

第9章 句法分析

(2/2)

宗成庆

中国科学院自动化研究所

cqzong@nlpr.ia.ac.cn


1. 概述

◆句法分析任务(syntactic parsing)

句法分析任务的目标就是识别句子的结构关系。在自然语言处理中，通常有两种句法分析任务：

- 短语结构分析(constituent phrase parsing)
 - 完全句法分析
 - 局部句法分析
- 依存关系分析(dependency parsing)

本章内容

- 
1. 概述
 2. 依存关系分析方法
 3. 分析结果评价
 4. 短语结构与依存关系
 5. 英汉句法结构特点对比
 6. 习题

1. 概述

◆依存语法理论

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 (Lucien Tesnière, 其姓氏也被译作: 特思尼耶尔或特尼耶尔)(1893.5-1954.12)。他的主要成就是提出“结构句法”理论, 后称“依存语法”或“从属语法”。为了提出一种普适的语法理论, 他作了大量的语言对比研究, 从1939年起他开始撰写巨著《结构句法基础》(Éléments de Syntaxe Structurale), 边写边改, 历时十余载, 直到1950年才完成。1954年泰尼埃去世之后, 他的朋友们整理了他的遗稿, 于1959年出了《结构句法基础》的初版, 1965年出了第二版。

冯志伟: “泰尼埃与依存语法”,
见《现代语文》2014年第11期

1. 概述

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。

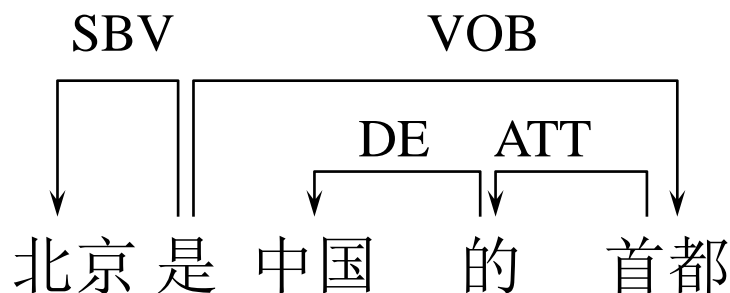
1. 概述

Tesnière 在《结构句法基础》中将化学中“价”的概念引入到依存语法中。“价”亦称“配价”或“向”(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。例如，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。

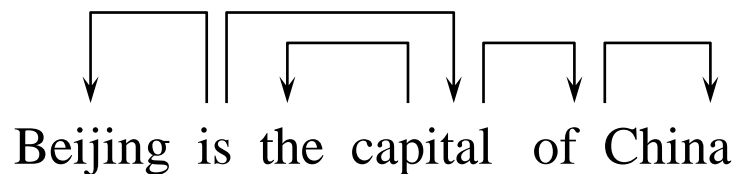
在依存语法理论中，“依存”就是指词与词之间支配与被支配的关系，这种关系**不是对等的，而是有方向的。处于支配地位的成分称为支配者(governor, regent, head)，而处于被支配地位的成分称为从属者(modifier, subordinate, dependency)。**

1. 概述

◆ 依存关系表示



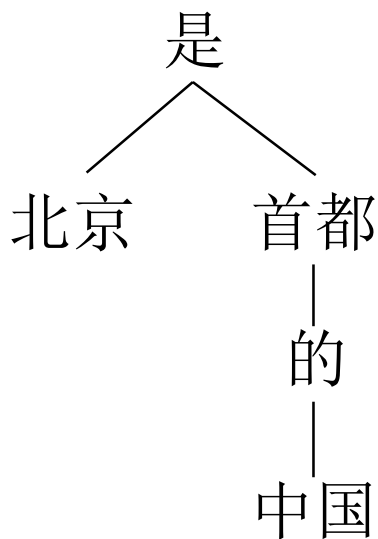
(a) 有向图



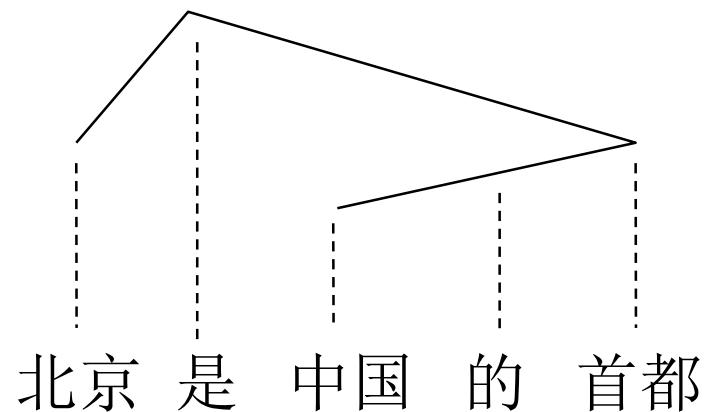
(b) 有向图

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。

1. 概述



(c) 依存树



(d) 依存投射树

图(c)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(d)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。

1. 概述

◆依存语法的四条公理

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。

1. 概述

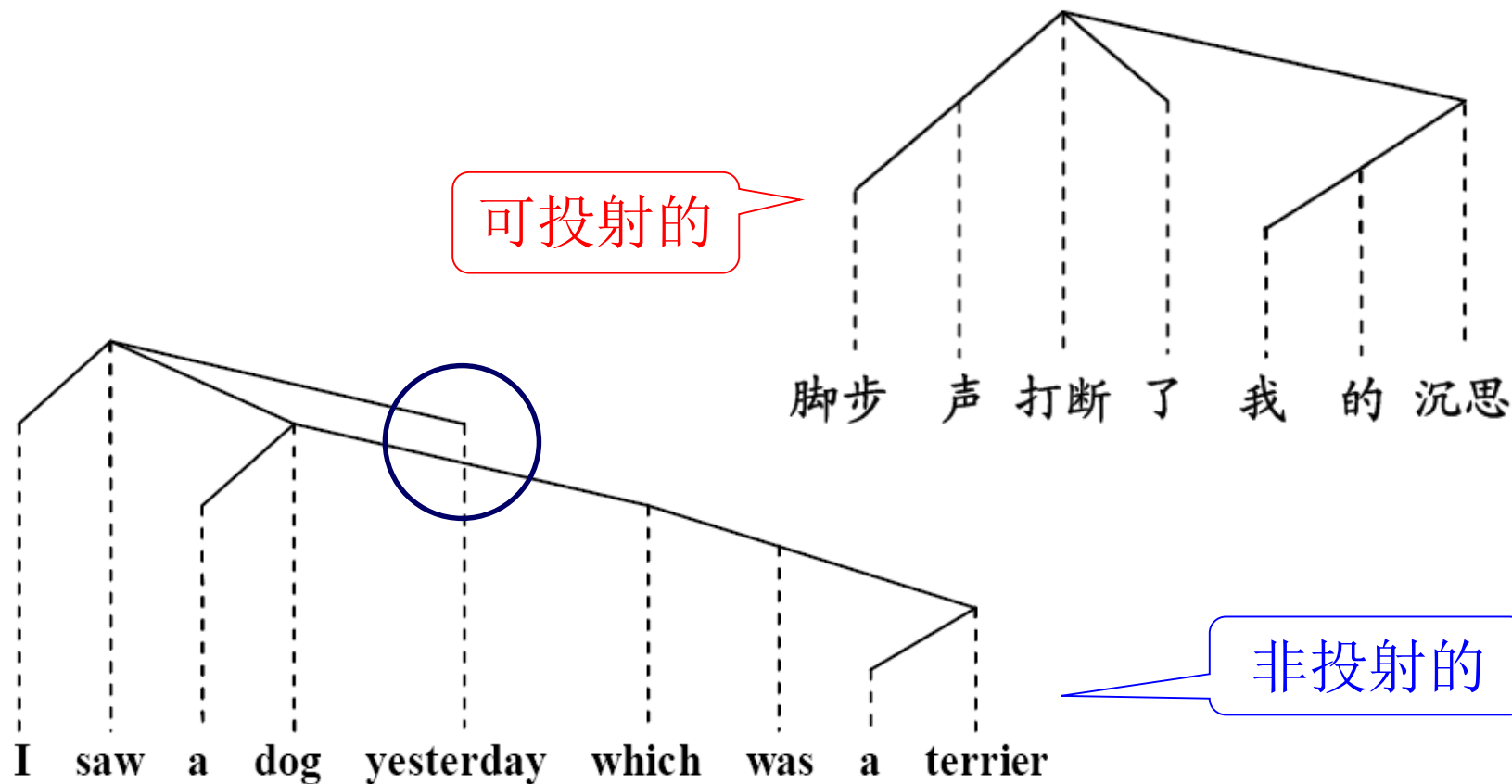
这4条公理相当于对依存图和依存树的形式约束为：

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

由此保证了句子的依存分析结果是一棵有“根(root)”的树结构。

1. 概述

◆ 投射投射(projective)与非投射(no-projective)



1. 概述

◆依存语法的优势

- 简单，直接分析词语之间的依存关系，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；
- 与短语结构语法相比，其形式化程度较浅，对句法结构的表述更为灵活。

本章内容

1. 概述

 2. 依存关系分析方法

3. 分析结果评价

4. 短语结构与依存关系

5. 英汉句法结构特点对比

6. 习题

2. 依存关系分析方法

◆方法归纳

输入：句子；

输出：依存结构，通常采用有向图描述。

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几类：

- 生成式分析方法(generative parsing)
- 判别式分析方法(discriminative parsing)
- 决策式(确定性)分析方法(deterministic parsing)
- 基于约束满足的分析方法(constraint satisfaction parsing)
- 神经网络方法与决策方法的结合

2. 依存关系分析方法

◆生成式分析方法(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。这种搜索整个概率空间, 得到整个句子的依存分析结果。
- ✧ 优点: 如果有大规模高质量标注样本, 可获得较好的性能。
- ✧ 弱点: (1) 采用联合概率模型, 在进行概率乘积分解时常常需要做各种假设以简化, 不易加入语言特征; (2) 由于采用全局搜索, 算法的复杂度较高, 一般为 $O(n^3)$ 或 $O(n^5)$ 。

2. 依存关系分析方法

◆ 判别式分析方法(discriminative parsing)

- ✧ 基本思想：采用条件概率模型 $Score(x|y, \theta)$ ，使目标函数最大的 θ 作为模型的参数。
- ✧ 例如：最大生成树模型(maximum spanning trees, MST)

在点和边组成的生成树(spanning tree)中寻找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。

