具有自学习能力的语料库句法标注工具CSTT

周　明　　黄昌宁　　杨军东*

清华大学计算机科学与技术系

*烟台大学电子系

摘　要

实现大规模真实文本的处理，需要对语料库进行词法、句法、语义等方面的标注，而句法标注是核心环节。本文以依存语法为依据，采用上下文相关分析方法，设计了语料库句法标注工具CSTT。由于具有自学习功能，可显著提高标注的效率和标注的一致性。对1300个典型汉语简单陈述句做了试验，效果令人满意。

关键词：语料库　句法分析　句法标注　依存语法

1 引言

近年来，国际计算语言学界围绕其战略目标及相应的理论、方法问题展开了热烈的讨论[1]，1990年8月在赫尔辛基举行的第13届国际计算语言学大会（即Coling '90）会前讲座的主题是："处理大规模真实文本的理论、方法和工具"，明确地提出了计算语言学今后一个时期的战略目标。此外，1992年6月在蒙特利尔举行的第四届机器翻译的理论和方法国际会议（即TMI-92）的主题是："机器翻译中的经验主义和唯理主义方法"，所谓唯理主义是指以生成语言学理论为基础的方法，所谓经验主义则是指以大规模语料库的分析为基础的方法。由此可见，基于语料库的语言研究已经成为计算语言学研究的重点。这一点，新近在日本召开的第五届机器翻译的理论和方法国际会议（即TMI-93）和机器翻译高层会议（MT SUMMIT IV）给予了更大的关注[2][3]。

为了实现"大规模真实文本处理"这一战略目标，各国学者均十分强调语料库的作用。这是因为，从"大规模"和"真实文本"两个角度去观察，语料库是最理想的语言知识资源。然而，要使语料库成为名副其实的语言知识库，就必须对库存语料进行从词法、句法、语义等各种层次上的加工。这一步骤使语料由"生"变"熟"，才能使知识获取成为可能。所以，语料加工的理论、方法和工具是目前学术界的关注焦点[4]。以汉语为例，语料的加工由浅入深，按图1所示的流程进行。

图1中，设立分词、词性、句法、语义等四个层次的标注，语料标注深度依次提高。在标注每一级，理想情况下，均应有一个知识获取模块，该模块从(1)标注模块的人工操作(如遇到歧义时，人工排歧)中，获取知识；(2)从标注过的语料中获取知识[5][6]。所获取的知识又反馈到标注模块，以提高标注的自动化程度和标注的一致性。对汉语来讲，分词已趋向成熟，词性标注也有较大的进展[7]，而对句法层次来讲，无论是人工标注，还是自动标注都刚起步。原因是存在如下问题：其一，没有确立一个语言模型作为句法标注体系，如短语结构文法，还是依存文法或者其它文法，哪一个最适合语料库的标注？其二，如确立了某一语言模型，那么，要确立该模型完整的体系，以作为标注的规范。比如采用短语结构语法，要确立多少种短语类型？如采用依存文法，应确立多少种依存关系？其三，作为标注工具，为不断提高标注的自动化程度和标注的效率，保证标注的一致性，应该具有学习功能，对于这一点，国内外均没有很好解决。作者曾在[8]中论述了依存文法是适合语料库标注的语言模型，并确立了44类汉语的依存关系，对前两个问题做了初步回答，本文试图解决第三个问题。

