自然语言处理技术基础

王小捷　常宝宝　编著

北京邮电大学出版社

·北京·

内 容 简 介

本书包括了三个方面的内容。第一部分介绍基于规则的自然语言处理技术,分别从语法和语义两个层面入手。首先介绍了几种语法系统的形式化表示方案,在此基础上,介绍了几种典型的上下文无关句法分析和基于复杂特征的句法分析方法。在语义层面,分别从词义和句义两个层次介绍了进行词义和句义分析的方法。第二部分介绍基于统计的自然语言处理技术,包括词汇层的一些统计语言模型以及在句法层的概率上下文无关语法。第三部分介绍一种重要的应用——机器翻译,分别从规则和统计两个方面来介绍它的理论和实现。

图书在版编目(CIP)数据

自然语言处理技术基础/王小捷等编著.—北京:北京邮电大学出版社,2001.6

ISBN 7-5635-0527-X

Ⅰ.自… Ⅱ.王… Ⅲ.自然语言处理 Ⅳ.TP391

中国版本图书馆 CIP 数据核字(2001)第 084948 号

出 版 者:北京邮电大学出版社(北京市海淀区西土城路 10 号)

邮编:100876 电话:62282185(发行部) 62283578(传真)

网址:http://www.buptpress.com

经 销:各地新华书店

印 刷:北京源海印刷厂

印 数:4 000 册

开 本:787 mm×1 092 mm 1/16

印 张:9.75

字 数:229 千字

版 次:2002 年 12 月第 1 版 2002 年 12 月第 1 次印刷

书 号:ISBN 7-5635-0527-X/TP·54

定 价:19.00 元

目　　录

序　　言

自然语言处理:自然的人机交互

随着计算技术的飞速发展,计算机已成为辅助人类认识和改造世界最为强大的工具之一,自出现那一天起至今,帮助人类完成了许多自身难以完成的工作,使人类社会在这一段时期里获得了比以往任何时期都要快的发展。相信在可以预见的未来,计算机对人类发展的重要辅助作用还将持续。

为了让计算机能完成人类所赋予的各项任务,一个首要的问题就是人和计算机的通信问题,即如何把人类希望计算机完成的任务告诉计算机,以及计算机在完成任务后又如何把结果告诉人们。

人机通信经过了几个时期,编写二进制代码、汇编代码、高级语言、第四代语言,人类为了与计算机进行通信,创造了一系列人工语言。为了和计算机进行通信,人类付出了许多的努力。在人类使用工具的历史长河中,人类还从来没有为了和自己创造的工具进行交流而如此屈尊过;如此为了使用这种工具而使自己向这种工具靠拢。人机的矛盾、人因为工具而产生的异化在这里表现得十分突出。一些哲学家早就注意到这个问题,提出了哲学和社会学上的解决方案。

但是,也可以看出,所有这些不断发展新人工语言的努力,正在让人类在使用计算机时离计算机远一些,而离人类本身更近一些。然而,我们知道,人类表达自己思想最方便、最自然的方式是利用人类自身的语言——各种自然语言;人与人之间交流观点、传播消息最方便、最自然的方式也是利用自然语言。因此,最自然的人机通信不应该是任何人工语言,而应该是自然语言。

要使计算机与人能通过自然语言进行通信,就要使计算机能够理解和运用自然语言。早在计算机发明不久,人们就开始了这个方面的尝试,自然语言处理[①]技术就是几十年来人们在这个方向不断努力的产物。

从某种意义上来说,自然语言处理技术提供了一个解决人机异化问题的技术上的解决方案:计算机直接处理自然语言,无需人去适应机器。这将是一个更自然、消除了异化的人机环境,计算机将能帮助人类完成更多的工作。

为了让计算机能很好地进行自然语言处理,一个有益的工作是考察人类的自然语言运用方式,虽然计算机进行自然语言处理的方式很可能与人类不同,但是,毕竟到目前为止,人类的自然语言运用是自然语言处理的唯一原型。遗憾的是,迄今为止,人类对自身运用自然语言的机制还不甚了解,更多的研究还集中在外在的语言本身上。

---

① 自然语言处理也称为计算语言学,它们所指的是同一个研究领域,只是在使用时稍有不同。通常的使用习惯是,在偏重本研究领域的理论时,使用计算语言学这一术语;而偏重于本研究领域的应用方面时,常使用自然语言处理。

语言学:经验材料和理性规则

最简单地说,人类对于自身所使用的自然语言的研究称为语言学。这种研究从很早以来就一直没有终止过,通常分为几个交错的阶段。

最早的研究是由希腊人创立的所谓‘语法”,并在法国人波尔·洛瓦雅尔的‘唯理普遍语法”中得到了显著的体现。其特征是以逻辑为基础,制订出一些规则,用以区别正确的语言形式和非正确的语言形式。其对于语言材料本身缺乏科学的观察。

其后出现了语文学,其首要任务是确定、解释和评注各种文字的文献,通过比较不同时代的文献,确定每个作家的特殊语言,解读和说明用某种古代的或晦涩难懂的语文写出的碑铭。

随后,人们发现不仅可以进行这种比较,还可以进行不同语种间的比较,用一种语言阐明另一种语言,用一种语言的形式解释另一种语言的形式。这就是语言学的第三个阶段——历史比较语言学。

在这样一些研究的基础上,德·索绪尔建立了普通语言学,以此为界,标志着现代语言学的开始。在索绪尔那里,语言的研究重心转向共时语言学,研究语言体系的内部结构。这成为了结构主义语言研究的开始。在结构主义语言学派中,美国的描写派是最有影响的流派之一,他们注重记录实际语言,注重语言中各种单位的分布,基于分布信息的基础上对语言各单位进行切分、归并分类和组合。这时的语言学研究重视语言材料,具有很强的经验主义色彩。其主要原因是,美国语言学家十分强烈地受到一种需求的影响,这就是要把多达几百种以往没有文字记载的北美语言尽可能多地描写出来。最初的代表人物是弗朗兹·博厄斯(1858～1942 年),他认为每一种语言都有其独特的语法结构,语言学家的任务就是要为每一种语言找到适合于该语言的描写范畴。其后,从 1924 年美国语言学会成立到第二次世界大战开始这段时间内的重要代表人物之一是伦纳德·布隆菲尔德(1887～1949年),他明确采用行为主义作为语言描写的框架。为了按照他所理解的‘科学性”来描写语言,他认为应排除一切不能直接观察到的、也不能进行物理测量的素材,因此,语义的研究并不属于正规的语言学研究范围。这些观点,直到 20 世纪 60 年代,由美国后布隆菲尔德学派的结构主义语言学家齐格律·哈里斯(Zellig Harris)进一步继承。

20 世纪 50 年代中后期,诺姆·乔姆斯基(Noam Chomsky,1928～)提出了转换生成语法,他秉承波尔·洛瓦雅尔‘唯理普遍语法”的衣钵,重新确立了理性主义在语言研究中的地位。他认为:语言描写和分析的目的不在于分类,而在于建立一种理论,研究人的语言生成能力,即怎样用有限的成分和规则生成无限的句子,其目标是提出一个能产生所有句子的语法系统。他认为:人存在着先天语言能力,语言的结构是由人类的心理结构决定的,而语言的某些特征所具有的普遍性也证明了人类天性的这一部分为全体成员所共有,不论其种族或阶级如何,也不论其智力、性格和体质方面所显然具有的区别。乔姆斯基的理性主义观点曾经在语言学研究中占据着主导地位,时至今日,依然有着重要的影响。

与此同时,注重语言材料的语料库语言学仍然是一个重要的分支,并在 80 年代随着计算机计算能力的迅猛发展得到越来越多的重视。在 80 年代,一些语言学家、哲学家还发展了把语言纳入认知范畴来研究的认知语言学。

可以看到,在语言学发展的过程中,存在着经验主义(注重语言材料)和理性主义(注重语言机制)的交替发展。这种情形也出现在了计算语言学的发展过程中。

从语言学到计算语言学

计算语言学诞生之日正值乔姆斯基学派的理论大行其道之时,自然语言处理的主流技术是基于规则的,从各种句法分析技术到句法语义分析技术,利用规则来描述语言现象使之能为计算机所处理是计算语言学的主导方法。

20 世纪 80 年代末和 90 年代初,由于大量联机语料的出现以及计算机处理能力的大幅度提高,也由于规则方法迟迟未能达到人们预期的目标,统计自然语言处理逐渐兴起,成为自然语言处理中与规则方法比肩发展的两个方向。

在统计方法开始盛行之初,规则方法和统计方法存在着很多的对立,但是不久,人们便认识到二者并不是不可调和的两个对立面,而是互为补充的。詹姆士·艾伦(James Allen)在他的《Natural Language Understanding》(第二版)一书中,在保留规则方法的同时,增加了一些统计方法的内容,在序言中,他谈到,老方法(基于规则)和新方法(基于统计)是互为补充的,谁也不能替代谁。

全 书 安 排

全书分为三个部分。

第一部分用来介绍一些重要的基于规则的自然语言处理技术,这部分是从第一章开始直到第五章。其中,第一章介绍面向计算机处理的上下文无关语法及其形式化表示方式;第二章介绍了几种基于上下文无关语法的句法分析算法;第三章介绍基于特征的增强上下文无关语法以及基于该类语法的句法分析方法。后面两章介绍语义层面,其中,第四章介绍词汇语义的表示和处理;第五章介绍句义的表示和处理。

第二部分从第六章到第八章,介绍一些基于统计的自然语言处理技术。其中,第六章介绍 n 元语言模型;第七章介绍隐马尔科夫模型;第八章介绍概率上下文无关语法。(王伟、孙健两位博士参与了第六和第七章的选材和编写。)

在第三部分介绍一个典型的自然语言处理的应用——机器翻译,为本书的第九章。这部分主要从技术的角度来考察、分析各种机器翻译系统在规则和统计技术下是如何来实现的,而不过多地介绍某个具体的系统。

作者

2002 年 1 月

第一章　上下文无关语法

说到语法(在本书中,语法均在句子层面使用,因此将不仔细区分语法和句法两个词的使用),人们可能首先会想到语言学课程。在语言学的教科书中,语法是一个主要的内容。在那些语法中,规定了如何用词构造句子,何种用法是不允许的等等。通常,语法可以用来辅助人们完成两件事情,其一是作为判定一个句子构造得是否合适的重要依据,也即,一个句子是否合乎语法;其二,依据语法来分析句子的结构,帮助人们理解句子内容,这一过程在人们学习外语时是尤为明显和重要的。(由此也可见,利用语法来进行句子结构分析对于进行自然语言理解是有一定认知依据的。)

对于计算机自然语言处理,利用认知依据来建立计算模型是一种可行的途径。因而,让计算机能够利用语法来分析句子是进行自然语言处理的一个重要阶段。与人类使用语法相同,计算机利用语法来分析句子也可以有两个层次:其一是识别一个句子是否合乎语法。通常把能完成该任务的计算机程序称为句子识别器。其二是分析句子的内部结构,确定句子的语法成分,为进一步的句子分析和理解提供足够的基础。通常把能完成第二个任务的计算机程序称为句法分析器。显然可以看出,句法分析器比识别器具有更强的能力。

为了实现句子识别器或句法分析器,需要预先赋予计算机两个东西。

第一个是语法:通常语言学教材中的语法是面向人的,为了让机器分析句子,需要让机器知道这些语法,这种面向机器处理的语法也称为形式语法,它是规定语言中允许出现的结构的形式化说明。其中很重要的是如何表示形式语法,即形式语法的表示方式。本章将介绍两种表示方式:重写规则和转移网络。

第二个是语法分析算法:机器依据形式语法来识别和分析句子并决定其结构的方式。在计算机自然语言处理中,我们更多地关心句法分析器的算法,因为句法分析器比识别器具有更强的能力,能够提供更多的信息。句法分析算法还应包括其中采用的数据结构的构造,在分析之后如何表示句子的句法结构等各个方面。在通常的人类自然语言中,未经分析的句子是线性的符号串表示。本章将介绍在经过分析后产生的句子结构的树形表示,以及两种表示对于理解句子所带来的差异,也即句子的结构歧义问题。

本章主要明确两个方面的内容,其一是形式语法的表示;其二是句子结构的表示。各部分是这样安排的:1.1 节一般性介绍形式语法的描述问题;1.2 节利用重写规则描述上下文无关语法;在 1.3 节介绍用转移网络和递归转移网络来描述上下文无关语法;在 1.4 节介绍句子在经过句法分析后产生的句法结构的树形表示;最后是对本章的小结。

1.1 形式语法描述

最简单的描述语法的方式是把一种语言中所有可能的句子都列举出来作为这种语言的语法。

这种描述语法的方式其问题是明显的,可以从以下两个方面来看。

第一,在这种语法描述方式下,为了要完成句子识别的任务,即判断一个句子是否符合该语法,也即判断该句子是否是这种语言中的一个合法的句子,就需要列出这种语言中所有可能的句子,这样要判断一个句子是否合乎语法,只需要把该句子和这种语言中的句子逐一比较,看看是否有和该句子完全相同的句子。而通常,我们所使用的语言其句子是无穷多的,无论是对于计算机还是对于人,穷举都是不可能的,对于计算机处理,一个可行的方案是编制一个程序来按某种算法生成并输出这种语言的所有句子,显然,对于有无穷个可能句子的语言而言,这个输出过程是无限的。对于这类语言,有如下的定义:

对于一种语言,如果能编写一部程序,使得能按某种次序输出该语言的所有句子,则称该语言是可递归枚举的。形式语言理论的一个结论是,可递归枚举语言是一种很强的语言,对它的句子进行是否合乎语法的判断并不一定能完全实现。假设给定某种可递归枚举语言,并编写出了一部程序能生成其所有的句子,现在来判断一个句子是否合乎语法,即该句子是否能和程序输出的某个句子完全匹配,如果找到一个完全匹配的句子,那么,可以说该句子是合乎语法的。但是,如果一直没有找到匹配的句子,也不能断定该句子不合乎语法,因为它还可能与后面输出的句子相匹配。由于在句子个数无限多时程序的输出过程是不会终止的,因而它与后面输出的句子相匹配的可能性就一直存在,也即是说,对句子的合法性判断可能不会在有限步骤结束,这对于计算机处理而言,是不可实现的。

可用计算机实现合法性判定的语言应该如下定义:

如果对于一种语言,能编写一个程序在有限步骤内完成上述判断,则该语言称为是可递归的。一种语言是可递归枚举的,却不一定是可递归的。

可见,自然语言句法分析的任务只能在可递归语言上实现,因此,相应的语法描述也应该是可递归的。

第二,如果用列举所有句子的方法作为语法描述,那么这种描述是无助于对新句子进行结构分析的,而只能实现新句子的合法性识别。其方法是通过把新句子与该语法所列举出的句子进行匹配来判定新句子是否来自于这些合法句子的集合中,即是否是一个合法的句子。除此之外,不能得出关于句子结构的进一步信息。

从上述的两点可以看出,用列举句子的方法作为语法描述难以完成对句子结构进行分析的任务。

另外,用列举句子的方法作为语法描述对于解释人类语言构造的方式没有任何帮助,好的语法应该具有推广能力,能够抓住语言现象中的共同点,这对于理解语言、发掘语言运用的认知原理,进而发掘人类思维的本质都具有重要意义。更具现实意义的是,从计算资源的角度来看,具有推广能力的语法更能节约存储空间。

从上面的分析可以看到改进语法描述的某些端倪:它描述的应该是可递归语言,并且

它描述的句子应该是有内部结构的,而且这种内部结构是具有共性的,因而这种语法是有推广能力的。

一个让语法具有推广能力的方法是首先建立一些语法范畴(也常被称为语法类别、词性等),把具有相似语法行为的词归入一个相同的语法类别;然后,描述这些语法类别如何进一步组合的语法行为,这样构造的任何一个语法行为对于具有相似语法范畴的词都具有推广能力。

这时,语法描述就是列出语法类别的所有可能的组合模式。例如:(本章使用几种常用的语法类别,包括 ART(冠词)、N(名词)、V(动词)、ADJ(形容词)、ADV(副词)和 PRON(代词)等。)

ART + N

ART + N + V

ART + ADJ + N + V

就是几个在英语中允许的语言模式,这种以语法类别为单元的模式就比以词本身为单元的列举具有更强的推广能力。例如,上述的

ART + N

模式就可以用来描述诸如:a book, the sentence 等等很多个词串。

但是,如果模式有限,每个语法类别中的词有限,则这样的语法可以生成的句子是有限的。然而通过引入几个记号可以大大扩展上述模式的描述能力。

(1) Kleene 星,记为 *,例如:

$$ART + ADJ + ADJ^{*} + N \qquad (1\text{-}1\text{-}1)$$

* 号出现在 ADJ 的右上角,表示 ADJ 可以出现 0 次或 0 次以上。这样,可以描述在一个冠词和名词之间插有多个形容词的语言模式。

(2) Kleene 加,记为 +,例如:

$$ART + ADJ^{+} + N \qquad (1\text{-}1\text{-}2)$$

+ 号出现在 ADJ 的右上角,表示 ADJ 可以出现 1 次或 1 次以上。这个式子描述的内容与模式(1-1-1)是相同的。

(3) 圆括号,记为( ),例如:

$$ART + (ADJ) + N \qquad (1\text{-}1\text{-}3)$$

ADJ 外加一个圆括号表示 ADJ 可以出现 1 次,也可以 1 次也不出现。也就是说,ADJ 是可选的。

(4) 垂直线,记为|,例如:

$$N|PRON + V \qquad (1\text{-}1\text{-}4)$$

N 和 PRON 中间的直线表示可以是 N,也可以是 PRON,它们都可以与后面的 V 组成这个模式,但二者不能同时出现。

在引入了这几个记号后,基于有限个语法类别的组合模式就可以构造无限多个句子,比如,在模式

$$ART + ADJ^{+} + N + V$$

中,可以通过无限次重复出现 ADJ 而产生无限多个句子。

这样形成的上述语法描述称为正则表达式。与第一种用列举的方法来作为语法描述

相比,利用正则表达式来描述语法具有了一定的推广能力,并可以利用有限的语法类别的组合模式来生成无限的句子。但是,这种语法描述所表现出的推广能力还是远远不够的,还有进一步改进的余地。例如,上述的模式 (1-1-1) 和模式 (1-1-3) 通常会在句子中处于类似的位置,起着类似的作用,因此,应该可以进入更高一层次的抽象。这可以通过简单地定义一个比语法类别具有更高抽象的概念——短语来实现。简单地说,短语是经常反复出现的符号串,它构成确定的语法成分。例如,上述的两个字符串:模式(1-1-1)和模式(1-1-3)可以进一步用一个短语名称 NP (名词短语)来概括。

在引入短语概念的基础上,可以进一步扩展正则表达式的描述能力。如果在正则表达式中,除了可以包含原有的语法类别,还可以包含短语,就可以形成一种描述语言的语法——短语结构语法。

1.2 短语结构语法

为描述短语结构语法,需要先介绍重写规则。重写规则是一种形式化表示方式,可以用来描述规则,例如:

$$S \rightarrow NP\ VP$$

就是一个重写规则。其中,S 代表一个句子;NP, VP 表示两个短语,NP 表示一个名词短语,VP 表示一个动词短语。该规则的意思是说左边的符号 S 所代表的项可以被合乎语法地替换成右边符号所代表的两个项,即被重写为右边两项的组合。

一个形式语法可以包含若干条重写规则。通常一些重写规则的集合用 P 来表示。除此之外,组成一个完整的形式语法还有另外几个要素:其一是所谓终结符号集合,用 T 来表示,一个终结符号代表一个这样的项,它在此语法中不能再被重写为其他项的组合,通常是该形式语法所描述的语言中的词汇的语法类别(如 N, V 等等),或者就是该语言中使用的词汇(如英语中的单词 a, boy 等等);其二是非终结符号集合,用 NT 来表示,一个非终结符号代表一个这样的项,它在此语法中可能再被重写为其他项的组合,如果上述终结符号指的是语言中的词汇本身,那么非终结符号也包括词的语法类别;其三是一个特殊的非终结符号 S,表示句子。因为句法分析针对的单位均为句子,因而 S 就十分重要,它通常是对句子进行语法分析的开始或结束符号。

这样,一个完整的用来描述一种语言的形式语法就可以表示为四元组 (T, NT, S, P),且 $T \cap NT = \varnothing$,即一个符号不能同时既是终结符号又是非终结符号。令 $V = T \cup NT$, V^* 表示由 V 中的符号所构成的全部符号串(包括空符号串 ε),而 V^+ 表示 V^* 中除 ε 之外的一切符号串的集合。P 中的每条规则形如:

$$a \rightarrow b$$

其中,$a \in V^+$, $b \in V^*$,且 $a \neq b$。

一个简单的例子:

$$NT = \{S, NP, VP, ART, N, V\}$$

$$T = \{\text{the},\ \text{a},\ \text{boy},\ \text{sees},\ \text{cat},\ \text{dirty}\}$$

