

“自然语言处理导论”课程讲义

句法分析2： 依存句法分析(Dependency Parsing)

孙栩

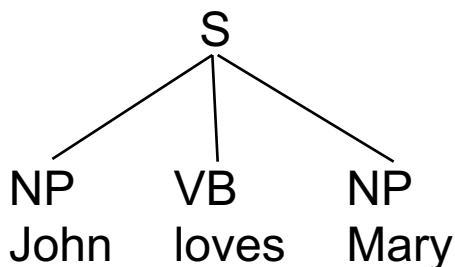
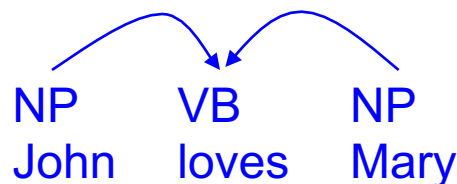
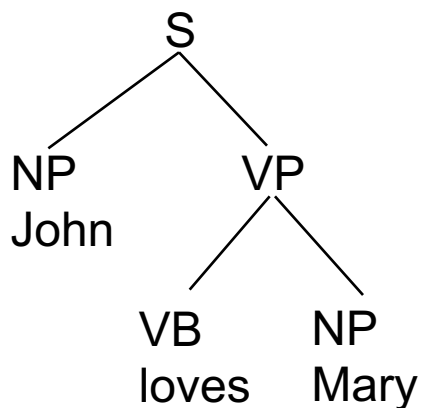
信息科学技术学院

xusun@pku.edu.cn

<http://xusun.org>

□ 输出的类别

- 传统句法分析模式（如之前讲的上下文无关文法CFG、PCFG），**依存分析模式**



- **依存句法分析(Dependency-based grammar parsing)**
 - 根据依存语法规则 G , 给定一个输入字符串 $x \in \Sigma^*$, 计算 x 的依存句法树 y

- **也有依存篇章分析(Dependency-based text parsing)**

依存语法 (Dependency Grammar)

□ 依存语法

- 依存语法 (Dependency grammar , DG) 是相对比较晚出现的语法分析理论，核心是基于依存句法关系，可以追溯到Lucien Tesnière提出的相关依存句法理论
- **基本思想：用若干基于词对的非对称依存箭头来刻画句法结构**

□ 依存关系

- 依存句法关系是以动词为依存结构的核心，一般把动词作为依存关系的头节点 (head)
- 依存结构取决于头节点 (head) 和附属节点 (dependents)之间的依存关系、类别

□ 动词为中心的思想

- 给定一个句子，认为所有的句法单元(e.g. 词) 要么直接或者间接地依赖于某个动词

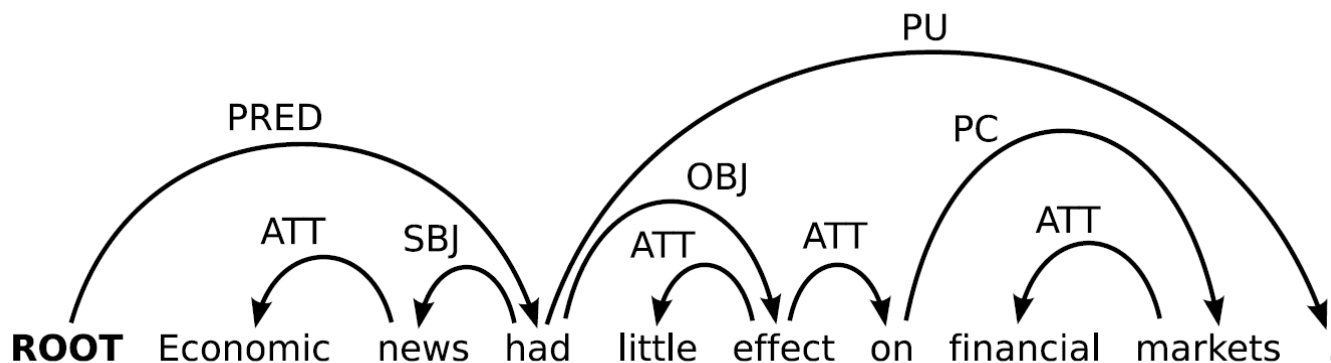
□ 依存语法的优点

- 很适合对词序比较自由、灵活的语言进行句法结构分析
- 适合捕捉长距离的句法结构依存关系

□ 有向图(Directed graphs)

