

依存语法分析

依存语法分析

◆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 Tesnière, 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 d'une syntaxe structurale)中。

依存语法分析

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。

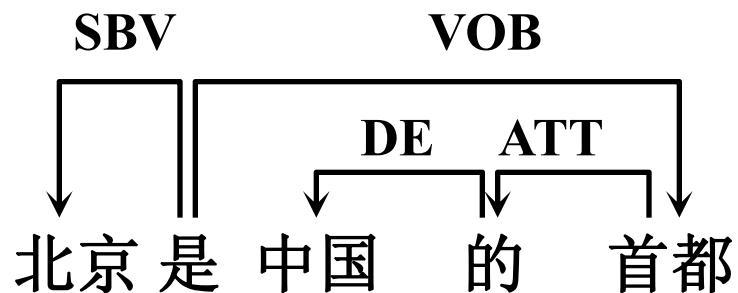
依存语法分析

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。

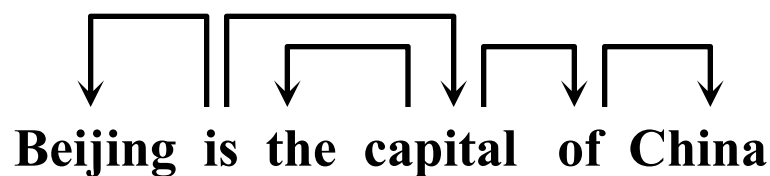
依存语法分析

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)。

依存语法分析



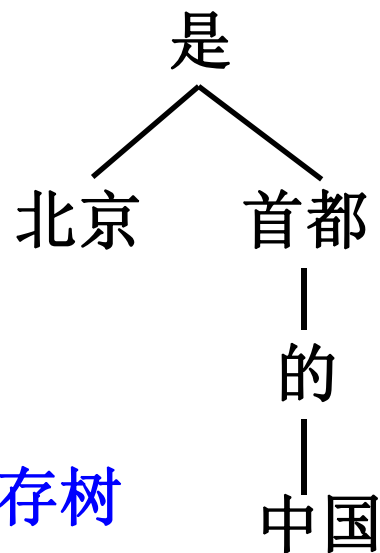
(e) 有向图-1



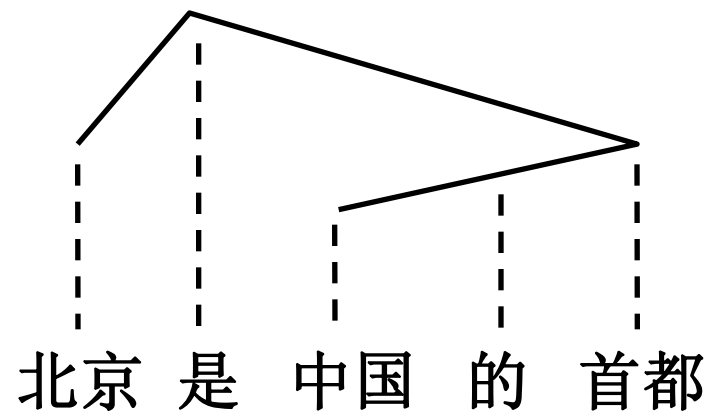
(e) 有向图-2

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。

依存语法分析



(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。

依存语法分析

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。

依存语法分析

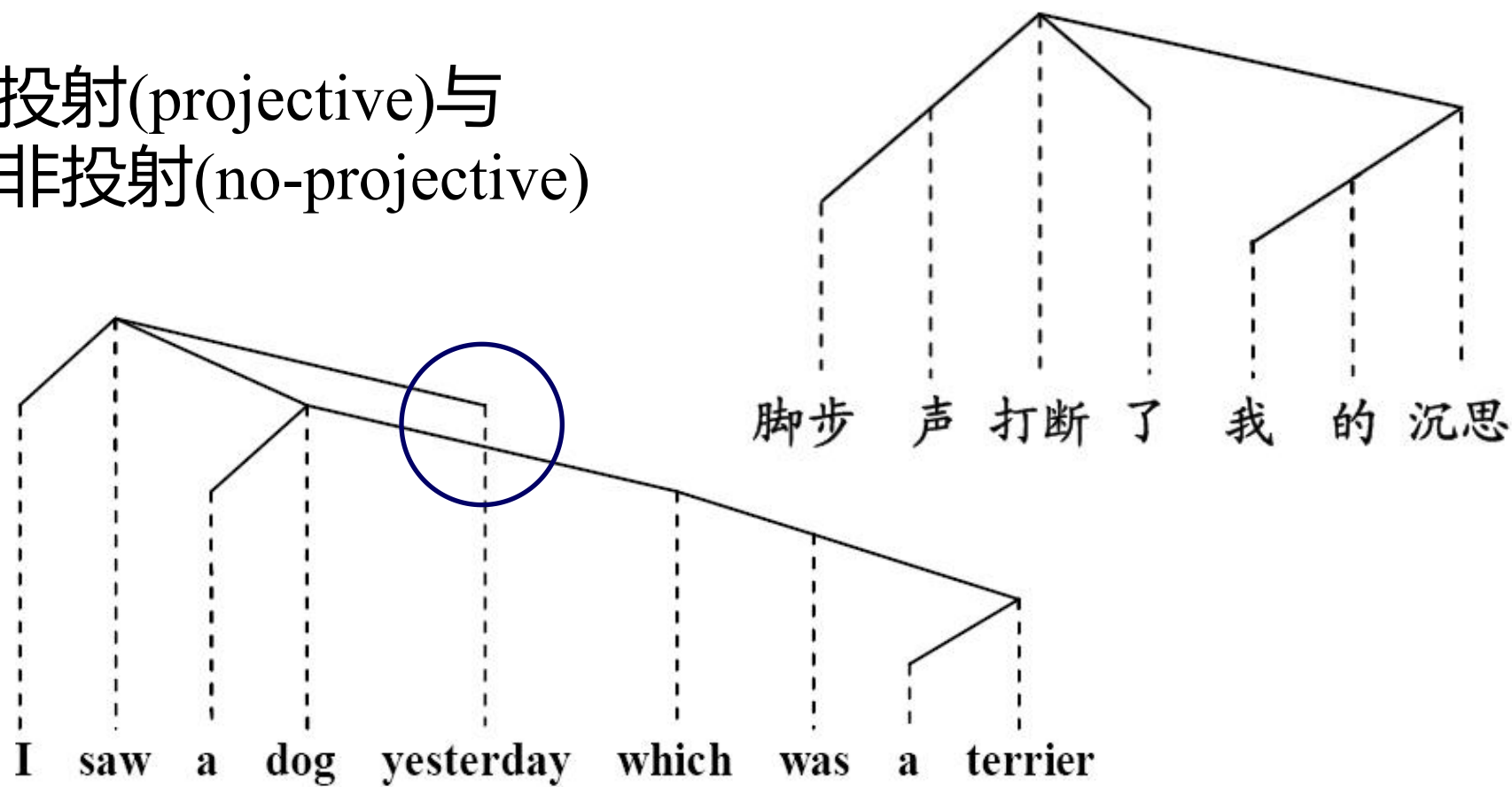
这4条公理相当于对依存图和依存树的形式约束为：

- 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。

依存语法分析

投射(projective)与
非投射(no-projective)



依存语法分析

◆ 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。

依存语法分析

◆依存句法分析方法

依存句法分(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) 语法规则或参数学习

依存语法分析

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)

依存语法分析

➤ 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。

依存语法分析

(1) 模型 A. 二元词汇亲和模型 (bigram lexical affinities)

$$p(\text{words, tags, links}) = p(\text{words, tags}) \times (\text{link presences and absences} | \text{words, tags})$$
$$\approx \prod_{1 \leq i \leq n} p(\text{tword}(i) | \text{tword}(i+1), \text{tword}(i+2)) \times \prod_{1 \leq i, j \leq n} p(L_{ij} | \text{tword}(i), \text{tword}(j))$$

其中, tword 是词性(tag)和词形(word)的联合, L_{ij} 是取值0或1的二值函数, $L_{ij}=1$ 表示 i 和 j 具有依存关系, $L_{ij}=0$ 表示 i 和 j 不具有依存关系; n 是句子长度。

依存语法分析

一个标记序列(tags)由马尔柯夫(Markov)过程产生，观察每一对词(words)是否可以构成链接关系(link)的决策依赖于(tags, words)，即 link 对词汇是敏感的。最终生成words、tags 和 links 的联合概率模型。

依存语法分析

(2) 模型B. 选择偏好模型 (selectional preferences)

$$\begin{aligned} p(\text{words}, \text{tags}, \text{links}) &= p(\text{words}, \text{tags}) \times p(\text{preferences} | \text{words}, \text{tags}) \\ &\approx \prod_{1 \leq i \leq n} p(\text{tword}(i) | \text{tword}(i+1), \text{tword}(i+2)) \times \prod_{1 \leq i, j \leq n} p(L_{ij} | \text{preference}(i), \text{tword}(i)) \end{aligned}$$

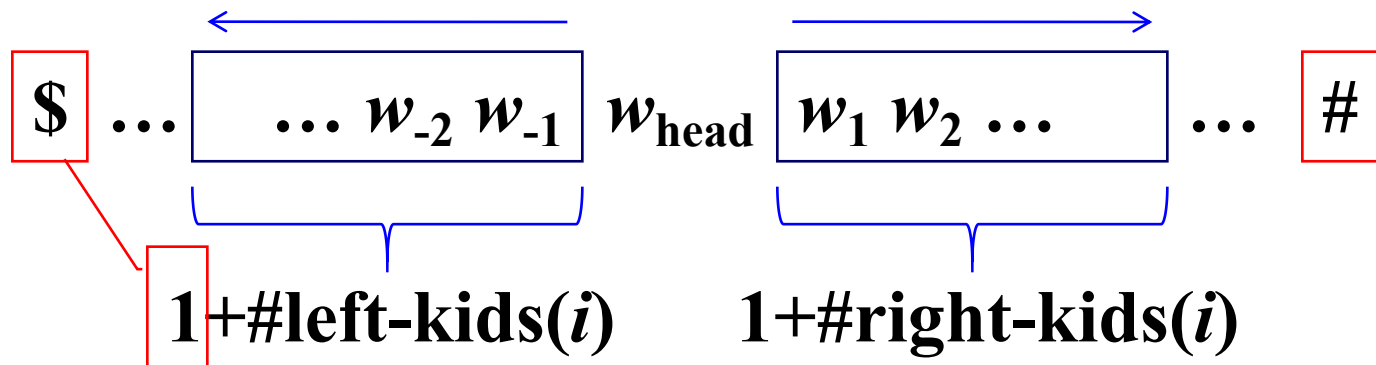
该模型加入了词的选择偏好($\text{preference}(i)$)信息，不再穷举所有连接(link)再根据约束进行剪裁，而是限制模型为每一个词只选择一个父结点。