T 中的符号串均为英语单词。

P 中包含如下几条重写规则:

$$S \rightarrow NP\ VP \tag{1-2-1}$$

$$NP \rightarrow ART\ N \tag{1-2-2}$$

$$NP \rightarrow ART\ ADJ\ N \tag{1-2-3}$$

$$VP \rightarrow V\ NP \tag{1-2-4}$$

$$ART \rightarrow the \mid a \tag{1-2-5}$$

$$N \rightarrow boy \mid cat \tag{1-2-6}$$

$$V \rightarrow saw \tag{1-2-7}$$

$$ADJ \rightarrow dirty \tag{1-2-8}$$

其中,(1-2-5)～(1-2-8)表明非终结符号的所属语法类别,四元组 (S, NT, T, P) 就表示了一个语法。

利用上述语法,不仅可以进行句子的合法性识别,还可以对一些句子进行结构分析。下面对这种合乎语法的可识别性以及结构分析的内容进行定义。

导出:某个句子被称为由一个语法导出的(一个语法导出了某个句子),如果能由 S 开始依据语法中的一系列重写规则重写出该句子。如果一个句子能由某个语法导出,则称这个句子是合乎该语法的。

看看下面的句子:

$$The\ boy\ saw\ a\ cat\ . \tag{1-2-9}$$

是否合乎上述语法,如果合乎语法,其结构又是怎样的。

(1) 由规则(1-2-1),合乎该语法的句子都是应该能被重写为一个名词短语加一个动词短语,因此,只需看句子(1-2-9)是否是由这两部分组成的。名词短语和动词短语是非终结符号,还可以进一步分解。

(2) 名词短语是应该能被重写为由一个冠词加一个名词组成的(规则(1-2-2)),或者由一个冠词加一个形容词再加一个名词组成的(规则(1-2-3));而动词短语是应该能被重写为由一个动词加一个名词短语组成的(规则(1-2-4))。显然,句子(1-2-9)中 the boy 可以组成一个名词短语,see a cat 可以组成一个动词短语,而其中 a cat 又是一个名词短语。

(3) 由上述两条,可以看到句子(1-2-9)是能够依据一系列重写规则导出的,规则的使用次序可以如下:

规则(1-2-1),(1-2-2),(1-2-5),(1-2-6),(1-2-4),(1-2-7),(1-2-2),(1-2-5),(1-2-6)

同时,得到其组成结构:

(NP1 (The boy) VP1(saw NP2(a cat)))

括号由内向外,反映了句子的组成,The 和 boy 组成一个名词短语 NP1,a 和 cat 组成一个名词短语 NP2,saw 和 NP2 组成一个动词短语 VP1,NP1 和 VP1 组成句子 S。

显然,很多自然语言的句子在上述几条规则(语法)下是不合法的,即有很多自然语言的句子不能由上述语法导出。通过增加重写规则、增加(非)终结符号都可以增加语法能导出的句子的数量。一般地,我们称能生成更多个句子的语法具有更强的生成能力,显然,对语法具有较少约束的语法具有更强的生成能力。

按照上一节定义的可递归枚举语言与可递归语言来分,上述的一般的短语结构语法是可以描述可递归枚举语言的,即某些短语结构语法导出的语言是可递归枚举的而不是可递归的。

通过对一般的短语结构语法进行限制,可以得到被称为乔姆斯基体系的 4 类语法。下面,按照生成能力由弱到强(约束由多到少)的次序分别简单介绍。

1 . 正则语法(3 型语法)

正则语法分为左线性语法和右线性语法。

在左线性语法中,所有重写规则必须采用如下的形式:

$$A \rightarrow Bt \text{ 或 } A \rightarrow t$$

其中,A,B 是非终结符号;t 而为终结符号。

而在右线性语法中,所有重写规则必须采用如下的形式:

$$A \rightarrow tB \text{ 或 } A \rightarrow t$$

正则语法是乔姆斯基体系中生成能力最弱的一个,一些常见的语言现象都不能用正则语法来生成。一个简单的例子是任意符号“ x ”两边成对匹配添加括号,通过不断嵌套的方式可以实现一系列句子:

$$x,(x),((x)),(((x))),((((x)))),\cdots$$

为了生成这种语言的句子,当生成到“ x ”时必须知道前面已经生成了多少个“ (”,以便能生成同样数量的“) ”相匹配。而对于正则语法,无论是左线性语法还是右线性语法,都只能独立地生成“ x ”某一侧的符号,无法进行匹配。

在自然语言中,也存在着类似的匹配模式。例如:“ 如果 A 那么…… ”、“ 因为 A 所以…… ”等句子结构(其中 A 表示一个符号串),通常都需要匹配出现,这种模式也可以进行不断地嵌套形成复杂句子:

如果 A 那么……,如果……如果 A 那么……那么……,……

同样,当生成到 A 时,也必须知道前面已经生成了多少个“ 如果 ”,以便能生成同样数量的“ 那么 ”相匹配。

2 . 上下文无关语法(2 型语法)

在上下文无关语法中,每一条规则都采用如下的形式:

$$A \rightarrow x$$

其中,A 是非终结符号,$x \in V^*$ 。这种规则的应用不依赖于 A 出现在什么上下文环境中,因此称为上下文无关语法。

上下文无关语法比正则语法具有更强的生成能力,能反映更多的自然语言现象。但是,还有一些自然语言现象并不能由上下文无关语法来描述,有些情况下,一条重写规则的应用是受上下文制约的。

3 . 上下文有关语法(1 型语法)

在上下文有关语法中,每一条重写规则都是这样的:

$$x \rightarrow y$$

其中,$x,y \in V^*$,且 y 的长度(即符号串 y 中的符号个数)总是大于或等于 x 的长度。

上下文有关语法的重写规则也可以这样来表示:

$A \rightarrow y / x\_z$

其中,A 是非终结符号;$y \in V^+$;$x, z \in V^*$,在这种表示中,可以很明显地看出所谓上下文有关的含义来:如果 A 出现在上下文 x_z 中,即前面紧挨着符号串 x,后面紧挨着符号串 z,则 A 可重写为 y,可以看到 A 可重写为 y 是有上下文约束的。

4.无约束短语结构语法(0 型语法)

0 型语法对规则没有任何约束,其定义的语言可能不是递归的,因而就不可能设计一个程序来判别一个输入的符号串是否是 0 型语言中的一个句子,所以 0 型语言很少被用来处理自然语言。

总而言之,在乔姆斯基体系中,如果一种语言可以被一部 i ($i = 0, 1, 2, 3$) 型语法所生成,就称它为 i 型语言。

由于在乔姆斯基体系中,语法的型号越高,对重写规则所附加的限制也越多,所以 3 型语言是 2 型语言的一个子集,2 型语言是 1 型语言的一个子集,依此类推,有:0 型语言 ⊃ 1 型语言 ⊃ 2 型语言 ⊃ 3 型语言。从语法的生成能力看,0 型语言最强,1 型到 3 型依次递减,3 型最弱。

在上述乔姆斯基体系的四种语法中,上下文无关语法是计算语言学的重要研究对象。由于其描述能力强,足以描述自然语言中的大部分结构,同时又是可递归的,可以构造有效的句法分析器来进行句子的分析,因此,目前大多数计算机处理用的语法都是基于上下文无关语法的。

1.3 转移网络

除了用重写规则来描述语法之外,转移网络也是一种方式。

转移网络是一个图,图由结点集合和边集合组成,每条边都是带标记的。结点集合中有一个结点是初始状态或称开始状态,还有一个或多个终止状态。

有限状态转移网络是一种较为简单的转移网络。上述乔姆斯基体系中的正则语法可以用有限状态转移网络来等价地描述。图 1-1 是一个有限状态转移网络。

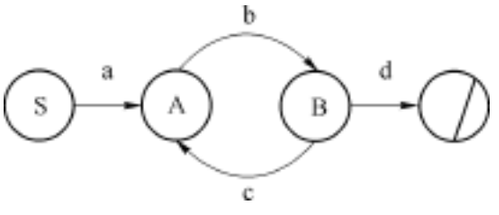


图 1-1 有限状态网络

图中标为 S 的结点是句子分析的起始点,从结点 S 有一条指向结点 A 的有向弧,弧上有一个语法类别标记,意为句子的第一个词如果具有语法类别 a,则 a 可以转移到结点 A;然后如果第二个词具有语法类别 b,则可转移到结点 B;在结点 B 时,有两个有向弧分别指向两个结点,其中标有 d 的有向弧指向的是终止结点(终止结点中标有一斜划线)。如果一个句子能沿着有向弧最终达到终止结点,同时句子也到最后的话,就可以断定该句子是符合该有限状态转移网络所描述的语法。

显然,上述转移网络能用来等价地描述如下的一部正则语法:

$$S \rightarrow aA$$

$$A \rightarrow bB$$

$$B \rightarrow cA$$

$$B \rightarrow d$$

因此,正则语法也可称之为有限状态语法。

对有限状态转移网络进行扩展,可以建立具有更强描述能力的递归转移网络。递归转移网络的描述能力与上述乔姆斯基体系中的上下文无关语法等价。例如,如下一个上下文无关语法:

$$S \rightarrow NP\ VP$$

$$NP \rightarrow ART\ ADJ^{*}\ N\ (PP)$$

$$VP \rightarrow V\ (NP)$$

$$PP \rightarrow PREP\ (NP)$$

可等价地用如图 1-2 中的递归转移网络来描述(PP 表示介词短语),图中每一个单独的网络与上述的一个规则相对应。

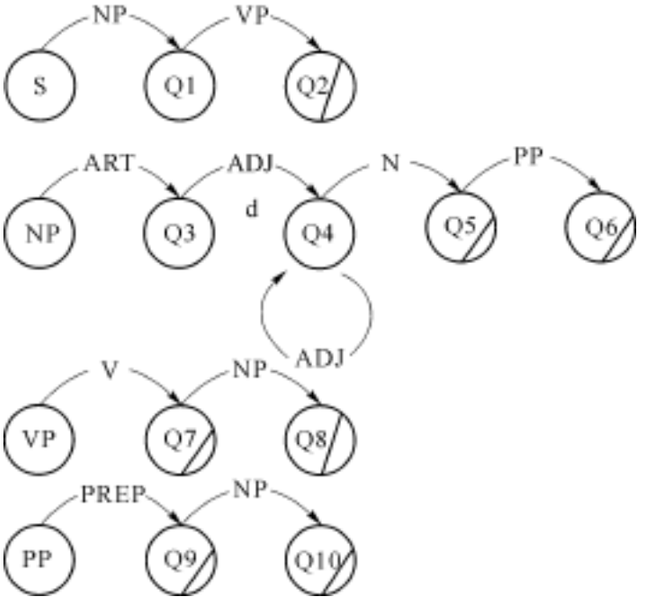


图 1-2　递归转移网络

可以看到,在递归转移网络中,有向弧上的标记不仅可以是某个语法类别,还可以是某个短语结构,这是递归转移网络与有限状态转移网络的一个重要区别所在,这使得递归转移网络能支持递归结构。例如:在图 1-2 的四个网络所构成的递归网络描述中,主网络是起始结点为 S 的网络,一旦在这个网络中沿着有向弧能到达终止结点,则可判定相应的句串是合乎该网络所描述的语法的。而为了从 S 结点转移到终止结点 Q2,首先要转移到 Q1 结点。而为了转移到 Q1 结点,首先要判定句子前面是否有一个有向弧上标的结构:NP。而为了判定句子中的 NP 结构,需要调用第二个网络,即 NP 网络。在 NP 网络中有两个终止结点,如果在第二个终止结点终止的话,将意味着在 N 后面有一个 PP 结构。为判定 N 后面是否有一个 PP 结构,需要调用第四个网络,即 PP 网络。在 PP 网络中若要达到终止结点,首先要找到一个 PREP,而后要判断后面是否有一个 NP 结构,这样反过来又需要调用第二个 NP 网络,如此形成递归调用。

1.4　短语结构与句法树

自然语言,无论是语音形式还是文本形式都表现为线性的,语音在时间上顺序出现,文本在空间上顺序出现。例如,下面以文本形式出现的句子:

I saw a boy .

其中的符号从左到右地顺序出现。

在这种线性描述下,没有任何关于句子构成的信息,无法知道句子是依据哪些语法规则以及如何构成的。而如前所述,一般认为,知道句子组成结构的信息,就能够帮助人们理解句子内容。为此,希望能有一种具有刻画句子语法结构的表示方法,这种表示能描述句子是如何由其子部分构成的。

最常用来描述句子语法结构的表示是树,通常描述语法结构的树称为句法树。树形结构反映了语言中小单元组成大单元,大单元组成句子的递增层次结构,我们可以在句法树中获得在线性结构中所不能得到的句子结构信息。例如,上述句子的句法树如图1-3所示。

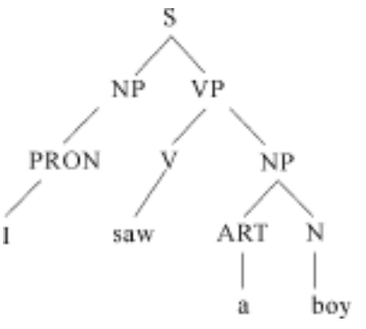


图 1-3　句法树

图中所示的树表现了这样一些句法结构信息:整个句子(S)是由一个名词短语(NP)和一个动词短语(VP)构成。而名词短语就包含一个代词(PRON),这个代词就是 I。而动词短语由一个动词和另一个名词短语组成,其中动词是 saw;名词短语还有一个内部结构,由一个冠词和一个名词组成,冠词是 a,名词是 boy。这是一个自顶向下的分析过程。或者也可以相反,自底向上地说:这个句子中,首先 a 作冠词,boy 作一个名词,二者组成一个名词短语;这个名词短语与动词 saw 一起构成了一个动词短语;而与此同时,I 作为代词形成一个名词短语,这个名词短语和动词短语一起构成了整个句子。

从句法树中同时也可以看到一个句子是如何运用多个语法规则组合而成的。如上面的句子中,就是运用了如下的几条语法规则:

S→NP VP

VP→V NP

NP→PRON

NP→ART N

PRON→I

V→saw

ART→a

N→boy

而生成的 (其中后 4 条我们后面通常会归入词典中,而非放在语法中)。其过程如下:

S

NP VP　　(重写 S)

PRON VP　　(重写 NP)

I VP　　(重写 PRON)

I V NP　　(重写 VP)

I saw NP　　(重写 V)

I saw ART N　　(重写 NP)

I saw a N　　(重写 ART)

I saw a boy　　(重写 N)

其中,使用重写规则的次序是可以不同的。比如,也可以先重写 VP 再重写 NP,最终的句子是一样的。这样,在得到一个句子的句法结构树之后,就很容易判定该句子是否合乎语法,这只需要看它在构成句子时所用的规则是否都是该语法中的规则。

本书后面将主要用树作为句法结构的主要描述工具,因而对其中的概念作如下的介绍和规定:

树是图的一种,是由结点集合和边集合组成的,树与其他图的主要差别在于树中不包含任何循环。一条边上端的结点管辖边下端的结点,没有被任何结点管辖的结点称为根结点,而没有管辖任何结点的结点是叶子结点。

除了用句法树之外,也可以通过增加辅助记号,用接近线性的方式来表示和句法树同样多的句法结构信息。例如,上述的句法树可以用如下加括号的方式等价描述:

```
(S ( NP (  PRON I) )
    (VP (V saw)
        (NP (ART the)
            (N cat )
         )
    )
)
```

在一些线性结构的句子中,各个词之间的结构关系可能用多个树来描述,例如下面的句子:

I saw a boy with a telescope .

句子中的介词短语结构 with a telescope 有可能是 boy 的后置修饰成分,与 a boy 一起构成一个名词短语;也有可能是谓语动词 saw 的状语,表明该行为所用的工具。

对应于第一种可能和第二种可能,可以分别用两棵不同的句法树来描述,如图 1-4 中的(a),(b)所示。

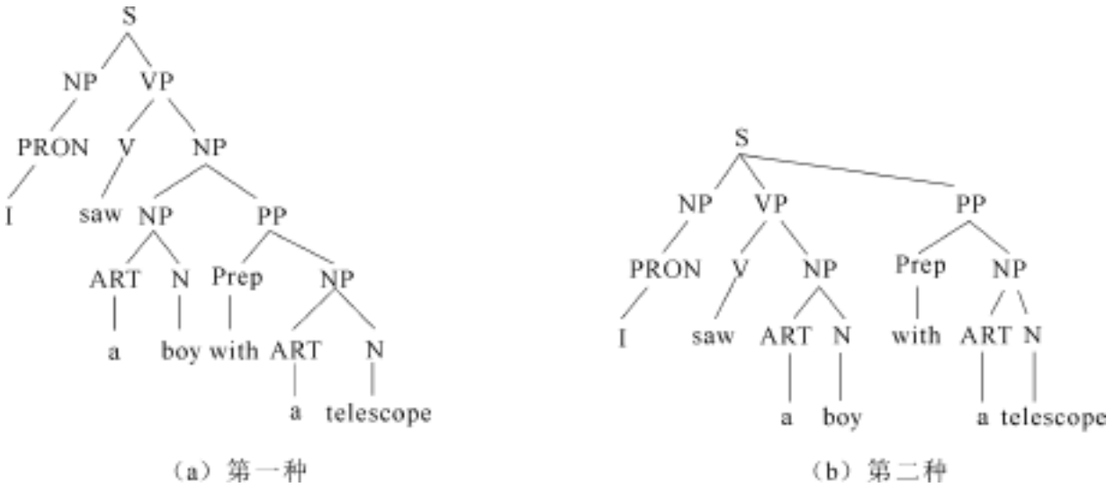


图 1-4　同一个句子的两棵句法树

可以看到,两棵句法树分别揭示了两种可能的句子结构方式。同一个句子存在多个

不同的句法树表明了句子存在结构上的歧义,揭示这种可能存在的歧义,把所有可能的句法结构分析出来正是句法分析的重要任务。

小　　结

本章主要介绍了自然语言语法处理中的表示问题。其一是形式语法的表示方法,包括利用重写规则表示语法和用转移网络来表示语法;其二是句子语法结构的表示形式,利用树来表示句子,可以消除线性结构中存在的语法结构上的歧义,这是自然语言处理的一个重要任务。在下面的一章中,将介绍如何基于上述不同的表示下的语法,来得到一个线性结构句子的所有可能的句法树,即如何进行句法分析。

第二章　上下文无关句法分析器

一个句法分析器可以实现以下两个方面的目标:

(1) 确认输入句子是否可以由给定的语法来描述,即输入句子是否合乎给定的语法;

(2) 识别句子各部分是如何依据语法规则组成合法句子,同时生成句法树。

要为某个语言建立一个达到上述目标的句法分析器,需要有两个方面的准备。一方面是该语言的形式化语法描述,规定在该语言中允许出现的结构。其中也包括在该语言各种语法类别中可以出现的所有终结符号,这部分通常以词典的形式单独存储。另一个方面是根据语法来分析句子并决定其结构的方法,即句法分析技术。这两个部分共同组成一个完整的句法分析器。

本章是这样安排的,首先简单考察构造句法分析器的第一个方面,即如何构造一个好的语法;之后主要的内容是基于一个简单的语法来介绍几种句法分析算法。

由于本章的句法分析算法所依据的语法都是上下文无关语法,因此,本章的句法分析器均为上下文无关句法分析器。

2.1　语　　法

第一章介绍了如何形式化描述一个语法系统,包括利用重写规则和转移网络来描述语法规则。但是并没有提到这些语法规则本身是如何构造的,因为这是与语言相关的。我们可以限定一个语法系统是上下文无关语法,但它也只是对每条语法规则的形式具有约束,而对每条语法规则中具体内容的确定没有帮助。

一般在为一个语言构造语法的时候,最感兴趣的有三个方面:语法的推广能力,这是指能被语法正确分析的句子范围;选择能力,即能被语法识别出问题的非句子的范围;可理解性,即语法本身的简单性。

现有语法书中的语法系统,可以作为一个有益的借鉴,它是语言学家们长期研究语言现象所得出的一些规律性的结论,其中的很多规则是具有上述三个方面的特点的。但已有的实践表明,由于自然语言的句子是无限的,而且处于不断地增长和更新中,语言中的新现象、例外现象、歧义现象等不断出现,所以单单利用语法书中的规则难以精确覆盖一种自然语言的所有语言现象。

对于复杂的语言现象,一种简化可行的研究方式是对要研究的语言对象进行一些限定,比如排除一些例外、排除一些不常用的词汇等等,或者根据语言使用的领域的特点进

行规则的限定等等。总之,是在庞大语言现象的一个限定子集上构造语法,这样就可能使上述三个方面都达到较好的性能。实际上,人们在日常高频使用的一般也就是所有语言现象的一个子集,所以,这种方法是有实际意义的。

在本章以下的部分,为了解释句法分析的算法,均采用一个十分简单的语法,它只包含如下几条重写规则:

S→NP VP (2-1-1)

VP→V (2-1-2)

VP→V NP (2-1-3)

VP→AUX VP (2-1-4)

NP→PRON (2-1-5)

NP→ART N (2-1-6)

NP→ART ADJ N (2-1-7)

NP→ADJ N (2-1-8)

PRON→I (2-1-9)

V→saw (2-1-10)

ART→a (2-1-11)

N→boy (2-1-12)

其中 AUX 表示系动词,规则(2-1-9)～(2-1-12)通常放在词典中,余下的几条语法规则(2-1-1)～(2-1-8)在后面统称为语法 2 .1。此外,在本书后面的例子中,如果出现其他的词,都假设其词汇类别信息已经存放在一个可用的词典中了。

2 .2 基于符号串的句法分析

一个句法分析算法可以表述为一个搜索过程,其搜索空间是语法规则,搜索过程就是检查各种语法规则所有可能的组合方式,搜索目的是最终找到一种组合,其中的语法规则能够生成一棵用来表示句子结构的句法树。通常在算法中,句法树都不是被显式地构造出来,而是隐含在所搜索出的语法规则的组合序列中,通过这个语法规则的序列,可以很容易构造出相应的句法树。我们会在后面的例子中看到这一点。搜索算法同时也可以确定该输入句子是否是合乎语法的。

句法分析的搜索过程可以有两个相反的方向,一个是自顶向下;另一个是自底向上。本节先介绍自顶向下的方法,在下一节介绍自底向上的方法。

自顶向下的分析方法是从符号(S)开始,S 称为这种句法分析的初始状态。算法试图通过搜索并应用语法中的重写规则来不断改变算法的状态序列,直到最终生成与输入句子的词汇类别序列相匹配的符号序列,就可以断定该输入句子是合乎语法的,并且最终所用到的那些重写规则序列就蕴涵了句子的句法结构。或者,当所有可能性都尝试后还不能生成输入的句子,则可以断定该输入句子不能由该语法分析,或该句子在该语法下是不合法的。

在算法进行的任何时刻,算法的分析状态都可以表示为一个符号列表,这个列表通常

称为符号串。例如,初始状态的符号串为 (S),在应用重写规则 S→NP VP 后,状态序列符号串就变成了(NP VP),对于其中的 NP,可以进一步再用规则 NP→ART N,这时状态序列符号串为(ART N VP)。当然这时候也可以用规则 NP→ART ADJ N,则符号串为(ART ADJ N VP)等等。过程将一直进行到符号串完全由终结符号(在句法分析时,一般把词汇类别作为终结符号)组成,然后检查是否与输入句子的词汇类别序列相匹配。但是这种方式是十分浪费的,因为在上述的方案中,每一条路径都要等到一次搜索走到尽头(符号串完全重写为终结符号)后才能判断其对错。而实际上,大量的错误路径通常在完成几个较早的搜索步骤后就可以判定出来,无需进一步试探该路径,从而可以减少后面无用的搜索步骤。而错误发现得越早,可以节省的搜索量就越多。因此,一个好的算法就是要能尽早发现搜索路径中的错误,在由语法生成可能的词类序列的同时,不断与输入句子可能的词汇类别序列对比,在这个过程中不断尽早排除一些不可能的结构,直到最终完全相符。

下面介绍一个简单的自顶向下的句法分析算法,例子中采用语法 2 .1,并假设所需的词汇类别信息均已在另外的词汇类别词典中给出。

为描述算法,需要做以下一些准备工作。

引入位置标记,包括两个方面的标记,一方面是在输入句子上的标记,即对于给定的输入句子中的词汇按次序标记其所在的位置。例如,对于句子:

The boy cried .

