汉语多重关系复句的关系层次分析*

鲁松，白硕，李素建，刘群

(中国科学院 计算技术研究所，北京 100080)

E-mail: lusong@ict.ac.cn

http://www.ict.ac.cn

摘要：汉语多重关系复句的句法分析问题主要由复句中的关系分析和层次分析两部分组成.将多重关系复句中的层次分析作为研究对象.它是针对多种逻辑或并列关系，按照一定层次组成复杂主从关系复句而进行的关系层次分析过程.为了有效地形式化地表示多重关系复句的层次结构，提出了关系层次树的概念，并以此为基础构造文法，采用部分数据驱动的确定性移进-归约算法实现多重关系复句的关系层次分析.通过开放测试对计算机实现的多重关系复句句法分析器进行考察，93.56%的正确率使所提出的分析方法的有效性和正确性得到了充分的验证.

关键词：多重关系复句；句法分析；关系层次树；改进的确定性移进-归约算法

中图法分类号：TP18 **文献标识码**：A

复句的分析和处理是诸多自然语言理解应用问题中最常见和最困难的问题之一.其中包括无关联词语逻辑关系复句的关系识别、描述性复句内在关系的判定、各种回指现象的消解及机器翻译中复句相应的英文的生成等问题.

本文的研究对象是汉语多重关系复句的句法分析.它是针对并列关系或多种逻辑关系，按照一定层次组成复杂主从关系复句的关系层次分析和识别过程.尽管已有的工作都是从语言学角度出发，并未做深入的形式化和算法化工作的，但其中的探索具有极大的可借鉴性.

为了直观地描述问题和介绍本文所提出的方法，在此给出汉语多重关系复句的典型例句，见例1[1](为了清楚地描述各分词间的关系和层次地位，例句中添加了各分句的序列号)：

例1：我们的确已经取得了很大的成绩①，但是如果因为有了这些成绩②，就骄傲起来③，认为可以歇一歇脚④，那就不妥当了⑤.

汉语复句的组成成分是分句.按照分句之间连接关系的不同，汉语复句被分为主从关系复句和联合关系复句.主从关系复句是两个分句按照一定的逻辑关系组成的复句，其逻辑关系通常由关联词语指示.联合关系复句是两个或多个分句按照并列关系组成的复句[1].主从关系和联合关系都是组成复句的原子关系.

作为复句中的一种，多重逻辑关系复句的组成成分同样也是分句.这一特点决定了多重逻辑关系复句的句法分析不同于单句的句法分析，必须且仅能依据分句这一基本表示单元来进行.

由于多重关系复句是汉语的议论文中常见的表述方式，在自然语言处理诸多应用问题时，如机器翻译，均需对其进行处理，所以将实现对此类复句关系层次结构的形式化描述确定为本文的研究

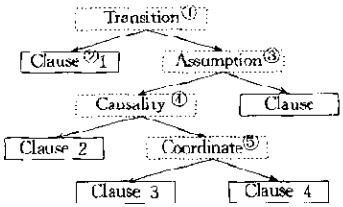
* 收稿日期：2000-09-02；修改日期：2001-03-26

基金项目：国家重点基础研究发展规划973资助项目(G1998030510)

作者简介：鲁松(1972－)，男，北京人，博士，主要研究领域为计算语言学，机器学习，信息检索；白硕(1956－)，男，辽宁沈阳人，研究员，博士生导师，主要研究领域为网络信息处理，计算语言学，人工智能；李素建(1975－)，女，山东荷泽人，博士生，主要研究领域为自然语言处理，机器翻译；刘群(1966－)，男，江苏萍乡人，副研究员，主要研究领域为自然语言处理，机器翻译.

初衷和目标.为了形式化地表示分句之间的连接关系和层次构成,本文将多重关系复句形式化为一棵关系层次树.例1的关系层次树如图1所示.

汉语多重关系复句的分析主要由两部分组成:分句间的构成关系的分析和其中关系层次的分析.构成关系的分析是在关联词语缺省的情况下,依靠相关语义分析来获取分句间逻辑关系的问题.关系层次分析是在构成关系可以确定的情况下构造关系层次树的过程.本文以层次分析为研究对象,不涉及关联词语缺省情况下分句间的构成关系的分析.