依存语法分析

(3) 模型C. 递归生成模型 (recursive generation)

$$p(\text{words}, \text{tags}, \text{links}) = \prod_{1 \leq i \leq n} \left(\prod_{c = -(1 + \# \text{left-kids}(i)), c \neq 0}^{1 + \# \text{right-kids}(i)} p \left(\text{tword}(\text{kid}_c(i)) \middle| \text{tag} \left(\begin{array}{l} \text{kid}_{c-1}(i) \\ \text{or } \text{kid}_{c+1} \text{ if } c < 0 \end{array} \right), \text{tword}(i) \right) \right)$$

该模型中每个词的左子结点和右子结点分别由各自的马尔可夫模型顺次产生：左子结点的产生方向是自右向左，右子结点的产生方向是自左向右的。每一个子结点的生成建立在支配词和它前一个子结点上，是自顶向下的递归生成式模型。



依存语法分析

✧ 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。

依存语法分析

➤ 判别式的分析方法 (discriminative parsing)

✧ 基本思想: 采用条件概率模型 $Score(x | y, \theta)$, 使目标函数 $\prod_{i=1}^n Score(x_i | y_i; \theta)$ 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:

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

依存语法分析

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。 $f(\bullet)$ 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:

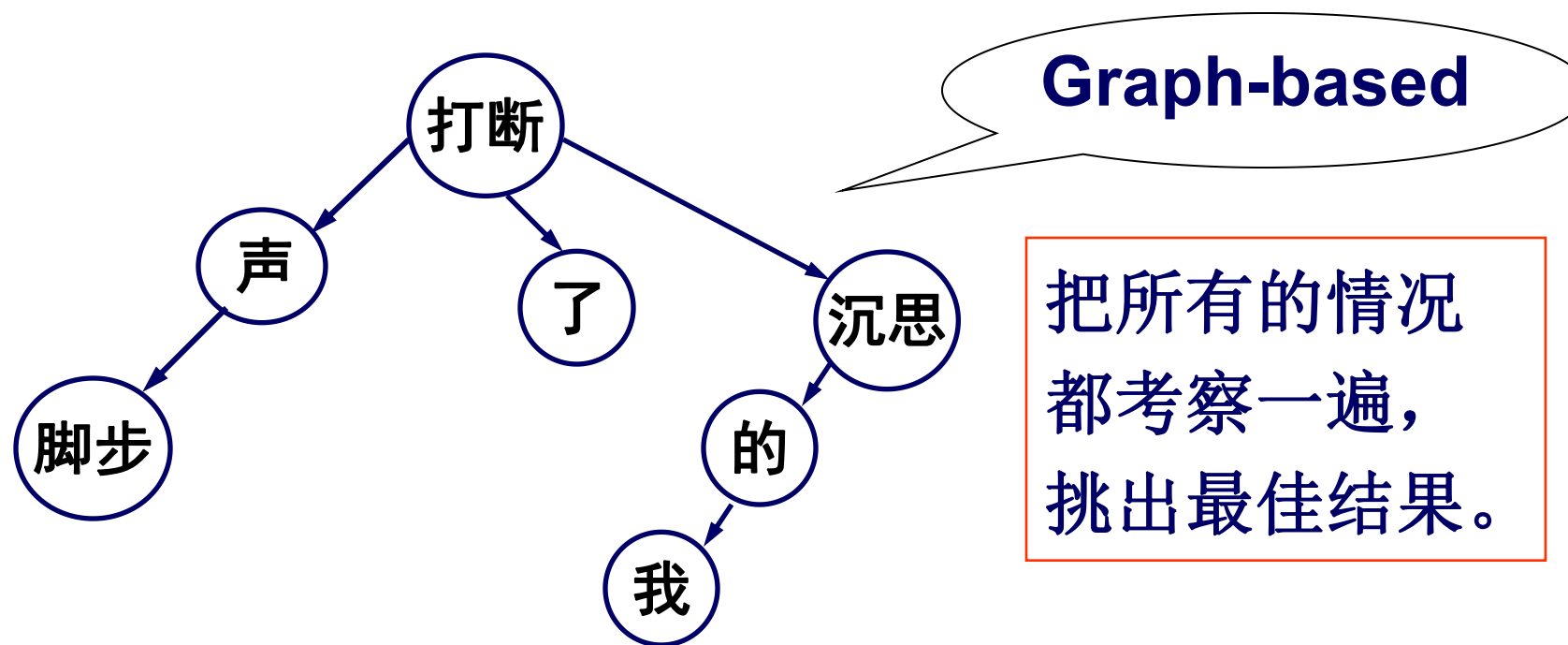
$$f(i, j) = \begin{cases} 1 & \text{如果 } x_i = \text{'打'} \text{ and } x_j = \text{'球'} \\ 0 & \text{其他} \end{cases}$$

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

依存语法分析

该方法基本思想就是，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。

依存语法分析



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

依存语法分析

✧ 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。

依存语法分析

- 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。

依存语法分析

(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、未处理结点序列和依存弧集合。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。

依存语法分析

初始: $[]_W$ 终止: $[\text{ROOT}]_W$

$$\begin{array}{l} \text{Right} \quad \frac{\dots [w_i, w_{i+1}]_W \dots}{\begin{array}{c} \dots [w_{i+1}]_W \dots \quad A \cup \{w_i \leftarrow w_{i+1}\} \\ | \\ w_i \end{array}} \end{array}$$

$$\begin{array}{l} \text{Left} \quad \frac{\dots [w_i, w_{i+1}]_W \dots}{\begin{array}{c} \dots [w_i]_W \dots \quad A \cup \{w_i \rightarrow w_{i+1}\} \\ | \\ w_{i+1} \end{array}} \end{array}$$

$$\begin{array}{l} \text{Shift} \quad \frac{\dots [w_i, w_{i+1}]_W \dots}{\dots [w_{i+1}, w_{i+2}]_W \dots} \end{array}$$

依存语法分析

(2) Arc-eager 分析算法 — 4种分析动作(Actions):

初始: $(\text{nil}, I, \emptyset)$ 终止: (S, nil, A)

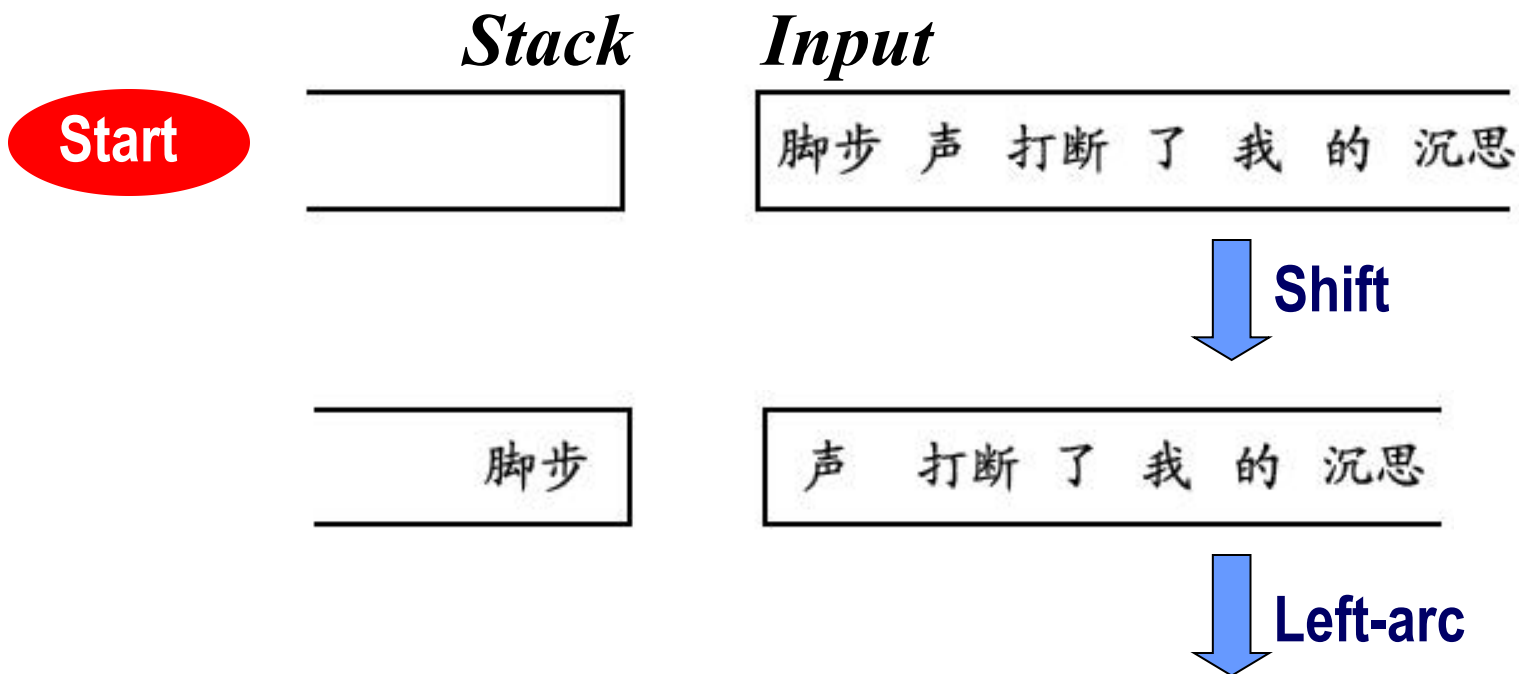
Left-Arc_l	$\frac{[\dots, w_i]_S \quad [w_j, \dots]_I \quad \neg \exists w_k \rightarrow w_i \in A \quad \text{条件}}{[\dots]_S \quad [w_j, \dots]_I \quad A \cup \{w_i \xleftarrow{l} w_j\}, \text{pop}(w_i)}$
Right-Arc_l	$\frac{[\dots, w_i]_S \quad [w_j, \dots]_I \quad \neg \exists w_k \rightarrow w_j \in A}{[\dots w_i, w_j]_S \quad [\dots]_I \quad A \cup \{w_i \xrightarrow{l} w_j\}, \text{push}(w_j)}$
Reduce	$\frac{[\dots w_i]_S \quad [\dots]_I \quad \exists w_k \rightarrow w_i \in A}{[\dots]_S \quad [\dots]_I \quad \text{pop}(w_i)}$
Shift	$\frac{[\dots]_S \quad [w_i, \dots]_I}{[\dots w_i]_S \quad [\dots]_I \quad \text{push}(w_i)}$

