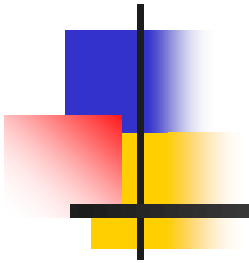
第9章 句法分析

(2/2)

北京市海淀区中关村东路95号
邮编：100190
电话： +86-10-8254 4688
邮件：cqzong@nlpr.ia.ac.cn



9.10 依存句法分析

9.10 依存句法分析

◆依存句法理论

现代依存语法(dependency grammar)理论的创立者是法国语言学家吕西安·泰尼埃 (其姓氏也被译作：特思尼耶尔、特尼耶尔等)(Lucien Tesni ère, 1893-1954)。他的主要思想反映在1953年出版的专著《结构句法概要》(Esquisse dune syntaxe structu-rale)中。

9.10 依存句法分析

L. Tesnière 的理论认为：

一切结构句法现象可以概括为关联(connexion)、组合(jonction)和转位(tanslation)这三大核心。句法关联建立起词与词之间的从属关系，这种从属关系是由支配词和从属词联结而成；动词是句子的中心，并支配其他成分，它本身不受其他任何成分的支配。

欧洲传统的语言学突出一个句子中主语的地位，句中其它成分称为“谓语”。依存语法打破了这种主谓关系，认为“谓语”中的动词是一个句子的中心，其他成分与动词直接或间接地产生联系。

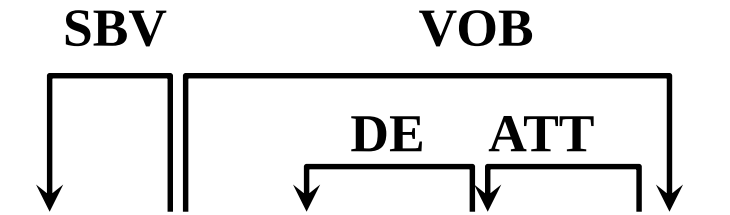
9.10 依存句法分析

Tesnière 还在《结构句法基础》中将化学中“价”的概念引入依存语法中。“价” 亦称“配价”或“向”(法文valence，德文valenz，英文valence/ valency)，一个动词所能支配的行动元(名词词组)的个数即为该动词的价数。也就是说，它能支配几个行动元，它就是几价动词。如汉语中的零价动词：“地震、刮风”；一价动词：“病、醉、休息、咳嗽、游泳”等；二价动词：“爱、采、参观、讨论”等；三价动词：“给、送、告诉、赔偿”等。1959年问世的《结构句法基础》(Elements de syntaxe structurale) 则标志着配价语法论的形成。

9.10 依存句法分析

在依存语法理论中，“依存”就是指词与词之间支配与被支配的关系，这种关系不是对等的，而是有方向的。**处于支配地位的成分称为支配者**(governor, regent, head)，**而处于被支配地位的成分称为从属者**(modifier, subordinate, dependency)。

9.10 依存句法分析

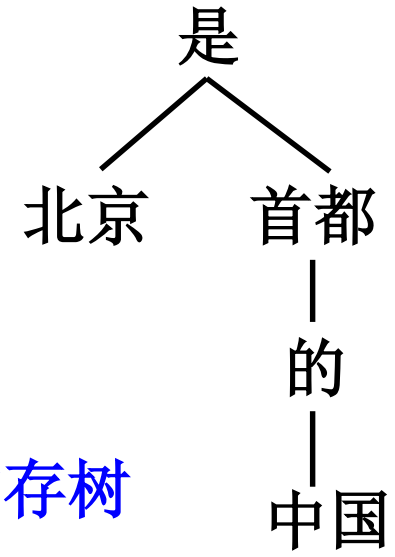


(e) 有向图-1

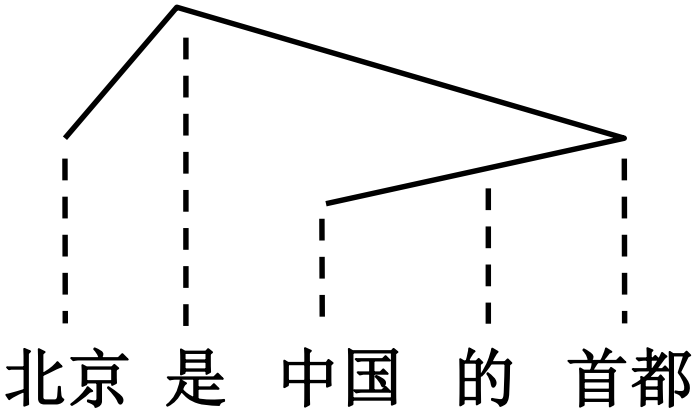
(e) 有向图-2

两个有向图用带有方向的弧(或称边，edge)来表示两个成分之间的依存关系，支配者在有向弧的发出端，被支配者在箭头端，我们通常说被支配者依存于支配者。

9.10 依存句法分析



(f) 依存树



(g) 依存投射树

图(f)是用树表示的依存结构，树中子节点依存于该节点的父节点。

图(g)是带有投射线的树结构，实线表示依存联结关系，位置低的成份依存于位置高的成份，虚线为投射线。

9.10 依存句法分析

1970年计算语言学家J. Robinson在论文《依存结构和转换规则》中提出了依存语法的4条公理：

(1) 一个句子只有一个独立的成分；

(2) 句子的其他成分都从属于某一成分；

(3) 任何一成分都不能依存于两个或多个成分；

(4) 如果成分A直接从属于成分B，而成分C在句子中位于A和B之间，那么，成分C或者从属于A，或者从属于B，或者从属于A和B之间的某一成分。

9.10 依存句法分析

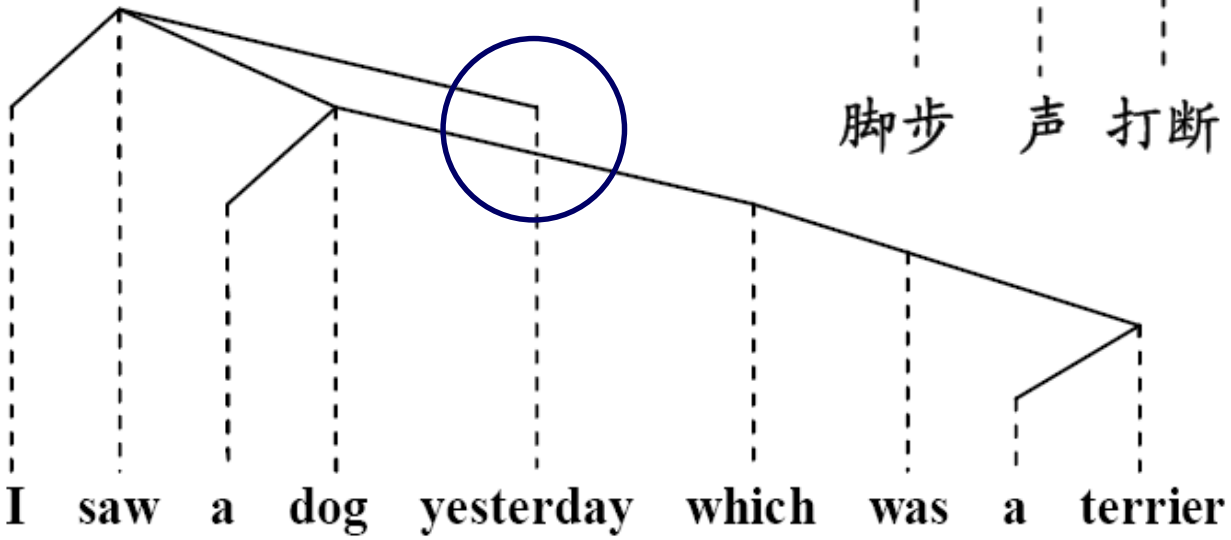
这4条公理相当于对依存图和依存树的形式约束为：

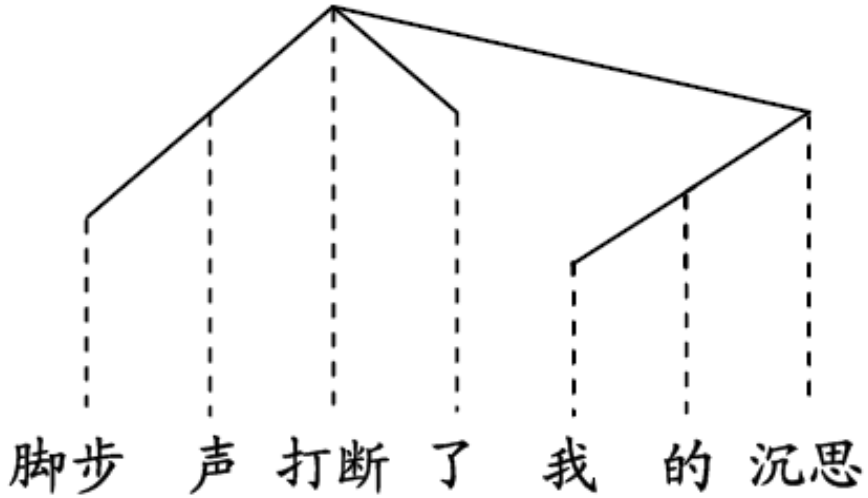
- 单一父结点(single headed)
- 连通(connective)
- 无环(acyclic)
- 可投射(projective)

由此来保证句子的依存分析结果是一棵有“根(root)”的树结构。

9.10 依存句法分析

投射(projective)与
非投射(no-projective)





9.10 依存句法分析

◆依存语法的优势

- 简单，直接按照词语之间的依存关系工作，是天然词汇化的；
- 不过多强调句子中的固定词序，对自由语序的语言分析更有优势；
- 受深层语义结构的驱动，词汇的依存本质是语义的；
- 形式化程度较短语结构语法浅，对句法结构的表述更为灵活。

9.10 依存句法分析

◆依存句法分析方法

依存句法分析(dependency parsing)的任务就是分析出句子中所有词汇之间的依存关系。

建立一个依存句法分析器一般需要完成以下三部分工作：

(1) 依存句法结构描述

(2) 分析算法设计与实现

(3) 文法规则或参数学习

9.10 依存句法分析

目前依存句法结构描述一般采用有向图方法或依存树方法，所采用的句法分析算法可大致归为以下4类：

- **生成式的分析方法**(generative parsing)
- **判别式的分析方法**(discriminative parsing)
- **决策式的（确定性的）分析方法**(deterministic parsing)
- **基于约束满足的分析方法**(constraint satisfaction parsing)

9.10 依存句法分析

- **生成式的分析方法** (generative parsing)
  - **基本思想**：采用联合概率模型$Score(x, y|\theta)$（其中，x为输入句子，y为依存分析结构，θ为模型的参数）生成一系列依存句法树，并赋予其概率分值，然后采用相关算法找到概率打分最高的分析结果作为最后输出。这是一种完全句法分析方法，它搜索整个概率空间，得到整个句子的依存分析结果。

9.10 依存句法分析

✧ 方法评价

- **优点：**
  - ✓ 此类方法的准确率较高
- **弱点：**
  - ✓ 采用联合概率模型，在进行概率乘积分解时做了不尽合理的假设，不易加入语言特征；
  - ✓ 因为采用全局搜索，算法的复杂度较高，一般为$O(n^3)$或$O(n^5)$；
  - ✓ 不易处理非投射现象。

9.10 依存句法分析

- **判别式的分析方法** (discriminative parsing)
  - **基本思想**：采用条件概率模型 $Score(x \mid y, \theta)$，使目标函数 $\prod_{i=1}^{n} Score(x_i \mid y_i; \theta)$ 最大的 θ 作为模型的参数。
  - **例如**：最大生成树模型(maximum spanning trees, MST)

    定义整棵句法树的打分是树中各条边打分的加权和：

$$s(\boldsymbol{x}, \boldsymbol{y}) = \sum_{(i,j)\in y} s(i, j) = \sum_{(i,j)\in y} \mathbf{w} \cdot \mathbf{f}(i, j)$$

9.10 依存句法分析

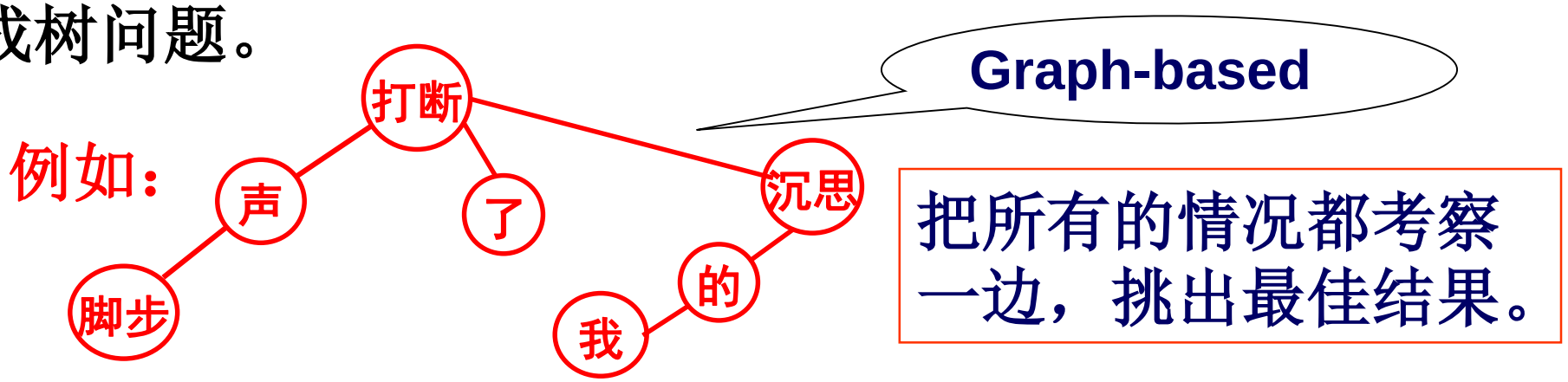
其中，s 表示打分值，y 是句子x 的一棵依存树，(i, j) 是y 中的结点对。$\mathbf{f}(\bullet)$是取值为 1 或 0 的高维二元特征函数向量，表示结点x_i 和 x_j 之间的依存关系，如果一棵依存分析树中两个词存在依存关系，例如：“打”和“球”，则 $f(i, j)=1$，否则，$f(i, j)=0$。即：

$$f(i,j)=\begin{cases}1 & \text{如果} x_i=\text{'打'} \; and \; x_j=\text{'球'} \\ 0 & \text{其他}\end{cases}$$

$\mathbf{w}$ 是特征 $\mathbf{f}(i, j)$ 的权值向量，$\mathbf{w}$在确定了特征后由样本训练得到。

9.10 依存句法分析

该方法**基本思想**是：在点和边组成的生成树(spanning tree)中找到加权和分值最高的边的组合。生成树中任意两个由词表示的节点之间都有边，根据特征和权值为每条边打分，求解最佳分析结果转化为搜索打分最高的最大生成树问题。



R. McDonald, K. Lerman and F. Pereira. 2006. Multilingual Dependency Analysis with a Two-Stage Discriminative Parser. *Proc. CoNLL-X*, pp. 216–220

9.10 依存句法分析

✧ 方法评价

- **优点：**
  - ✓ 采用判别式模型，避开了联合概率模型所要求的独立性假设；
  - ✓ 较好的可计算性，使诸多机器学习和运筹学的方法得以应用，并可处理非投射现象；
  - ✓ 分析准确率较高。
- **弱点：**
  - ✓ 整句内的全局搜索，不易使用动态特征；
  - ✓ 同样由于是全局搜索，算法复杂度较高。

9.10 依存句法分析

- **决策式的(确定性的)分析方法**(deterministic parsing)
  - **基本思想**：模仿人的认知过程，按照特定方向每次读入一个词。每读入一个词，都要根据当前状态做出决策(比如判断是否与前一个词发生依存关系)。一旦决策做出，将不再改变。所做决策即“采取什么样的分析动作(action)”。分析过程可以看作是一步一步地作用于输入句子之上的分析动作(action)的序列。

