基于扩充ＬＲ分析的数据库汉语接口的句法分析

王力红

（云南工业大学　信息与电子工程学院　　昆明　650051）

摘　要：本文提出了一种基于扩充的LR分析技术的低限制性句法分析方法。研究表明，该方法在不改变原分析矩阵规模的前提下，较大地扩充了可处理的语言集合，提高了对句子的容错能力，降低了句法限制。

关键词：数据库，自然语言理解，句法分析，LR分析，汉语接口

A method of syntax analysis based on an improved LR technique for Chinese interface to database

Wang LiHong

(Sch. of Information & Electronic Eng. , Yunnan Poly. Univ. , Kunming, 650051)

Abstract: In this paper, a low-limitative syntax analysis method based on improved LR technique is proposed. It is showed by the study that this method can extented the analysed language set largely, improve fault -tolerance to sentences and reduce the limit to syntax in the condition of the same matrix scale.

Key words: Database, natural language understand, syntax analysis, LR technique, Chinese interface

一、前言

自然语言理解是人工智能的一个重要分支，而汉语接口是自然语言理解的重要应用。数据库系统汉语接口就是将用户输入（或语音识别后输出）的汉语句子转换为数据库系统上可执行的操作语言，其中语言理解是接口的基础工作。通常自然语言理解包括词法分析，句法分析，语义语用分析三个部份，其中句法分析是语言理解的核心，它直接影响到语义的生成和对用户的限制程度。因此设计一个句法限制少，处理成本低，易于语义生成的句法分析机制，对自然语言接口是十分必要的。

二、基本思想

1965年由D. Knuth提出的LR(K) 分析法是一种自底向上的无回溯分析法，其处

理成本低（时间复杂度为O(0)），易于实现且便于语义语用分析。但是由于它只适合于上下文无关文法的一个子集（自然语言至少是上下文相关的），而且它的实现要建立包含全部终结符和非终结符在内的分析矩阵，即使每一终结符可以代表某一类的词汇，自然语言复杂的句法关系也会使分析矩阵过于庞大而占用太多的存储空间和降低分析效率，因而LR分析在处理大范围自然语言理解时受到了限制。

数据库系统汉语接口涉及到受限领域的自然语言理解问题，句型相对简单（如只有祈使句，疑问句和省略句等），是自然语言的一个子集，可近似用LR文法描述。但汉语句型的千变万化，使即使在受限领域内，也很难用一个文法将所有句法概括完毕，况且为某些偏僻句型建立专门的规则也是不经济的。于是本文提出一种基于扩充LR分析的低限制句法分析方法，即通过对高效分析矩阵（即状态少而句型覆盖面广的LR分析矩阵）进行扩充改造，实现用较简单的分析矩阵进行较大范围的句法分析的目的。实验证明，这一方法是可行的，LR分析矩阵的大量“空白项”为这种扩充提供了可能。

三、基本句型的高效分析矩阵

如前所述，实际的数据库汉语接口的常见句型可概括为祈使句，疑问句，相应的省略句等基本句型，如：

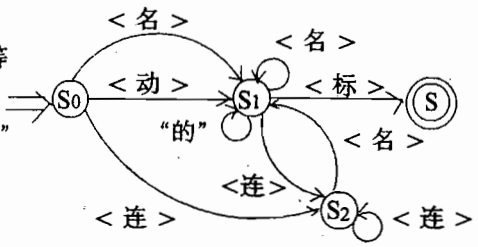
打印贵单位所有工程师的名字。　　(祈使句)

平均成绩大于90分的学生有多少?　　(疑问句)

大于80分的呢?　　(省略句)

以及在基本句型上添加其它成份的复合句型，如：请你告诉我 . . . ，请问 . . . ，等。句中“请你”，“请问”对语义来说是多余的。

一个描述基本句型的最小有限自动机如附图。图中“＜＞”表某一类终结符，“的”代表汉字“的”。具体地，＜动＞代表动词，＜标＞代表标点符号，如“?”、“。”等，＜连＞代表“且”、“和”、“与”、“为”、“在”、“是”、“大于”、“有”等连词或具有连词功能的词，而＜名＞除代表名词外，还代表形容词，带量词单位的数词以及“哪些”、“什么”等连词助词，“呢”、“吗”等疑问词。