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

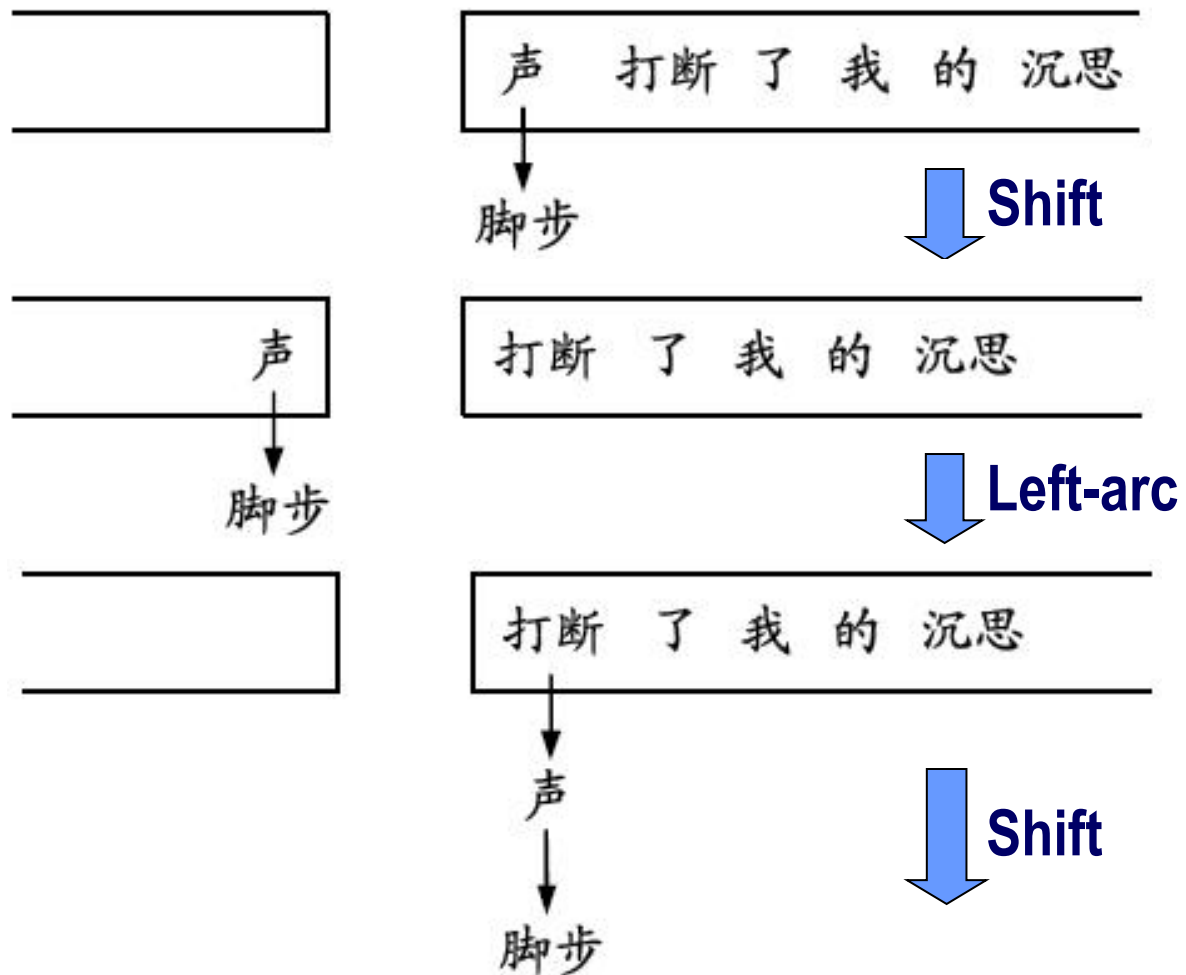
依存语法分析

✧举例：分析如下句子：

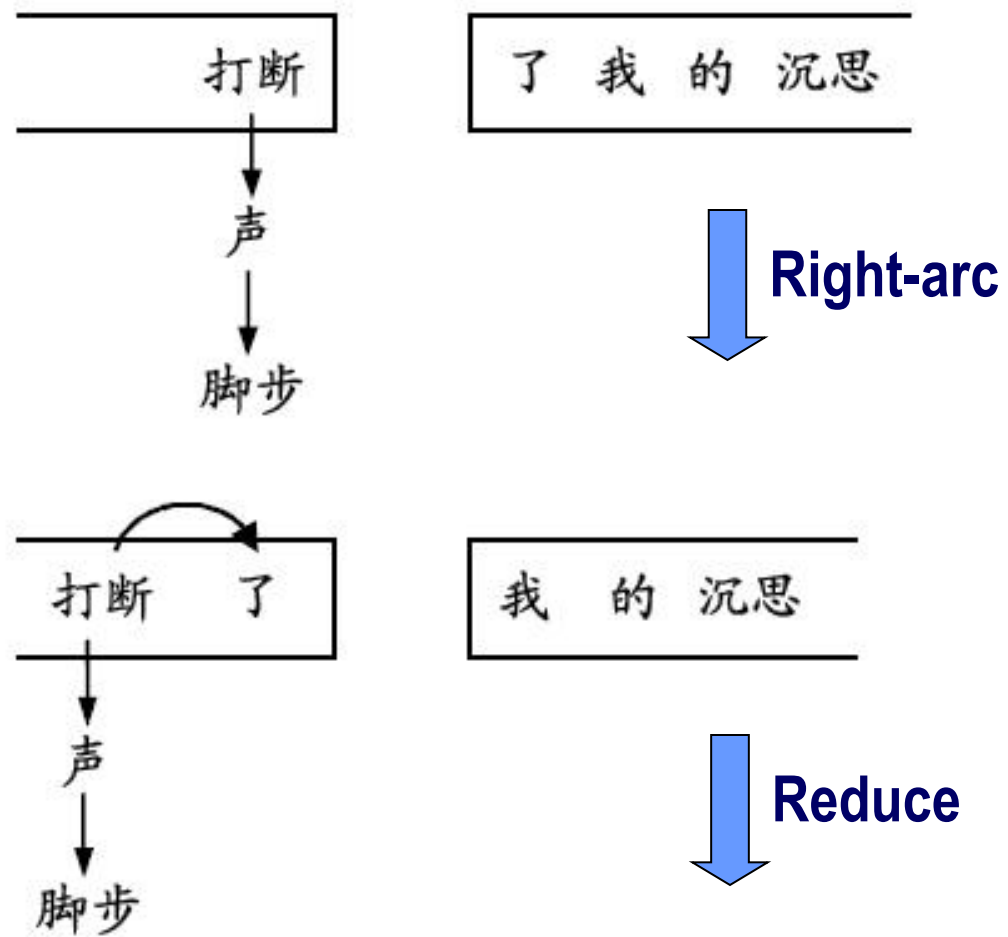
脚步 声 打断 了 我 的 沉思



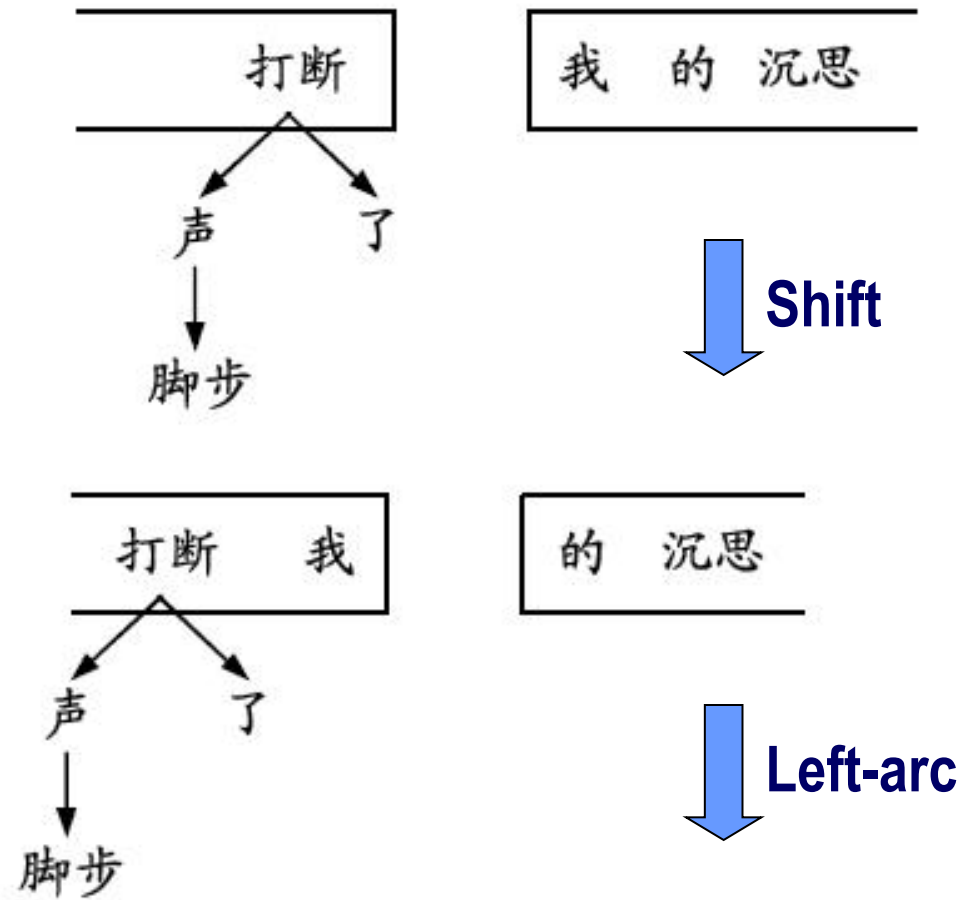
依存语法分析



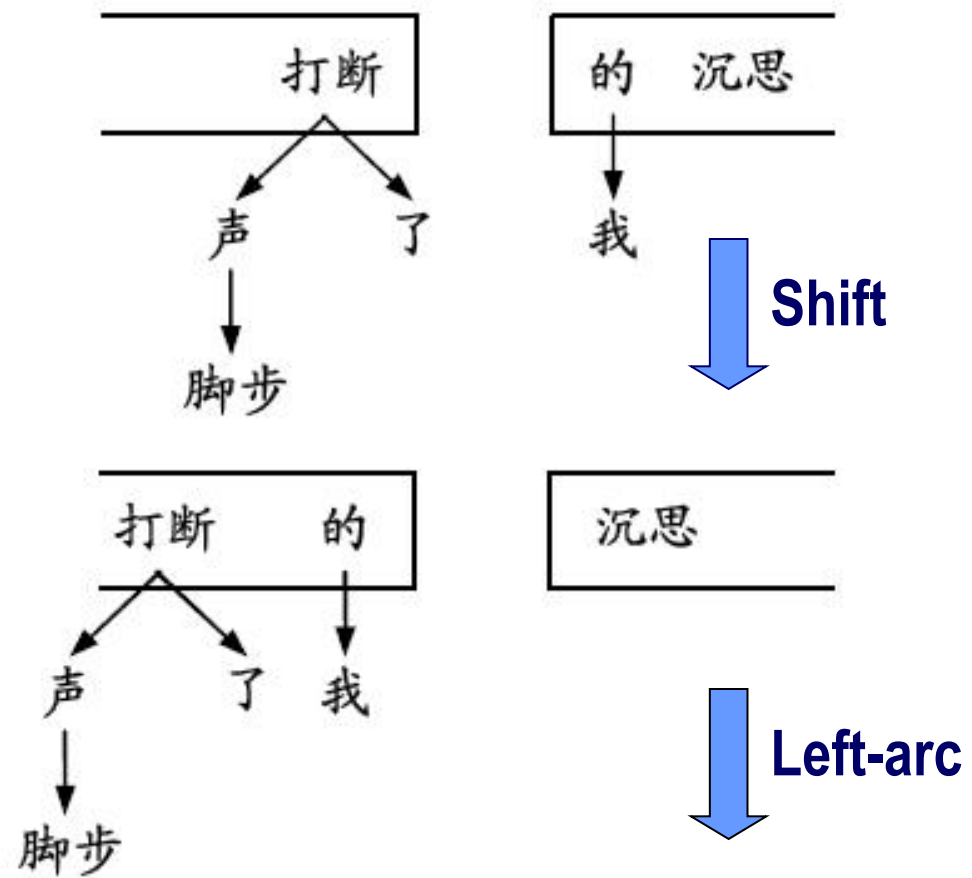
依存语法分析



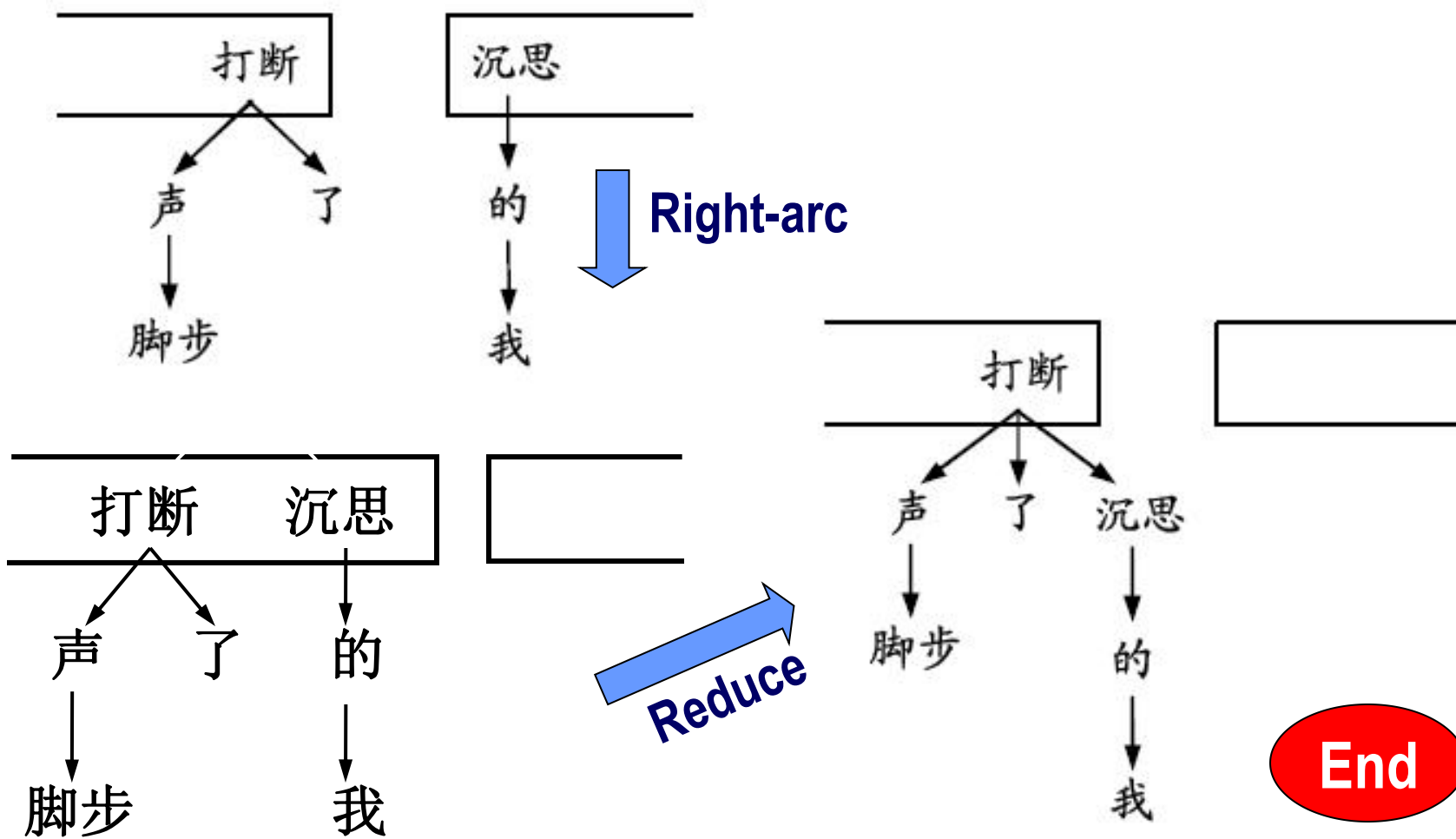
依存语法分析



依存语法分析



依存语法分析

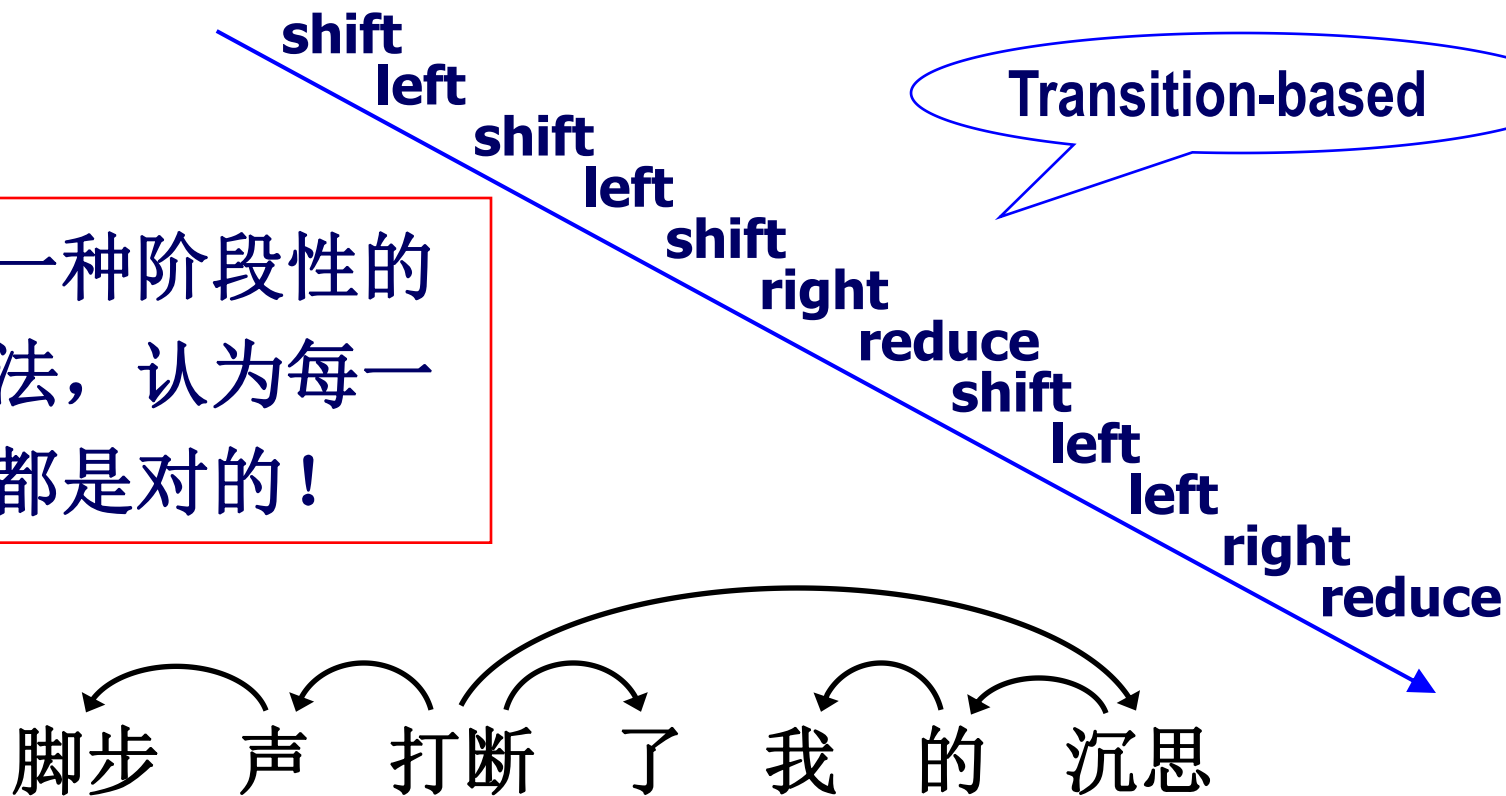


依存语法分析

脚步 声 打断 了 我 的 沉思

是一种阶段性的方法，认为每一步都是对的！

Transition-based



依存语法分析

✧ 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。

依存语法分析

➤ 基于约束满足的分析方法(Constraint Satisfaction parsing)

- ✧ 基本思想：将依存句法分析过程看作可以用约束满足问题(Constraint satisfaction problem, CSP)来描述的有限构造问题(finite configuration problem)。它不像上下文无关文法那样探索性地生成，而是根据已规定好的约束进行剪裁，把不符合约束的分析去掉，直到留下一棵合法的依存树。
- ✧ 存在的问题：可能不存在能满足所有约束的分析树，也可能有多个树满足所有约束，无法消歧。