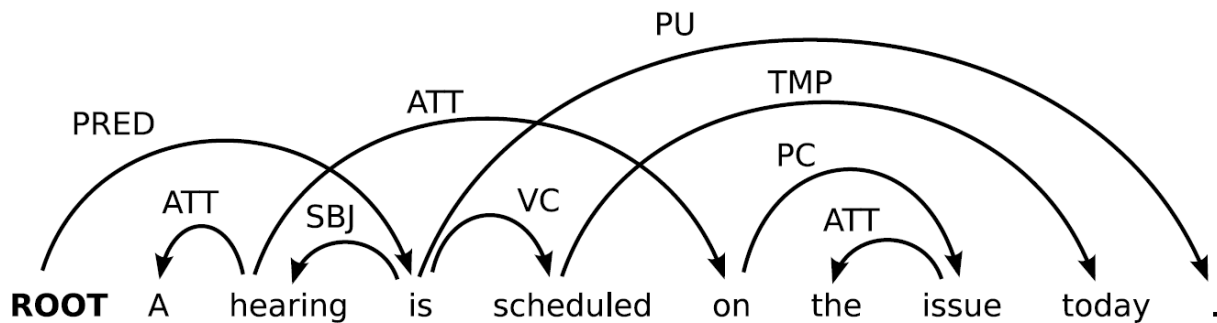
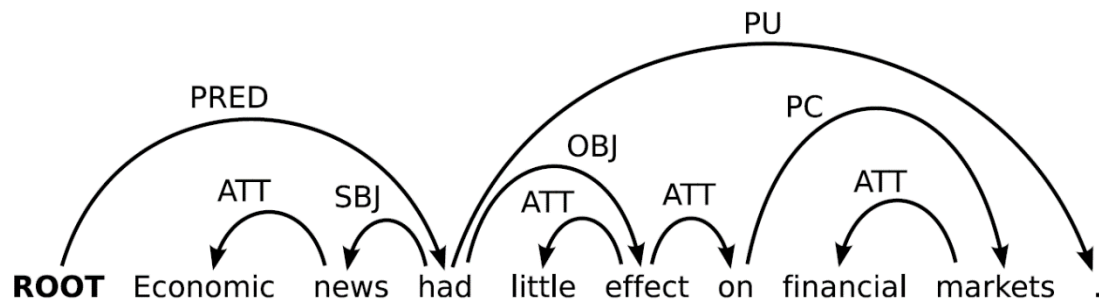
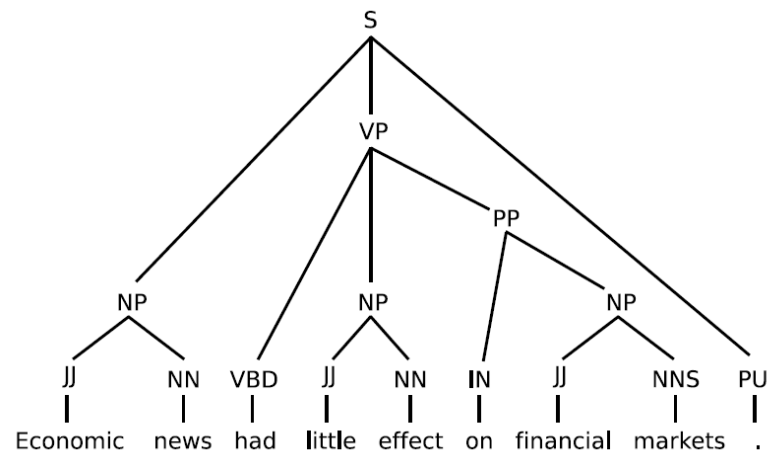
- V 代表节点的集合(词)
- E 代表箭头的集合(依存关系)
- L 代表 E 的类别集合(依存关系的类别)

□ 举例:



依存表示(Dependency Representations)

依存结构表达的信息和传统的短语结构句法树表达的信息是不一样的，可以表达更长距离的信息依存关系



依存表示(Dependency Representations)

怎么确定头结点(head, H)和其依赖节点(dependent, D)

□ 一些比较通用的判断原则:

1. H 决定了依存关系的类别
2. H 决定了依存关系的语义的类别; D 只是对语义信息进行了一定的补充
3. H 在结构上是必须的; D 在结构上不是必须的

□ 依存结构的一些限制条件

□ 单头结点(single-head)

- 对于任何一个节点/词，其只能有最多一个头节点

□ 联通图(connectedness)