①转折关系,②分句,③假设关系,④因果关系,⑤并列关系.

Fig.1 Relation hierarchical tree of example 1
图1 例1的关系层次树

至此,多重关系复句的句法分析任务进一步明确为构建多重关系复句的层次树.针对这一目标,本文给出了一种具有预测机制、自底向上、部分数据驱动的确定性移进-归约关系层次分析算法,并在开放测试中使其理论的正确性和算法的有效性得到了验证.

本文第1节介绍多重关系复句的文法和关联词语的价值及作用.第2节详尽地描述了本文提出的层次分析算法及其实现过程.第3节通过模拟栈和队的工作过程,给出了一个多重关系复句实例的完整分析过程.第4节是对实验过程和实验结果的描述与分析.第5节对本文提出的分析算法进行了讨论.

1 关联词语和多重关系复句的文法*

1.1 多重关系复句中的关联词语

关联词语作为对关系和层次进行指示的重要形式化标记,在多重关系复句的关系和层次判定问题上是唯一和必须依靠的资源.但类似于其他自然语言现象,由于关联词语省略、多个关联词语同处一个分句及其出现位置灵活等特点,决定了依托于关联词语的多重关系复句层次判定不能像数学算式中左、右括号的匹配运算那样整齐划一和易于形式化.

(1) 关联词语的成分归属

尽管关联词语在句法形式上与分句结合紧密,但关联词语不是分句内部的有效语义和语法成分,而是复句中关系组合的有效连接成分.它之所以没有成为必要成分,是因为如果在语境已提供足够多的信息,并能够依靠意合,明确表明分句间关系的情况下,关联词语则可以省略.所以,我们认为关联词语是复句内分句有效组合的黏着点和句法表现,仅受复句的句法的限制.

(2) 关联词语在关系组合中的位置及零关联词语

关联词语在关系组合中的前、后位置为层次分析的提供信息,但部分关联词语存在类似于兼类词语的表现关系组合时的位置歧义.见表1.其中"√"表示该关联词语可以成为此部关联词语;"×"表示该关联词语不能作为此部关联词语.

定义1(零关联词语). 在关联词语省略的情况下,将其定义为零关联词语,标记为∅,尽管缺省,但仍然作为一个客观实体,与其他关联词语等同对待.

---

* 限于篇幅,本文不罗列所有逻辑关系和相关关联词语,相关内容可参见文献[1].

(3) 关联词语与分句间关系的组合

关联词语的搭配与组合关系的一一对应是本文所提出的层次分析算法的关键.关联词语不仅用于指示分句间的连接关系,而且是分析算法所必须依靠的形式化标记和处理时基本的数据操作单元.

Table 1 Position of conjunction

表 1 关联词语可现位置表

Class①	Conjunction$_{first}$②	Conjunction$_{latter}$③	Meaning④	Example⑤
1	✓	×	Only as conjuction$_{first}$⑥	"只要/只有/..."
2	✓	✓	Both as conjuction$_{first}$ and as conjunction$_{latter}$⑦	"因为/如果/..."
3	×	✓	Only as conjuction$_{first}$⑧	"但是/所以/..."

①类别,②关联词语$_{前}$,③关联词语$_{后}$,④含义,⑤举例,⑥仅能做关联词语$_{前}$,⑦做前/后关联词语均可,⑧仅能做关联词语$_{后}$.

1.2 多重关系复句的文法及其特点

依据复句中的关联词语,可以采用上下文无关文法(context free grammars)形式化地表述多重关系复句,其文法 $L(G_1)$如下:

$$
\begin{aligned}
&G_1=(V_N,V_T,S,P),\\
&V_N=\{S,R,R\text{-transition},R\text{-assumption},R\text{-causality},\dots\},\\
&V_T=\{\text{因为},\text{因此},\text{所以},\text{但是},\text{如果},\text{就},\varnothing,\dots\},\\
&S=\{S\},\\
&P=\{S\rightarrow R,\\
&\quad R\rightarrow R\text{-transition}\mid R\text{-assumption}\mid R\text{-causality}\mid\dots\\
&\quad R\text{-causality}\rightarrow\text{causality-fore}+\text{causality-back}\\
&\quad \text{causality-fore}\rightarrow\text{因为}\mid\text{因}\mid\dots\mid R\\
&\quad \text{causality-back}\rightarrow\text{所以}\mid\text{因此}\mid\dots\mid R\\
&\quad \dots\\
&\quad \}.
\end{aligned}
$$