9.10 依存句法分析

(1) 移进—归约算法

J. Nivre等(2003)提出的自左向右、自底向上的分析算法：

当前分析状态的格局(configuration)是一个三元组：**(*S*, *I*, *A*)**，***S*, *I*, *A*分别表示栈、未处理结点序列(Input)和依存弧集合(Arcs)**。分析体系主要包含两种分析动作组合，一种是采用标准移进—规约方式，使用Left-Reduce、Right-Reduce 和 Shift 三种动作。

J. Nivre, and J. Nilsson. 2003. Three Algorithms for Deterministic Dependency Parsing. *Proc. of the 15th NODALIDA*, pp. 47-56

9.10 依存句法分析

初始：$[\ ]_W$　　终止：$[\text{ROOT}]_W$

Right

$$\frac{\ldots\ [w_i, w_{i+1}]_W\ \ldots}{\ldots\ [w_{i+1}]_W\ \ldots \quad A \cup \{w_i \leftarrow w_{i+1}\}}$$

w_i

Left

$$\frac{\ldots\ [w_i, w_{i+1}]_W\ \ldots}{\ldots\ [w_i]_W\ \ldots \quad A \cup \{w_i \rightarrow w_{i+1}\}}$$

w_{i+1}

Shift

$$\frac{\ldots\ [w_i, w_{i+1}]_W\ \ldots}{\ldots\ [w_{i+1}, w_{i+2}]_W\ \ldots}$$

9.10 依存句法分析

(2)Arc-eager 分析算法— 4种分析动作(Actions):

初始：(nil, I, Ø)　　终止：(S, nil, A)

Left-Arc$_l$

$$\frac{[\ldots,w_i]_S\ [w_j,\ldots]_{Input}\quad \neg\exists w_k \rightarrow w_i \in A}{[\ldots]_S\ [w_j,\ldots]_{Input}\quad A \cup \{w_i \overset{l}{\leftarrow} w_j\},\ \mathbf{pop}(w_i)}$$

条件

Right-Arc$_l$

$$\frac{[\ldots,w_i]_S\ [w_j,\ldots]_{Input}\quad \neg\exists w_k \rightarrow w_j \in A}{[\ldots w_i,w_j]_S\ [\ldots]_{Input}\quad A \cup \{w_i \overset{l}{\rightarrow} w_j\},\ \mathbf{push}(w_j)}$$

Reduce

$$\frac{[\ldots w_i]_S\ [\ldots]_{Input}\quad \exists w_k \rightarrow w_i \in A}{[\ldots]_S\ [\ldots]_{Input}\quad \mathbf{pop}(w_i)}$$

Shift

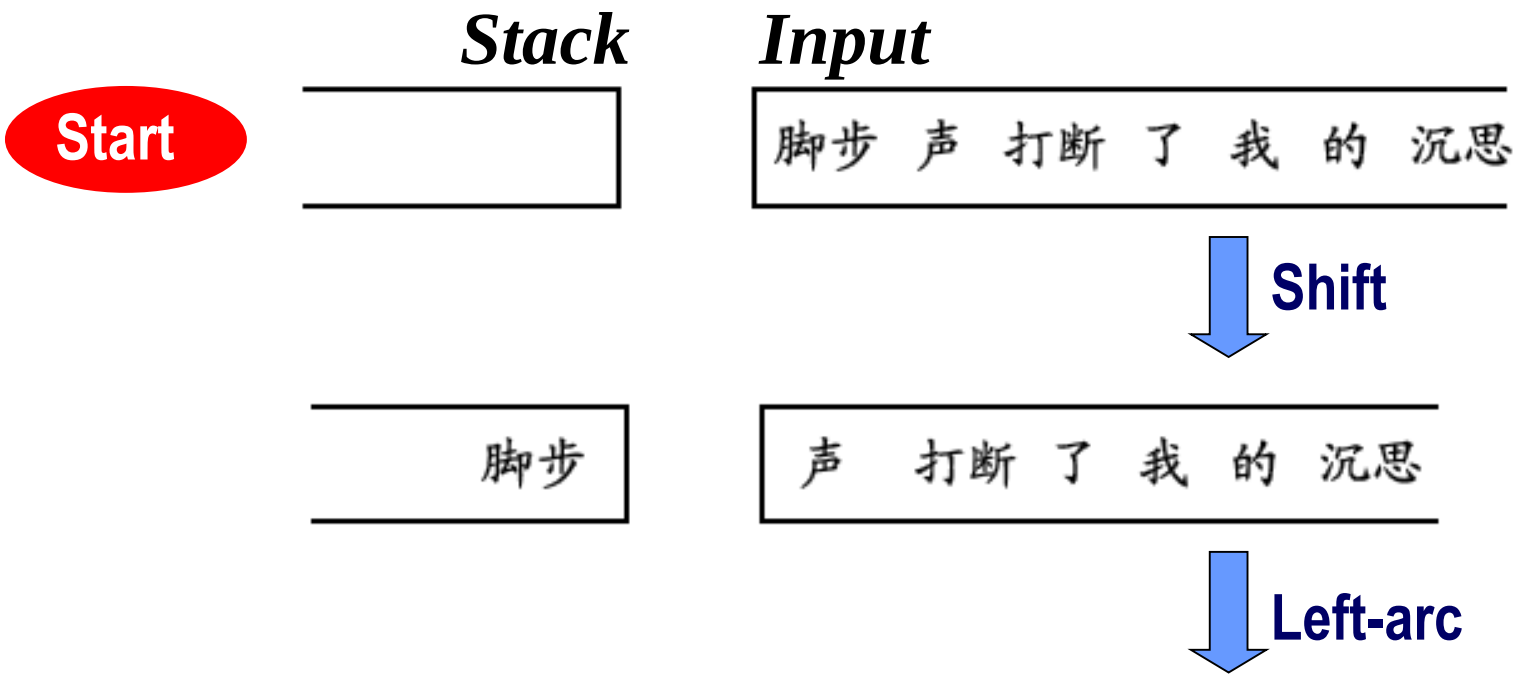
$$\frac{[\ldots]_S\ [w_i,\ldots]_{Input}}{[\ldots w_i]_S\ [\ldots]_{Input}\quad \mathbf{push}(w_i)}$$

Joakim Nivre and Mario Scholz ,Deterministic Dependency Parsing of English Text. *Proc*. COLING'2004.

