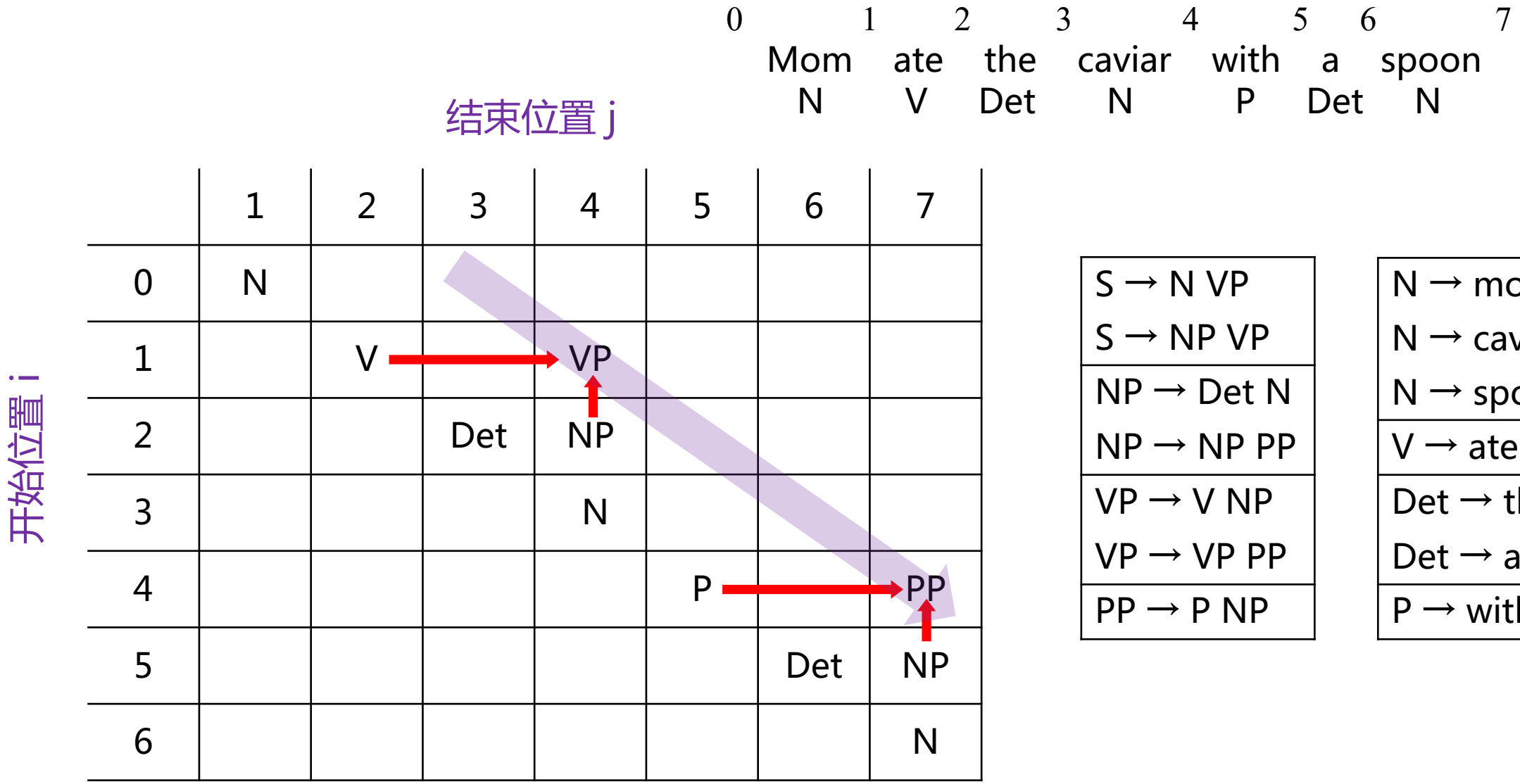
S → N VP	N → mom
S → NP VP	N → caviar
NP → Det N	N → spoon
NP → NP PP	V → ate
VP → V NP	Det → the
VP → VP PP	Det → a
PP → P NP	P → with

CKY算法示例



S → N VP S → NP VP	N → mom N → caviar N → spoon
NP → Det N NP → NP PP	V → ate
VP → V NP VP → VP PP	Det → the Det → a
PP → P NP	P → with