- 没有完全独立的节点/词

□ 无环图(acyclicity)

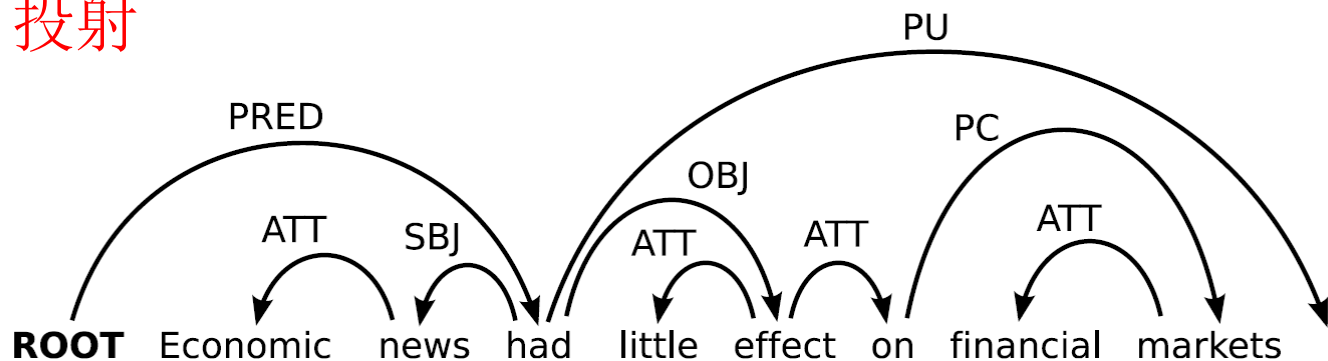
- 依存结构不会存在环形

□ 可选：投射性(projectivity)

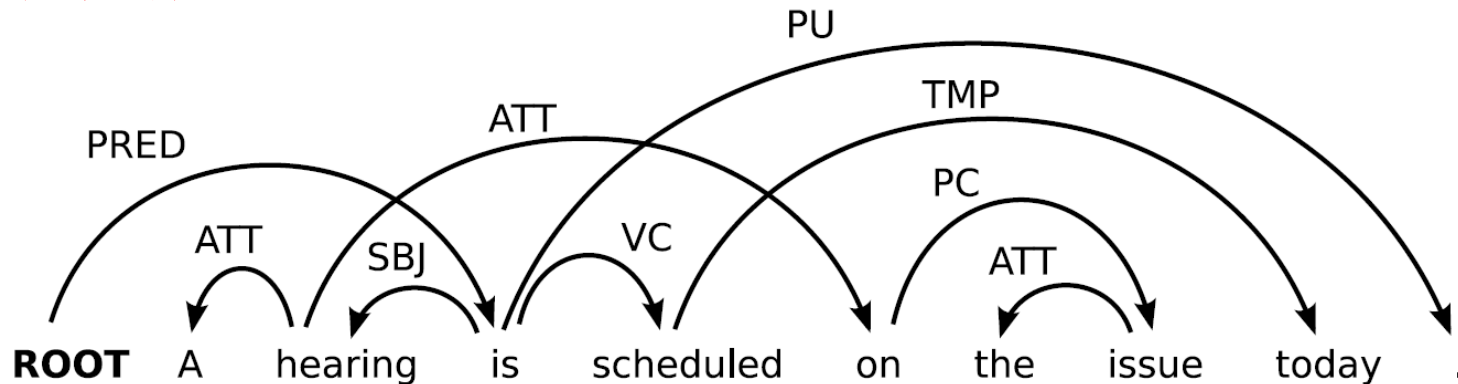
- 没有交叉的依存箭头
- 也就是依存结构是嵌套的、局部的
- 这主要是为了高效率的依存句法分析算法，为了实现动态规划
- 这是可选的，不是必须的。也有非投射性的依存语法、对应的算法

依存表示(Dependency Representations)

