

第 11 章 句 法 分 析

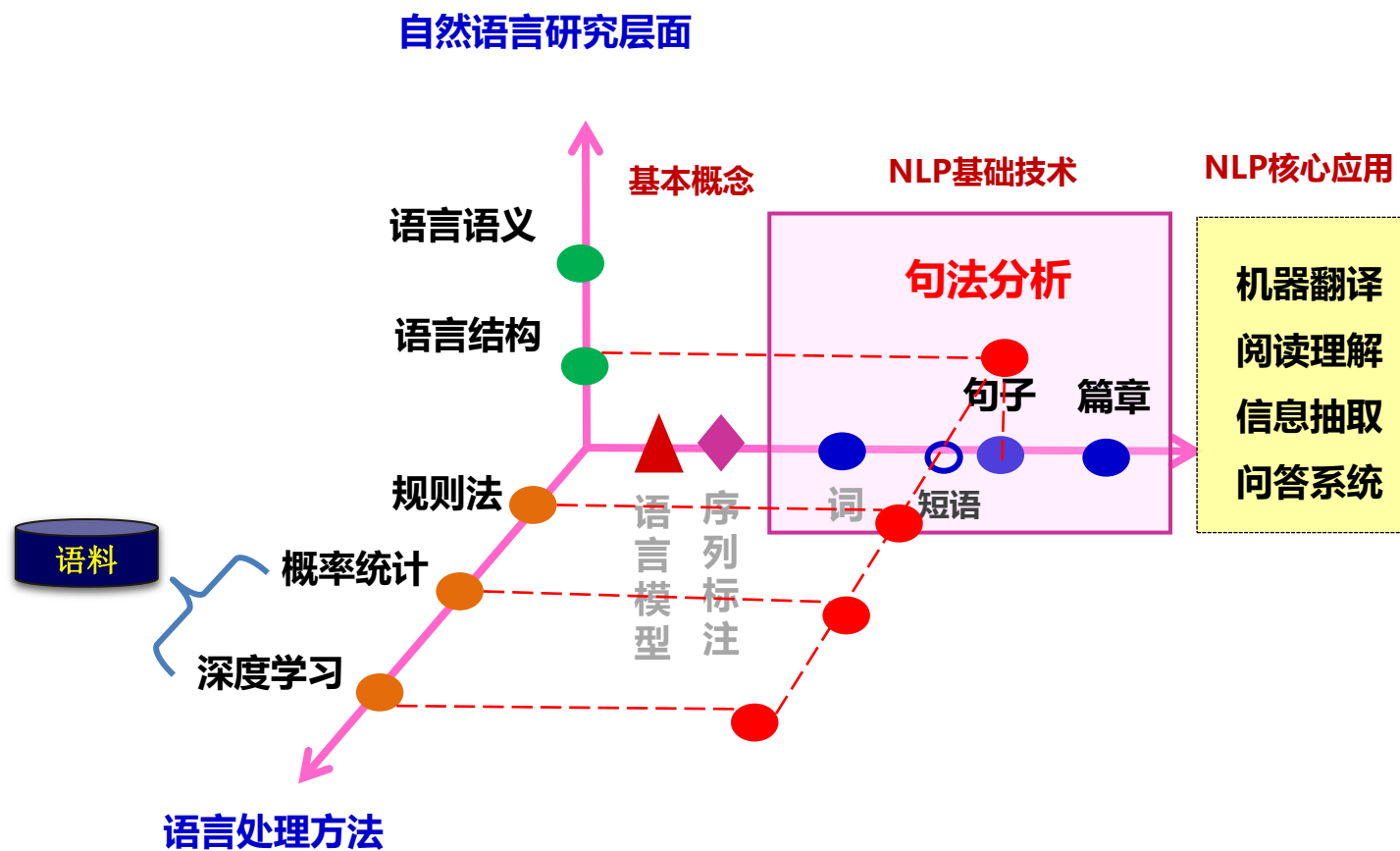
中科院信息工程研究所第二研究室

胡玥

huyue@iie.ac.cn

自然语言处理课程内容及安排

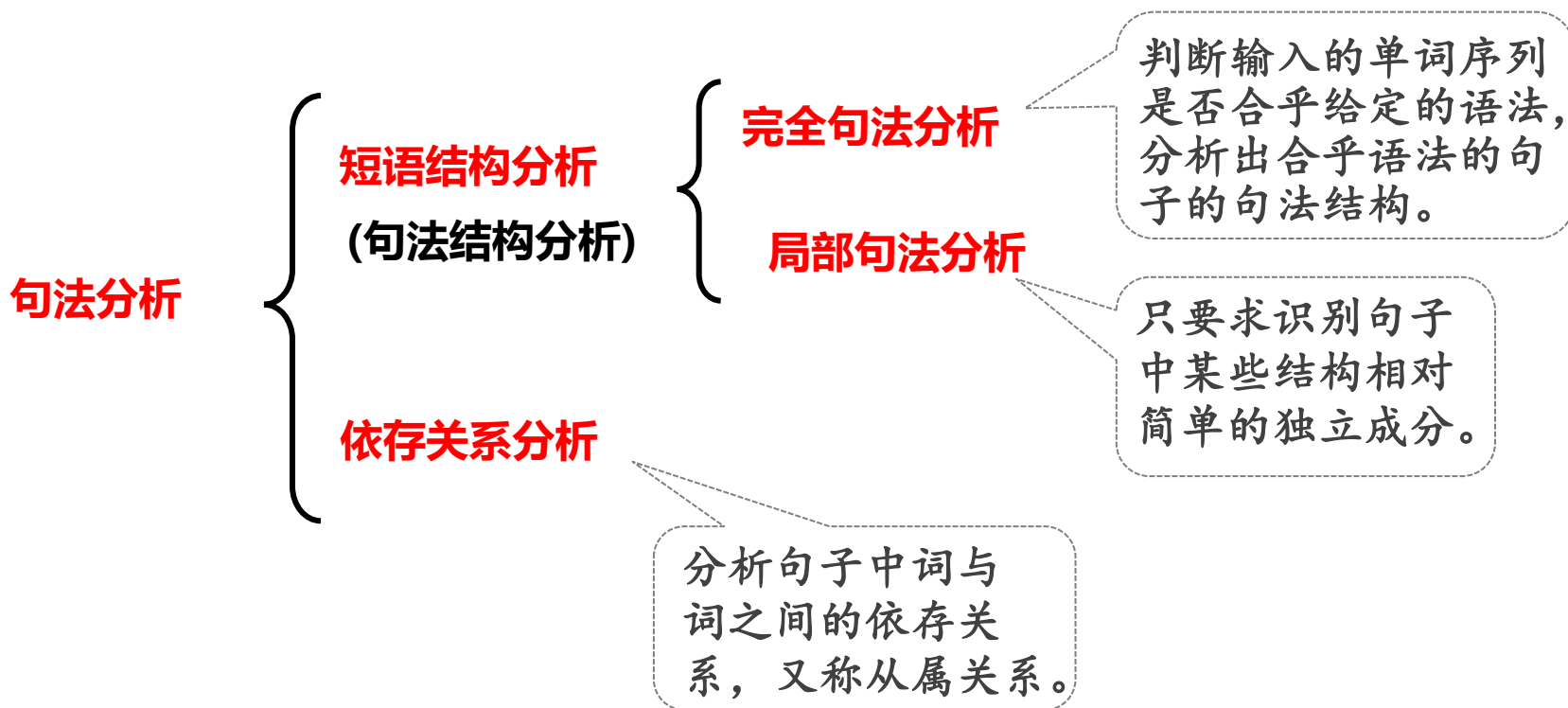
◇ 课程内容：



句法分析概述

句法分析(syntactic parsing) :

任务是确定句子的句法**结构**或句子中词汇之间的**依存关系**



内 容 提 要

第一部分：完全句法分析

第二部分：局部句法分析

第三部分：依存关系分析

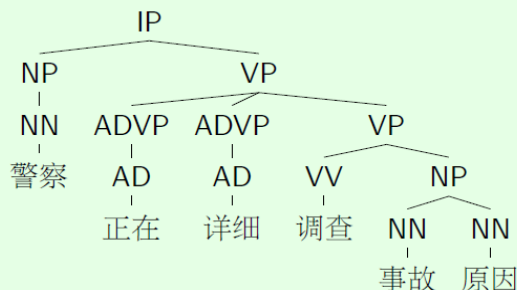
完全句法分析概述

完全句法分析任务

输入： 警察/NN 正在/AD 详细/AD 调查/VV 事故/NN 原因/NN

层次分析法（语言-句法）

输出： 短语结构树



规则法

概率统计

深度学习

技术方法

规则法

Chomsky形式文法
句法分析算法

深度学习

递归神经网络
神经网络完全句法分析

概率统计

概率上下文无关文法
概率句法分析算法

完全句法分析-内容提 要

11.1.1 层次分析法

11.1.2 规则法完全句法分析

11.1.3 概率统计法完全句法分析

11.1.4 神经网络法完全句法分析

11.1.5 句法分析评价

11.1.1 层次分析法

层次分析法（语言学-句子结构层面分析法）

1. 把句子划分为 主语 谓语 宾语 定语 状语 补语 六个成分

(1) 主语

主语是说话人所要陈述的对象，在汉语里主语一般可以理解为话题。

（1）弟弟在看电影。（2）信寄走了。

(2) 谓语

谓语是陈述主语的，是句子结构和语义解释的核心，抽出它，句子就散架了

（1）我读书（2）我吃饭

11.1.1 层次分析法

(3) 宾语

宾语跟动语相对应的，是动语后边表示人物或事件的成分。

(1) 他洗衣服 (2) 你写字

(4) 定语

给人、事物的性质、状态分类或划定范围。

(1) 绍兴老酒 (2) 金灿灿的阳光

(5) 状语

动词性和形容词性词语前边起修饰作用的成分。

(1) 非常高兴 (2) 少抽烟

(6) 补语

动词和形容词后面的补充说明成分。

(1) 他讲完了 (2) 他讲得深刻动人

11.1.1 层次分析法

2. 词、词组作为划分成分的基本单位
3. 根据六个成分的搭配排列按层次顺序确定句子的格局。

一般以树结构表示结果（短语结构分析--句法分析树）

11.1.1 层次分析法

例1：我弟弟已经准备好了一切用品。

分析的时候，往往找出主语和谓语作为句子的主干，以其他成分作为枝叶，描述整个句子的结构。

层次分析：

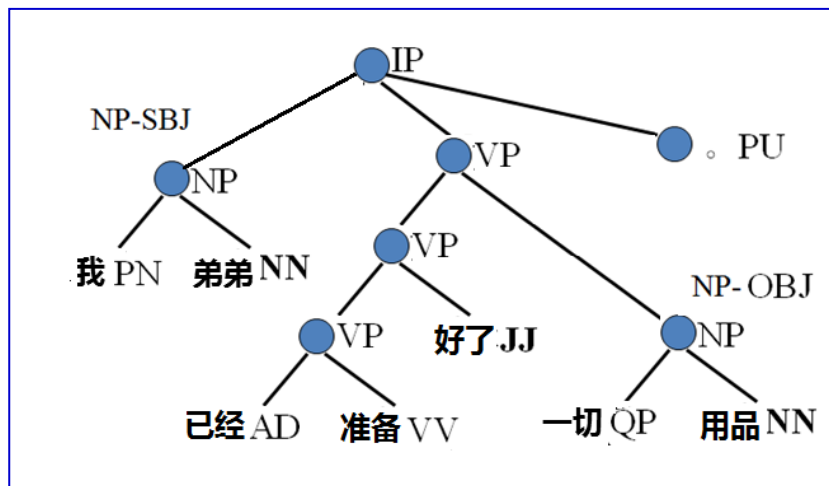
我 弟弟	已经 准备 好了	一切 用品
(主语)	(谓语)	
我 弟弟 (定语) (主语)	准备 好了 (谓语) (补语)	(宾语)
	已经 准备 (状语) (谓语)	一切 用品 (定语) (宾语)

11.1.1 层次分析法

层次分析结果：

我 弟弟	已经 准备 好了	一切 用品
(主语)	(谓语)	
我 弟弟 (定语) (主语)	准备 好了 (谓语) (补语)	(宾语)
	已经 准备 (状语) (谓语)	一切 用品 (定语) (宾语)

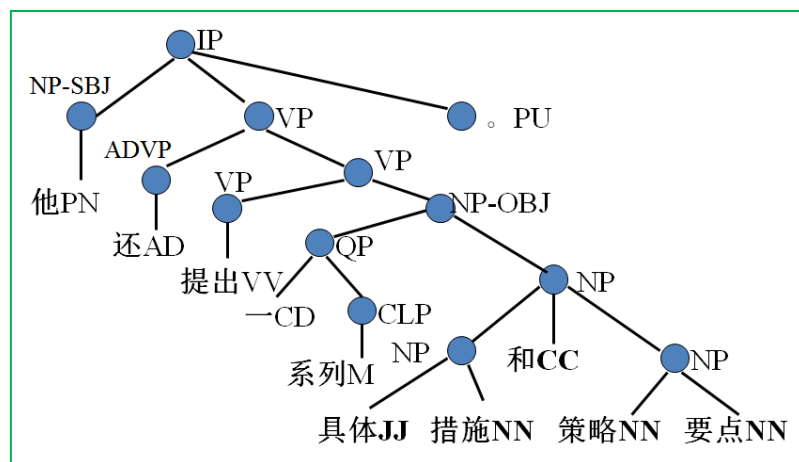
句法分析树



11.1.1 层次分析法

例2：他还提出一系列具体措施和策略要点。

句法分析树



||

括号层次表示

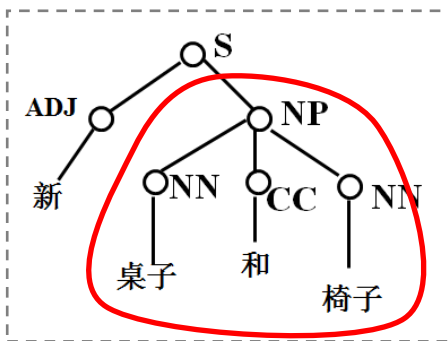
```
( IP (NP-SBJ (PN 他 ))
  (VP (ADVP ( AD 还 ))
    (VP (VV 提出 ))
      (NP-OBJ(QP (CD 一)
        (CLP ( M 系列 )))
        (NP (NP(ADJP ( JJ 具体)
          (NP (NN 措施))))
          (CC 和)
          (NP ( NN 政策)
            (NN 要点 )))))
      (PU 。 ))
```

11.1.1 层次分析法

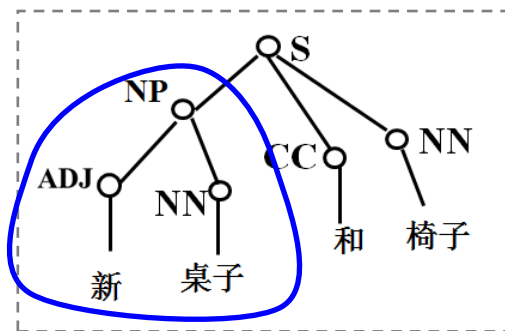
层次结构句法分析问题：

层次分析法枝干分明，便于归纳句型。但会遇到大量歧义问题。

如：新桌子和椅子

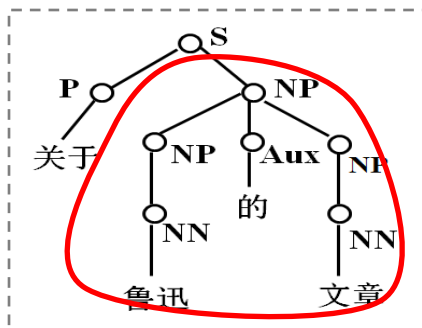


短语派树 - (1)

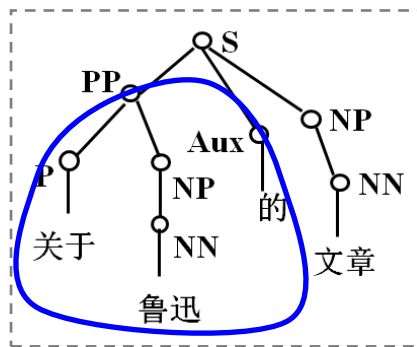


短语派生树 - (2)

关于鲁迅的文章



短语派树 - (1)

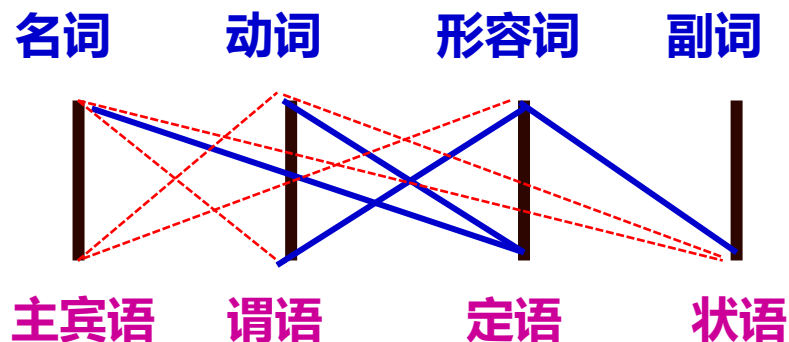


短语派生树 - (2)

11.1.1 层次分析法

层次结构句法分析困难：

(1) 汉语里，词类跟句法成分之间的关系比较复杂，除了副词只能作状语，属于一对一之外，其余的都是一对多，即一种词类可以作多种句法成分。



黑线的是表示该词类的主要功能，划蓝线的是次要功能，划红线的是局部功能

11.1.1 层次分析法

(2) 词存在兼类，词类的多功能---词类与句子成分不存在一一对应关系

如：“会(会议，名词)、会(能够、动词)”

每次他都**会**在**会**上制造点新闻。

(3) 短语存在多义(歧义)，各类短语与句子成分不存在一一对应关系

A.发现敌人的哨兵

定) 中

动 | 宾

B.发现敌人的哨兵

动 | 宾

定) 中

A.看打乒乓球的孩子

动 | 宾

定) 中

动 | 宾

B.看打乒乓球的孩子

定) 中

动 | 宾

动 | 宾

完全句法分析-内容提 要

11. 1. 1 层次分析法

11. 1. 2 规则法完全句法分析

11. 1. 2. 1 Chomsky形式文法

11. 1. 2. 2 自然语言形式文法

11. 1. 2. 3 句法分析算法

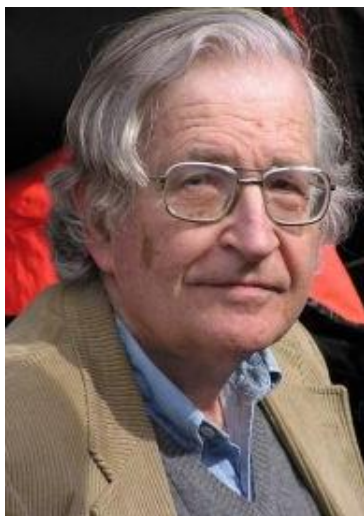
11. 1. 3 概率统计法完全句法分析

11. 1. 4 神经网络法完全句法分析

11. 1. 5 句法分析评价

11.1.2.1 Chomsky的形式文法

Chomsky的形式语言



Chomsky

自诞生之日起至今，历经古典理论、标准理论、扩充式标准理论、管辖约束理论和最简理论五个阶段的发展变化。它在语言学界产生了重大影响，被誉为—场Chomsky式的革命，其影响力波及到语言学之外的心理学、哲学、教育学、逻辑学、翻译理论、通讯技术、计算机科学等领域。

11.1.2.1 Chomsky的形式文法

Chomsky文法：

用 G 表示形式语法， G 定义为四元组：

$$G = (V_n, V_t, S, P)$$

V_n ：非终结符(non-terminals)的有限集合，不能处于生成过程的终点，即在实际句子中不出现。在推导中起变量作用，相当于语言中的语法范畴。

V_t ：终结符(terminals)的有限集合，只处于生成过程的终点，是句子中实际出现的符号，相当于单词表。

S ： V_n 中的初始符号，相当于语法范畴中的句子。

P ：重写规则(rewriting rules)，又称生成规则(production rules)，一般形式为 $\alpha \rightarrow \beta$ ，其中 α 和 β 都是符号串， α 至少含有 V_n 中的一个符号。

- 语法 G 的**不含非终结符**的句子形式称为 **G 生成的句子**。
- 由语法 G 生成的语言，记做 $L(G)$ ，指 G 生成的**所有句子的集合**。

11.1.2.1 Chomsky的形式文法

例1： 设有一种语言

{ ab , aab , aaab , aaaab... }

如何表示？

解： 采用文法方法 → 生成方式

终结符： { a,b }

非终结符： A , B, S

生成规则P： $S \rightarrow aA$ $A \rightarrow aA$; $A \rightarrow b$;

$$G = (\{ A \}, \{ a, b \}, S, P)$$
$$P : S \rightarrow aA \quad A \rightarrow aA \mid b$$

用有限的规则表示
(生成) 无限的语句

11.1.2.1 Chomsky的形式文法

例2： 设有一种语言

{ 001, 0011, 00111, 00111 1... }

如何表示？

解： 采用文法方法 → 生成方式

终结符： { 0,1 }

非终结符： A, S

生成规则P： $A \rightarrow 0$; $A1 \rightarrow A11$; $S \rightarrow 0A1$

$$G = (\{ A \}, \{ 0, 1 \}, S, P)$$
$$P : S \rightarrow 0A1 \quad A1 \rightarrow A11 \quad A \rightarrow 0$$

11.1.2.1 Chomsky的形式文法

例2： 设有一种语言

$\{ 001, 0011, 00111, 001111 \dots \}$

如何表示？

解： 采用文法方法 \rightarrow 生成方式

终结符： $\{ 0, 1 \}$

非终结符： A, S

生成规则P： $A \rightarrow 0$; $A1 \rightarrow A11$;

如, $S \rightarrow A$? $L = \{ 0 \}$

如, $S \rightarrow A1$? $\{ 01, 011, 0111, 01111 \dots \}$

11.1.2.1 Chomsky的形式文法

Chomsky根据重写规则的形式，把形式语法分为4级：

0型文法（无约束文法）

1型文法(上下文有关文法)

2型文法(上下文无关文法)

3型文法(正则文法)

11.1.2.1 Chomsky的形式文法

0 型文法(无约束文法)：

重写规则为 $\alpha \rightarrow \beta$, 其中 $\alpha, \beta \in (V_n \cup V_t)^*$ 。该文法对规则形式没有任何限制, 因此也称为**无约束文法或无限制重写文法**。

V^* 是符号串集合 V 的闭包, 定义为: $V^* = V^0 \cup V^1 \cup V^2 \dots$

例如：

$$V = \{a, b\}$$

$$V^0 = \{\epsilon\}, V^1 = V, V^2 = \{aa, ab, bb, ba\}$$

$$V^* = \{\epsilon, a, b, aa, ab, bb, ba, aaa, \dots\}$$

11.1.2.1 Chomsky的形式文法

1型文法（上下文有关文法）

重写规则为 $\alpha A \beta \rightarrow \alpha \gamma \beta$, 其中 $A \in V_n$, $\alpha, \beta, \gamma \in (V_n \cup V_t)^*$