附图、描述基本句型的最小有限自动机

由附图的自动机可以构造相应的左线性方法(3.1)[1]：

(3.1)

(1) S → S1＜标＞　(6) S1 → S2＜名＞

(2) S1 → ＜动＞　(7) S2 → ＜连＞

(3) S1 → ＜名＞　(8) S2 → S1＜连＞

(4) S1 → S1＜名＞　(9) S2 → S2＜连＞

(5) S1 → S1“的”

根据文法(3.1)，可以作出基本句型的高效 L R 分析矩阵[1]，如表一。表中S_i (i=0，1，…，11)表状态，而GOTO表中的S_1，S_2表文法中的非终结符。表二给出了基于该表的对基本句型进行句法分析的例子。

终结符 / 状态	ACTION						GOTO	
	<动>	<名>	<连>	<标>	'的'	#	S_1	S_2
S_0	S_2	S_3	S_5				1	4
S_1		S_7	S_9	S_6	S_8			
S_2		r_2	r_2	r_2	r_2			
S_3		r_3	r_3	r_3	r_3			
S_4		S_{10}	S_{11}					
S_5		r_7	r_7					
S_6						acc		
S_7		r_4	r_4	r_4	r_4			
S_8		r_5	r_5	r_5	r_5			
S_9		r_8	r_8					
S_{10}		r_6	r_6	r_6	r_6			
S_{11}		r_9	r_9					

表一、基本句型的高效LR分析矩阵

例句：打印 10 月份 的 资金平衡表 。 ✓

分词后：<动><名><名>'的' <名> <标>#

状态栈	分析栈	待分析句子	动作	归约产生式
S_0	#	<动><名>…<标>#	进	
S_0S_2	#<动>	<名><名>…<标>#	归	(2)S_1→<动>
S_0S_1	#S_1	同 上	进	
$S_0S_1S_7$	#S_1<名>	<名>'的'…<标>#	归	(4)S_1→S_1<名>
S_0S_1	#S_1	同 上	进	
$S_0S_1S_7$	#S_1<名>	'的'<名><标>#	归	(4)S_1→S_1<名>
$S_0S_1S_7$	#S_1<名>	'的'<名><标>#	归	(4)S_1→S_1<名>
$S_0S_1S_8$	#S_1'的'	<名><标>#	归	(5)S_1→S_1'的'
S_0S_1	#S_1	同 上	进	
$S_0S_1S_7$	#S_1<名>	<标>#	归	(4)S_1→S_1<名>
S_0S_1	#S_1	同 上	进	
$S_0S_1S_6$	#<标>	#	'acc'	分析成功

表二、基于表一的句法分析举例

四、高效 L R 分析矩阵的扩充改造

由于受 L R 分析的固有局限性的制约，表一的分析矩阵存在以下问题：

(1)受文法(3.1)的制约，只能处理正则语言，句型及其变换较少；

(2)容错能力差，没有部份匹配能力；

(3)不考虑上下文。

仅仅通过扩充文法规则，显然不能解决全部问题，而且还将付出相当的存储和处理效率的代价。如何在不增加分析矩阵规模的前提下，最大限度地解决这一问题，是问题的关键所在。

从表一存在着的大量的空白项可看出，基于该表的句法分析存在着众多的“错误出口”，而这些“错误”仅仅是由于句子不符合分析表所依赖的文法。由前所述，汉语复杂的句法关系，使任何文法都不可能概括所有的汉语句子，因此‘错误出口’中一定含有事实上正确的句子，而且考虑到人类理解自然语言时，可以不受某些语法错误的影响，故对某些确有错误的句子，只要不影响语义，仍应予以“接收”。此外，某些语法上不完全的句子，如果考虑其上下文，语义也是完整的。