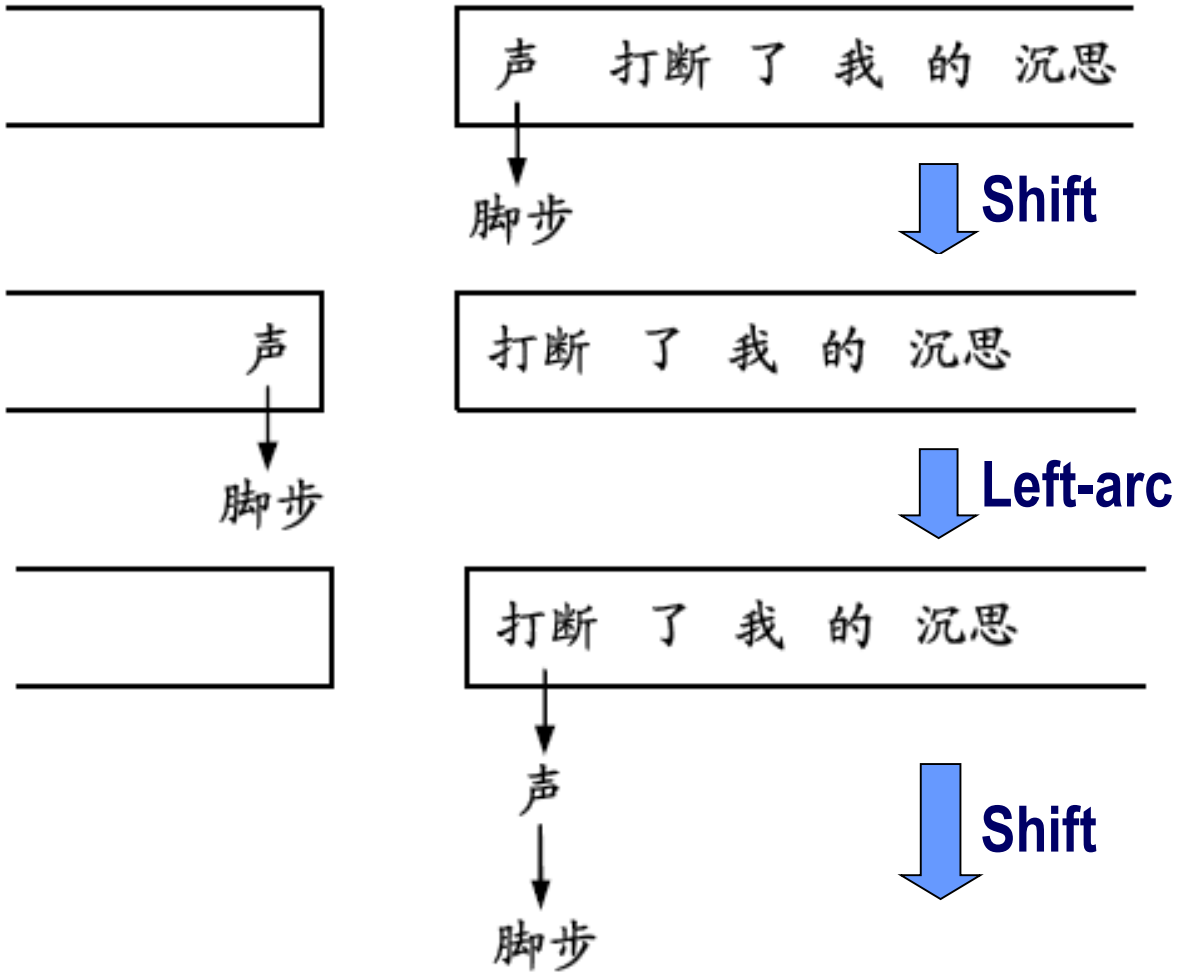
9.10 依存句法分析

✧举例：分析如下句子：

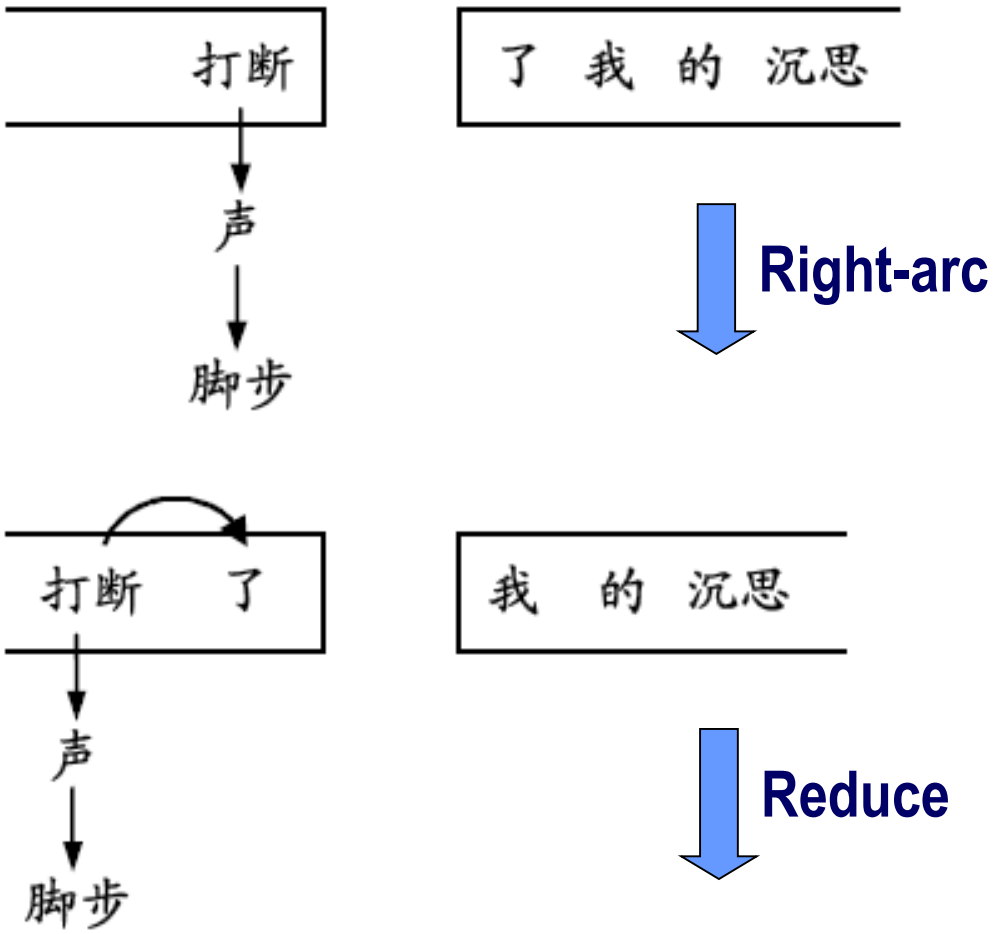
脚步　声　打断　了　我　的　沉思



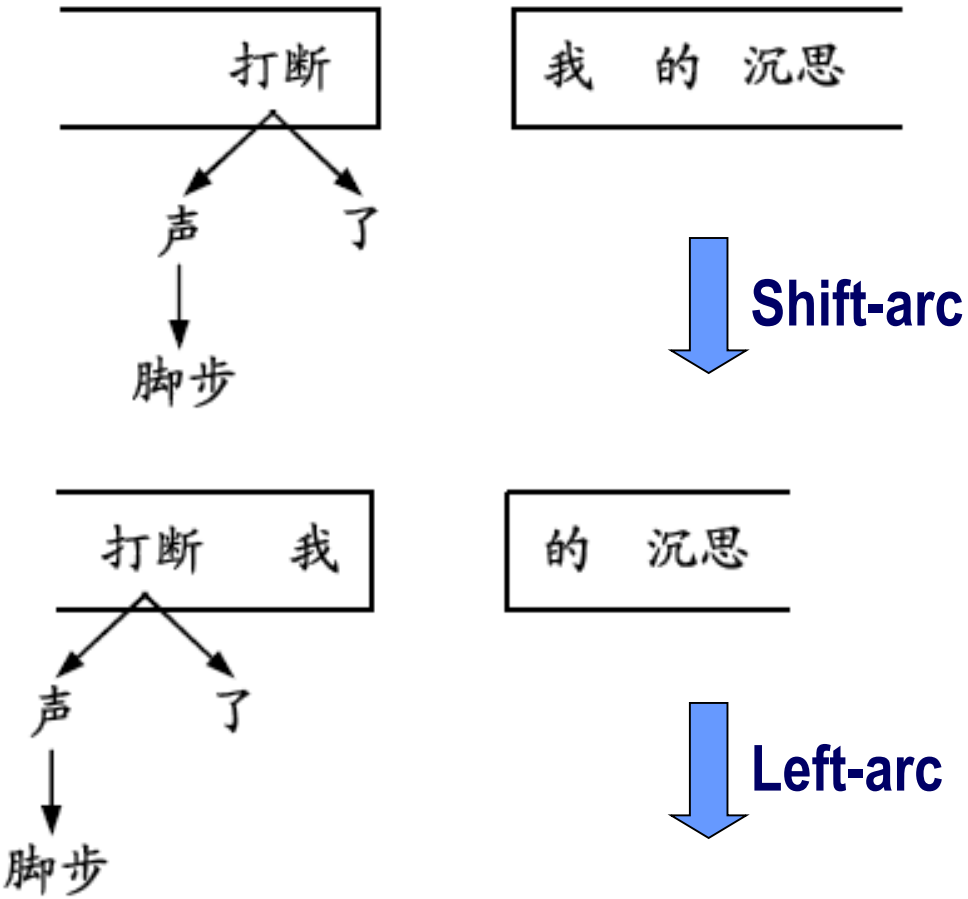
9.10 依存句法分析



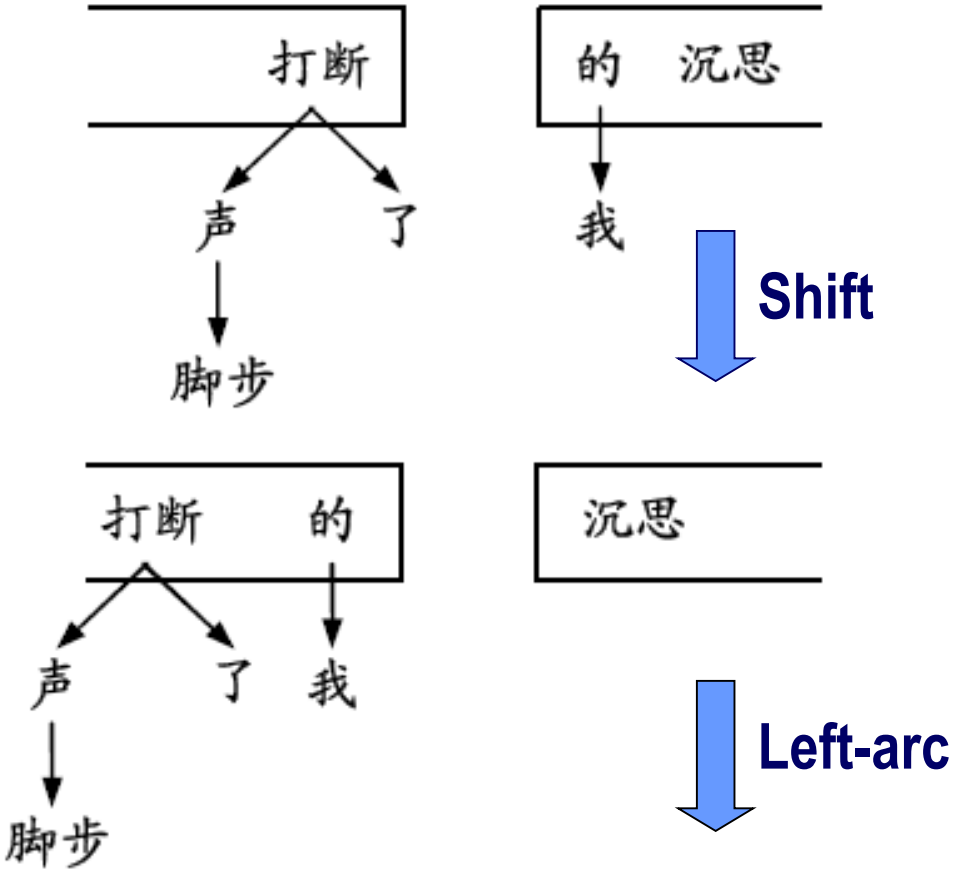
9.10 依存句法分析



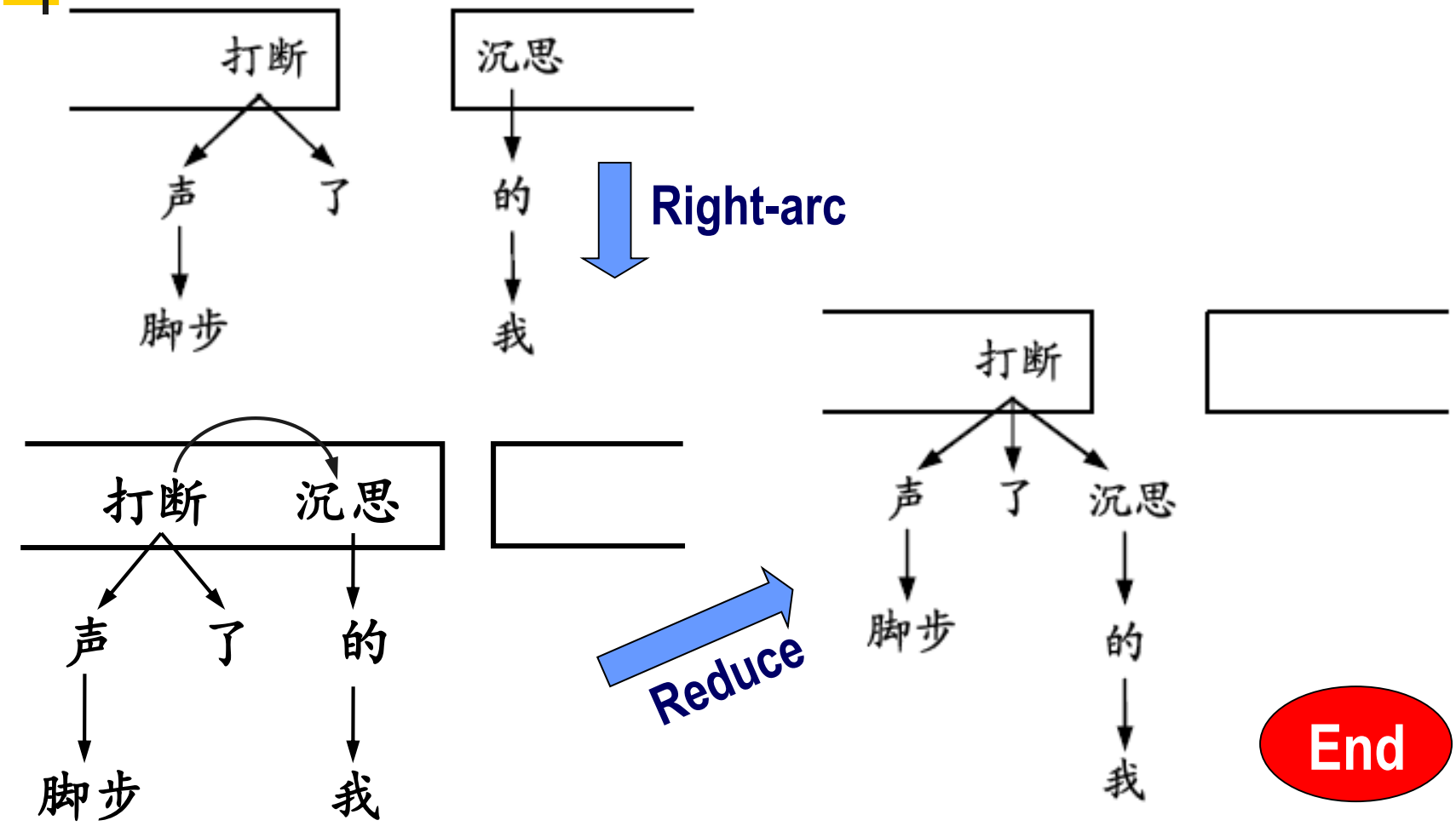
9.10 依存句法分析



9.10 依存句法分析



9.10 依存句法分析



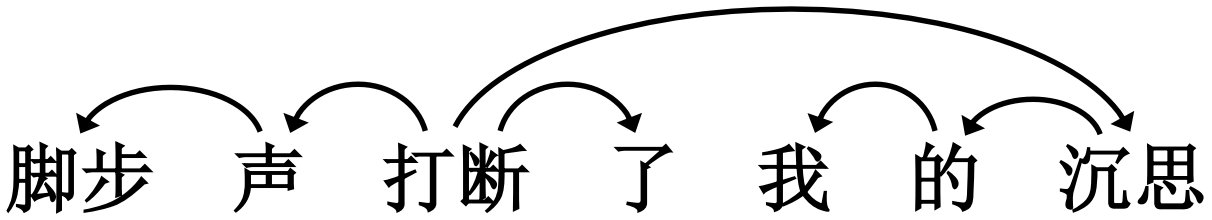
9.10 依存句法分析

脚步 声 打断 了 我 的 沉思

shift
left
shift
left
shift
right
reduce
shift
left
left
right
reduce

Transition-based

是一种阶段性的方法，认为每一步都是对的！



9.10 依存句法分析

✧ 方法评价

- **优点：**
  - ✓ 算法可以使用之前产生的所有句法结构作为特征；
  - ✓ 可以达到线性复杂度：$O(n)$。
- **弱点：**
  - ✓ 以局部最优的加和代替全局最优，导致错误传递；
  - ✓ 不可处理非投射现象，准确率稍逊于全局最优算法。

9.10 依存句法分析

➢ 实现一个依存句法分析器

以由 **Arc-eager** 算法实现基于转换的(transition-based)句法分析器为例。

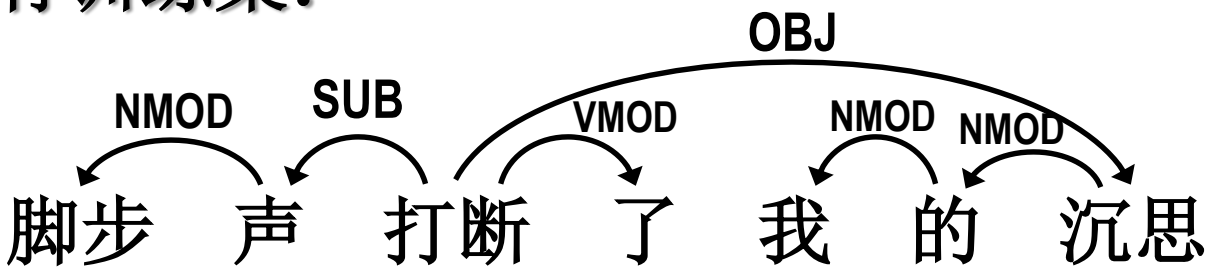
✧ **基本思路：** 在每一个状态(configuration)下根据当前状态提取特征，然后通过分类决定下一步应该采取的“动作”(action)：移进(shift)、左弧(left-arc)、右弧(right-arc)、归约(reduce)，执行分类器选择的最优动作，转换到下一个状态。

✧ **具体实现：** ①标注大量的依存关系句法树，建立训练集。每个句子都可以一对一地转换为动作序列；②确定特征集合，以构造动作分类器。

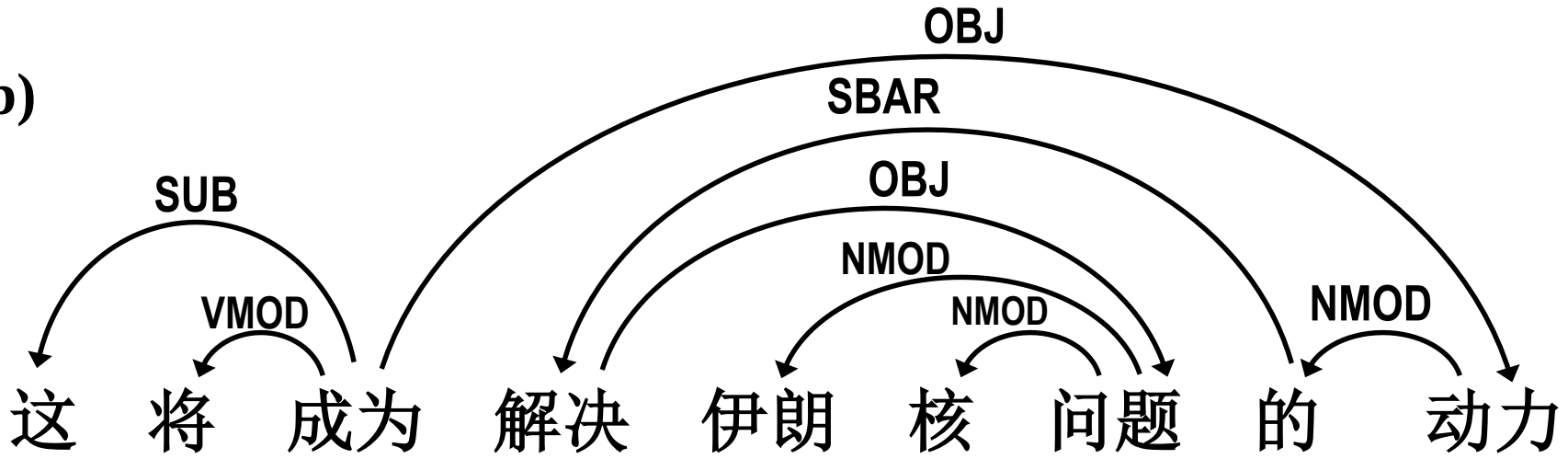
9.10 依存句法分析

例如，有训练集：

(a)



(b)

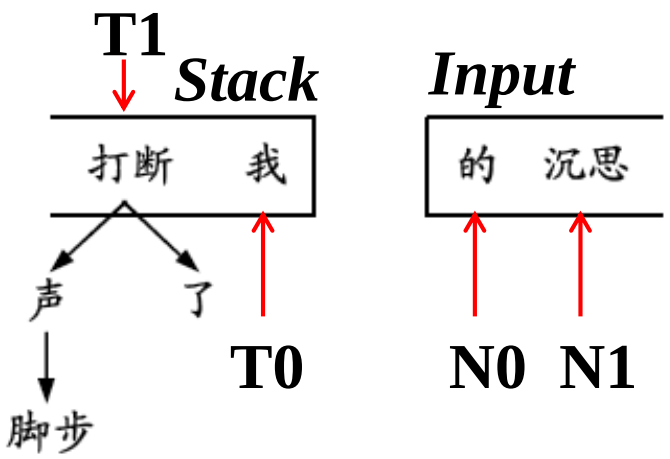


9.10 依存句法分析

假设选取如下特征构造分类器：word/w: 词；pos/p: 词性；lc：left_most_child；rc: righ_most_child；rel: 依存关系

	word	pos	lc_w	lc_p	lc_rel	rc_w	rc_p	rc_rel
T1	√							
T0	√	√		√	√	√		√
N0	√	√		√	√		√	√
N1	√	√						

其中，N0、N1分别表示Input序列中(即“I”)最前面的token(N0)和第2个token(N1)，依次向后排；T0、T1分别表示Stack栈中(即“S”)最顶的token(T0)、次顶的token(T1)，依次向下排。表中的“√”是指在当前状态下，分类器的特征从哪里取、取什么。如T0和pos对应的框里画√，表示取栈顶词的词性作为特征之一。依此类推。



9.10 依存句法分析

根据前面的图(b)得到如下训练实例：

动作	特征序列
Shift	T0w:这 T0p:PN N0w:将 N0p:AD N1w:成为 N1p:VV
L_A	T1p:PN T0w:将 T0p:AD N0w:成为 N0p:VV N1w:解决 N1p:VV
L_A	T0w:这 T0p:PN N0w:成为 N0p:VV N1w:解决 N1p:VV N0lc_p:AD N0lc_rel:VMOD N0rc_w:将 N0rc_rel:VMOD
Shift	T0w:成为 T0p:VV T0lc_p:PN T0lc_rel:SUB N0w:解决 N0p:VV N1w:伊朗 N1p:NR
Shift	T1w:成为 T1p:VV T1lc_p:PN T0lc_rel:SUB T0w:解决 T0p:VV N0w:伊朗 N0p:NR N1w:核 N1p:NN

……

9.10 依存句法分析

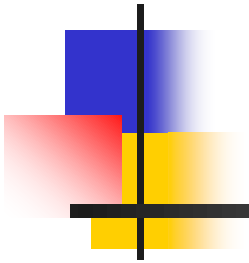
根据前面的图(b)得到如下训练实例

动作	特征序列
Shift	T0w:这 T0p:PN N0w:将 N0p:AD N1w:成为 N1p:VV
L_A	T1p:PN T0w:将 T0p:AD N0w:成为 N0p:VV N1w:解决 N1p:VV
L_A	T0w:这 T0p:PN N0w:成为 N0p:VV N1w:解决 N1p:VV N0lc_p:AD N0lc_rel:VMOD N0rc_w:将 N0rc_rel
Shift	T0w:成为 T0p:VV T0lc_p:PN T0lc_rel:SBV N0w:伊朗 N0p:NR N1w:伊朗 N1p:NR