加上位置标记为:

$_1$The $_2$boy $_3$cried$_4$

每个词前面的下标数字表示该词的位置,而句子最后一个下标数字标志句子的结束。在上面的句子中,单词 The 在句子的位置是 1,达到位置 4 就表明句子已经结束。通过引入标记,句法分析程序可以判定其当前处理的词是哪一个以及是否已完成对一个句子的处理。在句子上的这种标记在句子输入后就可以立即完成。对句子的位置标记在后面的其他部分还会用到,本书将一直沿用这里的约定方法。

另一方面,句法分析过程中产生的符号串也包含某种标记。例如,在分析过程中有这样一个符号串:

((N VP)2)

该符号串后面的标记 2 表明算法对输入句子的第一个词的分析已完成,分析算法已经进入到对第二个词的匹配;同时,由前面的(N VP)部分可知:算法期望第二个词的句法类别是 N,并且在其后是一个 VP。

符号串在分析过程中是不断更新变化的,其改变状态的操作根据当前符号串的第一个符号是否是词汇类别符号而不同。

如果是词汇类别符号(例如在上述的符号串((N VP)2)中第一个符号为 N,就是一个词汇类别符号),那么就继续检查此时符号串后面的数字标记所对应的句子中的单词是否属于该词汇类别(例如在上例中,符号串中数字 2 对应的单词为 boy,其词汇类别为 N),如果是,就可以把该词汇类别从符号串中删去,生成新符号串(上例中把 N 去掉后符号串就剩下(VP)),同时位置标记加 1(综合起来得到新的状态序列为((VP)3))。

如果不是词汇类别符号(例如上述新生成的符号串(VP),VP 不是词汇类别符号),那么就从语法规则中寻找所有可能的规则来重写该符号(例如在语法 2 .1 中可以用来重写

符号 VP 的规则有规则(2-1-2),(2-1-3)和(2-1-4),如果有多个可用的规则,那么就会产生多个可能的新符号串(例如用规则(2-1-2),(2-1-3)和(2-1-4),产生的新符号串分别为(V),(V NP)和(AUX VP)),这时先取一个符号串,其余未用的符号串要保存以备回溯之用。所谓回溯就是如果用先选的符号序列在后面的分析中得不到句子的句法结构,那么就退回到有多个选择的地方,选择未使用的其他可能的符号串来进行下一步的分析。

下面举例说明这个算法是如何分析一个句子的。句子:

I saw a boy .

显然在进行算法之前,需要对句子加位置标记:

$_1$I $_2$saw $_3$a $_4$boy$_5$

算法过程如表 2-1 所示。

表 2-1 句子 I saw a boy 的算法过程

步骤	当前状态	备份状态(用于回溯)	步骤	当前状态	备份状态(用于回溯)
1	((S)1)		7	((V NP)2)	
2	((NP VP)1)		8	((NP)3)	
3	((PRON VP)1)	((ADJ N VP)1) ((ART ADJ N VP)1) ((ART N VP)1)	9	((PRON)3)	((ADJ N)3) ((ART ADJ N)3) ((ART N)3)
4	((VP)2)		10	((ART N)3)	
5	((V)2)	((V NP)2) ((AUX VP)2)	11	((N)4)	
6	(3)		12	(5)	

说明:

第一步是自顶向下句法分析算法的初始状态,符号串为(S),分析算法处于句子开头,位置标记为 1。

第二步判定符号串。第一个符号显然是非词汇类别符号,因此搜索语法中可用来重写该符号的规则。在语法 2 .1 只有规则(2-1-1)可以,因此依据语法规则更新符号串为(NP VP)。

第三步同样判定后,语法中有四条规则(2-1-5),(2-1-6),(2-1-7)和(2-1-8)可以重写 NP,先用规则(2-1-5)重写 NP 产生的新状态,其余三种可能的状态按次序保存,在回溯时使用。

第四步,由于此时符号串的第一个符号为词汇类别符号 PRON,因此可以按后面的标记位置找到句子中的对应的词。在例子中 1 号位置为 I,它的词汇类别恰好是 PRON,这样可以把该符号从符号串中去掉得到新的符号串;同时,后面的位置标记加 1,即得到第四步给出的状态序列((VP)2)。

第五步,在语法 2 .1 中可以找到三条规则重写 VP,先使用第一条规则(2-1-2),使用另两条规则产生的可能状态保存以备回溯用。

第六步和第四步类似,由于符号串的第一个符号已是词汇类别符号 V,句子的 2 号位置为 saw,其词汇类别是 V,则去掉已匹配的符号,得到新的符号串。此时的符号串为空,即按所用的语法规则来看,句子应该结束了,但后面的位置标记按算法是加 1,即为 3,这表明对句子的分析才进行到位置 3,而没有到句子结束的位置 5。因此可以断定前面的语

法规则的使用有问题,需要回溯。

第七步,回溯就是从最近保存的可能状态按照后进先出的原则取出来,这里最近的保留状态是在第五步保存的((V NP)2)。

第八步与第六步的前面部分一样,从符号串中去掉 V,但此时的符号串不为空,而是有(NP),位置标记加 1,得到新的状态序列((NP)3)。

第九步和第三步完全一样,依据语法 2 .1 可以产生四种可能的状态,先取规则(2-1-5)所产生的状态,其余三种状态按次序保存,在回溯时使用。

第十步之前省略了一个和第六步一样的分析,即按所用语法规则句子应该结束了,但位置标记却没有达到句子结束位置,因此要回溯,把在第九步最后保留的一个状态取出来,按后进先出原则,选用((ART N)3)。

第十一步,由于此时符号串的第一个符号为词汇类别 ART,回到句子的第 3 个位置,为 a,其词汇类别为 ART,从符号串中删去 ART,位置标记加 1,为 4。

第十二步,此时符号串的第一个符号为词汇类别 N,回到句子的第 4 个位置,为 boy,其词汇类别为 N,从符号串中删去 N。此时符号串为空,位置标记为 4 + 1 = 5,语法规则的使用和句子的位置标记均标识了句子的结束,因此,该句子的分析完成。

在上述的搜索过程中,省略掉被回溯的规则,最终使用的重写规则及其使用次序如下,它们构成了一条可以最终生成句子的路径。

S	
NP VP	(重写 S)
PRON VP	(重写 NP)
I VP	(重写 PRON)
I V NP	(重写 VP)
I saw NP	(重写 V)
I saw ART N	(重写 NP)
I saw a N	(重写 ART)
I saw a boy	(重写 N)

按照上述句子真正使用的句法规则集及其使用的顺序,就可以获得其句法树,如第一章图 1-3 所示。

现在,可以把上述算法一般地描述如下,其中用堆栈来保存供回溯之用的状态序列,称为回溯栈。

第一步,初始化状态序列为((S)1),回溯栈为空。

第二步,选择当前状态序列:

```
if(该状态序列为空)
{
  if(状态序列的位置标记是句子的最后位置)
    then{算法停止(成功地进行了句法分析)}
  else(状态序列的位置标记不是句子的最后位置)
    {
      if(回溯栈为空)
```

then {算法停止(没有成功的进行句法分析)}
else(回溯栈不为空)
then{从回溯栈弹出一个状态序列作为当前状态}
}
}
else(该状态序列不为空)
then {从状态序列中取出第一个状态,把它称为 C}
第三步,处理当前状态序列的第一个符号:
if(C 是终结符号)
{
if(当前状态序列的位置标记在句子中的下一个词可以是该终结符号类)
then {把 C 从当前状态序列中去掉,得到一个新的当前状态序列,位置标记 + 1,转到第二步}
else(当前状态序列的位置标记在句子中的下一个词不可能是该终结符号类)
{
then {从回溯栈弹出一个状态序列作为当前状态}
if (回溯栈为空)
then {算法停止(没有成功的进行句法分析)}
}
}
else(C 不是终结符号)
{
then {按重写规则重写该符号,并替换进当前状态序列,生成新的当前状态序列}
if(有多个可应用的重写规则)
then {取任意一个,而把其他几个压入回溯栈,转到第二步}
}

对于自顶向下的句法分析过程,要避免出现左递归的问题。如果在语法中有类似如下的直接递归(左递归)的重写规则

$$VP \rightarrow VP\ NP$$

就有可能出现无穷循环的问题。即当使用上述规则重写 VP 时,替换后的符号仍含有 VP,则又可以用上述规则进行重写,这个过程可以无限进行下去,这就是所谓的左递归规则的问题。对于这个问题,一种解决办法是规定在使用一次左递归规则后,就要用非左递归规则来代替它。例如可以用

$$VP \rightarrow V\ NP$$

来重写 VP。

上面的算法实际上是一个深度优先的搜索过程。在搜索过程的每一步,都有一个期望,只有当前一步的期望实现后,才能进行下一步。这个过程是顺序进行的。

与此不同的另一种搜索策略是广度优先。在图 2-1 中,标记了对同一个句子用两种搜索方式进行分析时,状态序列出现的次序。图中,每个状态左边的数字是按深度优先来

计的,而右边的数字是按广度优先来计的。

深度优先和广度优先各有利弊,通常深度优先可能会需要较少的回溯。

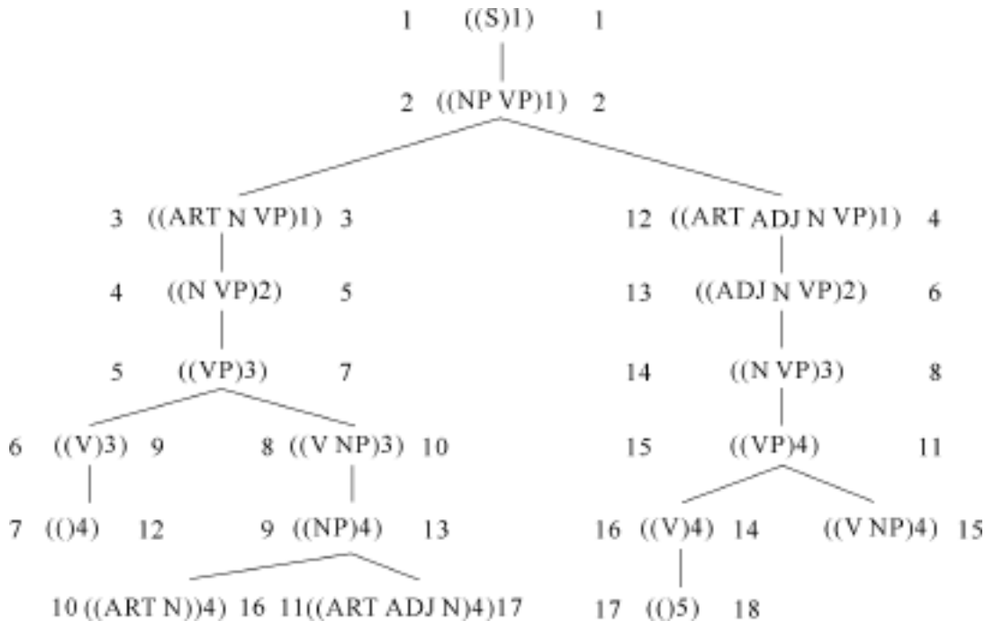


图 2-1 分别按深度优先和广度优先进行句法分析

2.3 自底向上的图句法分析

上一节介绍了句法分析搜索过程的自顶向下方式,本节介绍自底向上的方法。自底向上和自顶向下句法分析的主要差别在于语法规则的使用上。例如,对于语法规则

NP→ART ADJ N

自顶向下算法中使用本规则的方式是:如果期望在句子中找到一个 NP,那么按此规则就需要在句子中找到(ART ADJ N)序列;而在自底向上算法中使用该规则时是已经在句子中找到了符号串(ART ADJ N)后利用该规则把此符号串确认为是一个 NP。

在自底向上算法中,对句子的分析是从词串开始的,通过不断匹配重写规则来减少符号串的长度以组成更大的句子成分,直到组成 S 符号。

与上述的自顶向下算法一样,可以把匹配过程形式化为一个搜索过程从而建立自底向上的句法分析算法。搜索过程中的主要操作可以描述为:

(1) 用可能的词汇类别重写词;

(2) 用重写规则左边的符号替换与其右边相匹配的符号序列。

但是这种简单的搜索方法是十分昂贵的,由于算法可能会一次又一次尝试相同的匹配,因此,其中有很多的计算是浪费的。为避免重复计算,可以使用一种称为图的数据结构,利用这种数据结构可以保存已经进行的部分匹配的结果,并能在后面被直接使用。

在介绍基于图这种数据结构的句法分析算法之前,先介绍几个概念。

关键符:在自底向上的算法中需要进行匹配的符号,可以是词汇类别符号,也可以是短语结构符号。例如,分析一个以 ART 开始的句子,则首先以 ART 为关键符,在语法中搜索右边以 ART 开头的规则,仍以语法 2.1 为例,则有规则(2-1-6)和(2-1-7)可以匹配上,

其他的规则不可用。为进一步分析下面的词,需要保留这个信息,同时为了标记已经匹配上的部分,需要在这两个规则上引入一个记号,即在已经匹配了的部分之后、没有匹配的部分之前加上一个小圆圈( ),则规则(2-1-6)和(2-1-7)变为:

$$NP \rightarrow ART \ \circ \ N \qquad (2\text{-}1\text{-}6')$$

$$NP \rightarrow ART \ \circ \ ADJ \ N \qquad (2\text{-}1\text{-}7')$$

如果 ART 的后续符号为 ADJ,那么上述规则(2-1-7′)可以扩展为:

$$NP \rightarrow ART \ ADJ \ \circ \ N \qquad (2\text{-}1\text{-}7'')$$

所有上述的规则都没有完全被匹配,只有当小圆圈位于一个规则的末尾时,该规则才被完全匹配,例如:

$$NP \rightarrow ART \ ADJ \ N \ \circ \qquad (2\text{-}1\text{-}7\ )$$

成分:句子中的一部分。

未完成成分:只与某个规则的一部分相匹配的成分。

完全成分:已完全与某个规则匹配的成分。

弧:从一个标识位置指向另一个标识位置的有向线段,上面标有重写规则。

活动弧:其上所标规则为部分匹配规则的弧。

非活动弧:被完全匹配的规则所标的弧。

图:一种数据结构,不仅能保留在句法分析过程中从句子导出的所有成分的记录,也能保留上述那种只被部分匹配上的规则的记录。图 2-2 为一个示例,描述了对成分 a little 进行分析时图结构中应保留的记录内容。

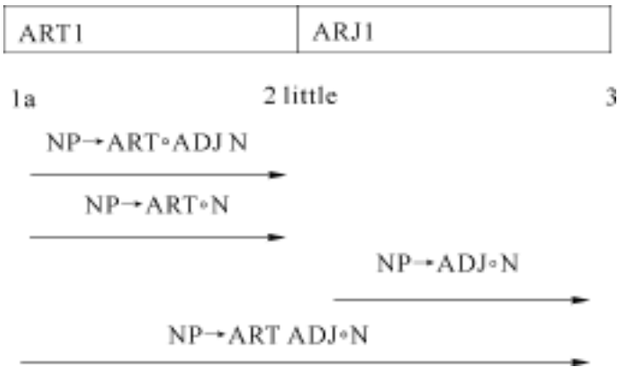


图 2-2　图结构内容示例

图中包含了如下的信息:有两个已完成的句子成分,位于位置标记 1 到 2 之间的 ART1 和位于 2 与 3 之间的 ADJ1。有四个活动弧,它们的含义分别是,第一行的弧暗示可能有一个 NP,它从位置 1 开始(关键符为 ART),而在位置 2 还需要一个 ADJ,ADJ 后面还需要一个 N;第二行的弧暗示可能有一个 NP,它从位置 1 开始(关键符为 ART),而在位置 2 还需要一个 N;第三行的弧暗示可能有一个 NP,它从位置 2 开始(关键符为 ADJ),而在位置 3 还需要一个 N;最后一行的弧表示可能有一个 NP,它从位置 1 开始,然后是一个 ADJ,而在位置 3 还需要一个 N。前三个弧都是利用关键符匹配直接得到的,最后一个则是通过弧扩展而来的。

弧扩展是在此句法分析算法中的一个基本操作,它是通过把一个活动弧与一个已完成成分相结合而实现的。经过弧扩展后或是产生了一个新的已完成成分,或者是对原活

动弧的扩展。所有已完成成分均保存在图内的表格中(以下称之为进程表),并加入到图中。整个过程构成一个弧扩展过程,可由下面的算法描述:

在位置 p_1 到 p_2 之间获取一个输入成分 C 后,可以执行如下的弧扩展算法:

第一步,把 C 放到图中的 p_1 到 p_2 之间。

第二步,对于任意从位置 p_0 到 p_1 且形如 $X \rightarrow X_1 \cdots C \cdots X_n$ 的活动弧,增加一条从位置 p_0 到 p_2 的弧,同时弧上的规则扩展为:$X \rightarrow X_1 \cdots C \cdots X_n$ 。

第三步,对于任意从位置 p_0 到 p_1 且形如 $X \rightarrow X_1 \cdots X_n$ C 的活动弧,在 p_0 到 p_2 的位置增加一个新的名为 X 的成分到进程表中。

弧扩展算法是本节要介绍的自底向上句法分析算法中的一个重要操作。有了这个算法,就可以建立基于图的自底向上的句法分析算法。以下仍然先用一个例子来说明该算法的执行过程。句子:

The large can can hold the water .

