依存句法分析

汇报人：刁永祥 || 2018年12月26日

Part 1 概述

Part 2 依存句法的形式化

Part 3 依存句法分析方法

Part 4 结束语

Part 1 概述

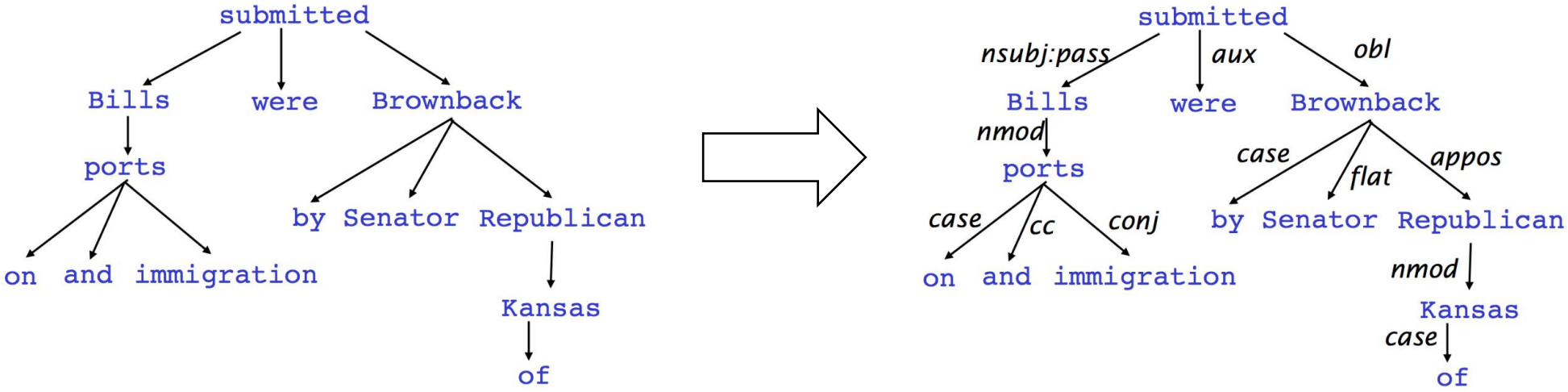
Part 2 依存句法的形式化

Part 3 依存句法分析方法

Part 4 结束语

概述

✓句法分析研究主要有两方面内容：句法结构分析、依存句法分析



✓依存句法理论聚焦于词汇间的依存关系（dependency relation）[如果一个单词修饰另一个单词，则称该单词依赖于另一个单词]

Part 1 概述

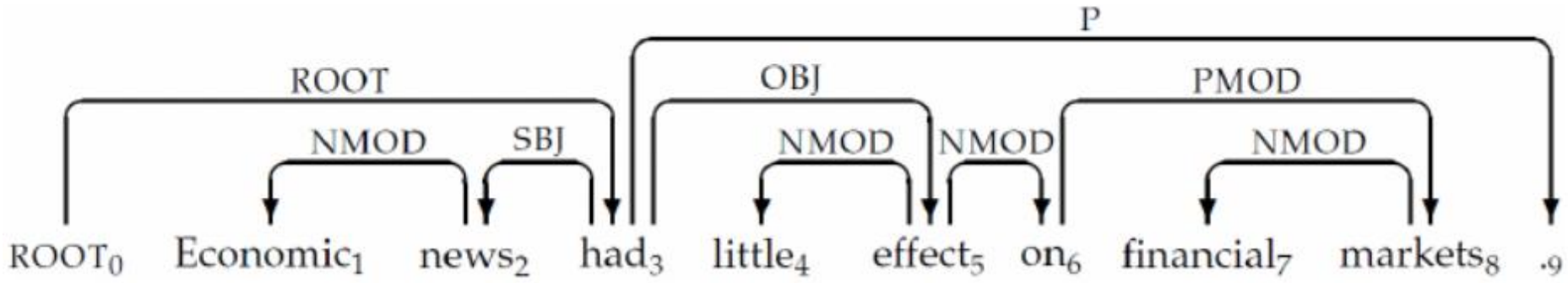
Part 2 依存句法的形式化

Part 3 依存句法分析方法

Part 4 结束语

依存图（Dependency Graph）

- ✓ 句子的语法结构采用依存图建模
- ✓ 依存图：使用带有标签（label）的弧（arc）表示每个词及其语法依赖
- ✓ (Penn TreeBank) eg：Economic news had little effects on financial markets.