基于以上的考虑，本文主要针对 L R 分析矩阵的“错误出口”，同时兼顾对原文法(3.1)的完善，进行了以下改造：

(1)在表一的 ACTION 表中增加一列<其它>。<其它>代表除表一所列<动>、<名>、<连>、<标>、“的”、“#”以外的其它类的终结符，如副词等；

(2)对于仅含有移进项和空白项的行（即状态S_0，S_1，S_4），表明目前栈顶已没有可归约项，需移进下一词。移进项表明下一词正确，而“空白项”表明下一词不符合文法(3.1)。由前面的讨论，可将ACTION[S_i，X]=“空”(X≠“#”)的项置为“T”，“T”表示将指向下一词的指针再移后一词（即跳过或剔除出错的词）；

(3)对于仅含有归约项和空白项的行（即状态S_2，S_3，S_5，S_7～S_{11}），表明句柄已在分析栈栈顶形成，故“空白项”也可置为相应的归约产生式序号r_j(j=2，3，7，4，5，8，6，9)，而把对下一词的分析留到归约后的新状态行处理（此部份同 L R(0) 分析表对归约态的处理）；

(4)对于仅含“接收”项和空白项的行（即状态S_6），表明某一带标点的句子已满足文法(3.1)，根据标点后的下一词的不同，需作如下处理：

(a)置 ACTION[S_6，“#”]=“$Tacc_1$”。标点后的词为“#”，按理应“接收”，但考虑到附图的自动机在最小化时，降低了对某些错误的检测能力（如不能检测出缺宾语的句子：“打印数学成绩大于等于90分。”）。为弥补这一缺陷，将原“acc”态置为“$Tacc_1$”（即准接收态），表明首先应检查句子的完整性（即看其形式语是否完整），(a.1)若“完整”，则“接收”；(a.2)若不完整，则检查该句之前的句子（若有的话）是否完整，能否弥补本句所缺成份，若能，则此句作省略句处理；(a.3)否则，则“出错”。

即：

	“acc”	(句子完整)
$Tacc_1$	补足成份并“acc”	(省略句)
	“error”	(其它)

这样，扩充后的LR分析可处理某些具有上下文相关性的语句；

(b)置 ACTION[S_6，X]=“$Tacc_2$”(X≠“#”)。标点后还有其它词，不符合原文法。“$Tacc_2$”亦为“准接收”态，表示需先检查标点及其前面的句子的完整性，(b.1)若完整，则“接收”（即“acc”）标点以前的句子，而将标点后的词作为另一省略句处理；(b.2)若不完整，则跳过该标点，分析下一词，即弹出分析栈和状态栈的栈顶<标>和S_6（即视该标点在语法上为多余，语义上为.AND.）。

即：　$Tacc_2$　　“acc”　　（标点前的句子完整）

　　剔除标点　　（其它）

(5)对于仅含移进项和空白项的行，即状态S_0，S_1，S_4，剩余空白项的处理：

(a)仍置 ACTION[S_0，“#”]=“空”。句子以“#”开头，显然“出错”；

(b)置ACTION[S_1，“#”]=“$Tacc_3$”，ACTION[S_4，“#”]=“$Tacc_3$”。句子尚不完全，下一词却是“#”（即“回车”，表句子结束）。“$Tacc_3$”为“准接收”态，指示需检查此句之前的句子（若有的话）能否补上本句所缺成份，若能，此句作省略句处理；否则，则出错。

终结符 / 状态	ACTION							GOTO	
	<动>	<名>	<连>	<标>	'的'	#	<其它>	S_1	S_4
S_0	S_2	S_3	S_5	T	T		T	1	4
S_1	T	S_7	S_9	S_8	S_8	$Tacc_3$	T		
S_2	r_2	r_2	r_2	r_2	r_2	r_2	r_2		
S_3	r_3	r_3	r_3	r_3	r_3	r_3	r_3		
S_4	T	S_{10}	S_{11}	T	T	$Tacc_3$	T		
S_5	r_7	r_7	r_7	r_7	r_7	r_7	r_7		
S_6	——	——	$Tacc_2$	——	——	$Tacc_1$	$Tacc_2$		
S_7	r_4	r_4	r_4	r_4	r_4	r_4	r_4		
S_8	r_5	r_5	r_5	r_5	r_5	r_5	r_5		
S_9	r_8	r_8	r_8	r_8	r_8	r_8	r_8		
S_{10}	r_6	r_6	r_6	r_6	r_6	r_6	r_6		
S_{11}	r_9	r_9	r_9	r_9	r_9	r_9	r_9		