在此文法的基础上可以构建相应的句法分析器.为了把握关系层次分析的关键,进一步简化文法和分析器,本文给出一种基于关联词语前、后部所属的更为简洁的文法 $L(G_2)$,其形式如下:

$$
\begin{aligned}
&G_2=(V'_N,V'_T,S',P'),\\
&V'_N=\{S,R,\text{conjunction-fore},\text{conjunction-back}\},\\
&V'_T=\{\text{因为},\text{因此},\text{所以},\text{但是},\text{如果},\text{就},\varnothing,\dots\},\\
&S'=\{S\},\\
&P'=\{S\rightarrow R\\
&\quad R\rightarrow\text{conjunction-fore}\quad+\quad\text{conjunction-back}\\
&\quad R\rightarrow\text{conjunction-fore}\quad+\quad\varnothing\\
&\quad R\rightarrow\text{conjunction fore}\quad+\quad R\\
&\quad R\rightarrow\varnothing\quad+\quad\text{conjunction-back}\\
&\quad R\rightarrow R\quad+\quad\text{conjunction-back}\\
&\quad R\rightarrow\varnothing\quad+\quad\varnothing\\
&\quad \text{conjunction-fore}\rightarrow\text{因为}\mid\text{只要}\mid\text{只有}\mid\dots\\
&\quad \text{conjunction-back}\rightarrow\text{因为}\mid\text{但是}\mid\text{所以}\mid\dots\\
&\quad \}.
\end{aligned}
$$

文法可以反映出多重关系复句的3个基本特点:逻辑关系组合的二分性、成分或关系组合的完

备性和关系组合的就近匹配性.需要指出的是,尽管多重关系复句基于关联词语存在与否和前、后部所属的构成文法比较简单,但同样存在着严重的移归冲突和归归冲突的歧义情况.

2 层次分析算法及其实现

2.1 算法框架

层次分析算法是基于文法 $L(G_2)$ 构造的,它是一种确定性移进-归约的改进算法[2~5].由于多重关系复句中表示严格的逻辑关系在一定程度上限制了自然语言的灵活性,所以我们选用了部分数据驱动的策略,由此实现无回溯的确定性分析算法.其基本数据结构为栈和队列,待处理数据存放在队列中,工作区为栈,主要操作有移进、归约、拒绝和接受.整个层次的分析过程是以关联词语提取(包括零关联词语)预处理为前提的,针对关联词语进行移进-归约操作过程.

2.2 基本数据结构和语言知识库

在队列和栈的操作过程中,五元组是操作的基本数据单元,其定义形式为

$$I=(RW,F/B,C\_S,M\_S,R).$$

其各个符号的含义如下:

RW——关联词语名称;

F/B —— 该关联词语在当前多重关系复句中的前、后部所属;

C_S——在当前多重关系复句中,该关联词语所处分句的序列号;

M_S——关系归约后生成的关系序列号;

R ——关系归约后生成的关系名称.

其中,在进行入栈操作之前,F/B 必须确定完毕;在有关该关联词语的关系归约之前,M_S-C_S,且 R 为空.

语言知识库中包括了关联词语前、后部所属,搭配组合情况及其构成逻辑关系的详细知识,它是多重关系复句中逻辑关系指示的关键和层次分析的基础.