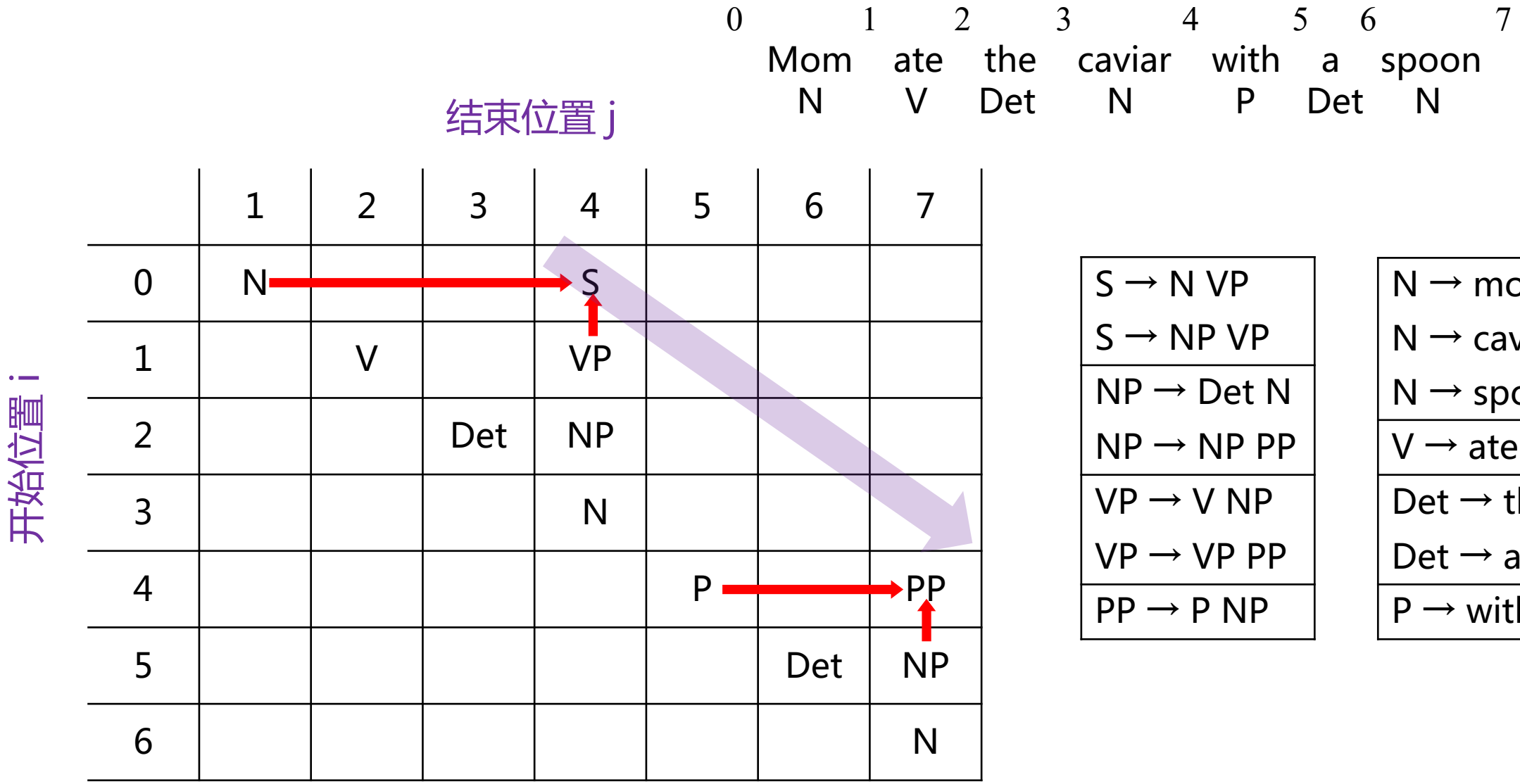
CKY算法示例



S → N VP
S → NP VP
NP → Det N
NP → NP PP
VP → V NP
VP → VP PP
PP → P NP

N → mom
N → caviar
N → spoon
V → ate
Det → the
Det → a
P → with

CKY算法示例



S → N VP
S → NP VP
NP → Det N
NP → NP PP
VP → V NP
VP → VP PP
PP → P NP

N → mom
N → caviar
N → spoon
V → ate
Det → the
Det → a
P → with

CKY算法示例

0	1	2	3	4	5	6	7
	Mom	ate	the	caviar	with	a	spoon
	N	V	Det	N	P	Det	N

结束位置 j

开始位置 i

	1	2	3	4	5	6	7
0	N			S			
1		V		VP			
2			Det	NP			NP
3				N			
4					P		PP
5						Det	NP
6							N

S → N VP	N → mom
S → NP VP	N → caviar
NP → Det N	N → spoon
NP → NP PP	V → ate
VP → V NP	Det → the
VP → VP PP	Det → a
PP → P NP	P → with

CKY算法示例

	0	1	2	3	4	5	6	7
		Mom	ate	the	caviar	with	a	spoon
		N	V	Det	N	P	Det	N

结束位置 j

开始位置 i

	1	2	3	4	5	6	7
0	N			S			
1		V		VP			VP
2			Det	NP			NP
3				N			
4					P		PP
5						Det	NP
6							N

S → N VP S → NP VP	N → mom N → caviar N → spoon
NP → Det N NP → NP PP	V → ate
VP → V NP VP → VP PP	Det → the Det → a
PP → P NP	P → with

CKY算法示例

	0	1	2	3	4	5	6	7
		Mom	ate	the	caviar	with	a	spoon
		N	V	Det	N	P	Det	N

结束位置 j

开始位置 i

	1	2	3	4	5	6	7
0	N			S			S
1		V		VP			VP
2			Det	NP			NP
3				N			
4					P		PP
5						Det	NP
6							N

S → N VP
S → NP VP
NP → Det N
NP → NP PP
VP → V NP
VP → VP PP
PP → P NP

N → mom
N → caviar
N → spoon
V → ate
Det → the
Det → a
P → with

CKY算法总结

- 动态规划
  - 将分析中间结果存放在表中，减少重复计算
- 复杂度：$O(n^3|G|)$
  - n：句子长度
  - $|G|$：文法中规则的数量
- 算法前提：上下文无关文法CFG转换为乔姆斯基范式CNF

CKY的问题

	0	1	2	3	4	5	6	7
		Mom	ate	the	caviar	with	a	spoon
		N	V	Det	N	P	Det	N

结束位置 j

开始位置 i

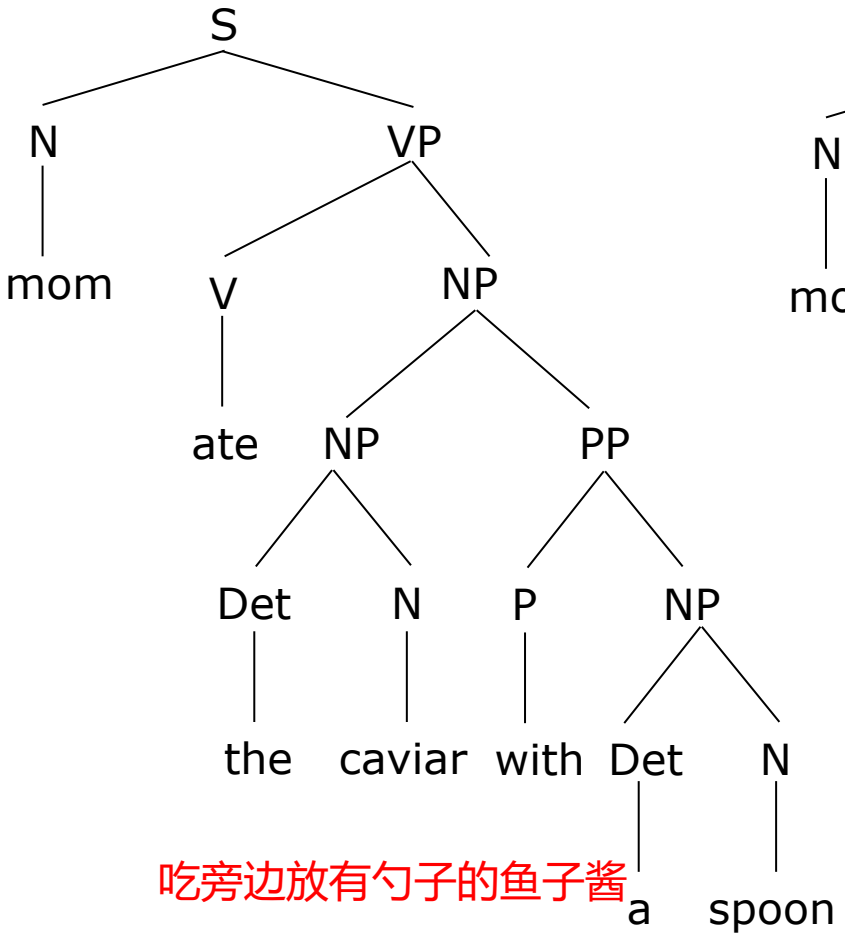
	1	2	3	4	5	6	7
0	N			S			
1		V		VP			VP
2			Det	NP			NP
3				N			
4					P		PP
5						Det	NP
6							N