γ 不为空。在上下文 $\alpha - \beta$ 中, 单个的非终结符 A 被重写为符号串 γ , 因此是上下文敏感的。

例 : $G = (N, \Sigma, P, S)$

$N = \{S, A, B, C\},$

$\Sigma = \{a, b, c\},$

P: (a) $S \rightarrow A B C$

(b) $A \rightarrow a A \mid a$

(c) $B \rightarrow b B \mid b$

(d) $B C \rightarrow B c c$

$L(G) = ? \quad \{a^n b^m c^2\}, n \geq 1, m \geq 1$

11.1.2.1 Chomsky的形式文法

2型文法 (上下文无关文法CFG)

重写规则为 $A \rightarrow \alpha$, 其中 $A \in V_n$, $\alpha \in (V_n \cup V_t)^*$, A 重写为 α 时没有上下文限制。上下文无关文法。

例 $G = (N, \Sigma, P, S)$,

$$N = \{S, A, B, C\}, \quad \Sigma = \{a, b, c\},$$

$$P: (a) S \rightarrow A B C \quad (b) A \rightarrow a A \mid a$$

$$(c) B \rightarrow b B \mid b \quad (d) C \rightarrow B A \mid c$$

$$L(G) = ? \quad \{a^n b^m a^k c^\alpha\}, n \geq 1, m \geq 1, k \geq 0, \alpha \in \{0, 1\}$$

(如果 $k = 0$ 的话 , $\alpha = 1$, 否则 , $\alpha = 0$ 。)

11.1.2.1 Chomsky的形式文法

3型文法 (正则文法RG)

重写规则为 $A \rightarrow Bx$, 或 $A \rightarrow x$, 其中 $A, B \in V_n$ $x \in V_t$; $A \rightarrow x$ 是 $A \rightarrow Bx$ 中当 B 为空符号时的一种特殊情况。**正则文法又称3型文法。**

如果把 A 和 B 看作不同的状态, 那么由重写规则可知, 由状态 A 转入状态 B 时, 可生成一个终结符 x , 因此正则文法也称作有限状态文法(finite state grammar)。

上述定义的是左线性正则文法, 如果 $A \rightarrow xB$ 则是右线性正则文法。

11.1.2.1 Chomsky的形式文法

例

$$G = (N, \Sigma, P, S),$$

$$N = \{S, A, B\},$$

$$\Sigma = \{a, b\},$$

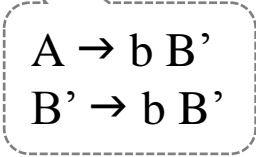
$$P: (a) S \rightarrow a A$$

$$(b) A \rightarrow a A$$

$$(c) A \rightarrow b b B$$

$$(d) B \rightarrow b B$$

$$(e) B \rightarrow b$$


$$\begin{array}{l} A \rightarrow b B' \\ B' \rightarrow b B' \end{array}$$

$$L(G) = ? \quad \{a^n b^m\}, n \geq 1, m \geq 3$$

11.1.2.1 Chomsky的形式文法

例1： 设有一种语言

{ ab , aab , aaab , aaaab... }

如何表示？

解： 采用文法方法 → 生成方式

终结符： { a,b }

非终结符： A , B, S

生成规则P： $S \rightarrow aA$ $A \rightarrow aA$; $A \rightarrow b$;

$$G = (\{ A \}, \{ a, b \}, S, P)$$
$$P : S \rightarrow aA \quad A \rightarrow aA \mid b$$

文法是几型文法？

3型文法

11.1.2.1 Chomsky的形式文法

例2： 设有一种语言

{ 001, 0011, 00111, 00111 1... }

如何表示？

解： 采用文法方法 → 生成方式

终结符： { 0,1 }

非终结符： A, S

生成规则P： $A \rightarrow 0$; $A1 \rightarrow A11$; $S \rightarrow 0A1$

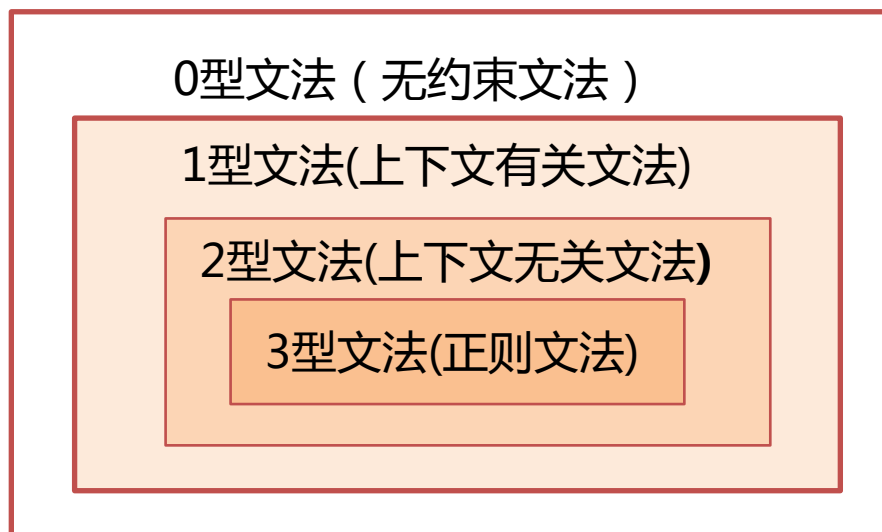
$G = (\{ A \}, \{ 0, 1 \}, S, P)$
 $P : S \rightarrow 0A1 \quad A1 \rightarrow A11 \quad A \rightarrow 0$

文法是几型文法？

1型文法

11.1.2.1 Chomsky的形式文法

各级文法之间关系：



如果一种语言能由几种文法所产生，则把这种语言称为在这几种文法中受限制最多的那种文法所产生的语言。

每一个正则文法都是上下文无关文法，每一个上下文无关文法都是上下文有关文法，每一个上下文有关文法都是0型文法。Chomsky把0型文法生成的语言叫0型语言，把由上下文有关文法、上下文无关文法、正则文法生成的语言分别叫作上下文有关语言、上下文无关语言、正则语言(或有限状态语言)，因此有：

$$L(G_0) \supseteq L(G_1) \supseteq L(G_2) \supseteq L(G_3)$$

完全句法分析-内容提 要

11. 1. 1 层次分析法

11. 1. 2 规则法完全句法分析

11. 1. 2. 1 Chomsky形式文法

11. 1. 2. 2 自然语言形式文法

11. 1. 2. 3 句法分析算法

11. 1. 3 概率统计法完全句法分析

11. 1. 4 神经网络法完全句法分析

11. 1. 5 句法分析评价

11.1.2.2 自然语言形式文法

自然语言的形式文法

在自然语言处理中采用Chomsky形式文法作为刻画语言规律、表示语言的形式文法

- 用几型文法描述自然语言

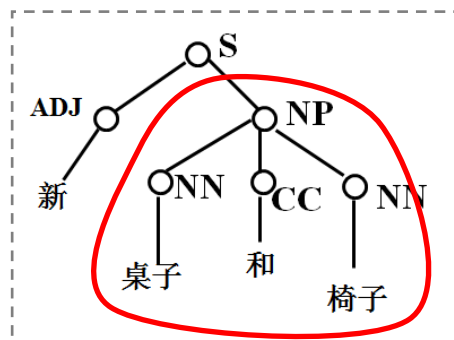
- 从描述能力上，正则文法描述能力太弱，上下文无关文法也不足以表述自然语言，因为自然语言中上下文相关情况非常常见。
- 从计算复杂度上，上下文有关文法的复杂度为NP完全，上下文无关文法的复杂度为多项式，复杂度可以接受。
- 因此，用上下文无关文法来描述自然语言最为普遍，并需要用一些其它手段来增强其描述能力。

11.1.2.2 自然语言形式文法

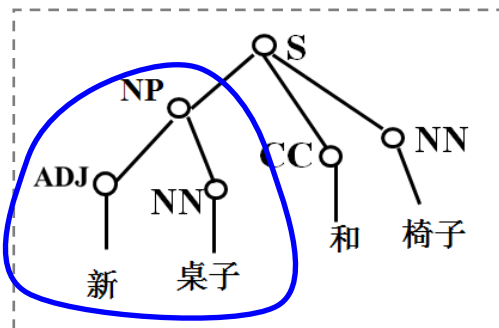
- 上下文无关文法的二义性问题

一个文法 G ，如果存在某个句子有不只一棵分析树与之对应，那么称这个文法是二义的。**上下文无关文法处理语言会产生二义性**

如：新桌子和椅子



短语派树 - (1)



短语派生树 - (2)

11.1.2.2 自然语言形式文法

- Chomsky范式

Chomsky证明，任何的由上下文无关文法生成的语言，均可由重写规则为 $A \rightarrow BC$ 或者 $A \rightarrow x$ 的文法生成，其中 $A, B, C \in V_n$, $x \in V_t$ 具有这样的重写规则的上下文无关文法，它的推导树均可简化为二元形式，这样就可以用二分法来分析自然语言，采用二叉树来表示自然语言的句子结构。上述重写规则称为 **Chomsky范式** (Chomsky normal form)。

11.1.2.2 自然语言形式文法

Chomsky形式语言各部分与自然语言的对应关系

形式语法G定义为四元组：

$$G = (V_n, V_t, S, P)$$

V_n ：非终结符(non-terminals)的有限集合，：

对应语言语法单位（一般为“词”或“词组”的词性）

V_t ：终结符(terminals)的有限集合：

对应语言基本组成单位（一般为“词”）

S ： V_n 中的初始符号，**相当于语法范畴中的句子。**

P ：生成规则(production rules)：

对应语法结构规则（不同的语言有不同的规则）（人工编写）

11.1.2.2 自然语言形式文法

例：设有如下语句：

{ 我吃饭，我洗衣，我看书，我喝水，
你吃饭，你洗衣，你看书，你喝水，
他吃饭，他洗衣，他看书，他喝水 }

如何用形式文法表示这些语句？

解：采用文法方法 → 生成方式

终结符 V_t ： {我，你，他，吃，洗，看，喝，饭，衣，书，水}

非终结符 V_n ： S，IP，PR，VV，NN

生成规则P： $S \rightarrow IP$ ； $IP \rightarrow PR \ VV \ NN$ ；

$PR \rightarrow \text{我}|\text{你}|\text{他}$ $VV \rightarrow \text{吃}|\text{洗}|\text{看}|\text{喝}$ $NN \rightarrow \text{饭}|\text{衣}|\text{书}|\text{水}$

11.1.2.2 自然语言形式文法

例：设有如下语句：

{ 我吃饭，我洗衣，我看书，我喝水，
你吃饭，你洗衣，你看书，你喝水，
他吃饭，他洗衣，他看书，他喝水 }

如何用形式文法表示这些语句？

解：语句表示文法

$G = (V_n, V_t, S, P)$

$V_n = \{S, IP, PR, VV, NN\}$

$V_t = \{\text{我, 你, 他, 吃, 洗, 看, 喝, 饭, 衣, 书, 水}\}$

P:

1. $S \rightarrow IP$
2. $IP \rightarrow PR \ VV \ NN$
3. $PR \rightarrow \text{我} \mid \text{你} \mid \text{他}$
4. $VV \rightarrow \text{吃} \mid \text{洗} \mid \text{看} \mid \text{喝}$
5. $NN \rightarrow \text{饭} \mid \text{衣} \mid \text{书} \mid \text{水}$

文法是几型文法？

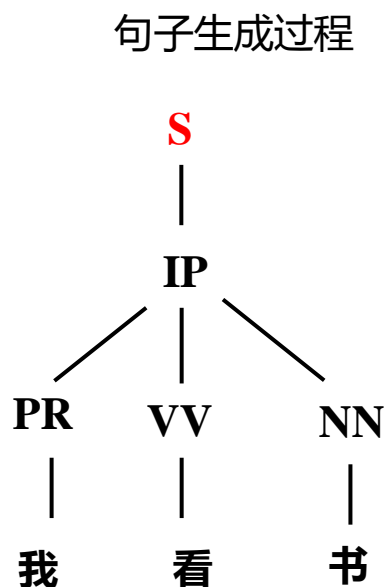
2型文法

11.1.2.2 自然语言形式文法

验证句子是否能由所求文法生成：

{ 我吃饭，我洗衣，我看书，我喝水，
你吃饭，你洗衣，你看书，你喝水，
他吃饭，他洗衣，他看书，他喝水 }

以“我看书”为例：



$G = (V_n, V_t, S, P)$

$V_n = \{S, IP, PR, VV, NN\}$

$V_t = \{我, 你, 他, 吃, 洗, 看, 喝, 饭, 衣, 书, 水\}$

$P :$

1. $S \rightarrow IP$
2. $IP \rightarrow PR \ VV \ NN$
3. $PR \rightarrow 我 \mid 你 \mid 他$
4. $VV \rightarrow 吃 \mid 洗 \mid 看 \mid 喝$
5. $NN \rightarrow 饭 \mid 衣 \mid 书 \mid 水$

所使用的规则

1
2
3 4 5

11.1.2.2 自然语言形式文法

例：设有如下语句：

{ 我吃饭，我洗衣，我看书，我喝水，
你吃饭，你洗衣，你看书，你喝水，
他吃饭，他洗衣，他看书，他喝水 }

如何用形式文法表示这些语句？

解：语句表示文法

$G = (V_n, V_t, S, P)$
 $V_n = \{S, IP, PR, VV, NN\}$
 $V_t = \{\text{我, 你, 他, 吃, 洗, 看, 喝, 饭, 衣, 书, 水}\}$
 $P:$

1. $S \rightarrow IP$
2. $IP \rightarrow PR \ VV \ NN$
3. $PR \rightarrow \text{我} \mid \text{你} \mid \text{他}$
4. $VV \rightarrow \text{吃} \mid \text{洗} \mid \text{看} \mid \text{喝}$
5. $NN \rightarrow \text{饭} \mid \text{衣} \mid \text{书} \mid \text{水}$

如想表示 “我看你喝水” ？

$IP \rightarrow PR \ VV \ NN \mid PR \ VV \ IP;$

11.1.2.2 自然语言形式文法

例：设有如下语句：

{ 我吃饭，我洗衣，我看书，我喝水，我看你喝水，我看他喝水
你吃饭，你洗衣，你看书，你喝水，我看你看书，我看你吃书
他吃饭，他洗衣，他看书，他喝水，我看你喝水，他看我吃饭.....}

如何用形式文法表示这些语句？

解：语句表示文法

$G = (V_n, V_t, S, P)$

$V_n = \{S, IP, PR, VV, NN\}$

$V_t = \{\text{我, 你, 他, 吃, 洗, 看, 喝, 饭, 衣, 书, 水}\}$

$S = S$

P: 1. $S \rightarrow IP$

2. $IP \rightarrow PR \ VV \ NN$

3. $IP \rightarrow PR \ VV \ IP$

4. $PR \rightarrow \text{我} \mid \text{你} \mid \text{他}$

5. $VV \rightarrow \text{吃} \mid \text{洗} \mid \text{看} \mid \text{喝}$

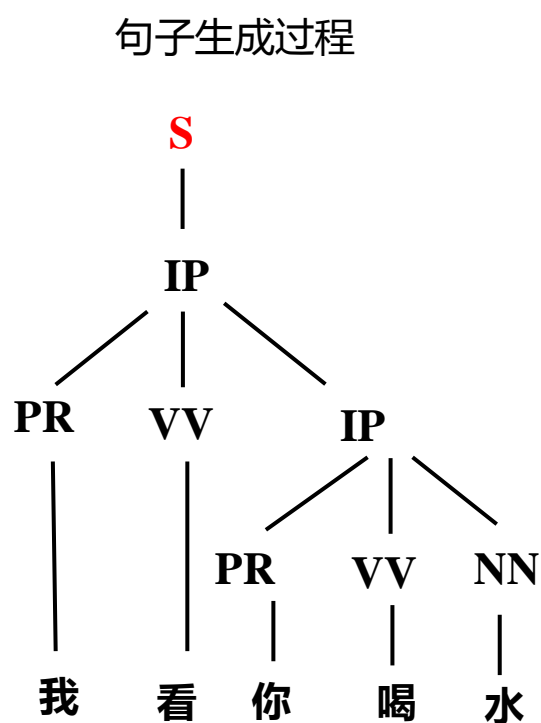
6. $NN \rightarrow \text{饭} \mid \text{衣} \mid \text{书} \mid \text{水}$

11.1.2.2 自然语言形式文法

验证句子是否能由所求文法生成：

{ 我吃饭, 我洗衣, 我看书, 我喝水, 我看你喝水, 我看他喝水
你吃饭, 你洗衣, 你看书, 你喝水, 我看你看书, 我看你吃书
他吃饭, 他洗衣, 他看书, 他喝水, 我看你喝书, 他看我吃饭.....}

以“我看你喝水”为例



所使用的规则

1
3
4 5 2
4 5 6

$G = (V_n, V_t, S, P)$
 $V_n = \{S, IP, PR, VV, NN\}$
 $V_t = \{我, 你, 他, 吃, 洗, 看, 喝, 饭, 衣, 书, 水\}$
 $S = S$
 $P :$ 1. $S \rightarrow IP$
2. $IP \rightarrow PR VV NN$
3. $IP \rightarrow PR VV IP$
4. $PR \rightarrow 我 | 你 | 他$
5. $VV \rightarrow 吃 | 洗 | 看 | 喝$
6. $NN \rightarrow 饭 | 衣 | 书 | 水$

“我去公园运动” ✗

11.1.2.2 自然语言形式文法

句子分析过程是生成过程的逆过程，由于形式文法中生成规则是根据语法规则制定，所以在分析句子是否由某文法产生的同时就等同于对句子进行语法结构分析。

完全句法分析-内容提 要

11. 1. 1 层次分析法

11. 1. 2 规则法完全句法分析

11. 1. 2. 1 Chomsky形式文法

11. 1. 2. 2 自然语言形式文法

11. 1. 2. 3 句法分析算法

11. 1. 3 概率统计法完全句法分析

11. 1. 4 神经网络法完全句法分析

11. 1. 5 句法分析评价

11.1.2.3 句法分析算法

分析算法有三种策略：

- 自底向上 (Bottom-up)
- 从上到下 (Top-down)
- 从上到下和从下到上结合

1. 自顶向下

从符号S 开始搜索，用每条产生式**右边**的符号来改写**左边**的符号，然后通过不同的方式搜索并改写非终结符，直到生成了输入的句子或者遍历了所有可能的句子为止。

2. 自底向上

从句子中的词语开始，基本操作是将一个符号序列匹配归约为其产生式的左部（用每条产生式**左边**的符号来改写**右边**的符号），逐渐减少符号序列直到只剩下开始符S 为止。

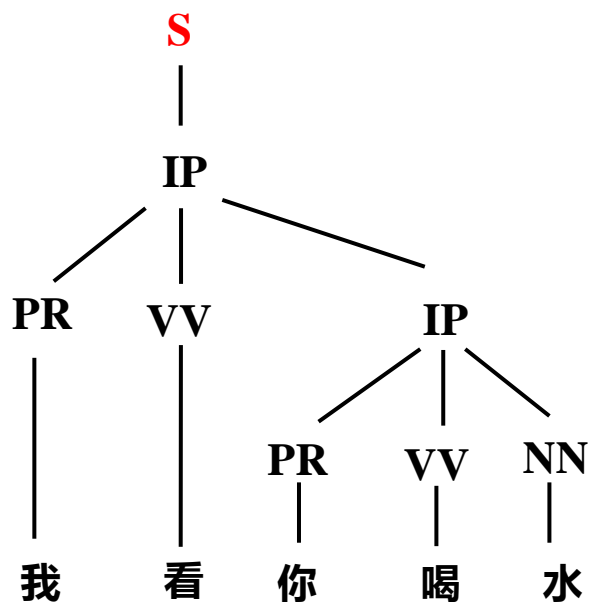
11.1.2.3 句法分析算法

自底向上分析法

从句子中的词语开始用每条产生式左边的符号来改写右边的符号。

句子：我看你喝水

句法分析树



文法：

$$G = (V_n, V_t, S, P)$$

$$V_n = \{ S, IP, PR, VV, NN \}$$

$$V_t = \{ \text{我, 你, 他, 吃, 洗, 看, 喝, 饭, 衣, 书, 水} \}$$

$$P: 1. S \rightarrow IP$$

$$2. IP \rightarrow PR \ VV \ NN$$

$$3. IP \rightarrow PR \ VV \ IP$$

$$4. PR \rightarrow \text{我} \mid \text{你} \mid \text{他}$$

$$5. VV \rightarrow \text{吃} \mid \text{洗} \mid \text{看} \mid \text{喝}$$

$$6. NN \rightarrow \text{饭} \mid \text{衣} \mid \text{书} \mid \text{水}$$

所使用的规则

1

3

3

4 5 4 5 6

11.1.2.3 句法分析算法

句法分析算法

前人做了大量工作提出许多句法分析算法：

- 厄尔利 (Earley) 分析算法、富田胜 (Tomida) 分析算法、
线图 (Chart) 分析算法、CYK 分析算法
- 基于扩充转移网络的分析算法
- 链分析算法等

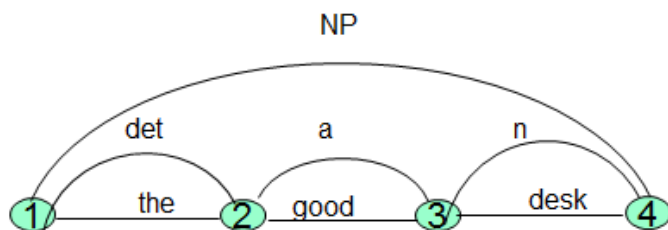
1. 线图分析法 (Chart)

主要特点是可以用chart数据结构来保存以前分析的结果，分析中能够找出所有符合规则的短语结构，对于多义性的句子，可以生成多个语法树。

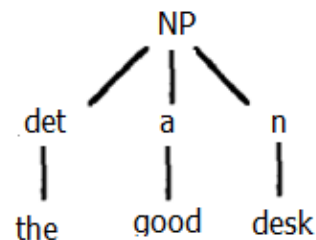
1. 线图分析法 (Chart)

线图与短语结构树

名词短语：The good desk



线图



树型图

线图 是一组节点(node)和边(edge)的集合

节点：对应着输入字符串中的字符间隔

边：<起点, 终点, 标记>，其中标记为非终结符或终结符

如：<2, 3, a>，<1, 4, NP>

1. 线图分析法 (Chart)

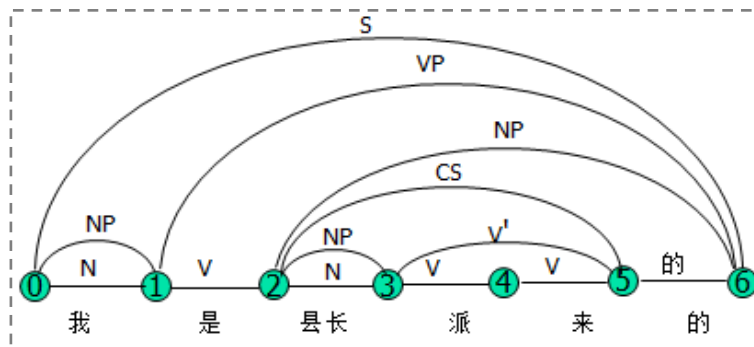
Chart 分析算法：

- 给定一组 CFG 规则: $XP \rightarrow \alpha_1 \dots \alpha_n$ ($n \geq 1$)
- 给定一个输入句子的**词性序列**：
- 从输入串开始，一步步形成chart，使得存在一条边可以覆盖全部节点，并且边上标记为S。