“这”为栈顶T0，“将”为输入序列第一个token N0，动作类别标签是Shift。

“将”为栈顶，“成为”为输入序列第一个token，执行动作“left-arc”(“将”是“成为”的孩子)。

可以看出，一旦子节点在父节点的左边，执行left_arc后可以马上归约，但如果子节点在父节点的右边，则不能马上把右边的孩子归约掉，因为也许这个孩子还有孩子在更右边的位置，还没有遍历到。(主要是由从左到右的遍历顺序导致的)。



9.11 依存句法分析器性能评价

9.11 依存句法分析器性能评价

- **无标记依存正确率**(unlabeled attachment score, UA)：所有词中找到其正确支配词的词所占的百分比，没有找到支配词的词(即根结点)也算在内。
- **带标记依存正确率**(labeled attachment score, LA)：所有词中找到其正确支配词并且依存关系类型也标注正确的词所占的百分比，根结点也算在内。
- **依存正确率**(dependency accuracy, DA)：所有非根结点词中找到其正确支配词的词所占的百分比。

9.11 依存句法分析器性能评价

- **根正确率**(root accuracy, RA)：有两种定义方式：

  (1)正确根结点的个数与句子个数的比值；

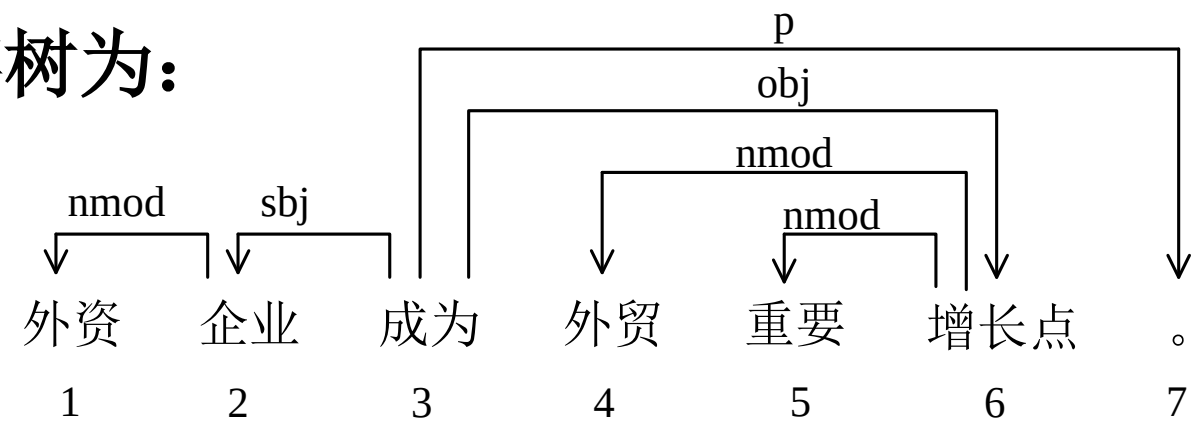
  (2)另一种是所有句子中找到正确根结点的句子所占的百分比。

  对单根结点语言或句子来说，二者是等价的。

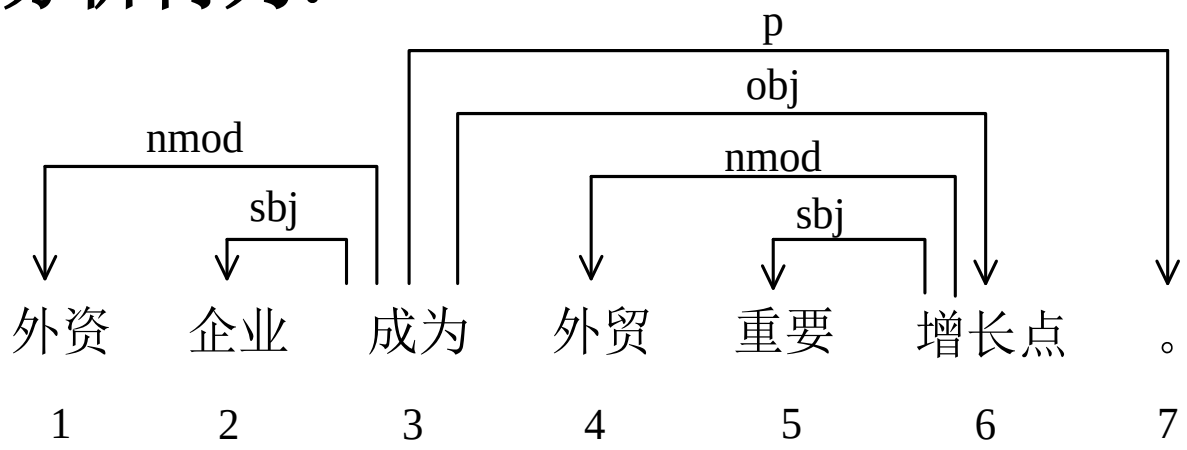
- **完全匹配率**(complete match, CM)：所有句子中无标记依存结构完全正确的句子所占的百分比。

9.11 依存句法分析器性能评价

✧例如，答案依存树为：

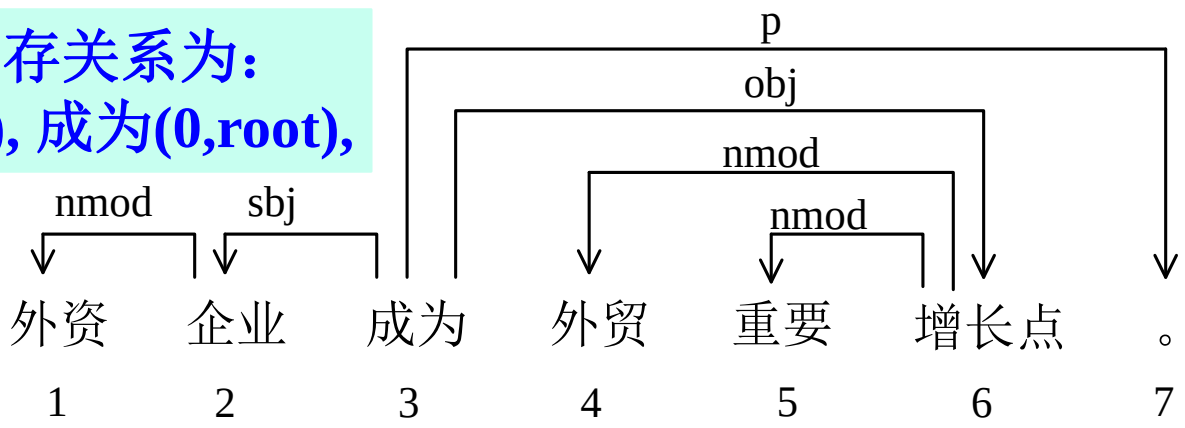


✧系统输出的依存分析树为：



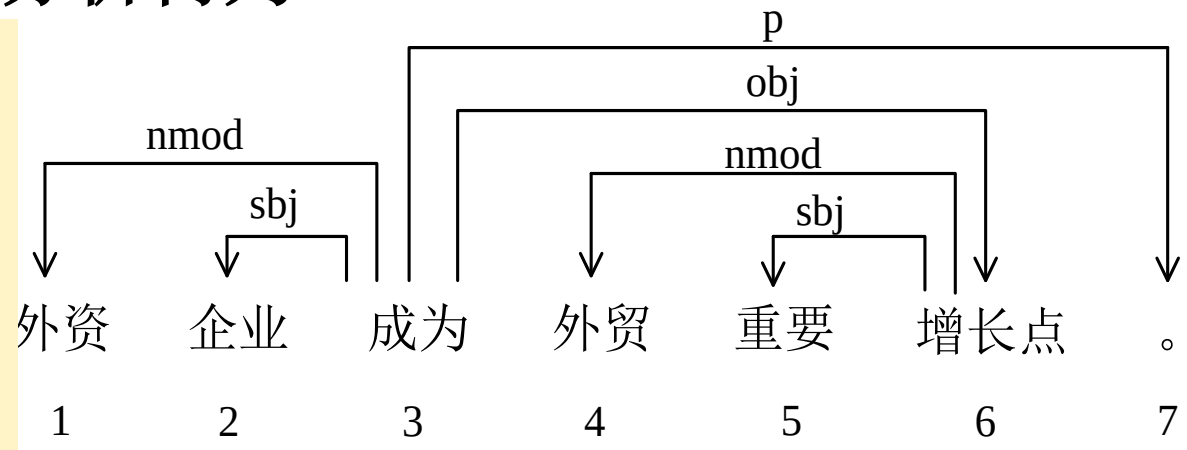
9.11 依存句法分析器性能评价

每个词对应的支配词及依存关系为：
外资(2,nmod), 企业(3,sbj), 成为(0,root),
外贸(6,nmod),
重要(6,nmod),
增长点(3,obj), 。(3,p)



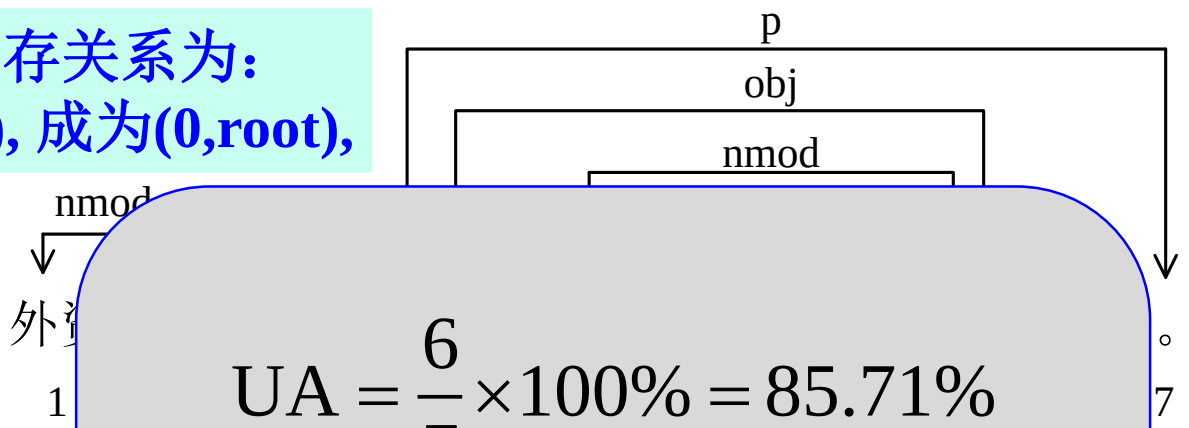
✧ 系统输出的依存分析树为：

每个词对应的支配词及依存关系为：外资(3,nmod)，
企业(3,sbj)，成为(0,root)，
外贸(6,nmod)，
重要(6,sbj)，
增长点(3,obj)，。(3,p)



9.11 依存句法分析器性能评价

每个词对应的支配词及依存关系为：
外资(2,nmod), 企业(3,sbj), 成为(0,root),
外贸(6,nmod),
重要(6,nmod),
增长点(3,obj), 。(3,p)



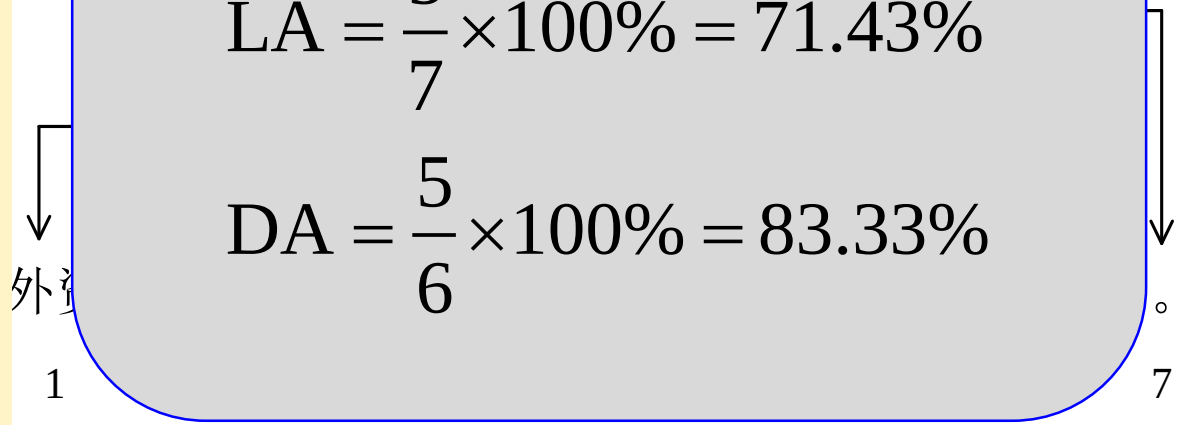
$$UA = \frac{6}{7} \times 100\% = 85.71\%$$

$$LA = \frac{5}{7} \times 100\% = 71.43\%$$

$$DA = \frac{5}{6} \times 100\% = 83.33\%$$

✧ 系统输出的依存分

每个词对应的支配词及依存关系为：外资(3,nmod)，企业(3,sbj)，成为(0,root)，外贸(6,nmod)，重要(6,sbj)，增长点(3,obj)，。(3,p)



9.11 依存句法分析器性能评价

◆ 性能现状

- 英语依存句法分析器测试
  - ✧ 训练语料：《华尔街日报》第02-21章；
  - ✧ 测试语料：《华尔街日报》 第23章。

9.11 依存句法分析器性能评价

分析器	UAS (%)	DA (%)	RA (%)	CM (%)	POS (%)
Charniak00	—	92.1	95.2	45.2	97.1
Collins97	—	91.5	95.2	43.3	97.1
Yamada03	—	90.3	91.6	38.4	97.1
Nivre04	87.1	87.3	84.3	30.4	96.1

1、Charniak00 和 Collins97 均指短语结构句法分析器，用中心词提取规则将短语结构分析结果转化依存分析树。

2、Yamada03 和 Nivre04 是基于转换的确定性方法。

3、POS(%)指词性标注器自动标注的正确率。

9.11 依存句法分析器性能评价

- 汉语依存句法分析器测试
  - 利用宾州中文树库进行转换

	树库句子	词数
训练集	001-815, 1001-1136	434,936
开发集	886-931, 1148-1151	21,595
测试集	816-885, 1137-1147	50,319

9.11 依存句法分析器性能评价

分析器	DA (%)	RA (%)	CM (%)
Duan07	84.36	73.70	32.70
Zhang08	86.21	76.26	34.41

[Duan07]X. Duan, J. Zhao and B. Xu. 2007. Probabilistic models for action-based Chinese dependency parsing. In *Proc. of ECML-ECPPKDD*, pp: 559-566

[Zhang08]Y. Zhang and S. Clark. 2008. A tale of two parsers: investigating and combining graph-based and transition-based dependency parsing using beam-search. In *Proc. of EMNLP*, pp: 562-571

9.11 依存句法分析器性能评价

分析器	UAS (%)
Chen08	86.52
Yu08	87.26
Zhao09	86.1

训练和测试数据都来源于《人民日报》标注语料。

说明：依据不同训练语料和测试语料获得的性能指标没有可比性；相同的分析方法采用不同的参数可获得不同的分析结果；来自不同论文的测试结果往往没有可比性，除非作者明确说明与对比系统具有严格相同的测试条件和数据。

9.11 依存句法分析器性能评价

新的进展：

- Performance of neural network transition-based dependency parser (with novel features) evaluated on the CoNLL'09 shared task dependency TreeBanks: English: UAS:94.10%, LAS:92.55%

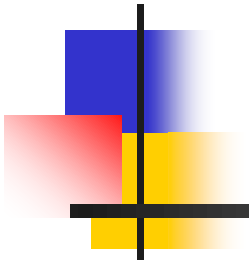
  Chris Alberti *et al*., Improved Transition-Based Parsing and Tagging with Neural Networks. *Proc. EMNLP*'2015

- **Using LSTMs: Chinese: UAS:85.96%, LAS:84.40%;**
  English: UAS:92.57%, LAS:90.31%.

