



数据科学与计算机学院
School of Data and Computer Science

自然语言处理

Natural Language Processing

权小军 教授

中山大学数据科学与计算机学院

quanxj3@mail.sysu.edu.cn

课程回顾

中文分词

- 词法分析 {
 1. 自动分词
 2. 命名实体识别
 3. 词性标注

中文分词

- 词法分析 {
 1. 自动分词
 2. 命名实体识别
 3. 词性标注
- 语言模型 {
 1. 统计语言模型
 2. 神经网络语言模型

中文分词

- 词法分析 {
 1. 自动分词
 2. 命名实体识别
 3. 词性标注
- 语言模型 {
 1. 统计语言模型
 2. 神经语言模型
- 句法分析

句法分析

1、概述

概述

- 句法分析是自然语言处理中的基础性工作，它分析句子的句法结构（主谓宾结构）和词汇间的依存关系（并列，从属等）；

概述

- 句法分析是自然语言处理中的基础性工作，它分析句子的句法结构（主谓宾结构）和词汇间的依存关系（并列，从属等）；

概述

- 句法分析是自然语言处理中的基础性工作，它分析句子的句法结构（主谓宾结构）和词汇间的依存关系（并列，从属等）；
- 句法分析可以为语义分析、情感倾向、观点抽取等NLP应用场景打下坚实的基础。

概述

- 句法分析是自然语言处理中的基础性工作，它分析**句子的句法结构**（主谓宾结构）和**词汇间的依存关系**（并列，从属等）；
- 句法分析可以为语义分析、情感倾向、观点抽取等NLP应用场景打下坚实的基础。

句法分析是语言理解的重要基础

概述

句法分析不是自然语言处理任务的最终目标，但它往往是实现最终目标的一个关键环节！

概述

- 句法分析是自然语言处理中的基础性工作，它分析句子的句法结构（主谓宾结构）和词汇间的依存关系（并列，从属等）；
- 句法分析可以为语义分析、情感倾向、观点抽取等NLP应用场景打下坚实的基础。

例如：习近平结束对尼泊尔国事访问回到北京

这句话表达了什么意思？

概述

- 句法分析是自然语言处理中的基础性工作，它分析**句子的句法结构**（主谓宾结构）和**词汇间的依存关系**（并列，从属等）；
- 句法分析可以为语义分析、情感倾向、观点抽取等NLP应用场景打下坚实的基础。

I can't love you

概述

- 句法分析是自然语言处理中的基础性工作，它分析句子的句法结构（主谓宾结构）和词汇间的依存关系（并列，从属等）；
- 句法分析可以为语义分析、情感倾向、观点抽取等NLP应用场景打下坚实的基础。

*I can't love you **more**.*

概述

□ 分类：句法分析分为两类：

- 分析句子的主谓宾定状补的句法结构；

概述

□ 分类：句法分析分为两类：

- 分析句子的主谓宾定状补的句法结构；
- 分析词汇间的依存关系，如并列、从属、比较、递进等。

概述

□ 分类：句法分析分为两类：

- 分析句子的主谓宾定状补的句法结构；
- 分析词汇间的依存关系，如并列、从属、比较、递进等。

习近平结束对尼泊尔国事访问回到北京

概述

□ 任务类型:

❖ 短语结构分析(Phrase parsing)

- 完全句法分析(Full parsing)：以获取整个句子的句法结构为目的；
- 局部句法分析(Partial parsing)：以获得局部成分为目的；

概述

□ 任务类型:

❖ 短语结构分析(Phrase parsing)

- 完全句法分析(Full parsing)：以获取整个句子的句法结构为目的；
- 局部句法分析(Partial parsing)：以获得局部成分为目的；

❖ 依存句法分析(Dependency parsing)

通过分析语言单位内成分之间的依存关系揭示其句法结构

1、概述

2、短语结构分析

短语结构分析

□ 句法分析的例子

例子：他还提出一系列具体措施的政策要点。

短语结构分析

□ 句法分析的例子

例子：他还提出一系列具体措施的政策要点。

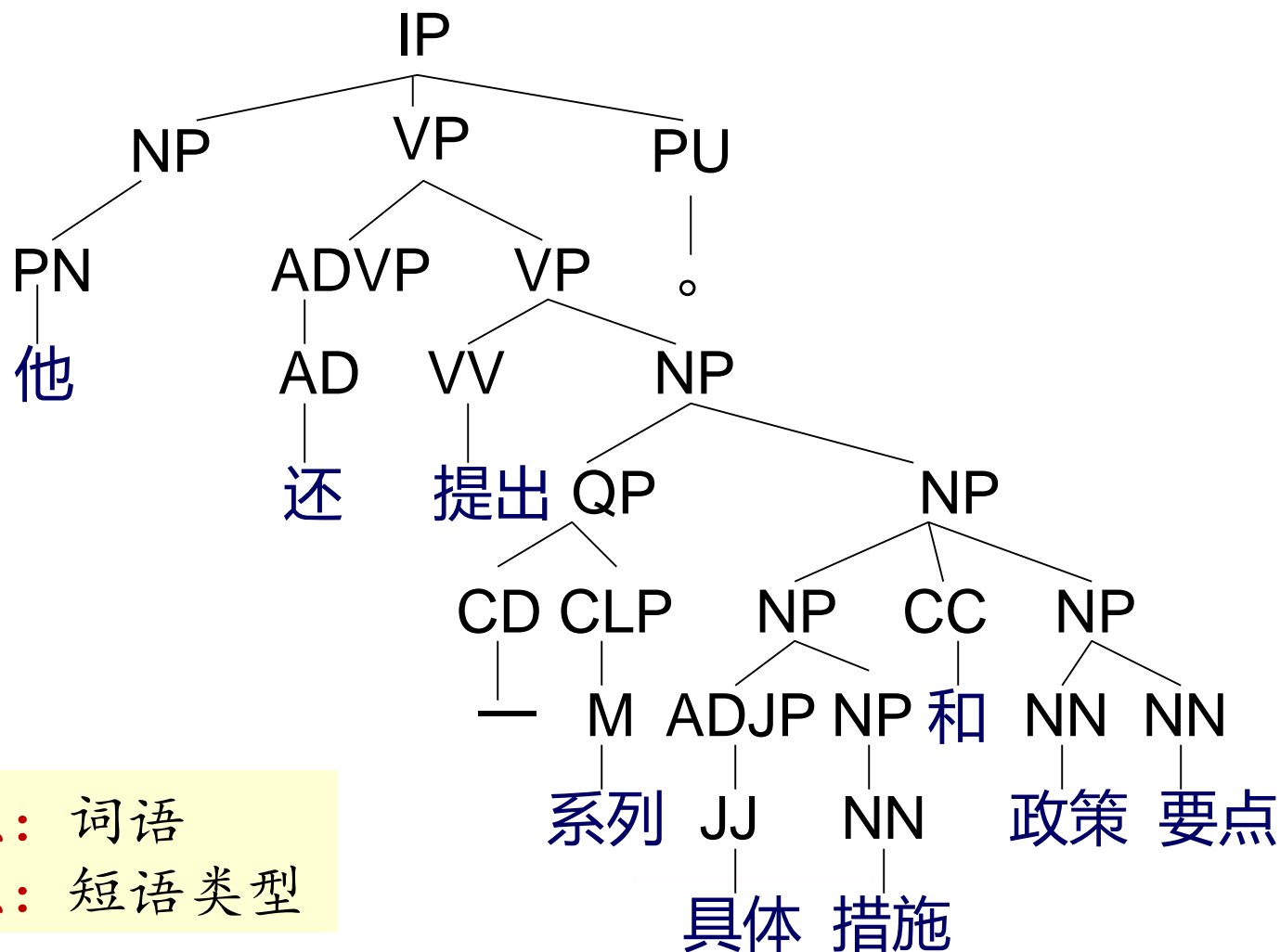
短语结构分析

括号嵌套表示:

(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 。))

短语结构分析

树状表示:



- 叶子节点：词语
- 内部节点：短语类型

短语结构分析

- 符号解释:
- NP: 名词短语
 - VP: 动词短语
 - PU: 断句符, 通常是句号、问号、感叹号等标点符号
 - PP: 介词短语
 - CP: 由‘的’构成的表示修饰性关系的短语
 - ADVP: 副词短语
 - ADJP: 形容词短语
 - DP: 限定词短语
 - QP: 量词短语
 - NN: 常用名词
 - NT: 时间名词
 - PN: 代词
 - VV: 动词
 -

短语结构分析

- 目标：实现高正确率、高鲁棒性(robustness)、高速度的自动句法分析过程；

短语结构分析

- **目标**: 实现高正确率、高鲁棒性(robustness)、高速度的自动句法分析过程;
- **困难**: 自然语言中存在大量的复杂的结构歧义(structural ambiguity);

短语结构分析

□ 结构歧义

例如：(1) I saw a boy **in the park**.

[I saw a boy] in the park.

I saw a [boy in the park].

(2) I saw a boy **in the park** **with a telescope**.

(3) I saw a boy swimming **on the bridge**.