文法规则：

- (1) $S \rightarrow NP VP$
- (2) $NP \rightarrow N$
- (3) $NP \rightarrow CS$ 的
- (4) $VP \rightarrow V NP$
- (5) $CS \rightarrow NP V'$
- (6) $V' \rightarrow V V$

输出：



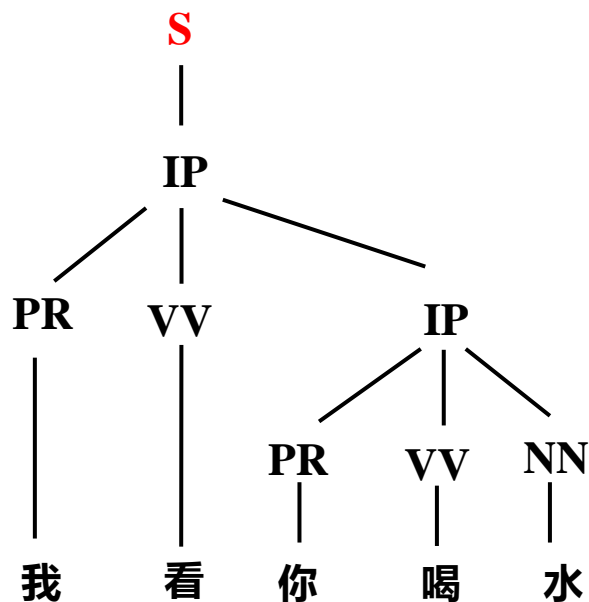
输入： 我 是 县 长 派 来 的
N V N V V 的

1. 线图分析法 (Chart)

句法分析过程

句子：我看你喝水

句法分析树



文法：

$G = (V_n, V_t, S, P)$
 $V_n = \{ S, IP, PR, VV, NN \}$
 $V_t = \{ \text{我, 你, 他, 吃, 洗, 看, 喝, 饭, 衣, 书, 水} \}$
P:
1. $S \rightarrow IP$
2. $IP \rightarrow PR \ VV \ NN$
3. $IP \rightarrow PR \ VV \ IP$
4. $PR \rightarrow \text{我} \mid \text{你} \mid \text{他}$
5. $VV \rightarrow \text{吃} \mid \text{洗} \mid \text{看} \mid \text{喝}$
6. $NN \rightarrow \text{饭} \mid \text{衣} \mid \text{书} \mid \text{水}$

所使用的规则

1
3
3
4 5 4 5 6

分析：匹配过程

匹配过程中，有些规则右部未被完全匹配完，需要后继成分匹配完才能继续匹配，算法中要能保留这些规则，并记录已匹配的位置。

定义：**活动边**表示未被完全匹配完的规则；**点规则**表示已匹配的位置

1. 线图分析法 (Chart)

实现算法需要变量

- **活动边集(ActiveArc)** : 记录那些右端符号串与输入串的某一段相匹配, 但还未完全匹配的重写规则, 通常以数组或列表存储。
- **线图Chart (非活动边)** : 保存分析过程中已经建立的成分(包括终结符和非终结符)、位置(包括起点和终点)。通常以 $n \times n$ 的数组表示(n 为句子包含的词数)。
- **代理表(待处理表)(Agenda)** : 记录刚刚得到的一些重写规则所代表的成分, 这些重写规则的右端符号串与输入词性串(或短语标志串)中的一段完全匹配, 通常以栈或线性队列表示。
- **输入缓冲区** : 输入串

Chart分析算法描述

从输入串的起始位置到最后位置，循环执行如下步骤：

- (1) 如果待处理表(Agenda)为空，则找到下一个位置上的词，将该词对应的(所有)词类 X 附以 (i, j) 作为元素放到待处理表中，即 $X(i, j)$ 。其中， i, j 分别是该词的起始位置和终止位置， $j > i, j - i$ 为该词的长度。
- (2) 从 Agenda 中取出一个元素 $X(i, j)$ 。
- (3) 对于每条规则 $A \rightarrow X\gamma$ ，将 $A \rightarrow X \circ \gamma(i, j)$ 加入活动边集ActiveArc 中，然后调用
扩展弧子程

扩展弧子程序：

- (a) 将 X 插入图表(Chart)的 (i, j) 位置中。
- (b) 对于活动边集(ActiveArc)中每个位置为 (k, i) ($1 \leq k < i$) 的点规则，如果该规则具有如下形式：
 $A \rightarrow \alpha \circ X$ ，如果 $A=S$ ，则把 $S(1, n+1)$ 加入到 Chart 中，并给出一个完整的分析结果；否则，将 $A(k, j)$ 加入到Agenda表中。
- (c) 对于每个位置为 (k, i) 的点规则： $A \rightarrow \alpha \circ X\beta$ ，则将 $A \rightarrow \alpha X \circ \beta(k, j)$ 加入到活动边集

1. 线图分析法 (Chart)

例：输入句子

我 看 你 洗 衣

我 看 你 吃 饭

我 看 你 喝 水

.....

PR VV PR VV NN

文法：

$G = (V_n, V_t, S, P)$

$V_n = \{ S, IP, PR, VV, NN \}$

$V_t = \{ \text{我, 你, 他, 吃, 洗, 看, 喝, 饭, 衣, 书, 水} \}$

P:

1. $S \rightarrow IP$

2. $IP \rightarrow PR \quad VV \quad NN$

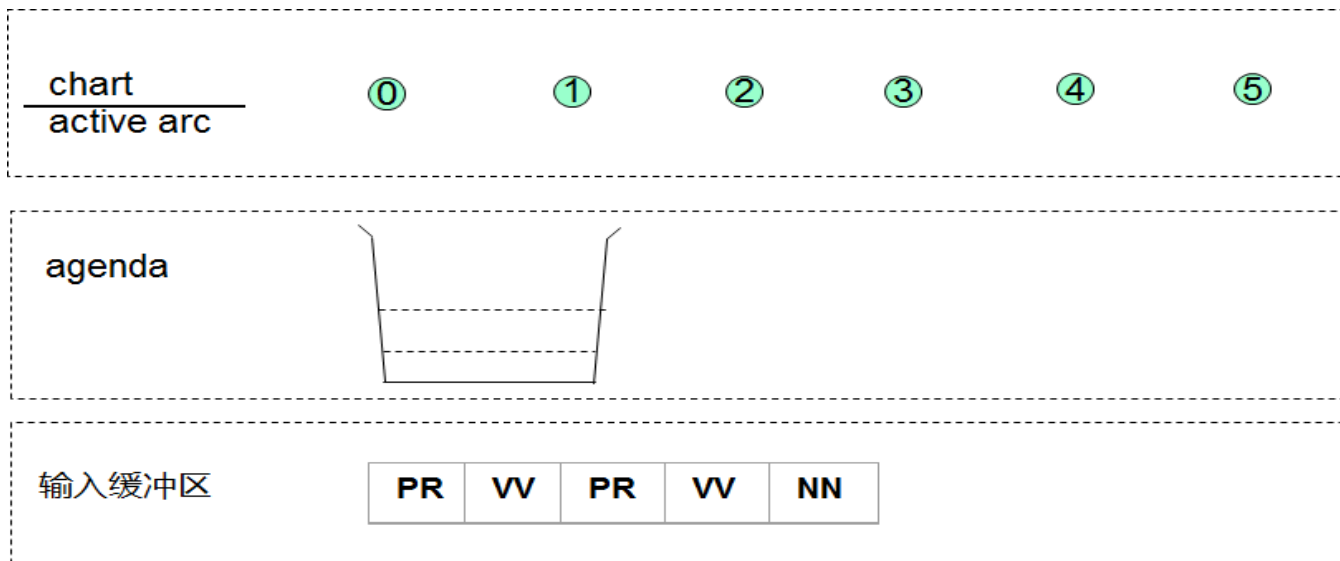
3. $IP \rightarrow PR \quad VV \quad IP$

4. $PR \rightarrow \text{我} \mid \text{你} \mid \text{他}$

5. $VV \rightarrow \text{吃} \mid \text{洗} \mid \text{看} \mid \text{喝}$

6. $NN \rightarrow \text{饭} \mid \text{衣} \mid \text{书} \mid \text{水}$

Chart算法变量



- P:
1. $S \rightarrow IP$
 2. $IP \rightarrow PR \quad VV \quad NN$
 3. $IP \rightarrow PR \quad VV \quad IP$

agenda

0

1

2

3

4

5

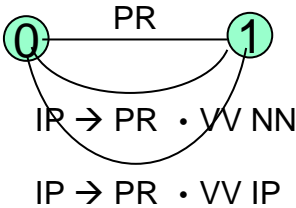
输入缓冲区

PR	VV	PR	VV	NN
----	----	----	----	----

我 看 你 喝 水

c
h
a
r
t
—
a
c
t
i
v
e
a
r
c

(0,1) PR

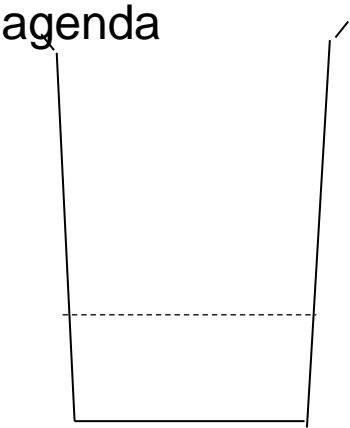
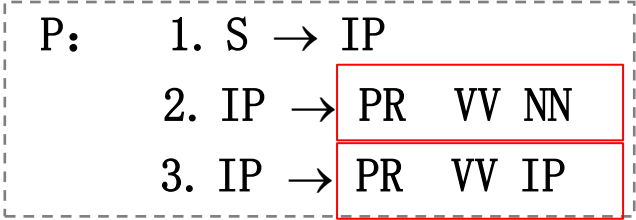


②

③

④

⑤



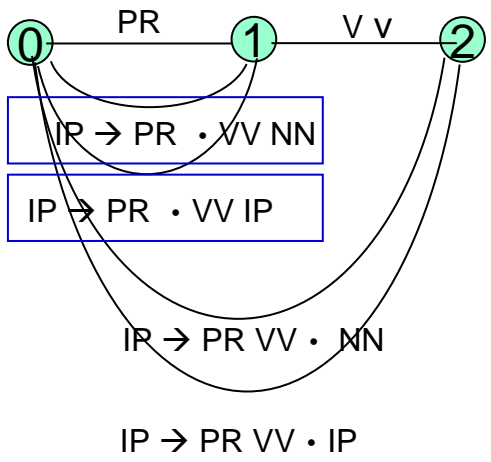
输入缓冲区

VV	PR	VV	NN
看	你	喝	水

- P:
1. $S \rightarrow IP$
 2. $IP \rightarrow PR \quad VV \quad NN$
 3. $IP \rightarrow PR \quad VV \quad IP$

PR **VV**
我 **看**

(1,2) VV



agenda

输入缓冲区

PR	VV	NN
你	喝	水

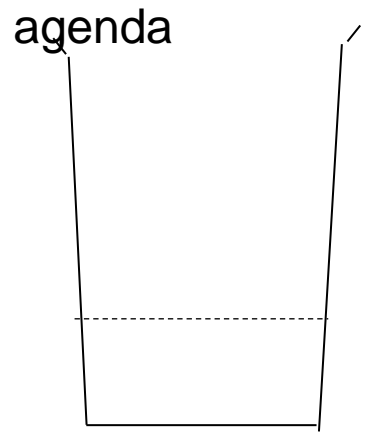
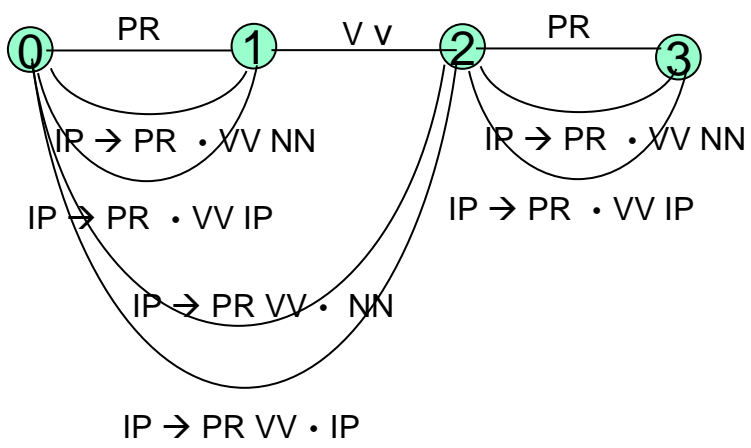
c
h
a
r
t
—
a
c
t
i
v
e
a
r
c

chart
—
active
arc



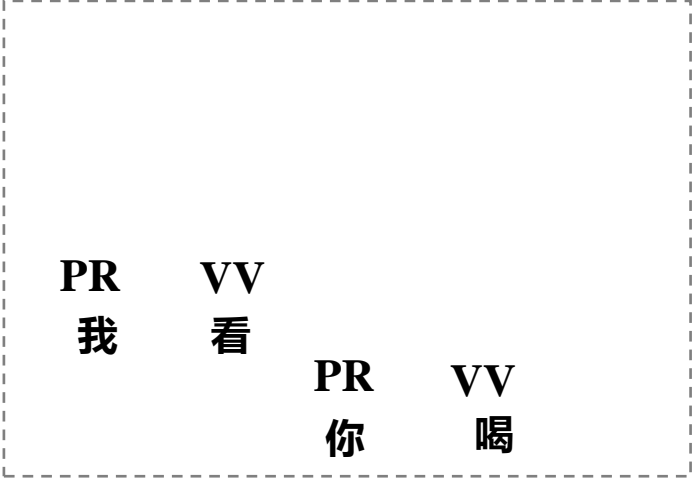
- P:
1. $S \rightarrow IP$
 2. $IP \rightarrow PR \quad VV \quad NN$
 3. $IP \rightarrow PR \quad VV \quad IP$

(2,3) PR



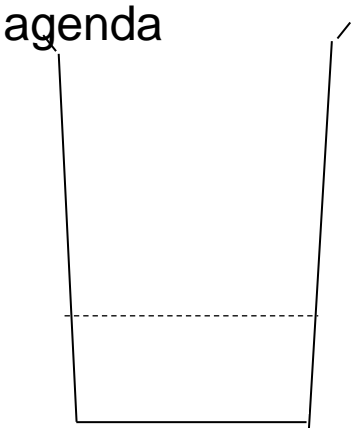
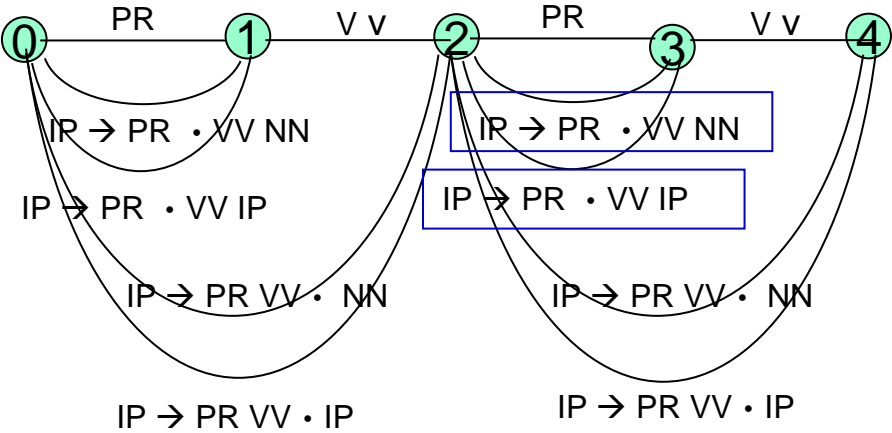
输入缓冲区

VV	NN
喝	水



- P:
- 1. S → IP
 - 2. IP → PR VV NN
 - 3. IP → PR VV IP

(3,4) VV



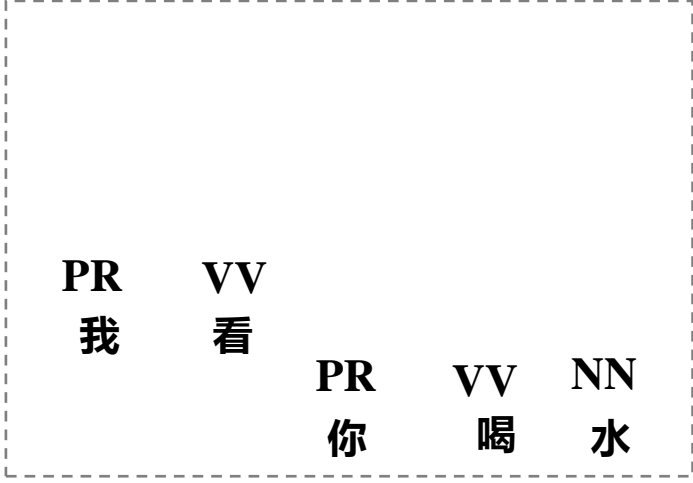
输入缓冲区



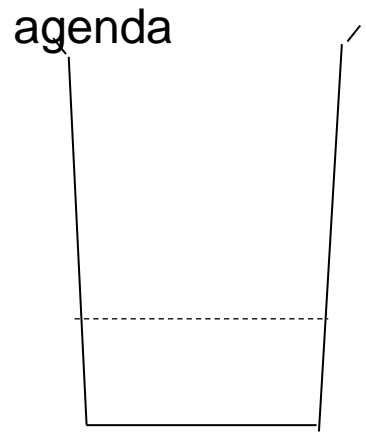
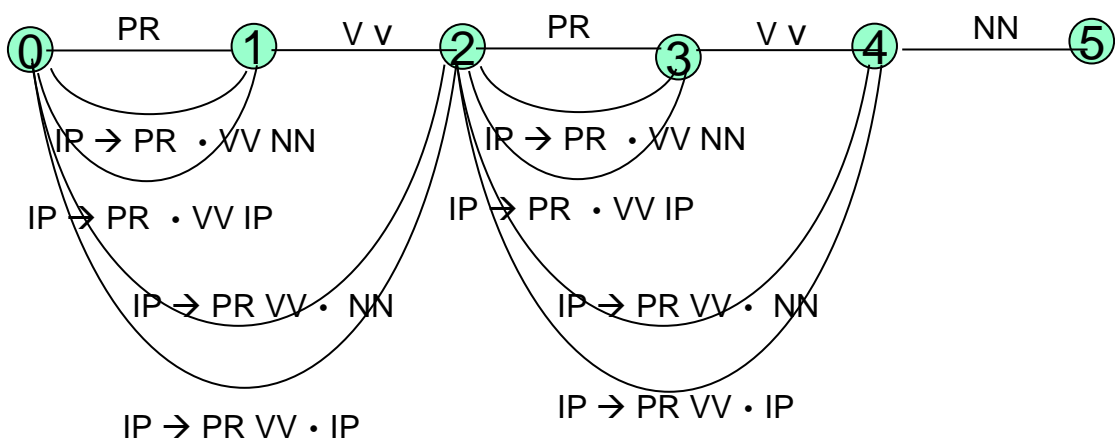
水

chart
—
active
arc

(4,5) NN

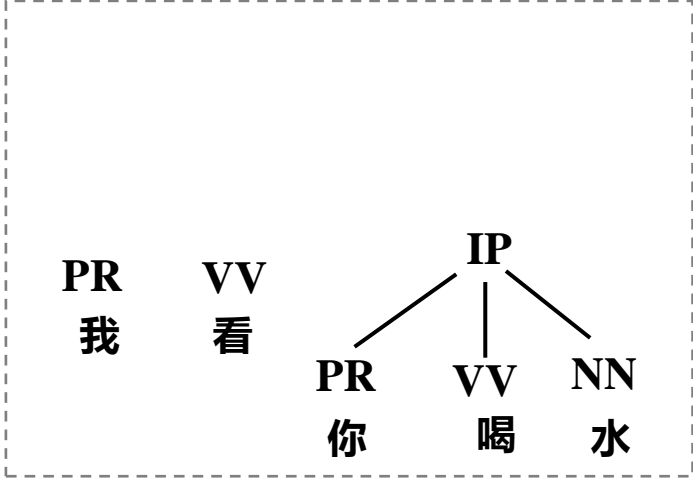


- P:
1. $S \rightarrow IP$
 2. $IP \rightarrow PR \quad VV \quad NN$
 3. $IP \rightarrow PR \quad VV \quad IP$



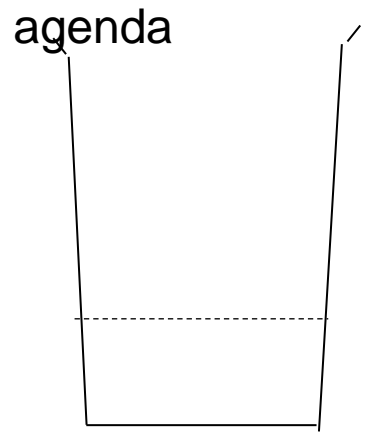
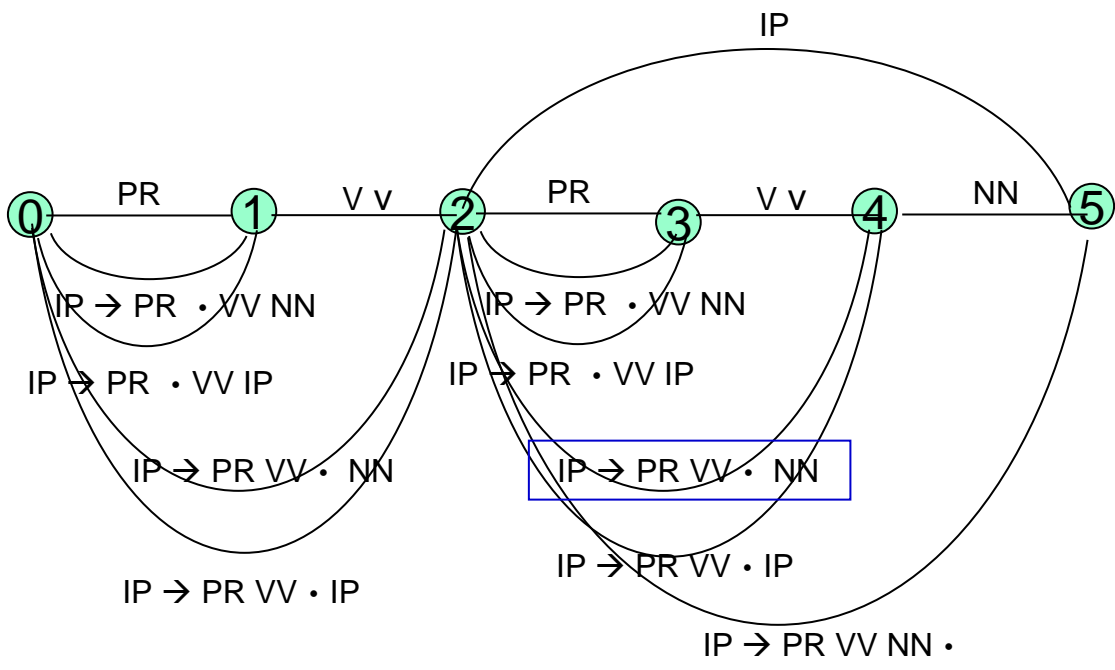
输入缓冲区

chart
—
active
arc

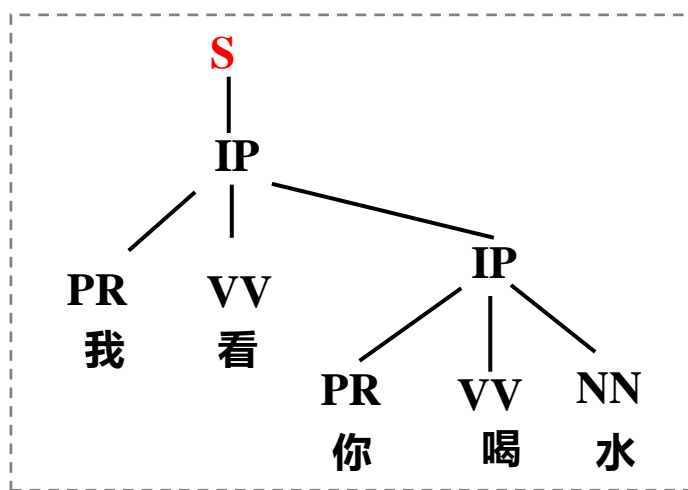


- P:
1. $S \rightarrow IP$
 2. $IP \rightarrow PR \quad VV \quad NN$
 3. $IP \rightarrow PR \quad VV \quad IP$