表三、改造后的高效LR分析矩阵

即：　$Tacc_3$　　补足成份并“acc”　　（省略句）

　　“error”　　（其它）

至此，对表一的LR分析矩阵改造完毕。表三列出了改造后的高效 LR分析矩阵，表四给出了基于该表的句法分析的例子。

五、结论

本文通过对基本句型的高效LR分析矩阵的扩充改造，得到一个可进行低限制句法分析的扩充LR分析矩阵。该分析矩阵较原分析表有以下改进：

(1)在不增加原分析矩阵规模的前提下，较大地扩充了可处理的语言集合，提高了对句

子的容错能力，降低了句型限制，并具有部份匹配能力；

⑵具有分析与上文相匹配的省略句的能力（如果稍加修改，同样可具有分析与下文相匹配的省略句的能力），使LR分析的适用范围扩大至上下文相关语言的一个子集；

⑶通过与语义分析的结合，在不增加状态的前提下，提高了分辨力，并改善了自动机最小化时带来的某些缺陷。

该方法不仅适用于数据库系统自然语言接口的句法分析；对其它领域的自然语言理解也有一定启发作用。

例句：　请　显示　平均分数　高于　90分　，　职务　为　班委　的　学生名单　。　✓

分词后：<其它> <动> <名> <连> <名> <标> <名> <连> <名>'的' <名> <标> #

状态栈	分析栈	待分析句子	动作	归约产生式
S_0	#	<其它><动>……<名><标>#	T	
S_0	#	<动><名>……<名><标>#	进	
S_0S_2	#<动>	<名><连><名>……<名><标>#	归	(2)S_1→<动>
S_0S_1	#S_1	同　上	进	
$S_0S_1S_7$	#S_1<名>	<连><名>……<名><标>#	归	(4)S_1→S_1<名>
S_0S_1	#S_1	同　上	进	
$S_0S_1S_9$	#S_1<连>	<名><标>……<名><标>#	归	(8)S_2→S_1<连>
S_0S_4	#S_2	同　上	进	
$S_0S_4S_{10}$	#S_2<名>	<标><名>……<名><标>#	归	(6)S_1→S_2<名>
S_0S_1	#S_1	同　上	进	
$S_0S_1S_6$	#S_1<标>	<名><连>……<名><标>#	$Tacc_2$	句子不完整，弹出
S_0S_1	#S_1	同　上	进	
$S_0S_1S_7$	#S_1<名>	<连><名>……<名><标>#	归	(4)S_1→S_1<名>
S_0S_1	#S_1	同　上	进	
$S_0S_1S_9$	#S_1<连>	<名>'的'<名><标>#	归	(8)S_2→S_1<连>
S_0S_4	#S_2	同　上	进	
$S_0S_4S_{10}$	#S_2<名>	'的'<名><标>#	归	(6)S_1→S_2<名>
S_0S_1	#S_1	同　上	进	
$S_0S_1S_8$	#S_1'的'	<名><标>#	归	(5)S_1→S_1'的'
S_0S_1	#S_1	同　上	进	
$S_0S_1S_7$	#S_1<名>	<标>#	归	(4)S_1→S_1<名>
S_0S_1	#S_1	同　上	进	
$S_0S_1S_6$	#S_1<标>	#	$Tacc_1$	句子完整，接收

表四、基于表三的句法分析举例

参考文献

[1]. 蒋立源，编译原理，西北工业大学出版社，1994年，P46-49，P135-143.

[2]. 管纪文，黄善祥，自然语言接口的设计方法评述，计算机科学，1988年1月.

[3]. Charniak E，McDermott D. Introduction to artificial intelligence Addison -Wesley Publishing Company，1985年，P194-244.　（其它略）