2. 依存关系分析方法

定义整棵句法树的打分是树中各条边打分的加权和：

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

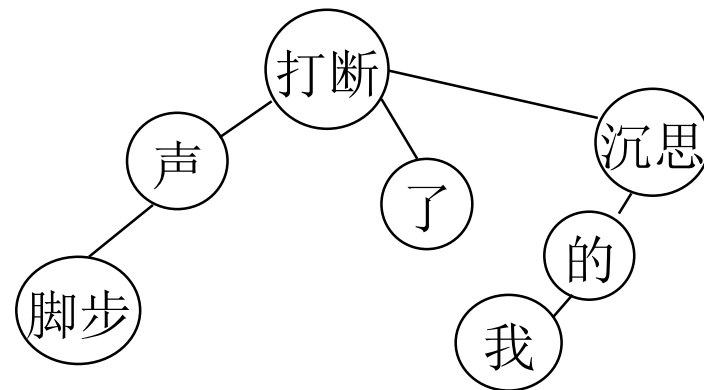
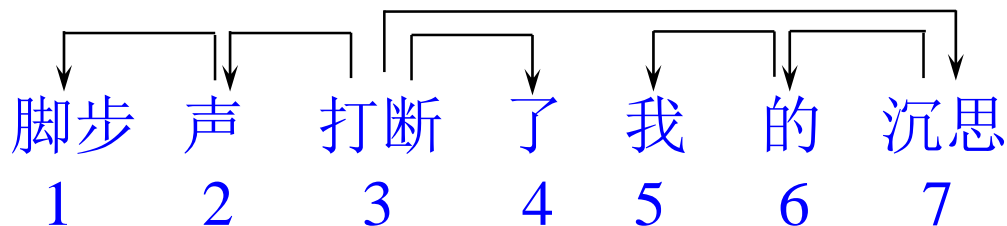
其中， s 是打分函数， y 是句子 x 的一棵依存分析树， (i, j) 是 y 中的一对结点。 $\mathbf{f}(\bullet)$ 是取值为 1 或 0 的高维二元特征函数向量，表示结点 i 和 j 之间的依存关系。如“我的”词对中，“的”字支配“我”，则 $f(i, j)=1$ ，否则 $f(i, j)=0$ 。即：

$$f(i, j) = \begin{cases} 1 & \text{如果 } y_i = \text{'的'} \text{ and } y_j = \text{'我'} \\ 0 & \text{其他} \end{cases}$$

\mathbf{w} 是特征 $f(i, j)$ 的权值向量，可以由样本训练得到。

2. 依存关系分析方法

例如：



$(1, 2) = 0, (1, 3) = 0, \dots$

$(2, 1) = 1, (2, 3) = 0, \dots$

$(3, 1) = 0, (3, 2) = 1, (3, 4) = 1 \dots (3, 7) = 1$

\dots

Graph-based method

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

2. 依存关系分析方法

✧方法评价

➤优点:

- 判别式模型避开了联合概率模型所要求的独立性假设;
- 有较好的可计算性, 使很多机器学习方法得以应用, 并可处理非投射现象;
- 分析准确率较高。

➤弱点:

- 整句内全局搜索, 不易使用动态特征;
- 由于是全局搜索, 算法复杂度较高。

2. 依存关系分析方法

◆ 决策式(确定性)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，每次都要根据当前状态做出决策，如判断当前读入的词是否与前一个词发生依存关系。一旦做出决策，之后不再改变。所谓的决策就是“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一系列分析动作(action)。

2. 依存关系分析方法

◇ 移进-归约算法(shift-reduce algorithm)

J. Nivre 等人2003年提出了一种自左向右、自底向上的分析算法，基本思路如下：

当前分析状态的格局(configuration)是一个三元组：(S, I, A), S 表示栈顶词(Stack)、I表示未处理序列中的当前词(Input), A 表示依存弧集合(Arcs)。4种操作(actions): Left-Arc (依存弧向左指)、Right-Arc (依存弧向右指)、Reduce (归约) 和 Shift (移进)。用一个stack栈和一个queue输入序列来分别储存已经被处理的词和未被分析的句子中的词。

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

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

2. 依存关系分析方法

◇ Arc-eager 分析算法:

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

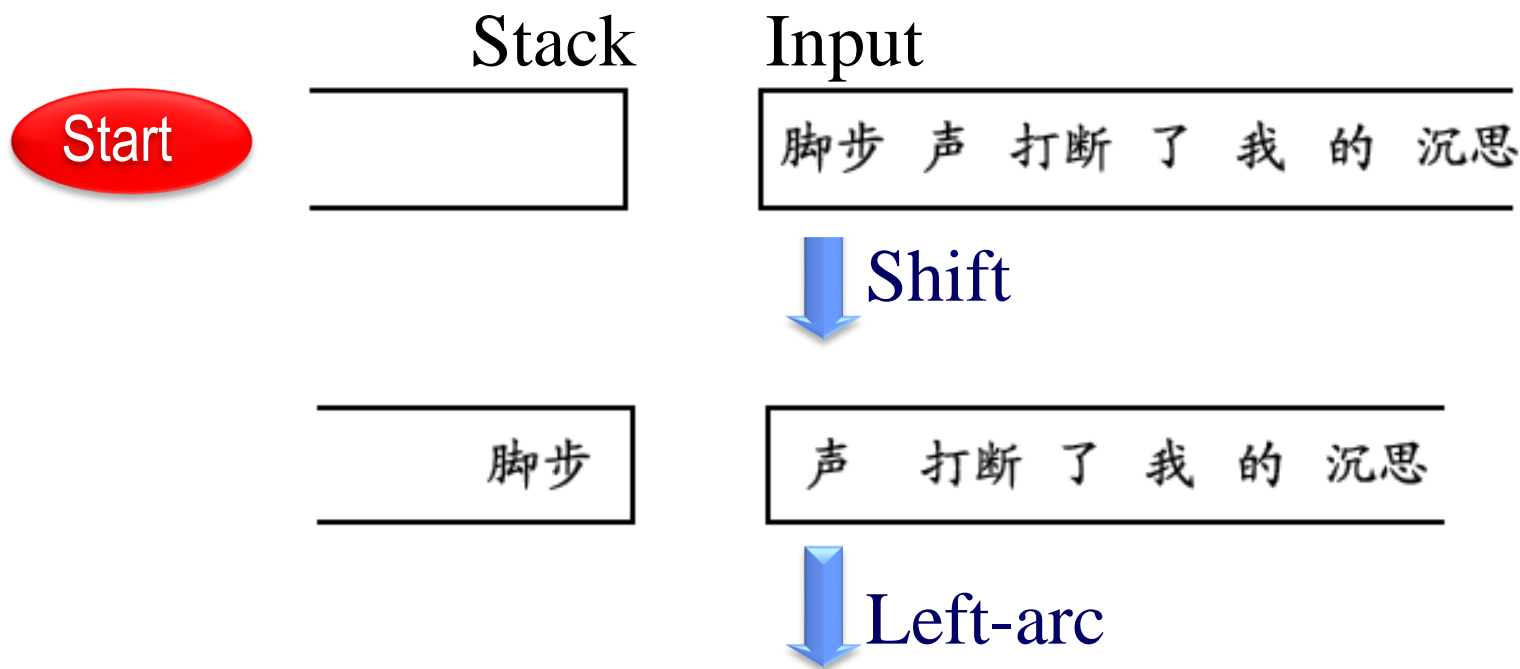
Left-Arc _l	$\frac{[\dots, w_i]_S \quad [w_j, \dots]_{\text{Input}} \quad \neg \exists w_k \rightarrow w_i \in A \quad \text{条件}}{[\dots]_S \quad [w_j, \dots]_{\text{Input}} \quad A \cup \{w_i \xleftarrow{l} w_j\}, \text{pop}(w_i)}$
Right-Arc _r	$\frac{[\dots, w_i]_S \quad [w_j, \dots]_{\text{Input}} \quad \neg \exists w_k \rightarrow w_j \in A_l}{[\dots w_i, w_j]_S \quad [\dots]_{\text{Input}} \quad A \cup \{w_i \xrightarrow{r} w_j\}, \text{push}(w_j)}$
Reduce	$\frac{[\dots w_i]_S \quad [\dots]_{\text{Input}} \quad \exists w_k \rightarrow w_i \in A}{[\dots]_S \quad [\dots]_{\text{Input}} \quad \text{pop}(w_i)}$
Shift	$\frac{[\dots]_S \quad [w_i, \dots]_{\text{Input}}}{[\dots w_i]_S \quad [\dots]_{\text{Input}} \quad \text{push}(w_i)}$