投射

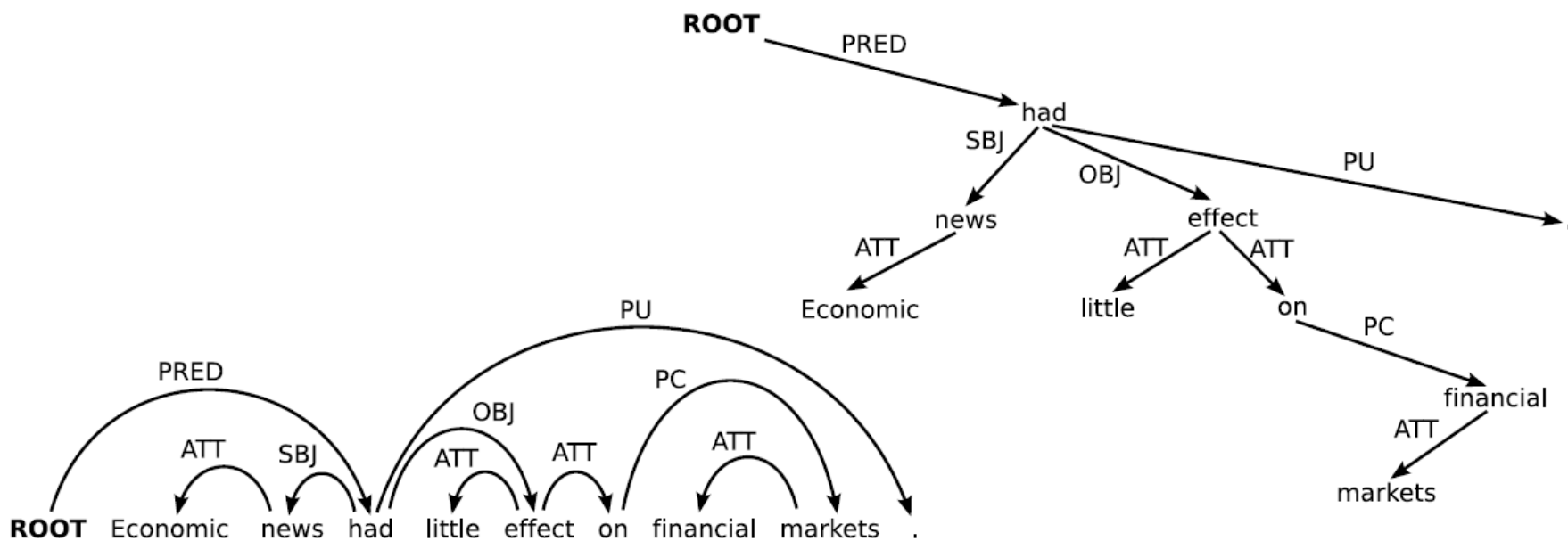


非投射



依存表示(Dependency Representations)

- 依存句法分析其实也是树状结构！



□ 主要的依存句法分析算法

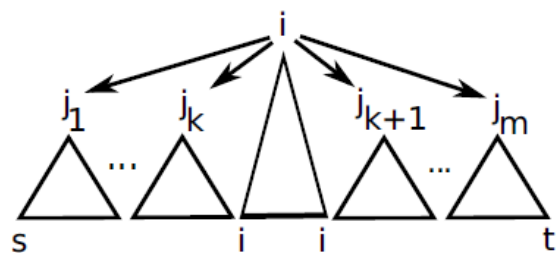
□ 图方法(Graph-Based Parsing)

- 用动态规划算法寻找全局最优的句法树
- 3次方复杂度

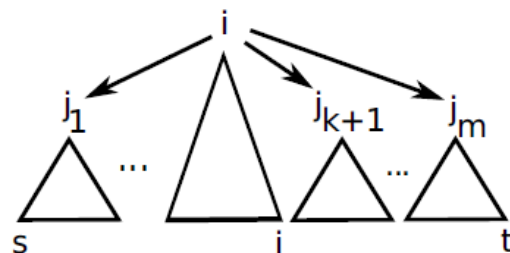
□ 局部分类方法(Deterministic parsing)