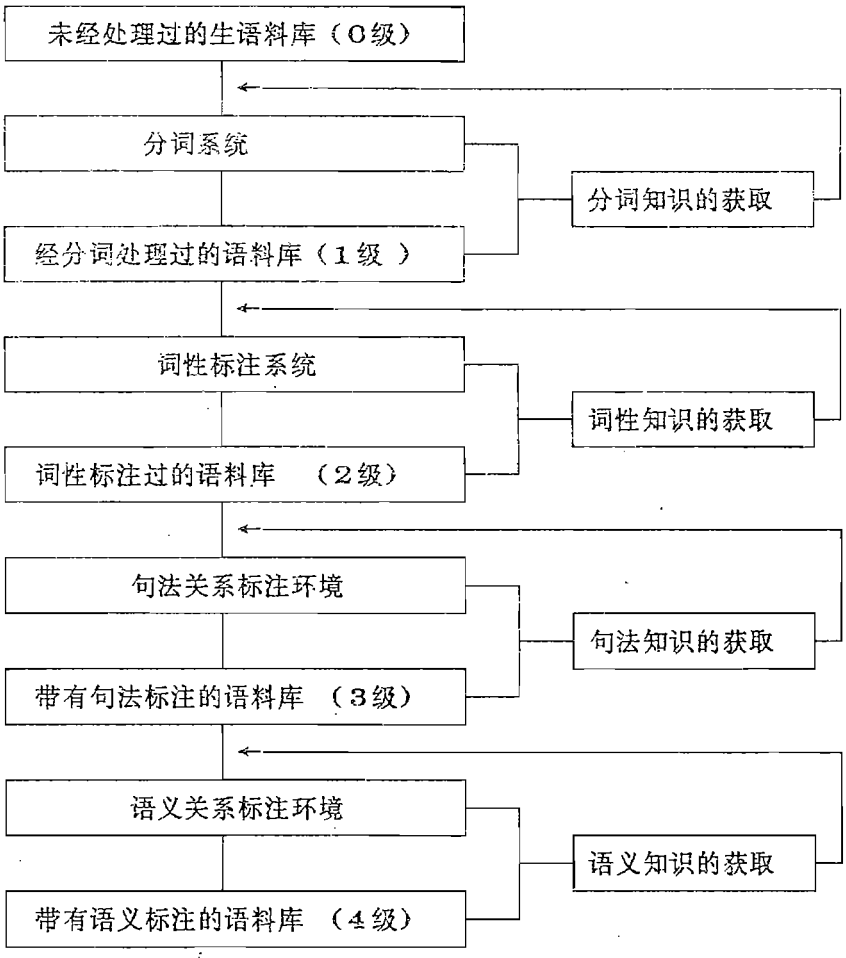

图1 语料库的加工和知识的获取

Robert F. Simmons 和 Yeong- Ho Yu 在研究英语分析中，引入上下文约束规则(Context-Dependent Grammar)，通过交互式示教机制，获取并积累规则，通过基于栈的移进/归约分析器，进行短语结构分析和格分析，成功率99%[9]。我们受其启发，借鉴其分析机制，在依存文法体系下，设计了一个汉语句法标注工具CSTT．它能学习人工标注的思路，在开始阶段，人标注，机器学习，到一定阶段，机器可以帮助人标注，自动化程度愈来愈高，甚至一半以上的操作可由机器自动完成，理论上讲，本标注工具发展到一定程度，也就成为一个依存句法分析器。我们作了1300句的测试，证明非常有效。

2 语料库句法工具CSTT的设计

2．1系统的设计思想

在句法标注过程中获取标注知识，是最为关键问题。我们采用了上下文有关移进/ 归约分析器，对输入句子，进行人工/机器分析，在每一步分析时，记录当前上下文环境， 获取规则。移进/归约分析原理是：每当接收一个句子，首先判断栈和输入串的状态，假设分析未完成，则根据栈顶元素来查寻规则库，这里我们采用上下文有关规则，在匹配时，要检查上下文环境，如果没有相匹配的规则，则由人来判断移进还是归约操作，如是归约操作，还要判断归约为何种语法单位，形成何种依存关系等，机器要记下当前的上下文环境，以及用户的操作，形成一条规则。如此循环直到栈中只剩下中心词，且输入串为空。对于依存标注来讲，由于依存文法强调的是两个词之间的直接关系[8]，所以每次归约操作， 均取栈顶两个元素进行归约，此外除了要说明归约成何种语法单位外，还要说明所归约的两个元素之间的依存关系。所以规则的形式为：

$\alpha\ xy\ \beta \rightarrow s$ （移进时）

$\alpha\ xy\ \beta \rightarrow z \mid \gamma$ （归约时）

其中α、β是栈顶元素x和y的上下文（此处仅以词性代表），z是归约后的语法单位，γ是x与y之间的依存关系。为简单起见，z是x和y中的中心词。在下一节中，我们将讨论引入非终结符号（短语结构）。

首先举一个简单的例子:

我 是 她 的 好 朋友。
r vy r usde a ng。

栈	分隔符	输入串	动作依存关系	
- - - - -	#	R VY R USDE A NG 。	移进	
- - - - R	#	VY R USDE A NG 。	移进	
- - - R VY	#	R USDE A NG 。	归约	SUB
- - - - VY	#	R USDE A NG 。	移进	
- - - VY R	#	USDE A NG 。-	移进	
- - VY R USDE	#	A NG 。- -	归约	DEP
- - - VY USDE	#	A NG 。- -	移进	
- - VY USDE A	#	NG 。- - -	移进	
- VY USDE A NG	#	。- - - -	归约	ATTA
- - VY USDE NG	#	。- - - -	归约	ATTA
- - - VY NG	#	。- - - -	归约	OBJ
- - - - VY	#	。- - - -	移进	
- - - - VY 。	#	- - - - -	归约	MARK
- - - - VY	#	- - - - -	归约	GOV
- - - - GOV	#			

每一步标注，都得到一条规则，说明在当前栈状态和输入串下，要进行的动作(归约 / 移进)，以及归约时形成的依存关系。列举两条规则如下:

```
- - - R VY     # R  USDE A NG 。  B     SUB
- - R VY USDE  # A  NG 。- -      S     -
```

规定当归约时，A表示中心词为栈顶第二个元素，B表示中心词为栈顶第一个元素，归约后，栈中去除非中心词，在栈顶保留中心词。注意，当动作为移进时(动作='S')，关系为空。由于句子长度不定，为避免繁琐，我们取栈中三个元素（不包括栈顶两个元素）和输入串的前五个元素。即只考虑栈顶两词前后共八个词的上下文环境。

2．2基于上下文有关移进/归约分析的标注算法

输入是所给定句子的词性序列，输出是依存关系，以（从词，主词，依存关系）的三元组形式表示，同时获取若干上下文有关规则。

CDG是以前获取的上下文有关的语法规则(系统初启用时为空)。

```
BEGIN
STACK:=EMPTY
DO UNTIL (INPUT=EMPTY AND STACK=(GOV))
  WINDOW_CONTEXT:=APPEND(TOP_FIVE(STACK),FIRST_FIVE(INPUT)) /*取上下文环境*/
    OPERATION:=CONSULT_CDG(WINDOW_CONTEXT,CDG)    /* 到规则库中查规则  */
    IF(OPERATION==NIL                  /* 没有查到规则  */
       OR NUMBER(OPERATION)>=3)        /* 规则冲突      */
       OPERATION=CONSULT_TO_HUMAN()    /* 人机对话      */
    IF FIRST(OPERATION)='S'
    THEN STACK:=PUSH(FIRST(INPUT),STACK)
    ELSE
    IF FIRST(OPERATION)='A'
    THEN BULID_DEPENDNECY_RELATION(FIRST(STACK) , SECOND(STACK)
          ,SECOND(OPERATION))          /* 建立依存关系三元组 */
         STACK:=POP(STACK)            /* 归约为栈顶第二词(中心词) */
    ELSE
    IF FIRST(OPERATION)='B'
```

```
THEN BULID_DEPENDNECY_RELATION(SECOND(STACK) , FIRST(STACK)
        , SECOND(OPERATION))        /* 建立依存关系三元组 */
     STACK:=PUSH(FIRST(STACK), POP(POP(STACK)))
                                    /* 归约为栈顶第一词(中心词) */
ENDDO
```

函数TOP_FIUE, FIRST_FIVE 分别返回栈和输入串的前五个元素，如果不够，则以空格补足。APPEND 将两个表合并，以得到分析器当前状态。函数CONSULT_CDG 查找规则库， 如果存在匹配规则且不冲突，则系统自动完成标注，如果规则不存在或规则冲突，则与用户交互，询问用户下一步的操作。用户确定后，系统继续执行，在此期间进行知识的获取，获取的新规则加入到CDG规则库中。函数PUSH 为压栈操作， POP 为弹出操作。 SECOND 返回表中第二项，FIRST 返回表中第一项。

上下文有关规则数目很大，为便于快速查到规则，我们建立了两级索引。另外匹配规则时我们引用[9]的加权匹配策略：

设全部规则之集合$R=\{R1,R2,..,Rm\}$，其中某一条规则（左部）为$Ri=\{ri1,ri2,..,ri10\}$，设当前分析器的上下文$C=\{c1,c2,..,cn\}$， 设$\mu(cj,rij)$为匹配函数， 当 cj=rij时， 为1， 否则为0。则评价函数为：

$$SCORE=\sum_{i=1}^{3}\mu(ci,ri)\cdot i+\sum_{i=6}^{10}\mu(ci,ri)(11-i)$$

一个规则被选中，当且仅当该规则的SCORE>=某一预先定义的门限值ζ。当ζ=30时，就是完全匹配。如果ζ的值较大，则找到匹配规则少，而ζ过小则找到匹配规则多，造成冲突增加。即系统刚开始的时候，规则较少，宜采用完全匹配（即ζ=30）。当规则数目较大时， 我们在加权匹配分值一致时，使用频度高的规则优先，如果一个规则使用频率极小， 可以将其淘汰，这样规则库始终处于动态管理状态，以保证效率和速度。