(4,5) NN

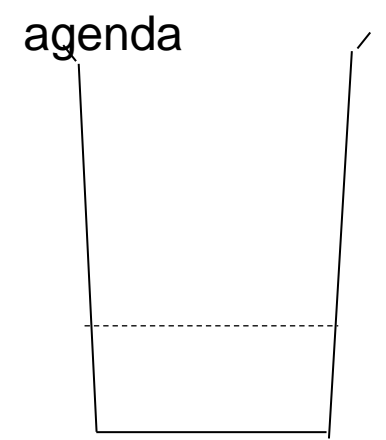
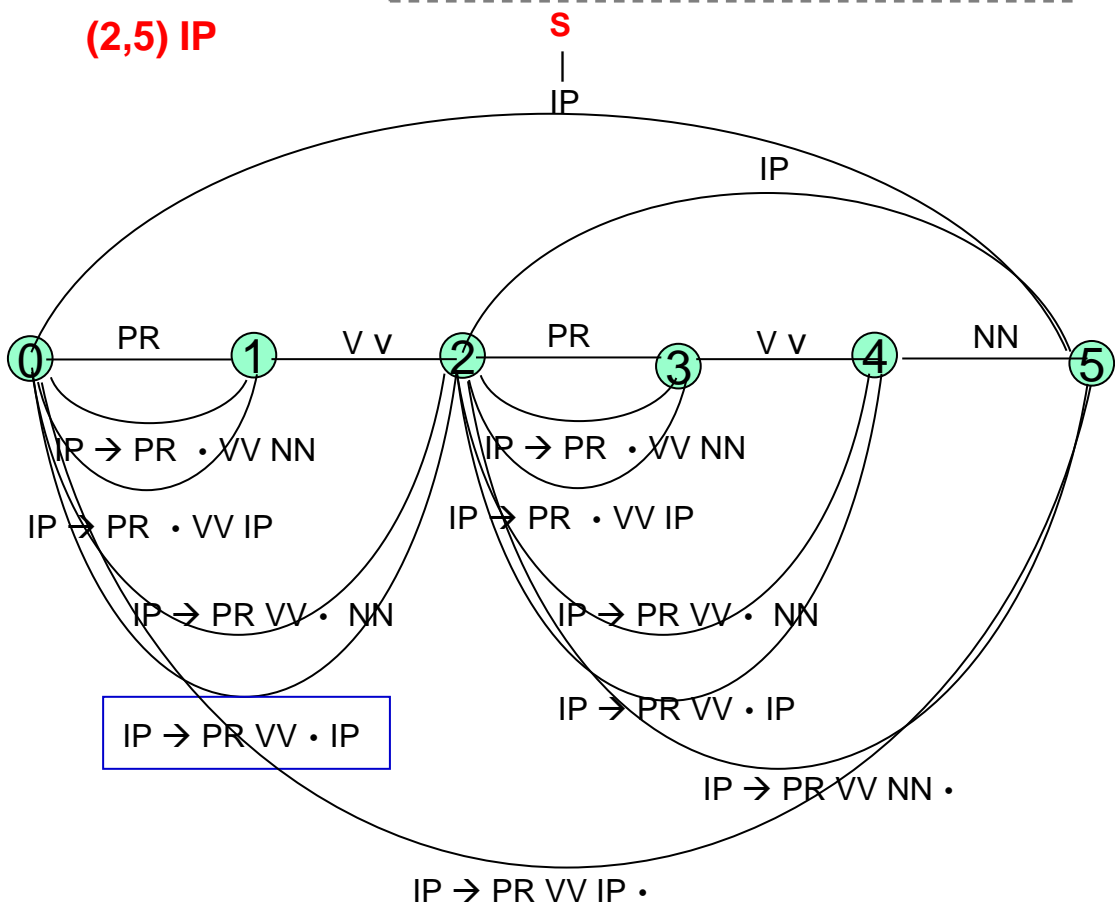


输入缓冲区



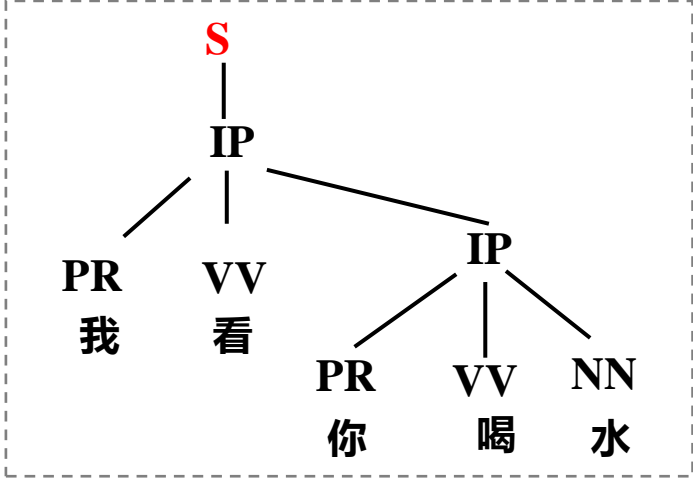
- P:
1. $S \rightarrow IP$
 2. $IP \rightarrow PR \quad VV \quad NN$
 3. $IP \rightarrow PR \quad VV \quad IP$

(2,5) IP



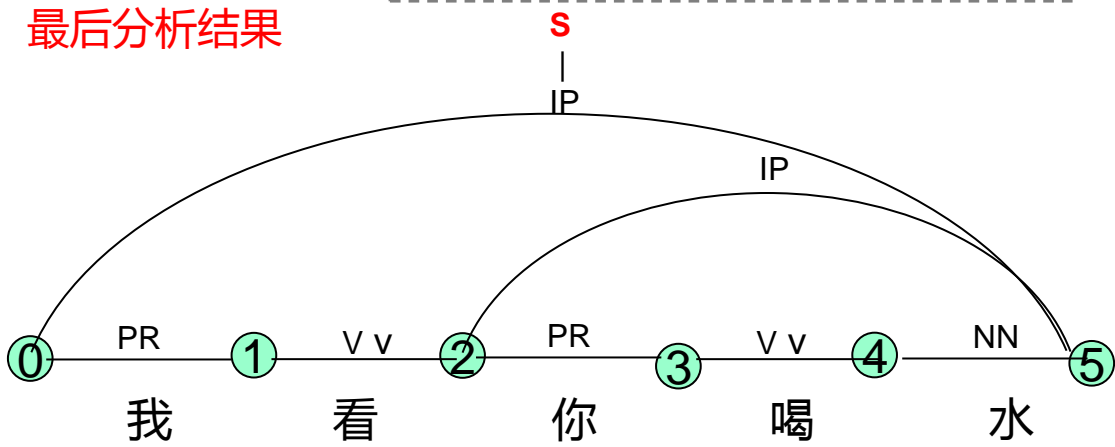
输入缓冲区

c
h
a
r
t
—
a
c
t
i
v
e
a
r
c



- P:
- 1. $S \rightarrow IP$
 - 2. $IP \rightarrow PR \quad VV \quad NN$
 - 3. $IP \rightarrow PR \quad VV \quad IP$

最后分析结果



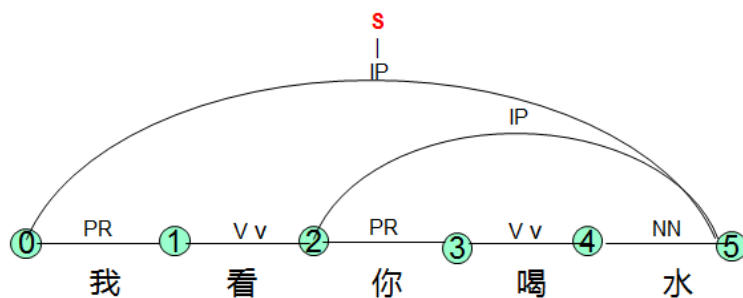
线图Chart

PR	VV	PR	VV	NN
我	看	你	喝	水

Chart算法的时间复杂度为： $O(Kn^3)$
n 为句子的长度，K 为一常数

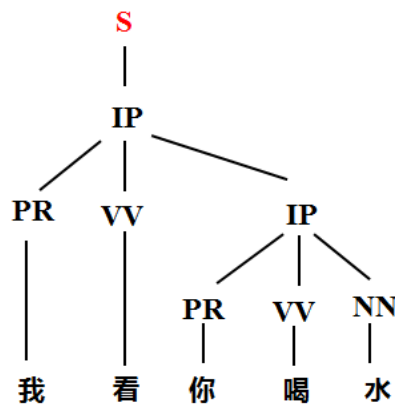
1. 线图分析法 (Chart)

线图



将上图中的边改为结点，将结点改为边，得到分析结果的树型图。

树型图



1. 线图分析法 (Chart)

Chart parsing 算法评价

优点：

- 算法简单，容易实现，开发周期短。

弱点：

- 算法效率低，时间复杂度为 Kn^3 ；
- 需要高质量的规则，分析结果与规则质量密切相关；
- 难以区分歧义结构。

2. CYK 分析算法

2. CYK (Coke-Younger-Kasami) 分析算法

CYK 算法首先是由Ney(1991)描述的，但后来采用的概率CYK 算法的版本来自 Collins (1999) 和 Aho and Ullman (1972)

- 自下而上的分析方法
- 需要对 Chomsky 文法进行范式化：

$$A \rightarrow a \text{ 或 } A \rightarrow BC$$

$$A, B, C \in V_N, \quad a \in V_T, \quad G=(V_N, V_T, P, S)$$

算法思想：通过构造识别矩阵进行分析

2. CYK 分析算法

CYK 分析算法描述

- (1) 首先构造主对角线，令 $t_{0,0}=0$ ，然后，从 $t_{1,1}$ 到 $t_{n,n}$ 在主对角线的位置上依次放入输入句子 x 的单词 a_i 。
- (2) 构造主对角线以上紧靠主对角线的元素 $t_{i, i+1}$ ，其中， $i = 0, 1, 2, \dots, n-1$ 。对于输入句子 $x = a_1 a_2 \dots a_n$ ，从 a_1 开始分析。

如果在文法 G 的产生式集中有一条规则：

$$A \rightarrow a_1 \quad \text{则 } t_{0,1} = A。$$

依此类推，如果有 $A \rightarrow a_{i+1}$ ，则 $t_{i, i+1} = A$ 。

- (3) 按平行于主对角线的方向，一层一层地向上填写矩阵的各个元素 $t_{i,j}$ ，其中， $i = 0, 1, \dots, n-d$ ， $j = d+i$ ， $d=2, 3, \dots, n$ 。如果存在一个正整数 k ， $i+1 \leq k \leq j-1$ ，在文法 G 的规则集中有产生式 $A \rightarrow BC$ ，并且， $B \in t_{i,k}$ ， $C \in t_{k,j}$ ，那么，将 A 写到矩阵 $t_{i,j}$ 位置上。

判断句子 x 由文法 G 所产生的充要条件是： $t_{0,n}=S$ 。

2. CYK 分析算法

例： 给定文法 $G(S)$ ：

- (1) $S \rightarrow P VP$
- (2) $VP \rightarrow V V$
- (3) $VP \rightarrow VP N$
- (4) $P \rightarrow \text{他}$
- (5) $V \rightarrow \text{喜欢}$
- (6) $V \rightarrow \text{读}$
- (7) $N \rightarrow \text{书}$

用 CYK 算法分析句子：

他喜欢读书

解： (1) 汉语分词和词性标注以后：

他/P 喜欢/V 读/V 书/N $n=4$

(2) 构造识别矩阵：

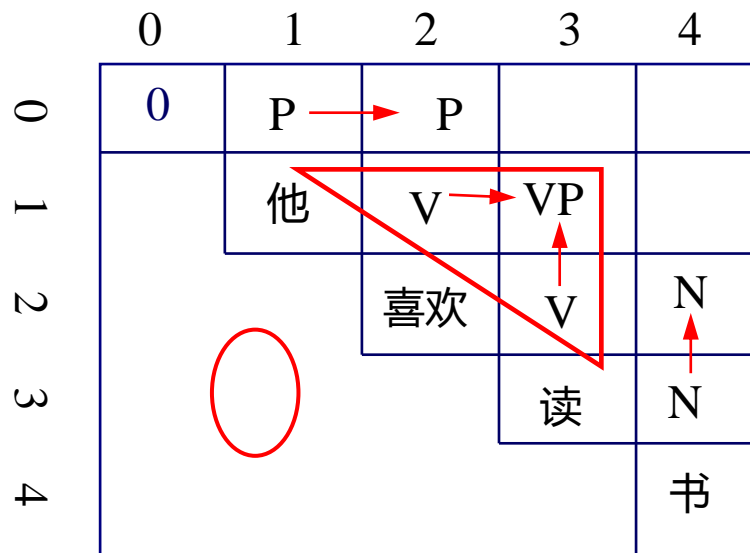
	0	1	2	3	4
0	0	P			
1		他	V		
2			喜欢	V	
3				读	N
4					书

2. CYK 分析算法

(3) 执行分析过程

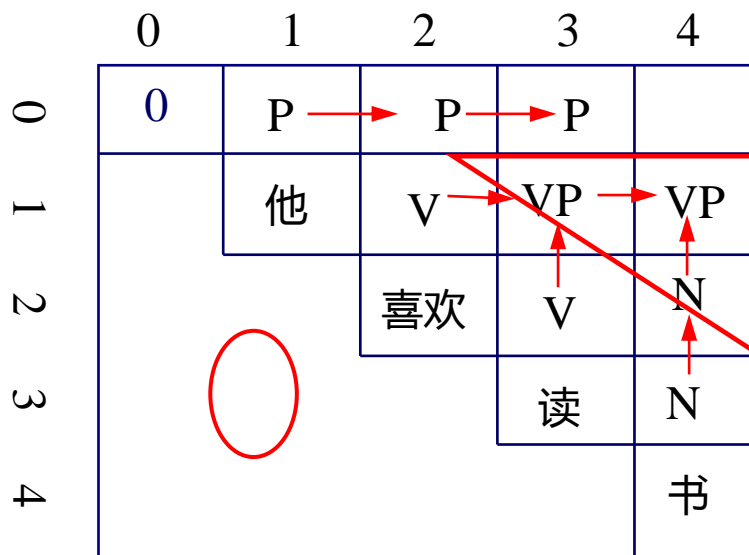
第一遍

- (1) $S \rightarrow P VP$
- (2) $VP \rightarrow V V$
- (3) $VP \rightarrow VP N$
- (4) $P \rightarrow \text{他}$
- (5) $V \rightarrow \text{喜欢}$
- (6) $V \rightarrow \text{读}$
- (7) $N \rightarrow \text{书}$



第二遍

- (1) $S \rightarrow P VP$ ✗
- (2) $VP \rightarrow V V$
- (3) $VP \rightarrow VP N$
- (4) $P \rightarrow \text{他}$
- (5) $V \rightarrow \text{喜欢}$
- (6) $V \rightarrow \text{读}$
- (7) $N \rightarrow \text{书}$



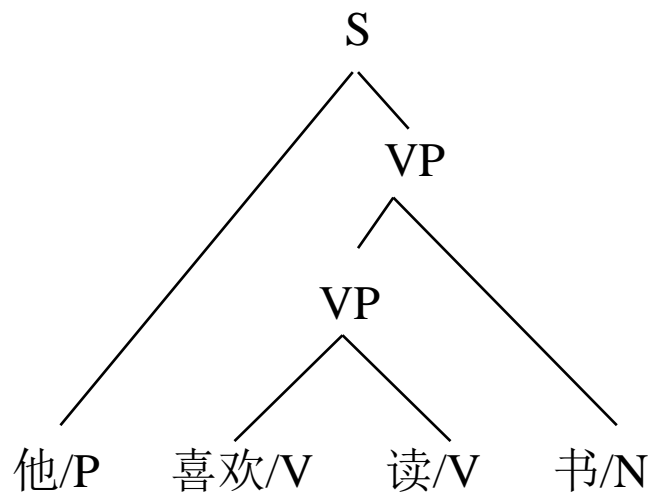
2. CYK 分析算法

第三遍

- (1) $S \rightarrow P VP$
- (2) $VP \rightarrow V V$
- (3) $VP \rightarrow VP N$
- (4) $P \rightarrow \text{他}$
- (5) $V \rightarrow \text{喜欢}$
- (6) $V \rightarrow \text{读}$
- (7) $N \rightarrow \text{书}$

	0	1	2	3	4
0	0	P →	P →	P →	S
1		他	V →	VP →	VP
2			喜欢	V	N
3				读	N
4					书

分析结果



2. CYK 分析算法

CYK 算法的评价

优点：

- 简单易行，执行效率高

弱点：

- 必须对文法进行范式化处理
- 无法区分歧义

完全句法分析-内容提 要

11.1.1 层次分析法

11.1.2 规则法完全句法分析

11.1.3 概率统计法完全句法分析

11.1.4 神经网络法完全句法分析

11.1.5 句法分析评价

11.1.3 概率统计法完全句法分析

主要采用概率上下文无关文法

概率上下文无关文法 (PCFG) 是CFG的概率拓广,可以直接统计语言学中词与词、词与词组以及词组与词组的规约信息,并且可以由语法规则生成给定句子的**概率**。

主要作用：可以定量的计算分析树的概率

{ 句法分析树的消歧
求最佳分析树

11.1.3 概率统计法完全句法分析

主要内容

1. 概率上下文无关文法 (PCFG) 定义
2. PCFG参数学习问题
3. 分析树概率计算 (句法分析树的消歧)
4. 求最佳分析树 (PCFG Viterbi 算法)

11.1.3 概率统计法完全句法分析

1. 概率上下文无关文法定义

定义：一个概率上下文无关语法 (PCFG) 由以下5部分组成：

- (1) 一个非终结符号集 N
- (2) 一个终结符号集 Σ
- (3) 一个开始非终结符 $S \in N$
- (4) 一个产生式集 R
- (5) 对于任意产生式 $r \in R$ ，其概率为 $P(r)$

产生式具有形式 $X \rightarrow Y, P$

其中, $X \in N, Y \in (N \cup \Sigma)^*$

$$\sum_{\lambda} P(X \rightarrow \lambda) = 1$$

11.1.3 概率统计法完全句法分析

如：

$S \rightarrow NP VP, 1.00$	$NP \rightarrow NP PP, 0.40$
$VP \rightarrow V NP, 0.70$	$VP \rightarrow VP PP, 0.30$
$PP \rightarrow P NP, 1.00$	$NP \rightarrow \text{ears}, 0.18$
$NP \rightarrow \text{saw}, 0.04$	$P \rightarrow \text{with}, 1.00$
$NP \rightarrow \text{stars}, 0.18$	$NP \rightarrow \text{telescopes}, 0.1$
$V \rightarrow \text{saw}, 1.00$	$NP \rightarrow \text{astronomers}, 0.10$

对于任意产生式 $r \in R$ ，其概率为 $P(r)$

产生式具有形式 $X \rightarrow Y, P$

其中， $X \in N, Y \in (N \cup \Sigma)^*$ ， $\sum_{\lambda} P(X \rightarrow \lambda) = 1$

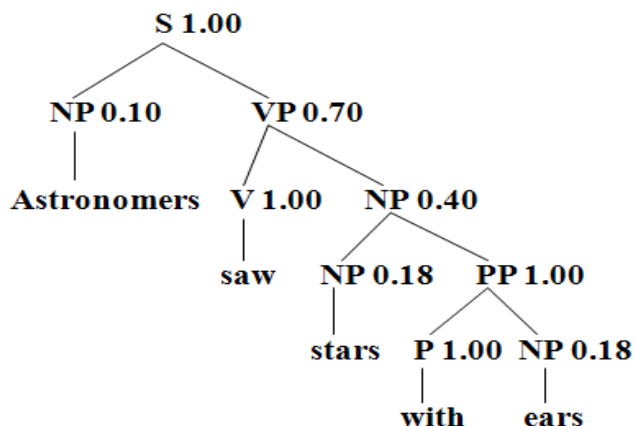
11.1.3 概率统计法完全句法分析

PCFG 与 CFG 对比

概率上下文无关文法 (PCFG) :

$S \rightarrow NP VP, 1.00$	$NP \rightarrow NP PP, 0.40$
$VP \rightarrow V NP, 0.70$	$VP \rightarrow VP PP, 0.30$
$PP \rightarrow P NP, 1.00$	$NP \rightarrow ears, 0.18$
$NP \rightarrow saw, 0.04$	$P \rightarrow with, 1.00$
$NP \rightarrow stars, 0.18$	$NP \rightarrow telescopes, 0.1$
$V \rightarrow saw, 1.00$	$NP \rightarrow astronomers, 0.10$

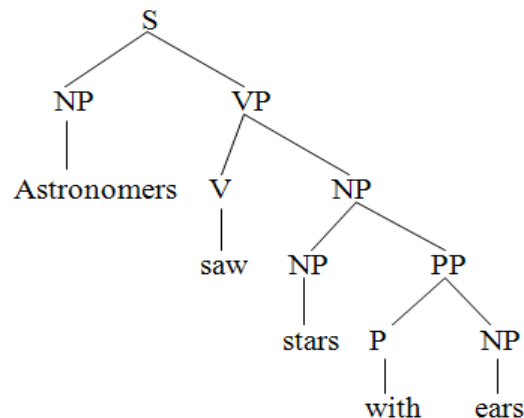
CFG分析树



上下文无关文法 (CFG) :