对当前词 w_i 的操作。横上是动作执行之前的句子状态，线下执行动作及之后的句子状态。

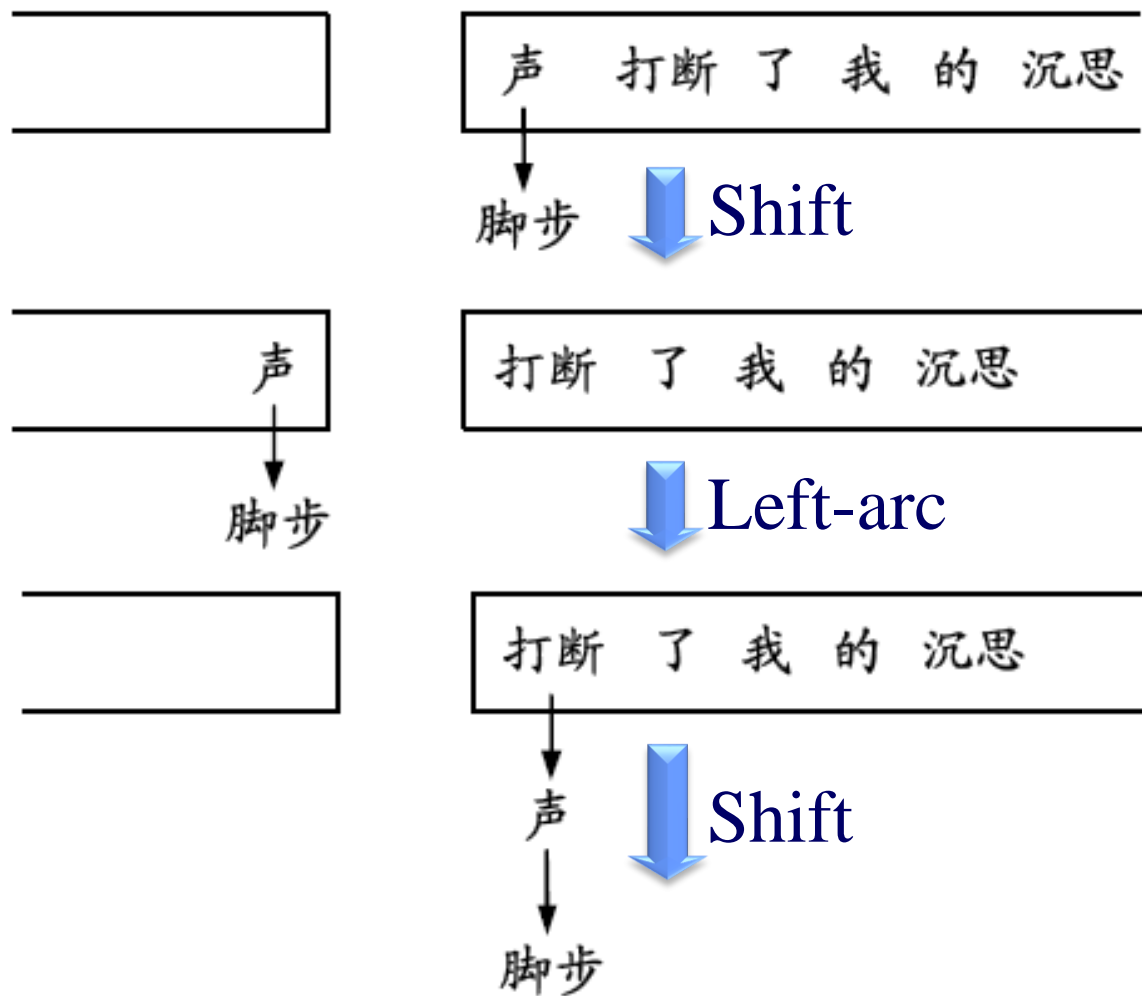
2. 依存关系分析方法

◇ 举例

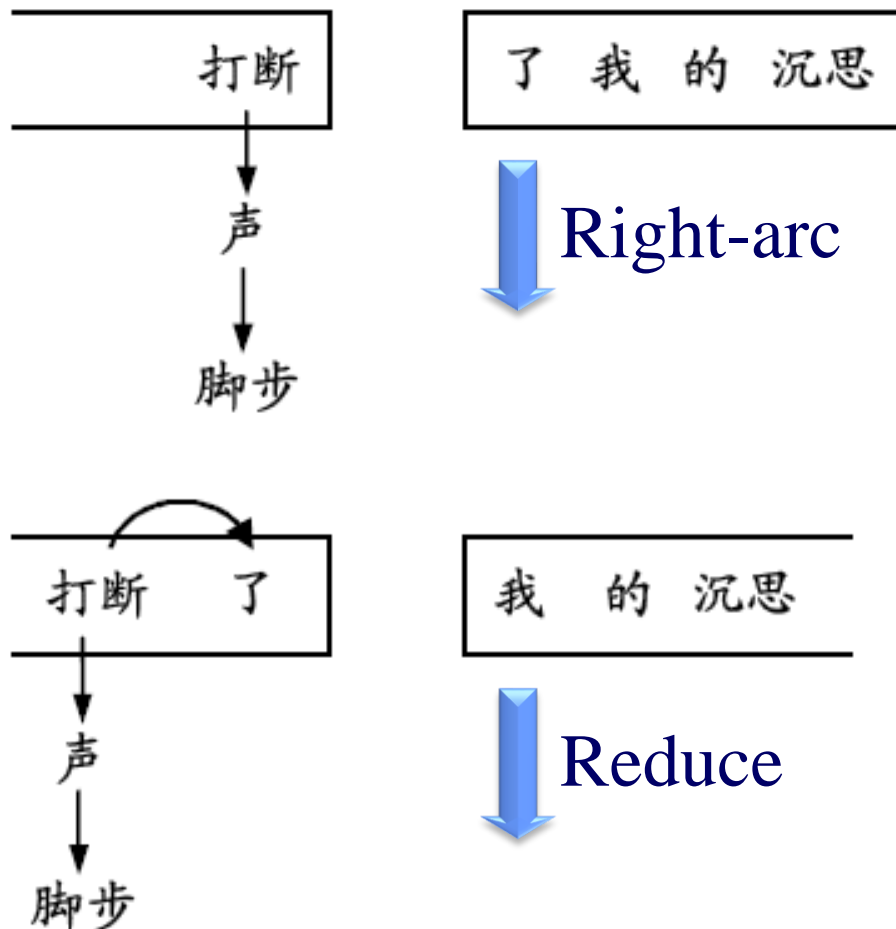
给定如下句子：脚步声打断了我的沉思



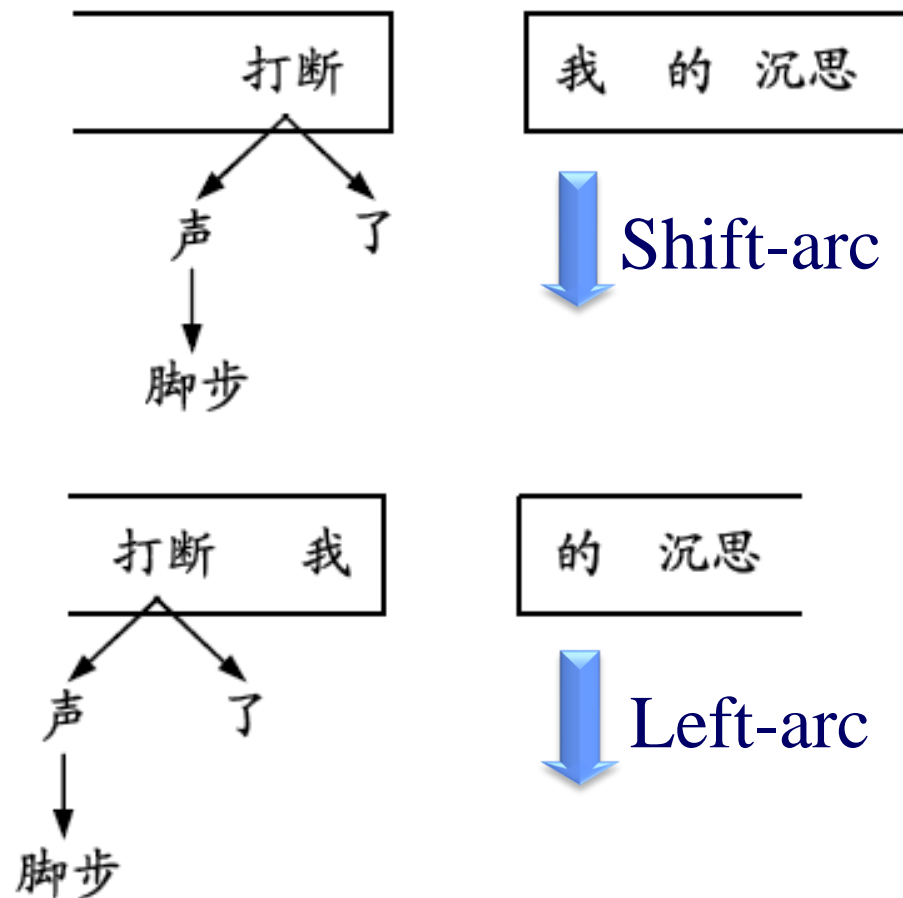
2. 依存关系分析方法



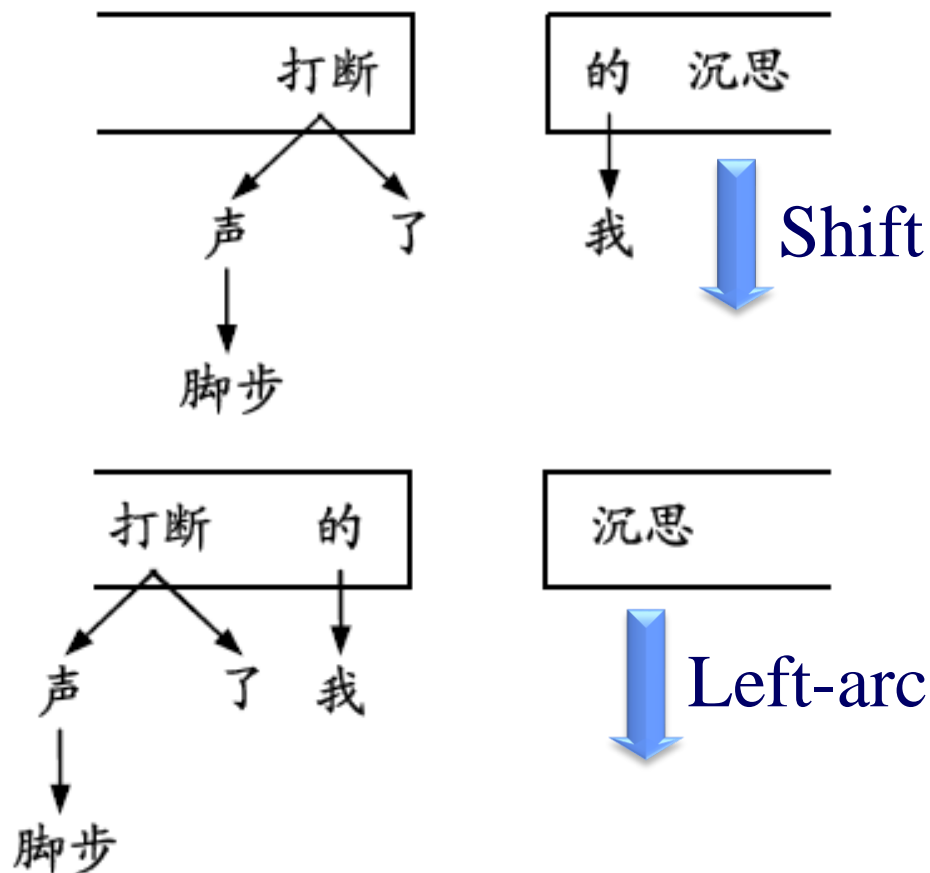
2. 依存关系分析方法



2. 依存关系分析方法

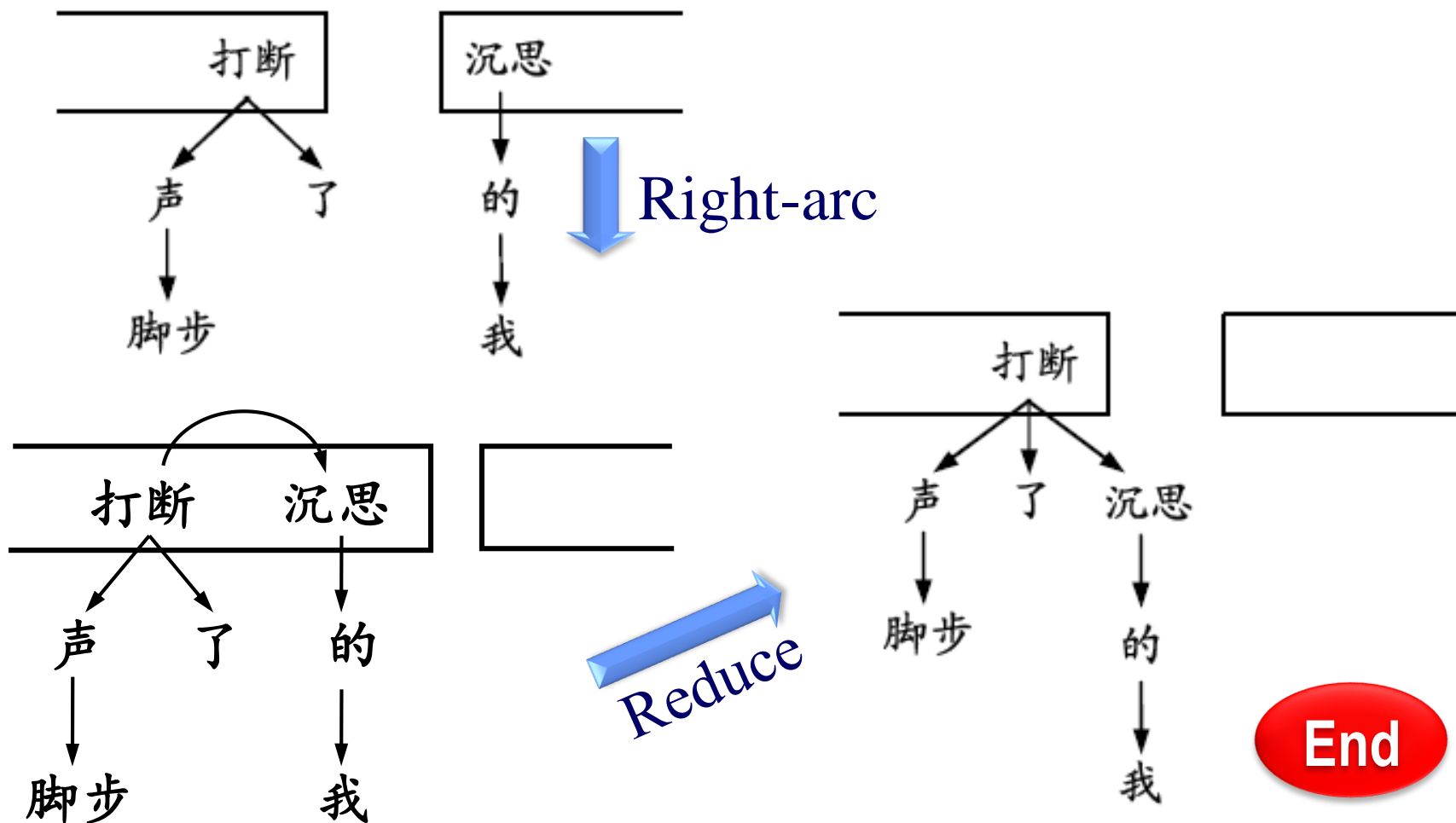


2. 依存关系分析方法

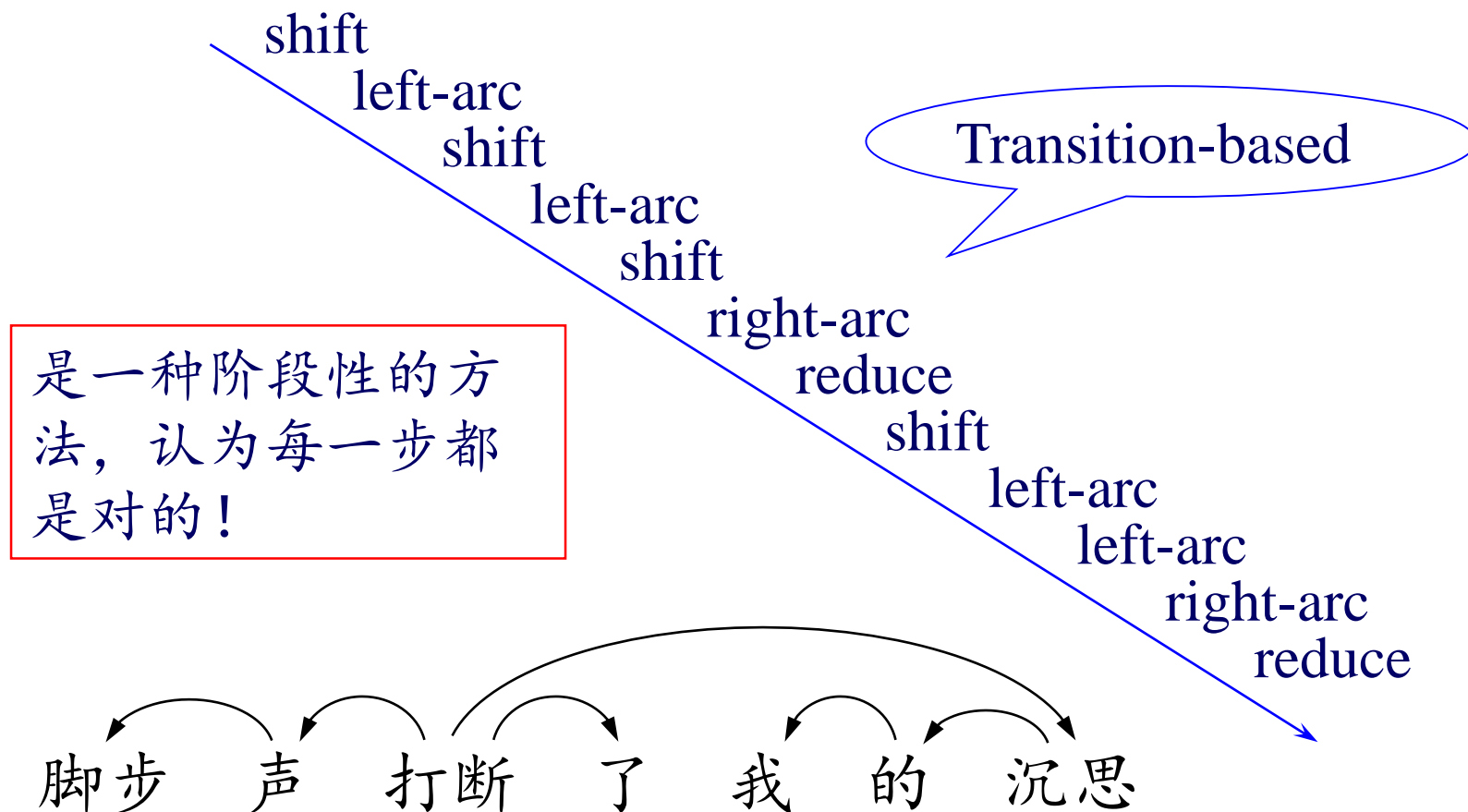


可以看出,一旦子节点在父节点的左边,执行 left_arc 后可以马上归约,但如果子节点在父节点的右边,则不能马上把右边的孩子归约掉,因为也许这个孩子还有孩子在更右边的位置,还没有遍历到。(主要是由从左到右的遍历顺序导致的)。

2. 依存关系分析方法



2. 依存关系分析方法



2. 依存关系分析方法

✧ 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。

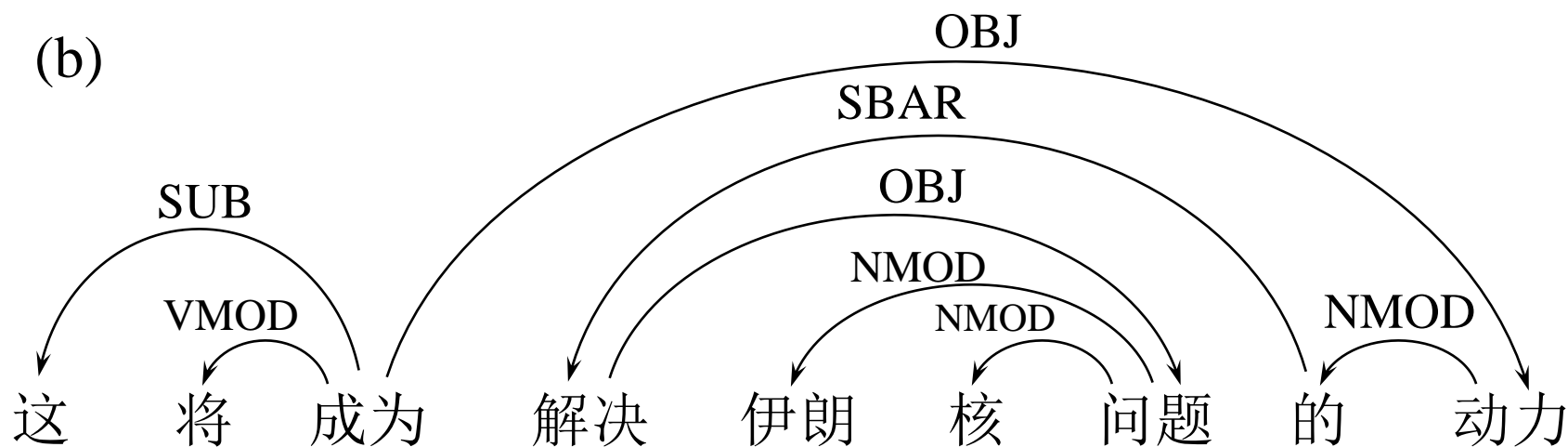
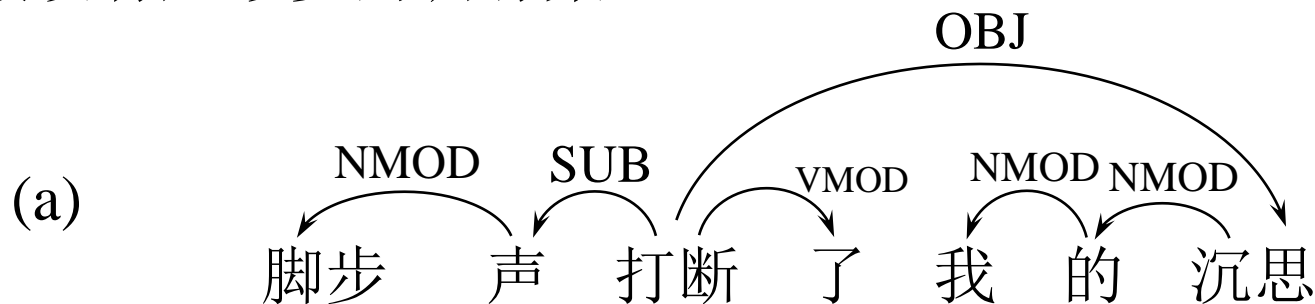