- 用局部分类方法寻找局部最优的句法树
- 线性复杂度

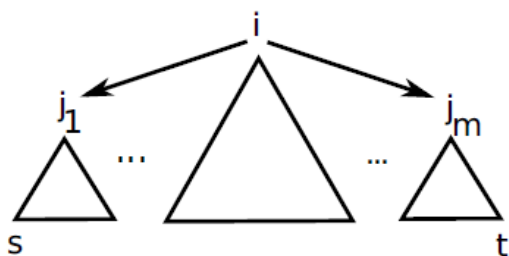
图方法1 : Cocke-Kasami-Younger (CKY)算法



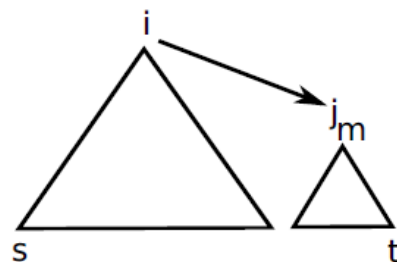
a



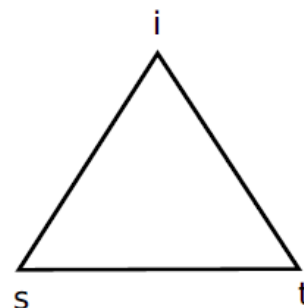
b



c



d



e

动态规划+分治法

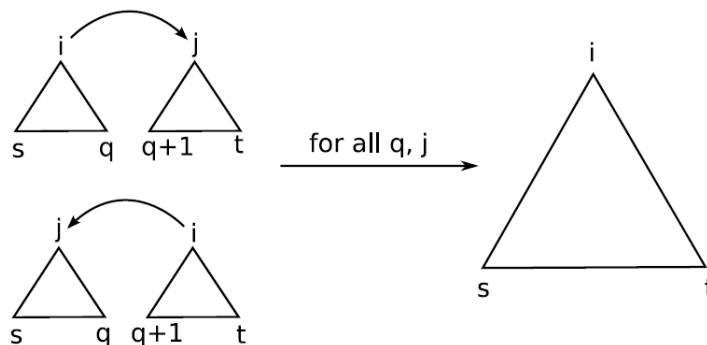
□ 图方法1 : Cocke-Kasami-Younger (CKY)算法

初始化

$$C[i][i][i] = 0.0, \text{ for all } 0 \leq i \leq n$$

递归计算

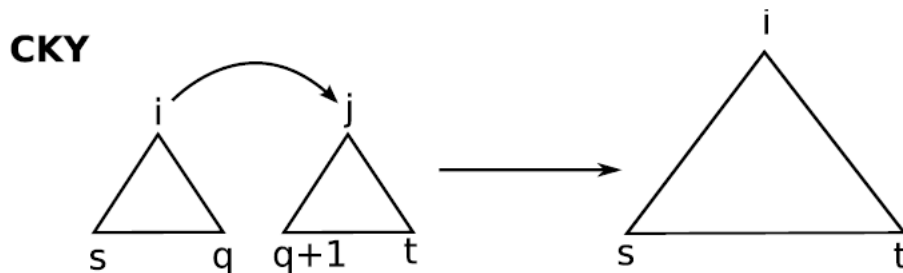
$$C[s][t][i] = \max_{s \leq q < t, s \leq j \leq t} \begin{cases} C[s][q][i] + C[q+1][t][j] + \lambda(w_i, w_j) & \text{if } j > i \\ C[s][q][j] + C[q+1][t][i] + \lambda(w_i, w_j) & \text{if } j < i \end{cases}$$