依存语法分析

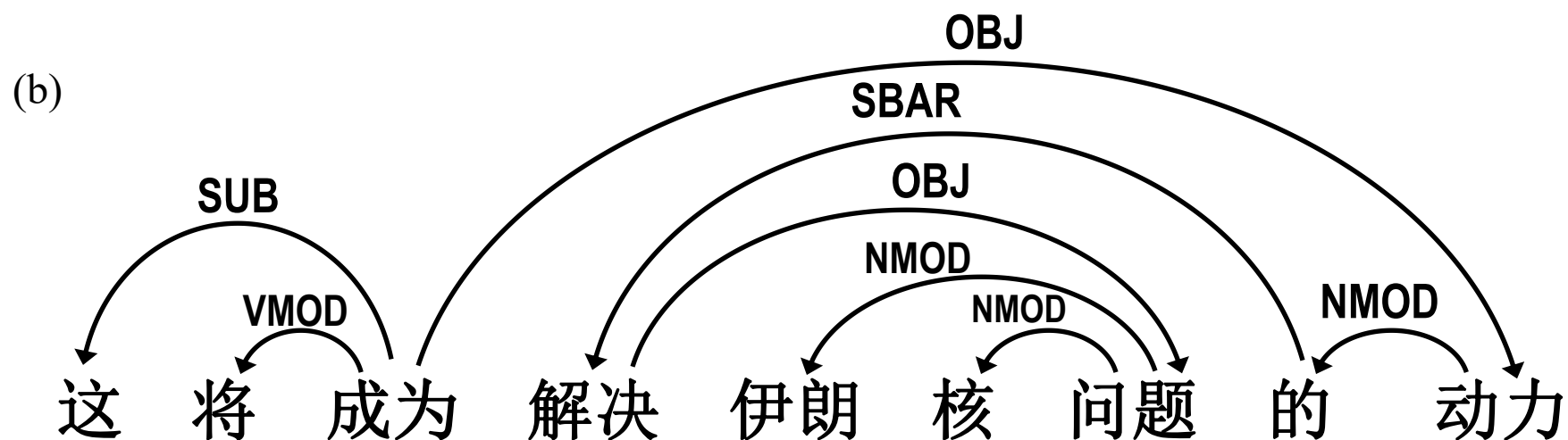
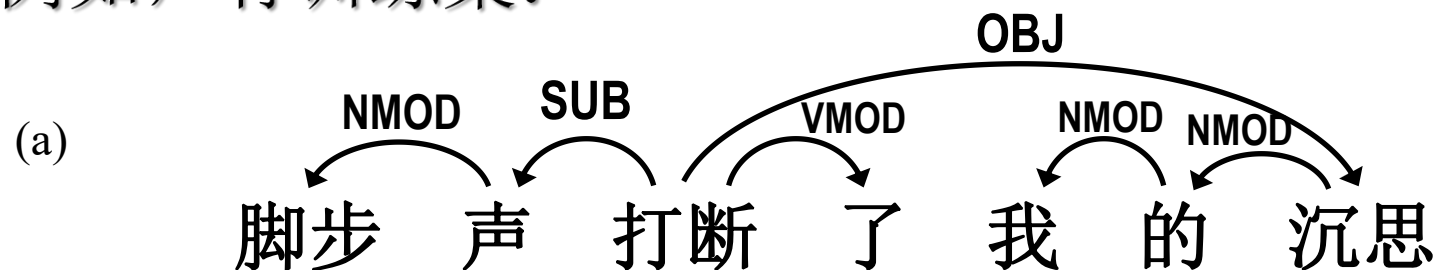
➤ 实现一个依存句法分析器

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。

依存语法分析

例如，有训练集：

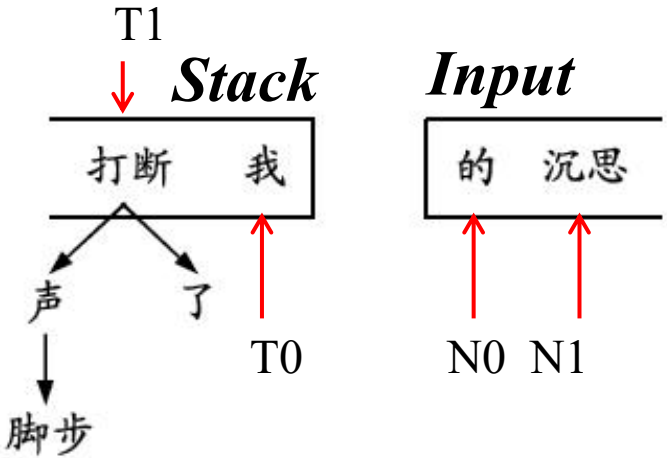


依存语法分析

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: right_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。



依存语法分析

根据前面的图(b)得到如下训练实例：

动作 特征序列

Shif 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

.....

依存语法分析

根据前面的图(b)得到如下训练实例

动作 特征序列

Shif 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:VMOD
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）。

依存句法分析器评价指标

- 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。

依存句法分析器评价指标

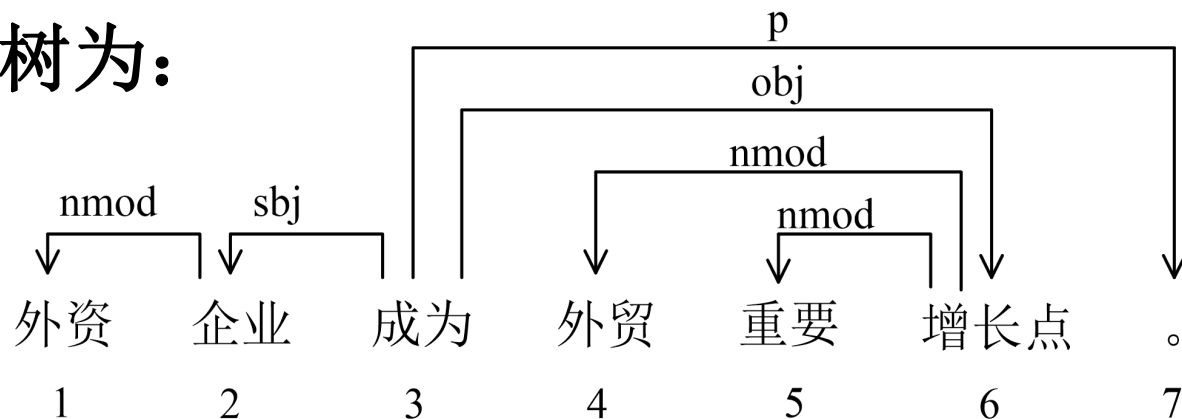
- 根正确率(root accuracy, RA): 有两种定义方式:
 - (1)正确根结点的个数与句子个数的比值;
 - (2)另一种是所有句子中找到正确根结点的句子所占的百分比。

对单根结点语言或句子来说,二者是等价的。

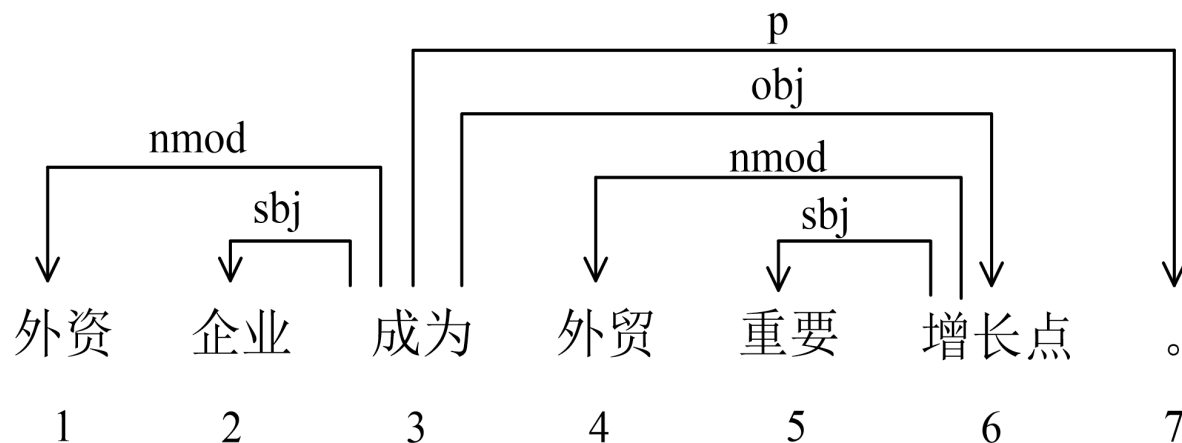
- 完全匹配率(complete match, CM): 所有句子中无标记依存结构完全正确的句子所占的百分比。