依存语法的形式化

✓**定义2.1** 给定一个依存标签（dependency label）的集合L = {l1,l2, …. , ln}，那么句子 x = (w0, w1, … ,wn) 的依存图是一个带关系（label）的有向图（directed graph），有：

(1) V = {0, 1, ..., n} 是顶点集

(2) $A \subseteq V \times L \times V$ 是带标签的有向边集

✓ **说明**：

① V 是包含 0 ~ n 的整数集合，每个整数对应一个单词（包括根节点）

② A 是三元组构成集合，每个元素为形（i, l, j）的三元组，i、j 是点，l 是依存关系。i 为核心词（head），j 是 i 的修饰词（dependent）

依存语法的形式化

✓ **定义2.2** 一个依存图 G = (V, A) 是合式（well-formed），当且仅当：

① 节点 0 是 root

② 每个节点至多有一个核心词、依存类型

③ 图 G 是无环的，即不存在 A 的非空子集满足如下条件：

$$\{(i_0, l_1, i_1), (i_1, l_2, i_2), \ldots, (i_{k-1}, l_k, i_k)\} \subseteq A \text{ 其中 } i_0 = i_k$$

✓ **定义2.3** 一个依存图 G = (V, A) 是可投影的（projective），当且仅当：

对 A 中 任意弧（i, l, j）和 V 中顶点 k, 若能满足 i < k < j or j < k < i,

则存在子集（non-projective 暂不讨论）：

$$\{(i, l_1, i_1), (i_1, l_2, i_2), \ldots, (i_{k-1}, l_k, i_k)\} \text{ 使得 } i_k = k$$

Part 1 概述

Part 2 依存句法的形式化

Part 3 依存句法分析方法

Part 4 结束语

依存句法分析的方法

✓ 基于图（graph based）的方法（姑且闲之）

① Eisner 的三个概率模型

✓ 基于转移的方法（transition based）

① Yamada & Matsumoto & Nivre 的基于栈的方法
② Covington 的基于表方法（暂不讨论）

Transition based - Definition

✓ **定义3.1** 一个句法分析的**转移系统**（transition system）视为四元组

$$S = (C, T, c_s, C_t)$$

其中：

(1)C是状态(configuration)集合，其中的元素包含一个缓存（buffer）β 还有余下的节点，一个依存弧的集合A

(2)T是转移(transition)集合，每个元素是一个函数$t: C \rightarrow C_t$

(3) c_s 是初始化函数，将句子 $x = (w_0, \ldots, w_n)$映射到一个状态 $\beta = [1, \ldots, n]$

✓ **说明：**

① 每个分析状态（configuration）至少包含一个buffer、A

② buffer 初始化包含与句子 x = (w0,…, wn) 对应的[1, … , n] 节点

Transition based - Definition

- ✓ **定义3.2** 设转移系统 $S=(C,T,c_s,C_t)$，句子 x = (w0, w1, …, wn) 在 S 定义下的转移序列（transition sequence）是一个状态的序列 $C_{0,m}=(c_0,\ldots,c_m)$ 满足：

(1) $c_0=c_s(x), c_m \in C_t$

(2)对任意的 i $(1 \le i \le m)$, 存在 t $\in$ T , 使得 $c_i=t(c_{i-1})$

Transition based - Definition

✓ **定义3.3** 对句子 x = (w0, w1, …, wn), 基于栈（stack based）的**状态表示** 为一个三元组 $c=(\sigma,\beta,A)$ ，其中：

(1)σ是词的栈，词 i 满足 $i \le k (k \le n)$

(2) β是词的缓存，词 j 满足 $j > k$

(3)A是依存弧的集合，且 $G = (\{0, 1, \ldots, n\})$ 是句子 x 的一个依存图

✓ **定义3.4** 一个**基于栈的转移系统**是一个四元组 $S=(C,T,c_s,C_t)$ ，其中：

(1)C是所有基于栈的状态集合

(2)$c_s(x=(w_0,\ldots,w_n))=([0],[1,\ldots,n],[\varnothing])$

(3)T是一个转移动作集合，每个动作以函数 $t: C \rightarrow C_t$

(4)$C_t=\{c \in C \mid c=(\sigma,[],A)\}$

Transition based - Algorithm

✓ 基于栈的依存句法分析算法主要分两种：ARC-Standard、ARC-Eager
✓ ARC-Standard （用得较多）三个分析动作：
① 左弧（左归约，left-arc）
② 右弧（右归约，right-arc）
③ 转移（shift）

ARC-Standard分析动作

	分析动作
左弧$_l$（LEFT-ARC$_l$）	$(\sigma\|i, j\|\beta, A) \Rightarrow (\sigma, j\|\beta, A \cup \{(j, l, i)\})$
右弧$_l$（RIGHT-ARC$_l$）	$(\sigma\|i, j\|\beta, A) \Rightarrow (\sigma, i\|\beta, A \cup \{(i, l, j)\})$
移进（SHIFT）	$(\sigma, i\|\beta, A) \Rightarrow (\sigma\|i, \beta, A)$