2. 依存关系分析方法

✧ Arc-eager 算法的实现

- **基本思路**：在每一个状态(configuration)下根据当前状态提取特征，然后通过分类器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)，选择最优动作执行，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，构造动作分类器。

2. 依存关系分析方法

假设有足够多的训练集：

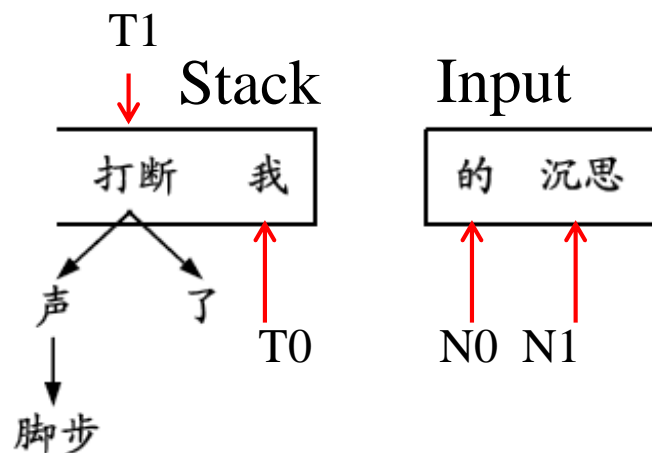


2. 依存关系分析方法

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”)的前两个token(N0)和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。



2. 依存关系分析方法

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

.....

2. 依存关系分析方法

根据前面的图(b)

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

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

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

.....