依存句法分析器评价指标

✧例如，答案依存树为：



✧系统输出的依存分析树为：



依存句法分析器评价指标

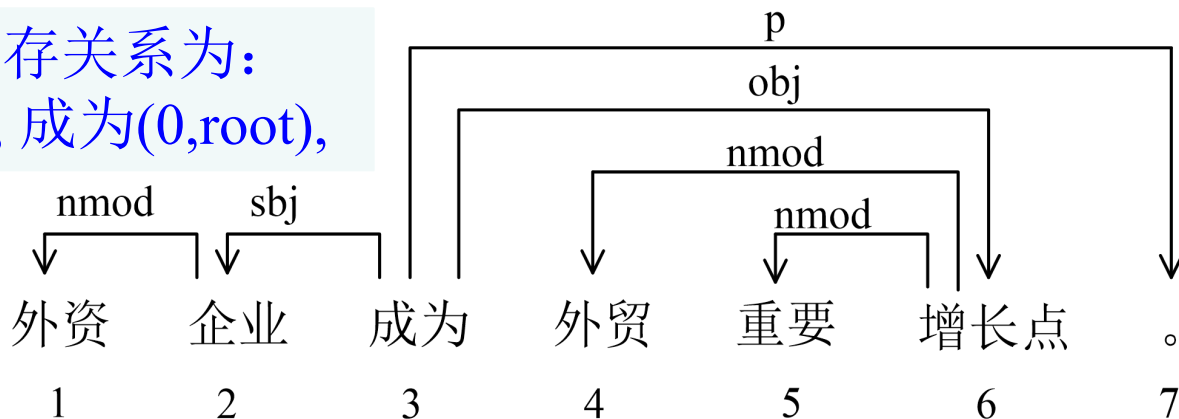
每个词对应的支配词及依存关系为：

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

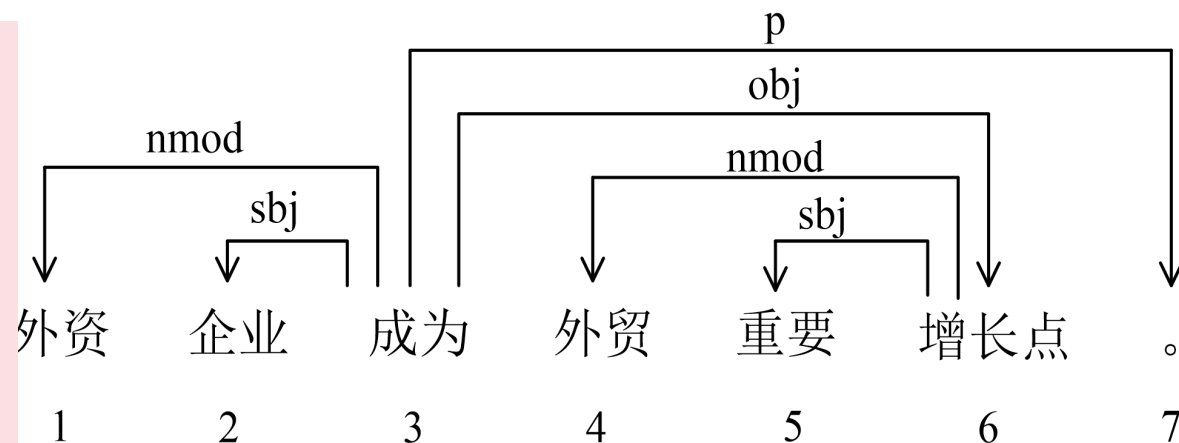
存关系为：外资(3,nmod),

企业(3,sbj), 成为(0,root),

外贸(6,nmod),

重要(6,sbj),

增长点(3,obj), 。(3,p)

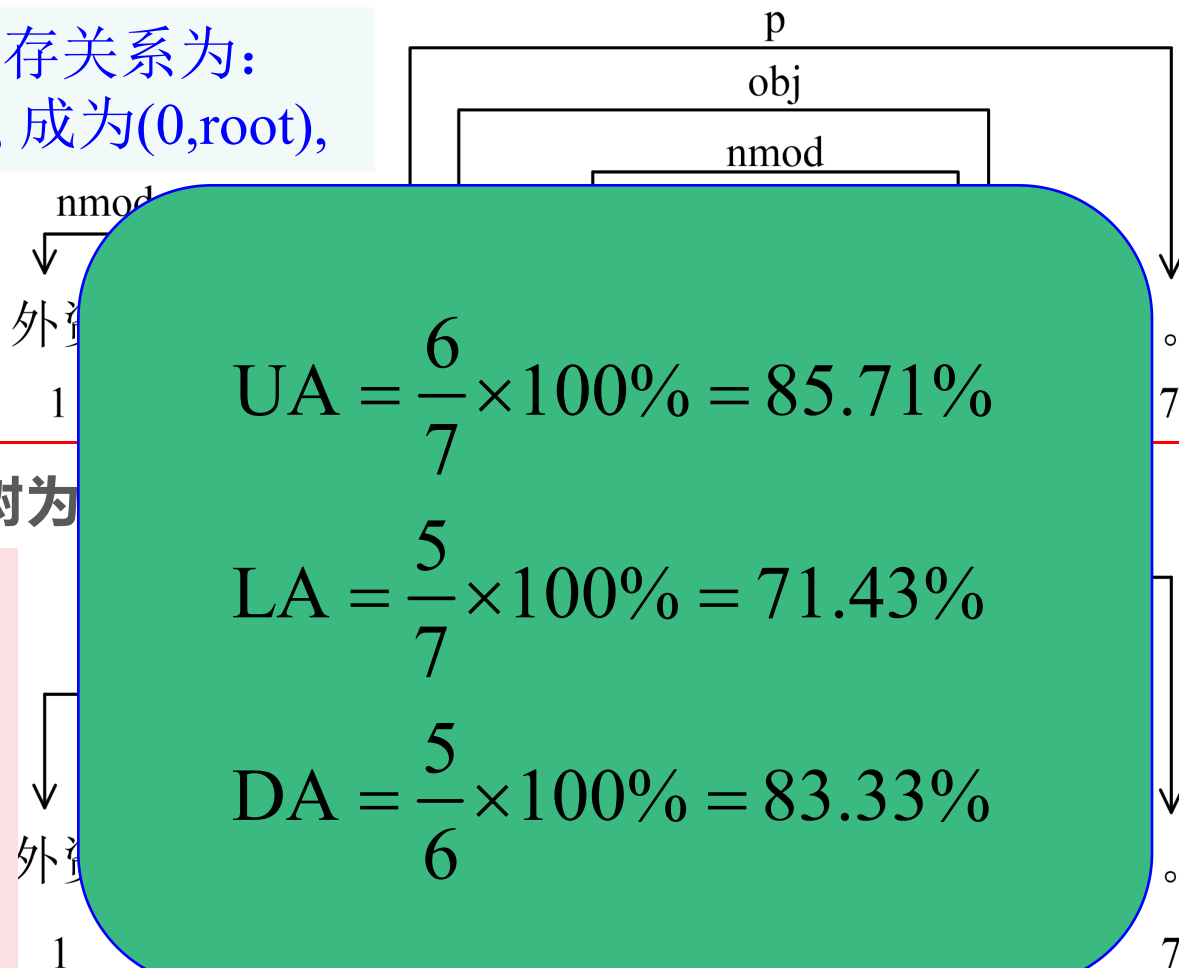


依存句法分析器评价指标

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)



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)