(4) 关于鲁迅的文章。

(5) 把重要的书籍和手稿带走了。

短语结构分析

□ 基本方法和开源的句法分析器：

○ 基于CFG规则的分析方法

- CFG: Context-Free Grammar (上下文无关文法)
- 代表：线图分析法(chart parsing)

○ 基于 PCFG 的分析方法

- PCFG: Probabilistic Context-Free Grammar (概率上下文无关文法)

短语结构分析

□ 基本方法和开源的句法分析器：

○ 基于CFG规则的分析方法

- CFG: Context-Free Grammar (上下文无关文法)
- 代表：线图分析法(chart parsing)

○ 基于 PCFG 的分析方法

- PCFG: Probabilistic Context-Free Grammar (概率上下文无关文法)

1、概述

2、短语结构分析

a) 上下文无关文法

上下文无关文法 (CFG)

- CFG由一系列规则组成，每条规则给出了语言中的符号可以被组织或排列在一起的规则，以及符号和单词构成的字典。

上下文无关文法 (CFG)

- CFG由一系列规则组成，每条规则给出了语言中的某些符号可以被组织或排列在一起的方式。

符号被分成两类：

- 终结点(叶子节点)：就是指单词，例如 book；
- 非终结点(内部节点)：句法标签，例如 NP 或者 NN；

规则是由一个“ \rightarrow ”连接的表达式：

- 左侧：只有一个 non-terminal；
- 右侧：是一个由符号组成的序列；

上下文无关文法 (CFG)

CFG示例:

□ 符号:

- 终结点: rat, the, ate, cheese;
- 非终结点: S, NP, VP, DT, VBD, NN;

□ 规则:

$S \rightarrow NP VP$

$NP \rightarrow DT NN$

$VP \rightarrow VBD NP$

$DT \rightarrow the$

$NN \rightarrow rat$

$NN \rightarrow cheese$

$VBD \rightarrow ate$

上下文无关文法 (CFG)

下面我们试着利用上面这个CFG来生成句子，通常总是以**S**做为开始：

1. S

上下文无关文法 (CFG)


下面我们试着利用上面这个CFG来生成句子，通常总是以S做为开始：

1. S

2. 应用规则 $S \rightarrow NP VP$ ，则有

上下文无关文法 (CFG)

下面我们试着利用上面这个CFG来生成句子，通常总是以S做为开始：

1. S
 2. 应用规则 $S \rightarrow NP VP$ ，则有
 3. NP VP
- 

上下文无关文法 (CFG)

下面我们试着利用上面这个CFG来生成句子，通常总是以S做为开始：

1. S

2. 应用规则 $S \rightarrow NP VP$ ，则有

3. NP VP



4. 应用规则 $NP \rightarrow DT NN$ ，则有

上下文无关文法 (CFG)

下面我们试着利用上面这个CFG来生成句子，通常总是以S做为开始：

1. S

2. 应用规则 $S \rightarrow NP VP$ ，则有

3. NP VP

4. 应用规则 $NP \rightarrow DT NN$ ，则有

5. DT NN VP



上下文无关文法 (CFG)

下面我们试着利用上面这个CFG来生成句子，通常总是以S做为开始：

1. S

2. 应用规则 $S \rightarrow NP VP$ ，则有

3. NP VP

4. 应用规则 $NP \rightarrow DT NN$ ，则有

5. DT NN VP

6. 应用规则 $DT \rightarrow the$, $NN \rightarrow rat$ ，则有

上下文无关文法 (CFG)

下面我们试着利用上面这个CFG来生成句子，通常总是以S做为开始：

1. S

2. 应用规则 $S \rightarrow NP VP$ ，则有

3. NP VP

4. 应用规则 $NP \rightarrow DT NN$ ，则有

5. DT NN VP

6. 应用规则 $DT \rightarrow the$, $NN \rightarrow rat$ ，则有

7. the rat VP



上下文无关文法 (CFG)

下面我们试着利用上面这个CFG来生成句子，通常总是以S做为开始：

8. 应用规则 $VP \rightarrow VBD\ NP$ ，则有

9. the rat VBD NP

上下文无关文法 (CFG)

下面我们试着利用上面这个CFG来生成句子，通常总是以S做为开始：

8. 应用规则 $VP \rightarrow VBD\ NP$ ，则有

9. the rat VBD NP

10. 应用规则 $VBD \rightarrow ate$ ，则有

11. the rat ate NP

上下文无关文法 (CFG)

下面我们试着利用上面这个CFG来生成句子，通常总是以S做为开始：

8. 应用规则 $VP \rightarrow VBD\ NP$ ，则有

9. the rat VBD NP

10. 应用规则 $VBD \rightarrow ate$ ，则有

11. the rat ate NP

12. 应用规则 $NP \rightarrow DT\ NN$ ，则有

13. the rat ate DT NN

上下文无关文法 (CFG)

下面我们试着利用上面这个CFG来生成句子，通常总是以S做为开始：

8. 应用规则 $VP \rightarrow VBD\ NP$ ，则有

9. the rat VBD NP

10. 应用规则 $VBD \rightarrow ate$ ，则有

11. the rat ate NP

12. 应用规则 $NP \rightarrow DT\ NN$ ，则有

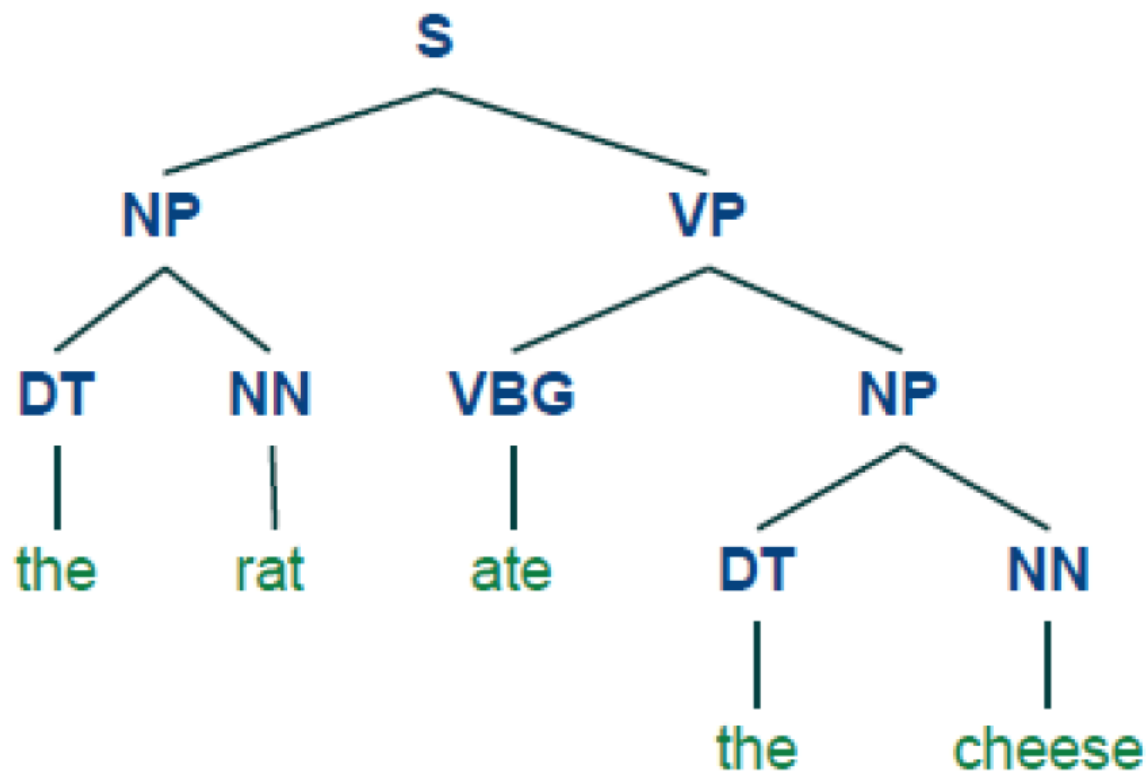
13. the rat ate DT NN

14. 应用规则 $DT \rightarrow the$, $NN \rightarrow cheese$ ，则有

15. the rat ate the cheese

上下文无关文法 (CFG)

上述过程用树表示非常方便，terminals是叶子节点，而non-terminals是非叶子节点：



基于上下文无关文法的句法分析

基于上下文无关文法（CFG）的句法分析是指基于预定义的语法规则，为输入语句生成恰当的句法树，要求该树：

- ✓ 符合给定语法规则；

基于上下文无关文法的句法分析

基于上下文无关文法（CFG）的句法分析是指基于预定义的语法规则，为输入语句生成恰当的句法树，要求该树：

- ✓ 符合给定语法规则；
- ✓ 终（叶子）节点包含所有的词；

基于上下文无关文法的句法分析

基于上下文无关文法（CFG）的句法分析是指基于预定义的语法，为输入语句生成恰当的句法树，要求该树：

- ✓ 符合给定语法；
- ✓ 叶子节点包含所有的词；

符合这样条件的树通常有很多！

1、概述

2、短语结构分析

a) 上下文无关文法

b) 线图分析法

线图分析法

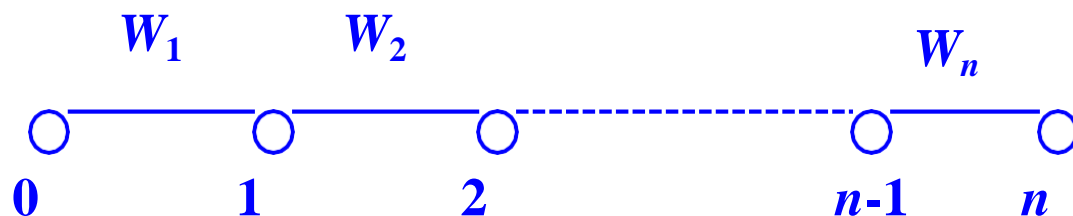
□ 三种策略

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

线图分析法

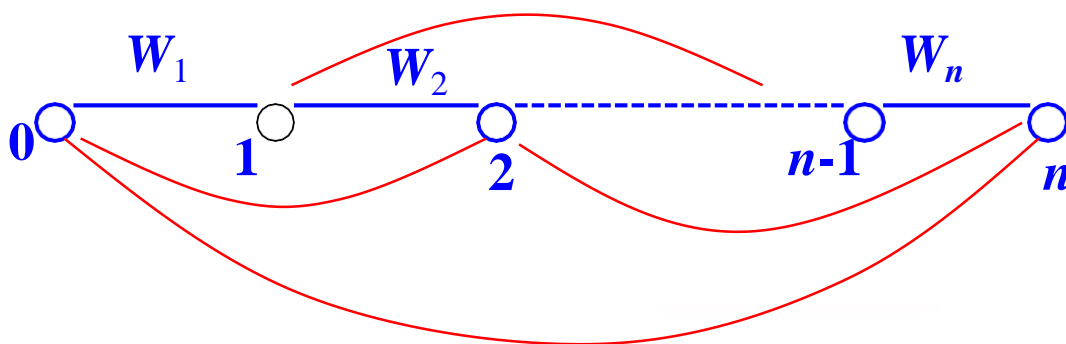
□ 自底向上的线图分析算法

- 给定一组 CFG 规则: $XP \rightarrow \alpha_1 \dots \alpha_n$ ($n \geq 1$)
- 给定一个句子的词性序列: $S = W_1 W_2 \dots W_n$
- 构造一个线图: 一组结点和边的集合;