S → N VP	N → mom
S → NP VP	N → caviar
	N → spoon
NP → Det N	V → ate
NP → NP PP	
VP → V NP	Det → the
VP → VP PP	Det → a
PP → P NP	P → with

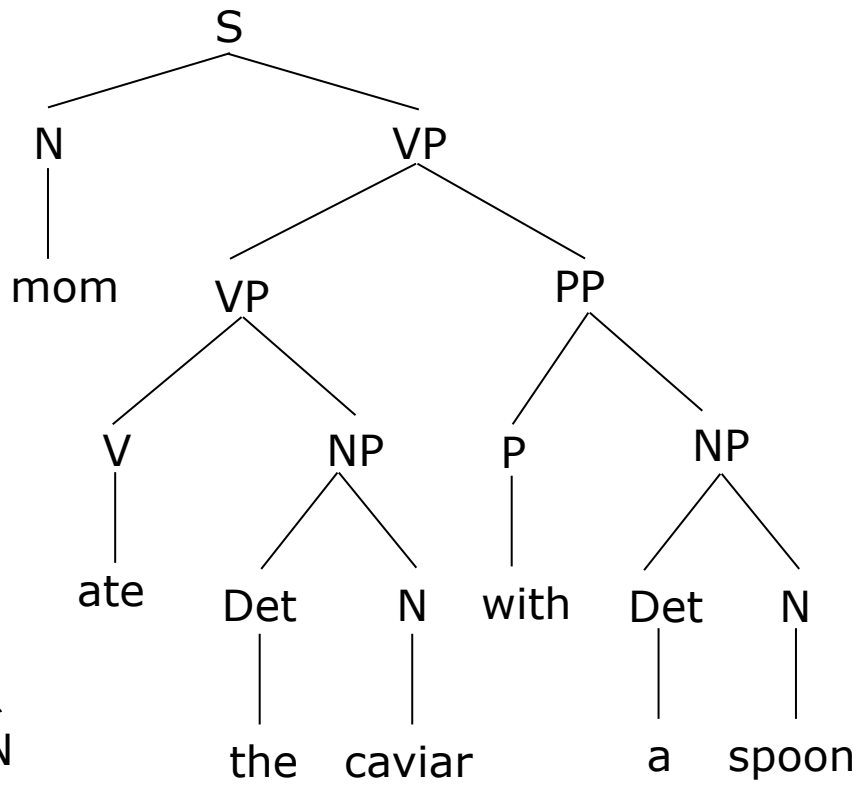
CKY的问题

- 歧义

Mom ate the caviar with a spoon
N V Det N P Det N



吃旁边放有勺子的鱼子酱



用勺子吃鱼子酱

S → N VP S → NP VP	N → mom N → caviar N → spoon
NP → Det N NP → NP PP	V → ate
VP → V NP VP → VP PP	Det → the Det → a
PP → P NP	P → with

概率上下文无关文法PCFG

- PCFG一般由五元组构成$G=(N, T, S, R, P)$
  - N：非终结符集合
  - T：终结符集合
  - S：开始符号
  - R：产生式规则集合
    - 形式一般为 $X \rightarrow Y_1\ Y_2 \ldots Y_n$, $for\ n \geq 0, X \in N, Y_i \in (N \cup T)$
  - $P(X \rightarrow Y_1\ Y_2 \ldots Y_n)$：产生式对应的概率