假设在词典中,上述句子中的词汇类别如下指定:

the: ART

large: ADJ

can: N, AUX, V

hold: N, V

water: N, V

对句子加上位置标记:

$_1$The $_2$large $_3$can $_4$can $_5$hold $_6$the $_7$water$_8$

以下开始算法,同时完成一个图。

第一步,初始化,图中的各个部分均初始化为空,没有弧,进程表中也没有已完成的成分。

第二步,按自底向上算法的原则,先在位置 1 取第一个词,读入 the,词汇类别为 ART,这是第一个出现的 ART,记为 ART1,它是一个完成成分,可以把它加入到进程表中;同时,以该成分为关键符匹配语法规则,并形成活动弧,显然与 ART 能匹配的有两个规则,因此向图中加入两个活动弧,从位置 1 开始,在位置 2 之前结束。

NP→ART ADJ N

NP→ART N

此时,就完成对第一个词的处理,位置标记进到 2。图由空变为图 2-3。

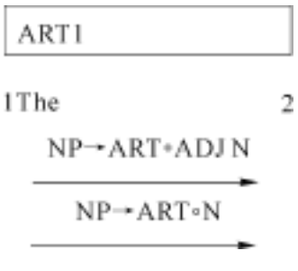


图 2-3 处理完第 1 个词

第三步,读入在位置 2 的词 large,词汇类别为 ADJ,是第一个出现的 ADJ,记为 ADJ1,它是一个已完成成分,将它加入到进程表中 ART1 的后面;同时以它为关键符匹配规则,可以找到一个活动弧 NP→ADJ N 加入到图中,从位置 2 开始,在位置 3 之前结束。此时,由于在第二步已有活动弧,当前为一个完成成分,因此可以尝试弧扩展。当前成分为 ADJ,因此可以对活动弧 NP→ART ADJ N 进行扩展为

NP→ART ADJ N

它仍为一个活动弧,没有生成完成成分,还不能向进程表中加入什么,位置标记后移至 3。

此时的图如图 2-4 所示。

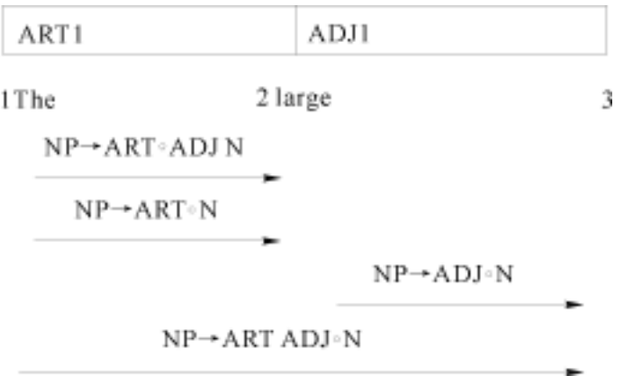


图 2-4　处理完第 2 个词

第四步，读入在位置 3 的词 can，它有三个可能的词汇类别：N，AUX，V，均为第一次出现的类别，分别记为 N1，AUX1，V1，由于都可能是完成成分，因此都加到进程表中，然后分别处理。

先取成分 N1，以它为关键符没有能匹配的语法规则，因此无需加活动弧。进一步考察能否进行弧扩展，显然有两个活动弧都可以，从 1 到 3 的弧 NP→ART ADJ N 可扩展为

NP→ART ADJ N

完成了一个从 1 到 4 的 NP 成分，记为 NP1。由于产生了新的完成成分，又需要以此成分为关键符来匹配规则，显然可以生成活动弧 S→NP VP，加入到图中，从位置 1 到位置 4。而从 2 到 3 的弧 NP→ADJ N 可扩展为

NP→ADJ N

完成了一个从 2 到 4 的 NP 成分，记为 NP2。同样，以新产生的完成成分为关键符来匹配规则，同样可以生成活动弧 S→NP VP，加入到图中，这个活动弧与 NP2 相应，是从位置 2 到位置 4。同时所得的两个完成成分都分别保存到进程表的合适位置上。

再取成分 AUX1，以它为关键符可以匹配一条规则，在图中增加从 3 到 4 的活动弧 VP→AUX VP；再检查能否进行弧扩展，结果是没有能扩展 AUX 的弧。

再取第三个可能的成分 V1，以它为关键符可以匹配一条规则，在图中增加从 3 到 4 的活动弧 VP→V NP；再检查能否进行弧扩展，结果是也没有能扩展 V 的弧，位置标记后移到 4。此时的图变为图 2-5。

第五步，读入在位置 4 的词，还是一个 can，它有三个可能的词汇类别：N，AUX，V，均为第二次出现的类别，分别记为 N2，AUX2，V2，由于都可能是完成成分，因此都加到进程表中，然后进行类似上一步的分别处理。下面只写出操作的结果：

为 N2 时，不增加活动弧，也没有弧扩展；

为 AUX2 时，增加从 4 到 5 的活动弧 VP→AUX VP，无弧扩展；

为 V2 时，增加从 4 到 5 的活动弧 VP→V NP，无弧扩展。

位置标记后移一位到 5，此时得到图 2-6 所示的图结构。

第六步，读入在位置 5 的词 hold，有两个可能的类别，分别记为 N3，V3。操作结果如下：

为 N3 时，不增加活动弧，也没有弧扩展；

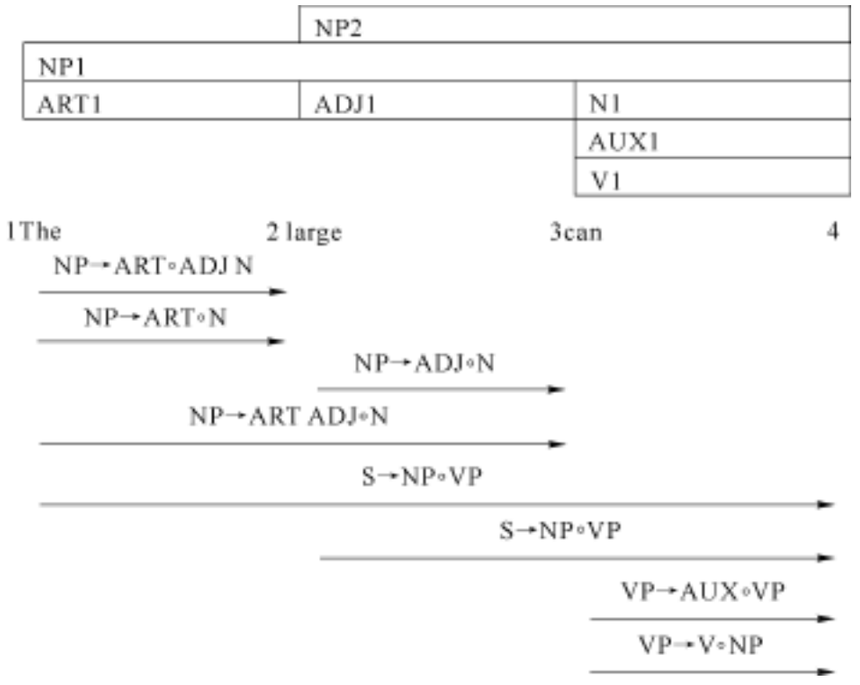


图 2-5　处理完第 3 个词

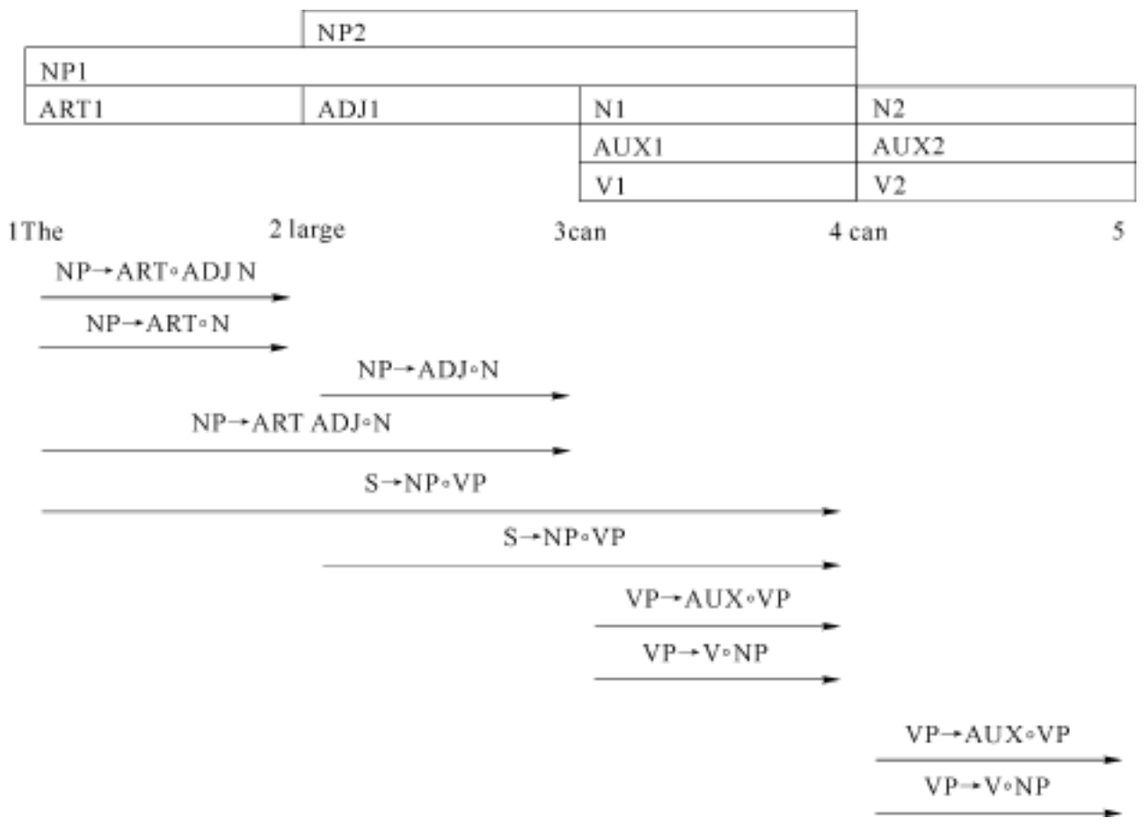


图 2-6　处理完第 4 个词

为 V3 时，增加从 5 到 6 的活动弧 VP→V NP，无弧扩展。

位置标记后移一位到 6，此时得到图 2-7。

第七步，读入在位置 6 的词 the，标记为 ART2 加入进程表中，同时，以此为关键符匹配后，可以增加两个从 6 到 7 的活动弧：

NP→ART ADJ N

NP→ART N

位置标记后移一位到 7，未达到句子的结束，继续。

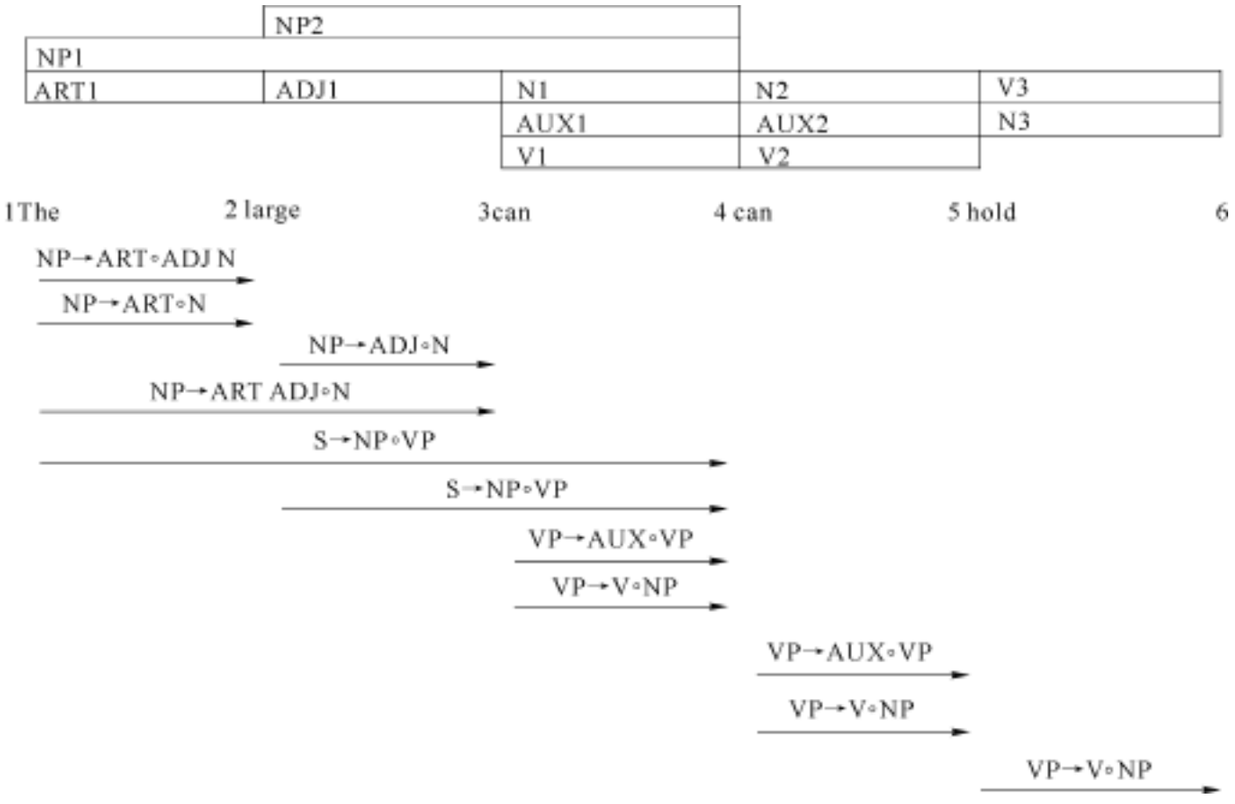


图 2-7　处理完第 5 个词

第八步,读入在位置 7 的词 water,**有两个可能的标记**:N4 **与** V4,**分别加入进程表中。**

对于 N4:

没有可加入的活动弧。

可扩展从 6 到 7 的活动弧 NP→ART N,**生成一个** NP,**记为** NP3(**从 6 到 8 包含** the water **两个词**),**要进一步对这两个新生成的成分进行处理。**

对于 NP3:

可以加入一个 6 到 8 的活动弧 S→NP VP。

可扩展 5 到 6 的活动弧 VP→V NP,**生成一个新完成的成分** VP(**从 5 到 8 包含** hold the water **三个词**),**记为** VP1,**立即对此成分进行处理。**

对于 VP1:

没有可加入的活动弧。

可扩展 4 到 5 的活动弧 VP→AUX VP,**生成一个新完成的成分** VP(**从 4 到 8 包含** can hold the water **四个词**),**记为** VP2,**立即对此成分进行处理。**

对于 VP2:

没有可加入的活动弧。

可扩展三个弧,其一是扩展从 3 到 4 的活动弧 VP→AUX VP,**生成一个** VP(**从 3 到 8 包含** can can hold the water **几个词**),**记为** VP3;**其二是扩展从 2 到 4 的活动弧** S→NP VP,**生成一个** S(**从 2 到 8 包含** large can can hold the water **几个词**),**记为** S1;**其三是扩展从 1 到 4 的活动弧** S→NP VP,**生成另一个** S(**从 1 到 8 包含** the large can can hold the water **几个词**),**记为** S2。**上述三个新成分中,** S1, S2 **已达到算法分析的最终目标,无需进一步处理,但** VP3 **还需要进一步处理。**

对于 VP3:

没有可加入的活动弧。

没有可扩展的弧。

位置标记后移一位到8,已达到句子的结束。至此,所有的成分均得到了处理,算法分析完成,最终生成图2-8。

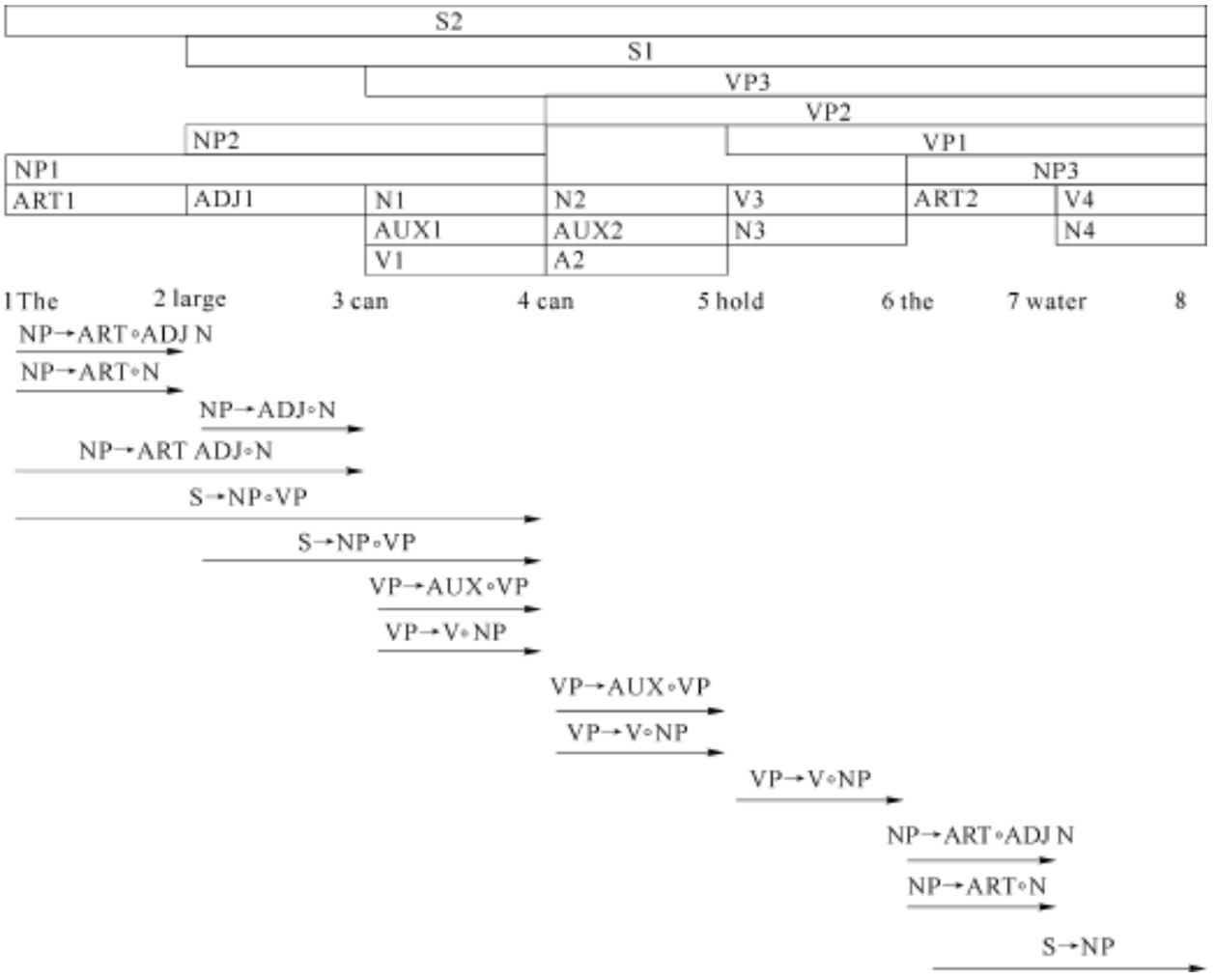


图2-8　最终的图

在图中,可以找到句子中所有可能的结构,最终组成的句子有多少个,就说明可能的句法树有多少棵。在本例中,只有1个是包含所有词的,因此只有一个句法树,可以很容易地从上述图中得到如图2-9所示的句法树。

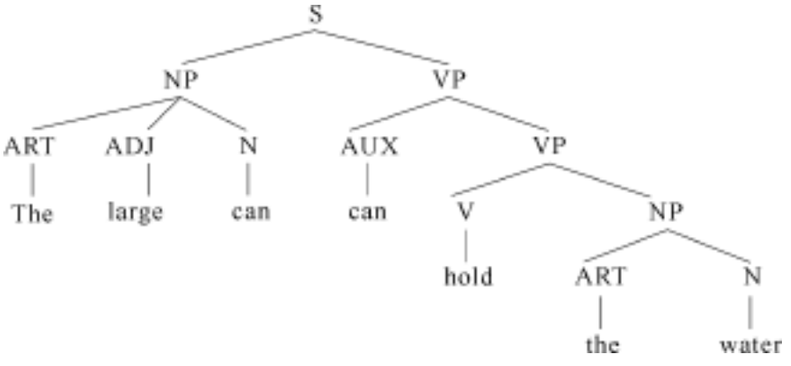


图2-9　句法树

算法描述如下:

初始化工作:给句子进行位置标记,初始化一个空图,进程表为空,无活动弧。

(1) 执行以下操作,直到位置标记达到句子末尾。

设当前的位置标记为 i,从输入句子中获取当前词汇类别C,把它加入到进程表中,并执行下一步。

(2) 以C为关键符,将匹配的活动弧加入到图中,即在寻找到语法中形如 $X \rightarrow CX_1 \cdots X_n$ 的规则后,在从 $i \sim i+1$ 的位置加入活动弧 $X \rightarrow C\ X_1 \cdots X_n$。

(3) 把C和前面出现的活动弧结合使用弧扩展算法,可能有以下三种不同的结果:

① 没有可结合的活动弧;

② 产生一个新的活动弧,如果原活动弧是从 $(i-m) \sim i$,则新的活动弧是从 $(i-m) \sim (i+1)$;

③ 产生一个新的完成成分,对每一个新的完成成分C,把它加入到进程表中。

(4) 把位置标记后移1,即 $i = i+1$,返回(1)。

基于图的句法分析同样也是个搜索过程,所以也可以采用宽度优先和深度优先两种策略。上述算法介绍的是深度优先的策略。如果在上述过程中,放入进程表的已完成成分的处理是堆栈方式(即后进先出方式),那么整个过程就是一个宽度优先的方式。此外,对于宽度优先方式,需要先把所有的已完成成分得到以后才开始对它们的处理过程。

可以看到,基于图这种数据结构,可以保存在分析过程产生的所有可能的成分,每个成分只需要生成一次就可以为整个的句法分析过程所用。这在提高算法分析效率的同时,也使这种算法不仅能分析句子结构,也可以用于分析任何的短语结构。

如上所述,在基于图的句法分析算法中,句子中的每一个成分在第一次构成后就保存下来了,不会再次构造同一个成分。这样它就比通常的搜索算法更有效。一个纯粹的搜索算法分析一个长度为 n 的句子需要 H^n 次操作,H 为一个与特定算法有关的常数。即使 H 不大,在句子长度增加时而导致的操作次数的指数增加也使算法不可用。而对于基于图的算法,在最坏情况(每两个成分都可能组成新的成分),其复杂性为 $K \cdot n^3$,n 为句子长度,K 为某个与算法相关的常数值,可能比 H 大,但是,这只是一个多项式复杂性。在一定的句子长度后,H^n 会远比 $K \cdot n^3$ 增长得快。例如,设 $H=10$,$K=1\,000$,在句子长度 $n=12$ 时,纯粹搜索的算法需要的操作次数是 $10^{12} \approx 1\,000\,000\,000\,000$,而基于图的算法是 $1\,000 \times 12^3 \approx 1\,728\,000$,前者是后者的500 000倍。