3 CSTT的试验及分析

3.1 系统的试验

该系统是在486机上用TURBO C实现的，由以下五个模块组成。

- 输入模块：输入已标注好词性的句子。
- 标注模块：标注的结果以三元组的形式来表示。
- 知识获取模块：它记录分析过程中的新规则，经过整理后存入规则库中。
- 输出模块：把标注的结果以句法树形式显示给用户，在用户确认后将它存入文件中。
- 人机界面：因为系统是人－机共栖方式，为此设计了一个用户有好的界面。

对1300个汉语简单陈述句子（选自日常生活语料）作了试验：（1）词性自动标注[7]。（2）利用CSTT标注。共获取了6521条规则，在试验中， 以100个句子为单位做测试，并对标注的过程作了记录。前300句我们得到了1455条规则，作为系统的初始知识库，具体过程见表1。

句 子	1-3	4	5	6	7	8	9	10	11	12	13
规则数	1455	447	384	455	486	628	565	572	564	483	492
冲突数	39	22	41	37	40	40	32	46	27	40	39
动作数	2072	768	776	792	851	834	846	837	1153	1164	1111
自动	487	291	336	281	317	121	237	210	572	641	580
干预	1546	455	399	474	494	673	587	581	654	483	492
自动完成%比	23.5	37.8	43.3	35.5	37.3	14.5	30.0	25.1	49.6	55.1	52.2

表 1 测试结果表(句子数以百为单位。)

由表1知，当标注到1000句以后自动完成百分比已为50%左右。上述属开放测试，至于封闭测试，即在对1300句获取完规则后，再重做1300句，自动标注率为90%以上。

3．2　问题及分析

（1）规则的冲突问题

我	劝	她	努力	学习。	(句一)
R	VG	R	D	VG	。
我	听见	她	唱	歌	。 (句二)
R	VG	R	VG	NG	。
我	送	礼物	给	她	。 (句三)
R	VG	NG	VG	R	。

以句一为例说明分析过程：首先移进“我”与“劝”，下一步是归约，归约后栈顶是“劝”，然后移进“她”，下一步是归约，归约为'劝'然后又是移进这时栈顶是“劝努力”不能归约，继续移进，栈顶是“努力　学习”应归约，归约为“努力”栈中剩下“劝　学习”，他们的词性是“VG VG'，输入为“。”。其它两个句子分析到最后，栈中也是为“VG VG”，输入为“。”，但是它们的关系却分别为 SOC，OBJ，VA，说明规则发生了冲突。

原因之一：　虽然本系统引入了上下文的信息，但只是词性的信息，　如下几种典型歧义解决不了：

1.　VG　A　NG

处理　好　关系

养成　好　习惯

2.　NG　NG

飞机　大炮　(联合)

木头　桌子　(偏正)

3.　VG　A　USDE NG

喜欢　漂亮　的　人　(述宾)

喜欢　漂亮　的　人　(偏正)

4.　VG　NG　USDE NG

咬死　老鼠　的　猫。

砍断　猎人　的　树。

歧义分为两种，他宥性歧义结构和自宥性歧义结构，对他宥性歧义，本移进－归约分析器能解决一部分歧义：如“咬死猎人的狗”是有歧义的，如果我们考虑了上下文，　即使只考虑词性，也可能消除歧义。如：

咬死　猎人　的　狗　死　了。　　一　只　咬死　猎人　的　狗

VG　NG　USDE　NG　VG　Y　　M　Q　VG　NG　USDE　NG：

本移进-归约器可以解决这类冲突。对需借助语义才能解决的歧义或自宥性歧义，本移进－归约分析器无能为力，须借助人机对话。如：“飞机　大炮”〈-〉“木头　桌子”；“教育　孩子　的　问题”〈-〉“教育　经费　的　问题”。