其中 $\sigma|i$ 表示栈顶为 i 的栈，$j|\beta$ 表示队首为 j 的缓存

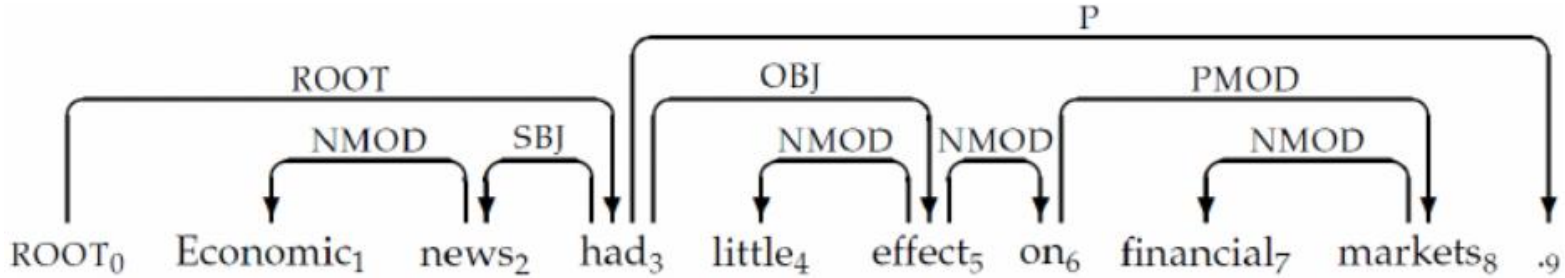
Transition based - Algorithm

ARC-Standard 分析动作（简化版）

	分析动作
左弧 $_l$（LEFT-ARC$_l$）	$(\sigma\|ij,\beta,A)\Rightarrow(\sigma,\beta,A\cup\{(j,l,i)\})$
右弧 $_l$（RIGHT-ARC$_l$）	$(\sigma\|ij,\beta,A)\Rightarrow(\sigma,\beta,A\cup\{(i,l,j)\})$
移进（SHIFT）	$(\sigma,i\|\beta,A)\Rightarrow(\sigma\|i,\beta,A)$

动作可执行的先决条件

动作	先决条件
左弧 $_l$（LEFT-ARC$_l$）	$\neg[i=0]$，$\neg\exists k\exists l'\,[(k,l',j)\in A]$
右弧 $_l$（RIGHT-ARC$_l$）	$\neg\exists k\exists l'\,[(k,l',j)\in A]$



分析动作		状态 (栈 σ,	缓存 β,	弧集 A)
初始化	$\Rightarrow$	([0],	[1, . . . , 9],	$\emptyset$)
SHIFT	$\Rightarrow$	([0, 1],	[2, . . . , 9],	$\emptyset$)
SHIFT	$\Rightarrow$	([0, 1, 2],	[3, . . . , 9],	$\emptyset$)
LEFT-ARC$_{\text{NMOD}}$	$\Rightarrow$	([0, 2],	[3, . . . , 9],	$A_1 = \{(2, \text{NMOD}, 1)\}$)
SHIFT	$\Rightarrow$	([0, 2, 3],	[4, . . . , 9],	A_1)
LEFT-ARC$_{\text{SBJ}}$	$\Rightarrow$	([0, 3],	[4, . . . , 9],	$A_2 = A_1 \cup \{(3, \text{SBJ}, 2)\}$)
SHIFT	$\Rightarrow$	([0, 3, 4],	[5, . . . , 9],	A_2)
SHIFT	$\Rightarrow$	([0, 3, 4, 5],	[6, . . . , 9],	A_2)
LEFT-ARC$_{\text{NMOD}}$	$\Rightarrow$	([0, 3, 5],	[6, . . . , 9],	$A_3 = A_2 \cup \{(5, \text{NMOD}, 4)\}$)
SHIFT	$\Rightarrow$	([0, 3, 5, 6],	[7, 8, 9],	A_3)
SHIFT	$\Rightarrow$	([0, 3, 5, 6, 7],	[8, 9],	A_3)
SHIFT	$\Rightarrow$	([0, 3, 5, 6, 7, 8],	[9],	A_3)
LEFT-ARC$_{\text{NMOD}}$	$\Rightarrow$	([0, 3, 5, 6, 8],	[9],	$A_4 = A_3 \cup \{(8, \text{NMOD}, 7)\}$)
RIGHT-ARC$_{\text{PMOD}}$	$\Rightarrow$	([0, 3, 5, 6],	[9],	$A_5 = A_4 \cup \{(6, \text{PMOD}, 8)\}$)
RIGHT-ARC$_{\text{NMOD}}$	$\Rightarrow$	([0, 3, 5],	[9],	$A_6 = A_5 \cup \{(5, \text{PMOD}, 6)\}$)
RIGHT-ARC$_{\text{OBJ}}$	$\Rightarrow$	([0, 3],	[9],	$A_7 = A_6 \cup \{(3, \text{OBJ}, 5)\}$)
SHIFT	$\Rightarrow$	([0, 3, 9],	[],	A_7)
RIGHT-ARC$_{\text{P}}$	$\Rightarrow$	([0, 3],	[],	$A_8 = A_7 \cup \{(3, \text{P}, 9)\}$)
RIGHT-ARC$_{\text{ROOT}}$	$\Rightarrow$	([0],	[],	$A_9 = A_8 \cup \{(0, \text{ROOT}, 3)\}$)