$S \rightarrow NP VP$	$NP \rightarrow NP PP$
$VP \rightarrow V NP$	$VP \rightarrow VP PP$
$PP \rightarrow P NP$	$NP \rightarrow ears$
$NP \rightarrow saw$	$P \rightarrow with$
$NP \rightarrow stars$	$NP \rightarrow telescopes$
$V \rightarrow saw$	$NP \rightarrow astronomers$

CFG分析树



句子 : Astronomers saw stars with ears

11.1.3 概率统计法完全句法分析

2. PCFG参数学习问题：

如何通过统计的方法得出 PCFG 中各产生式的概率？

如：

$S \rightarrow NP VP, 1.00$	$NP \rightarrow NP PP, 0.40$
$VP \rightarrow V NP, 0.70$	$VP \rightarrow VP PP, 0.30$
$PP \rightarrow P NP, 1.00$	$NP \rightarrow ears, 0.18$
$NP \rightarrow saw, 0.04$	$P \rightarrow with, 1.00$
$NP \rightarrow stars, 0.18$	$NP \rightarrow telescopes, 0.1$
$V \rightarrow saw, 1.00$	$NP \rightarrow astronomers, 0.10$

情况1：有大规模标注的树库语料

解决方案：用最大似然估计方法从树库直接统计

情况2：无标注语料

解决方案：用向内向外算法（借助 EM 迭代算法估计PCFG的概率参数）

11.1.3 概率统计法完全句法分析

情况1：从树库直接统计

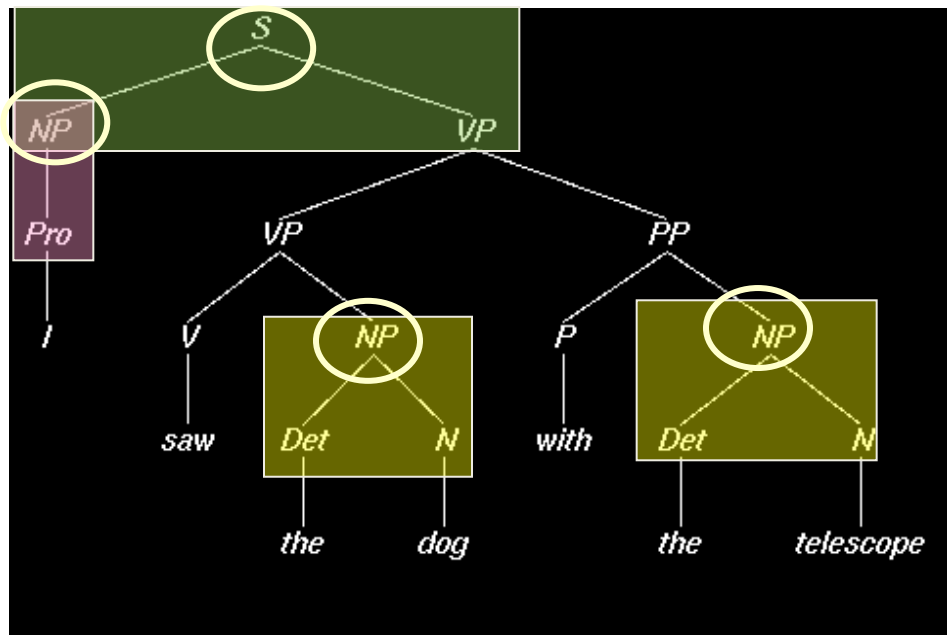
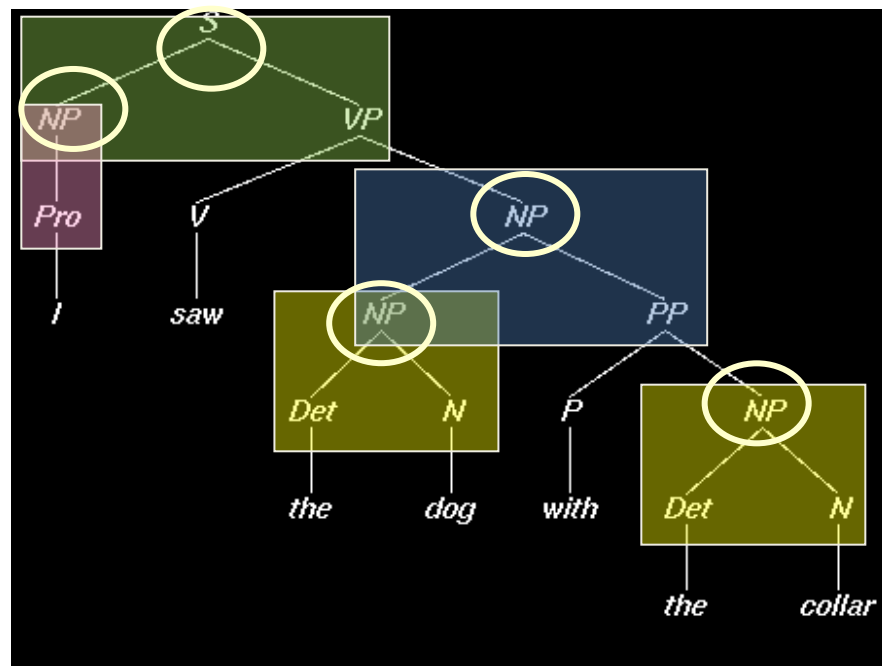
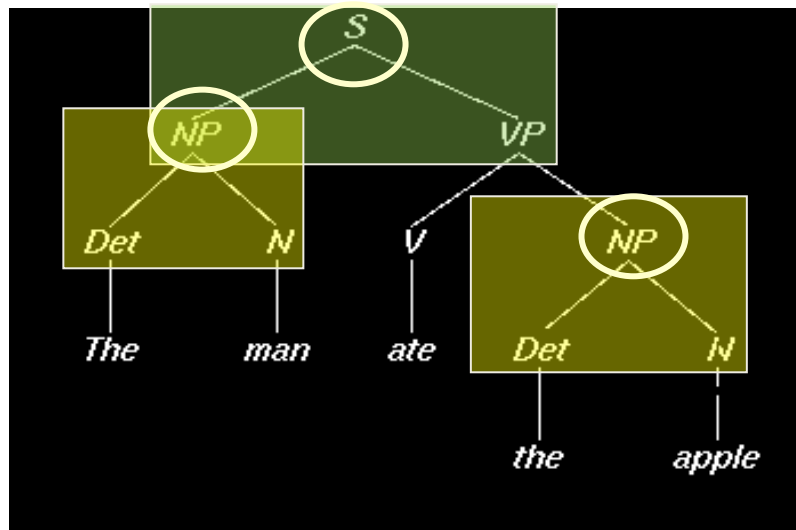
如果有大量已标注语法结构的训练语料，则可直接通过计算每个语法规则的使用次数，用最大似然估计方法计算 PCFG 规则的概率参数，即：

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

其中： $N^j \rightarrow \zeta$, $N^j \rightarrow \gamma$ 为产生式

11.1.3 概率统计法完全句法分析

PCFG参数学习过程



$$S \rightarrow NP VP \quad 3 / 3 = 1.0$$

$$NP \rightarrow Pro \quad 2 / 9 = 0.22$$

$$NP \rightarrow Det N \quad 6 / 9 = 0.67$$

$$NP \rightarrow NP PP \quad 1 / 9 = 0.11$$

$$VP \rightarrow V NP \quad 3 / 4 = 0.75$$

$$VP \rightarrow VP PP \quad 1 / 4 = 0.25$$

$$PP \rightarrow P NP \quad 2 / 2 = 1.0$$

11.1.3 概率统计法完全句法分析

3.分析树概率计算（句法分析树的消歧）

计算分析树概率的基本假设

- **位置不变性**：子树的概率与其管辖的词在整个句子中所处的位置无关，即对于任意的 k , $p(A_{k(k+C)} \rightarrow w)$ 一样。
- **上下文无关性**：子树的概率与子树管辖范围以外的词无关，
即 $p(A_{kl} \rightarrow w | \text{任何超出 } k \sim l \text{ 范围的上下文}) = p(A_{kl} \rightarrow w)$ 。
- **祖先无关性**：子树的概率与推导出该子树的祖先结点无关，
即 $p(A_{kl} \rightarrow w | \text{任何除 } A \text{ 以外的祖先结点}) = p(A_{kl} \rightarrow w)$ 。

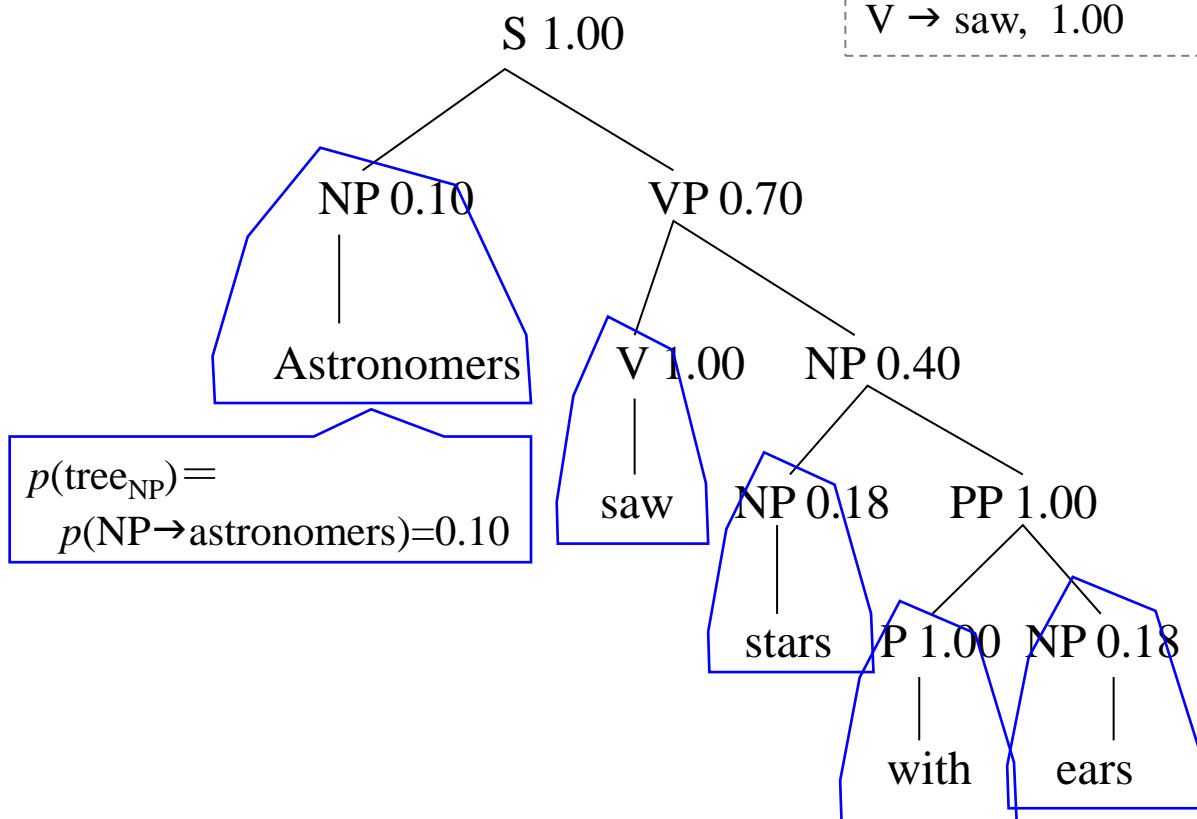
讨论中只考虑满足Chomsky 规范形式文法

11.1.3 概率统计法完全句法分析

一棵句法分析树概率计算：

设： t_1 是PCFG的一颗分析树

$S \rightarrow NP VP, 1.00$	$NP \rightarrow NP PP, 0.40$
$VP \rightarrow V NP, 0.70$	$VP \rightarrow VP PP, 0.30$
$PP \rightarrow P NP, 1.00$	$NP \rightarrow ears, 0.18$
$NP \rightarrow saw, 0.04$	$P \rightarrow with, 1.00$
$NP \rightarrow stars, 0.18$	$NP \rightarrow telescopes, 0.1$
$V \rightarrow saw, 1.00$	$NP \rightarrow astronomers, 0.10$

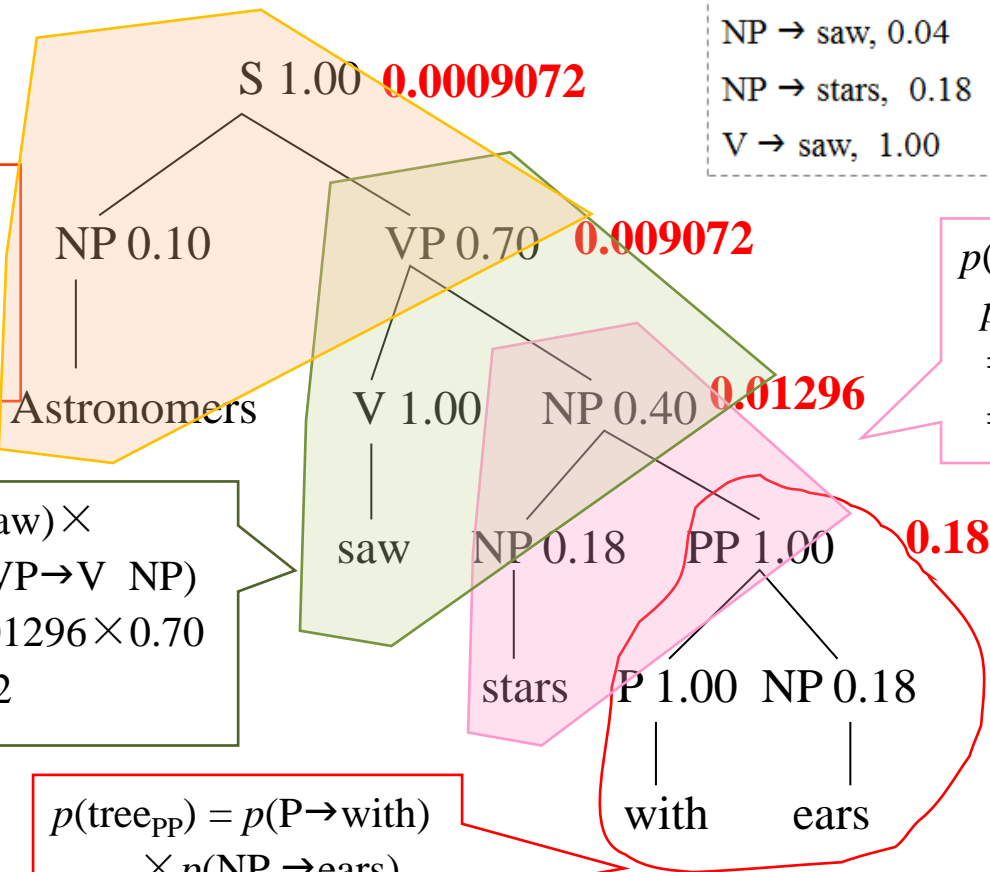


11.1.3 概率统计法完全句法分析

$S \rightarrow NP VP, 1.00$	$NP \rightarrow NP PP, 0.40$
$VP \rightarrow V NP, 0.70$	$VP \rightarrow VP PP, 0.30$
$PP \rightarrow P NP, 1.00$	$NP \rightarrow ears, 0.18$
$NP \rightarrow saw, 0.04$	$P \rightarrow with, 1.00$
$NP \rightarrow stars, 0.18$	$NP \rightarrow telescopes, 0.1$
$V \rightarrow saw, 1.00$	$NP \rightarrow astronomers, 0.10$

t_1 概率 :

$$\begin{aligned}
 p(t_1) &= p(\text{tree}_{NP}) \times \\
 & p(VP) \times p(S \rightarrow NP VP) \\
 &= 0.10 \times 1.0 \times 0.009072 \\
 &= 0.0009072
 \end{aligned}$$



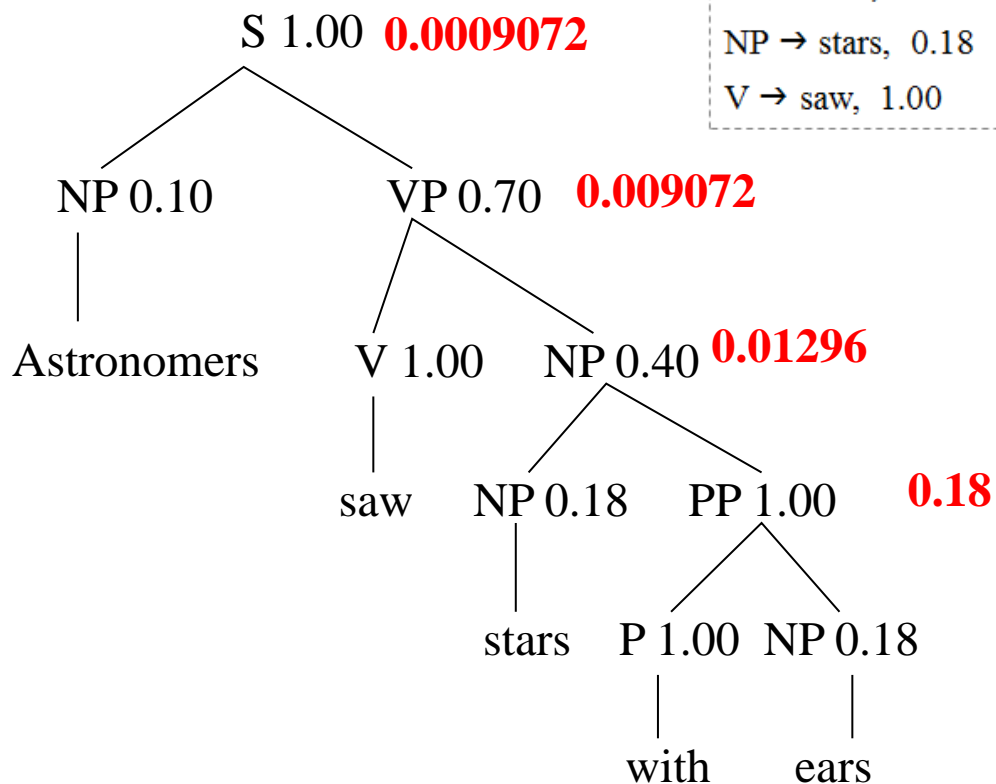
$$\begin{aligned}
 p(VP) &= p(V \rightarrow saw) \times \\
 & p(\text{tree}_{NP}) \times p(VP \rightarrow V NP) \\
 &= 1.0 \times 0.01296 \times 0.70 \\
 &= 0.009072
 \end{aligned}$$

$$\begin{aligned}
 p(\text{tree}_{NP}) &= p(NP \rightarrow stars) \times \\
 & p(\text{tree}_{PP}) \times p(NP \rightarrow NP PP) \\
 &= 0.18 \times 0.18 \times 0.4 \\
 &= 0.01296
 \end{aligned}$$

$$\begin{aligned}
 p(\text{tree}_{PP}) &= p(P \rightarrow with) \\
 & \times p(NP \rightarrow ears) \\
 & \times p(PP \rightarrow P NP) \\
 &= 1.00 \times 0.18 \times 1.00 \\
 &= 0.18
 \end{aligned}$$

11.1.3 概率统计法完全句法分析

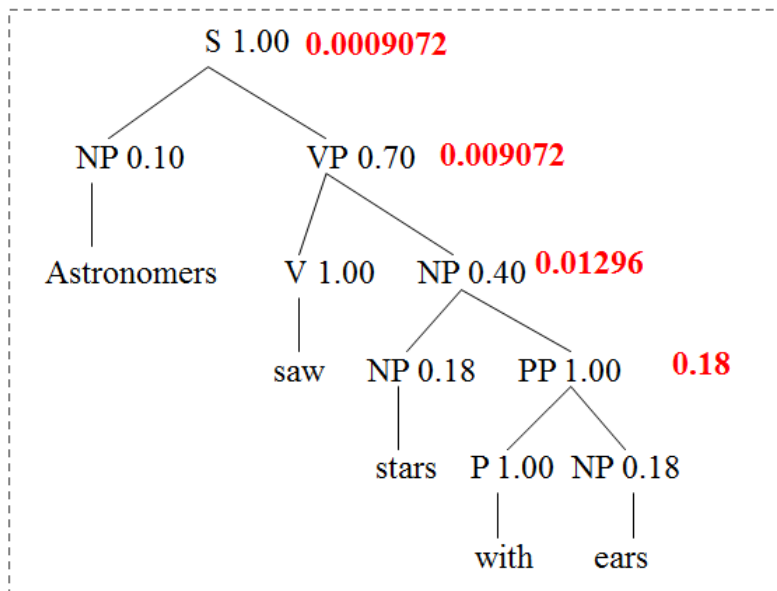
t_1 概率：



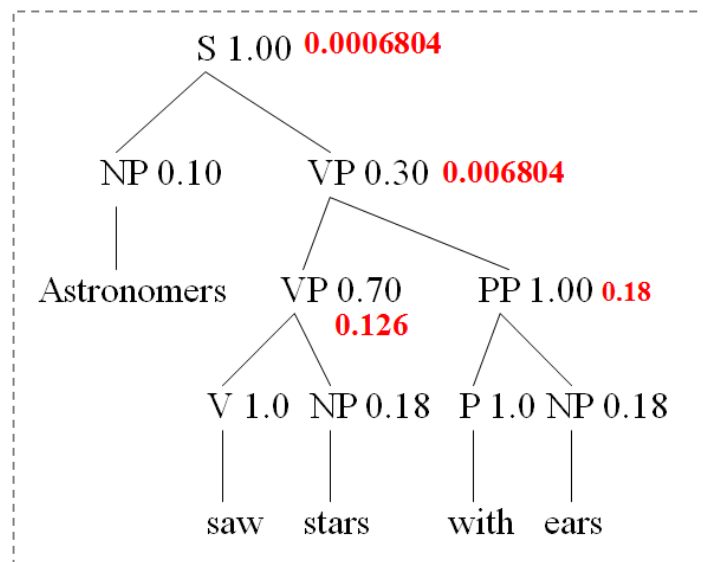
$S \rightarrow NP VP, 1.00$	$NP \rightarrow NP PP, 0.40$
$VP \rightarrow V NP, 0.70$	$VP \rightarrow VP PP, 0.30$
$PP \rightarrow P NP, 1.00$	$NP \rightarrow ears, 0.18$
$NP \rightarrow saw, 0.04$	$P \rightarrow with, 1.00$
$NP \rightarrow stars, 0.18$	$NP \rightarrow telescopes, 0.1$
$V \rightarrow saw, 1.00$	$NP \rightarrow astronomers, 0.10$

11.1.3 概率统计法完全句法分析

t_1 :



t_2 :



对于给定的句子 S ，如有两棵句法分析树的概率不等且 $P(t_1) > P(t_2)$ ，则分析结果 t_1 正确的可能性大于 t_2 。可以用此法进行**句法歧义消解**

语句在文法的概率等于所有分析树概率之和（可以用内向算法求）

11.1.3 概率统计法完全句法分析

4. 求最佳分析树

问题描述：

在语句 W 的句法结构有歧义的情况下,给定句子 $W = w_1w_2...w_n$ 和 PCFG , 如何快速的选择最佳句法结构树 ?