原因之二：系统归约为依存关系的中心词，虽然简单，但丢失了许多有用的信息。

例如：”我　听见　他　唱　歌。”与“信息　的　处理　已经　完成。”会产生如下的冲突：“听见　唱歌”　两词的词性与　“处理　完成”两词的词性完全相同，但二者的关系却不一样，“听见　唱歌”的关系是OBJ　而“处理　完成”的关系是SUB。所以必须引进中间结点，例如我们在归约把　“我　听见”两词归约为SV（主谓结构），而“信息的处理”归约为NP（名词短语），就会避免冲突。这样一来，规则的形式仍为：

$\alpha\ x\ y\ \beta \rightarrow s$　　（移进时）

$\alpha\ x\ y\ \beta \rightarrow z|\ \gamma$　　（归约时）

其中α、β是栈顶元素x和y的上下文（此处仅以词性代表），z是归约后的语法单位，　此时是表示短语结构的非终结符号，γ是x与y之间的依存关系，这种方法结合了依存文法和短语结构文法的长处，有很大优势。另外，可以实现依存表示与短语结构的转换，这样，用依存表示的语料库可以转换为用短语结构表示的语料库，有助于获取短语结构语法的知识，如概率化短语结构语法的概率数据。

（2）规则是否收敛的问题

[9]估计，对英文来讲，２５，０００条CDG规则，就可以处理大多数的语言现象，由此认为，规则是收敛的。如果我们只想做句法标注，就可以撇开收敛问题，譬如当我们得到２００００条规则时就停止获取，那么标注器已可提供极大的帮助。我们还可考虑对该标注器进行改进，对出现的冲突问题加以克服，使之发展为一个依存句法分析系统。

参考文献

1．黄昌宁,关于大规模真实文本的谈话,第三届中文信息处理国际学术会议论文集，１９９２年１０月，北京。

2．Proceedings of the Conference on Theoretical and Methodological Issues inMachine Translation, July 14-16, 1993, Kyoto, Japan.

3．Proceedings of the MT SUMMIT IV, July 20-22, 1993, Cobe, Japan.

4．黄昌宁，关于处理大规模真实文本的谈话，语言文字应用，１９９３年第二期。

5．周明、黄昌宁、张敏、白栓虎、吴升，统计与规则并举的汉语句法分析模型，计算 机研究与发展，待发表。

６．张敏，语料库，知识获取与汉语依存分析，清华大学计算机系硕士论文，１９９３．３。

7．白栓虎,汉语自动词性标注系统的研究与实现,清华大学计算机系硕士论文,１９９２．３。

8．周明、黄昌宁，面向语料库标注的汉语依存体系的探讨，中文信息学报，待发表。

9．Robert F.Simmons, Yeong-Ho Yu,The Acquisition and Use of Context- Dependent Grammarsfor English , Computational Linguistics, Vol.18,No.4, 1992

附录１　语料库的标注样本

例句：我是他的好同学。

1	我	2	SUB
2	是	0	GOV
3	他	4	DEP
4	的	6	ATTA
5	好	6	ATTA
6	同学	2	OBJ
7	。	2	MARK

附录２　获取的部分规则

usde ng vg　# p t vg 。　# ATTA b

ng vg　# ng p ng vg 。 # COOR a

p ng vg　# cm vg 。　# # s

vg ng vg　# usde r q ng 。 # ATTA b

vh a ng　# usde vg ng 。　# # s

a ng　# p a ng d a # ATTA b

vg ng　# 。　# OBJ a

CSTT: Chinese Syntactic Tagging Tool with Self-Learning Ability

Zhou Ming　Huang Changning　Yang Jundong*

Dept. of Computer Science, Tsinghua University

*Dept. of Electronic Engineering, Yantai University

ABSTRACT

As a strategic target of the global computational linguistic circle, the large scale authentic text processing becomes more and more significant for the modern informative society. To acquire knowledge from the corpus, we must annotate the corpus with part of speech,syntactic relation and semantic relation , since this work is timeconsuming and troublesome,an effective tool which has the ability of self-learning from human's operation is requested. In this paper, a Chinese syntactic tagging tool (CSTT) is described, in accordance with Depepdency Grammar formalism,CSTT uses context- sensitive shift/reduce parsing method, and accumulates the syntactic tagging knowledge during the tagging process, thus greatly improve the efficiency and keeps consistency.1300 Chinese sentences were tested, and satisfactory result was obtained.

Key Words: Corpus,Syntatic tagging, Parsing, Dependency Grammar