2.3 确定关联词语的前、后部

确定关联词语的前、后部位置所属时层次分析的前提条件.确定算法的类 Pascal 描述如下:

算法 1. 确定关联词语前、后部所属算法

```
/* 设当前非零关联词语五元组为 cw_i(下标 i 表示当前关联词语的序列号) */
If (关联词 cw_i.RW 既可为关联词语前,也可为关联词语后)
Then  If (cw_i.C_S=1)  /* 该句是第 1 分句 */
        Then cw_i.F/B="关联词语前"
        Else  If (该句是最后一分句)
                Then cw_i.F/B="关联词语后"
                Else  If (cw_{i-1}.C_S=c_i.C_S)
                        Then cw_i.F/B="关联词语承"
                        Else CASE (cw_{i-1}.F/B) of
                             {"关联词语前":cw_i.F/B="关联词语后"
                              default: If (cw_{i-2}.RW=cw_i.RW)
                                         Then cw_i.F/B="关联词语承"
                                         Else cw_i.F/B="关联词语后"
                             }
Else 依据关联词语数据库可得 cw_i.RW 的前、后部归属;
```

2.4 移进-归约表

部分数据驱动分析算法的移进-归约表见表 2. 其中:R 表示关系归约后得到的组合关系,N 表示层次分析器拒绝接受该复句,认为其为非法;∅表示零关联词;reduce(*i*)中,*i* 表示归约的编号,与第 2.5 节中的归约操作程序的编号一一对应. *s*2 表示栈顶第 2 元素.

Table 2 Reduce-Shift table

表 2 移进 归约表

(*q*1) Unit waiting for push① / (*s*1) Top unit in stack②	$Conjunction_{first}$③	$Conjunction_{latter}$④	∅	R
$Conjunction_{first}$	IF (*q*1.*RW*=*s*1.*RW*) and (|*q*1.*C*_*S*−*s*1.*C*_*S*|=1) THEN reduce(9) ELSE shift	IF (*s*1.*RW*()*s*2.*RW*) THEN shift ELSE IF (|queue|=1) THEN reduce(12) ELSE reduce(8)	shift	IF (*q*1.*C*_*S*=*s*1.*C*_*S*) THEN reduce(11) ELSE reduce(7)
$Conjunction_{latter}$	IF (*s*1.*C*_*S*=*q*1.*C*_*S*) THEN shift ELSE reduce(5)	IF (*q*1.*RW* refer purpose relation) THEN shift ELSE reduce(6)	shift	IF (*s*1.*C*_*S*=*q*1.*C*_*S*) THEN reduce(10) ELSE N
∅	reduce(1)	IF (*s*1.*C*_*S*=1) and |stack|=1 THEN shift ELSE IF (*s*2.*F*/*B*=conjuction$_{first}$) THEN reduce(2) ELSE reduce(3)	reduce(4)	N
R	N	shift	N	N

①待入栈元素,②栈顶元素,③关联词语$_{前}$,④关联词语$_{后}$.

2.5 归约操作表

根据归约条件和方式的不同,可将归约分为两种:关系归约和句法归约.

(1) 关系归约

关系归约是按照前、后部关联词语指示的逻辑关系或并列关系进行的前、后分句或已归约为关系的组合,设 *p*1 为后位五元组,*p*2 为前位五元组,*R*(*p*1,*p*2)=相应逻辑关系(logic relation,简称 LR),其运算产生两个旁效:① 生成关系五元组 *s*=("*","*",*p*2.*C*_*S*,*n*,*LR*)*;② 在关系归约记录表(relation reduce table,简称 RRT)中记录 *n*=(*p*1.*M*_*S*,*p*2.*M*_*S*,*LR*).

(2) 句法归约

句法归约是由于关联词语同处一分句而进行的归约. 即 *R*(*p*1,*p*2)中 *p*1.*C*_*S*=*p*2.*C*_*S*;*R*(*p*1,*p*2)没有返回值,仅有旁效:生成关系五元组 *s*−(*p*2.*RW*,*p*2.*F*/*B*,*p*2.*C*_*S*,*p*1.*M*_*S*,*p*1.*R*).

归约操作程序举例如下:

设待入栈元素为 *q*1,栈顶第 1 个元素为 *s*1,栈顶第 2 个元素为 *s*2.