线图分析法

执行：查看任意相邻几条边上的词性串是否与某条规则的右部相同，如果相同，则增加一条新的边跨越原来相应的边，新增加边上的标记为这条规则的头(左部)。重复这个过程，直到没有新的边产生。



线图分析法

例：G (S): $S \rightarrow NP \ VP,$

$VP \rightarrow V \ NP,$

$PP \rightarrow Prep \ NP$

$NP \rightarrow Det \ N$

$VP \rightarrow VP \ PP$

输入句子: the boy hits the dog with a rod

线图分析法

例：G (S): $S \rightarrow NP \ VP$,

$VP \rightarrow V \ NP$,

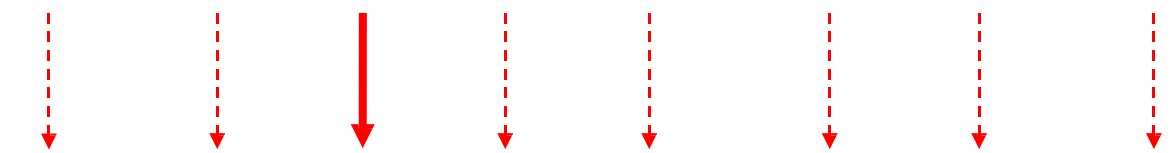
$PP \rightarrow Prep \ NP$

$NP \rightarrow Det \ N$

$VP \rightarrow VP \ PP$

输入句子: the boy hits the dog with a rod

①形态分析: the boy hit the dog with a rod



线图分析法

例：G (S): $S \rightarrow NP \ VP$,

$VP \rightarrow V \ NP$,

$PP \rightarrow Prep \ NP$

$NP \rightarrow Det \ N$

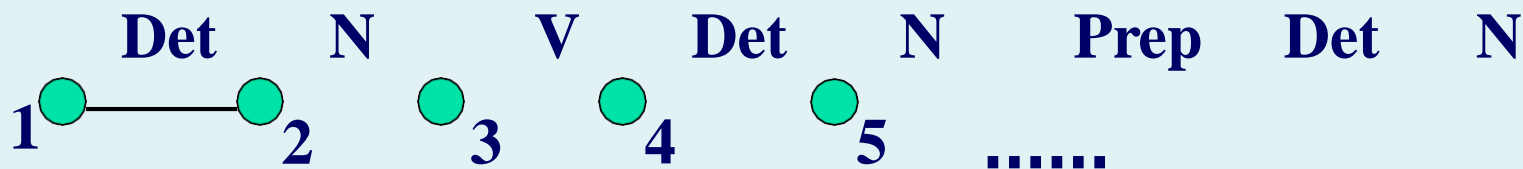
$VP \rightarrow VP \ PP$

输入句子: the boy hits the dog with a rod

①形态分析: the boy hit the dog with a rod

②词性标注: Det N V Det N Prep Det N

线图分析法



(1) $S \rightarrow NP VP$

(4) $VP \rightarrow V NP$

(2) $NP \rightarrow Det N$

(5) $PP \rightarrow Prep NP$

(3) $VP \rightarrow VP PP$