概率上下文无关文法PCFG

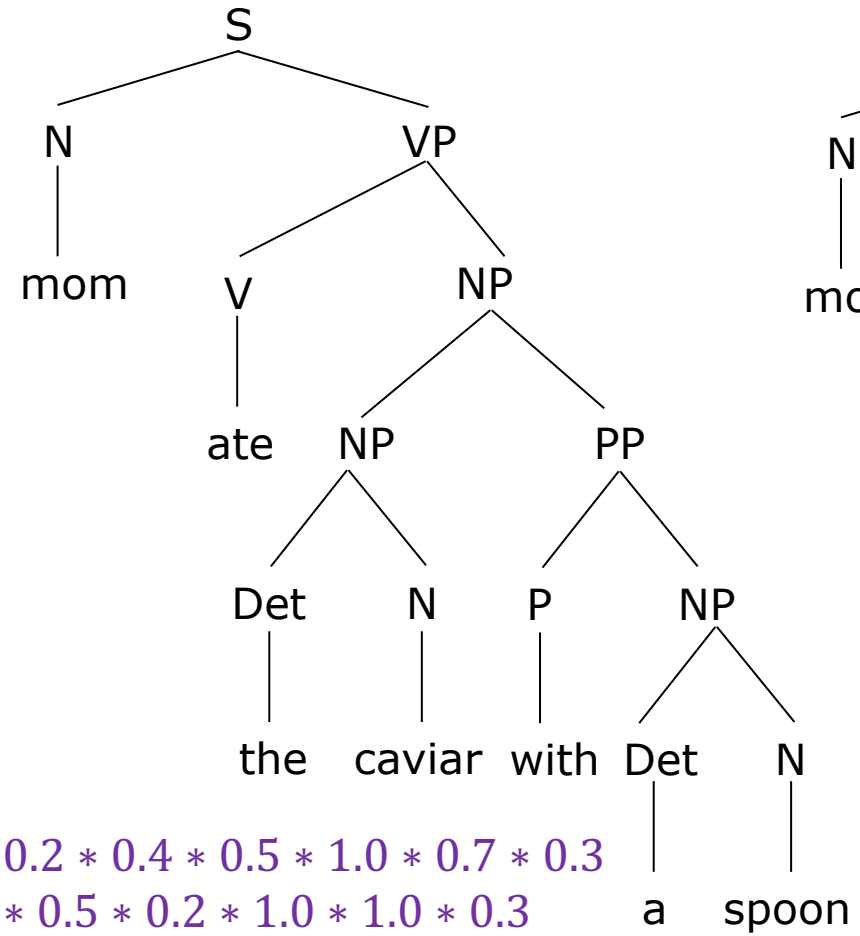
- PCFG由五元组构成*G=(N, T, S, R, P)*
- 利用PCFG计算一棵句法树 t 的概率
  - 句法树 t 通过以下产生式得到：
    - $\alpha_1 \rightarrow \beta_1, \alpha_2 \rightarrow \beta_2, \ldots, \alpha_n \rightarrow \beta_n$
  - 句法树 t 的概率为：
    - $p(t) = \prod_{i=1}^{n} p(\alpha_i \rightarrow \beta_i)$

Rules	p
S → N VP	0.2
S → NP VP	0.8
NP → Det N	0.3
NP → NP PP	0.7
VP → V NP	0.5
VP → VP PP	0.5
PP → P NP	1.0

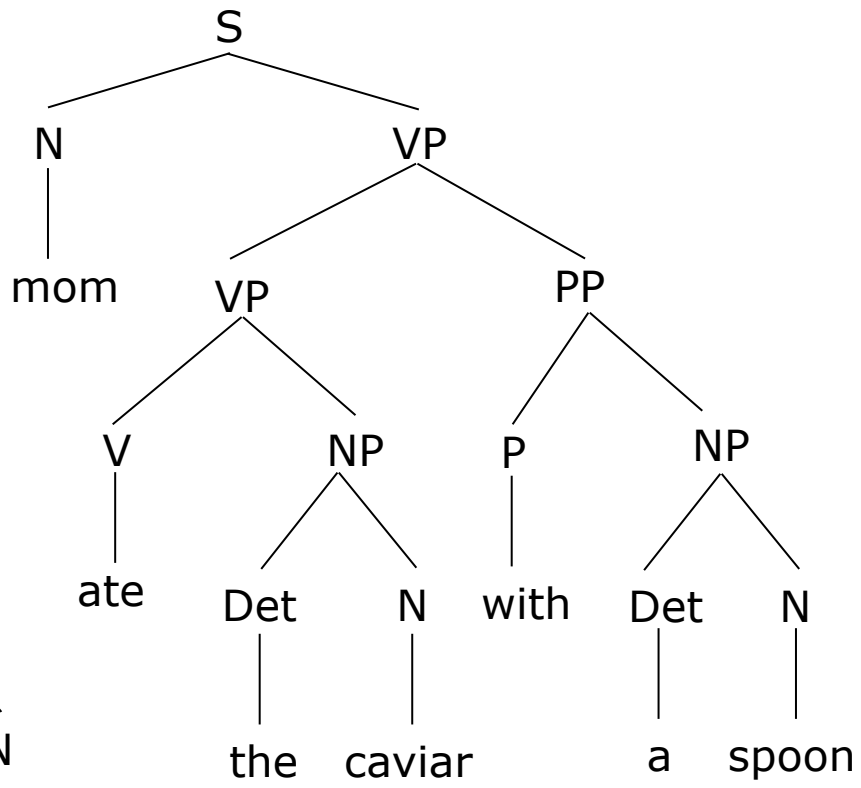
Rules	p
N → Mom	0.4
N → caviar	0.2
N → spoon	0.2
V → ate	1.0
Det → the	0.5
Det → a	0.5
P → with	1.0

概率上下文无关文法PCFG

- PCFG由五元组构成$G=(N, T, S, R, P)$



$$0.2 * 0.4 * 0.5 * 1.0 * 0.7 * 0.3 * 0.5 * 0.2 * 1.0 * 1.0 * 0.3 * 0.5 * 0.2 = 2.52 * 10^{-5}$$



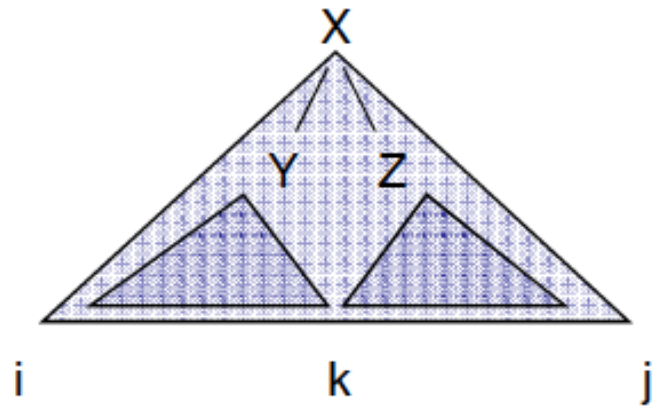
$$0.2 * 0.4 * 0.5 * 0.5 * 1.0 * 0.3 * 0.5 * 0.2 * 1.0 * 1.0 * 0.3 * 0.5 * 0.2 = 1.8 * 10^{-5}$$

Rules	p
S → N VP	0.2
S → NP VP	0.8
NP → Det N	0.3
NP → NP PP	0.7
VP → V NP	0.5
VP → VP PP	0.5
PP → P NP	1.0

Rules	p
N → Mom	0.4
N → caviar	0.2
N → spoon	0.2
V → ate	1.0
Det → the	0.5
Det → a	0.5
P → with	1.0