* 其中的 *I*.*M*_*S*=*n*(表示新的关系序列号),*R*=*LR*(归并产生的逻辑关系名称).

reduce(1): IF ($s2.F/B$=前$_{i-2}$)&($s1.F/B$=$\varnothing_{i-1}$)&($q1.F/B$=前$_i$)
Then {$R(s1,s2)$="并列关系";
s 入栈;重新申请 $q1$ 入栈}

reduce(2): IF ($s2.F/B$=前$_{i-2}$)&($s1.F/B$=$\varnothing_{i-1}$)&($q1.F/B$=后$_i$)
Then {if {$s2$ 与 $q1$ 有逻辑关系)
Then {$R(s1,s2)$="并列关系";s 入栈}
Else {$R(s1,s2)$=相应逻辑关系;申请 s 入栈}
重新申请 $q1$ 入栈}

reduce(3): IF ($s2.F/B$=后$_{i-2}$)&($s1.F/B$=$\varnothing_{i-1}$)&($q1.F/B$=后$_i$)
Then {$R(s1,s2)$="并列关系";s 申请入栈;重新申请 $q1$ 入栈}

reduce(4): IF ($s1.F/B$=$\varnothing_{i-1}$)&($q1.F/B$)=$\varnothing_i$)
Then {$R(s1,q1)$="并列关系";s 入栈}

reduce(5): IF ($s2.C\_S$=$i-2$)&($s1.F/B$=后$_{i-1}$)&($q1.F/B$=前$_i$)
Then {$R(s1,s2)$=相应逻辑关系;申请 s 入栈;重新申请 $q1$ 入栈}

reduce(6): IF ($s2.C\_S$=$i-2$)&($s1.F/B$=后$_{i-1}$)&($q1.F/B$=后$_i$)
Then {$R(s1,s2)$=相应逻辑关系;申请 s 入栈;重新申请 $q1$ 入栈}

reduce(7): IF ($s1.F/B$=前$_{i-1}$)&($q1.F/B$=*$_i$)
Then {$R(s1,q1)$=相应逻辑关系;申请 s 入栈}

reduce(8): IF ($s2.C\_S$=$i-2$)&($s1.F/B$=前$_{i-1}$)&($q1.F/B$=后$_i$)
Then If ($s2.RW$=$s1.RW$)
Then {$R(s1,s2)$="并列关系";s 入栈;重新申请 $q1$ 入栈}

reduce(9): IF ($s1.F/B$=前$_{i-1}$)&($q1.F/B$=前$_i$)
Then If $q1.RW$=$s1.RW$
Then {$R(s1,q1)$="并列关系";s 入栈入栈}

reduce(10): IF ($s1.F/B$=后$_i$)&($q1.F/B$=*$_i$)
Then {$R(q1,s1)$句法归约;s 入栈}

reduce(11): IF ($s1.F/B$=前$_i$)&($q1.F/B$=*$_i$)
Then {$R(q1,s1)$句法归约;s 入栈}

reduce(12): IF ($s1.F/B$=前$_i$)&($q1.F/B$=后$_{i+1}$)
Then {$R(s1,q1)$=相应逻辑关系;申请 s 入栈}

reduce(13): IF ($s2.F/B$=前$_i$)&($s1.F/B$=后$_{i+1}$)& 队列空
Then {$R(s2,s1)$=相应逻辑关系;s 入栈}

reduce(14): IF ($s1.F/B$=$\varnothing_i$)&($s1.F/B$=后$_{i+1}$)& 队列空
Then {$R(s2,s1)$=相应逻辑关系;s 入栈}

reduce(15): IF ($s2.F/B$=前$_i$)&($s1.F/B$=$\varnothing_{i+1}$)& 队列空
Then {$R(s2,s1)$-相应逻辑关系;s 入栈}

关于上述归纳操作中的符号说明如下:

(1) s 被标记为 $R(x,y)$后,归约旁效生成的五元组;

(2)"前"、"后"和"∅"分别表示为该关联词语是关联词语$_{前}$、关联词语$_{后}$和零关联词语.

(3) 表中的空位表示该位置关联词语(包括零关联词语)存在与否任意,但若存在,则必须符合前、后部所属逻辑.

(4) 下标 $i,i-1,i-2$ 是关联词语的所在分句序列号,其中针对关系归约结果"*"表示前部关联词语所在分句.