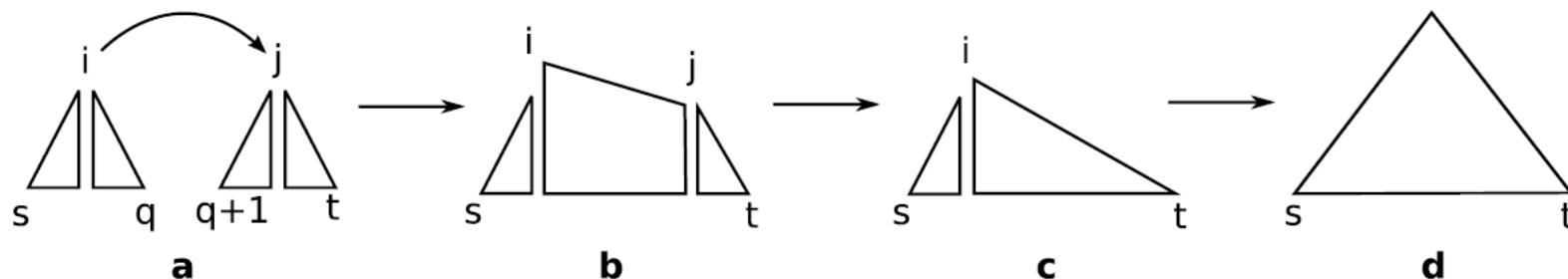
图方法1 : Cocke-Kasami-Younger (CKY)算法

时间复杂度 $O(n^5)$

改进 : Eisner动态规划算法



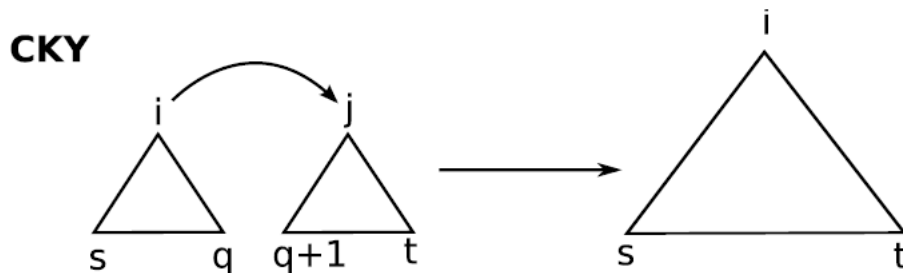
Eisner



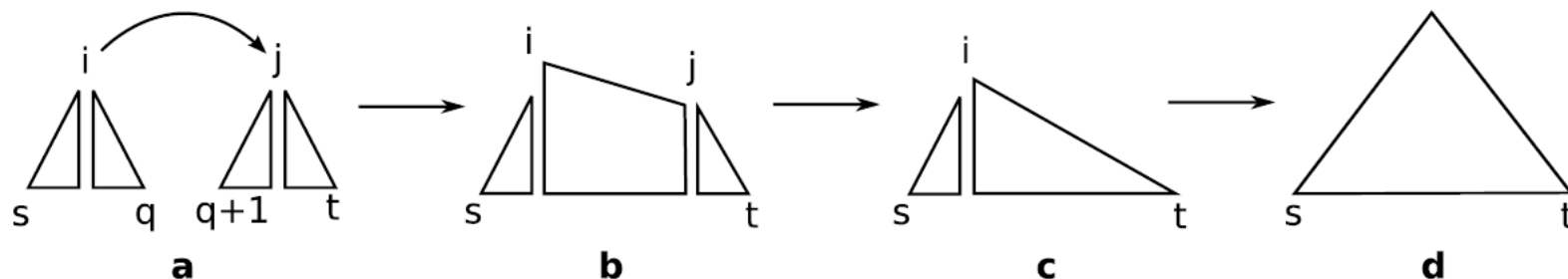
图方法1 : Cocke-Kasami-Younger (CKY)算法

时间复杂度 $O(n^5)$

改进 : Eisner动态规划算法

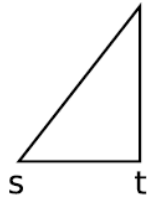


Eisner

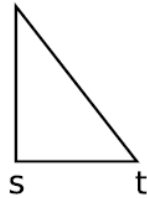


图方法(Graph-Based Parsing)

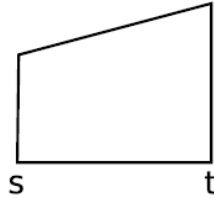
$E[s][t][0][0]$



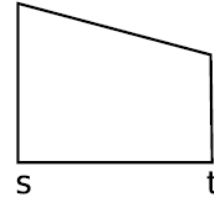
$E[s][t][1][0]$



$E[s][t][0][1]$



$E[s][t][1][1]$



Eisner(S, Γ, λ)

Sentence $S = w_0w_1 \dots w_n$

Arc weight parameters $\lambda_{(w_i, w_j)} \in \lambda$

- 1 Instantiate $E[n][n][2][2] \in \mathbb{R}$
- 2 Initialization: $E[s][s][d][c] = 0.0$ for all s, d, c
- 3 for $m : 1..n$
- 4 for $s : 1..n$
- 5 $t = s + m$
- 6 if $t > n$ then break

% Create subgraphs with $c = 1$ by adding arcs (step a-b in figure 4.5)

- 7 $E[s][t][0][1] = \max_{s \leq q < t} (E[s][q][1][0] + E[q+1][t][0][0] + \lambda_{(w_t, w_s)})$
- 8 $E[s][t][1][1] = \max_{s \leq q < t} (E[s][q][1][0] + E[q+1][t][0][0] + \lambda_{(w_s, w_t)})$

% Add corresponding left/right subgraphs (step b-c in figure 4.5)

- 9 $E[s][t][0][0] = \max_{s \leq q < t} (E[s][q][0][0] + E[q][t][0][1])$
- 10 $E[s][t][1][0] = \max_{s < q \leq t} (E[s][q][1][1] + E[q][t][1][0])$