概率CKY算法

- 输入：待分析的句子s，文法PCFG *G*
- 输出：句子s对应的概率最大的句法树以及最大概率bestscore(s)
- 算法流程
  - for i := 1 to n
    - for X := tags[s[i-1]]
      - score[X][i-1][i] = score(X → s[i-1])
  - for width := 2 to n
    - for i := 0 to n-width
      - j := start + width
      - for k := i+1 to j
        - for every rule X →YZ in the CNF G
          - score[X][i][j]=max(score[X][i][j], score(X →YZ)*score[Y][i][k]*score[Z][k][j])



PCFG中的概率估计

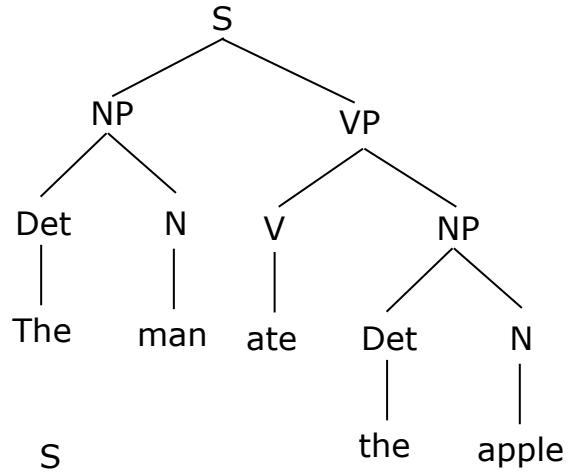
- 利用树库中的统计频率估计产生式概率
- 产生式$A \to \alpha$对应的概率为：

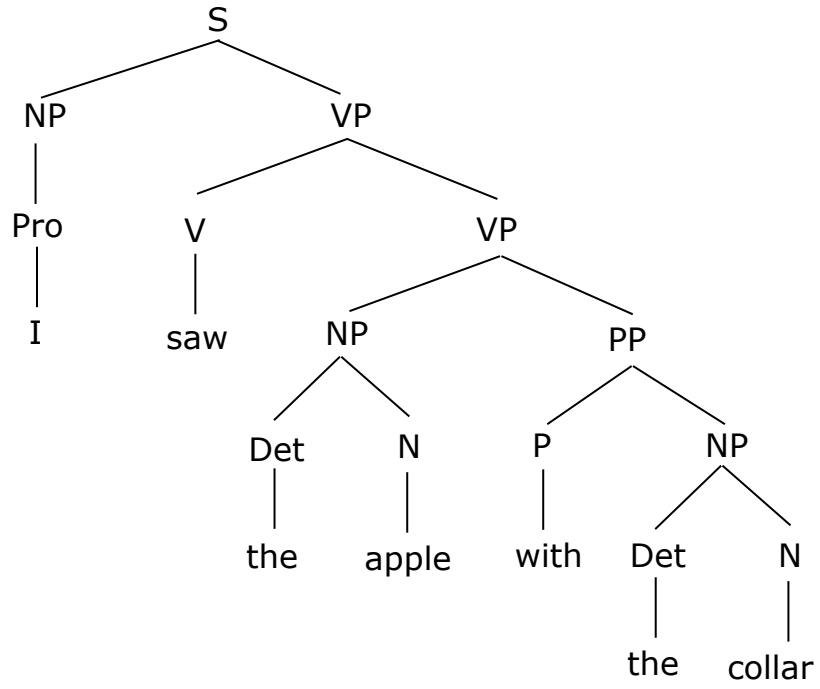
$$p(A \to \alpha) = \frac{Number(A \to \alpha)}{\sum_{\gamma} Number(A \to \gamma)}$$

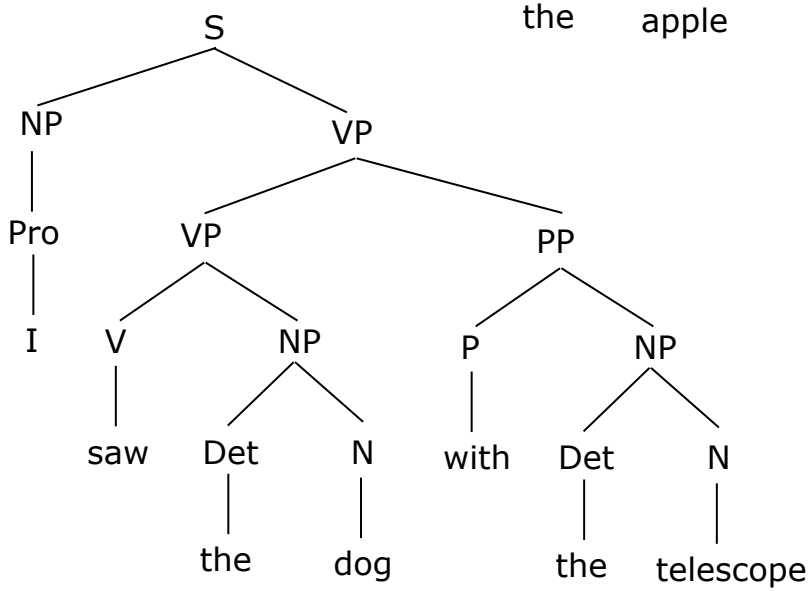
$$p(VP \to V) = \frac{Number(VP \to V)}{Number(VP \to V) + Number(VP \to VT\ NP) + Number(VP \to VP\ PP)}$$