线图分析法



(1) $S \rightarrow NP VP$

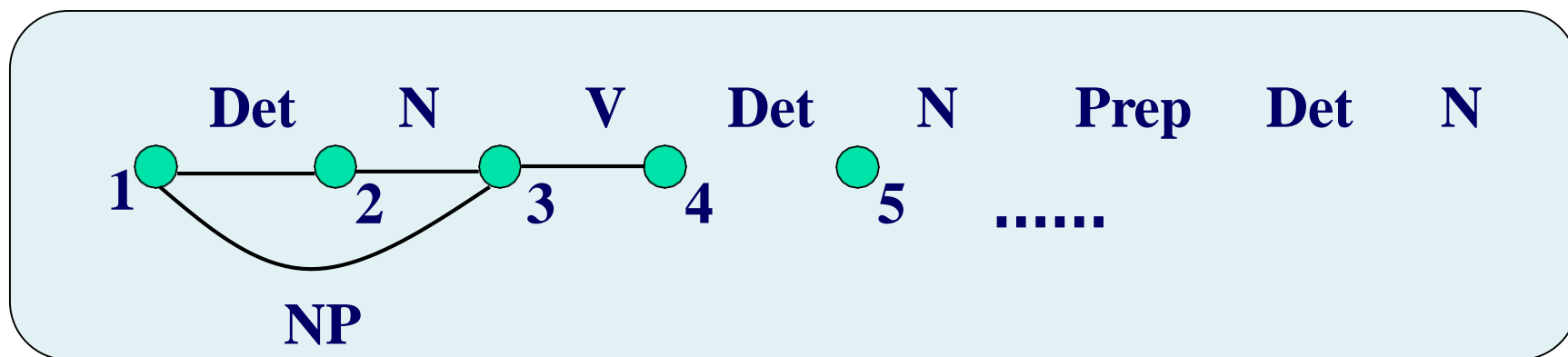
(2) $NP \rightarrow Det N$

(3) $VP \rightarrow VP PP$

(4) $VP \rightarrow V NP$

(5) $PP \rightarrow Prep NP$

线图分析法



(1) $S \rightarrow NP VP$

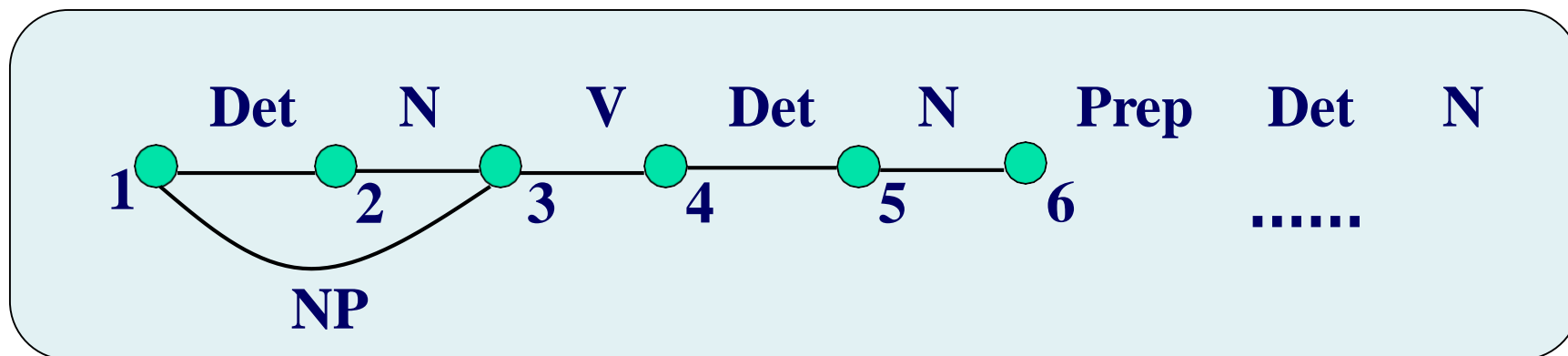
(4) $VP \rightarrow V NP$

(2) $NP \rightarrow Det N$

(5) $PP \rightarrow Prep NP$

(3) $VP \rightarrow VP PP$

线图分析法



(1) $S \rightarrow NP VP$

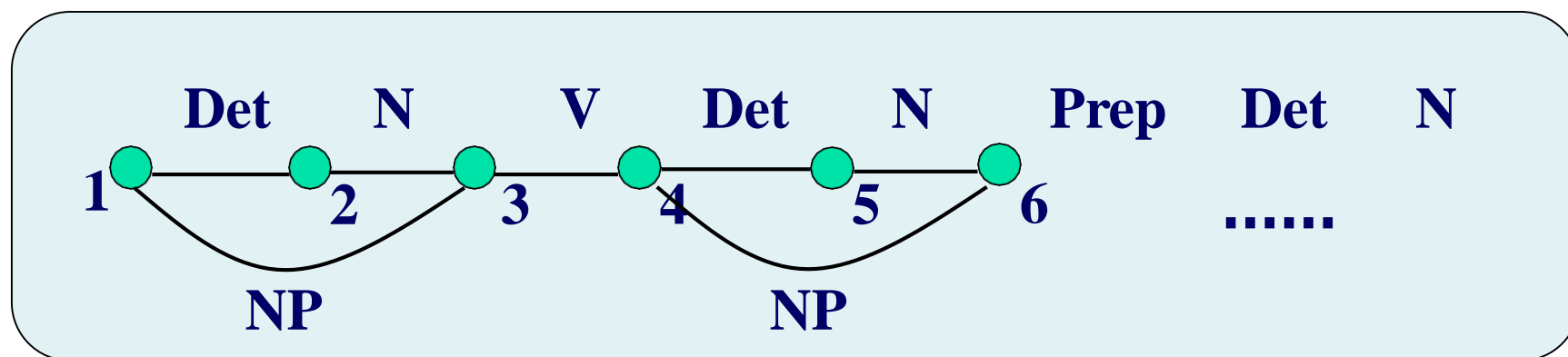
(2) $NP \rightarrow Det N$

(3) $VP \rightarrow VP PP$

(4) $VP \rightarrow V NP$

(5) $PP \rightarrow Prep NP$

线图分析法



(1) $S \rightarrow NP VP$

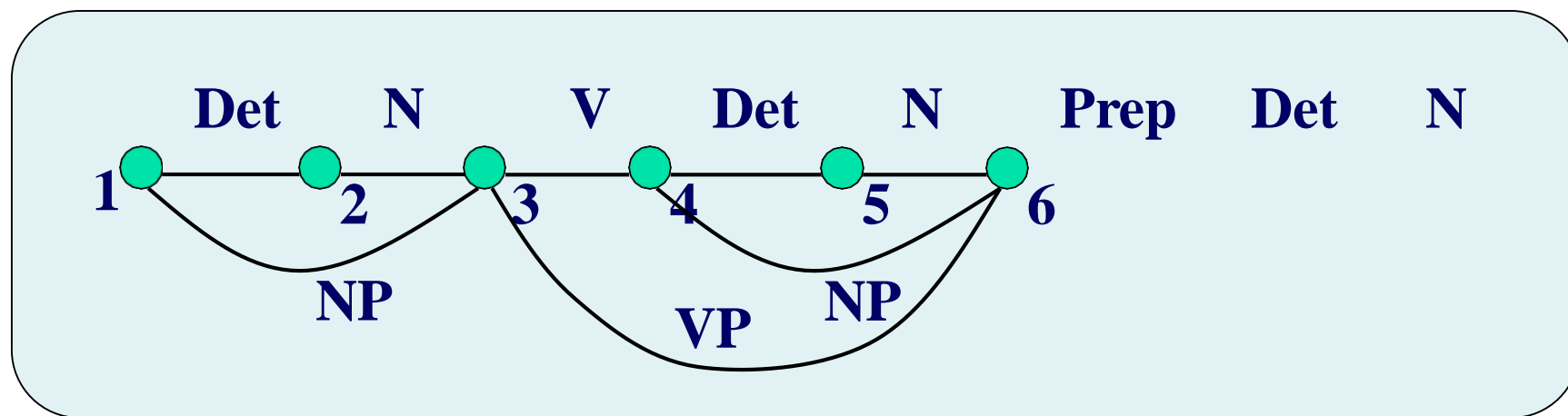
(2) $NP \rightarrow Det N$

(3) $VP \rightarrow VP PP$

(4) $VP \rightarrow V NP$

(5) $PP \rightarrow Prep NP$

线图分析法



(1) $S \rightarrow NP VP$

(4) $VP \rightarrow V NP$

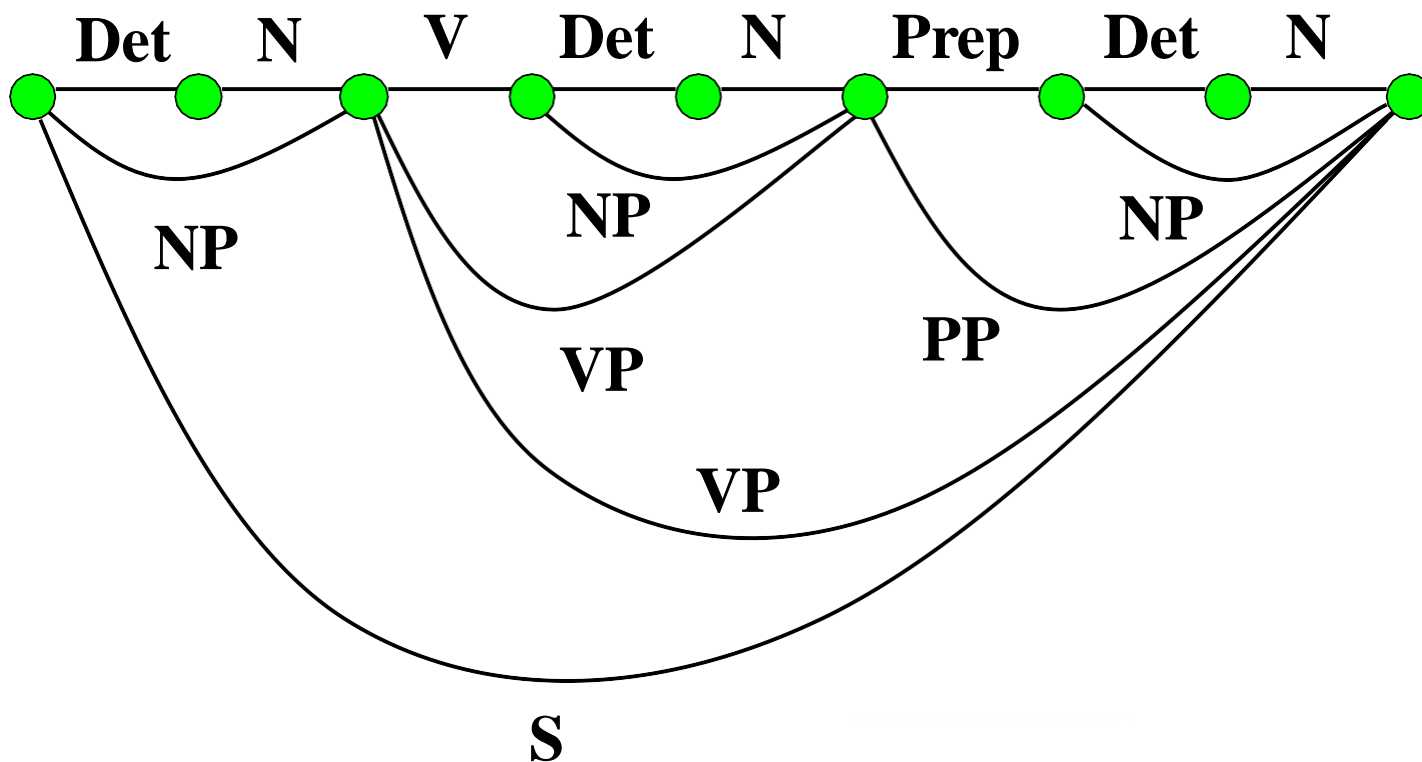
(2) $NP \rightarrow Det N$

(5) $PP \rightarrow Prep NP$

(3) $VP \rightarrow VP PP$

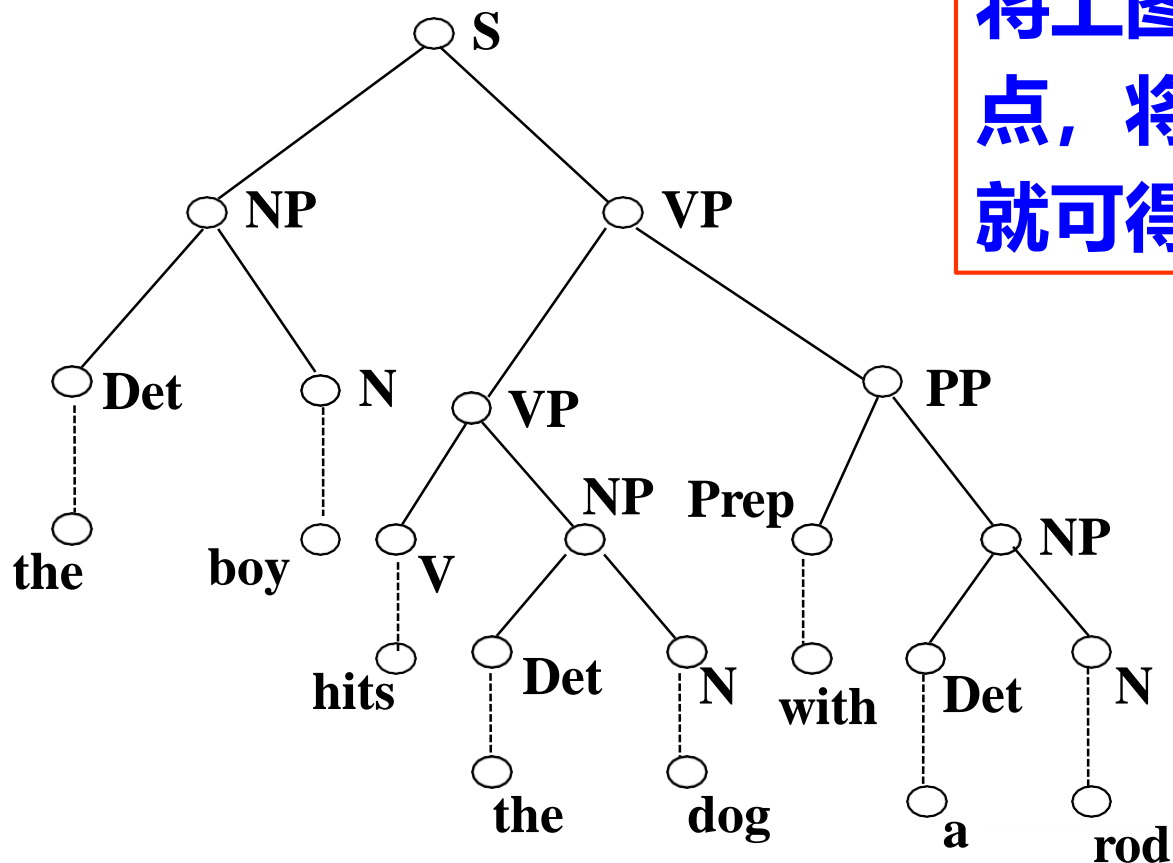
线图分析法

最后分析结果:



线图分析法

将上图中的边改为结点，将结点改为边，就可得到一棵句法树



1、概述

2、短语结构分析

a) 上下文无关文法

b) 线图分析法

c) 概率上下文无关文法

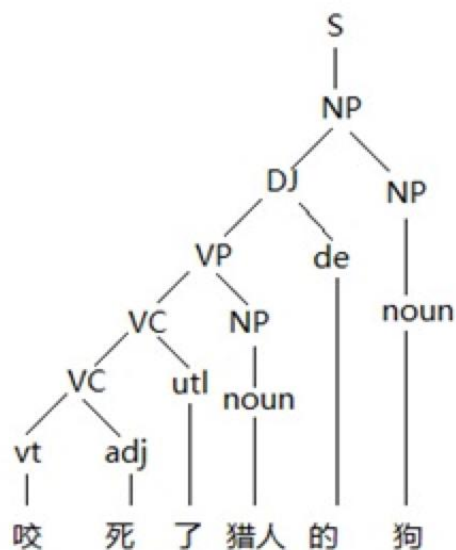
概率上下文无关文法

- CFG赋予了语言一种层次化的结构。但是根据一个CFG构建语法分析树，往往不止一个；

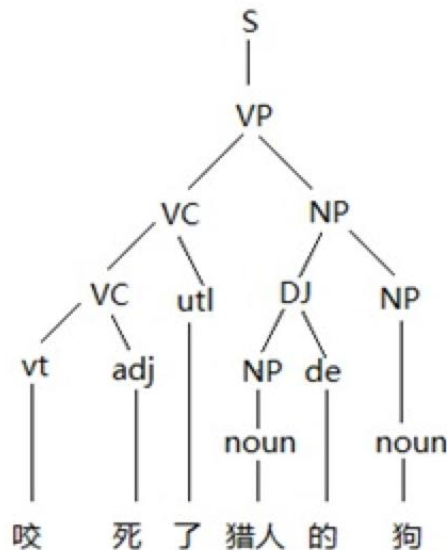
概率上下文无关文法

- CFG赋予了语言一种层次化的结构。但是根据一个CFG构建语法分析树，往往不止一个；

例如“咬死了猎人的狗”，有如下分析树：



(1)



(2)

概率上下文无关文法

- 对于可能产生多种语法分析结果的问题，我们该如何应对呢？

概率上下文无关文法

- 对于可能产生多种语法分析结果的问题，我们该如何应对呢？
- 引入概率上下文无关文法（PCFG, Probabilistic context-free grammar）：给每棵树计算一个概率！

概率上下文无关文法

□ PCFG 规则

形式: $A \rightarrow \alpha \quad [p]$

概率上下文无关文法

□ PCFG 规则

形式: $A \rightarrow \alpha$ $[p]$

- ▶ $NP \rightarrow DT\ NN$ $[p = 0.45]$
- ▶ $NN \rightarrow \text{leprechaun}$ $[p = 0.0001]$

概率上下文无关文法

□ PCFG 规则

形式: $A \rightarrow \alpha \quad [p]$

约束: $\sum_{\alpha} p(A \rightarrow \alpha) = 1$

概率上下文无关文法

□ PCFG 规则

形式: $A \rightarrow \alpha \quad [p]$

约束: $\sum_{\alpha} p(A \rightarrow \alpha) = 1$

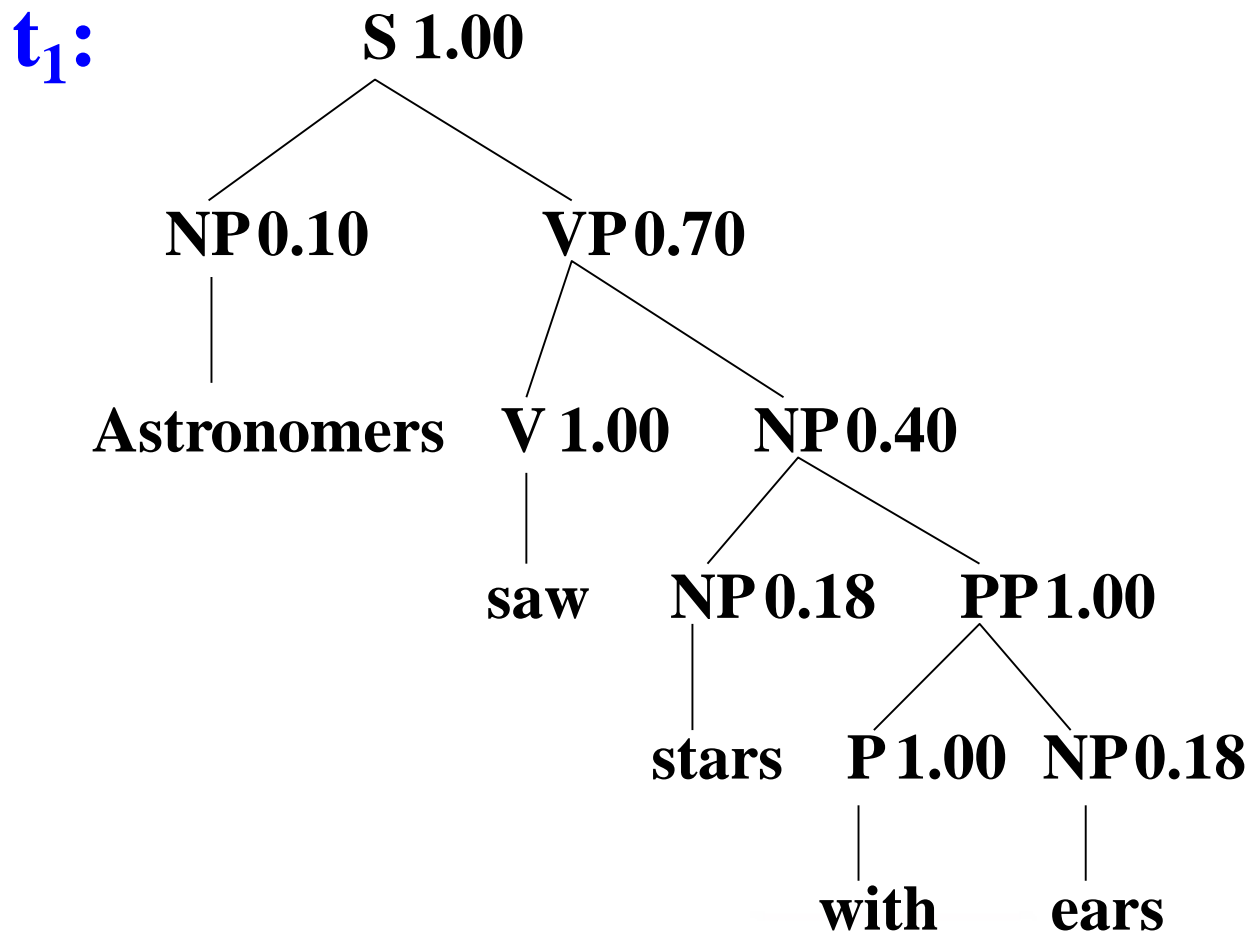
例如: $\left. \begin{array}{l} \text{NP} \rightarrow \text{NN NN}, 0.60 \\ \text{NP} \rightarrow \text{NN CC NN}, 0.40 \end{array} \right\} \sum p = 1$

概率上下文无关文法

□ 例-1:	$S \rightarrow NP VP$, [1.00]	$NP \rightarrow NP PP$, [0.40]
	$NP \rightarrow \text{astronomers}$, [0.10]	
	$NP \rightarrow \text{ears}$, [0.18]	$NP \rightarrow \text{saw}$, [0.04]
	$NP \rightarrow \text{stars}$, [0.18]	$NP \rightarrow \text{telescopes}$, [0.10]
	$PP \rightarrow P NP$, [1.00]	$P \rightarrow \text{with}$, [1.00]
	$VP \rightarrow V NP$, [0.70]	$VP \rightarrow VP PP$, [0.30]
	$V \rightarrow \text{saw}$, [1.00]	

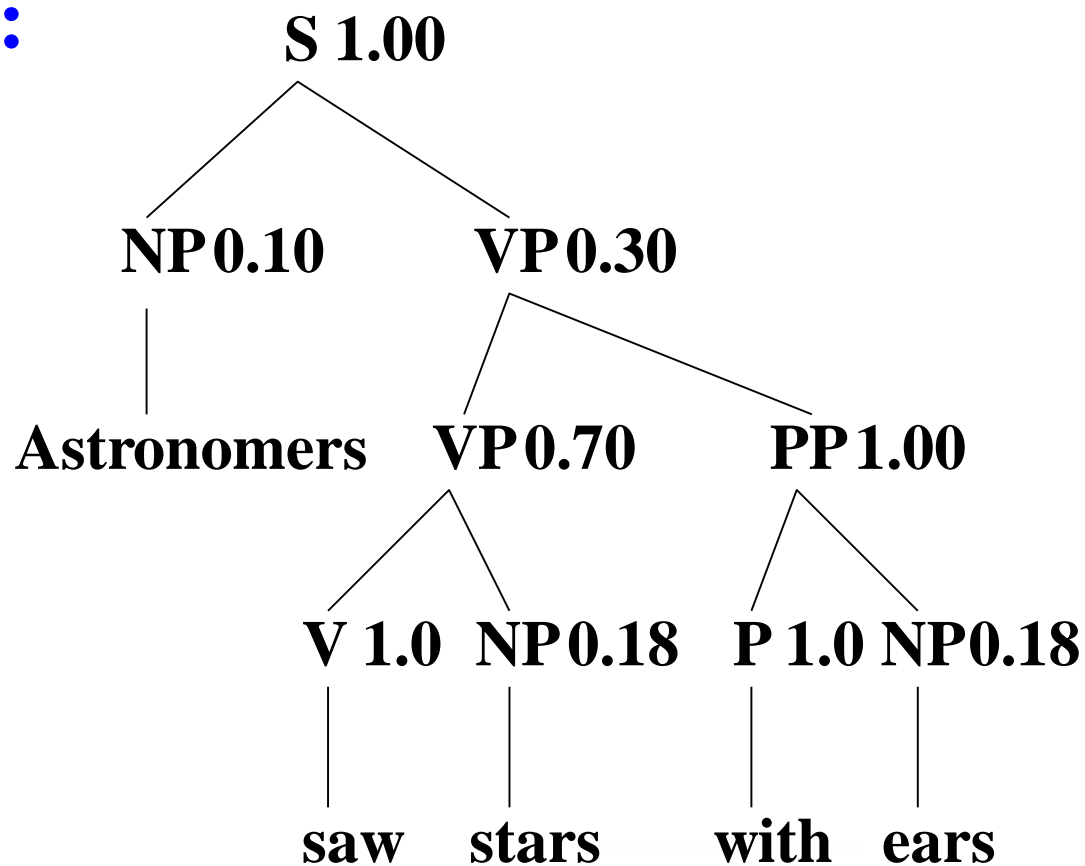
给定句子 S: *Astronomers saw stars with ears.*