通常,自顶向下的算法由于具有较高的预期性,因此比自底向上的方式效率要高。比如,一个孤立情况下有多种可能语法类别的词,放在一个期望的句法结构中时,有一些类别就不存在了,这些类别很可能在后面的分析中也就不用考虑了。下面可以看一个具体的例子:

the can hold water

其中,can有AUX,V和N三种可能的句法类别。如果是自顶向下的分析,首先从(S)开始,第一步用(NP VP)重写(S),假设重写NP有三种可能性:(ART ADJ N),(ART N)和(ADJ N)。先取第一个,分析器判定the能否为ART,通过;再判断can能否为ADJ,失败。取第

二个,分析器再次判定 the 能否为 ART,通过;再判断 can 能否为 N,成功。可以看到,在分析过程中,can 作为 AUX 和 V 的可能性不会被考察,因为语法中没有规则预期在这个位置会出现 AUX 或是 V。而自底向上的算法则不同,像在上面的基于图的自底向上的算法中所看到的,它需要考虑所有三种可能性,三种可能的语法类别都在图中列出。

从上面的分析,似乎可以认为自顶向下的算法确实要比自底向上的方式效率高。但是在上面的分析中也可以看到,算法中判定 the 能否为 ART 重复了两次,这种重复判定一个成分的现象在自顶向下算法中是常见的,而在自底向上算法中则不会出现,一个成分只会判定一次。

因此,一个自然的想法是如何能结合多种方法以同时避免上述的两类问题。由于图这种数据结构能把句子中的每一个成分在第一次构成后就保存下来,使算法无需再次构造同一个成分,这样就能避免自顶向下算法常见的需要重复分析某个句子成分的问题;另一方面,再采用自顶向下的分析算法,这样就能够同时避免上述两类问题。

下一节,介绍把图这种数据结构和自顶向下算法结合起来形成的自顶向下的图分析算法。

2.4 自顶向下的图句法分析

和自底向上的图分析算法一样,自顶向下的图分析算法也是由两个部分来驱动的。一个是保存已完成成分的进程表,另一个是未完成的活动弧。其中基本的算法仍是弧扩展算法,通过把已完成成分与活动弧组合,形成新的已完成成分或新的活动弧。而它们的主要差别在于从语法中引入活动弧的方式上。在自底向上的算法中,是以一个完成成分为关键符,来匹配语法规则右边的第一个成分,匹配上的规则就是要引入的活动弧,其中小圆圈是加在第一个成分之后的;而自顶向下算法中则不同,首先以一个例子来说明——分析与上节中同样的一个句子:

$_1$The $_2$large $_3$can $_4$can $_5$hold $_6$the $_7$water$_8$

自顶向下算法总是从符号 S 开始的。

第一步,句子位置标记在 1。首先用 S 来匹配语法规则中左边的成分,找到的匹配规则就是引入的活动弧,而小圆圈是加在右边的开头。例如,S 匹配了规则:S→NP VP,则引入的活动弧为 S→ NP VP。这表明,为了完成句子,首先要找到一个 NP,NP 不是词汇类别符,所以继续以 NP 为关键符在语法规则中找左边为 NP 的所有可能的规则。假设有:NP→ART N,NP→ART ADJ N 和 NP→ADJ N,此时规则右边已有词汇类别符了,需要进入下一种操作,此时算法分析的状态如图 2-10 所示,还没有完成成分,只有几个活动弧。

第二步,读入 The。取 The 的词汇类别 ART,第一个完成成分,加入进程表,考察它能否和前面的活动弧进行扩展。显然有两个弧可以扩展,即

NP→ ART N 扩展为 NP→ART N 从 1 到 2

NP→ ART ADJ N 扩展为 NP→ART ADJ N 从 1 到 2

如图 2-11 所示,同时句子位置标记后移到 2。

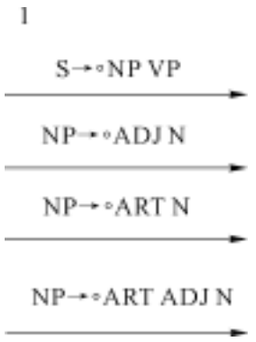


图 2-10　第一步

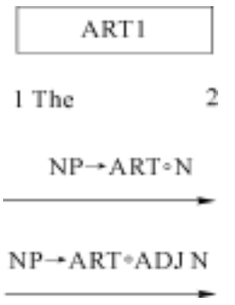


图 2-11　第二步

第三步,继续取 2 位的词 large。把新的成分 ADJ1 加入进程表,它把 NP→ART ADJ N 扩展为 NP→ART ADJ N 从 1 到 3,句子位置标记后移到 3。

第四步,取 3 位的 can,为 AUX1 时,没有扩展;为 V1 时,也没有扩展;为 N1 时,NP→ART ADJ N 扩展为 NP→ART ADJ N 从 1 到 4,即获得了一个新的完成成分 NP1(The large can),加入到进程表中,这时可把 S→ NP VP 扩展为 S→NP VP。此时为完成 S,需要找 VP,匹配规则右边可得两个规则:VP→ AUX VP 和 VP→ V NP。这时的算法状态如图 2-12 所示,句子位置标记后移至 4。

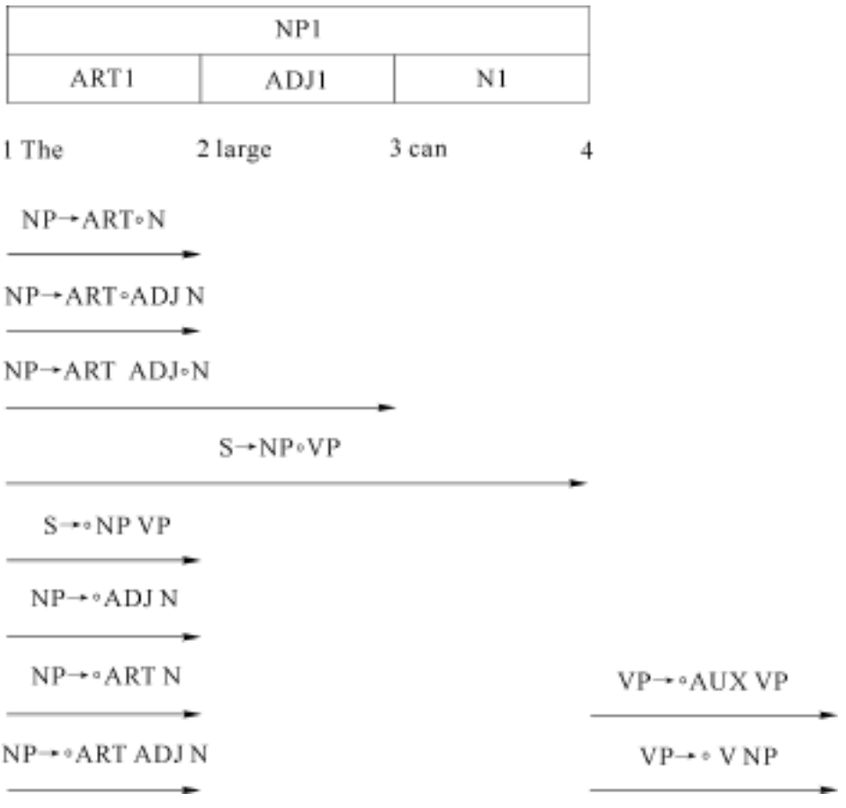


图 2-12　第四步

第五步,取 4 位的 can,为 AUX 时,把 VP→ AUX VP 扩展为 VP→AUX VP。此时,期望一个从 5 位开始的 VP。查找重写 VP 的规则有 VP→ AUX VP 和 VP→ V NP,为 V 时,把 VP→ V NP 扩展为 VP→V NP。此时,期望一个从 5 位开始的 NP。同样查找重写 NP 的规则有 NP→ ART N,NP→ ART ADJ N 和 NP→ ADJ N,而为 N 时,不能扩展任何弧。在这一步共生成了 5 个活动弧。句子的位置标记后移到 5 位。

第六步,取 5 位的 hold,当其语法类别为 N 时,没有弧可扩展;为 V 时,可以把

VP→ V NP扩展为从5到6的VP→V NP。此时,期望一个从6位开始的NP。查找重写NP的规则有NP→ ART N,NP→ ART ADJ N和NP→ ADJ N,它们是这一步生成的3个活动弧。句子位置标记后移至6。

第七步,取6位的the,可以扩展两个活动弧,把NP→ ART N扩展为NP→ART N,把NP→ ART ADJ N扩展为NP→ART ADJ N。句子位置标记后移至7。

第八步,取7位的water,为V时,没有弧扩展;为N时,可以把第七步形成的活动弧NP→ART N扩展为NP→ART N ,即得到一个新的完成成分NP从6到8,该NP可以把第六步形成的5到6的VP→V NP扩展成一个从5到8的完成成分VP。进一步,该VP又可以把第五步形成的活动弧VP→AUX VP扩展为一个新的VP,最终与第四步形成的S→NP VP扩展为S。得到了一个完整的句子,而此时句子的位置标记也已到句尾,因此完成了对该句子的句法分析,最终得到的进程表如图2-13所示。

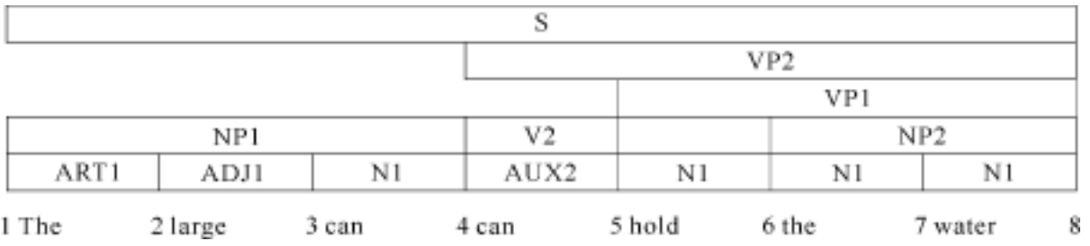


图2-13 最终的进程表

总结算法,可以描述如下:

初始化工作:给句子进行位置标记,初始化一个空图,进程表为空,当前活动弧为所有形如S→ $X_1 \cdots X_n$ 的弧。

(1) 执行以下循环,直到位置标记达到句子末尾。

当前的位置标记为 i,对当前所有活动弧小圆圈后的第一个符号C判断并执行下一步。

(2) 如果C为非终端符,则在语法中查找所有左边为C的规则,形成新的活动弧C→ $X_1 \cdots X_n$。

(3) 如果C为终端符,取当前位置标记的词,获取其所有可能的语法类别,看其中是否有和C相同的,如果有则可以使用弧扩展算法,并可能有以下两种不同的结果:

① 产生一个新的活动弧,如果原活动弧是从(i - m) ～ i,则新的活动弧是从(i - m) ～(i + 1);

② 产生一个新的完成成分,对每一个新的完成成分C,把它加入到进程表中,并判断其是否能扩展前面出现的活动弧。如果有则一直向前扩展。

(4) 位置标记后移一位, i = i + 1,返回1。

2.5 基于转移网络的句法分析

上面介绍的两类句法分析算法所用的语法描述都基于上下文无关的重写规则。在第

一章还介绍了描述上下文无关语法的另外一种方式——递归转移网络，本节介绍在用递归转移网络来描述语法时的句法分析算法。

为利用递归转移网络进行句法分析，需要在网络上进行一点修改，即在每个网络的可能终止的结点上向外伸出一条弧，命名为 POP 弧，表示该网络的分析已结束，如将第一章图 1-2 中给出的一个 VP 网络改写为如图 2-14 所示的形式，其中的 POP 只在结束结点时引出。此外，还划定了如表 2-2 所示的几个弧类别，弧类别是按弧上标记的不同来划分的。

图 2-14　VP 网络

表 2-2　几种类别的弧

弧　类　别	示　例	如何运用
CAT	N(某个词类别)	如果当前词的类别为该类别，则匹配成功，通过这条弧
WRD	of(某个词)	如果当前词为该词，则匹配成功，通过这条弧
PUSH	NP(非终端符)	当该名称的网络成功返回时
JUMP	只有 jump	遇到时则按弧指定的方向跳转
POP	只有 POP	当一个网络成功达到结束结点时

以下介绍一个基于递归转移网络的自顶向下的句法分析算法。实际上，基于递归转移网络是应该可以遵循与基于重写规则的算法相同的思路来建立的。首先可以由下列几个参数定义在算法中任意时刻句法分析器的状态：

当前位置——指向要分析的下一个词；

当前结点——分析中所使用的当前网络中的当前结点；

返回结点——在从当前网络中 POP 出来之后，为使分析继续而应该返回的其他网络中的结点。

在任意时刻，已知上述三个参数构成的状态，取当前结点所引出的弧作为当前要处理的弧。如果从当前结点引出了多个弧，那么任取其中的一条，根据该弧所属的上述类别的不同，算法可以执行如下的几种操作之一：

操作 1：如果该弧是 CAT 类的，并且当前位置与该 CAT 弧有相同的词汇类别，那么

① 把当前位置更新为下一个词；

② 把当前结点更新为该弧指向的下一个结点。

操作 2：如果该弧是 PUSH 弧，设其上标记的非终端符是 X，那么

① 把该弧指向的下一个结点加入返回结点列表；

② 把当前结点更新为与非终端符 X 相对应的网络 X 的开始结点。

操作 3：如果该弧是 POP 弧且返回结点列表非空，那么从返回结点列表中删除最后加

入的那一个结点,并把该返回结点作为当前结点。

操作 4:如果该弧是 POP 弧,返回结点列表为空且当前位置是句子的结束位置,那么句法分析成功完成。

下面举例说明该算法。语法用图 2-15 中的几个网络来描述。

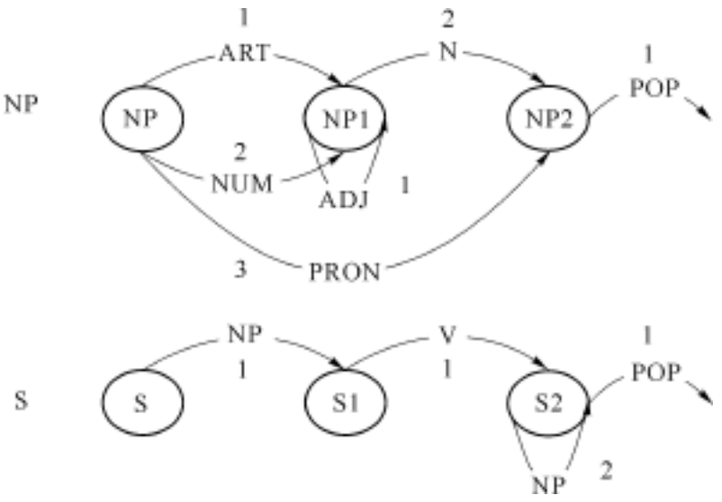


图 2-15　一个简单的语法

图中在弧上的 1,2,3 标记表示选取的次序,对算法描述没有实质影响。图中包括两个网络 NP 和 S,组成的语法与如下几条上下文无关重写规则组成的语法系统是等价的:

S→NP V (NP*)

NP→ART | NUM (ADJ*) N　(NUM 为数词)

NP→PRON

分析如下的句子:

The old man cried .

假设各词在词典中的语法类别为

ART ADJ N V

首先在句子上做位置标记如下:

$_1$The $_2$old $_3$man $_4$cried$_5$

分析过程的状态描述如表 2-3 所示。

表 2-3　句子 The old man cried 分析过程的状态描述

步骤编号	当前结点	当前位置	返回结点	下一步要处理的弧	步骤编号	当前结点	当前位置	返回结点	下一步要处理的弧
1	S	1	NIL	S/1	5	NP2	4	S1	NP2/2
2	NP	1	S1	NP/1	6	S1	4	NIL	S1/1
3	NP1	2	S1	NP1/1	7	S2	5	NIL	S2/1
4	NP1	3	S1	NP1/2					

在表中可以看出整个分析的过程。在第一步,状态的三个参数分别初始化为:当前结点在 S 网络的 S 结点,当前位置是 1 位 The,返回结点列表为空。下一步要处理的弧是从当前结点引出的弧,从网络中可以看到,S 结点引出的弧只有编号为 1 的一条,记为 S/1(意为从结点 S 引出的第一条弧)。对这条弧检查其符合前述四个操作中哪个操作的条

件,由于该弧上标的是 NP,所以它是一个 PUSH 弧,执行操作 2。

① 把该弧指向的下一个结点 S1 加入返回结点列表;

② 把当前结点 S 更新为 NP 网络的开始结点 NP。

当前位置没有发生变化,进入第二步,当前结点已经为 NP 网络中 NP,从它引出的弧有三个,不失一般性,按编号次序先选第一个弧作为当前要处理的弧,记为 NP/ 1。对这条弧检查其符合前述四个操作中哪个操作的条件,由于该弧上标的是 ART,所以它是一个 CAT 弧。那么,需要检查当前位置的语法类别是否与此相同。当前位置为 The,已知其语法类别为 ART,二者相同,因此执行操作 1。

① 把当前位置更新为下一个词 old,标记为 2;

② 把当前结点更新为该弧指向的下一个结点 NP1。

进入第三步,状态如表 2-3 所示,从它引出的弧有两个,不失一般性,按编号次序先选第一个弧作为当前要处理的弧,记为 NP1/ 1。判断后可知应执行操作 1。

① 把当前位置更新为下一个词 man,标记为 3;

② 把当前结点更新为该弧指向的结点,仍为 NP1(该完形弧是指向自身的)。

进入第四步,状态如表 2-3 所示,从它引出的弧有两个,不失一般性,按编号次序先选第一个弧作为当前要处理的弧,记为 NP1/ 1。该弧是 CAT 类的,判断后可知当前位置的词汇类别 (N)与该 CAT 弧标的词汇类别(ADJ)不同,不能执行操作 1;其他几个操作的条件也不匹配,因此按次序选第二个弧作为当前要处理的弧,记为 NP1/ 2。该弧是 CAT 类的,判断后可知当前位置的词汇类别 (N)与该 CAT 弧标的词汇类别(N)相同,执行操作 1。

① 把当前位置更新为下一个词 cried,标记为 4;

② 把当前结点更新为该弧指向的结点 NP2。

进入第五步,要处理的弧是 POP 弧,且此时返回结点列表中有一个结点 S1,因此执行操作 3,从返回结点列表中删除结点 S1,并把该返回结点作为当前结点。而其余的状态变量保持不变。

进入第六步,执行操作 1。

最后进入第七步,成功完成对此句子的句法分析过程。

从整个过程经过的弧可以得到句子的句法树。在上述分析过程中,在 S 网络中经过了 NP 和 V 两个弧,在 NP 网络中经过了 ART, ADJ 和 N 三个弧,因此可得如图 2-16 所示的句法树。

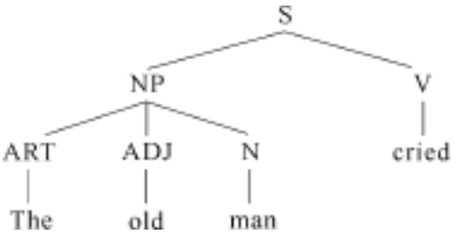


图 2-16 句法树

需要说明的是,上述的例子中,在 NP 结点恰好第一个候选的弧是句子的正确结构,所以整个句子的分析都很顺利。而在一般情况下,并不能保证第一次的选择就是正确的,因此,有必要对选择之前的状态进行备份以被后面发现错误后回溯之用。

最后,把基于转移网络的句法分析算法描述如下:

---

(1) 初始化状态,包括三个变量和第一个要处理的弧。

(2) 设已处理到句子的第 i 个位置,当前结点为 d_i。

(3) 处理从结点 d_i 引出的第 j 条弧(设从结点 d_i 引出的弧有 n 条),执行:

① 如果满足执行上述四个操作的某一个,则执行;

② 如果没有匹配的操作,则

如果 $j < n$,取第 $j = j + 1$,回到步骤(3)的起点;

如果 $j = n$,判断:

——如果 $i > 1$,执行 $i = i - 1$,d_i 回到上一个状态结点的 d_i,选择未用的一条弧回到步骤(2);

——如果 $i = 1$,结束,表明该句子不能用给定的递归转移网络分析。

在递归转移网络算法中,也可以引入类似于图的数据结构来记录分析过程中已经得到的句法成分,而这种结构通常称为完形子串表 WFST(Well-Form Substring Table),即表中记录的都是具有完整句法结构(例如一个句法类别、一个短语结构等等)的串。显然,在每执行完一个 POP 弧后,都能向 WFST 添加一个成分。

使用 WFST 的递归转移网络算法与基于图的自底向上算法具有相同的复杂性 $K \cdot n^3$,其中 n 为句子长度,K 为某个与算法相关的常数值。

小　　结

本章分别介绍了基于重写规则和基于递归转移网络描述下的句法分析算法。在基于重写规则的语法描述下,又分别介绍了利用符号串和图两种数据结构的算法。基于图的算法,由于图能保留更多的中间分析结果,从而减少重复,因此效率更高。而句法分析算法本身都可以看成是搜索过程,因而有深度搜索和广度搜索两种。从语法使用的方式来看,分为自底向上和自顶向下两种情况。