PCFG中的概率估计

- 利用树库中的统计频率估计产生式概率



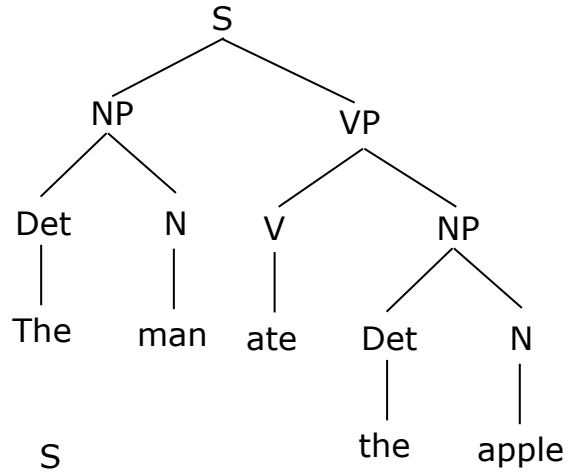


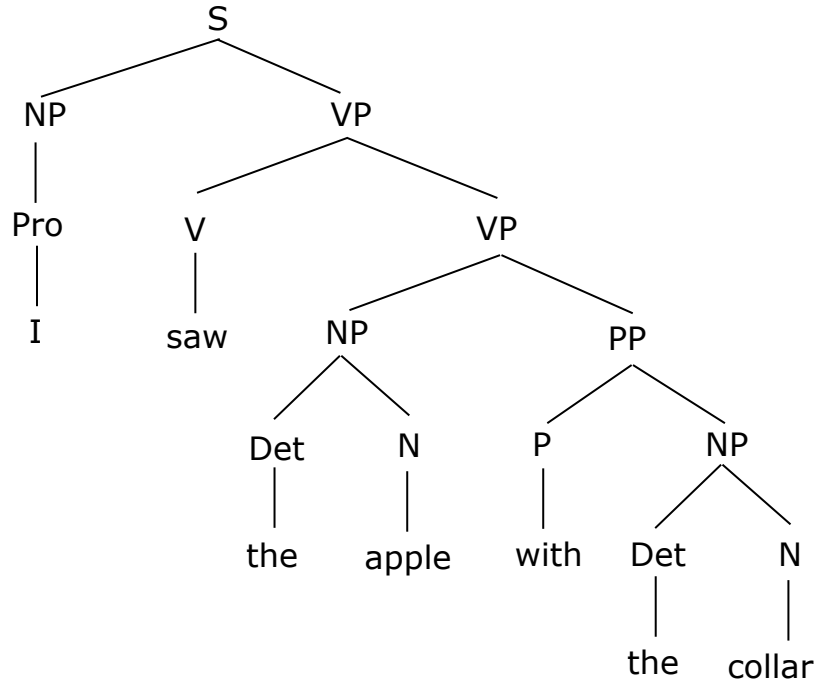


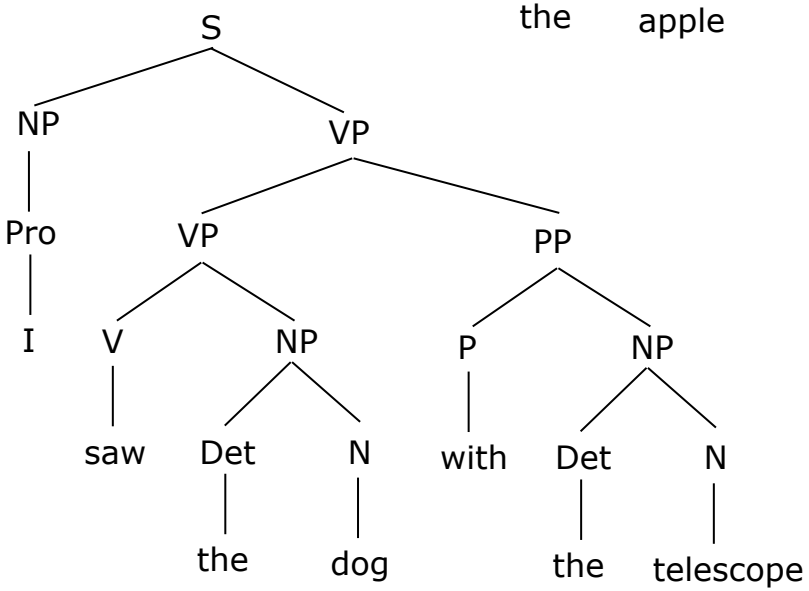
Rules	Number
S → N VP	3
NP → Pro	2
NP → Det N	6
NP → NP PP	1
VP → V NP	3
VP → VP PP	1
PP → P NP	2

PCFG中的概率估计

- 利用树库中的统计频率估计产生式概率







Rules	Probability
S → N VP	3/3=1.0
NP → Pro	2/9=0.22
NP → Det N	6/9=0.67
NP → NP PP	1/9=0.11
VP → V NP	3/4=0.75
VP → VP PP	1/4=0.25
PP → P NP	2/2=1.0

南京大学
NANJING UNIVERSITY

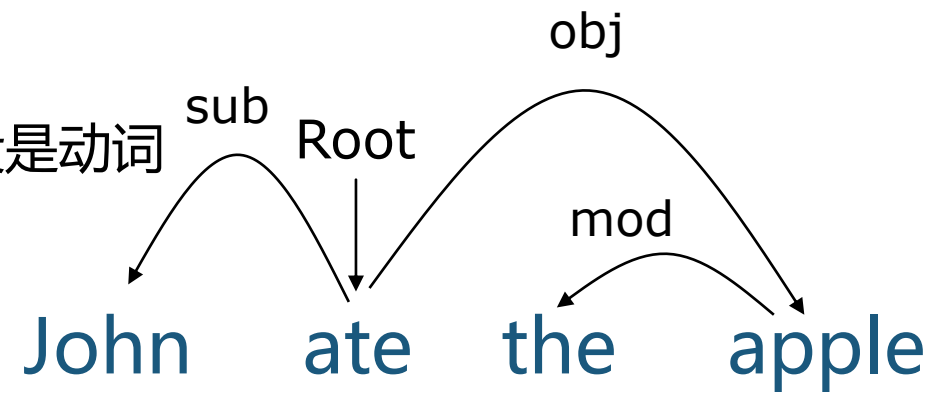
03

依存句法分析

DEPENDENCY PARSING

依存句法分析

- 识别词之间的依赖（或支配）关系
  - 依存是有向的：词与词之间的依赖关系是二元不对称的（“箭头”），“箭头”头部指向的词依赖“箭头”尾部指向的词（称为依存头）
  - 依存边是有类型的：表明两个词之间的依赖关系类型，如主语 (sub)、宾语 (obj)等
  - 每个词只有一个依存头：没有环，依存关系是树结构
- “John ate the apple”
  - John依赖于ate，是ate的主语
  - ate依赖于虚拟根节点 (ROOT)，根据点一般是动词
  - the依赖于apple，是apple的修饰词
  - Apple依赖于ate，是ate的宾语