  Miguel Ballesteros *et al*. Improved Transition-based Parsing by Modeling Characters instead of Words with LSTMs. *Proc. EMNLP*'2015

部分公开的依存句法分析器

- Stanford Parser
  http://nlp.stanford.edu/downloads/lex-parser.shtml
- MST Parser (Minimum-Spanning Tree Parser)
  http://www.seas.upenn.edu/~strctlrn/MSTParser/MSTParser.html
- MaltParser http://maltparser.org/index.html
- MINIPAR Parser (only for English)
  http://webdocs.cs.ualberta.ca/~lindek/minipar.htm
- Layer-based Dependency Parser (LDPar)
  http://www.openpr.org.cn/ (访问Download→ NLP Toolkit)



9.12 短语结构与依存结构的关系

9.12 短语结构与依存结构的关系

◆短语结构可转换为依存结构

➢实现方法：

(1) 定义中心词抽取规则，产生中心词表；

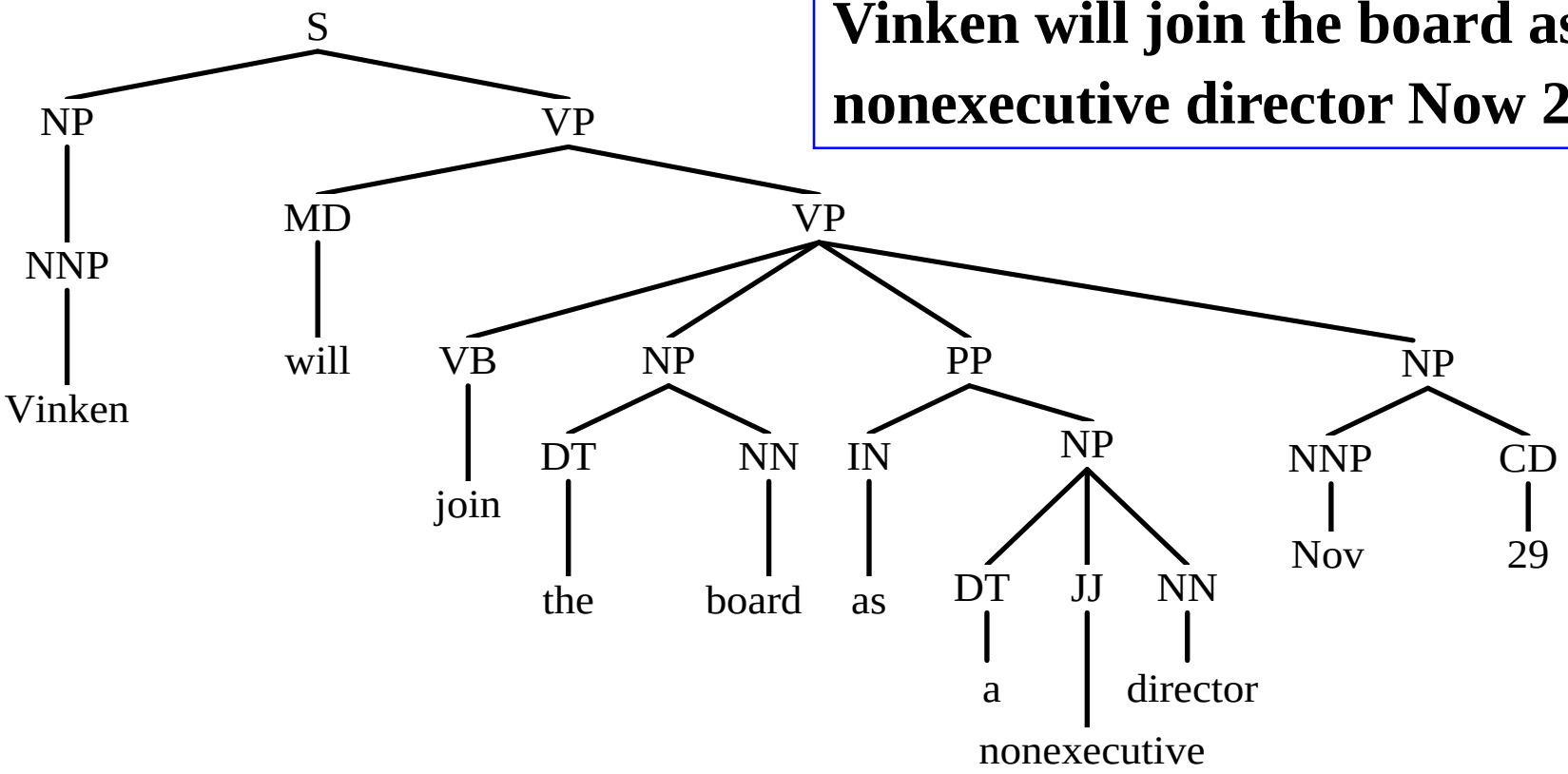
(2) 根据中心词表，为句法树中每个节点选择中心子节点；

(3) 将非中心子节点的中心词依存到中心子节点的中心词上，得到相应的依存结构。

9.12 短语结构与依存结构的关系

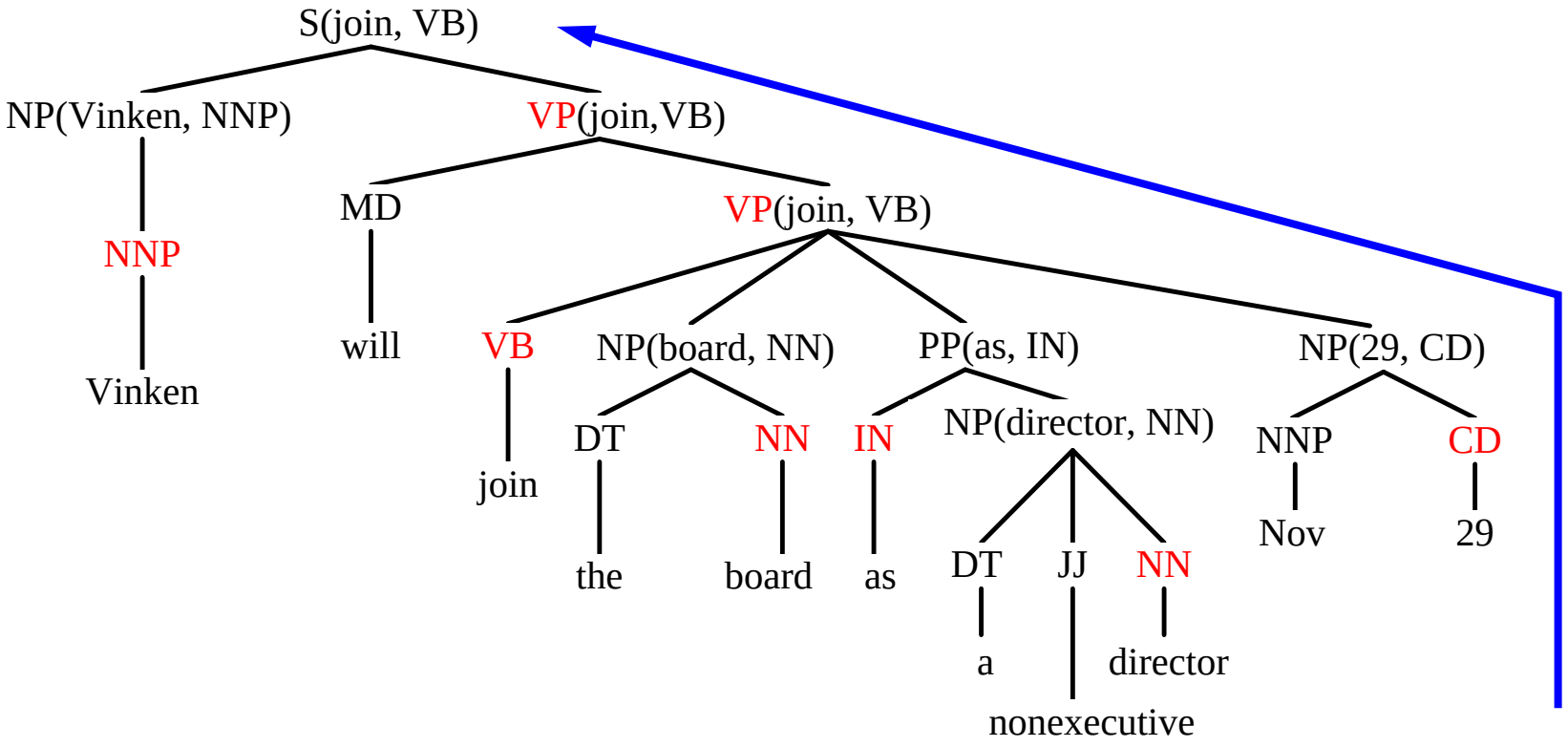
例如：给定如下短语结构树

Vinken will join the board as a nonexecutive director Now 29



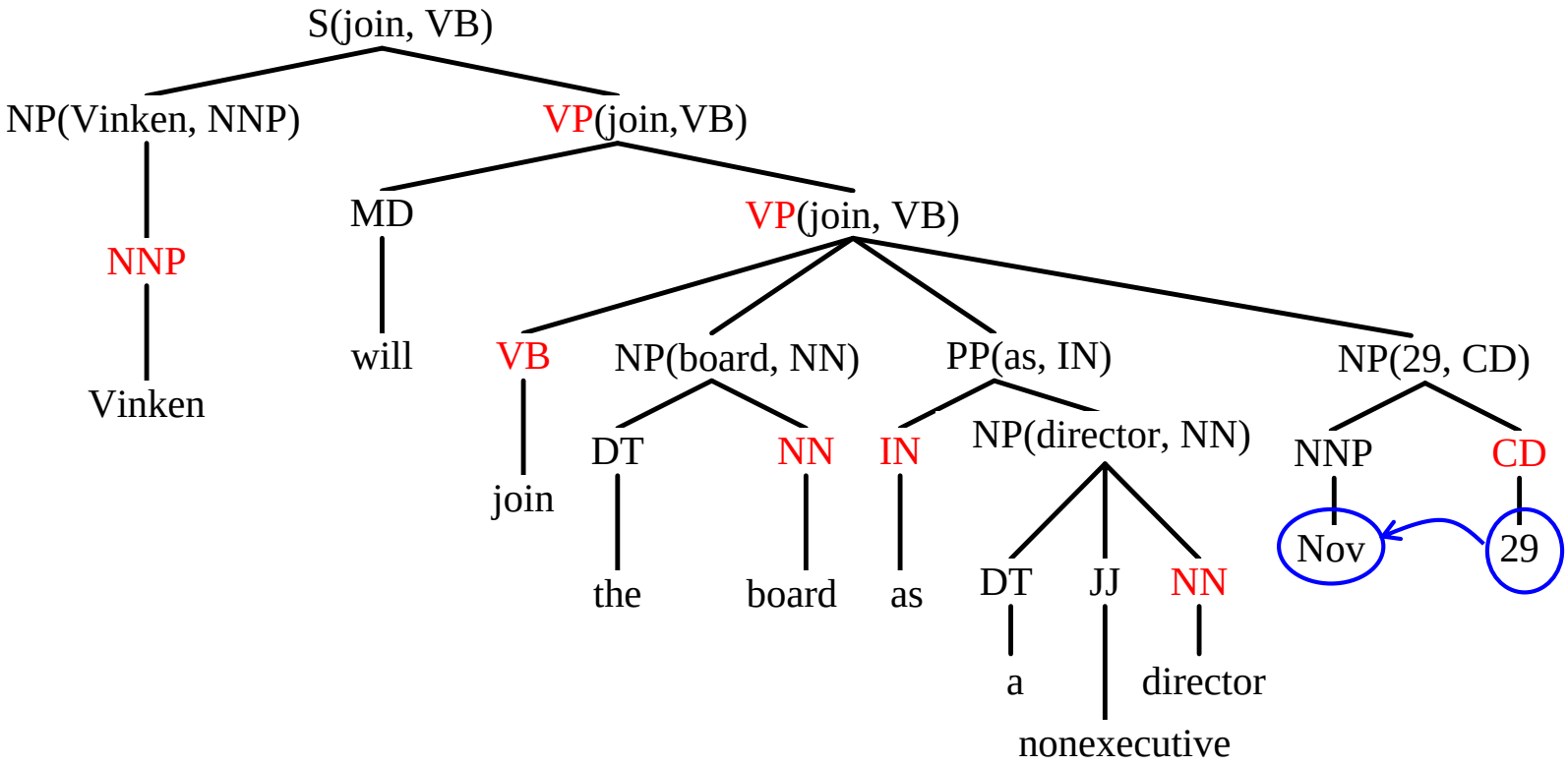
9.12 短语结构与依存结构的关系

- 根据中心词表为每个节点选择中心子节点 (中心词通过自底向上传递得到)