概率上下文无关文法



概率上下文无关文法

t_2 :

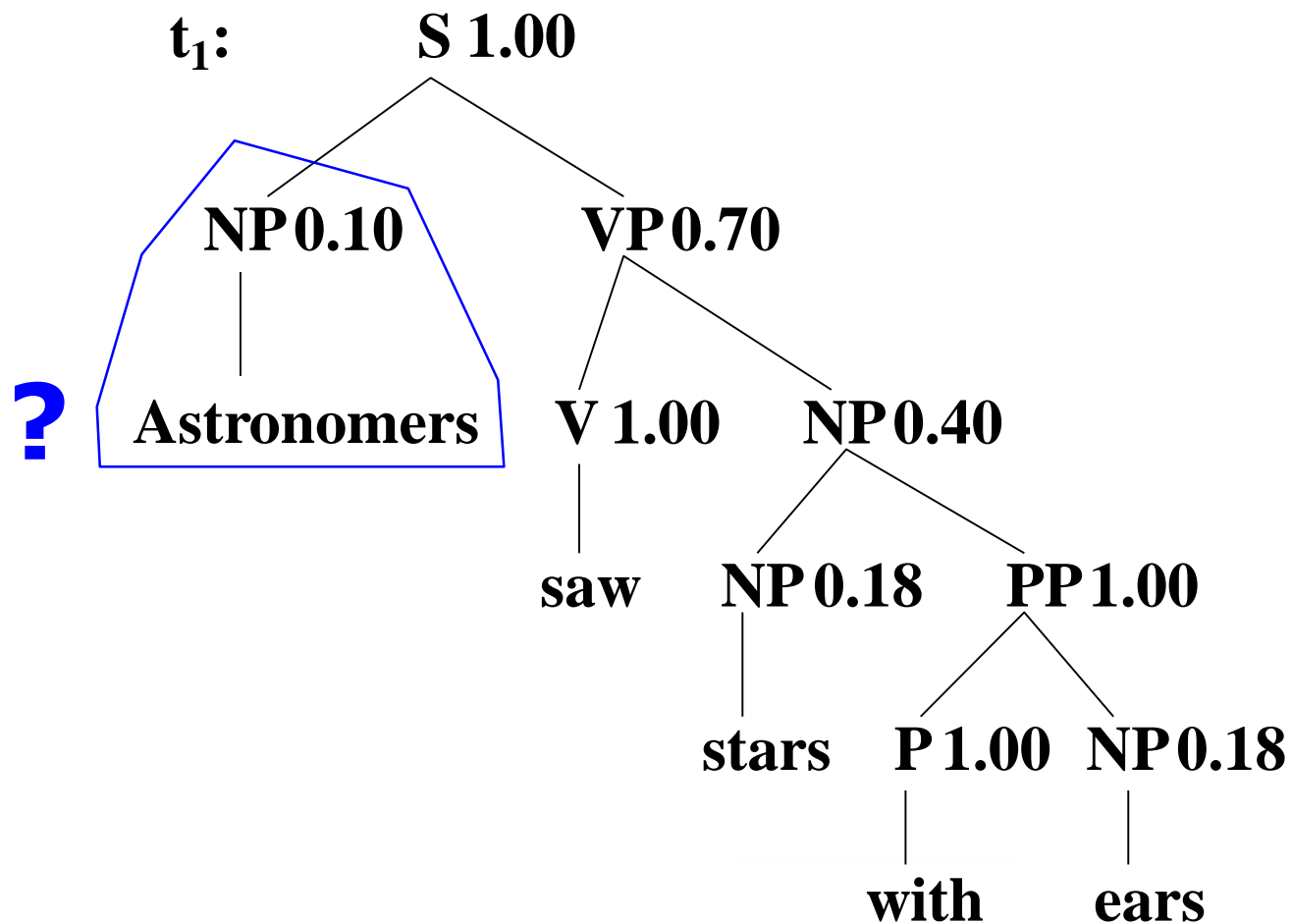


概率上下文无关文法

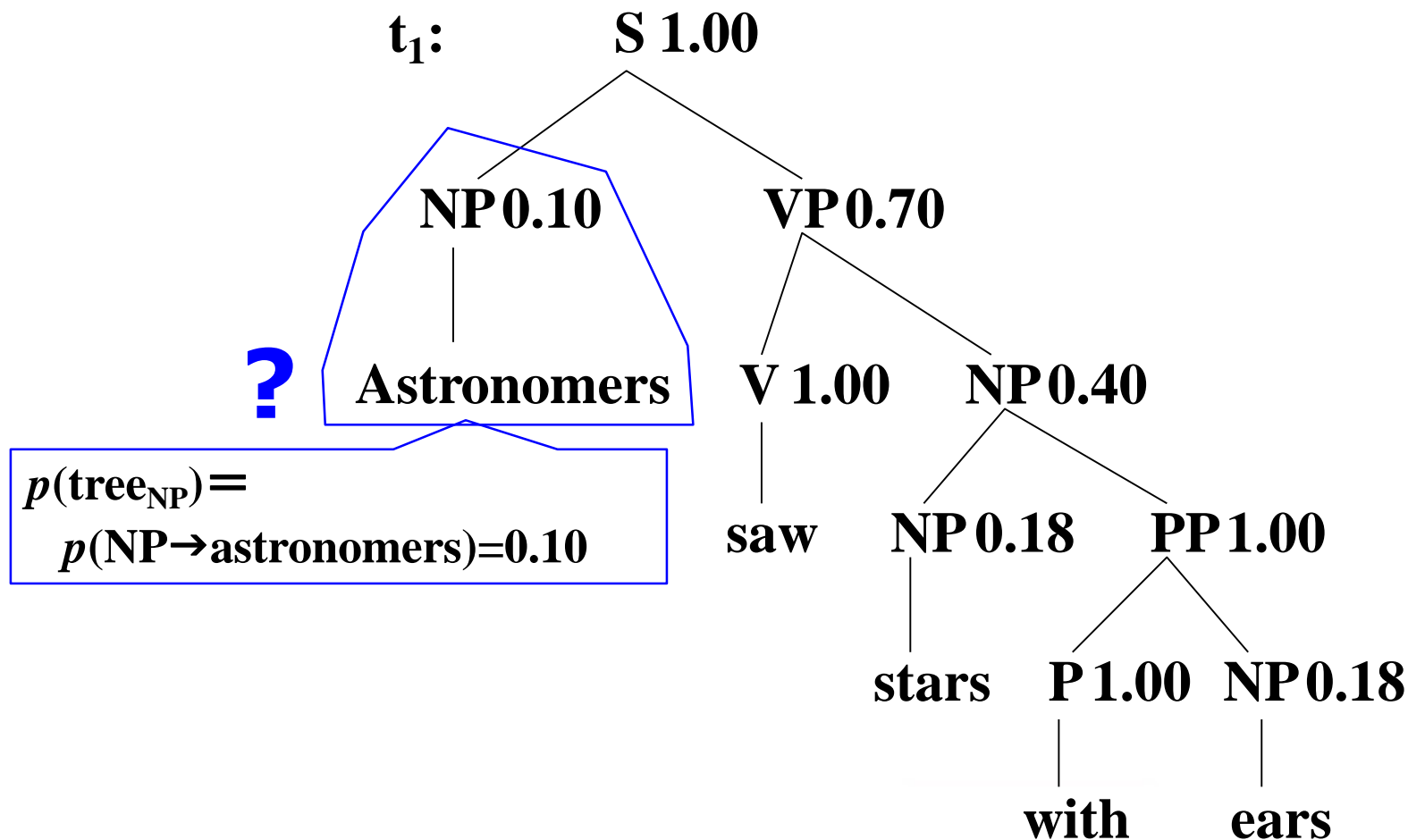
给定一个语法分析树，我们可以计算它的概率：

$$P(T) = \prod_{i=1}^n P(\text{RHS}_i | \text{LHS}_i)$$

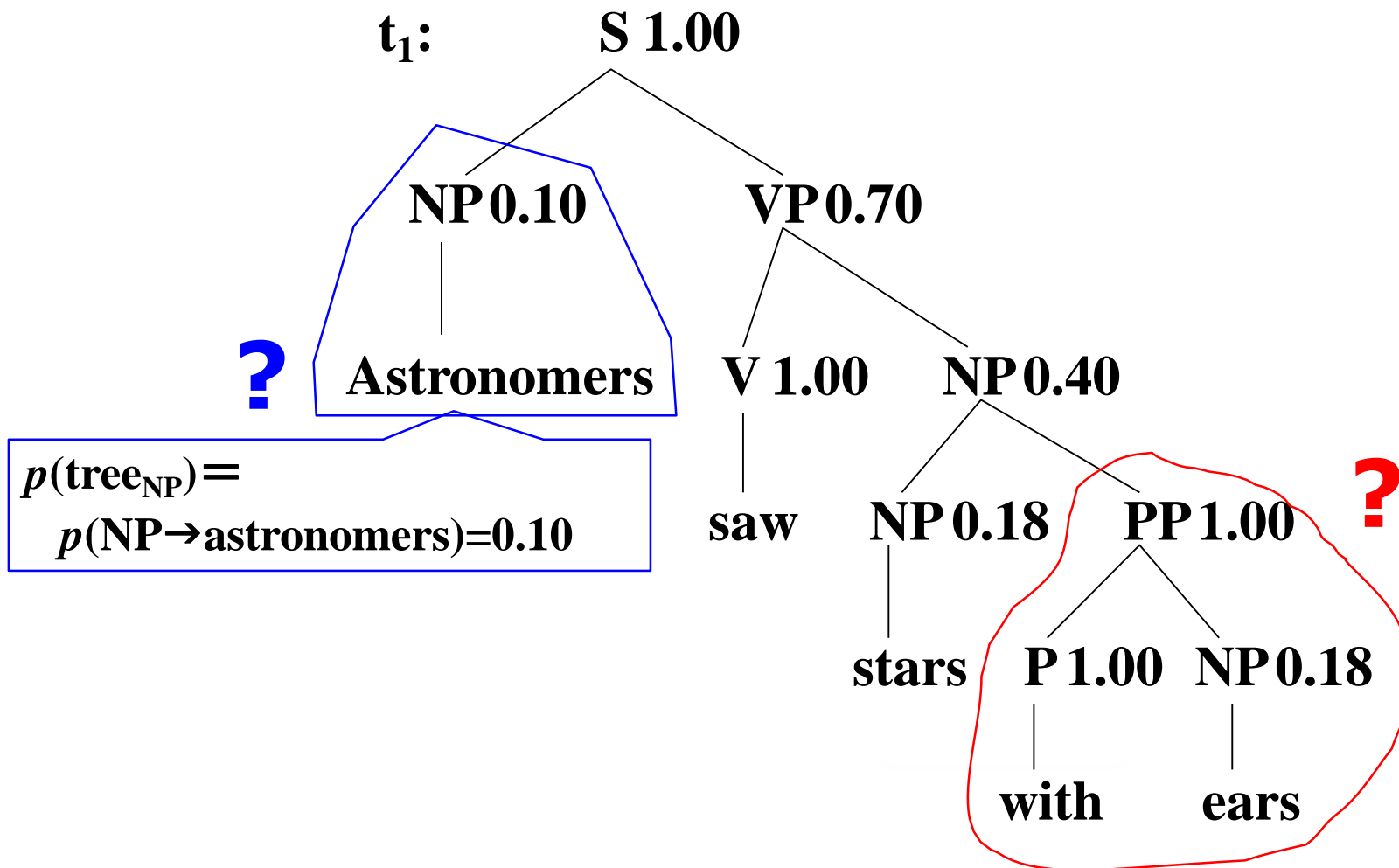
概率上下文无关文法



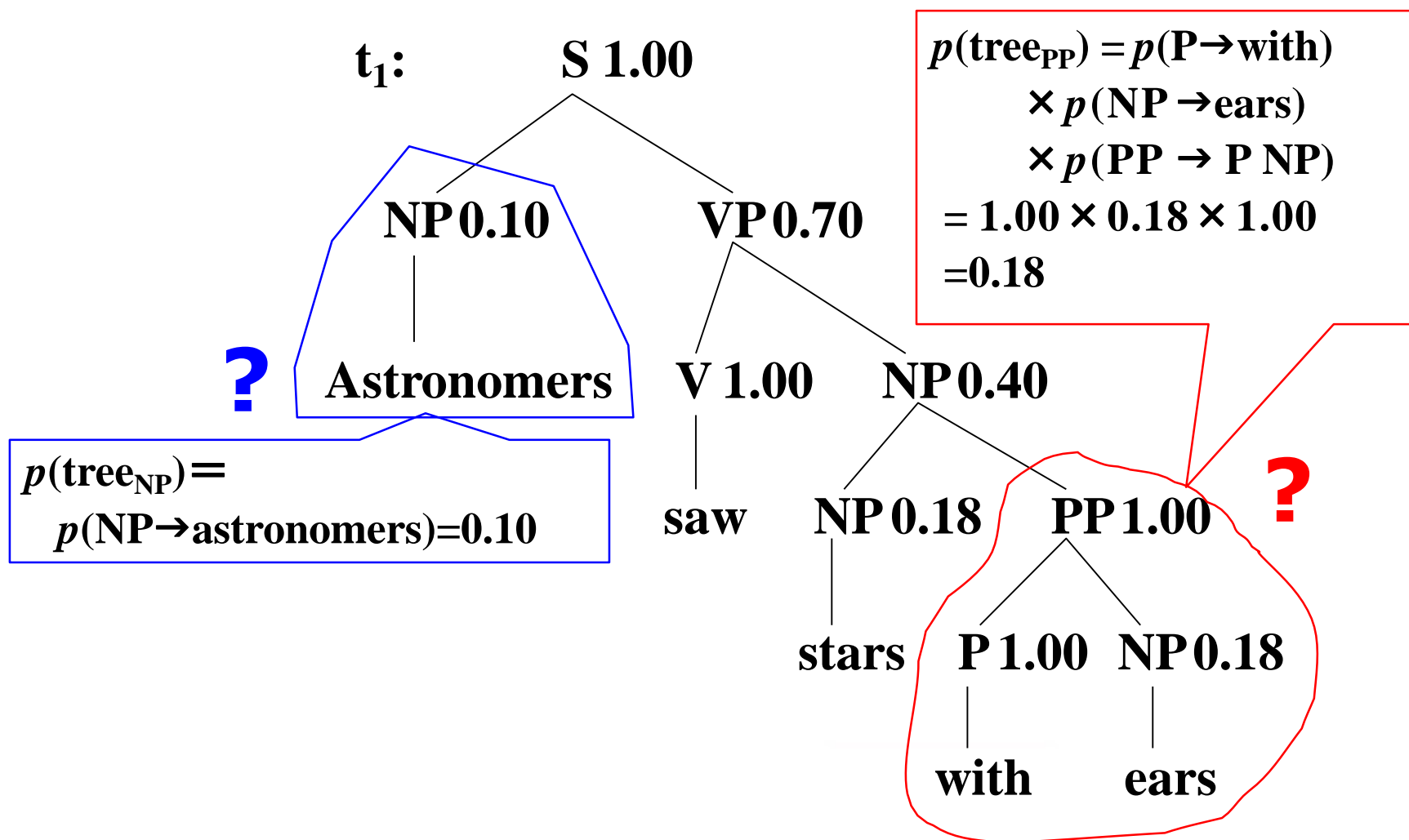
概率上下文无关文法



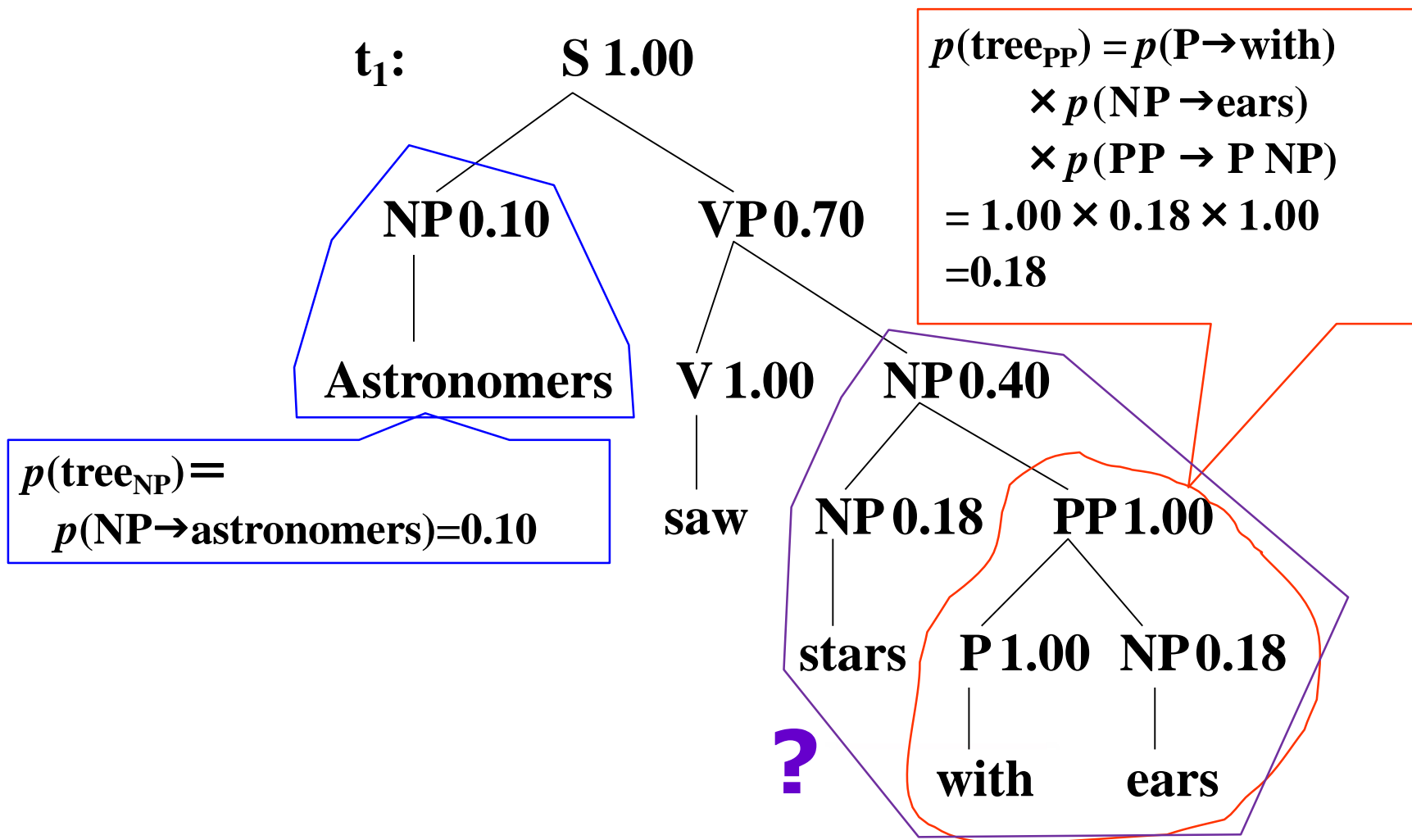
概率上下文无关文法



概率上下文无关文法

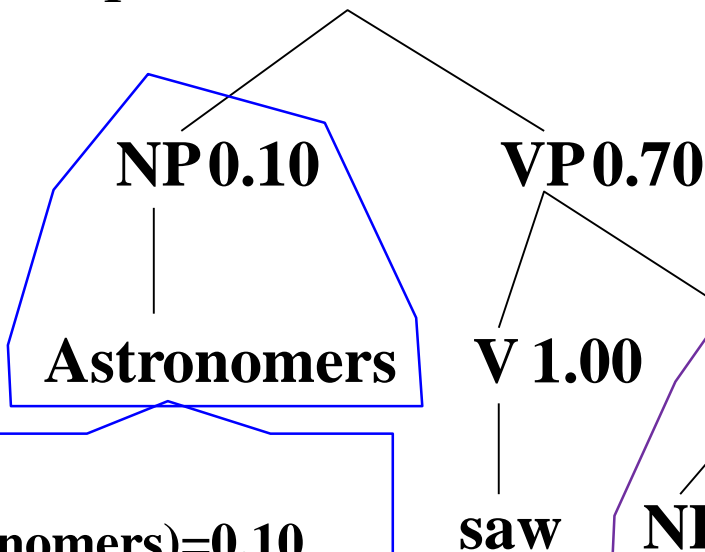


概率上下文无关文法



概率上下文无关文法

t_1 : S 1.00



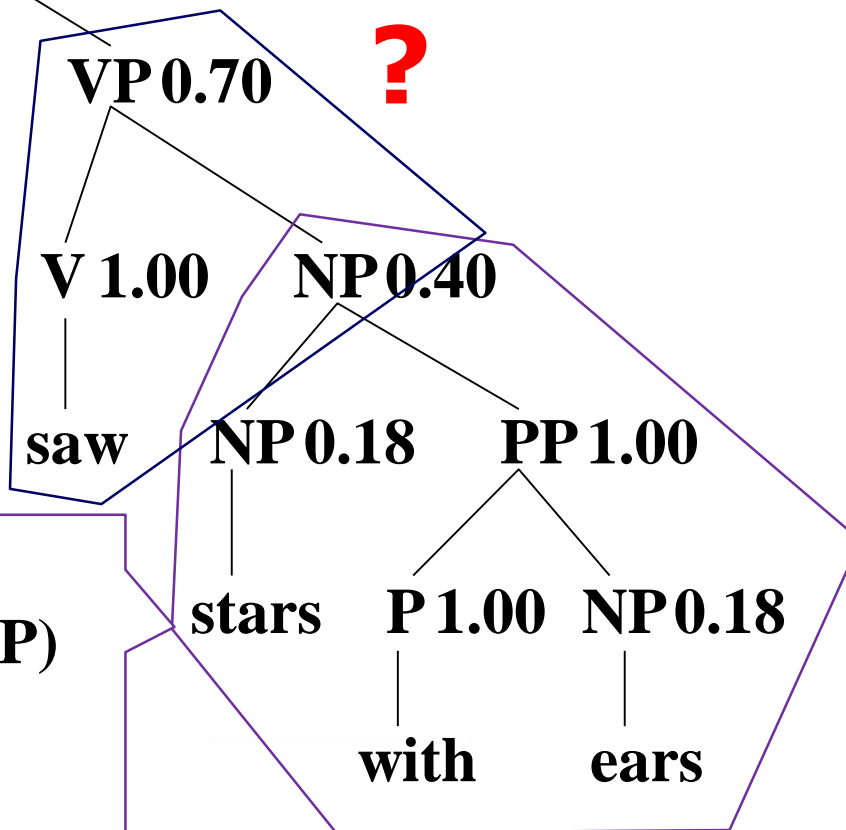
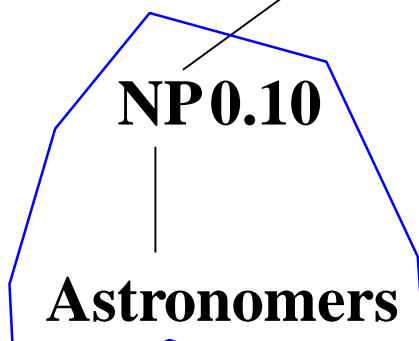
$$\begin{aligned}
 p(\text{tree}_{PP}) &= p(P \rightarrow \text{with}) \\
 &\quad \times p(NP \rightarrow \text{ears}) \\
 &\quad \times p(PP \rightarrow P NP) \\
 &= 1.00 \times 0.18 \times 1.00 \\
 &= 0.18
 \end{aligned}$$

$$\begin{aligned}
 p(\text{tree}_{NP}) &= \\