图方法2: Eisner动态规划算法

时间复杂度 $O(n^3)$

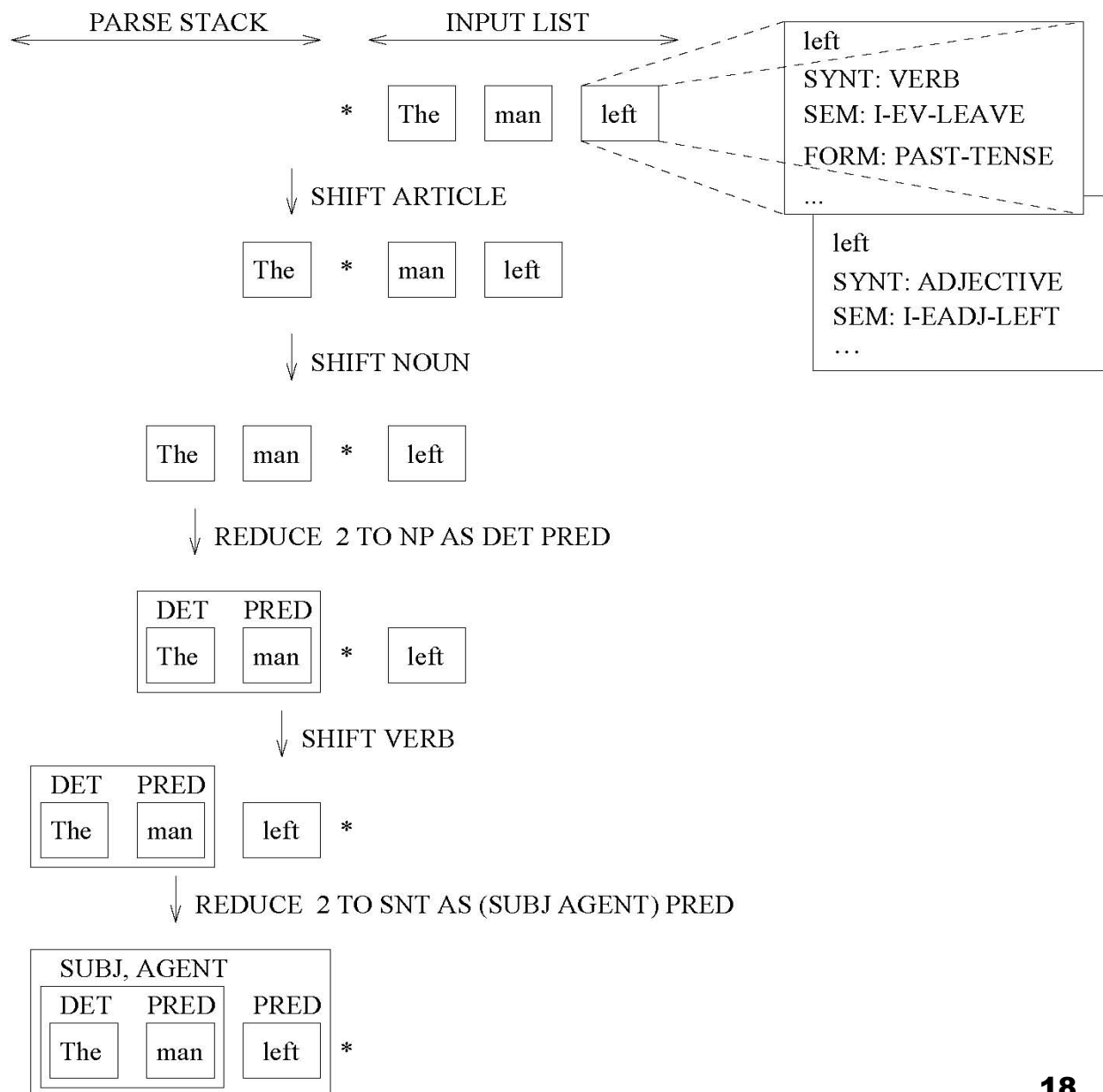
□ 主要思路

- 类似一个滑动窗口的方法，一步一步进行局部分类，最后获得句法树
- 无法获得全局最优的句法树，但是速度快，线性复杂度

□ 代表方法

- Shift-reduce 句法分析算法
 - Standard (Kudo, Matsumoto, Yamada)
 - Arc-eager (Nivre)