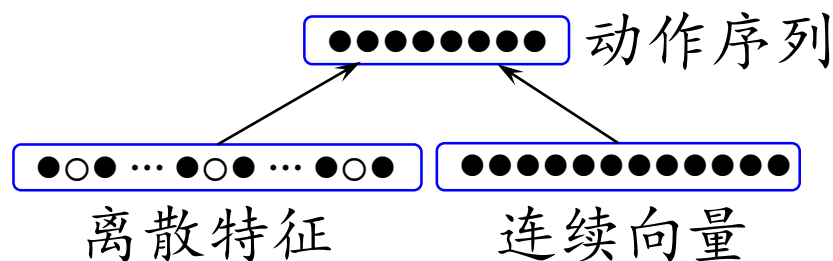
2. 依存关系分析方法

◆神经网络方法与决策方法的结合

✧基本思想

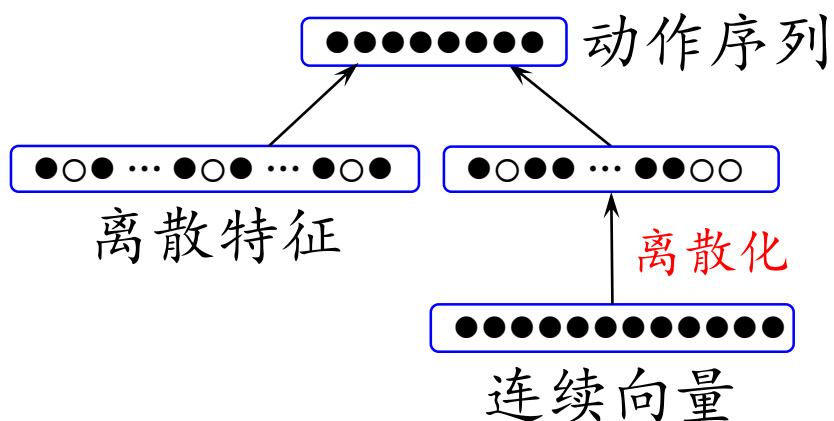
之前的决策式依存分析方法采用离散的特征、线性模型，而神经网络方法采用连续的实数向量表示词汇，词向量可以从大量未标注的样本中训练获得。神经网络模型可以获得非线性关系。那么，是否可以将离散特征与连续特征结合起来，实现它们各自优势的互补？

✧尝试-1: Word2Vec + CRFs



2. 依存关系分析方法

◇尝试-2: 连续特征离散化后与原离散特征结合



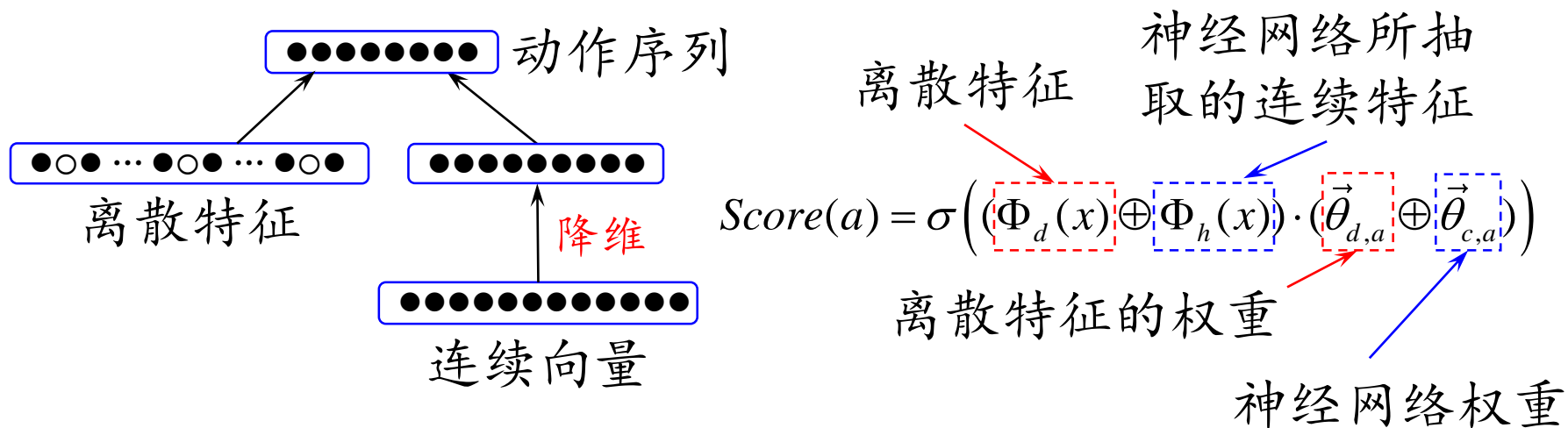
Jiang Guo, Wanxiang Che, Haifeng Wang, Ting Liu. Revisiting Embedding Features for Simple Semi-supervised Learning. *Proc. EMNLP'2014*

离散特征有很多优点:

- 特征的增加或减少都容易处理, 易于模型快速迭代;
- 稀疏向量内积乘法运算速度快, 计算结果方便存储;
- 离散化后的特征对异常数据有较好的鲁棒性...

2. 依存关系分析方法


◇尝试-3：连续特征隐层向量与原离散特征结合



M. Zhang and Y. Zhang, Combining Discrete and Continuous Features for Deterministic Transition-based Dependency Parsing. *Proc. EMNLP'2015*

- 变量 a 表示依存分析动;
- 运算符 \oplus 表示向量拼接;
- x 表示当前状态;
- d 和 h 分别是离散或隐藏特征的标记;
- $\vec{\theta}_{c,a}$ 是隐藏层和输出层之间的权重。

本章内容

1. 概述
2. 依存关系分析方法
-  3. 分析结果评价
4. 短语结构与依存关系
5. 英汉句法结构特点对比
6. 习题

3. 分析结果评价

◆性能指标

- 无标记依存正确率(unlabeled attachment score, UA 或 UAS):
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 或 LAS):
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。

3. 分析结果评价

- 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)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。

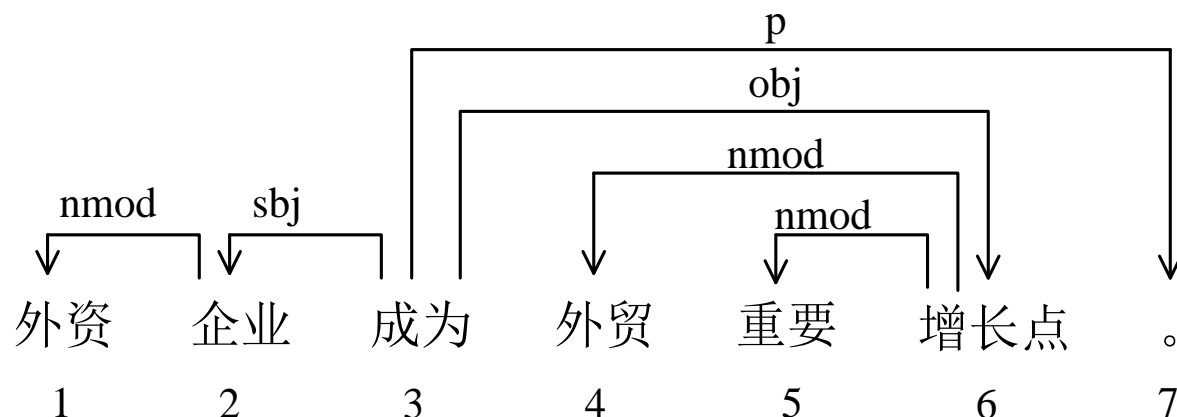
- 完全匹配率(complete match, CM):

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**。

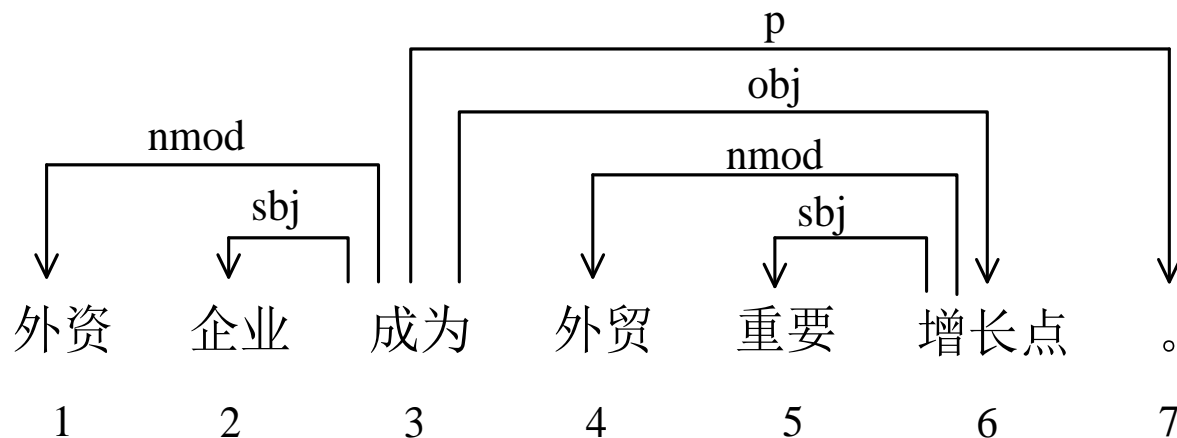
3. 分析结果评价

◆ 举例说明

◇ 答案依存树为：



◇ 系统输出的依存分析树为：



3. 分析结果评价

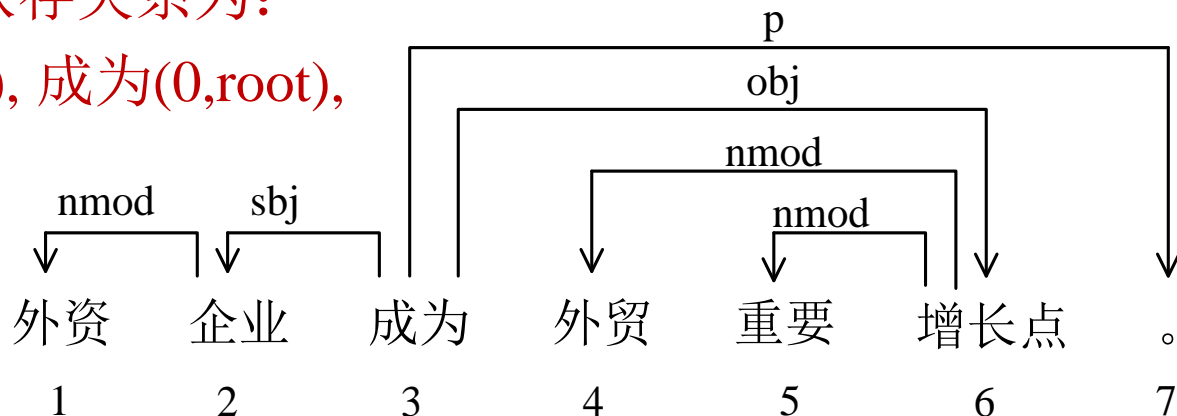
每个词对应的支配词及依存关系为：

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)