 & p(NP \rightarrow \text{astronomers}) = 0.10
 \end{aligned}$$

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

概率上下文无关文法

t_1 : S 1.00

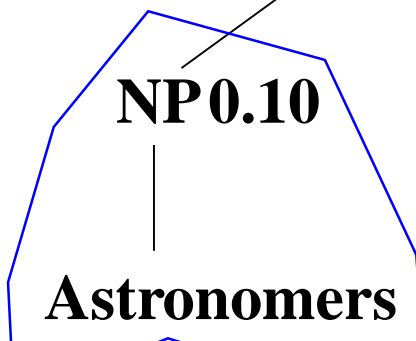


$$p(\text{tree}_{\text{NP}}) = p(\text{NP} \rightarrow \text{astronomers}) = 0.10$$

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

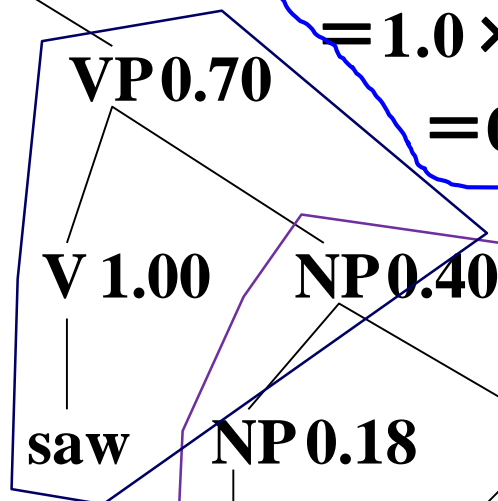
概率上下文无关文法

t_1 : S 1.00



$$p(\text{tree}_{\text{NP}}) = p(\text{NP} \rightarrow \text{astronomers}) = 0.10$$

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



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

概率上下文无关文法

t_1 :

S 1.00

NP 0.10

VP 0.70

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

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

$p(\text{tree}_{\text{NP}})$
 $p(\text{NP} \rightarrow$

1.00