第三章　基于特征的语法及其句法分析

上一章作为例子的语法系统只覆盖了英语语法的很小一个子集。但是,同时,它也能生成一些不在合法英语之列的句子。

考察包含如下几个重写规则的上下文无关语法,其中,VP 的语法范畴可以有三种形式出现:

$$S \rightarrow NP\ VP \tag{3-1a}$$

$$VP \rightarrow V \tag{3-1b}$$

$$VP \rightarrow V\ NP \tag{3-1c}$$

$$VP \rightarrow V\ NP\ NP \tag{3-1d}$$

词 denied 和 disappeared 都具有句法类别 V,因此利用上述上下文无关语法可以判定如下两个句子是合法的:

The defendant denied the accusation . (3-2a)

The problem disappeared . (3-2b)

但是,在上述上下文无关语法下,如下两个不合乎语法的句子同样也被认为是合法:

The defendant denied . (3-3a)

The teacher disappeared problem . (3-3b)

为了能避免诸如(3-3a)和(3-3b)这样的不合乎语法的句子,一个方法是在语法类别 V 中划分更小的子类别。例如,把具有句法类别 V 的词依据其后能接成分的不同而分为几个子类:一个不能接名词短语的动词子类别,就是通常命名为不及物动词的那一类动词,它在句子中组成动词短语 VP 的方式如上述语法规则(3-1b),这类动词用 IV 表示;第二个子类别是可以接一个名词短语的动词,通常可以称这一类动词为及物动词,它在句子中组成动词短语 VP 的方式如上述语法规则(3-1c),这类动词记为 TV;还有一类动词后面可以接两个名词短语,这一类动词为可以带双宾语的动词,它在句子中组成动词短语 VP 的方式如上述语法规则(3-1d),用 DTV 来标记这一类。与子类划分相对应,需要把对应的规则(3-1b),(3-1c)和(3-1d)改写为

$$VP \rightarrow IV \tag{3-4b}$$

$$VP \rightarrow TV\ NP \tag{3-4c}$$

$$VP \rightarrow DTV\ NP\ NP \tag{3-4d}$$

这样,分别属于不同子类别的动词就可以分别利用各自的重写规则来判定其组成动词短语的合法性。

但是,由于是按照动词的差别进行子类划分,因此,一旦分子类后,IV,TV 和 DTV 就

成为独立的类别,它们之间的一些共性在规则中并没有得到体现。比如,与主语名词短语的人称数的一致性。

在一些自然语言中,存在着人称数的语法现象,例如,在英语中,大多数名词和动词都有单数和复数两种人称数。对名词来说,单数与复数的差别表明该词是在指代一个还是多个对象,如单数的 bird 表示一只或一种鸟,而其复数形式 birds 表示有多只鸟或多种鸟。而对动词而言,采用单数或复数是为了与充当其主语的名词的数的形态相一致。例如,动词 sing 在如下:

A bird sings . (3-5a)

Birds sing . (3-5b)

的两个句子中,在句子(3-5a)中,由于主语名词是单数 bird,所以采用单数形式 sings;而在句子(3-5b)中,由于主语名词是复数 birds,所以采用复数形式 sing。相反地,不满足这种数的一致性的句子是不合法的英语句子,例如:

A bird sing . (3-6a)

Birds sings . (3-6b)

是不合法的。但是这种不合法在前述的上下文无关语法规则中并不能体现,因为无论是名词的单数还是复数形式,总是名词;而无论动词是单数还是复数形式,也总是动词。因此,无论是句子组(3-5a)和(3-5b)还是(3-6a)和(3-6b),都是符合语法规则(3-1a)的。也就是说,在上述的上下文无关语法中,句子因为数的不一致而导致的不合乎语法是得不到识别的。但是,真正的英语句子是要求数的一致。为了使得形式语法也能处理这种现实的语法现象,一种方案是对原来的上下文无关语法进行修改,例如把语法规则(3-1a)用如下的两个规则来代替:

S→NP-SING VP-GING (3-7a)

S→NP-PLU VP-PLU (3-7b)

也即把名词短语和动词短语都分别划分出单数(SING)和复数(PLU)两个子类。而 NP-SING,NP-PLU,VP-SING 和 VP-PLU 都是在前面的上下文无关语法中没有定义的,需要重新定义。显然名词短语和动词短语的单、复数取决于短语中中心词的单、复数。以动词为例,可以把动词短语的规则(3-1b),(3-1c)和(3-1d)分别按单、复数来改写,但在前面,动词按其后接成分划分了三个子类 IV,TV 和 DTV,并因此已把规则(3-1b),(3-1c)和(3-1d)由重写规则(3-4b),(3-4c)和(3-4d)替代,所以,现在需要直接在重写规则(3-4b),(3-4c)和(3-4d)上分别按单、复数来改写上下文无关语法,这样可以得到如下的几条重写规则:

VP-SING→IV-SING (3-8b)

VP-SING→TV-SING NP (3-8c)

VP-SING→DTV-SING NP NP (3-8d)

VP-PLU→IV-PLU (3-8b′)

VP-PLU→TV-PLU NP (3-8c′)

VP-PLU→DTV-PLU NP NP (3-8d′)

从上面的规则中可以看到,三个子类 IV,TV 和 DTV 已经分别按单、复数再分类为六个子类。或者,从对等的角度来看,六个子类划分的结果是从动词的两个属性来对动词进

行划分的结果。一个属性是动词的后接成分,按这个属性值来划分动词,有三个值:IV,TV 和 DTV;另一个属性是数,有两个值:单数(SING)和复数(PLU)。按这两个属性划分子类分别可以解决两个句子语法判定的问题:其一是不同动词可能后接的成分不一样;其二是动词应该与其主语的中心名词在数上是一致的。而在解决这两个问题的同时,原来的语法规则的数目也成倍增加了。这可以做一个简单的计算:原语法(3-1a)～(3-1d)包括 4 条重写规则,而在扩展后,包括规则(3-7a),(3-7b),(3-8b)～(3-8d)和(3-8b′)～(3-8d′)共 8 条规则,并且上述的扩展并没有完全展开,还没有把 NP 按人称数划分后的新类别进行扩展。

而最重要的是,这些扩展后的规则之间具有很大的相似性,只在局部有所区别。比如规则(3-8b)与(3-8b′)之间、(3-8c)与(3-8c′) 之间以及(3-8d)与(3-8d′)之间都只有一个相同的区别——数的不同。因此,扩展后规则的冗余度较大。冗余度大的规则导致其推广能力不足,语法规则多,利用这样的语法系统进行句法分析从实现上来看是十分不经济的;同时,从揭示语言结构的目的来看也是缺乏深度的。为此,要改进这种简单的扩展方式。

解决这种冗余问题的一个方案是让语法类别可分解为几个单元部分。在前述的上下文无关语法中,语法类别是唯一的组成依据,并且语法类别没有内部结构,是不可细分为更小单元的原子,两个语法类别或者相同,或者完全不同,前述的几个语法类别 IV-SING,TV-SING,IV-PLU 和 TV-PLU 就不能从其结构中得出哪个类别和另一个类别在某方面相似,而差别只是由于某个属性值不同而导致的结论。但是,它们之间的相似和差别是明显存在的。比如:IV-SING 和 IV-PLU 的相似性在于它们都是不能直接接名词短语的动词;而其差别之处在于 IV-SING 的主语名词应该是第三人称单数,IV-PLU 的主语名词应该是复数。而 IV-SING 与 TV-SING 的相似性在于它们的主语名词应该是第三人称单数;其差别之处在于 IV-SING 后是不能直接接名词短语的,而 TV-SING 后则需要接名词短语。显然这四个语法类别的相似和不同之处是由两个特征来表现的:其一是主语名词的人称数,记为特征 AGR(AGR 特征实际上可以看成由两个更小的特征组合成的:一个是人称(PER)特征,另一个是数(NUM)的特征。PER 有三个取值 1,2,3,分别表示第一、第二和第三人称;而 NUM 有两个可能的取值:s 和 p,分别表示单数和复数。二者结合形成的 AGR 可以有几种不同的取值组合,如取到 3s 表明是第三人称单数,而取到 p 表示人称不限,但都是复数),在这里就假设这个特征只取 3s 和复数 p 两个值。其二是后接成分的类型(记为 VAL),无后接成分(itr)和接一个名词短语(tr)是这个特征可以取到的两个值。另外,用 POS 来记语法类别特征,它们四者在这一特征上是一致的,都是 V。这样,上面四种语法类别可以用如下的方法来标记:

$$\text{IV-SING} = \begin{bmatrix} \text{POS} & \text{V} \\ \text{AGR} & \text{3s} \\ \text{VAL} & \text{itr} \end{bmatrix} \qquad \text{IV-PLU} = \begin{bmatrix} \text{POS} & \text{V} \\ \text{AGR} & \text{p} \\ \text{VAL} & \text{itr} \end{bmatrix}$$

$$\text{TV-SING} = \begin{bmatrix} \text{POS} & \text{V} \\ \text{AGR} & \text{3s} \\ \text{VAL} & \text{tr} \end{bmatrix} \qquad \text{TV-PLU} = \begin{bmatrix} \text{POS} & \text{V} \\ \text{AGR} & \text{p} \\ \text{VAL} & \text{tr} \end{bmatrix}$$

上述表示中,POS,AGR 和 VAL 分别为三个特征,其后为对应的取值。从中可以很清晰地看到动词的这四个子类之间的关系和差别所在,如 IV-SING 和 IV-PLU 在特征 AGR 上的取值不同,而 IV-SING 与 TV-SING 是在特征 VAL 上不同等等。根据特征取值的差别,也可以比较哪两者比较相近。例如 IV-SING 与 TV-SING 就比 IV-SING 与 TV-PLU 更相近。并且由于一个特征反映一类语法现象,那么具有某个相同特征的两个子类在该特征所涉及的语法现象上就会表现一致,这样,不仅能比较相似性,而且能精确地判定这种相似性的原因。

下面用新的表示来改写规则(3-1a),如下所示:

$$S\rightarrow\begin{bmatrix}\text{NP}\\ \text{AGR}\quad a\end{bmatrix}\begin{bmatrix}\text{VP}\\ \text{AGR}\quad a\end{bmatrix} \tag{3-9}$$

NP 和 VP 下面的部分表示这两个语法类别的 AGR 特征的取值应该是一样的,因为它们有一个相同的特征取值参数 a, a 可以取到{1s,1p,…}。这样一条规则(3-9)就可以包含多种情况,其中也包含(3-7a),(3-7b)两条规则,这是规则(3-1a)所不具备的能力。

从上面的几个简单例子可以看到引入新的语法特征表示结构的一些好处,通过这种方法也能有效地解决上下文无关语法的冗余性,它实际上可以认为是对上下文无关语法的增强。

下面对这种上下文无关语法的增强形式进行详细的介绍。

3.1 特征结构与基于特征的语法

首先引入上述语法类别描述方法的名称——特征结构。

特征结构是一些特征的集合,这些特征通常反映了被该特征结构所描述的对象的某个方面的信息,每一个特征都可以有一个或多个特定的值与之相联系。一个特征结构 F 通常可以用如下的方式来记:

$$F=\begin{bmatrix}\text{FEATURE}_1 & \cdots & \text{VALUE}_1\\ \cdots & & \cdots\\ \text{FEATURE}_i & \cdots & \text{VALUE}_i\\ \cdots & & \cdots\\ \text{FEATURE}_n & \cdots & \text{VALUE}_n\end{bmatrix}\qquad n\geq 1$$

其中,FEATURE_i ($i=1,\cdots,n$)为特征名称;而 VALUE_i ($i=1,\cdots,n$) 为相应特征的取值,它也可以是一个内嵌的特征结构。例如用这种方式来描述一个词特征结构:

$$\text{put}=\begin{bmatrix}\text{POS} & \text{V}\\ \text{AGR} & \text{p}\\ \text{VAL} & \text{itr}\end{bmatrix}$$

这个结构中描述了 put 的三个特征:POS,AGR 和 VAL,取值分别为 V,p 和 itr。

对于单一特征 FEATURE_i,当其值 VALUE_i 无内嵌的特征结构时,称之为一个原子。

如 VAL = itr 就是一个原子，其中的特征取值可以是单个的值，也可以是一些值的集合，如 VAL = {tr, dtr}，只要它不包含其他特征项。由于每个特征通常有确定且相异的取值范围，因此，通常也把其取值称为原子，例如{tr}，{tr, itr}。在上一章中，用来进行句法分析的单一特征——句法范畴的值就是原子；而值{3s}就不是一个原子特征，它包括了两个更基本的特征 PER(人称)和 NUM(数)。例如，上面的 AGR = p 就是包含了如下的特征结构：

$$\begin{bmatrix} \text{PER} & \{1,2,3\} \\ \text{NUM} & \text{p} \end{bmatrix}$$

不过，通常由于这两个特征结合得比较紧密，因此也可把{3s}，{p}等作为原子对待。

特征结构也可以表示为函数的形式：

$$\text{F}(\text{FEATURE}_i) = \text{VALUE}_i \quad (i = 1, \cdots, n)$$

即 F 的特征 FEATURE_i 取值为 VALUE_i $(i = 1, \cdots, n)$。则上述的词特征结构可以表示为如下三个函数：

$$\text{put}(\text{POS}) = \text{V};\ \text{put}(\text{AGR}) = \text{p};\ \text{put}(\text{VAL}) = \text{itr}$$

特征结构有如下的几个特点：

(1) 特征结构中各个特征之间的次序是不重要，无需区分的。例如上例中 put 的特征结构与下面的

$$\text{put} = \begin{bmatrix} \text{AGR} & \text{p} \\ \text{POS} & \text{V} \\ \text{VAL} & \text{itr} \end{bmatrix}$$

结构是等价的。

(2) 特征结构使得被描述对象具有组织上的层次性。

前述 IV-SING 与 IV-PLU 的特征结构中，有两个特征的名称和取值都是完全一样的，把这两个共同部分拿出来，即为

$$\begin{bmatrix} \text{POS} & \text{V} \\ \text{VAL} & \text{itr} \end{bmatrix}$$

这个特征结构描述的对象就是 IV，显然 IV-SING 与 IV-PLU 是对 IV 的 AGR 特征进行互相排斥的具体化后产生的，即 IV-SING 与 IV-PLU 是 IV 的两个不相交的子集，具有层次结构，如图 3-1 所示。

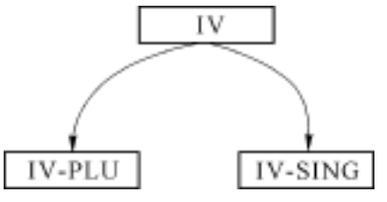


图 3-1　IV 的层次结构

同样，TV-SING 与 TV-PLU 的特征结构中，也有两个特征的名称和取值都是完全一样的，把这两个共同部分拿出来，即为

$$\begin{bmatrix} \text{POS} & \text{V} \\ \text{VAL} & \text{tr} \end{bmatrix}$$

这个特征结构描述的对象就是 TV，显然 TV-SING 与 TV-PLU 是对 TV 的 AGR 特征进行互相排斥的具体化后产生的，即 TV-SING 与 TV-PLU 是 TV 的两个不相交的子集，具有层次结构，如图 3-2 所示。

进一步把 TV 与 IV 的特征结构中的共同部分抽出来，有

$$[\text{POS} \quad \text{V}]$$

这个特征结构就是语法类别 V,显然 TV 与 IV 是对 V 的 VAL 特征进行互相排斥的具体化后产生的,即 TV 与 IV 是 V 的两个不相交的子集,具有如图 3-3 所示的层次结构。

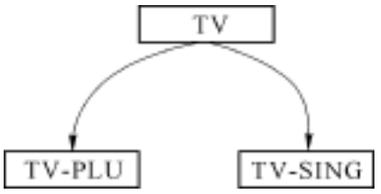


图 3-2　TV 的层次结构

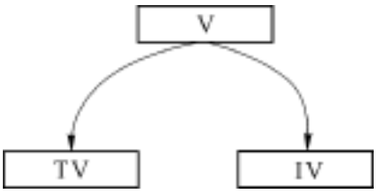


图 3-3　V 的二级层次结构

把上述三个层次结构综合起来,便是一个大的层次结构,如图 3-4 所示。

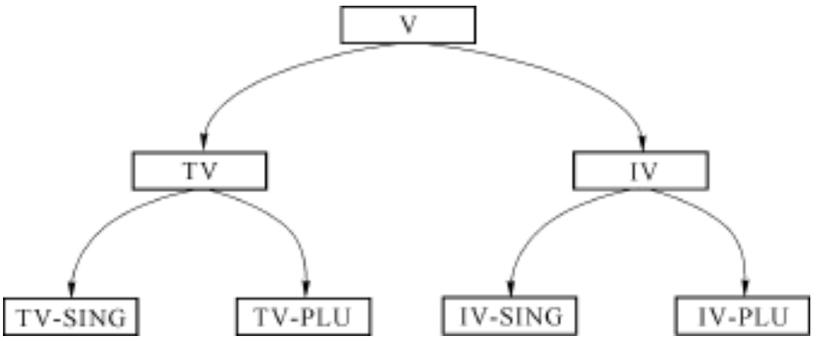


图 3-4　V 的三级层次结构

这种层次结构是特征结构所蕴涵的结果。

(3) 特征结构中特征的选择是由被描述的对象以及面向的任务所决定的。

(4) 特征结构除了上述的表述方法以外,还可以有其他的等价的表示方法,常用的包括以下两种:

① 直接无循环图 (DAG: Directed Acyclic Graph) 表示:在 DAG 中,有两类元素,结点和带标记的有向弧。弧上的标记为特征名称。结点有两类,一类是代表某个语法成分;另一类是特征的值。以上面的 put 为例,其特征结构的直接图如图 3-5 所示。

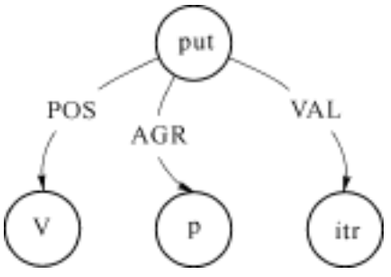


图 3-5　特征结构的直接图表示

在图中,只发出弧而没有弧进入的结点是根结点,本例中,从结点 put 发出了三条弧,但没有弧进入该结点,该结点为根结点。从这个根结点发出的三条有向弧,上面分别标着三种特征名,有向弧指向的结点的值就是该特征在此结构中的取值。此外,若从一个结点没有发出任何弧,那么该结点为叶结点,本例中,V,p 和 itr 三个结点均为叶结点。

② 括号表示方法:上面两种表示法均为平面图形,虽然直观,但不利于书写,而括号表示是线性的。仍以 put 为例,可表示为:(put(POS = V;AGR = p;VAL = itr))。其中,最外面的一对括号内为一个完整的特征结构;紧接着的 put 是特征结构的名称;后面的一对括号中就是该特征结构的内容,每个等号的前面是特征名称,而后面的是该特征在此特征结构中的取值。

在后面的介绍中,通常会使用线性的括号表示方法,但有时为了进行明确、直观的对比或其他特殊的需要也会使用其他表示方式。

前面已经提到,可以利用特征结构来改写上下文无关的重写规则,以获得能处理具有更多约束的语法来增强上下文无关语法。例如,把本章开始的几条上下文无关语法进行改写如下:

$$\begin{bmatrix} \text{S} \\ \text{AGR} \quad a \end{bmatrix} \rightarrow \begin{bmatrix} \text{phrase} \\ \text{POS} \quad \text{N} \\ \text{AGR} \quad a \end{bmatrix} \quad \begin{bmatrix} \text{phrase} \\ \text{POS} \quad \text{V} \\ \text{AGR} \quad a \end{bmatrix} \tag{3-1a′}$$

$$\begin{bmatrix} \text{phrase} \\ \text{POS} \quad \text{V} \\ \text{AGR} \quad a \end{bmatrix} \rightarrow \begin{bmatrix} \text{word} \\ \text{POS} \quad \text{V} \\ \text{AGR} \quad a \\ \text{VAL} \quad \text{itr} \end{bmatrix} \tag{3-1b′}$$

$$\begin{bmatrix} \text{phrase} \\ \text{POS} \quad \text{V} \\ \text{AGR} \quad a \end{bmatrix} \rightarrow \begin{bmatrix} \text{word} \\ \text{POS} \quad \text{V} \\ \text{AGR} \quad a \\ \text{VAL} \quad \text{tr} \end{bmatrix} \text{NP} \tag{3-1c′}$$

$$\begin{bmatrix} \text{phrase} \\ \text{POS} \quad \text{V} \\ \text{AGR} \quad a \end{bmatrix} \rightarrow \begin{bmatrix} \text{word} \\ \text{POS} \quad \text{V} \\ \text{AGR} \quad a \\ \text{VAL} \quad \text{dtr} \end{bmatrix} \text{NP NP} \tag{3-1d′}$$

上述各式中,a 为一个变量,在一个规则中,两个特征结构的相同特征都取相同的变量表明对它们在语法上的一致性要求。结构中的 phrase 为名词,表示该结构为短语的;word 类似,表示该结构是词的。

上述改写后的规则数量上没有增加,但是已经能够处理人称数的一致性、不同类别动词其后接成分也不同等问题。前面已经看到,这些问题在上下文无关语法下只能通过增加语法类别、增加规则才能解决,并导致语法规则数目的成倍增长和冗余度增大,不能抓住语言现象的特征。