3 分析过程举例

层次分析器有以下几个操作分析步骤:

(1) 关联词语的提取及其信息标注;

(2) 关联词语的前、后部所属的判定(依据算法 1);

(3) 层次的移进-归约操作(依据第 2 节的表 2 和归约操作程序).

以例 1 为输入的移进-归约分析全过程见表 3.

Table 3 Action of shift-reduce

表 3 移进-归约操作

Step①	Stack②	Action③	Queue④	Explanation⑤
0	#		[∅$_1$][但是$_{后2}$][如果$_{前2}$][因为$_{前2}$][就$_{后3}$][∅$_4$][就$_{后5}$]#	
1	#[∅$_1$]	shift	[但是$_{后2}$][如果$_{前2}$][因为$_{前2}$][就$_{后3}$][∅$_4$][就$_{后5}$]#	
2	#[∅$_1$][但是$_{后2}$]	shift	[如果$_{前2}$][因为$_{前2}$][就$_{后3}$][∅$_4$][就$_{后5}$]#	
3	#[∅$_1$][但是$_{后2}$][如果$_{前2}$]	shift	[因为$_{前2}$][就$_{后3}$][∅$_4$][就$_{后5}$]#	
4	#[∅$_1$][但是$_{后2}$][如果$_{前2}$][因为$_{前2}$]	shift	[就$_{后3}$][∅$_4$][就$_{后5}$]#	
5	#[∅$_1$][但是$_{后2}$][如果$_{前2}$][因为$_{前2}$][就$_{后3}$]	shift	[∅$_4$][就$_{后5}$]#	
6	#[∅$_1$][但是$_{后2}$][如果$_{前2}$][因为$_{前2}$][就$_{后3}$][∅$_4$]	shift	[就$_{后5}$]#	
7	#[∅$_1$][但是$_{后2}$][如果$_{前2}$][因为$_{前2}$]	reduce(3)	[*$_3$][就$_{后5}$]#	Coordinate⑥
8	#[∅$_1$][但是$_{后2}$][如果$_{前2}$]	reduce(7)	[*$_2$][就$_{后5}$]#	Causality⑦
9	#[∅$_1$][但是$_{后2}$][如果$_{前2}$]	reduce(11)	[就$_{后5}$]#	Syntactic reduction⑧
10	#[∅$_1$][但是$_{后2}$]	reduce(12)	[*$_2$]#	Assumption⑨
11	#[∅$_1$][但是$_{后2}$]	reduce(10)	#	Syntatic reduction
12	#[∅$_1$][但是$_{后2}$]	reduce(14)	#	Transition⑩
13	#[*$_1$]		#	The end of parsing⑪

①步骤,②栈,③操作,④队列,⑤说明,⑥并列关系,⑦因果关系,⑧句法归约,⑨假设关系,⑩转折关系,⑪分析结束.

对表 3 中数据表示的说明,关联词语数据单元的 BNF 如下:

〈关联词语数据单元〉::=[〈关联词语〉{〈前、后部所属〉}〈所在分句号〉]

〈关联词语〉::=因为|但是|...|即使|就|∅|*

〈前、后部所属〉::=前|后

〈所在分句号〉::=1|2|...|5|6

由表 3 的分析过程正好可以得到如图 1 所示的关系层次树.

4 实验、结果及其分析

为了进一步验证本文关系层次分析算法的正确性,我们依据文献[1]构造了包括因果、假设、条

件、推论、转折、让步、目的、并列、选择和递进这 10 种关系,有 218 个常见关联词语的关联词语知识库.在语料中抽取共计 328 句存在多重关系的复杂复句进行开放测试,每个复句的平均分句数为 6.31,平均组合关系数为 4.96,实验结果见表 4.

Table 4 Data of corpus

表 4 语料数据

Number①	Number of corpus②	Number of clauses③	Average number of clauses④
1	Seleted works of Mao Ze-dong, Vol.I,II,III⑤	162	7.35
2	The People's Daily (editorial edition), 96'&97'⑥	95	5.02
3	Guang Ming Daily, 96'⑦	71	5.65

①编号,②语料名称,③复句数量,④平均分句数,⑤《毛泽东选集》第一、二、三卷,
⑥《人民日报》1996 和 1997 年社论版,⑦1996 年《光明日报》.