依存句法分析算法

- 移进规约算法：分析过程是一个自底向上的动作序列生成过程
- 类似于shift-reduce分析，只是在归约时，增加左归约/右归约，表示两个节点的依赖方向
- 分析器维护三个数据结构
  - 一个栈 σ
  - 一个输入缓冲区 β
  - 一个依存边集合 A

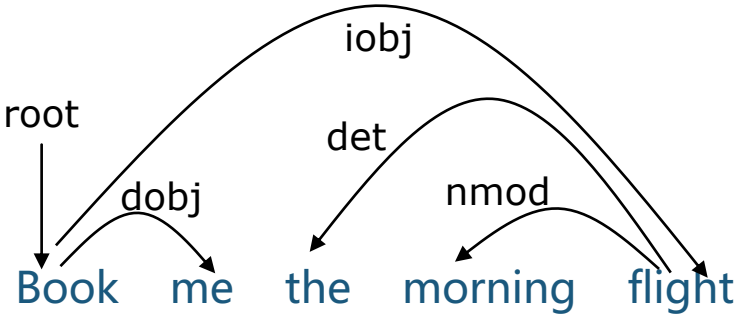
SHIFT-REDUCE PARSING

- 输入：待分析的句子s
- 输出：句子s对应的句法树
- 算法流程
  - Start: σ=[ROOT], $\beta = w_1, \ldots, w_n$, $\mathrm{A} = \emptyset$
    1. Shift $\sigma, w_i|\beta, \mathrm{A} \rightarrow \sigma|w_i, \beta, \mathrm{A}$
    2. Left-Arc $\sigma|w_i|w_j, \beta, \mathrm{A} \rightarrow \sigma|w_j, \beta, \mathrm{A} \cup \{r(w_j, w_i)\}$
    3. Right-Arc $\sigma|w_i|w_j, \beta, \mathrm{A} \rightarrow \sigma|w_i, \beta, \mathrm{A} \cup \{r(w_i, w_j)\}$
  - Finish: σ=[ROOT], $\beta = \emptyset$

SHIFT-REDUCE PARSING示例

- ” Book me the morning flight”

Step	Stack	Word List	Action	Relation Added
0	[root]	[book, me, the, morning, flight]	Shift	
1	[root, book]	[me, the, morning, flight]	Shift	
2	[root, book, me]	[the, morning, flight]	Right-Arc	(book → me)
3	[root, book]	[the, morning, flight]	Shift	
4	[root, book, the]	[morning, flight]	Shift	
5	[root, book, the, morning]	[flight]	Shift	
6	[root, book, the, morning, flight]	[]	Left-Arc	(morning ← flight)
7	[root, book, the, flight]	[]	Left-Arc	(the ← flight)
8	[root, book, flight]	[]	Right-Arc	(book → flight)
9	[root, book]	[]	Right-Arc	(root → flight)
10	[root]	[]	Done	

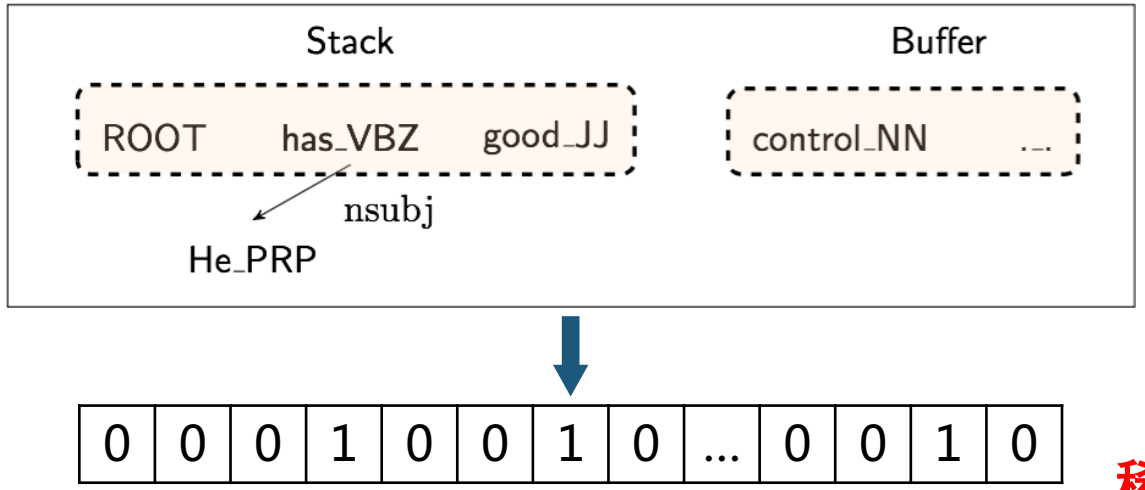


SHIFT-REDUCE PARSING

- 如何决定每一步的动作？
  - Shift
  - LeftArc
  - RightArc
- 机器学习
  - 统计学习
  - 深度学习

统计学习特征

- 词性、栈/缓冲区的单词
- 语言的形态学特征
- 已解析的关系
- 连词
- ……



特征含义

$$s1.w = \text{good} \wedge s1.t = \text{JJ}$$
$$s2.w = \text{has} \wedge s2.t = \text{VBZ} \wedge s1.w = \text{good}$$
$$lc(s_2).t = \text{PRP} \wedge s_2.t = \text{VBZ} \wedge s_1.t = \text{JJ}$$
$$lc(s_2).w = \text{He} \wedge lc(s_2).l = \text{nsubj} \wedge s_2.w = \text{has}$$