穷举法：找到每一个可能的句法树，计算概率，然后取概率最大的

效率非常低，尤其是当句子较长，生成句法树有多棵时效率极低。

解决方法：采用动态规划算法， **PCFG Viterbi 算法**

11.1.3 概率统计法完全句法分析

PCFG-Viterbi 算法描述：

变量定义：

$\gamma_{ij}(A)$: 非终结符 A 推导出语句 W 中子字串 $w_i w_{i+1} \cdots w_j$ 的最大概率

$\psi_{i,j}$: 记忆字串 $w_i w_{i+1} \cdots w_j$ 的 Viterbi 语法分析结果。

输入：文法 $G(S)$ ，语句 $W = w_1 w_2 \cdots w_n$

(1) 初始化： $\gamma_{ii}(A) = p(A \rightarrow w_i) \quad A \in V_N \quad 1 \leq i \leq j \leq n$

(2) 归纳计算： $j=1..n, i=1..n-j$, 重复下列计算：

$$\gamma_{i(i+j)}(A) = \max_{B, C \in V_N; i \leq k \leq i+j} p(A \rightarrow BC) \gamma_{ik}(B) \gamma_{(k+1)(i+j)}(C)$$

$$\psi_{i(i+j)}(A) = \max_{B, C \in V_N; i \leq k \leq i+j} p(A \rightarrow BC) \gamma_{ik}(B) \gamma_{(k+1)(i+j)}(C)$$

输出：分析树根结点为 s (文法开始符号)，从 $\psi_{1,n}(S)$ 开始回溯，得到最优树

11.1.3 概率统计法完全句法分析

例: 已知 PCFG如下:

$S \rightarrow NP VP, 1.00$	$NP \rightarrow NP PP, 0.40$
$VP \rightarrow V NP, 0.70$	$VP \rightarrow VP PP, 0.30$
$PP \rightarrow P NP, 1.00$	$NP \rightarrow \text{bone}, 0.18$
$NP \rightarrow \text{star}, 0.04$	$P \rightarrow \text{with}, 1.00$
$NP \rightarrow \text{fish}, 0.18$	$NP \rightarrow \text{John}, 0.1$
$V \rightarrow \text{ate}, 1.00$	$NP \rightarrow \text{telescope}, 0.1$

输入句子: John ate fish with bone

求: 最佳的语法分析树。

$S \rightarrow NP VP, 1.00$	$NP \rightarrow NP PP, 0.40$
$VP \rightarrow V NP, 0.70$	$VP \rightarrow VP PP, 0.30$
$PP \rightarrow P NP, 1.00$	$NP \rightarrow \text{bone}, 0.18$
$NP \rightarrow \text{star}, 0.04$	$P \rightarrow \text{with}, 1.00$
$NP \rightarrow \text{fish}, 0.18$	$NP \rightarrow \text{John}, 0.1$
$V \rightarrow \text{ate}, 1.00$	$NP \rightarrow \text{telescope}, 0.1$

输入：文法 $G(S)$ ，语句 $W = w_1 w_2 \cdots w_n$

(1) 初始化： $\gamma_i(A) = p(A \rightarrow w_i) \quad A \in V_N \quad 1 \leq i \leq n$

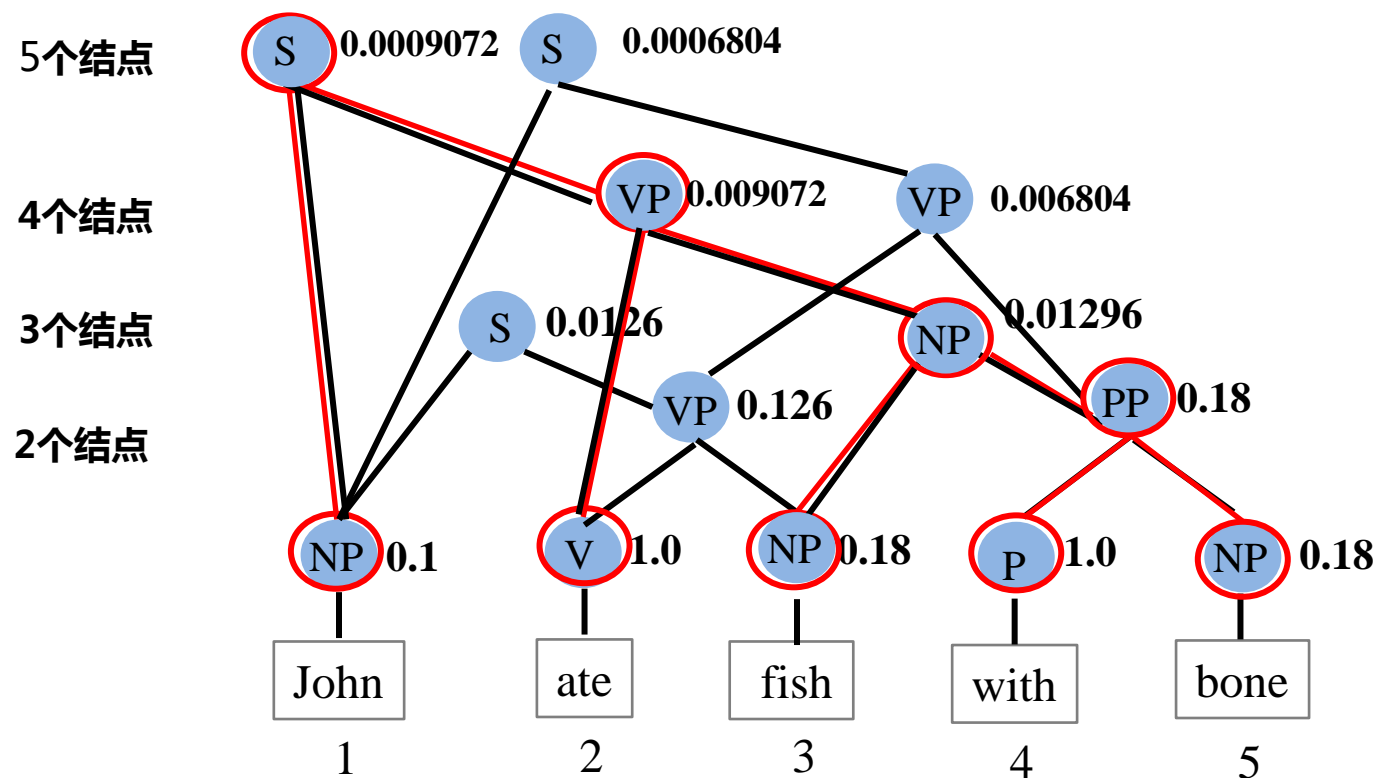
(2) 归纳计算： $j=1..n, i=1..n-j$, 重复下列计算：

$$\gamma_{i(i+j)}(A) = \max_{B, C \in V_N; i \leq k \leq i+j} p(A \rightarrow BC) \gamma_{ik}(B) \gamma_{(k+1)(i+j)}(C)$$

$$\psi_{i(i+j)}(A) = \max_{B, C \in V_N; i \leq k \leq i+j} p(A \rightarrow BC) \gamma_{ik}(B) \gamma_{(k+1)(i+j)}(C)$$

输出：分析树根结点为 s (文法开始符号)，从 $\psi_{1,n}(S)$ 开始回溯，得到最优树

解：输入句子 **John ate fish with bone**



$S \rightarrow NP VP, 1.00$	$NP \rightarrow NP PP, 0.40$
$VP \rightarrow V NP, 0.70$	$VP \rightarrow VP PP, 0.30$
$PP \rightarrow P NP, 1.00$	$NP \rightarrow \text{bone}, 0.18$
$NP \rightarrow \text{star}, 0.04$	$P \rightarrow \text{with}, 1.00$
$NP \rightarrow \text{fish}, 0.18$	$NP \rightarrow \text{John}, 0.1$
$V \rightarrow \text{ate}, 1.00$	$NP \rightarrow \text{telescope}, 0.1$

输入：文法 $G(S)$ ，语句 $W = w_1 w_2 \cdots w_n$

(1) 初始化： $\gamma_i(A) = p(A \rightarrow w_i) \quad A \in V_N \quad 1 \leq i \leq n$

(2) 归纳计算： $j=1..n, i=1..n-j$, 重复下列计算：

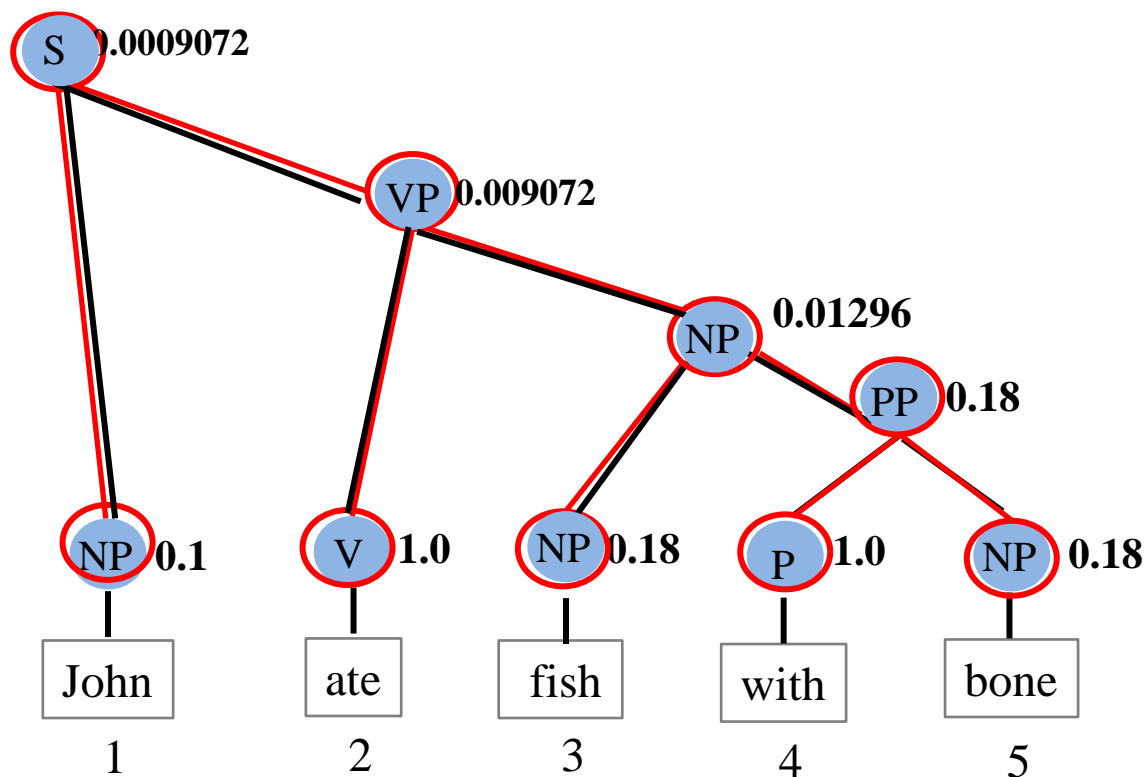
$$\gamma_{i(i+j)}(A) = \max_{B, C \in V_N; i \leq k \leq i+j} p(A \rightarrow BC) \gamma_{ik}(B) \gamma_{(k+1)(i+j)}(C)$$

$$\psi_{i(i+j)}(A) = \max_{B, C \in V_N; i \leq k \leq i+j} p(A \rightarrow BC) \gamma_{ik}(B) \gamma_{(k+1)(i+j)}(C)$$

输出：分析树根结点为 s (文法开始符号)，从 $\psi_{1,n}(S)$ 开始回溯，得到最优树

解：输入句子 **John ate fish with bone**

最佳语法分析树



完全句法分析-内容提 要

11. 1. 1 层次分析法

11. 1. 2 规则法完全句法分析

11. 1. 3 概率统计法完全句法分析

11. 1. 4 神经网络法完全句法分析

11. 1. 4. 1 递归神经网络

11. 1. 4. 2 神经网络句法分析

11. 1. 5 句法分析评价

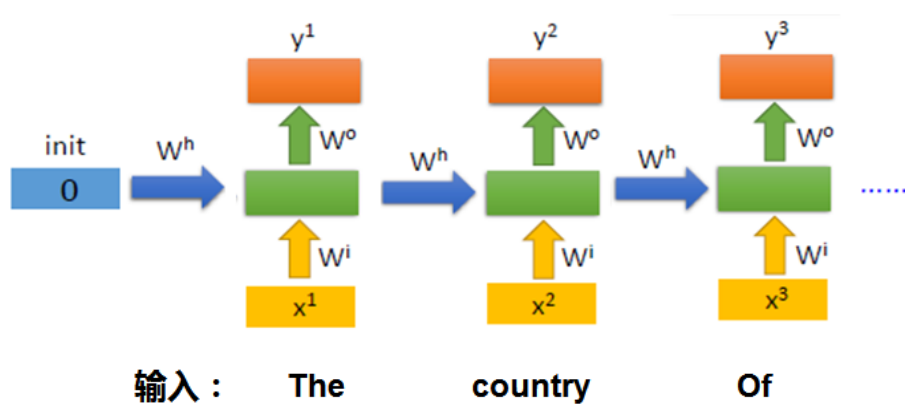
11.1.4.1 递归神经网络

问题引入：

RNN 适合处理时序上线性结展开的输入序列

RNN

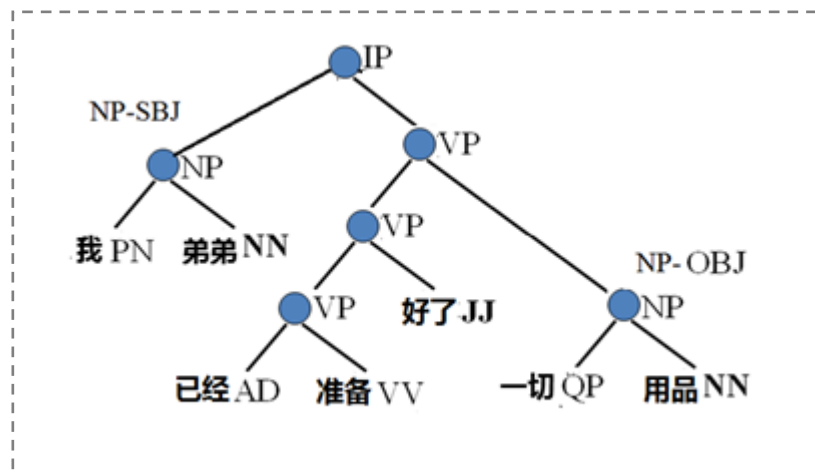
Input data: x^1 x^2 x^3 x^N



11.1.4.1 递归神经网络

对于结构展开的非线性句法关系表示能力差

例如： 短语结构树



而自然语言处理中语句内部的句法关系通常是非线性的

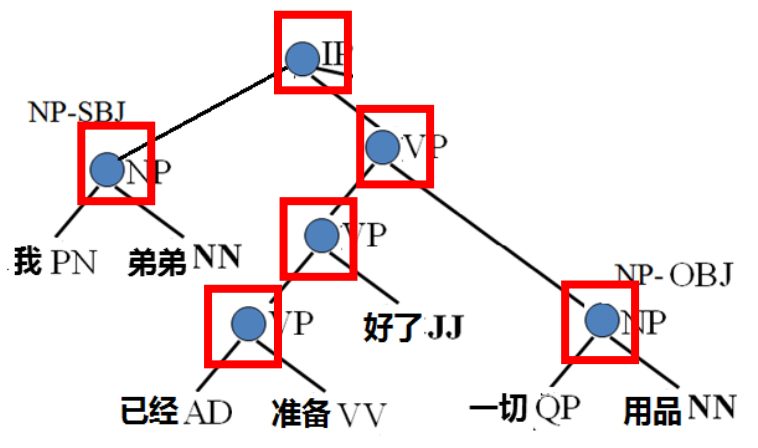
递归神经网络 (RvNN)

11.1.4.1 递归神经网络

递归神经网络基本思想：

将处理问题在结构上分解为一系列相同的“单元”，单元的神经网络可以在结构上展开，且能沿展开方向传递信息。

如：短语结构树

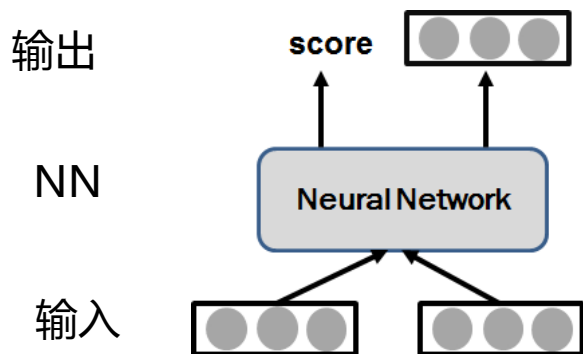


我 弟弟 已经 准备 好了 一切 用品

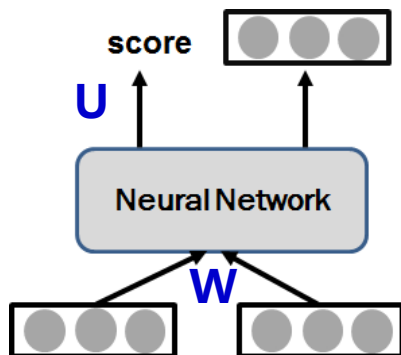
11.1.4.1 递归神经网络

RvNN单元：

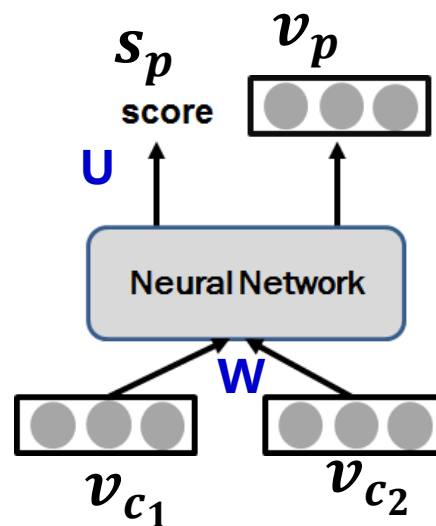
单元结构：



单元参数：

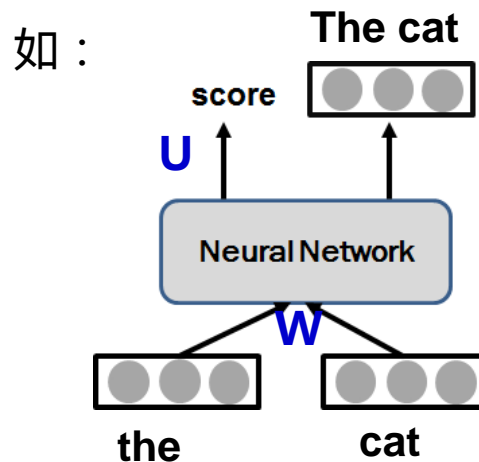


信息传播：



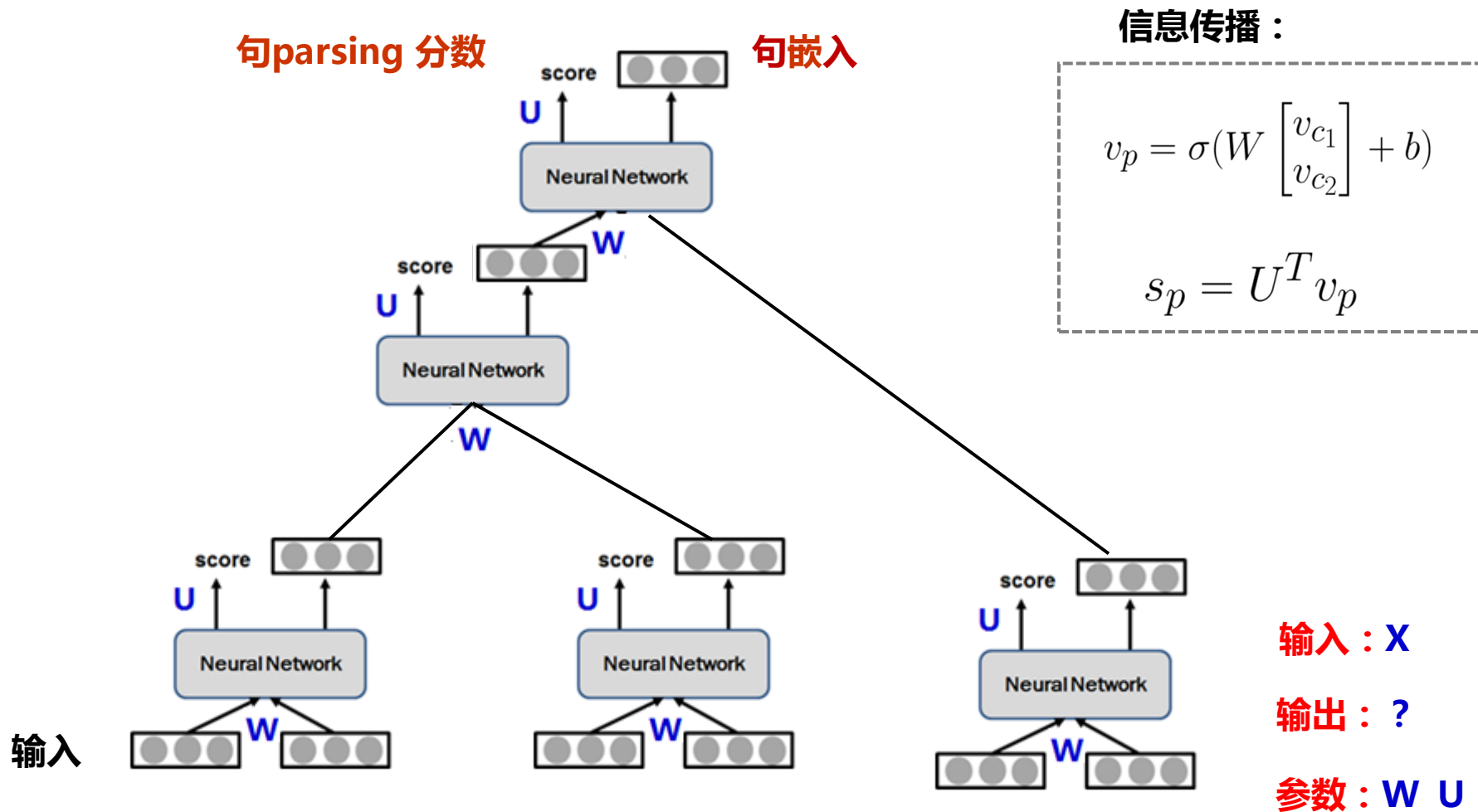
$$v_p = \sigma\left(W \begin{bmatrix} v_{c1} \\ v_{c2} \end{bmatrix} + b\right)$$

$$s_p = U^T v_p$$



11.1.4.1 递归神经网络

RvNN网络（按结构展开）：

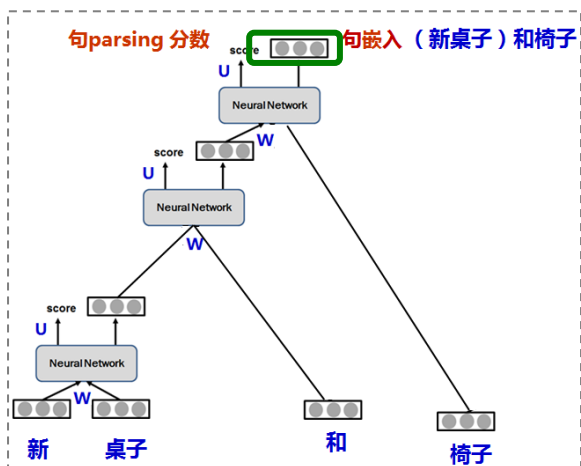


11.1.4.1 递归神经网络

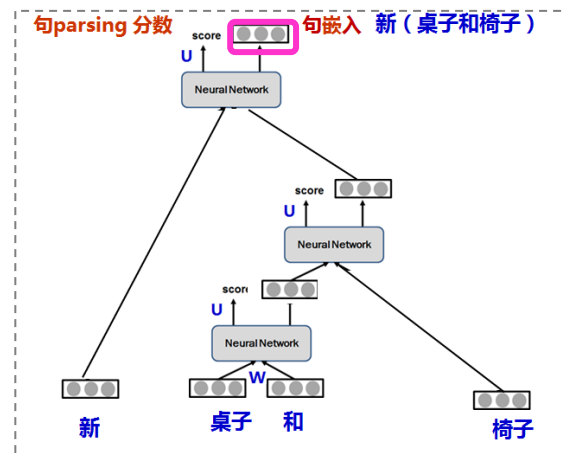
RvNN 句向量

如：新桌子和椅子

(新桌子)和椅子



新 (桌子和椅子)