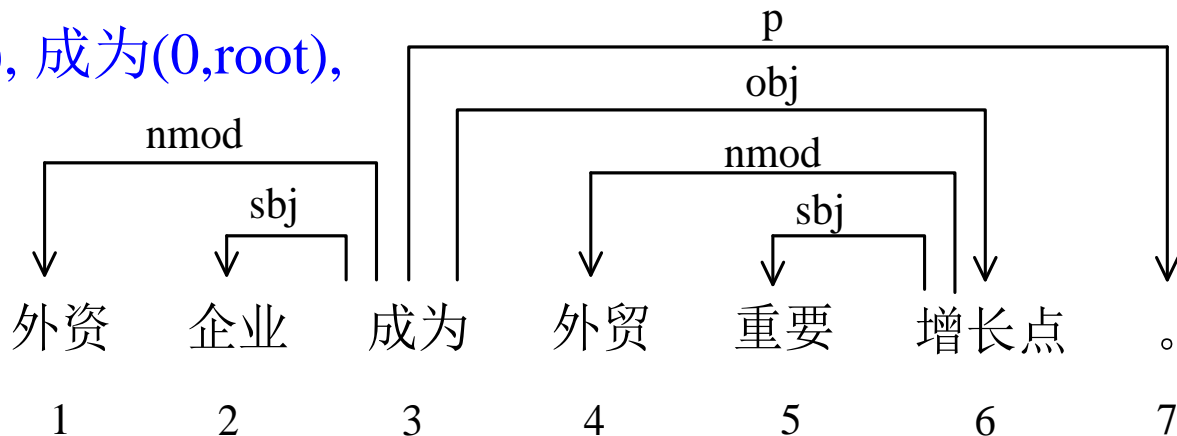
每个词对应的支配词及依存关系为：

外资(3,nmod), 企业(3,sbj), 成为(0,root),

外贸(6,nmod),

重要(6,sbj),

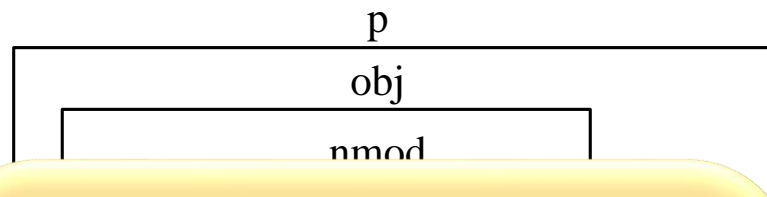
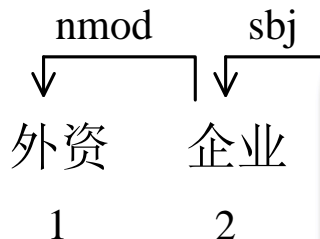
增长点(3,obj), 。 (3,p)



3. 分析结果评价

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)



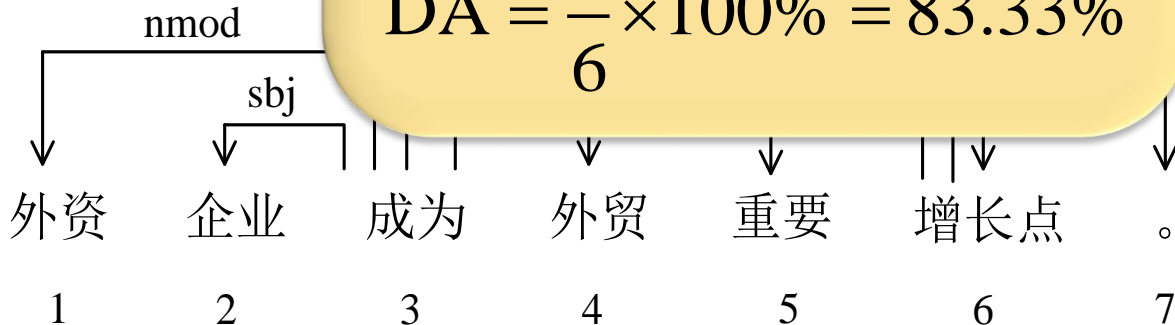
$$UA = \frac{6}{7} \times 100\% = 85.71\%$$

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

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

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)



3. 分析结果评价

◆目前的性能水平

➤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。

分析器	UA(%)	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

说明：①Charniak00 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。②Yamada03 和 Nivre04 是基于转换的确定性方法。③POS(%)指词性标注器自动标注的正确率。

3. 分析结果评价

➤ 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

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

[Duan07]X. Duan et al. 2007. Probabilistic models for action-based Chinese dependency parsing. *Proc. of ECML-ECPPKDD*, pp: 559-566

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

3. 分析结果评价

➤ 神经网络方法与决策方法结合后的性能

- ✧ 利用《华尔街日报》(Wall Street Journal, WSJ)语料做训练;
- ✧ 部分WSJ语料用于开发集;
- ✧ 在WSJ语料上训练出来的CRFs模型进行词性标注。

Domain	#Sent	#Word	TA(%)
WSJ-train	30060	731678	97.03
WSJ-dev	1336	32092	96.88
WSJ-test	1640	35590	97.51

TA 是词性标注的准确率

✧ 结果

Model	UAS	LAS	OOV	OOE
①	89.17	87.21	84.13	91.35
②	89.33	87.21	83.82	90.83
③	90.61	88.68	88.00	93.77

OOV: out-of-vocabulary;

OOE: out-of-embedding vocabulary

对于规范的英文文本，目前分析的总体水平(UAS)达到90%以上，汉语分析的性能接近90%。

3. 分析结果评价

◆ 代表系统

- Stanford Parser

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

- MST Parser (Minimum-Spanning Tree Parser)

<http://www.seas.upenn.edu/~strctrln/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)

本章内容

1. 概述
2. 依存关系分析方法
3. 分析结果评价
-  4. 短语结构与依存关系
5. 英汉句法结构特点对比
6. 习题

4. 短语结构与依存关系

◆ 短语句法结构可转换为依存结构

● 实现方法:

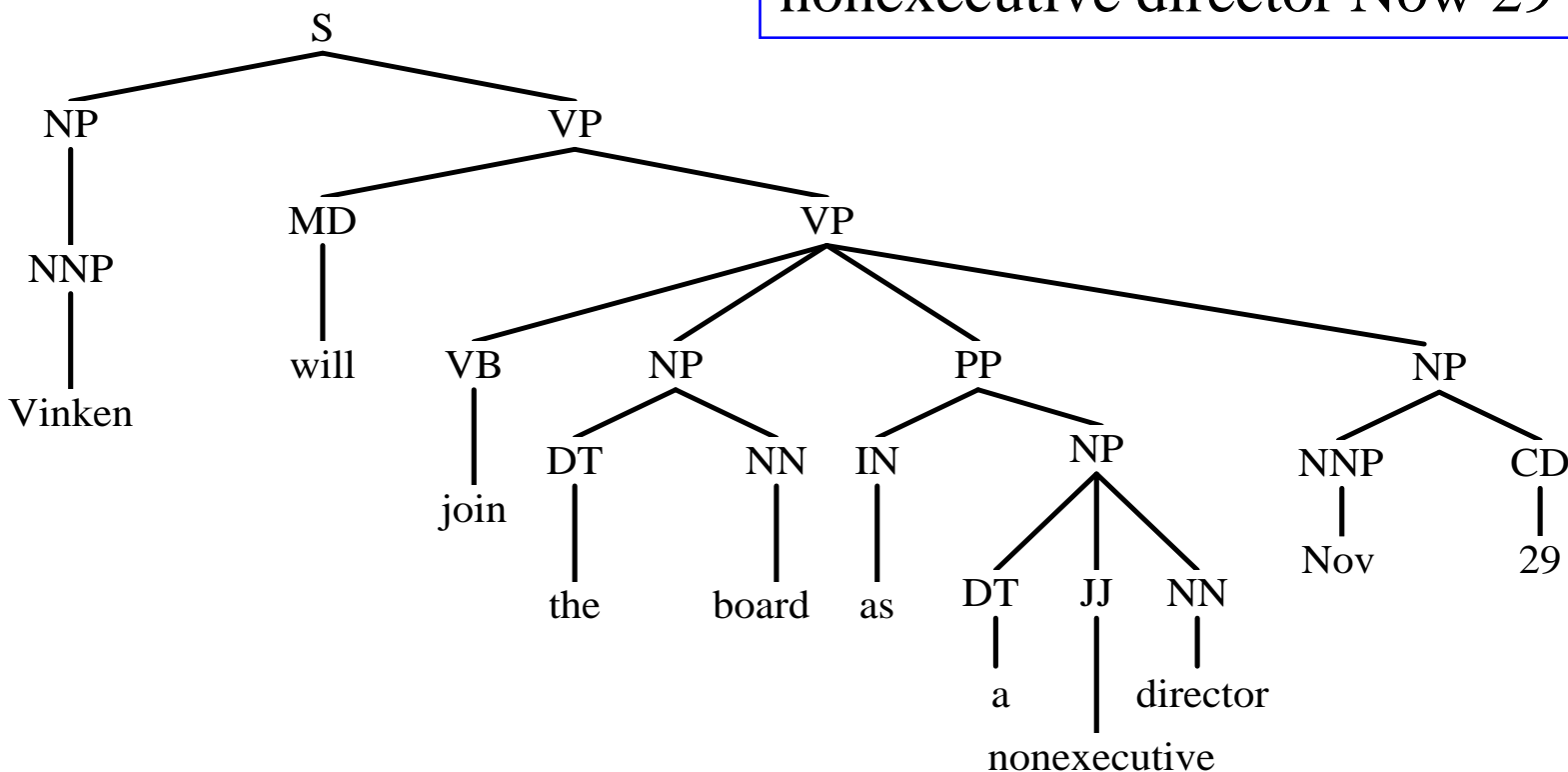
- (1) 定义中心词抽取规则，产生中心词表；
- (2) 根据中心词表，为句子短语结构树中的每个节点选择中心子节点；
- (3) 将非中心子节点的中心词依存到中心子节点的中心词上，得到相应的依存结构。