短语结构与依存结构的关系

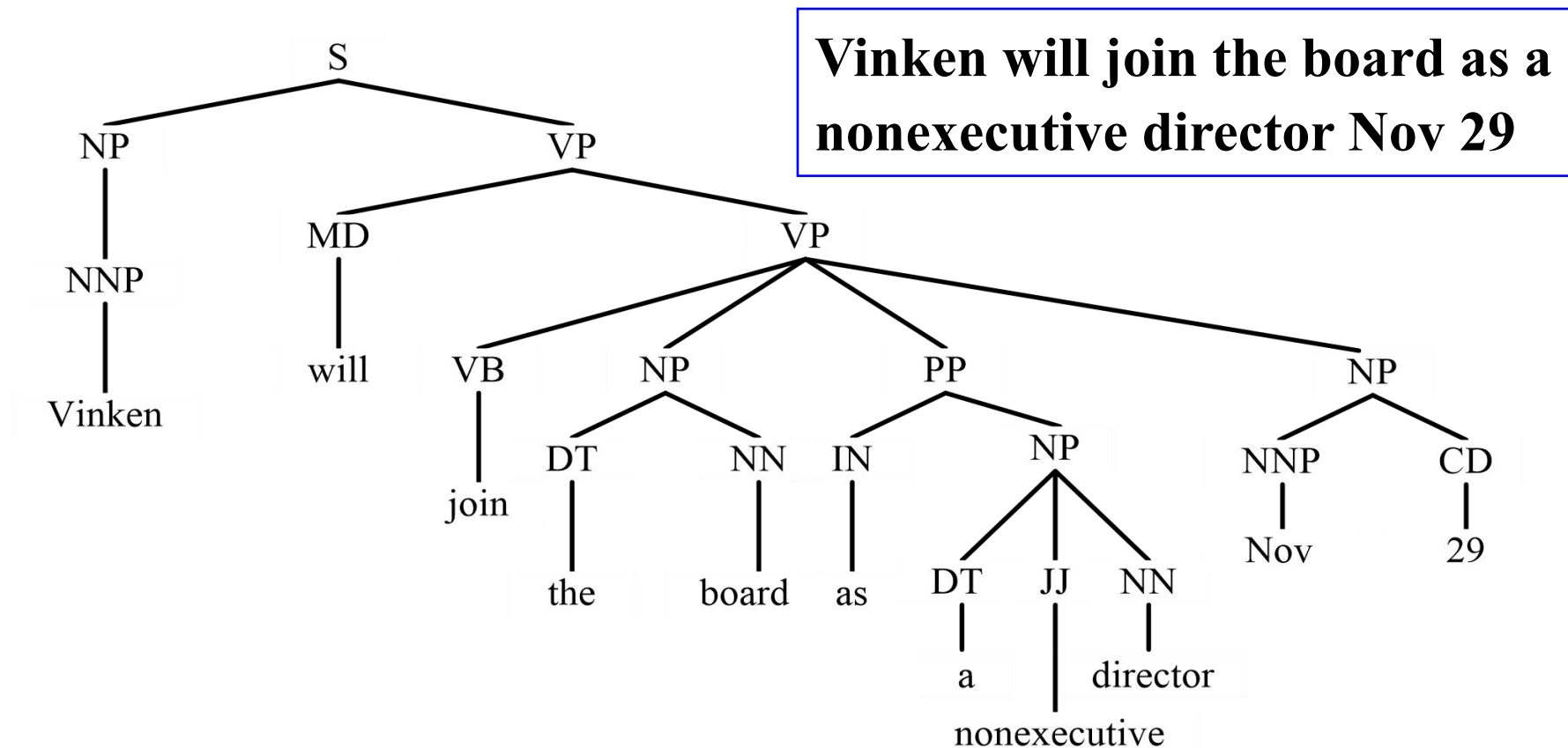
◆ 短语结构可转换为依存结构

➤ 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。

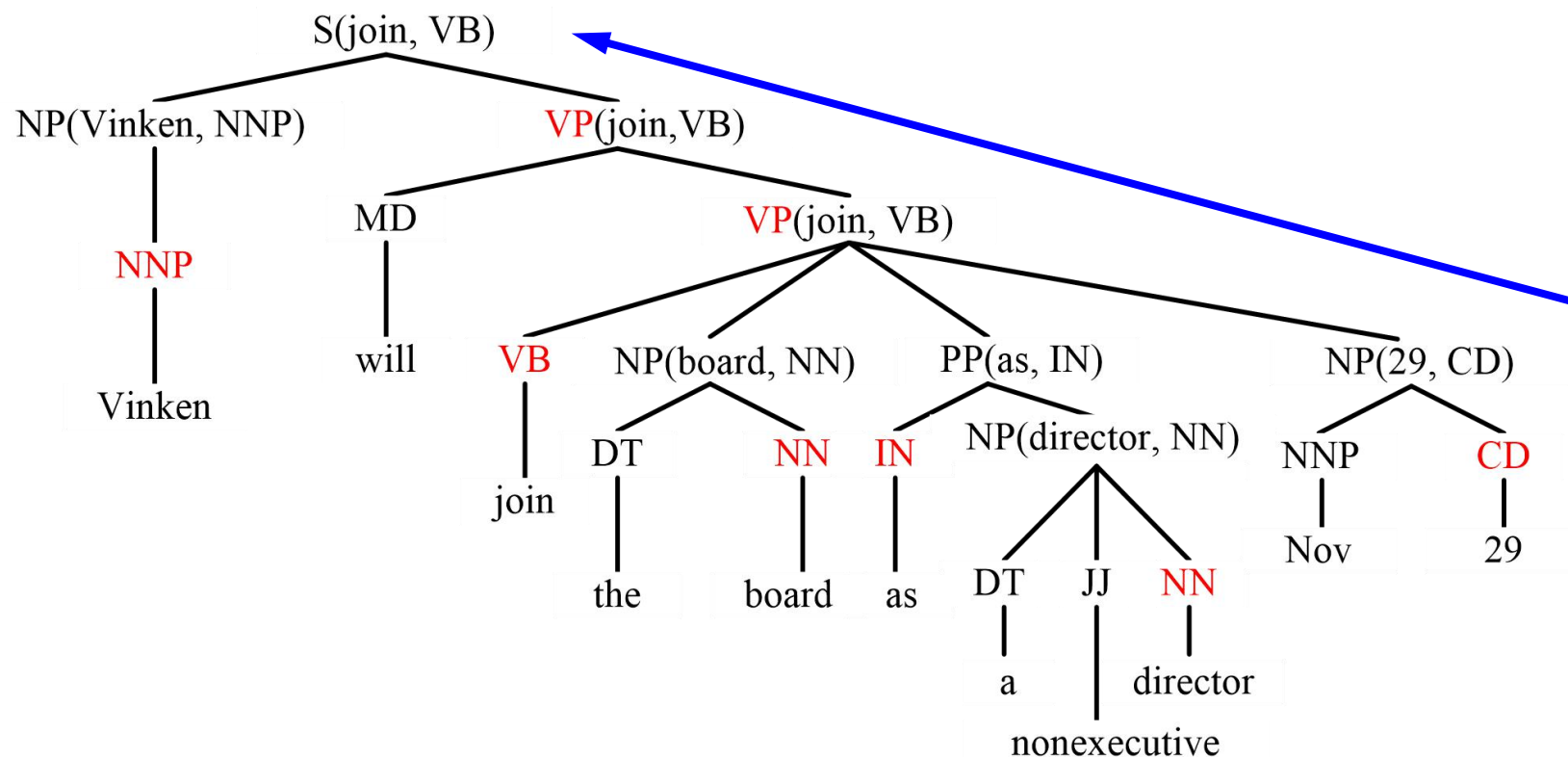
短语结构与依存结构的关系

例如：给定如下短语结构树



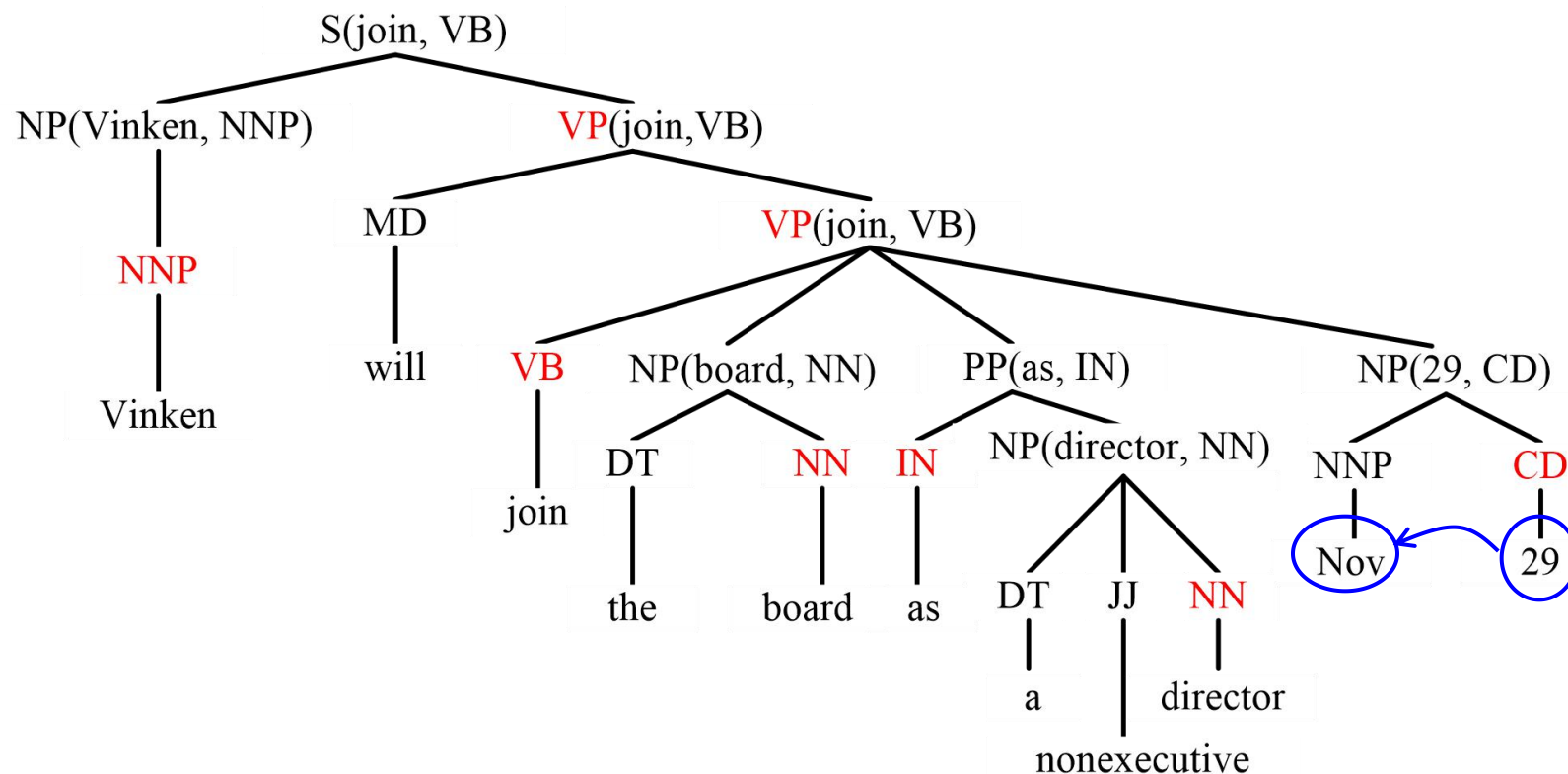
短语结构与依存结构的关系

- 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)