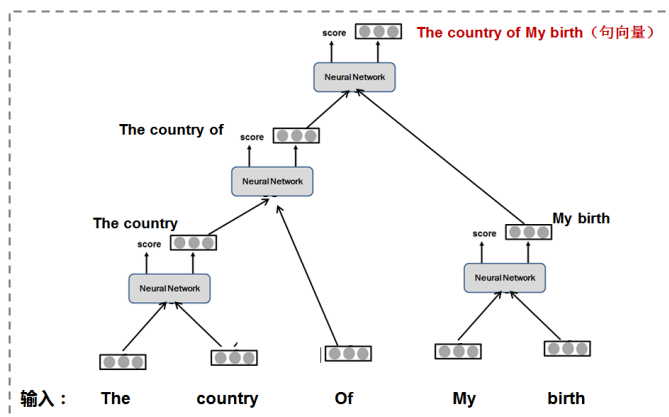


不同的结构
句向量不同

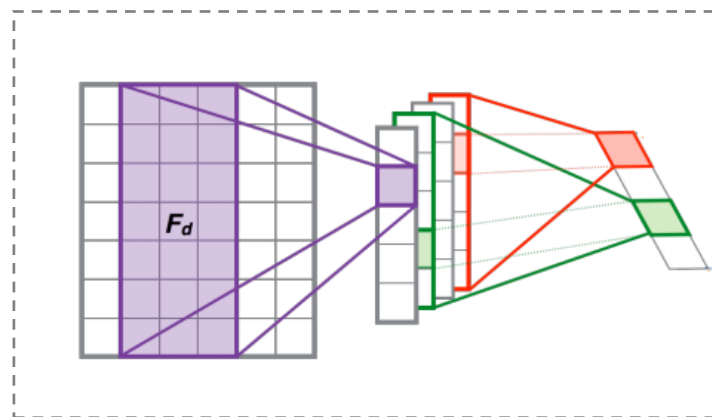
RvNN 句向量包含结构信息

11.1.4.1 递归神经网络

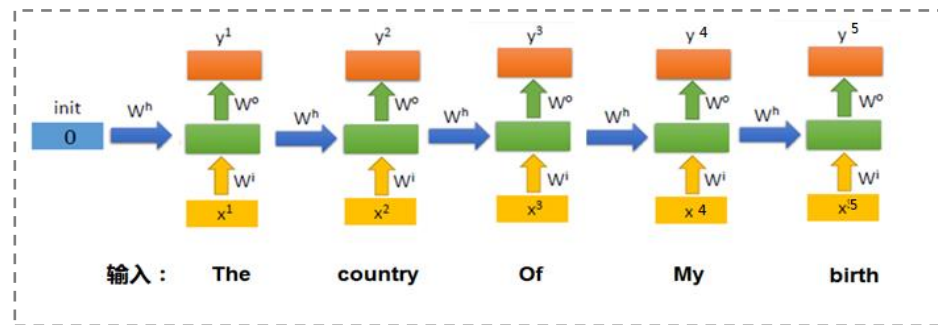
各种句向量



RvNN 句向量



CNN 句向量

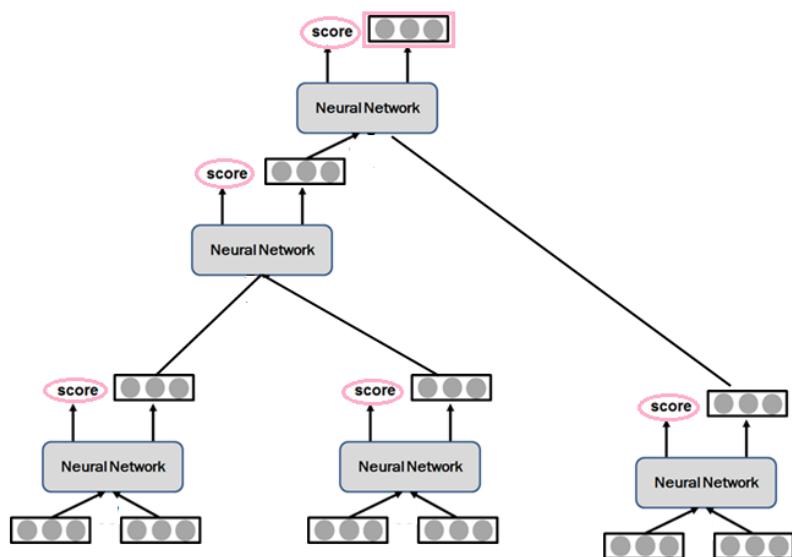


RNN 句向量

11.1.4.1 递归神经网络

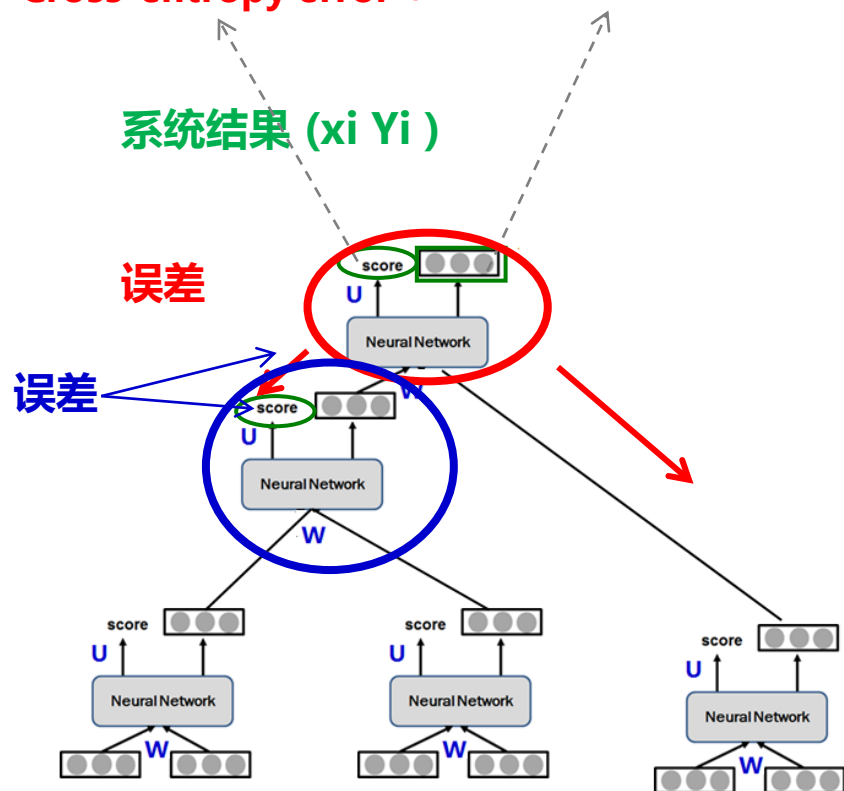
RvNN学习：(参数：W U)

语料实例 (xi Yi)



输入：词1 词2 词3 词4 词5 词6

- **损失函数：**
Cross-entropy error + Reconstruction error



输入：词1 词2 词3 词4 词5 词6

11.1.4.1 递归神经网络

- 参数训练（采用后向传播算法）

与传统的反向算法不同点：

- 每个节点误差由信息部分和打分部分误差组成
- 非根节点误差来源本位打分结点误差和父结点回传误差
- 每个节点误差反向传播时在都分别传给其子节点
- 参数 W 和 U 共享，调参时对 W 、 U 每步都调整