4. 短语结构与依存关系

◆ 举例

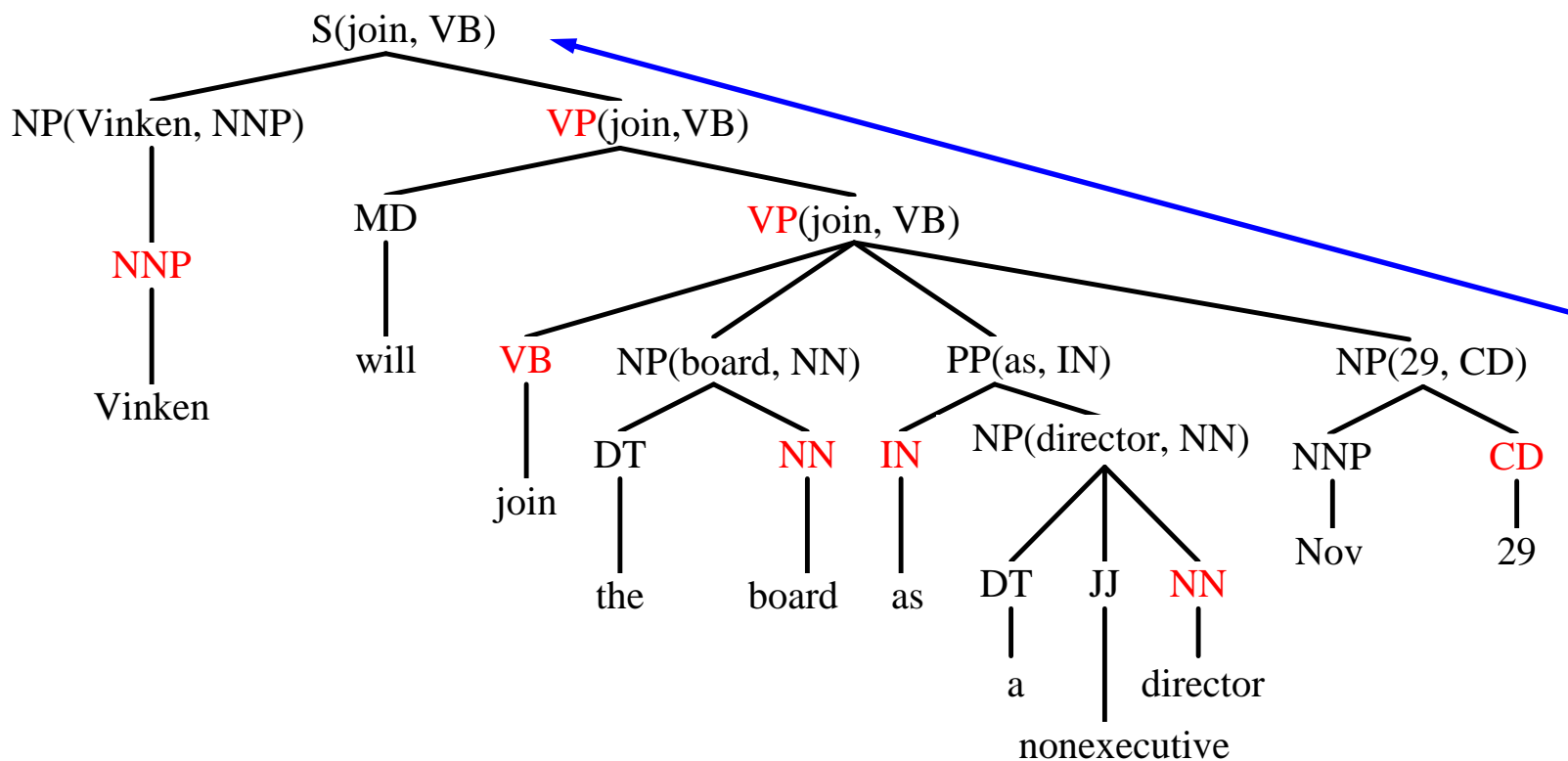
例1：给定如下短语结构树：

Vinken will join the board as a nonexecutive director Nov 29



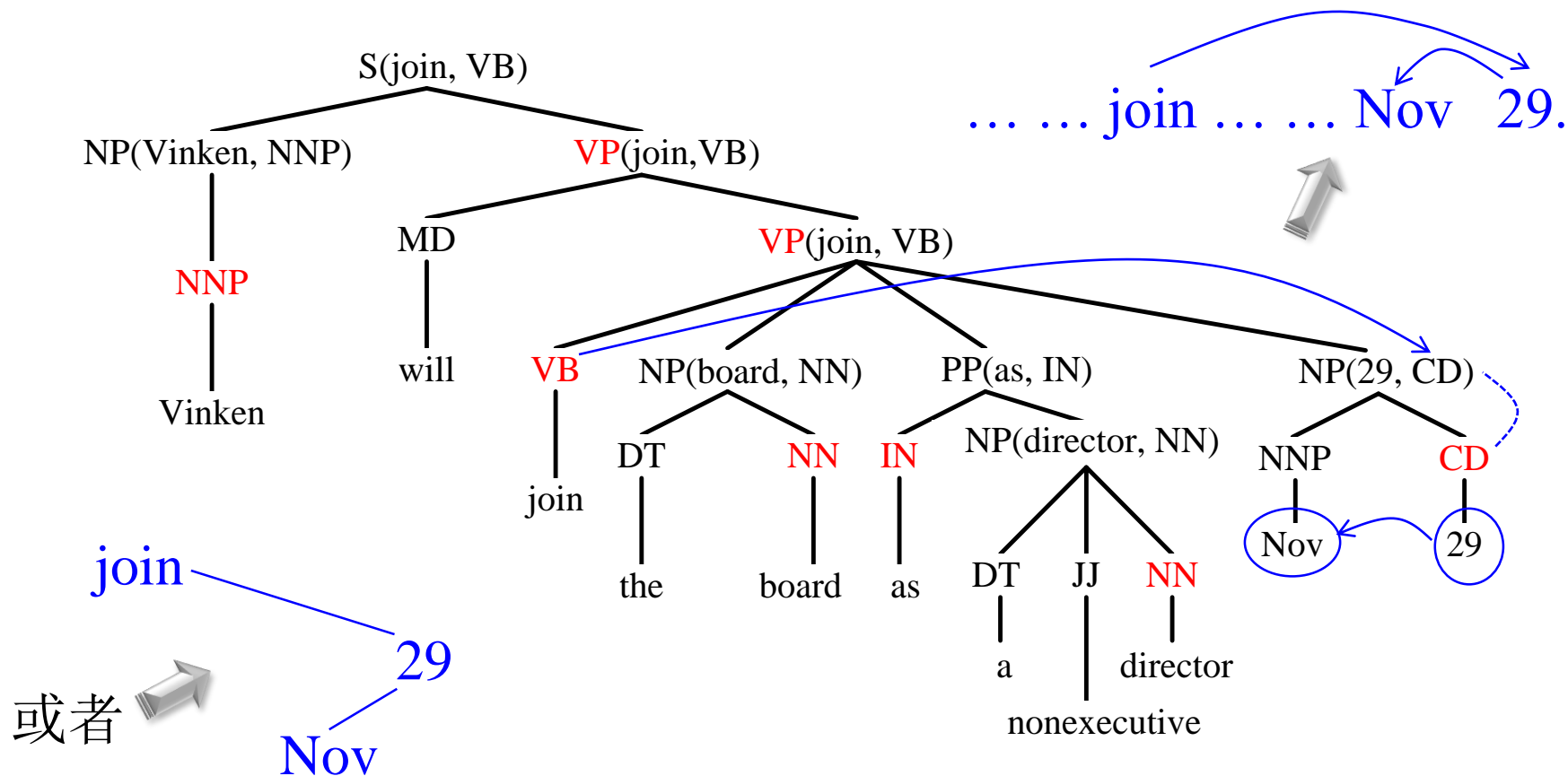
4. 短语结构与依存关系

(1)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）



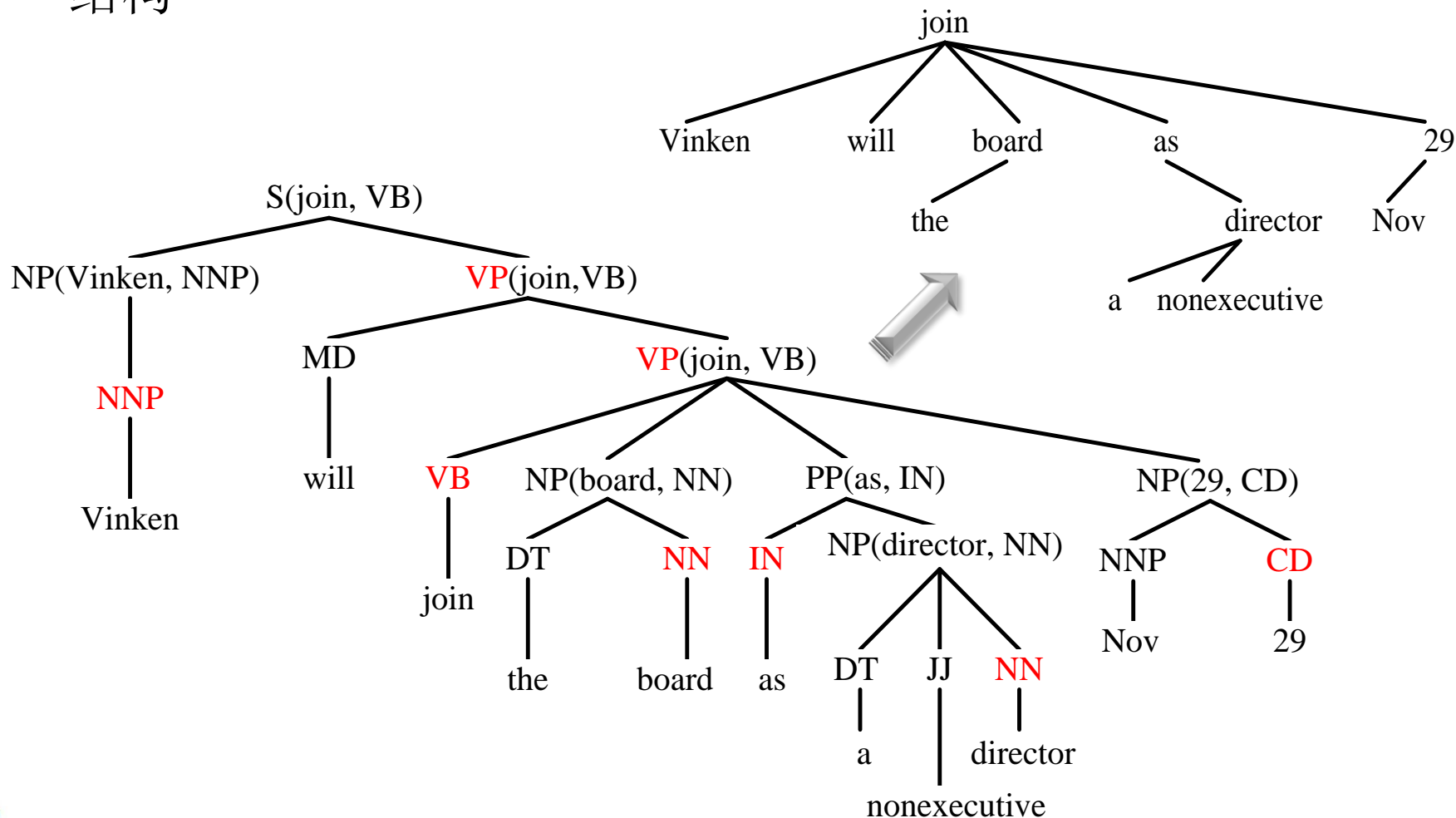
4. 短语结构与依存关系

(2)将非中心子节点的中心词依存到中心子节点的中心词上



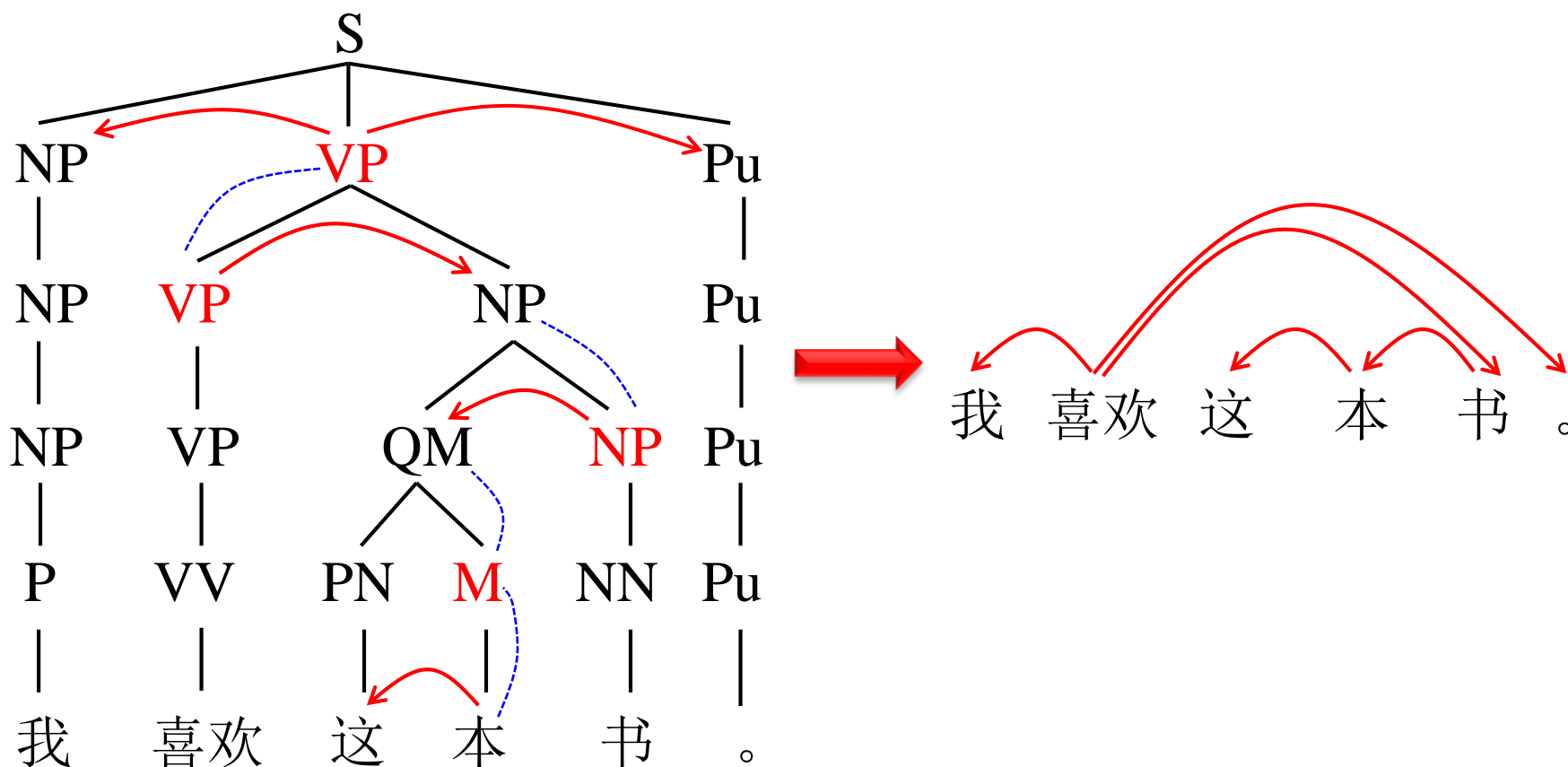
4. 短语结构与依存关系

(3)对所有的节点执行上述操作，最终得到整个句子的依存关系结构



4. 短语结构与依存关系

例2: 句子“我喜欢这本书。”的短语结构树如下:



本章内容

1. 概述
2. 依存关系分析方法
3. 分析结果评价
4. 短语结构与依存关系
- ➡ 5. 英汉句法结构特点对比
6. 习题

5. 英汉句法结构特点对比

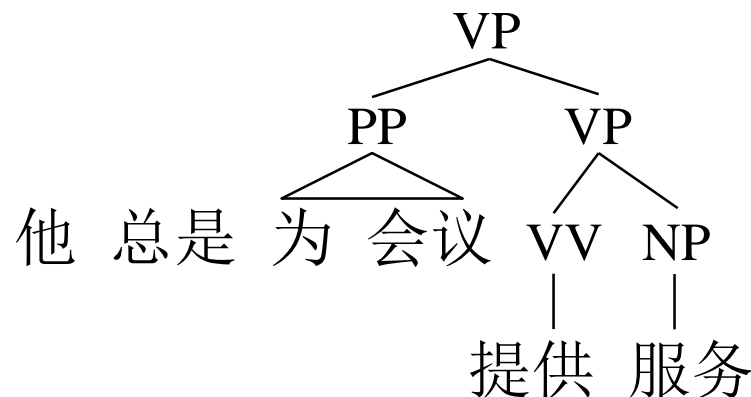
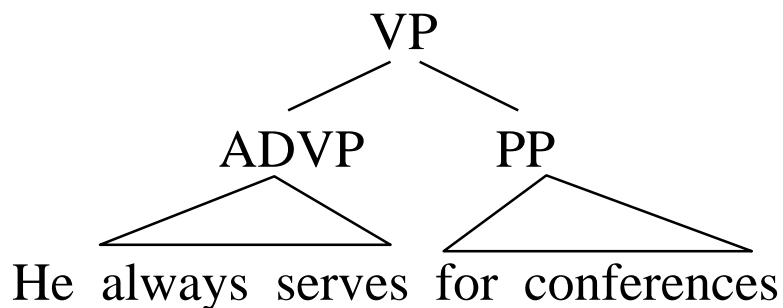
◆说明

- ① 此处撇开汉语的分词和词性兼类可能对句法分析带来的影响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。
- ② 以定性对比为主，没有充足的统计数据支撑相应的观点。

5. 英汉句法结构特点对比

◆特点归纳

- (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。
- (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:



5. 英汉句法结构特点对比

- (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。

我喜欢这里。/ I like it here.

- (4)本质上，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。

5. 英汉句法结构特点对比

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。