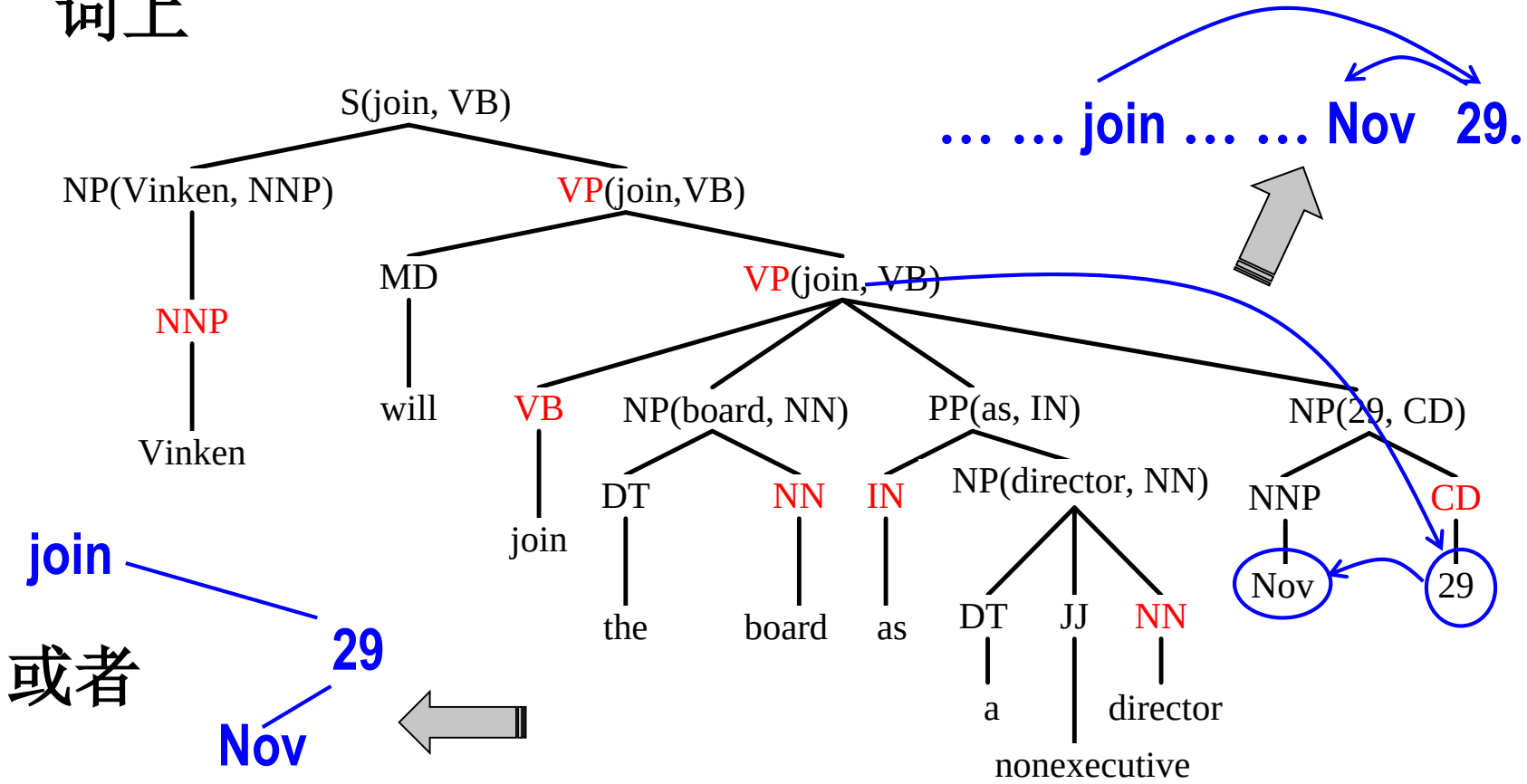
9.12 短语结构与依存结构的关系

- 将非中心子节点的中心词依存到中心子节点的中心词上



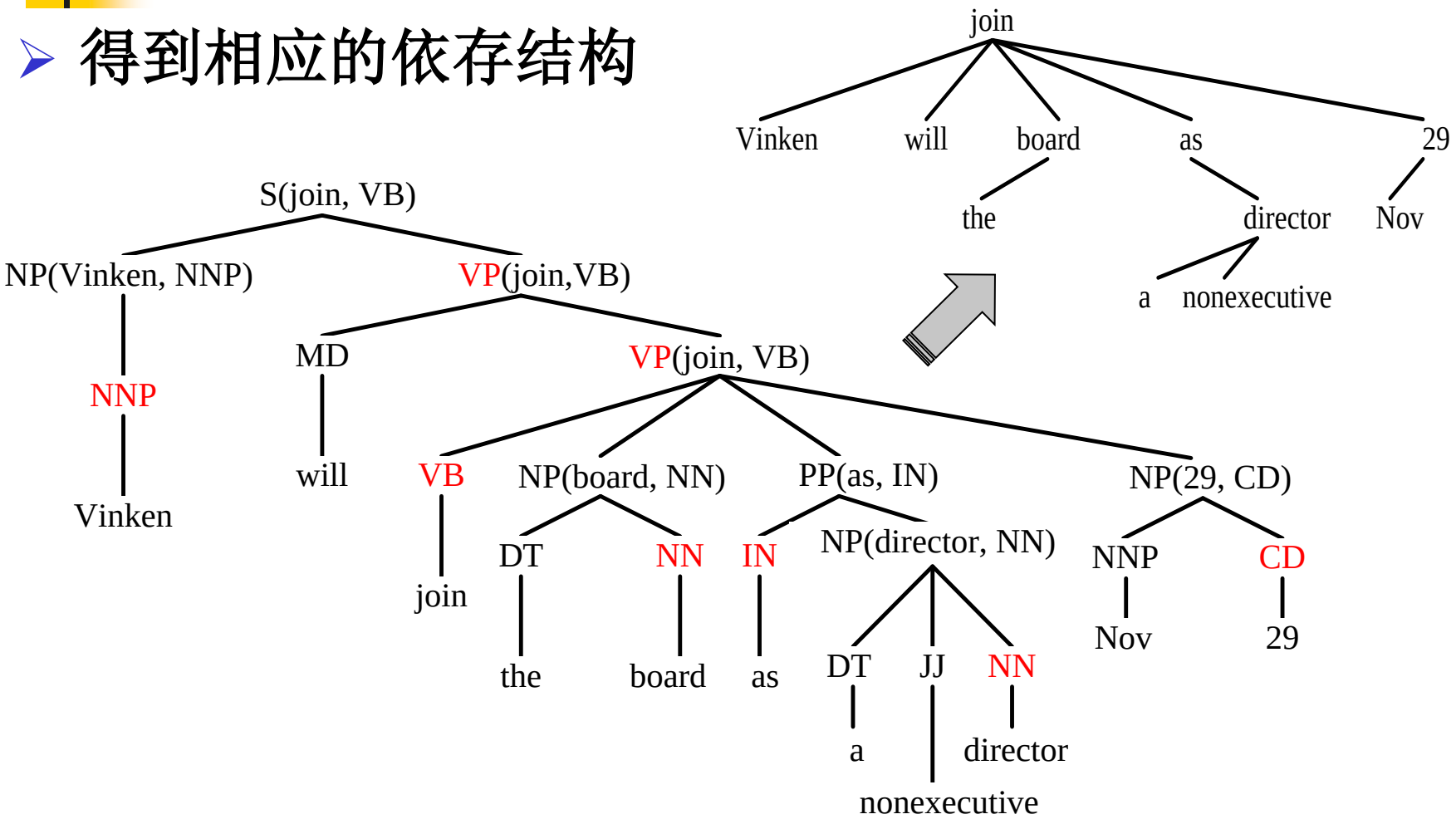
9.12 短语结构与依存结构的关系

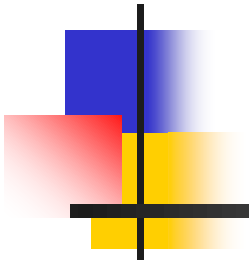
- 将非中心子节点的中心词依存到中心子节点的中心词上



9.12 短语结构与依存结构的关系

- 得到相应的依存结构





9.13 汉英句法结构特点对比

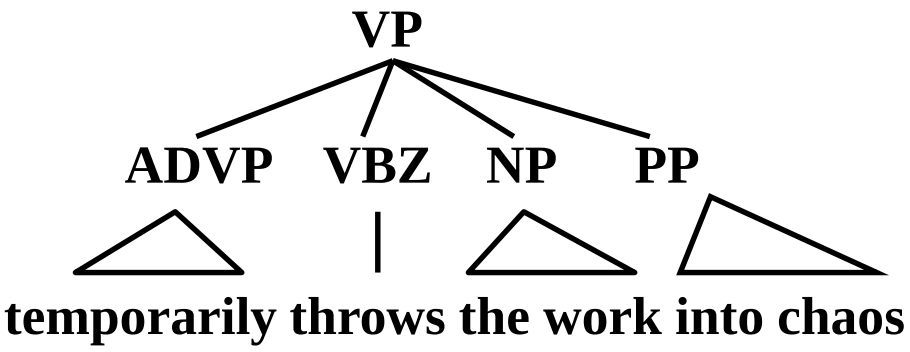
9.13 汉英句法结构特点对比

说明：撇开汉语的分词问题和词性消歧错误可能对句法分析器带来的影响来讨论汉英句法结构特点的比较问题，即保证句法分析器的输入为完全正确的词性序列，仅仅考虑句子结构本身的问题。

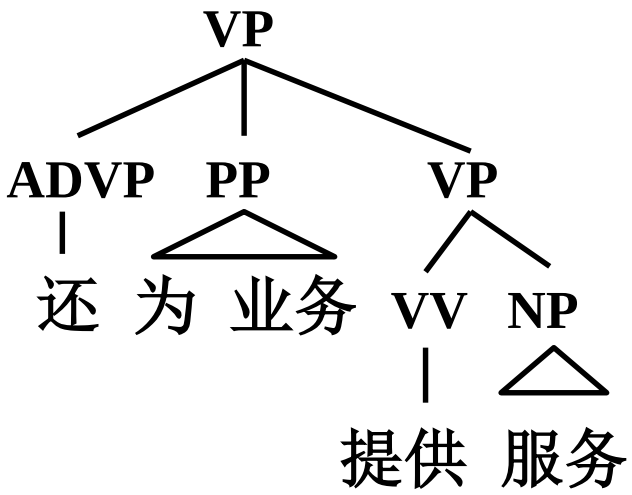
(1) 汉语比英语更少地使用功能词(function words)，且没有形态变化：汉语中不使用限定词(“这、这个、那个”等)的名词普遍存在，复数标记(“们”等)有限并且很少出现。

9.13 汉英句法结构特点对比

(2) 英语短语绝大多数以左部为中心，而汉语短语比较复杂，大多数短语类是以右部为短语中心，除了动词和介词的补语在它们的中心词之后。如：



(c)



(d)

9.13 汉英句法结构特点对比

在例图(c)中，介词短语 **into chaos** 在动词 **throw** 的右边，而在图 **(d)** 所示的汉语例子中恰好相反，介词短语“为业务”在动词前面。这种差异意味着在英语句子中附加在动词后面的补语引起的歧义是句法分析器需要解决的主要问题，而在汉语句子中很少有这种歧义存在。

9.13 汉英句法结构特点对比

(3) 在汉语句子中没有做主语的先行代词的情况普遍存在，但在英语中这种情况很少出现。这样就使得汉语句法分析器很难判断一个输入到底是没有主语的子句(IP)结构还是仅仅是一个动词短语VP，如：

He thinks it is true.　/ 他认为□是对的。

9.13 汉英句法结构特点对比

从本质上讲，英语是一种“结构型”语言，一个完整的句法结构即表示一个完整的句子。当多个单句连接起来构成复句的时候，单句与单句之间需要有显式的连接词或者短语。汉语则不同，汉语“表意型”的语言特点，使得汉语句子通常受语义的牵引，一个句子是表达一个完整意义的语言单元，这种特点在长句中表现得特别明显。因此，在汉语中存在一种独特的长句构成方式，就是一连串独立的简单句通过逗号或分号，连接成一个复杂的“句群”式的长句。

9.13 汉英句法结构特点对比

这些长句内部的各个简单句是为了表意的需要而连接在一起的，它们彼此的句法结构完全是独立的，表示彼此之间逻辑关系的连接词不是必需的。因此，在很多情况下，它们之间的分隔标记仅仅是一个逗号或者分号。这类长句在汉语中称之为“流水复句”，例如：

“我现已步入中年，每天挤车，搞得我精疲力尽，这种状况，直接影响我的工作，家里的孩子也没人照顾。”

9.13 汉英句法结构特点对比

从中文资源联盟 (Chinese LDC) 发布的汉语树库(TCT 973)中随机地抽取出 4431 个长度超过20个词的长句，其中，流水复句有1830个，占全部长句的41.3%[李幸, 2005]。

9.13 汉英句法结构特点对比

◆汉语长句的层次化句法分析方法

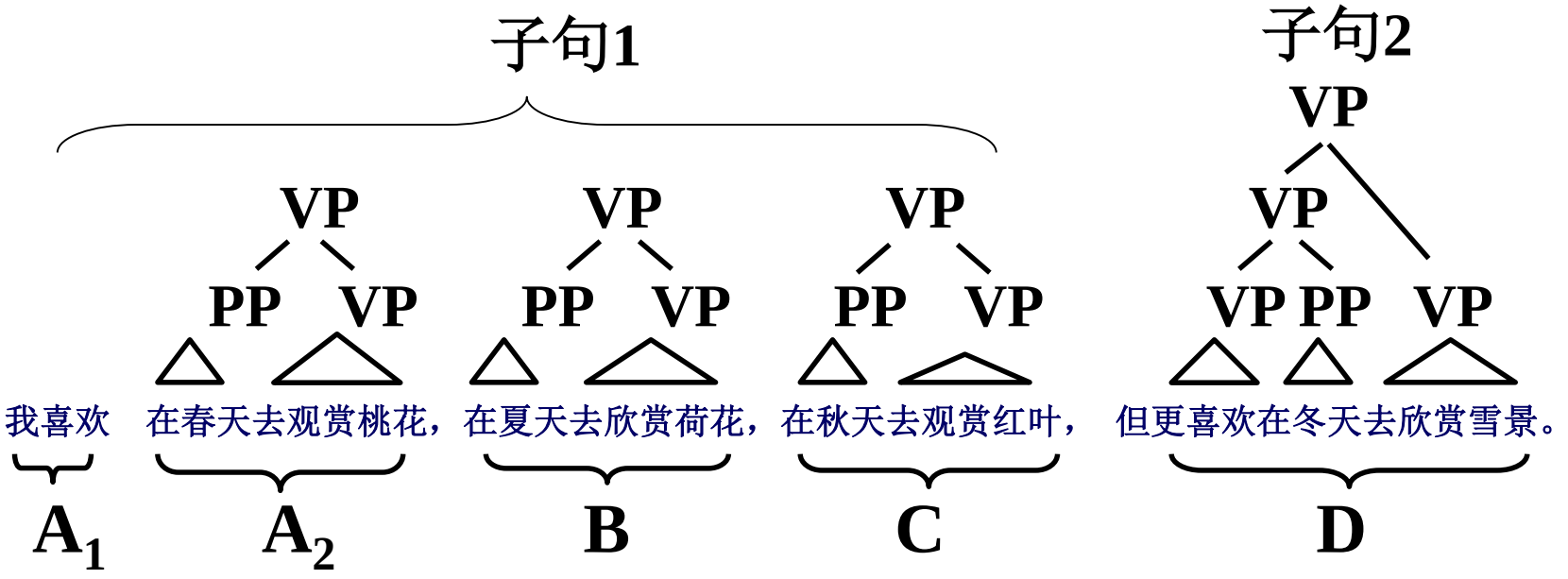
(1) 对包含“分割”标点的长句进行分割；

(2) 对分割后的各个子句分别进行句法分析(即第一级分析)，分析得到的各个最大概率的子树根节点的词类或者短语类别标记作为第二级句法分析的输入；

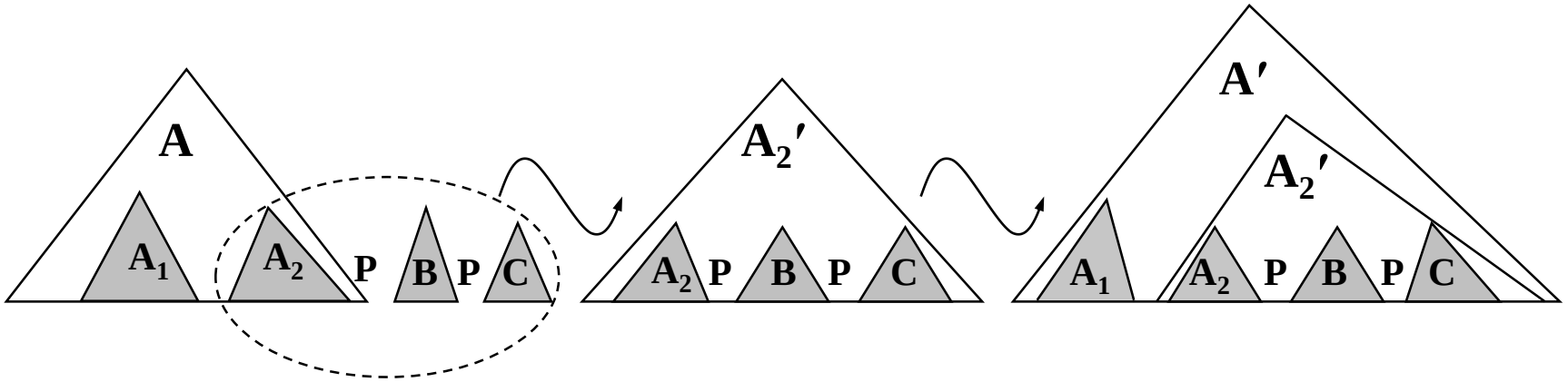
(3) 通过第二遍分析找到各子句或短语之间的结构关系，从而获得最终整句的最大概率分析树。

9.13 汉英句法结构特点对比

例句：我喜欢在春天去观赏桃花，在夏天去欣赏荷花，在秋天去观赏红叶，但更喜欢在冬天去欣赏雪景。



9.13 汉英句法结构特点对比



参见论文：

李幸，宗成庆，2006， 引入标点处理的层次化汉语长句句法分析方法，中文信息学报，20(4): 8-15

李幸，汉语句法分析方法研究[硕士学位论文]，中科院自动化所，2005 年6月

本章小结

- 句法分析的任务，面临的困难
- 短语结构分析方法
  - 基于规则的方法：
    - ❖ Chart Parsing
    - ❖ CYK 方法
  - 基于概率上下文无关文法 PCFG
    - 快速计算分析树的概率(内向算法)
    - 快速计算最大概率分析树(Viterbi 算法)
    - 参数估计(内外向算法)
  - 短语结构分析器实现方法、性能评测、开源工具