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

stars

P 1.00

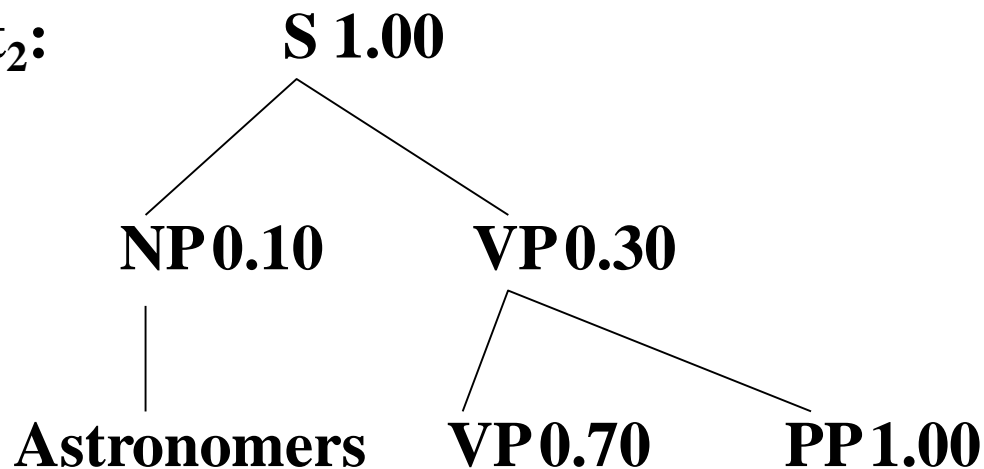
NP 0.18

with

ears

概率上下文无关文法

t_2 :



$$\begin{aligned} p(t_2) &= 1.00 \times 0.10 \times 0.30 \times 0.70 \\ &\times 1.00 \times 0.18 \times 1.00 \times 1.00 \times 0.18 \\ &= 0.0006804 \end{aligned}$$

18

saw stars with ears

概率上下文无关文法

对于给定的句子 S ，两棵句法分析树的概率不等， $P(t_1) > P(t_2)$ ，因此，可以得出结论：分析结果 t_1 正确的可能性大于 t_2 。

概率上下文无关文法

如何计算每条规则的概率？

句法分析工具

- ✧ **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>
- ✧ **Charniak Parser**: <http://www.cs.brown.edu/people/ec/#software>
- ✧ **Bikel Parser**
<http://www.cis.upenn.edu/~dbikel/software.html#stat-parser>
- ☼ **Oboe Parser** (可执行程序)
<http://www.openpr.org.cn/index.php/NLP-Toolkit-for-Natural-Language-Processing/>

Thank you!

权小军 中山大学数据科学与计算机学院