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。所以在很多情况下，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，工作压力也大，累得我精疲力尽，这种状况直接影响了我的生活，家里的孩子也没人照顾，身体一天天跨下来。”

中文资源联盟(Chinese LDC)中流水复句占41.3%[李幸, 2005]。

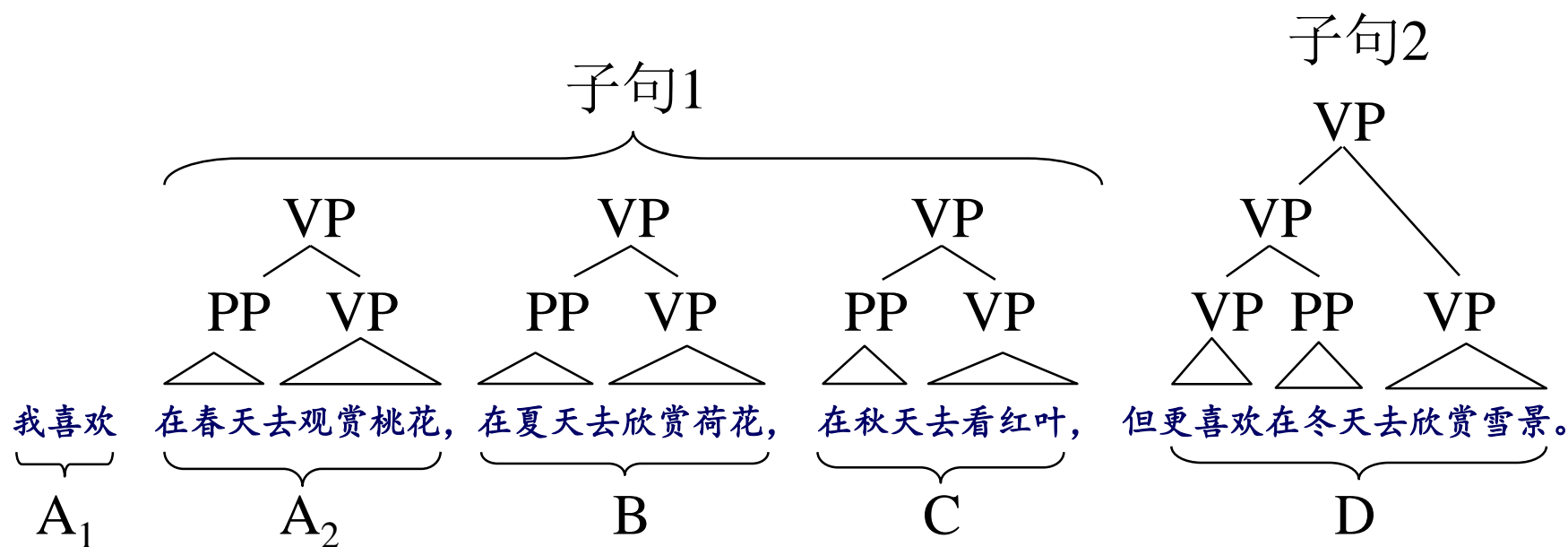
5. 英汉句法结构特点对比

◆ 汉语长句的层次化句法分析策略

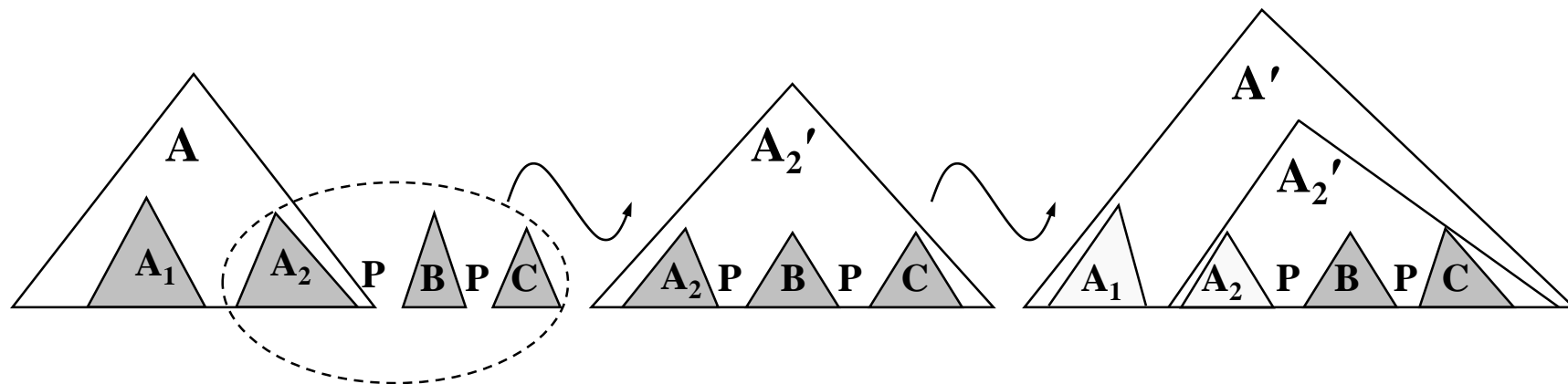
- (1) 对包含“分割”标点的长句进行分割；
- (2) 对分割后的各个子句分别进行句法分析(即第一级分析)，分析得到的各个最大概率的子树根节点的词类或者短语类别标记作为第二级句法分析的输入；
- (3) 通过第二遍分析找到各子句或短语之间的结构关系，从而获得最终整句的最大概率分析树。

5. 英汉句法结构特点对比

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



5. 英汉句法结构特点对比



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

◆ 短语结构分析方法

● 基于规则的方法

❖ 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)

● 基于神经网络的分析方法

● 短语结构分析器性能评测

● 局部句法分析

本章小结


◆依存关系分析

- 基本方法
- 依存句法分析器实现、性能评价
- 短语结构与依存结构

◆短语结构与依存关系之间的转换

◆汉语句法结构特点对比

本章内容

1. 概述
2. 依存关系分析方法
3. 分析结果评价
4. 短语结构与依存关系
5. 英汉句法结构特点对比
-  6. 习题

6. 习题

1. 如有条件，利用树库语料抽取PCFG规则，实现一个基于PCFG的短语结构句法分析器。
2. 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, 对比分析其性能。
3. 设计实现基于 LSTM-CRFs 的短语结构分析器。
4. 设计实现程序，对于任意给定的一棵短语结构树将其转换成依存关系图。
5. 设计实现基于端到端神经网络模型的句法分析器。
6. 思考一下，有没有可能在没有任何标注的句法树库情况下，实现无监督的句法分析(无论是短语结构还是依存关系)?
7. 了解线图分析方法的基本原理，编写程序分别实现自顶向下(top-down)和自底向上(Bottom-up)的线图分析器，对比两种算法的不同。

谢谢!

Thanks!