完全句法分析-内容提 要

11. 1. 1 层次分析法

11. 1. 2 规则法完全句法分析

11. 1. 3 概率统计法完全句法分析

11. 1. 4 神经网络法完全句法分析

11. 1. 4. 1 递归神经网络

11. 1. 4. 2 神经网络句法分析

11. 1. 5 句法分析评价

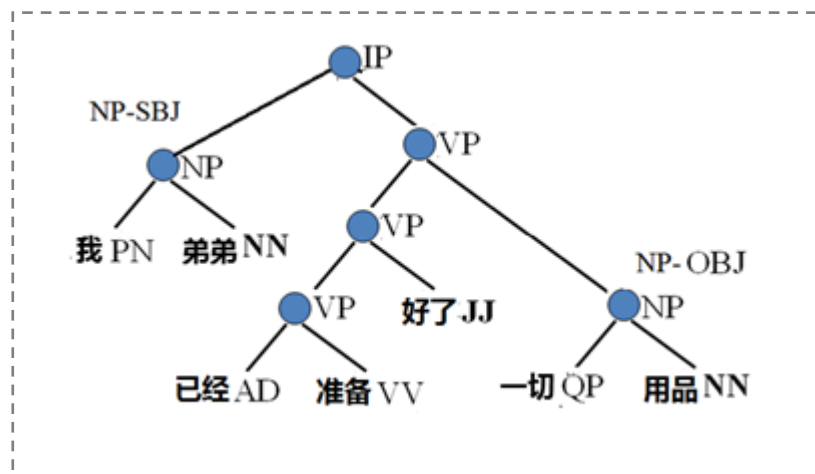
11.1.4.2 神经网络句法分析

短语结构分析

对给定的句子分析其短语结构树

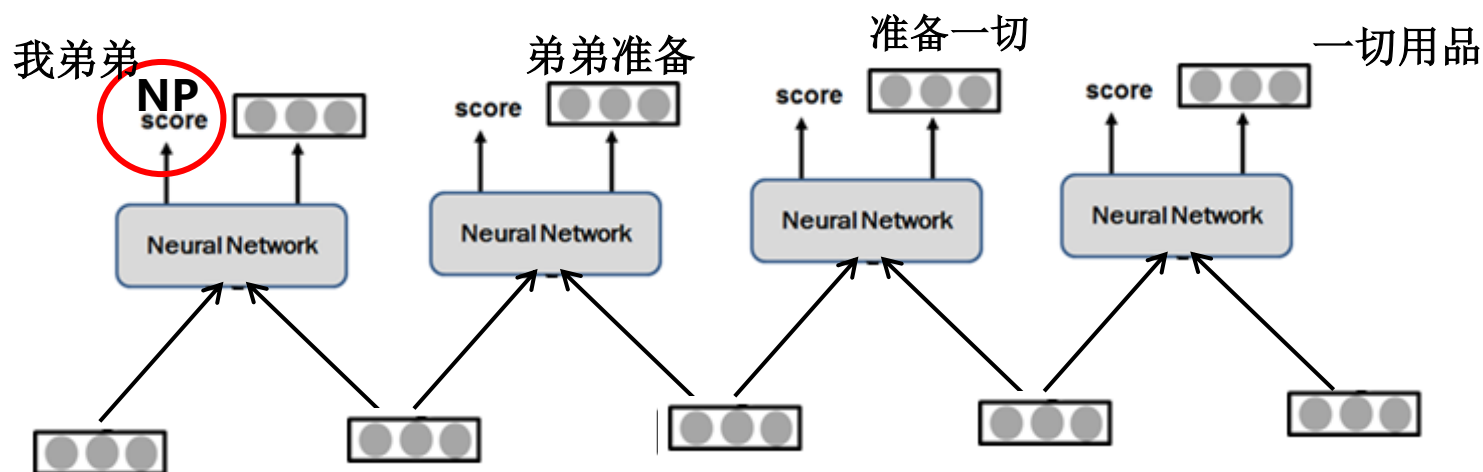
如：我 弟弟 已经 准备 好了 一切 用品

目标：生成短语结构树



11.1.4.2 神经网络句法分析

RvNN 短语结构句法分析：



输入：

我/PN

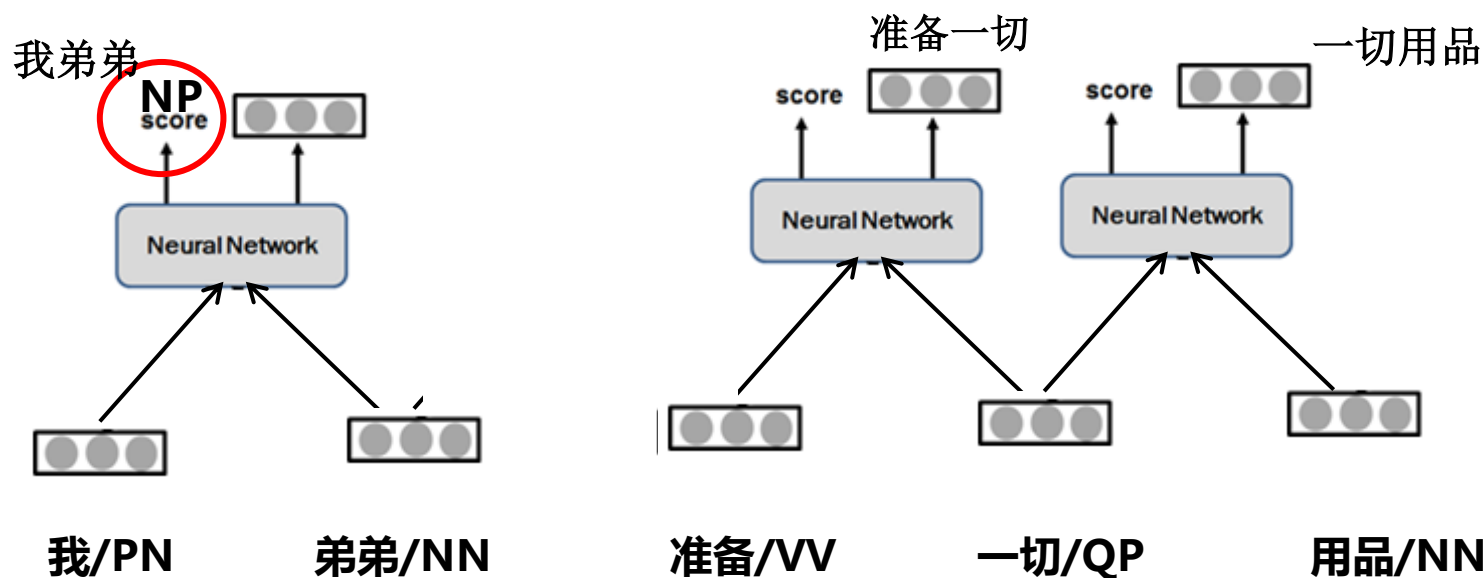
弟弟/NN

准备/VV

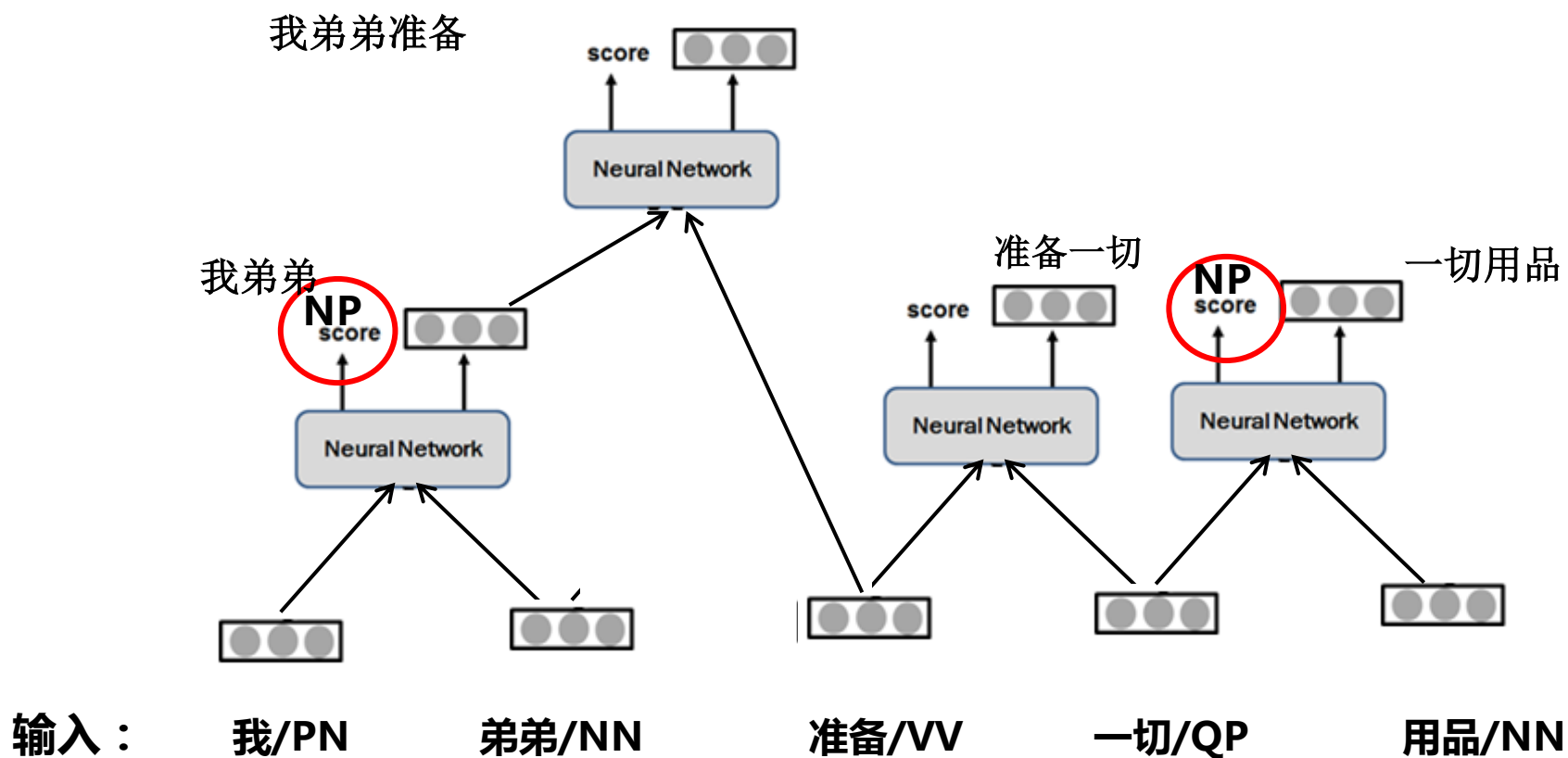
一切/QP

用品/NN

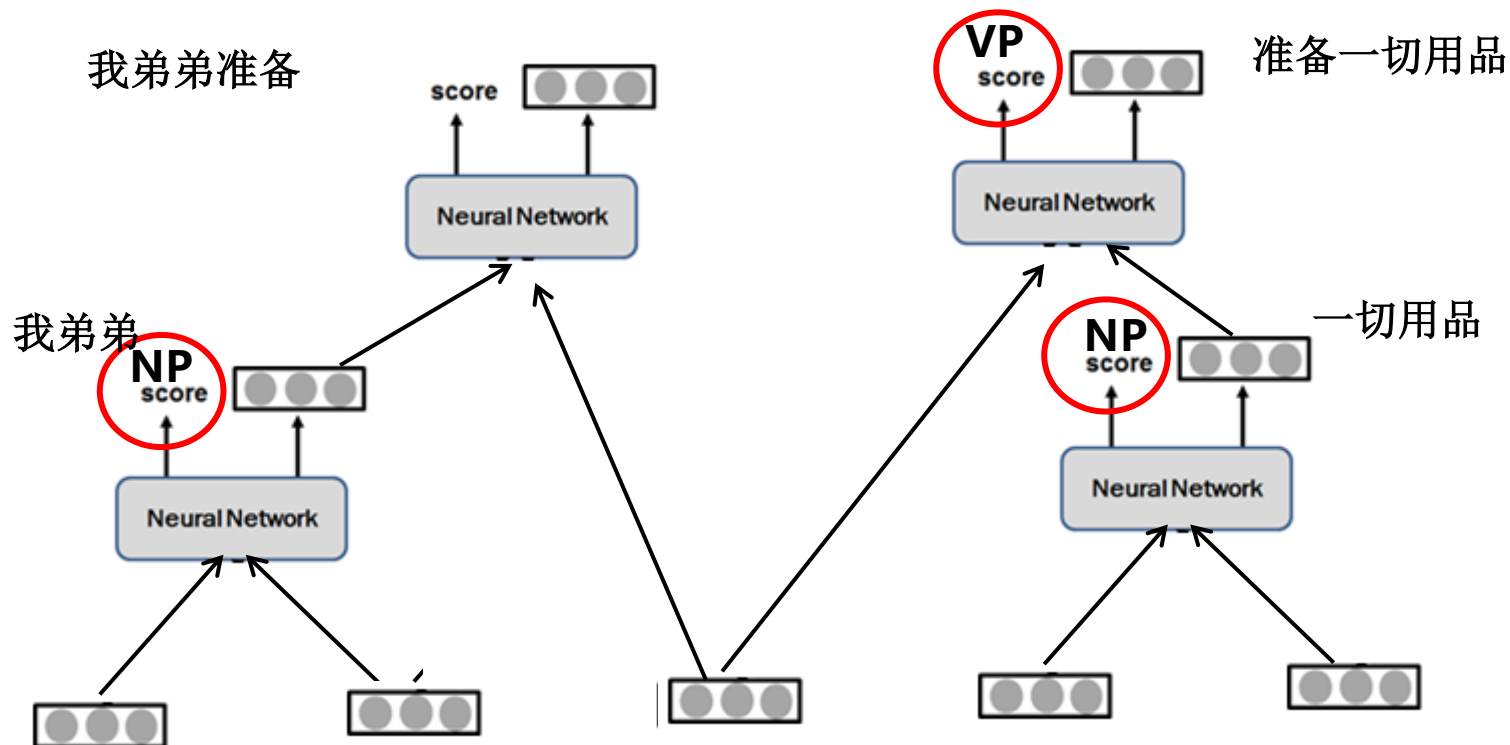
11.1.4.2 神经网络句法分析



11.1.4.2 神经网络句法分析



11.1.4.2 神经网络句法分析



输入：

我/PN

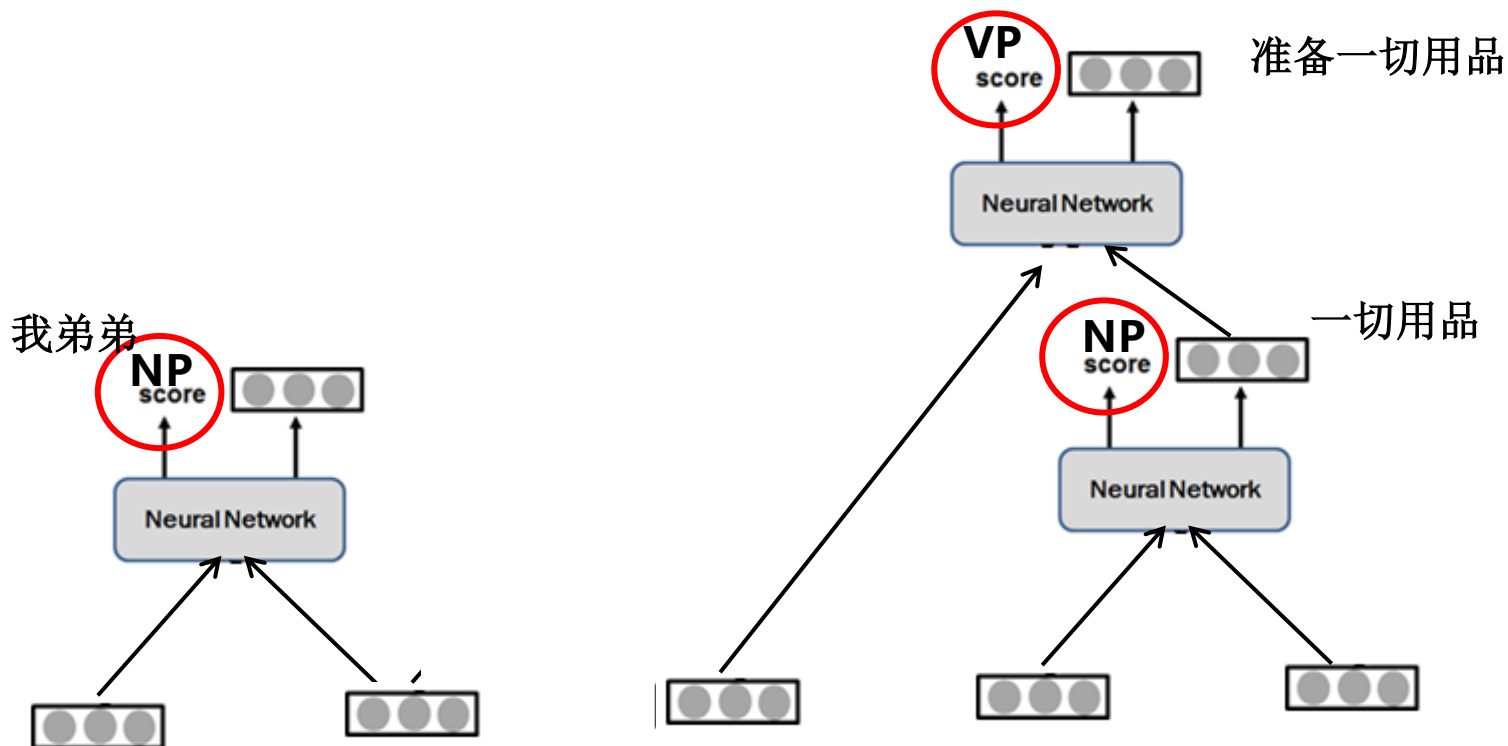
弟弟/NN

准备/VV

一切/QP

用品/NN

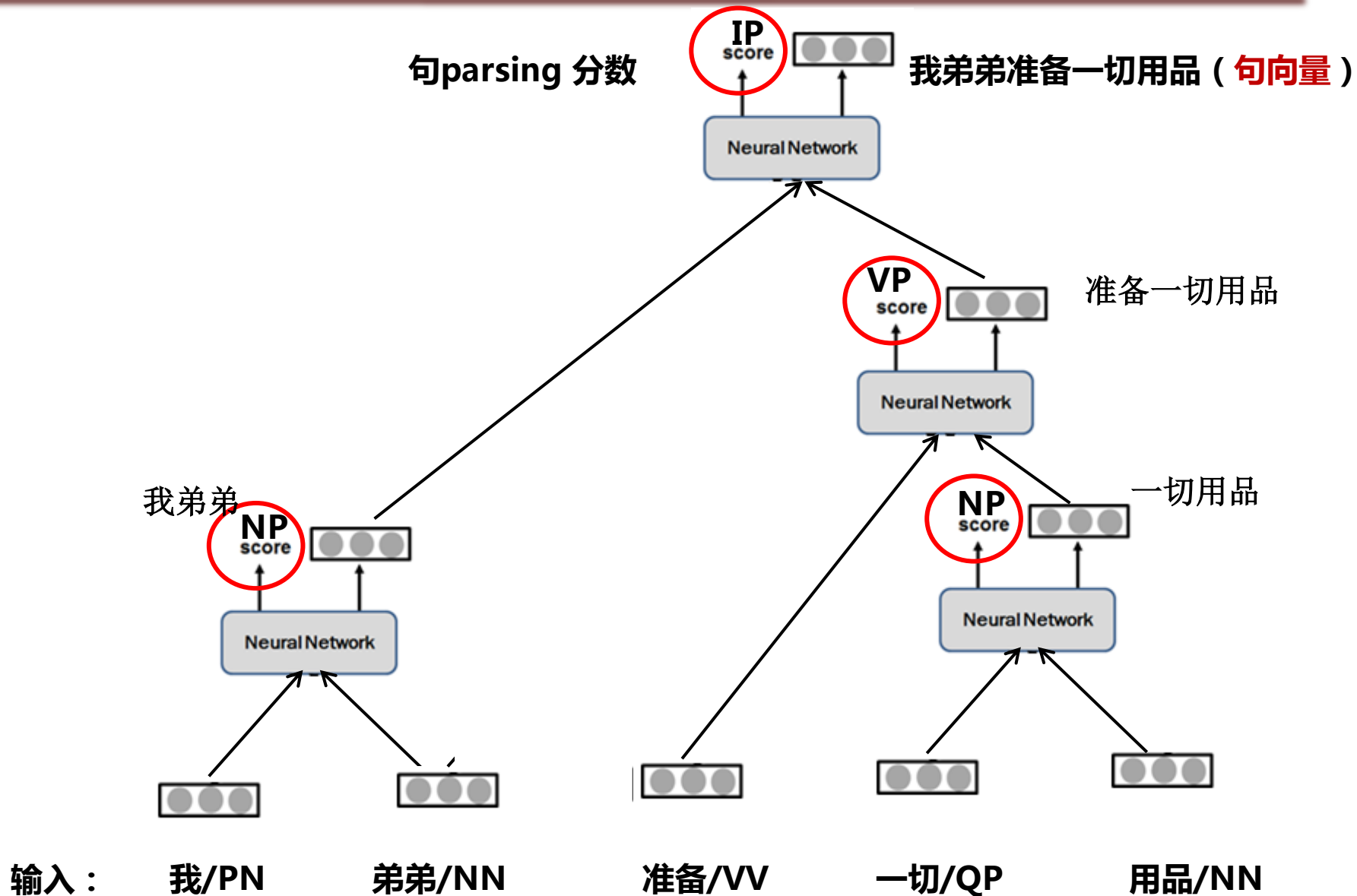
11.1.4.2 神经网络句法分析



输入： 我/PN 弟弟/NN

准备/VV 一切/QP 用品/NN

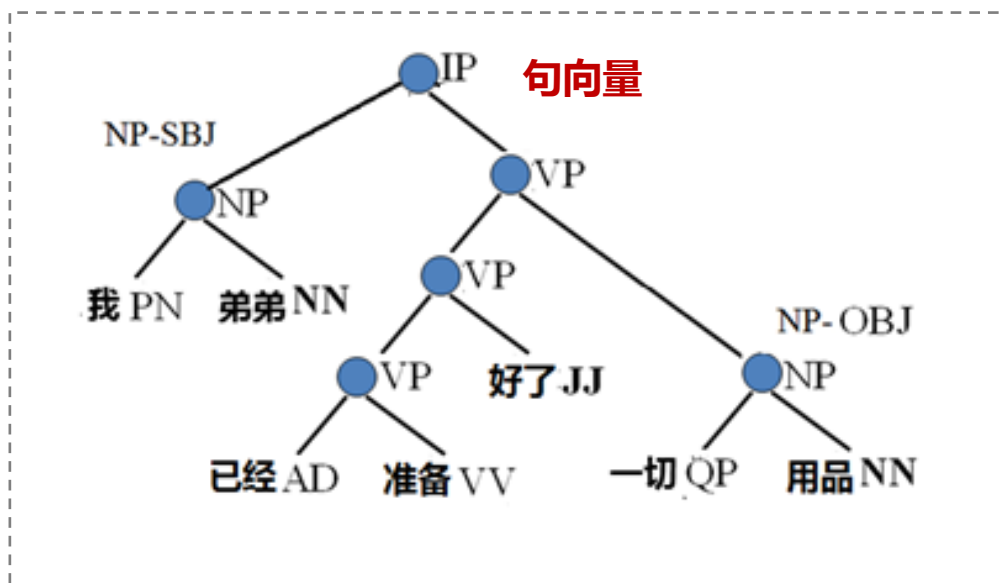
11.1.4.2 神经网络句法分析



11.1.4.2 神经网络句法分析

我 弟弟 已经 准备 好了 一切 用品

短语结构树



完全句法分析-内容提 要

11.1.1 层次分析法

11.1.2 规则法完全句法分析

11.1.3 概率统计法完全句法分析

11.1.4 神经网络法完全句法分析

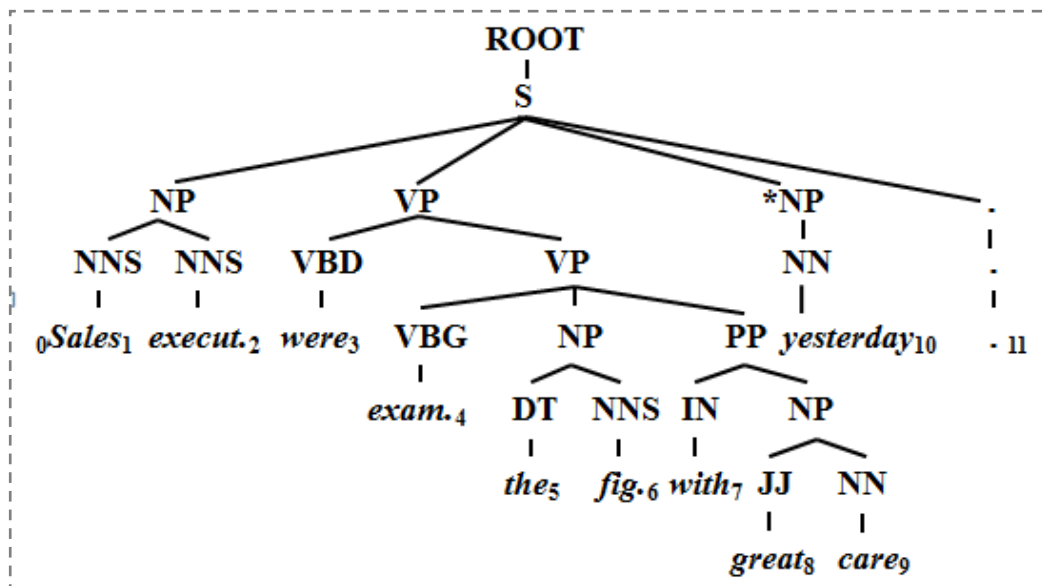
11.1.5 句法分析评价

11.1.5 句法分析评价

分析树中短语标记格式：

分析树中非终结符节点（短语）通常采用如下标记格式：XP-(起始位置：终止位置)。其中，XP为短语名称；(起始位置：终止位置)为该节点的跨越范围，起始位置指该节点所包含的子节点的起始位置，终止位置为该节点所包含的子节点的终止位置。

如：



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

11.1.5 句法分析评价

句法分析器性能指标

- **精度(precision)**：句法分析结果中正确的短语个数所占的比例，即分析结果中与标准分析树（答案）中的短语相匹配的个数占分析结果中所有短语个数的比例，即：

$$P = \frac{\text{分析得到的正确的短语个数}}{\text{分析得到的所有的短语个数}} \times 100\%$$

- **召回率(recall)**：句法分析结果中正确的短语个数占标准分析树中全部短语个数的比例，即：

$$R = \frac{\text{分析得到的正确的短语个数}}{\text{标准树库中(答案)的短语个数}} \times 100\%$$

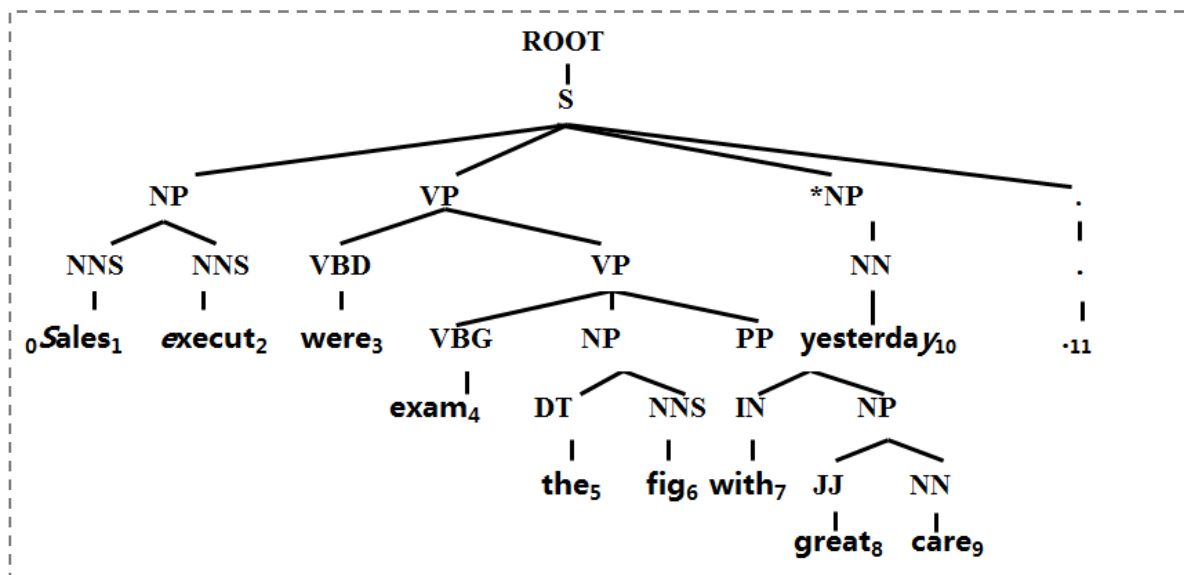
- **F-measure**：

$$F = \frac{(\beta^2 + 1) \times P \times R}{\beta^2 \times P + R} \times 100\% \quad \text{一般地，} \beta = 1, \text{称作 F1 测度。}$$

11.1.5 句法分析评价

例：句 Sales executives were examining the figures with great care yesterday.

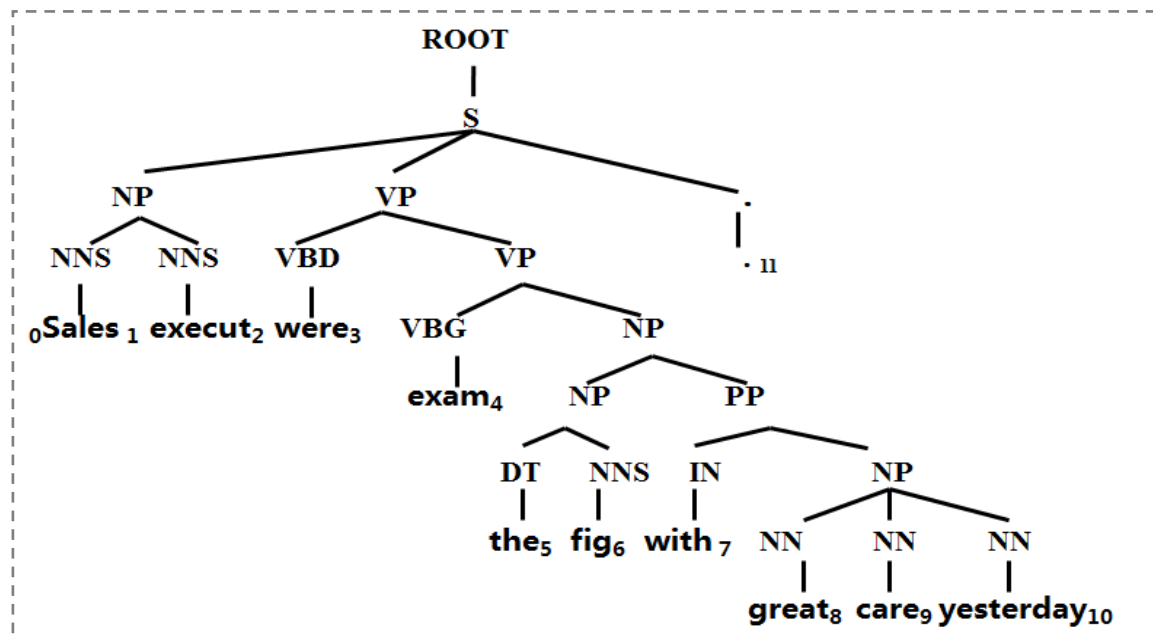
已知图(a)为正确分析树(标准答案)。图(b)为系统分析结果



(a) 句法分析实例 (标准答案)

短语有：S-(0:11), NP-(0:2), VP-(2:9), VP-(3:9), NP-(4:6), PP-(6:9),
NP-(7:9), *NP-(9:10)

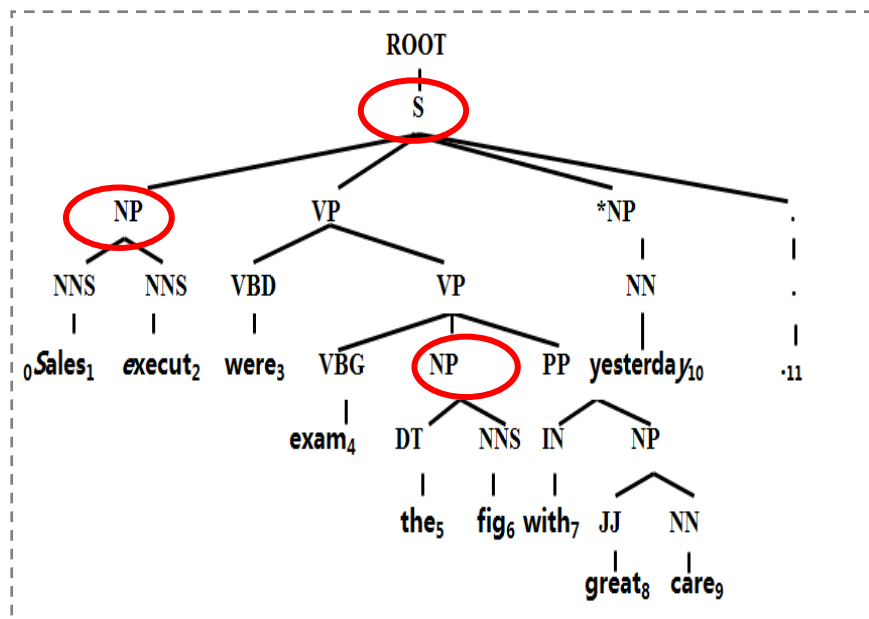
11.1.5 句法分析评价



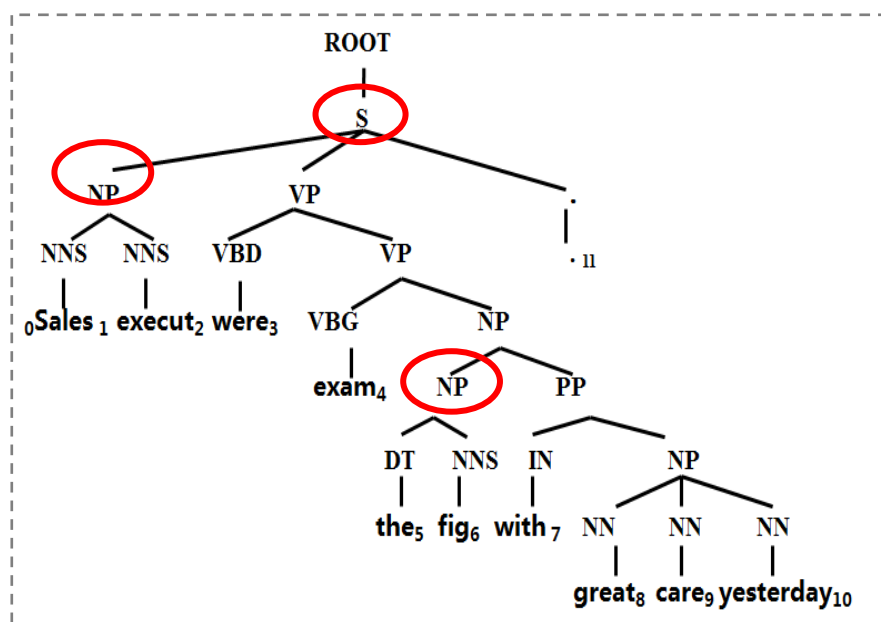
(b)系统分析结果

短语有：S-(0:11), NP-(0:2), VP-(2:10), VP-(3:10), NP-(4:10), NP-(4:6) ,
PP-(6:10), NP-(7:10)

11.1.5 句法分析评价



(a) 句法分析实例 (标准答案)



(b) 系统分析结果

系统分析结果只有3个短语与标准答案完全一样

$$\text{Precision} = \frac{3}{8} \times 100\% = 37.5\%$$

$$\text{Recall} = \frac{3}{8} \times 100\% = 37.5\%$$

$$F = 18.7\%$$

开源的短语句法分析器

✧ **Berkeley Parser**

<http://nlp.cs.berkeley.edu/Main.html#Parsing>

✧ **Stanford Parser**

<http://nlp.stanford.edu/downloads/lex-parser.shtml>

✧ **Collins Parser**

<http://people.csail.mit.edu/mcollins/code.html>

✧ **Bikel Parser**

<http://www.cis.upenn.edu/~dbikel/software.html#stat-parser>

✧ **Charniak Parser**

<http://www.cs.brown.edu/people/ec/#software>

✧ **Oboe Parser**(可执行程序)

<http://www.openpr.org.cn/index.php/NLP-Toolkit-for-Natural-Language-Processing/>

参考文献：

宗成庆，统计自然语言处理（第2版）课件

徐志明，概率句法分析（课件），哈工大语言技术中心

关毅，第七章 句法分析技术（课件）哈工大语言技术中心

詹卫东，第八章 句法分析（二），北京大学

孙薇薇，依存句法分析，北京大学

<https://www.csie.ntu.edu.tw/~yvchen/f106-adl/syllabus.html>

在此表示感谢！

谢谢各位！