上述语法对应的括号表示为:

$$\text{S} \rightarrow (\text{NP AGR ? } a)(\text{VP AGR ? } a) \tag{3-1a″}$$

$$(\text{VP}(\text{AGR ? } a)) \rightarrow (\text{V}(\text{AGR ? } a; \text{VAL} = \text{itr})) \tag{3-1b″}$$

$$(\text{VP}(\text{AGR ? } a)) \rightarrow (\text{V}(\text{AGR ? } a; \text{VAL} = \text{tr}))\ (\text{NP}) \tag{3-1c″}$$

$$(\text{VP}(\text{AGR ? } a)) \rightarrow (\text{V}(\text{AGR ? } a; \text{VAL} = \text{dtr}))\ (\text{NP})\ (\text{NP}) \tag{3-1d″}$$

改写后的重写规则显然已不再属于上下文无关语法,但是它们与上下文无关语法有着十分类似之处,可以通过某种方式回到上下文无关语法,在后面使用这种增强的上下文无关语法进行句法分析时,将可以仔细分析这一点。

3.2 基于特征的句法分析

上一章介绍的句法分析器在扩充后可以用来处理基于特征的句法分析。其中主要的扩展体现在匹配规则时,由于在语法规则中原有的上下文无关语法的单特征名称在这里

扩展成了具有内部结构的特征集,因此,在进行匹配时,除了结构名称的一致,还要匹配内部结构中每一个特征的一致性。下面以典型的图句法分析中的弧扩展算法为例来对这种扩展进行说明。

在弧扩展算法中,一个成分 X 可以把如下的未完成规则

$$C \rightarrow C_1 \cdots C_i \quad X \cdots C_n$$

扩展为

$$C \rightarrow C_1 \cdots C_i \; X \; \cdots C_n$$

在基于特征的语法中,上述规则中的每个成分都是特征结构,例如,一个未完成的规则如下:

$$(NP(AGR ? a)) \rightarrow (ART(AGR ? a))\ (N(AGR ? a)) \qquad (3\text{-}2\text{-}1)$$

其对应的简单值规则为

$$NP \rightarrow ART\ N$$

在原来的弧扩展算法下,一个句法范畴为 ART 的成分就可以把它扩展为

$$NP \rightarrow ART\ \ N$$

而无论这个 ART 的其他信息以及后面紧接着的是什么,如果后面一个成分是一个名词 N,那么又可以进一步扩展成一个完成成分 NP→ART N ,同样无论这个 N 的其他信息以及前面已出现的是什么,这正体现了所谓的上下文无关。但是,对于基于特征的未完成规则(3-2-1),要得到一个完成成分,就必须考察相关上下文的一致性。从规则可以看到,除了要考察是否分别是 ART 和 N 之外,还要判定它们的 AGR 特征是否相同,规则中对所有的 AGR 特征都用了一个相同的参数值,即表示它们的这个特征取值要相同。下面举例说明,如果有一个成分是:

$$(A(ROOT = A; POS = ART; AGR = 3s)) \qquad (3\text{-}2\text{-}2)$$

其中,ROOT 表示词根特征,A 的词根为 A 本身。为扩展(3-2-1)需要考察两部分:其一是特征结构中的 POS 是否为 ART,这可以确认;其二是特征结构中的 AGR 是否可以匹配。由于在规则(3-2-1)中 ART 的 AGR 特征取值不定,也就是说可以在所有可能的值中取,例如一个可能的取值集合为{1s,2s,3s,1p,2p,3p},这个取值集合与成分(3-2-2)中 AGR 的取值 3s 相匹配的就是 3s 本身。但是,同时要注意的是,由于在规则(3-2-1)中,三个结构中的AGR 特征是用相同的变量表示的,这也就表明,它们必须取相同的值,因此,成分(3-2-2)对规则(3-2-1)的扩展就包括这两部分的内容,在扩展结束后得到如下的结果:

$$(NP(AGR = 3s)) \rightarrow (ART(AGR = 3s))\ \ (N(AGR = 3s)) \qquad (3\text{-}2\text{-}3)$$

进一步考察,如果下一个成分是

$$(dog(ROOT = dog; POS = N; AGR = 3s)) \qquad (3\text{-}2\text{-}4)$$

成分(3-2-4)对规则(3-2-3)的扩展与成分(3-2-2)对规则(3-2-1)的扩展相同,除了要匹配 POS 是否为 N 之外,还要匹配 AGR 是否是规则(3-2-3)中所规定的。显然这个匹配是成功的,可以生成一个完成成分:

$$(NP(AGR = 3s)) \rightarrow (ART(AGR = 3s))\ \ \ (N(AGR = 3s)) \qquad (3\text{-}2\text{-}5)$$

即句子分析成功发现了一个 AGR 特征为 3s 的 NP。

从上面的匹配可以发现,前后两个成分是相互影响的。第一个成分把 AGR 的值通过未完成规则保留下来,并且对第二个成分进一步扩展规则产生约束。例如,如果成分

(3-2-4)换成如下情况:

(dog(ROOT = dog; POS = N; AGR = p)) (3-2-4′)

此时,成分(3-2-4′)就不能扩展规则(3-2-3)了,因为 AGR 特征不能匹配上。

因此,可以说,基于特征的规则在通过参数来传递上下文相关的信息,规则已不是完全的上下文无关。

上面是弧扩展算法在基于特征的扩展,整个的仅仅基于句法范畴的句法分析算法都可以进行这样的扩展。下面以图分析方法为例。句子:

He gives me a book .

最后形成的图如图 3-6 所示。

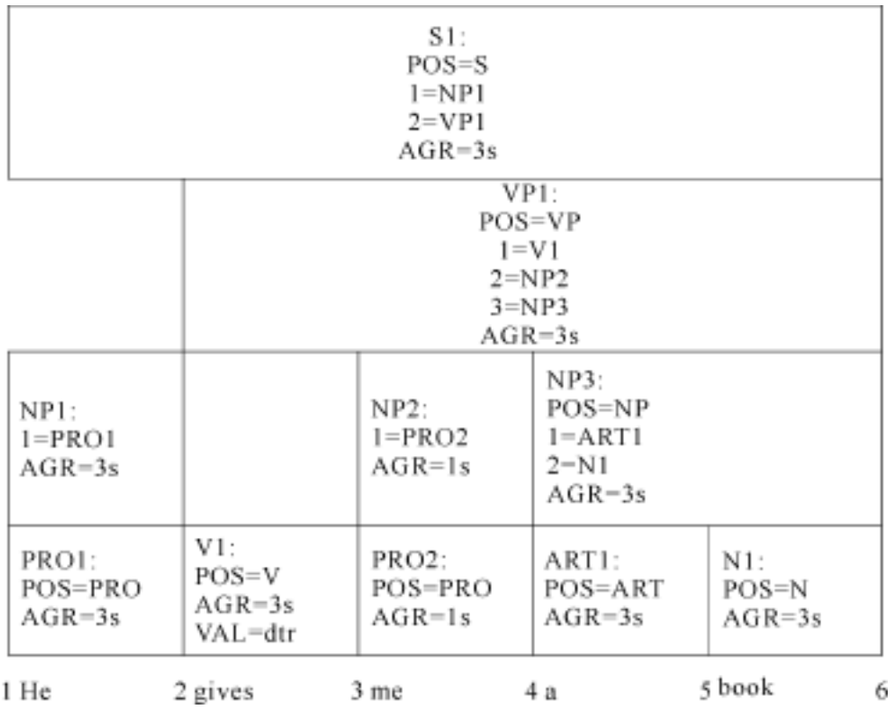


图 3-6 基于图的句子分析

可以看到,图的形成过程是,首先两个 PRO(代词)分别形成了两个 NP 结构,即 NP1 和 NP2;而后依据弧扩展 a book 形成一个新的完成成分 NP3。之前 V1 之所以没有独立或与 NP2 组成 VP 成分,是因为它的 VAL 特征为 dtr,所匹配的规则是(3-1d″),即需要后接两个名词短语才能组成一个 VP 完成成分。这样,在其后形成两个 NP 之后,就可以形成 VP1,说明一下 VP1 格中符号所表现的意思,其他格类似。第一行:VP1 是 VP;第二行:VP1 的第一个成分是 V1;第三行:其第二个成分是 NP2;第四行:其第三个成分为 NP3;第五行:该 VP1 是第三人称单数的。最后,NP1 与 VP1 具有相同的人称数,可以依据规则(3-1a″)组成 S。

3.3 基于扩充转移网络的句法分析

在上述的图分析器应用于增强的上下文无关语法之前,扩充转移网络(ATN)是第一

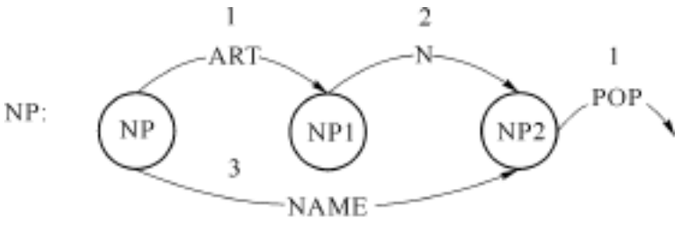
个基于增强的上下文无关语法的句法分析器。在第一章中,介绍了递归转移网络,作为描述上下文无关语法的另外一个工具。ATN 就是在递归转移网络的基础上进行扩充的,以便能处理增强的上下文无关语法,这与基于特征的重写规则对于原重写规则的扩展类似。

ATN 对于递归转移网络的扩充主要在两个方面,其一是在弧上除了原有的语法类别的约束要求之外,还增加一些对其他特征系统的测试功能,这样就达到了能够处理多特征约束的目的;其二是扩充转移网络通过设置寄存器可以保存句法分析的中间结果,这包括前述的完成成分以及其他一些特征信息。

首先来看看 ATN 是如何表示具有特征的句法规则的,然后通过一个例子来介绍如何使用 ATN 来进行句法分析。

图 3-7 是一个简单的描述 NP 的 ATN。

从图中可以看到两个部分。上面的有向图与前面所述的递归转移网络是一样的,不同的是增加了下面的一部分,其中对于每一个弧都有一行来说明对该弧除了弧上标明的判断外还需要进行什么测试,在测试通过后进行什么操作。例如,对于第一个弧,除了对通过该弧的词要判定是否是 ART(弧上本身就标有的)外,无需额外的测试。之后的操作有两个:其一是建立一个寄存器 ART,把通过的成分用一个变量名 * 来表示,并存储到寄存器 ART 中;其二是建立另一个寄存器 AGR,把通过成分的 AGR 值也赋予给寄存器 AGR。而对于第二个弧,有一个额外的测试,即判定新词的 AGR 与原有成分的 AGR 的取值是否有非空交集,如果没有,测试失败,不能通过该弧。如果有非空交集,则需要完成下一步的两个操作,其一是建立寄存器 HEAD,把该名词放入该寄存器;其二是把寄存器 AGR 的值更新为在测试中得到的非空交集。当从弧 1 和弧 2 经过后,对该网络的测试完成,并最终通过 POP 弧离开该网络,此时得到了一个 NP,其特征结构为



Arc	Test	Actions
1	none	ART:=* AGR:=AGR*
2	AGR^AGR*	HEAD:=* AGR:=AGR^AGR*
3	none	NAME:=* AGR:=AGR*

图 3-7　基于 ATN 的句法分析

$$(\mathrm{NP}(\mathrm{AGR} = \mathrm{AGR}_{\mathrm{ART}} \cap \mathrm{AGR}_{\mathrm{HEAD}}))$$

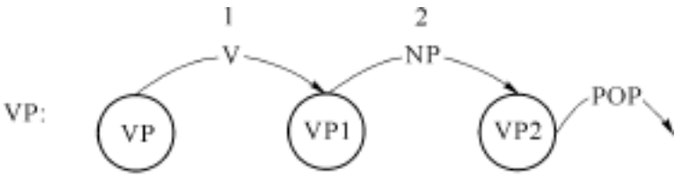
其中,$\mathrm{AGR}_{\mathrm{ART}}$表示寄存器 ART 中所保存的成分的 AGR 特征的取值;$\mathrm{AGR}_{\mathrm{HEAD}}$表示寄存器 HEAD 中所保存的成分的 AGR 特征的取值(后面的类似记号意义相同,将不再说明)。整个输出结构的 AGR 的值存储在 AGR 寄存器中,而寄存器 ART 和 HEAD 也分别存储了该 NP 的限定词和中心名词。

为了用 ATN 进行句法分析,再引入如下两个 ATN:

图 3-8 描述了规则(3-1c″),其 POP 输出结果为一个 VP,具有结构:

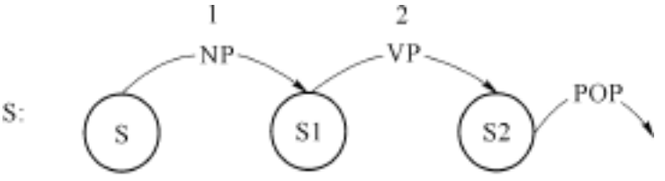
$$(\mathrm{VP}(\mathrm{AGR} = \mathrm{AGR}_{\mathrm{HEAD}}))$$

其中整个 VP 结构的 AGR 的值存储在 AGR 寄存器中,而寄存器 HEAD 和 OBJ 也分别存储了该 VP 的中心动词和后接名词短语。而图 3-9 描述规则(3-1a″),其 POP 输出结果为一个



Arc	Test	Actions
1	VAL?tr	HEAD:=* AGR:=AGR*
2	none	OBJ:=*

图 3-8　用 ATN 描述规则(3-1c″)



Arc	Test	Actions
1	none	SUBJ:=* AGR:=AGR*
2	AGR^AGR*	AGR:=AGR^AGR* PRED:=*

图 3-9　用 ATN 描述规则(3-1a″)

S,具有结构:

$$(S(AGR = AGR_{SUBJ} \cap AGR_{PRED}))$$

其中,AGR 的值存储在 AGR 寄存器中,而寄存器 SUBJ 和 PRED 也分别存储了 S 的 NP 部分和 VP 部分。

下面用一个例子说明如何使用 ATN 分析句子

Tom saw a cat .

的句法结构,其中与一般递归转移网络相同的部分在此不多叙述,主要阐明其中扩展的部分。

分析从 S 网络的 S 结点开始,由于弧上为 NP,进入 NP 网络。

匹配 NAME 弧,直接完成 NP 的分析,但此时与递归转移网络相比,要建立两个寄存器,一个是 NAME 用来保存当前词(Tom);另一个是 AGR1 用来保存该词的人称数(3s),由 POP 弧返回一个 NP1 回到 S 网络,进入 S1。

回到 S 网络后,再建立两个寄存器,一个是 SUBJ 存储 NP 结构(Tom);另一个是 AGR2 存储由 NP 网络返回的人称数(3s)。虽然这两组寄存器存储的内容相同,但完全在不同的层次上。分析在此之后进入 VP 网络。

除了首先判定下面的词的句法范畴是否是 V 之外,还要判定其 VAL 特征是否可以取到 tr,如果不能,那么就无需在此网络中继续了;如果能,则用 HEAD 存储该中心动词,AGR3 存储其人称数。

下一步再次进入 NP 网络,此时通过的是与 Tom 不同的另外一条路径。这里要建立

新的寄存器存储限定词、中心名词以及二者人称数的交集。如果二者人称数的交集为空，则得不到合法结构，把这个交集作为 NP 的人称数。

回到 VP 网络，得到 VP1，同时把返回的 NP 作为中心动词 HEAD 的宾语保存，把它的人称数作为 VP1 的人称数。

在完成 VP 网络后，最终回到 S 网络，把 NP1 与 VP1 的人称数取交集，如果非空，则得到 S，句子的人称数即为交集，VP1 存储在寄存器 PRED 中。至此完成分析。

3.4 基于合一的语法

在上面基于特征的句法分析中，有一个基本的操作——两个特征结构中 AGR 值的交。由于作为例子，在考察特征结构时主要注意了 AGR 等一两个特征。实际上，在基于特征的句法分析中，不同特征结构中同一特征取值集合的交是一个基本的操作。我们可以进一步把这种操作扩展到对不同的特征结构之间进行，这就是特征结构的合一运算。

定义　A 和 B 的合一运算(记为 $A\overline{\cup}B$)定义为：

(1) 若 A，B 均为原子，则

① 如果 $A\cap B$ 非空，则 $A\overline{\cup}B = A\cap B$；

② 如果 $A\cap B$ 为空，则 $A\overline{\cup}B$ 为空。

(2) 若 A，B 为两个特征结构，则

① 如果 A 中的任意一个特征 f，有 $A(f) = w$ (w 为原子值)，而该特征在 B 中没有定义，那么有 $A\overline{\cup}B(f) = w$；

② 如果 B 中的任意一个特征 f，有 $B(f) = w$，而该特征在 A 中没有定义，那么有 $A\overline{\cup}B(f) = w$；

③ 如果 A 中的任意一个特征 f，有 $A(f) = w$，且该特征在 B 中有 $B(f) = w'$(w'为原子值)，那么有 $A\overline{\cup}B(f) = w\cap w'$。

上述定义是递归的，可以对任意的复杂特征集实现合一运算。

例如：特征结构

$$(N1(POS = N; ROOT = fish; AGR = \{3s, 3p\}))$$

与特征结构

$$(N2(POS = N; AGR = \{3s\}))$$

的合一运算结果为

$$(N3(POS = N; ROOT = fish; AGR = \{3s\}))$$

而对于特征结构的 DAG 表示，可以根据图的特点将相应的两个 DAG 进行合一的算法，其算法也是递归的。下面是基于 DAG 的合一算法：

---

(1) 若 N1 和 N2 为两个叶结点，则考察二者的交集是否为空集，若非空，则创建一个新的结点，其值为 N1 和 N2 的非空交集；若为空，则没有新的结点产生。

(2) 若 N1 和 N2 不是叶结点，把从 N1 发出的标记特征为 F 的有向弧所指向的结点值

记为 N1F,并且

① 创建一个根结点 N;

② 如果从 N2 发出的弧没有标记为 F 的,则从 N 引出一条标记为 F 的弧,指向的结点值为 N1F;

③ 如果从 N2 发出的弧有标记为 F 的,其指向的结点值为 N2F,对结点 N1F 与结点 N2F 进行合一,如果它们都是叶结点,则执行(1);如果不是叶结点,则执行(2);

④ 如果从 N2 发出的弧有标记为其他在③中没有用到的特征时,从 N 引出所有这些特征,其值仍用 N2 中的值。

例如,如图 3-10 所示的两个 DAG 的合一运算结果如图 3-11 所示。

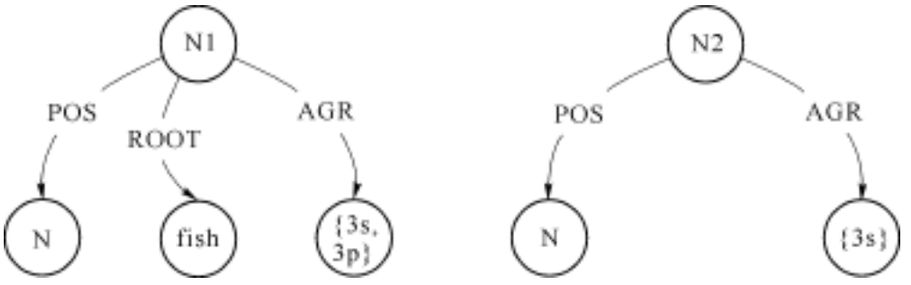


图 3-10　两个 DAG

在对运算进行扩展后,就可以把句法分析直接建立在特征结构的合一运算上进行。这样,甚至可以把整个语法系统看成是不同特征结构之间的约束集合,句法分析就是基于特征结构的合一运算。在这样一种观点下形成的语法系统通常称为合一语法。下面可以通过对比来看看合一语法的形式。

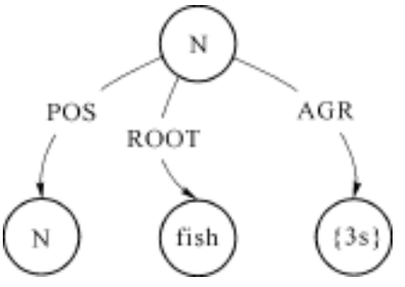


图 3-11　两个 DAG 的合一

例如,语法规则:

$$(S(AGR\ ?\ a) \rightarrow (NP(AGR\ ?\ a)\quad (VP(AGR\ ?\ a)$$

用合一语法来写,可以写成

$$X \rightarrow X_1\ X_2$$

POS0 = S

POS1 = NP

POS2 = VP

AGR0 = AGR1 = AGR2

其中规则部分是表示 X_0 可以被连续的符号序列 X_1 和 X_2 重写,它们分别受到四个特征结构的约束后就形成了上述规则。如果同样的 $X_0 \rightarrow X_1\ X_2$,换上不同的约束,就可以表示完全不同的规则,例如:

$$X_0 \rightarrow X_1\ X_2$$

POS0 = NP

POS1 = ART

POS2 = N

AGR0 = AGR1 = AGR2

此时,该合一语法表示的是名词短语的形成规则:

$$(NP(AGR\ ?\ a)\rightarrow(ART(AGR\ ?\ a)\quad(N(AGR\ ?\ a)$$

通常为了表述简单,可以把句法范畴写入到规则中,如上述合一规则可以表述为

$$NP\rightarrow ART\ N\qquad AGR0 = AGR1 = AGR2$$

则规则的形式结构与对规则的约束就完全分开了,从而能更有效地利用合一运算进行句法分析。

下面用基于特征结构的直接无循环图表示来介绍基于合一的句法分析。在介绍该算法之前,引入如下的记号:

给定规则 $X_0\rightarrow X_1,\cdots,X_n$,以 F_i 标记规则中第 i 个子成分的 F 特征,G_j 标记规则中第 j 个子成分的 G 特征,则有两类特征函数分别记为 $F_i = w$ 和 $F_i = G_j, SC_1,\cdots,SC_n$ 是分别与规则中 $X_1,\cdots,X_n$ 相对应的子成分,则以下算法建立一个满足所有特征函数的 DAG。

---

(1) 创建一个结点 CC_0 作为新特征结构的根结点。

(2) 把每个以 $SC_i(i = 1,\cdots,n)$ 为根结点的 DAG 复制到新的 DAG 中,根结点分别改名称为 $CC_i(i = 1,\cdots,n)$,从 CC_0 分别向 $CC_i(i = 1,\cdots,n)$ 发出一条有向弧,弧上标记为 i $(i = 1,\cdots,n)$。

(3) 对每一个形如 $F_i = V$ 的特征函数,寻找从结点 CC_i 发出的弧中是否有标记为 F 的,若有,设该弧指向的结点值为 N_i,则对 V 和 N_i 进行合一操作。

(4) 对每一个形如 $F_i = G_j$ 的特征函数,

① 如果存在从 CC_i 发出的标记为 F 特征的弧,设其指向的结点值为 N_i;如果同时存在从 CC_j 发出的标记为 G 特征的弧,设其指向的结点值为 N_j,对它们进行如下操作:

对 N_i 与 N_j 进行合一,把合一的结果赋给一个新创建的结点 X;

把原来指向 N_i 和 N_j 的弧均改为指向 X。

② 如果仅存在从 CC_i 发出的标记为 F 特征的弧,设其指向的结点值为 N_i;而不存在从 CC_j 发出的标记为 G 特征的弧,创建一个 CC_j 发出的标记为 G 特征的弧指向 N_j。

③ 如果存在从 CC_j 发出的标记为 G 特征的弧,设其指向的结点值为 N_j;而不存在从 CC_i 发出的标记为 F 特征的弧,创建一个 CC_i 发出的标记为 F 特征的弧指向 N_j。

下面给出一个算法工作的例子,首先给出例子中要用到的合一语法:

$$S\rightarrow NP\ VP\qquad AGR0 = AGR1 = AGR2 \tag{3-4-1}$$

$$NP\rightarrow ART\ N\qquad AGR0 = AGR1 = AGR2 \tag{3-4-2}$$

$$VP\rightarrow V\ ADJ\qquad AGR0 = AGR1 \tag{3-4-3}$$

考察分析如下的例句:

The man is luck .