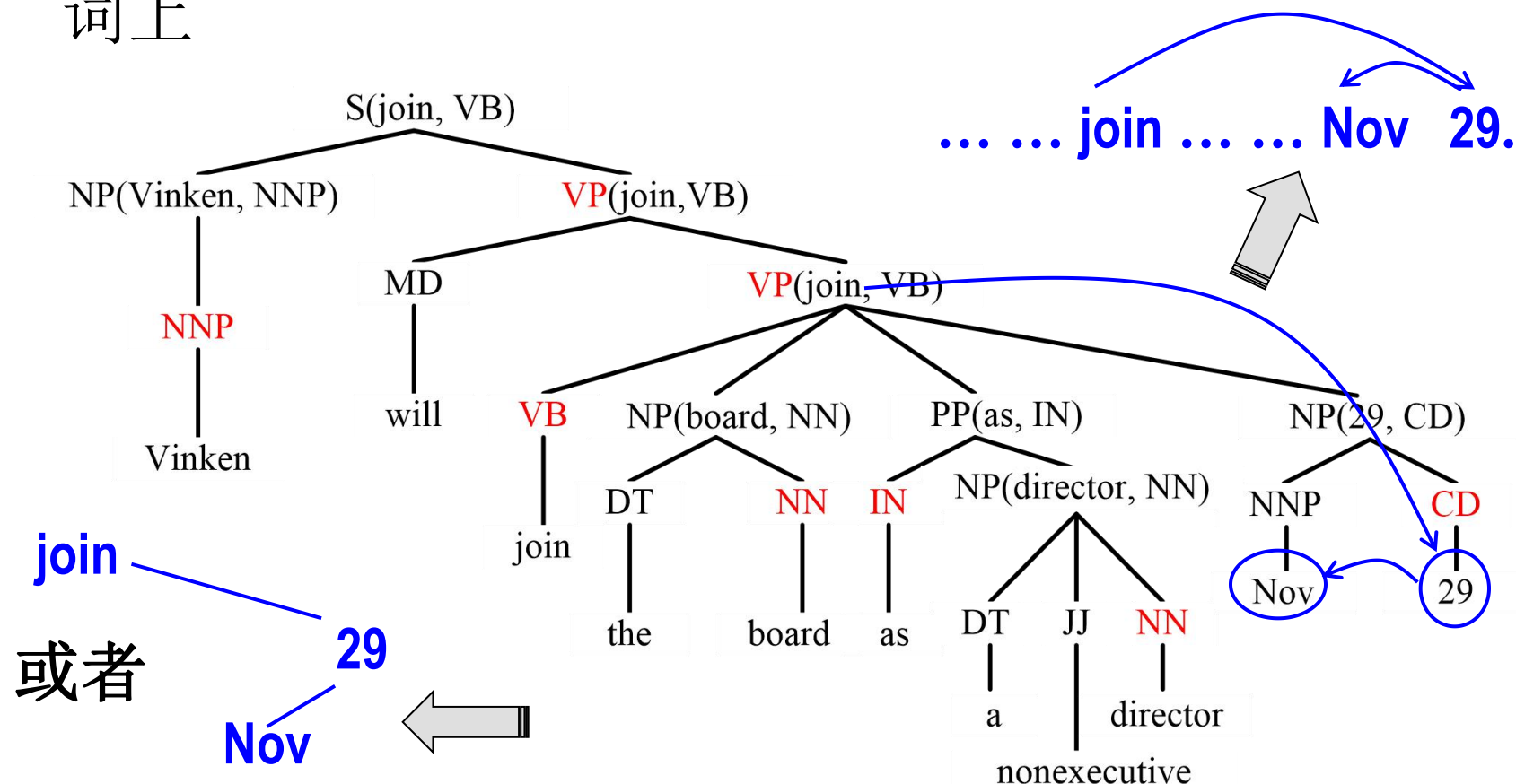
短语结构与依存结构的关系

- 将非中心子节点的中心词依存到中心子节点的中心词上



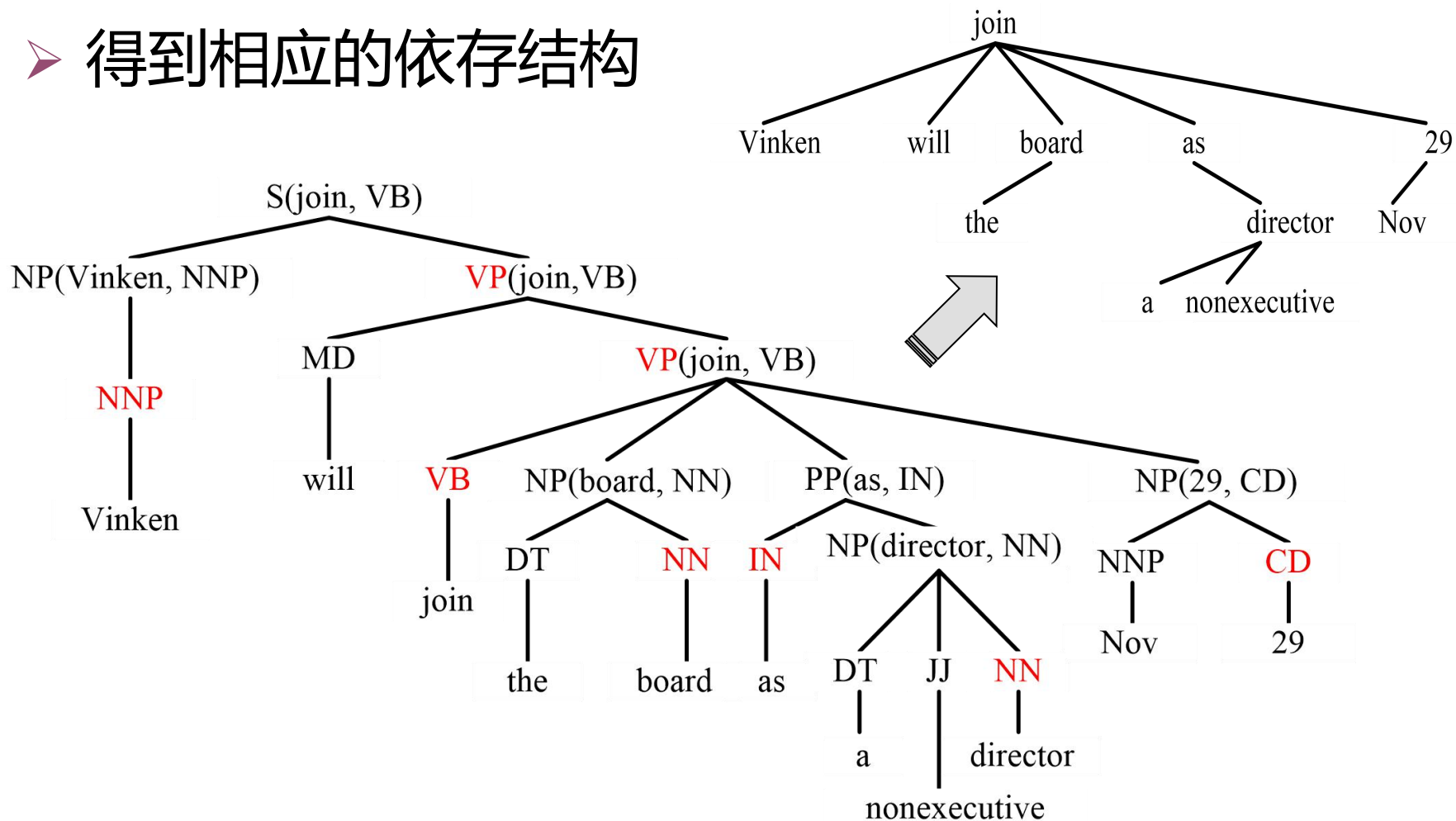
短语结构与依存结构的关系

- 将非中心子节点的中心词依存到中心子节点的中心词上



短语结构与依存结构的关系

➤ 得到相应的依存结构



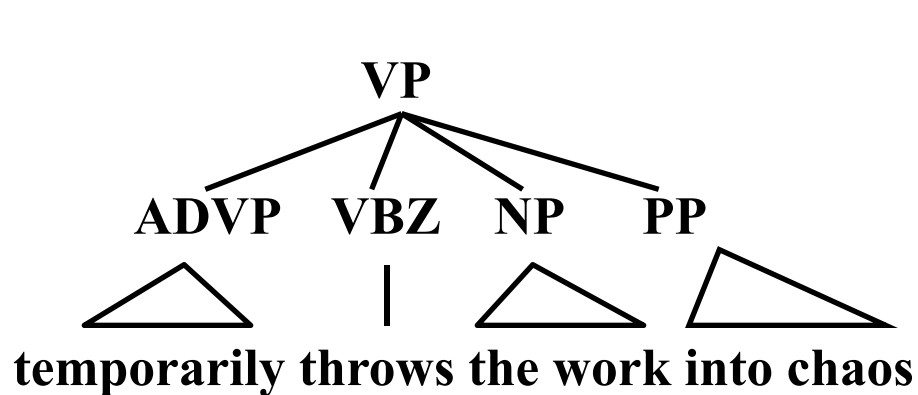
汉英句法结构特点对比

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。

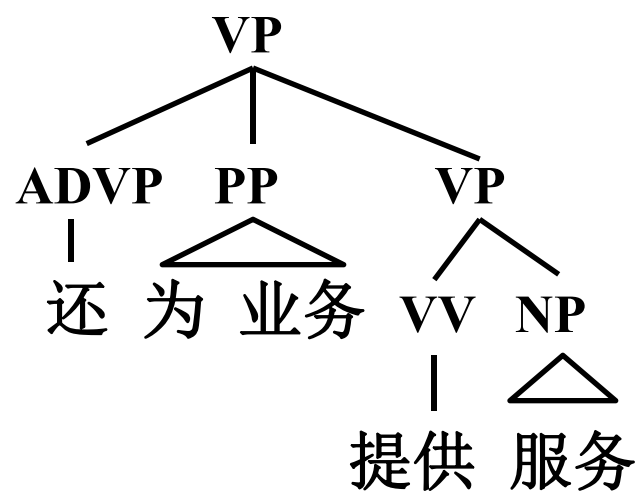
- (1) 汉语比英语更少地使用功能词(function words)，且没有形态变化：汉语中不使用限定词(“这、这个、那个”等)的名词普遍存在，复数标记(“们”等)有限并且很少出现。

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)

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。

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。

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。

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。”

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]。

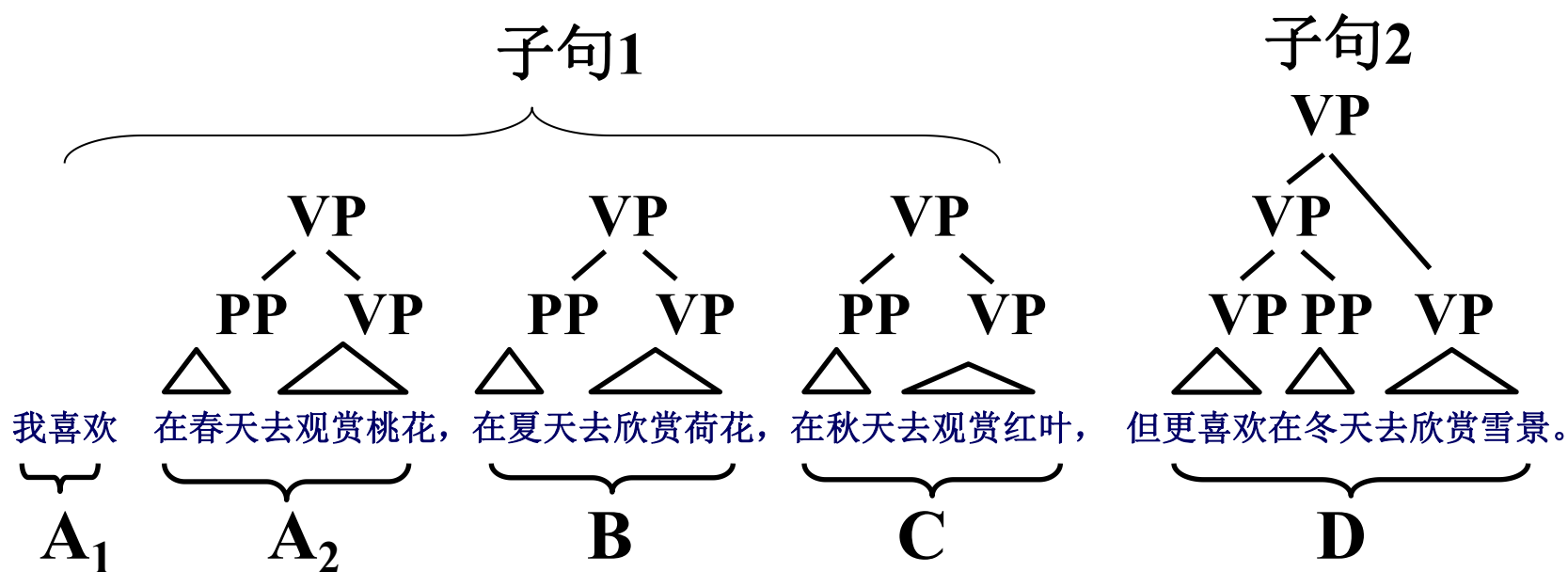
汉英句法结构特点对比

◆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。

汉英句法结构特点对比

例句1：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。



汉英句法结构特点对比

例句2：夫人穿着很得体，举止优雅，左臂上挂着一个暗黄色的皮包，右手领着一只白色的小狗，据说是京巴。

译文：Lady wearing a very decent, elegant manners, a dark yellow bag hanging on his left arm, the right hand took a white dog, is said to be a Pekingese. (百度翻译, 2016.4.19)

Lady wearing a very decent, elegant, wearing a dark yellow bag on his left arm, his right hand led a white dog, is said to Beijing bar. (Google 在线翻译, 2016.4.19)

- 译文：The lady is well dressed and elegant. She has a dark yellow leather bag on her left arm and a white dog on her right hand. It is said that it is Jingba. (百度翻译 2020.09)
- 译文：The lady is well dressed and elegant. She has a dark yellow leather bag on her left arm and a white dog in her right hand. It is said that she is Jingba. (百度翻译 2021.02)

例句2：夫人穿着很得体，举止优雅，左臂上挂着一个暗黄色的皮包。右手领着一只白色的小狗，据说是京巴。

- 译文：The lady is well dressed and elegant, with a dark yellow leather bag hanging on her left arm. It is said that the right hand of Jingba is a white dog. (百度翻译 2021.03)