Part 1 概述

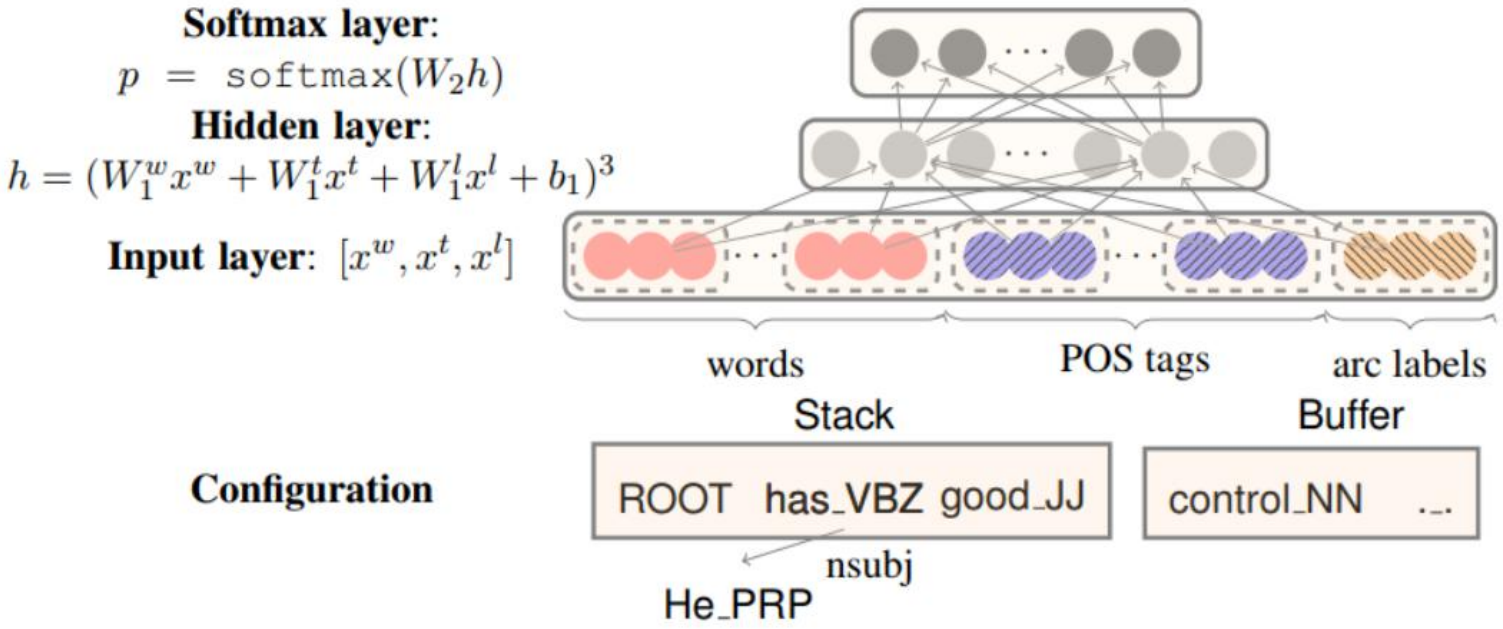
Part 2 依存句法的形式化

Part 3 依存句法分析方法

Part 4 结束语

结束语

- ✓ 基于转移的分析过程，可以用三元组（S, I, A）表示，S：堆栈、I：未处理节点序列、A：依存弧集合
- ✓ 如何从该三元组中，提取丰富特征，是依存句法分析器性能的关键
- ✓ 深度学习在依存句法分析中应用



谢谢聆听