不考虑局部效果,仅以整个复句关系层次分析的正误作为衡量标准,上述实验达到了 93.56% 的正确率.造成错误的原因来自两方面:一方面是测试集合中部分关联词语在语言知识库中未被登录,造成错判为零关联词所致;另一方面是少数多重关系复句中出现关联词语缺省却存在逻辑关系的复杂情况,这已超出本文的研究范围.

5 结 论

本文通过关系层次树的形式化描述,将经典单句分析的移进-归约算法借鉴到汉语多重关系复句的关系层次分析里,并在开放测试中取得了满意的效果.因为未见针对这一问题的研究文献,所以无法做针对性比较.但仅从正确率来看,本文提出的关系层次分析算法为汉语多重关系复句的句法分析提供了一种有效的处理算法.

在手工规则劳动量巨大以及自然语言知识表示困难,难以胜任诸多自然语言处理问题的情况下,针对多重关系复句分析问题,本文提出的分析方法之所以能够取得满意的结果,我们认为有以下原因:(1) 多重关系复句中有限的关系数量是手工规则能够胜任的关键;(2) 关联词语虽然在复句中表现灵活,但作为明显的形式化标记,在分析中起到了极为重要的作用;(3) 多重关系复句所要表达的严格的逻辑关系在一定程度上限制了它在自然语言表现形式上的灵活性,为分析的形式化提供了便利.

需要指出的是,本文的研究焦点问题是多重关系复句的关系层次分析.整个分析算法的形式化基础是关联词语.尽管没有任何关联词语指示的多重关系复句仅占多重关系复句总数的 0.74%,但在仅有两个分句构成的主从关系复句中,此种现象数量众多,特别是在口语理解中更是如此.因此,无关联词指示的逻辑关系判定将是我们的下一个研究目标.

References:

[1] Wu, Jing-cun, Hou, Xue-chao. Parsing Modern Chinese. Beijing: Beijing University Press, 1988 (in Chinese).

[2] Aho, A., Ullman, J. Principles of Compiler Design. Reading, MA: Addison-Wesley Publishing Company, 1977.

[3] Aho, A., Sethi, R., Ullman, J. Compilers: Principles, Techniques and Tools. Reading, MA: Addsion-Wesley Publishing Company, 1986.

[4] Allen, J.F. Natural Language Understanding. Rewood, CA: Benjamin/Cummings Publishing Company, Inc., 1995.

[5] Marcus, M. A Theory of Syntactic Recognition for Natural Language. Cambridge, MA: MIT Press, 1980.

[6] Weng, Fu-liang, Wang, Ye-yi. Introduction to Computational Linguistics. The Chinese Academy of Social Sciences Press, 1998 (in Chinese).

附中文参考文献:
[1] 吴竞存,侯学超.现代汉语句法分析.北京:北京大学出版社,1988.
[6] 翁富良,王野翊.计算语言学导论.北京:中国社会科学院出版社,1998.

Parsing the Logical Embedded Complex Sentences in Chinese *

LU Song, BAI Shuo, LI Su-jian, LIU Qun

(*Institute of Computing Technology, The Chinese Academy of Sciences, Beijing* 100080, *China*)
E-mail: lusong@ict.ac.cn
http://www.ict.ac.cn

Abstract: Parsing the logical embedded complex sentence in Chinese consists of recognition of both logical relation and logical level. Recognizing of logical level, which is to parse the level in this kind of sentences with lots of logical relations, is the aim in this paper. To formalize effectively the logical level in those complex sentences, in this paper, the concept of *relation hierarchical tree* is defined, and in view of this concept, the grammar is designed to construct the parser for this kind of complex sentences with shift-reduce algorithm improved. The parser in this paper is evaluated in open test. The accuracy, 93.56%, shows both the effectiveness and correctness of the theory in this paper.

Key words: logical embedded complex sentence; parse; hierarchical tree; improved algorithm of shift-reduce parser

* Received September 2, 2000; accepted March 26, 2001
Supported by the National Grand Fundamental Research 973 Program of China under Grant No. G1998030510