本章小结

- 依存句法分析
  - 基本方法
  - 依存句法分析器实现、性能评价、开源工具
  - 短语结构与依存结构
- 汉语句法结构特点对比

延伸阅读：

- 利用高阶依存信息构造句法重排序模型
- 浅层句法分析
  - Base NP 定义
  - 基于SVM 模型的 Base NP 识别方法

习题

9-1. 编写程序实现自顶向下(top-down)的 Chart 分析器，体会自顶向下和自底向上(bottom-up)分析算法的不同。

9-2. 自学Left Corner 分析算法和 Tomita GLR 句法分析算法。

9-3. 如有条件，利用树库语料抽取 PCFG 规则，结合 Chart 分析算法实现一个基于 PCFG 的句法分析器。

9-4. 参阅各种开源句法分析器网站(包括短语结构分析器和依存句法分析器)的相关内容，并试用相关分析器。

Thanks

进一步了解

附1：利用高阶依存信息构造句法重排序模型

附1: 利用依存信息进行句法重排序

- **基本思想**：在传统的基于PCFG的句法分析器中，要求的独立性假设过强，缺乏上下文词汇信息的帮助；词汇化方法 LPCFG(Lexicalized PCFG) 引入了中心词信息，改善了 PCFG 方法的性能。但是，中心词所携带的信息是有限的，因此，我们认为，可通过引入词汇依存信息，进一步改善句法分析器的性能。

附1: 利用依存信息进行句法重排序

- **实现方法**：利用判别式句法分析模型 —
  - 句法树的评价得分(*Score*)可以通过下式计算：

$$Score(x,c) = \Phi(x,c) \cdot \overline{\alpha}$$

    - 其中 ***x*** 为输入句子，***c*** 为句法树；
    - $\Phi(x, y) \in \Re^d$ 为句法树的特征向量；
    - $\overline{\alpha} \in \Re^d$ 为特征权值向量。
  - 句法分析的过程就是找到得分最高的句法树：

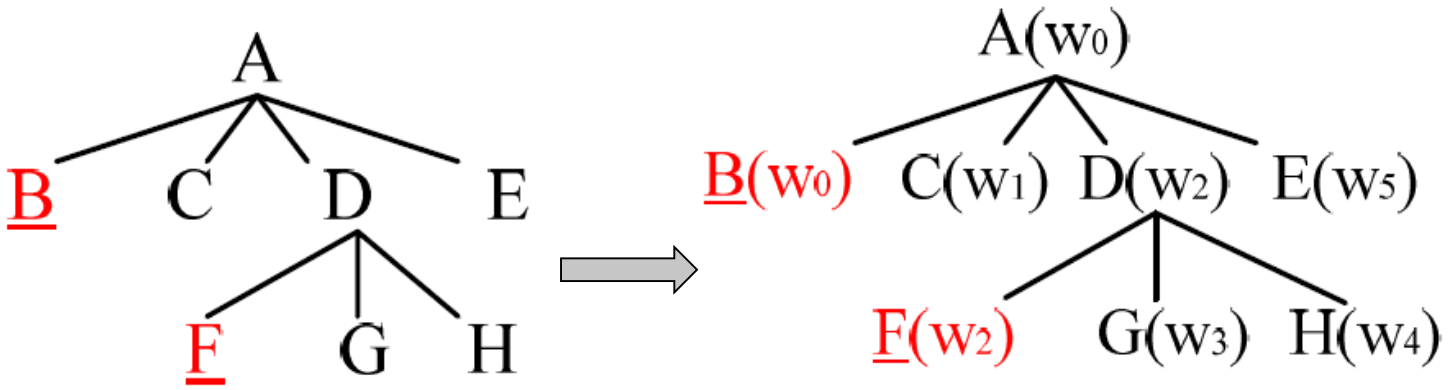
$$c^* = \arg \max_{c \in GEN(x)} Score(x,c)$$

其中，函数 GEN(x) 用于为句子 x 枚举出句法树候选。

附1: 利用依存信息进行句法重排序

- 将短语结构树转换为对应的依存分析树：

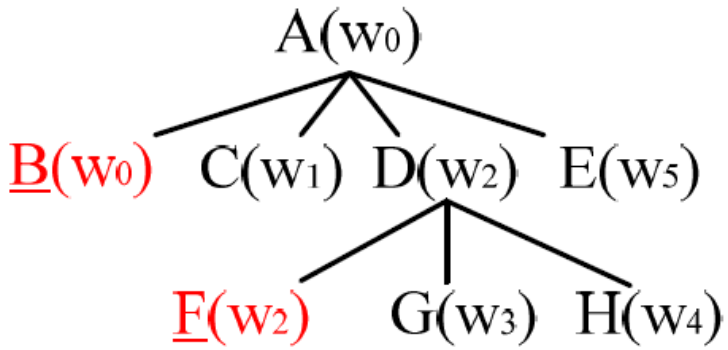
第一步：词汇化短语结构树



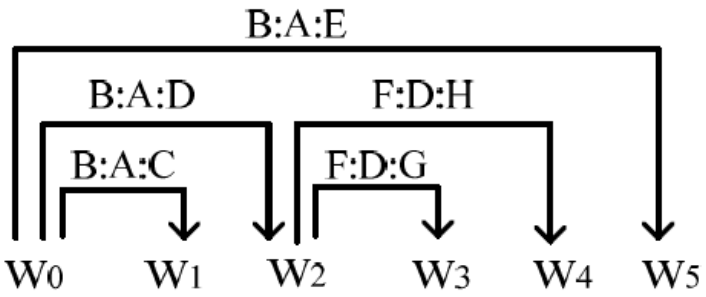
(a) Constituent tree　　(b) Lexicalized constituent tree

附1: 利用依存信息进行句法重排序

第二步：将词汇化的短语树转换为有标记的依存分析树



(b) Lexicalized constituent tree

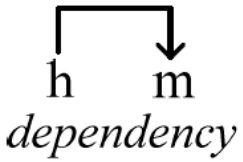


(c) Labeled dependency tree

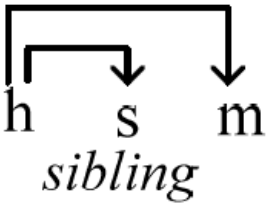
附1: 利用依存信息进行句法重排序

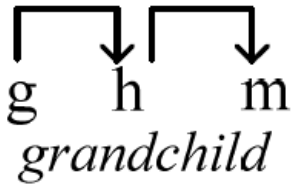
✧ 各种词汇依存结构

- 一阶词汇依存结构

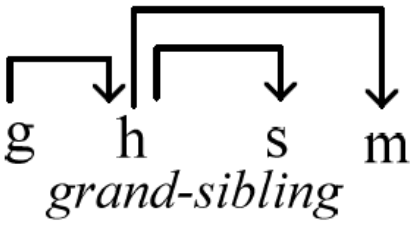


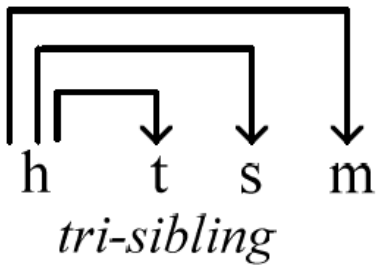
- 二阶词汇依存结构





- 三阶词汇依存结构

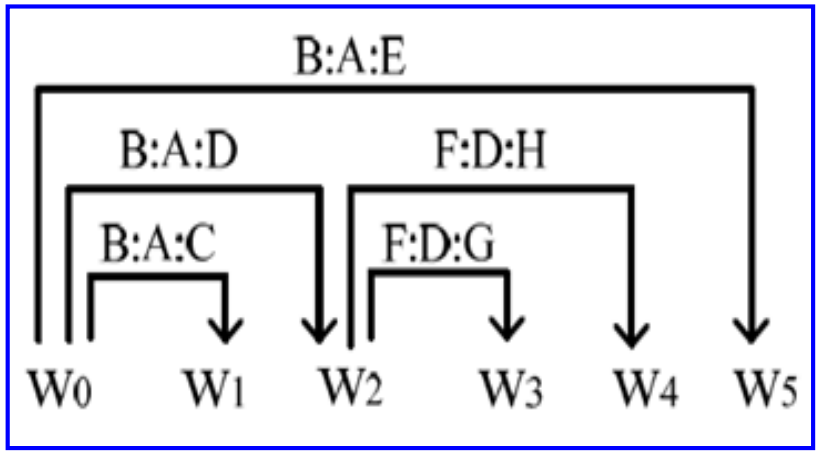


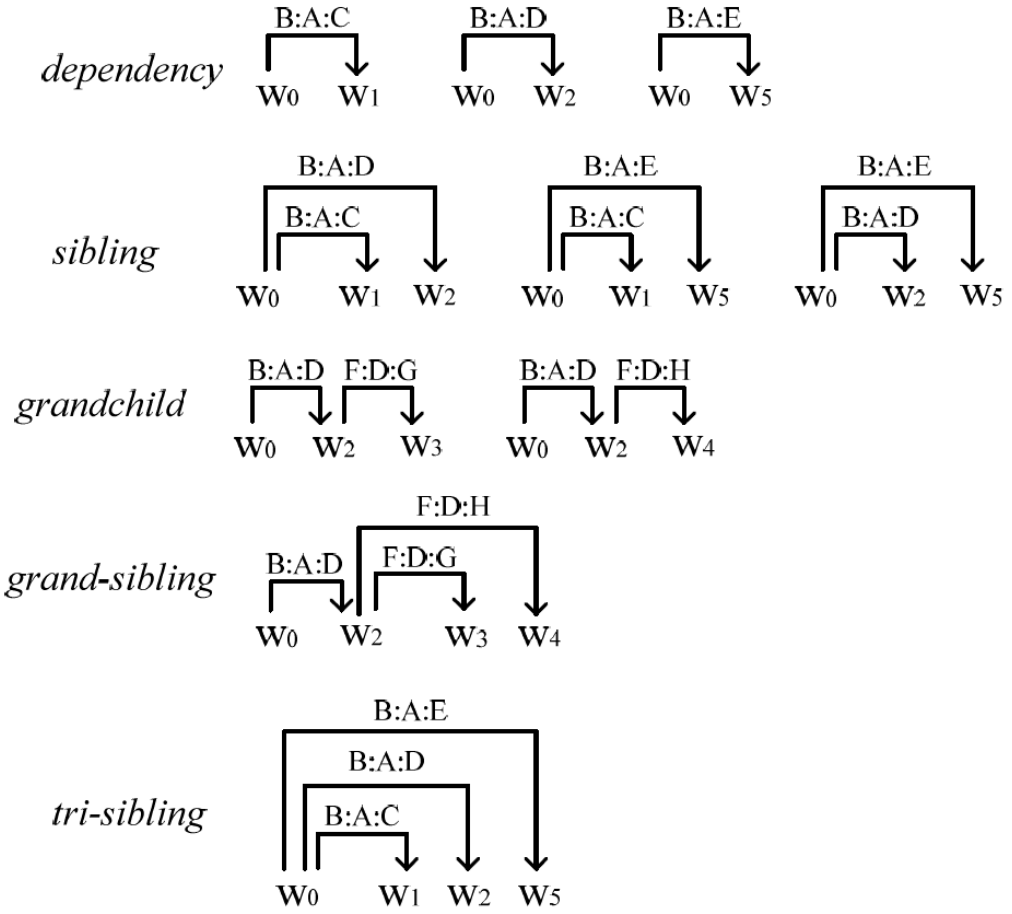


附1: 利用依存信息进行句法重排序

- 抽取词汇依存结构

如前面图中节点 *A* 的各种词汇依存结构为：





附1: 利用依存信息进行句法重排序

- 定义词汇依存特征模板，构造特征向量

h m *dependency*	**Basic Uni-gram Features** h , POS(h), N(h) h , POS(h) h , N(h) m , POS(m), N(m) m , POS(m) m , N(m) **Basic Bi-gram Features** P(m) , h , POS(h), N(h), m), POS(m), N(m) h , POS(h), N(h), m , POS(m), N(m) P(m) ,POS(h), N(h), POS(m), N(m) P(m) , h , N(h), m , N(m) P(m) ,h , POS(h), m , POS(m) P(m) ,h , m P(m) , POS(h),POS(m) P(m) , N(h), N(m) **Surrounding Word POS Features** P(m), N(h), POS(h), N(m), POS(m), POS(h)+1, POS(m)-1 P(m), N(h), POS(h), N(m), POS(m), POS(h)-1, POS(m)-1 P(m), N(h), POS(h), N(m), POS(m), POS(h)+1, POS(m)+1 P(m), N(h), POS(h), N(m), POS(m), POS(h)-1, POS(m)+1	h s m *sibling*	POS(h),N(h),POS(s),N(s),P(s),POS(m),N(m),P(m) POS(h),N(h),N(s),P(s),N(m),P(m) POS(h),N(h),POS(s),P(s),POS(m),P(m) POS(h),N(h),POS(s),N(s),POS(m),N(m) POS(h),POS(s),POS(m) N(h),N(s),N(m) N(h),P(s),P(m)
		g h m *grandchild*	POS(g),N(g),POS(h),N(h),P(h),POS(m),N(m),P(m) POS(g),N(g),N(h),P(h),N(m),P(m) POS(g),N(g),POS(h),P(h),POS(m),P(m) POS(g),N(g),POS(h),N(h),POS(m),N(m) POS(g),POS(h),POS(m) N(g),N(h),N(m) N(g),P(h),P(m)
		g h s m *grand-sibling*	POS(g),POS(h),POS(s),POS(m) N(g),N(h),N(s),N(m) N(g),P(h),P(s),P(m)
		h t s m *tri-sibling*	POS(h),POS(t),POS(s),POS(m) N(h),N(t),N(s),N(m) N(h),P(t),P(s),P(m)

附1: 利用依存信息进行句法重排序

- 短语结构分析器性能测试

(宾州中文树库)

Parser	F1 (%)	
	<=40	all
Charniak	85.97	82.41
Berkeley	86.57	83.13
NbestRerank	87.84	84.68
ForestRerank	88.91	85.72
EarlyUpdate	**88.97**	**85.74**

F1 scores on Test Set

附1: 利用依存信息进行句法重排序

- 短语结构对应的依存结构的准确率比较

Parsers	UA(%)
Charniak	82.31
Berkeley	84.05
NbestRerank	85.89
ForestRerank	85.69
EarlyUpdate	86.26
MST 1-ord (automatic POS)	79.62
MST 2-ord (automatic POS)	80.24
MST 1-ord (gold-standard POS)	85.23
MST 2-ord (gold-standard POS)	86.66

Unlabeled dependency accuracy (UA).

附1: 利用依存信息进行句法重排序

✧ 与其他系统的对比

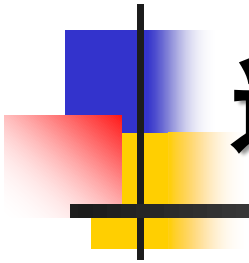
请参阅如下论文：

[1]Zhiguo Wang and Chengqing Zong. Phrase Structure Parsing with Dependency Structure. *Proc. 23rd COLING*, 2010, Pages 1292 -1300

[2]《软件学报》, 23(10): 2628−2642, 2012

Individual System	
(Petrov and Klein, 2007)	83.32
(Huang and Harper, 2009)	84.15
***N*-best Reranking**	
Charniak & Johnson Reranker[10]	83.30
Our NbestRerank System	84.68
Parsers Combination	
(Zhang et al., 2009)	85.45
Using Extra Resource	
(Burkett and Klein, 2008)	84.24
(Huang and Harper, 2009)	85.18
(Niu et al., 2009)	85.20
Reranking with Lexical Dependencies	
Our EarlyUpdate System	**85.74**

F1 (%) scores of state-of-the-art methods compared with ours on the Chinese Treebank.