稀疏
高维：10^6–10^7
费时：特征计算时间占比95%

深度学习

- Chen & Manning 2014
  - 特征表示维度低：约1000维
  - 特征为连续值：不稀疏

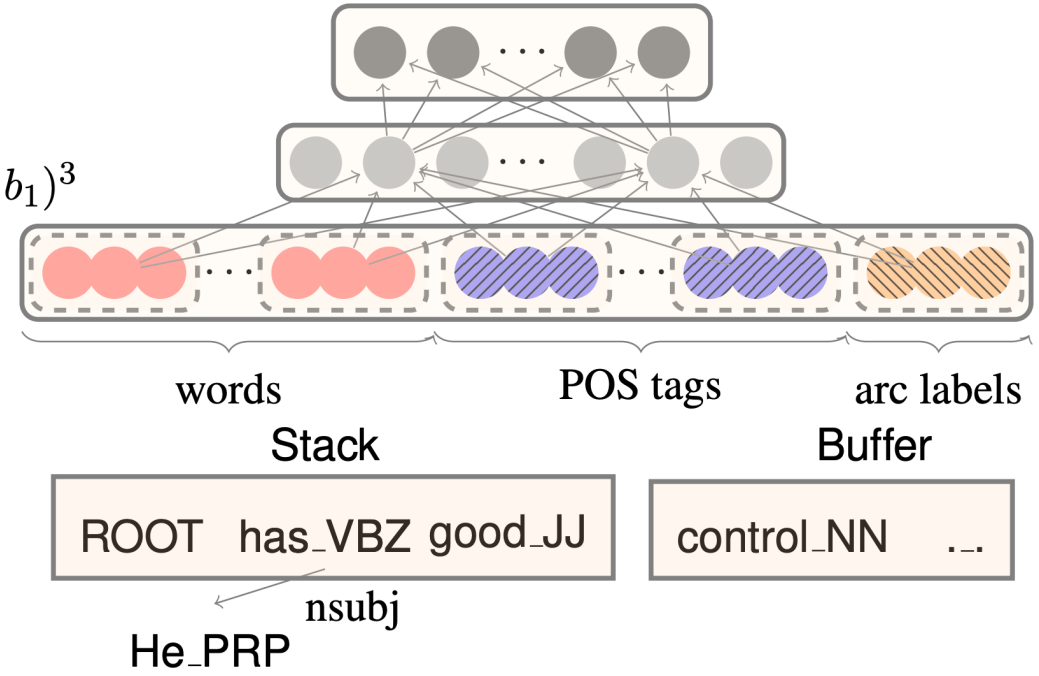
Softmax layer:

$$p = \texttt{softmax}(W_2 h)$$

Hidden layer:

$$h = (W_1^w x^w + W_1^t x^t + W_1^l x^l + b_1)^3$$

Input layer: $[x^w, x^t, x^l]$



Configuration

深度学习

- Chen & Manning 2014：性能

Parser	UAS	LAS	sent. / s
MaltParser	89.8	87.2	469
MSTParser	91.4	88.1	10
TurboParser	**92.3**	89.6	8
C & M 2014	92.0	**89.7**	**654**

错误传递：之前的某一步错误，会影响后续整个解析结果

04

评价方法

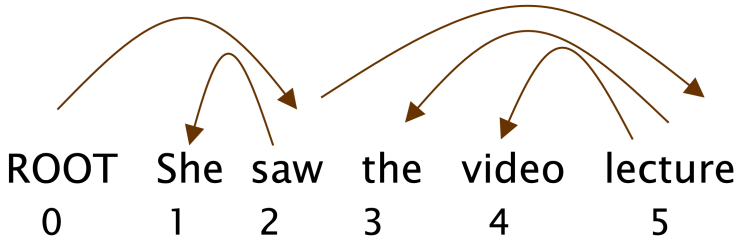
EVALUATION

评价方法

- 完全匹配
  - 将句法分析器得到的树结构与人工标注（树库）的树结构进行完成匹配
  - 指标很低：一棵树只要一条边不正确，整棵树会被判断预测错误
  - 关键的边预测正确即可
- 部分匹配
  - 预测正确的边数相对于标注边数的占比
  - UAS：unlabeled attachment score（不考虑边类型）
  - LAS：labeled attachment score（考虑边类型）

评价指标

- UAS：unlabeled attachment score
  - 预测正确的边数相对于标注边数的占比（不考虑边类型）



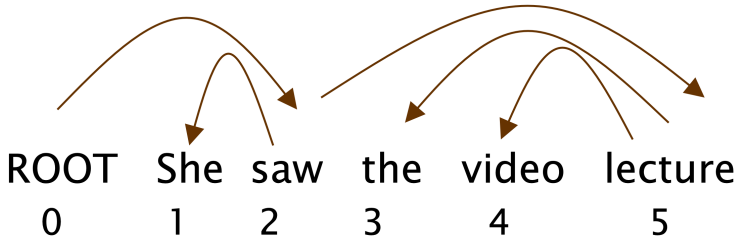
$$UAS = \frac{\#\ correct\ deps}{\#\ deps} = \frac{4}{5} = 80\%$$

Gold			
1	2	She	nsubj
2	0	saw	root
3	5	the	det
4	5	video	nn
5	2	lecture	obj

Parsed			
1	2	She	nsubj
2	0	saw	root
3	4	the	det
4	5	video	nsubj
5	2	lecture	ccomp

评价指标

- LAS：labeled attachment score
  - 预测正确的边数相对于标注边数的占比（考虑边类型）



$$LAS = \frac{\#\ correct\ deps\ with\ type}{\#\ deps} = \frac{2}{5} = 40\%$$

Gold

1	2	She	nsubj
2	0	saw	root
3	5	the	det
4	5	video	nn
5	2	lecture	obj

Parsed

1	2	She	nsubj
2	0	saw	root
3	4	the	det
4	5	video	nsubj
5	2	lecture	ccomp

REFERENCE

- https://web.stanford.edu/class/cs224n/slides/cs224n-2023-lecture04-dep-parsing.pdf
- https://www.cc.gatech.edu/classes/AY2020/cs7650_spring/
- 《Speech and Language Processing》, Chapter 8. Daniel Jurafsky & James H. Martin.

南京大学
NANJING UNIVERSITY

Thank you！
Q&A