举例

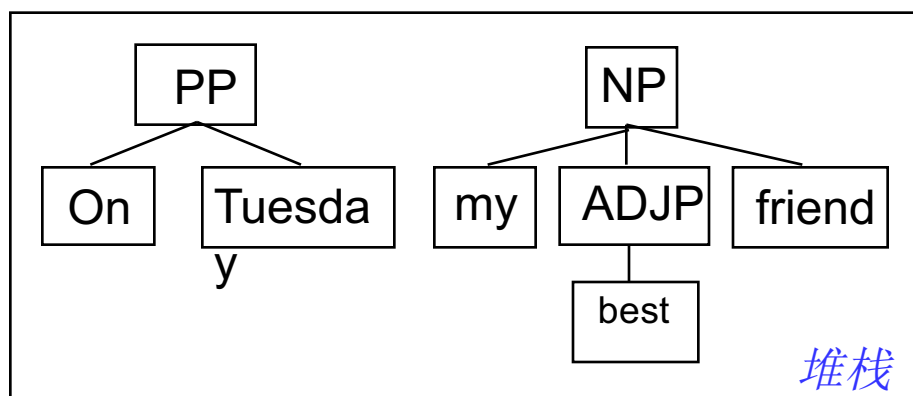


1. 输入序列

- 输入的词序列，也就是待分析的句子
- 逐步压入分析堆栈，直到词序列为空
- 输入序列为空的时候，分析完成，获得依存句法树

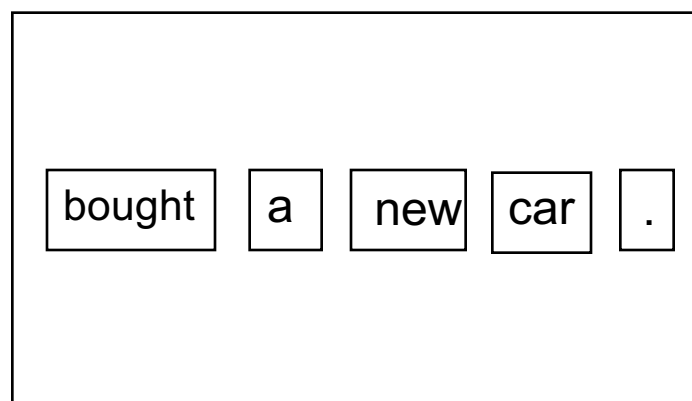
2. 分析堆栈

- 分析堆栈里面代表的是部分分析好的句法结构
- 堆栈的顶部是局部分类的“活跃部分”
- 输入序列为空的时候，堆栈中获得最后的完整句法分析树



堆栈

堆栈
的顶
部



输入序列

Shift-Reduce算法的步骤

- **2类主要的步骤:**
 - Shift步骤
 - 把元素从输入队列shift到堆栈
 - Reduce步骤
 - 在堆栈内分析依存结构

怎么定义特征(feature)

- 为了在堆栈进行准确的分类
 - 句法或者语义类别
 - 时态、数字、大小写信息
 - 词性信息
 - ...

Feature stem	Value
syntactic class of item at position 1	noun
semantic class of item at position 1	relative-temporal-interval
semantic class of object of item at position -1	monetary-quantity
tense of item at position -1	past tense
np-vp agreement of items at position -2 and -1	true
subcat affinity of 1 to -1 relative to -2	positive

□ 图方法

- 目前能达到93%以上

□ 局部分类方法

- 目前能达到91%以上

□ 优点:

- 依存关系和实际的语义关系比较接近，有助于对句子的语义方面的理解
- 依存关系的定义相对比较简单，有助于高效率的句法分析
- 因为能够有效建模长距离依赖关系，依存句法更适合词序列比较自由、灵活的语言

□ 参考书

- 《统计自然语言处理》第8章：句法分析
 - 8.9 依存语法理论简介
 - 8.10 依存句法分析
 - 8.11 依存分析器性能评价
 - 8.12 短语结构与依存结构之间的关系
 - Page 220 - 240

- McDonald, Ryan, Crammer, Koby and Pereira, Fernando (2005). [Online large-margin training of dependency parsers](#), Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL), Ann Arbor, MI, pp. 91–98.
- McDonald, Ryan and Pereira, Fernando (2006). [Online learning of approximate dependency parsing algorithms](#), Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL), Trento, Italy, pp. 81–88.

- 简述依存句法与CFG/PCFG的区别
- 何为依存句法树的投射性？
- 依存句法分析的两大类方法分别是什么？
- CYK算法的复杂度？Eisner算法的复杂度？Eisner算法的哪些假设导致了复杂度的降低？

Thanks!