进一步了解

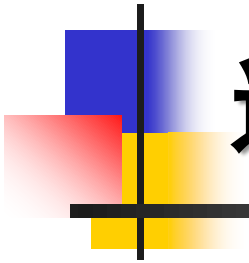
附2：最新进展

附2: 最新进展

1. Danqi Chen and Christopher Manning, A Fast and Accurate Dependency Parser using Neural Networks. *Proc. EMNLP*'2014 [用神经网络替换 SVM]
2. Meishan Zhang and Yue Zhang, Combining Discrete and Continuous Features for Deterministic Transition-based Dependency Parsing. *Proc. EMNLP*'2015 [将神经网络集成到 CRF 中]
3. Noah A. Smith, Chris Dyer, Wang Ling, Miguel Ballesteros and Austin Matthews, Transition-Based Dependency Parsing with Stack Long Short-Term Memory, *Proc. ACL-IJCNLP*'2015 [使用 LSTM]
4. Yue Zhang, Hao Zhou, Shujian Huang and Jiajun Chen, A Neural Probabilistic Structured-Prediction Model for Transition-Based Dependency Parsing, *Proc. ACL-IJCNLP*'2015 [将基于住搜索的转换依存方法扩展到神经网络方法]
5. Taro Watanabe and Eiichiro Sumita, Transition-based Neural Constituent Parsing, *Proc. ACL-IJCNLP*'2015 [使用神经网络方法进行短语结构分析]

附2: 最新进展

6. Oriol Vinyals, Lukasz Kaiser, Terry Koo et al., Grammar as a Foreign Language, https://papers.nips.cc/paper/5635-grammar-as-a-foreign-language.pdf [用端到端的神经机器翻译解码方法实现短语结构分析]
7. Wenduan Xu, Michael Auli and Stephen Clark,Expected F-Measure Training for Shift-Reduce Parsing with Recurrent Neural Networks. *Proc.NAACL-HLT*'2016 [将基于转换的方法与神经网络方法相结合，但训练目标函数不是最大似然估计，而是 F1-Measure]
8. Timothy Dozat and Christopher D. Manning, Deep Biaffine Attention for Neural Dependency Parsing. *Proc. ICLR*'2017. https://arxiv.org/pdf/1611.01734.pdf [将LSTM 用于基于图的依存句法分析]



进一步了解

附3: 局部句法分析

附3: 局部句法分析

S. Abney (1991) 首先提出了浅层句法分析的概念。浅层句法分析也称部分句法分析 **(partial parsing)** 或语块划分 **(chunking)**，它与完全句法分析不同，只要求识别句子中某些结构相对简单的独立成分，如：非递归的名词短语、动词短语等 。

浅层句法分析将句法分析任务分解为两个子任务：①语块的识别和分析；②语块之间的依附关系分析。

附3: 局部句法分析

根据 S. Abney 对语块的解释，语块是介于词和句子之间的具有非递归特征的核心成分。S. Abney(1995) 对英语语块的定义包含三个层次：

- 词 (words)
- 非递归的名词短语 (NPs)、动词词组 (VGs)、副词短语 (DPs)和介词短语 (PPs)
- 子句 (clause)

由于NPs、VGs 和 DPs、PPs属于不同的类，因此，又将第二类又进一步划分成“非递归的名词短语和动词词组”和“非递归的介词短语和副词短语”两类。

附3: 局部句法分析

◆ **Base NP 定义**

基本名词短语(base NP)指的是简单的、非嵌套的名词短语，不含有其他的子短语。

Base NP 的主要特点有两个：短语的中心语为名词；短语中不含有其他的子项短语，并且base NP之间结构上是独立的。

附3: 局部句法分析

Base NP 的形式化定义：

base NP → base NP + base NP

base NP → base NP + 名词 | 名动词

base NP → 限定性定词 + base NP | 名词

base NP → 限定性定词 + 名词 | 名动词

限定性定词 → 形容词 | 区别词 | 动词 | 名词 | 处所词 | 数量词 | 外文字串 | 数词和量词

附3: 局部句法分析

Base NP 识别 (base NP chunking) 就是从句子中识别出所有的 base NP。根据这种理解，一个句子中的成分可以简单地分为 base NP 和非 base NP 两类，那么，base NP识别就成为一个分类问题。

两种 base NP 表示方法：

- 括号分隔法 (the open/close bracketing)
- IOB 标注方法 (IOB tagging)

附3: 局部句法分析

例1: [Pierre Vinken], [61 years] old, will join [the board] as [a non-executive director] on [Nov. 29].

例2: When [it] is [time] for [their biannual powwow], [the nation]'s [manufacturing titans] typically jet off to [the sunny confines] of [resort towns] like [Boca Raton and Hot Springs].

例3：一个 于 [半个 世纪] 之后 重新 聚集 在 “[西南 联大]” [旗帜] 下 的 [奉献 活动] 开始了！

附3: 局部句法分析

在IOB标注方法中，字母‘B’(Begin)表示当前词语位于base NP的开端，字母‘I’(In) 表示当前词语在base NP内 (非短语首词语)，字母‘O’(Out) 表示词语位于base NP 之外。例如：

例4：外商/B 投资/I 成为/O 中国/B 外贸/I 重要/B 增长/I 。/O

与IOB方法类似的标注方法还有：IOE (In，Out，End) 表示方法，Start/End表示方法 (用5个标志符，O, B, E, I, S) 等。

附3: 局部句法分析

- Base NP 识别方法
  - 基于SVM 的识别方法
  - 基于WINNOW的 识别方法
  - 基于CRF的识别方法

附3: 局部句法分析

➢ 基于SVM 的识别方法

由于SVM算法解决的是二值分类问题，而base NP识别则是多值分类问题，因此，必须将 base NP识别转化为SVM可处理的问题，并充分利用句子中的上下文信息来提取特征。一般地，多值分类问题转化为二值分类问题有两种方法：配对策略(pairwise method) 和一比其余策略 (one vs. other method)。

附3: 局部句法分析

T. Kudo等(2003) 在利用 SVM 识别 base NP的系统(YamCha[1])中，主要使用了三类特征：

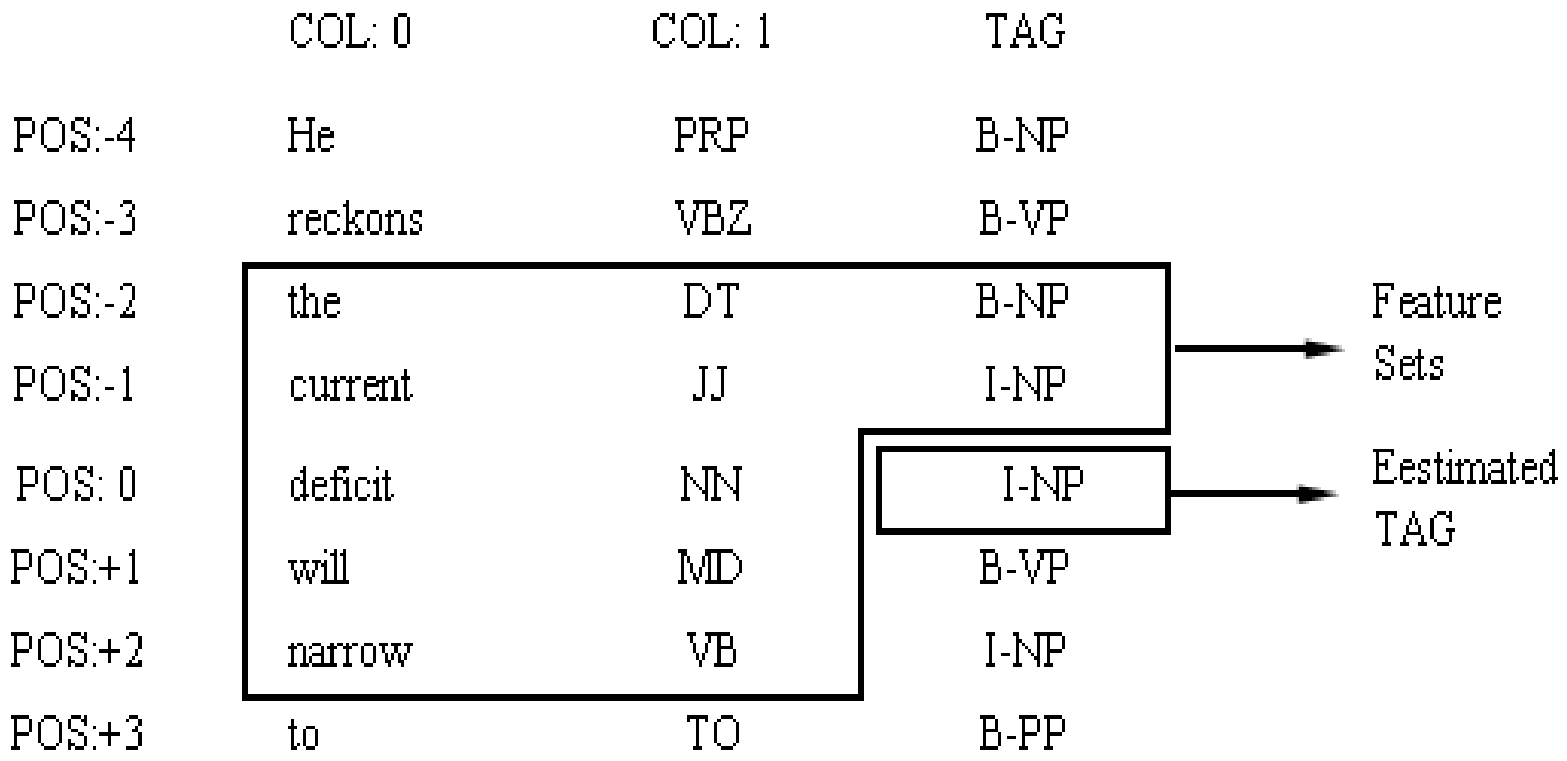
- 词： $w_{i-2}w_{i-1}w_i w_{i+1}w_{i+2}$
- 词性：$t_{i-2}t_{i-1}t_i t_{i+1}t_{i+2}$
- base NP 标志： $c_{i-2}c_{i-1}$

其中，w_i 为句子中位置 i 处的词，t_i 为词的词性，c_i 为要识别的第 i 个词的base NP标记。

[1] http://chasen.org/~taku/software/yamcha

附3: 局部句法分析

YamCha 系统识别 base NP 过程示意图：



附3: 局部句法分析

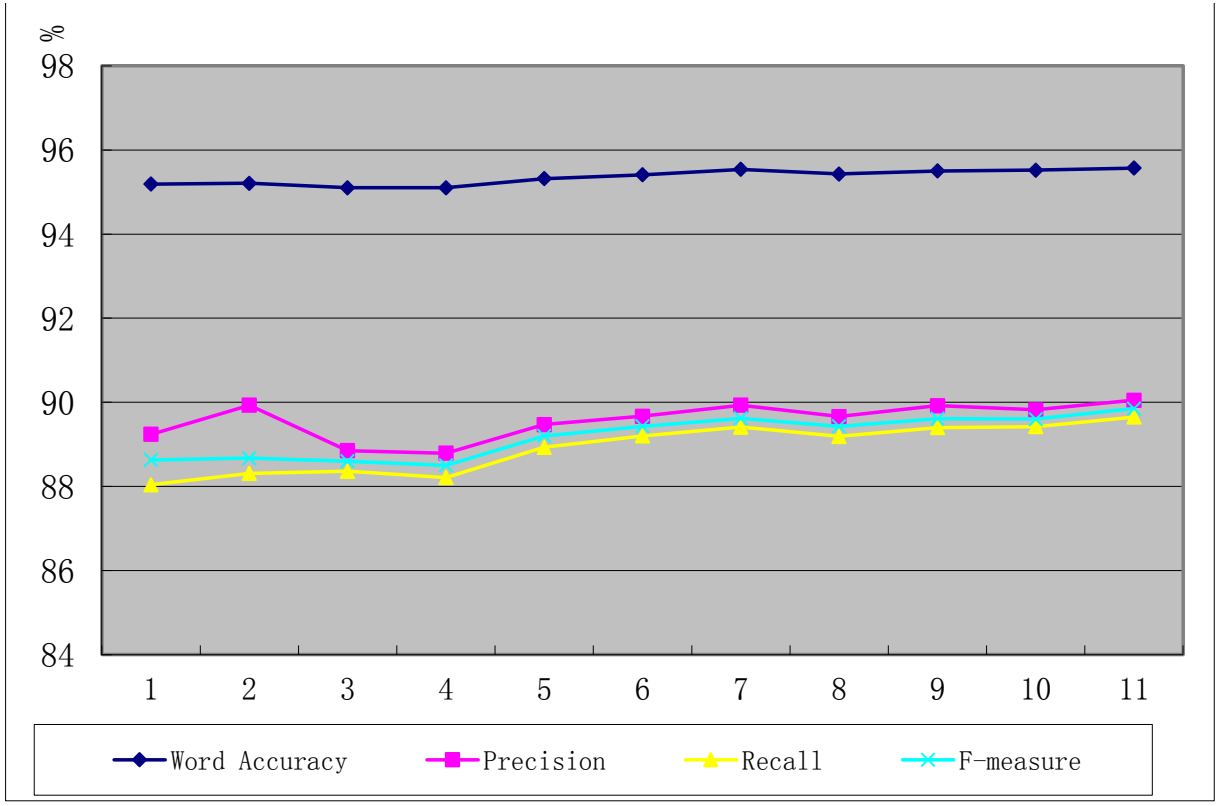
其中，POS列表示当前词(POS: 0)的前后词的位置；COL: 0 列表示给定句子，本例中为“He reckons the current deficit will narrow to”；COL: 1列为给定句子中各个词对应的词类标记，如，He的词类标记为PRP，reckons 的词类标记为VBZ等；TAG列为给定句子中的各个词被标记为base NP的情况，B-NP表示当前位置上的词为base NP的首词，I-NP表示当前位置上的词属于base NP。类似地，B-VP和I-VP分别表示当前位置上的词为VP的首词或内部词。当要估计位置POS: 0处词的base NP标记时，该词的前后各两个位置上的词及其他们的词性标记，以及前面两个词的base NP 标记共同作为被选取的特征。

附3: 局部句法分析

- 用于base NP 识别语料资源
  - ✧英文： CoNLL-2000 (Conference on Computational Natural Language Learning)提供的《华尔街日报》语料
    - 训练语料：15－18章，211,727个词
      http://www.cnts.ua.ac.be/conll2000/chunking/train.txt.gz
    - 测试语料： 第20章，47,377个词
      http://www.cnts.ua.ac.be/conll2000/chunking/test.txt.gz
    - 工具Chunkkink 用于语料格式转换。
      http://ilk.kub.nl/~sabine/chunklink/
  - ✧ 汉语：宾州 LDC 语料库，中文树库

附3: 局部句法分析

- 实验：训练语料规模对性能的影响：从CoNLL-2000中取100,000个词作为初始训练集，每次增加100,000个词。



F＝88.63~89.85

◆ 词的正确率
■ base NP正确率
▲ base NP召回率
× F值

附3: 局部句法分析

根据有关实验，最好的汉语base NP识别正确率比英语base NP 识别的正确率相差约5％左右。主要原因：

- 英语中大多数base NP有限定词，如形容词、冠词等，而汉语没有；
- 带多个名词修饰语的汉语 base NP 识别不完整，如：“高新技术”，“高”可能无法被识别；
- 连续名词序列组成的 base NP识别困难，如：中国/NR 红十字会/NR 名誉/NN 会长/NN 江泽民/NR；
- 汉语词性和语义更多样化。

徐昉，基本名词短语识别的关键技术研究[硕士学位论文]，中科院自动化所，2007年6月