句子中每一个词的特征结构的 DAG 表示如图 3-12 所示。

按自底向上的方法来进行。首先,匹配规则(3-4-2),约束为 AGR0 = AGR1 = AGR2,进入算法第(4)步执行①:创建一个新的根结点,并对 ART1 和 N1 的 AGR 特征合一,得到如图 3-13 所示的 DAG。

当第三和第四个词读入后,匹配规则(3-4-3),约束为 AGR0 = AGR1,同样进入算法的第(4)步执行③:创建一个从新的根结点到 V1 的 AGR 特征的弧,得到如图 3-14 所示的 DAG。

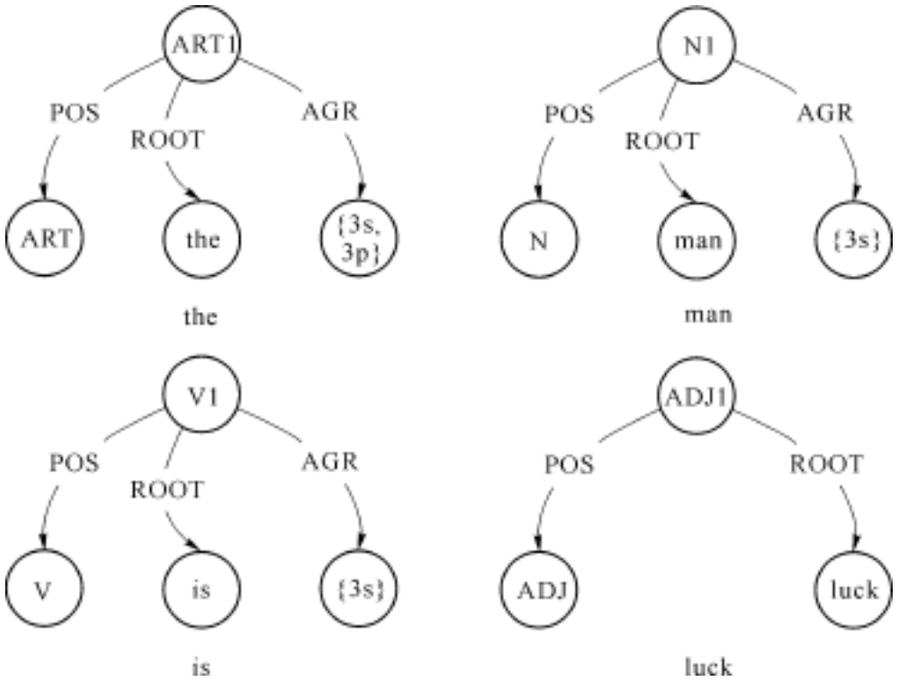


图 3-12　几个词的 DAG

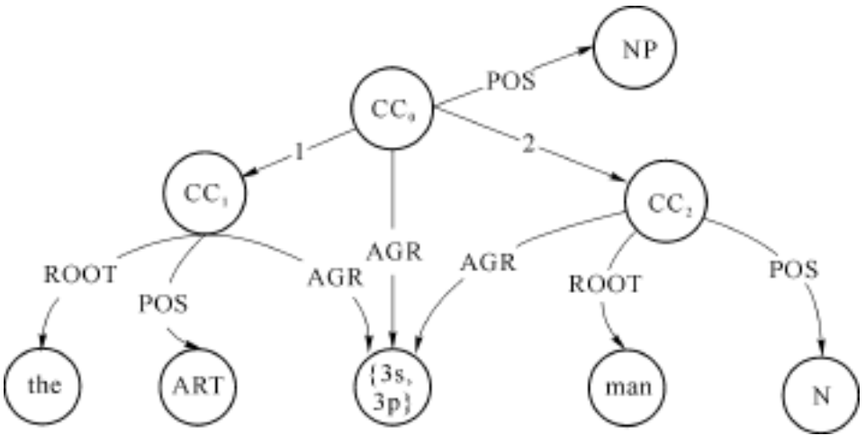


图 3-13　“The man”的 DAG

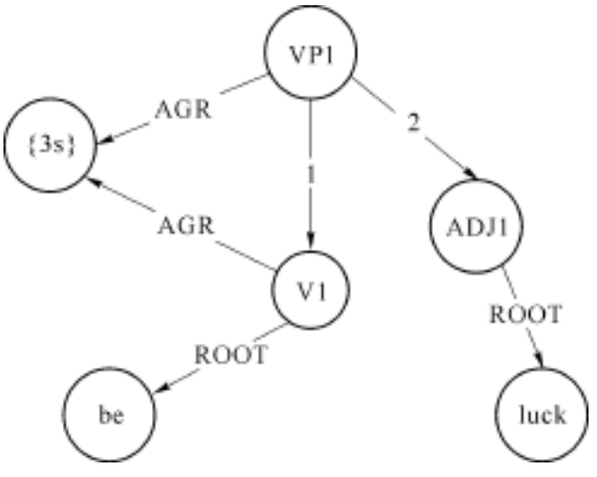


图 3-14　“is luck”的 DAG

最终,两个新的 DAG 匹配规则(3-4-1),约束为 AGR0 = AGR1 = AGR2,进入算法第(4)步的 ①,得到如图 3-15 所示的 DAG。

可以看到,基于合一的算法可以完成与增强上下文语法相同的句法分析任务。此外,

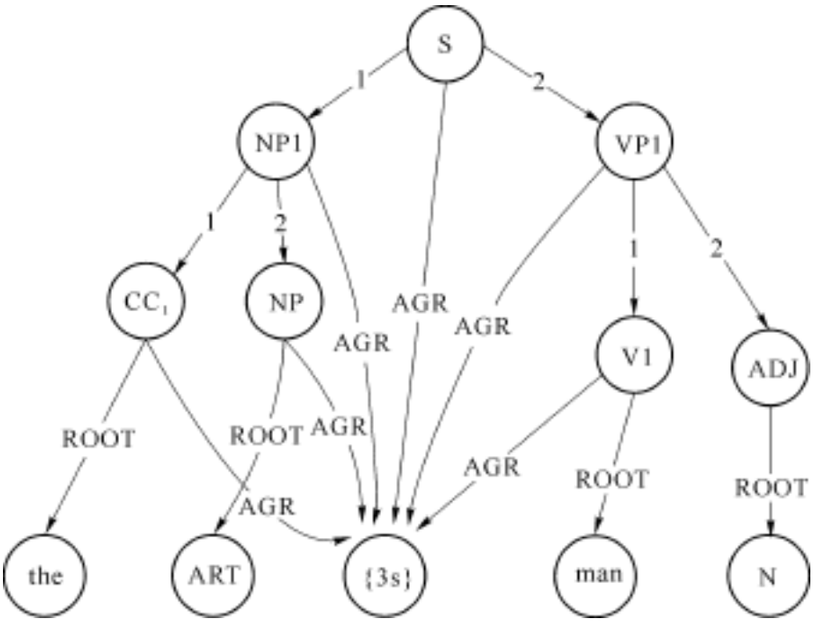


图 3-15 “The man is luck”的 DAG

基于合一语法表示方式的语法还比一般的增强上下文语法具有更多的优势,如:

合一方式的表示无需像一般的增强上下文语法那样只有依据句法范畴才能建立。例如,在英语中有一类结构为 NP be…,后面的部分可以是介词短语(如 He is in the house),也可以是名词短语(如 He is a student)或是形容词(如 He is happy),总之是具有某种名为 PRED 特征的成分就可以出现在 NP be 之后。这在标准的上下文无关语法中,需要用若干十分类似的规则,还不一定能完全表示出来,如:

VP→(V(ROOT be)) (NP PRED +)

VP→(V(ROOT be)) (PP PRED +)

VP→(V(ROOT be)) (ADJ PRED +)

可能才表示了部分情况;而在合一语法的表示方式下,上述三个规则可以表述为如下:

$X_0 \rightarrow X_1\ X_2$ POS0 = VP

POS1 = V

ROOT1 = be

PRED2 = +

即只要 X_2 具有特征 PRED 就可以出现在 be 之后形成 be…的结构,而无需指出 X_2 的句法范畴是什么。

上面利用特征结构增强上下文语法,降低了上下文语法的冗余度,从而提供了更大的推广能力。但是,对于上下文无关语法,还有一个问题是我们还不能理解的,即为什么有的上下文无关规则比另外一些更自然,例如,在下面的两个上下文无关语法中:

VP→V NP (3-4-4)

VP→P NP (3-4-5)

从语言学来看,规则(3-4-4)是合适的,而规则(3-4-5)是不能成立的。但是在上下文无关语法本身并没有能做出这种判定的任何依据。那为什么仍然采用规则(3-4-4)而认为规则(3-4-5)是不合适的呢?这就是上下文无关语法的任意性。一个比较显然的答案是在规则(3-4-5)中,箭头右边的两个成分不太可能组成左边的语法类别 VP。因为,直观

上,VP 中至少应该有一个 V,这种直观也正是命名 VP,NP 和 PP 的原因所在,在为短语结构命名时,应该以这种短语中必须包含的词汇类别来命名,即:如果在一种短语结构中必须包含至少一个 N,其他词汇类别可有可无,那么就称这种短语为 NP,VP 和 PP 类似。这个短语结构中所必须包含的相应词汇类别就称为该短语结构的中心词。中心词在自然语言中起着十分关键的作用,显然,引入中心词的概念就可以解释上下文无关语法中短语结构重写规则上的许多任意性。

小　　结

本章引入了特征结构,从多个特征来考察语法现象,并统一描述,这种语法是对上一章上下文无关语法的增强,但只考虑了人称、数以及动词的及物性等几种特征。在各种语言中语法特征各不相同,有的除此之外还有一些语法特征,如英语;而有些语言甚至连本章提到的这些特征都没有,例如,在汉语中并无人称、数的特征。这是在建立特征结构的时候需要考察的。

第四章　词 汇 语 义

在前面的两章中,介绍了如何从句子获得它的句法结构,通过句法结构,可以知道词是如何组合成句子的。但这还是远远不够的,自然语言处理的最终目的是使人类能直接用自然语言与计算机交流,其中句子的意义是要交流的主要内容,而与机器交流意义的首要问题就是任何给定的句子,其意义都应该是唯一确定的,也即是无歧义的。实际上,前述的句法分析也是解决这一问题的一个重要方面,例如,在句法分析后,已经可以处理如下的有歧义的句子:

I saw a boy with a telescope .

在这个线性结构下,可以得到两个不同的句法结构:

((I) (saw (a boy)) (with a telescope)) .

((I) ((saw (a boy)) (with a telescope))) .

分别表达了两个不同但确定的意义。但是,在句法分析后,并没有把所有自然语言的歧义完全消除,句法分析只是解剖了一部分的结构歧义,有些歧义单靠句法分析不能解析出来。如下面的例子:

Tom ran the machine . (4-1)

它的句法结构是唯一确定的:

(Tom (ran (the machine))) .

但是由于其中的单词 ran 在词典中至少有两种不同的意义,因而,其意义存在不确定性。这种不确定性主要存在于词汇意义层面,是由于词汇意义选择的不同而造成的。这种歧义是句子的句法结构所无法表现和解决的,是语义分析的一个重要方面。

再看看这个例子:

Every boy loves a dog . (4-2)

经过句法结构分析的结果是确定的:

((Every boy) (loves (a dog))) .

但在这同一个句法分析结果中,意义仍然是不明确的,其可以是下面两种意义解释中的任何一种:

(1) 有一只狗是每一个小男孩都喜欢的。(所有小男孩都喜欢同一只狗。)

(2) 每一个小男孩都有一只自己喜欢的狗。(所有小男孩都会有一只喜欢的狗,各人喜欢的狗可能是不同的。)

这两个可能的意义在已有的句法分析中是不能揭示出来的,需要作进一步的分析。这种分析需要揭示句子更深层的结构歧义,它是语义分析的另一个重要内容。

本章开始介绍为解决上述歧义而需要进行的语义分析的一些内容。将采用和句法分析一样的方式。首先描述词汇(无论是开始仅用句法范畴一个特征来描写,还是后面用特征结构),而后建立表征句子句法结构的表示,然后研究如何从最基本的词特征结构组合成短语、乃至更大的句子结构。而在语义分析中,同样先分析词汇语义的表示和分析,而后建立句子意义(结构语义)的描述方法,考察从最基本的词义结构逐级组合成句子意义的方法。在每部分同时介绍相应的解决词汇歧义和结构歧义的方案。

本章主要在词汇这一级,介绍词汇语义中的两个方面,其一是词汇语义的表示和组织;其二是介绍利用选择约束消除如上述句子(4-1)中出现的词汇歧义。

4.1 义　　位

词汇义是语义研究的首要对象。在上述句子(4-1)中,造成歧义的原因就是因为一个词有多个词义。在词典编撰中,称每一个词义为一个义项,在语义学中也称之为义位。例如在《现代汉语词典》中,"明白"有四个不同的意思:

内容、意思等使人容易了解;清楚;明确

公开的、不含糊的

聪明;懂道理

知道;了解

即是表明"明白"这个词包含四个不同的义位(义项)。

为特定句子中的某个词在它的多个可能义位中确定一个合适的义位是语义消歧的重要任务。一个可能的方案是依据上下文其他词的义位确定后产生的约束来进行歧义消除。这里主要考察不同义位之间搭配的可能性,例如在上述例句(4-1)中,machine 的义位确定为"机器"后,主要考察 ran 的几种可能的义位哪一个与 machine 的义位搭配更合适,即确定"跑"、"竞选"以及"操作"等义位与"机器"搭配的可能性,而确定这一点实际上是与语言本身无关的。如在本例中,利用世界知识就可以确定"跑机器"、"竞选机器"搭配的可能性要比"操作机器"小得多,因此可以确定 ran 在本句中应取义位"操作"。从上面的简单分析可以看到,不同义位的选取依赖于义位间的搭配情况,而义位间的搭配依赖于一些基本的义位间的关系。下面考察几种基本的义位间关系。

1.上下义关系

上下义关系是指在两个义位(分别称为上义义位和下义义位)间存在类属关系,下义义位是上义义位的子类,例如:"狮子"是"动物"的一个子类,即"狮子"和"动物"两个义位构成上下义关系,"狮子"是"动物"的下义位,"动物"是"狮子"的上义位。利用上下义关系可以进行如下的推理过程。对于某个义位 a,已有如下的判断:

X 是 a

如果存在另外一个义位 b,b 是 a 的上义位,则可以推出:

X 是 b

例如:由"X 是狮子"必然可以推出"X 是动物"。

注:

(1) 句子间的这种推理关系(一个句子表示的判定是另一个句子所表示的判定的必然结论)是后面考察句子语义时的主要研究对象,但并非所有的这种推理都需要利用上下义关系才能进行。

(2) 义位间的上下义关系也常用来进行概念定义,这即是所谓的属 + 种差的定义方法。例如在《现代汉语词典》中,狮子的定义是:哺乳动物。身长约 3 米,四肢强壮,有钩爪,掌部有肉块,尾巴细长,末端有一丛毛。雄狮的颈部有长鬣,全身长棕色毛。多产于非洲和亚洲西部。捕食羚羊、斑马等动物,吼声很大,有"兽王"之称。

在这个定义中,就是先指出狮子是一种动物(确定其所属),其后再给出狮子的一些性质特征(不同于所属类别其他子类的区别性特征)。

(3) 多个义位的上下义关系可以组成一个分类体系。图 4-1 就是一个简单的分类体系,上层的义位是下层义位的上义位。

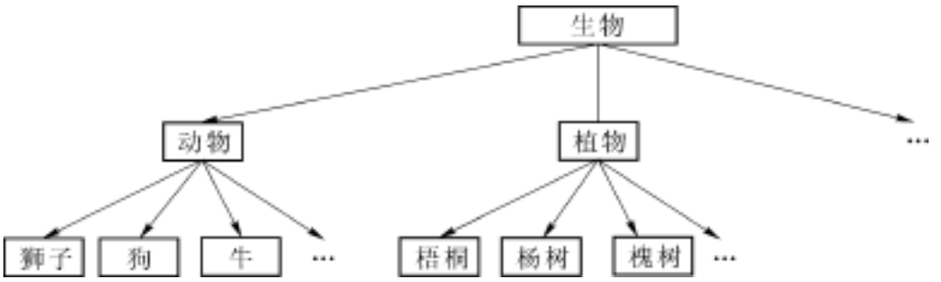


图 4-1　一个简单的分类体系

在上述的分类体系中,"狮子"、"狗"、"牛"等分别都与"动物"构成上下义关系,而这三个义位是处于相同层次的。

同时,也可以看到,上下义关系也不是绝对的,"动物"这个义位对于"狮子"而言是上义位;而对于"生物"这个义位来说,则是下义位,"生物"是上义位。

(4) 不单是表示名词事物的义位可以具有上下义关系,对于表示动作或事件的义位也可以构成上下义关系,如图 4-2 所示。

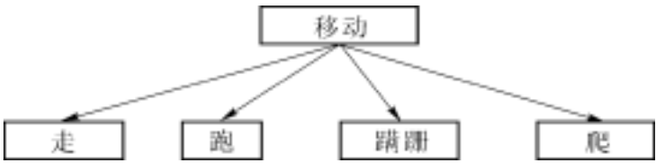


图 4-2　表示动作的义位构成上下义关系

很多歧义都可以通过上下义关系进行消歧。实际上,在例句(4-1)的消歧过程中,就隐含地用到了这种关系:义位"竞选"通常是选某个职位,"操作"通常作用在某个人工机构,"机器"正是一种"人造机构"的下义位,而非一种职位,因此不能用"竞选"与之搭配。

2 .整体-部分关系

这种关系表示在两个义位中,一个义位(部分义位)所表达的对象是另一个义位(整体义位)所表达的对象的组成部分。例如,"上肢"是"身体"的一部分。整体-部分关系与类属关系有许多类似的特点,例如,多个义位之间的部分整体关系也可以组成一个层次体

系,如图 4-3 所示。

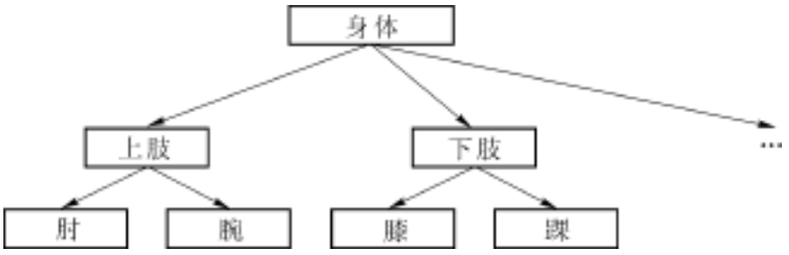


图 4-3　部分-整体关系组成的层次结构

下层的义位总是其上层义位的组成部分。可以利用这种关系推理,但是与上下义关系稍有不同,考察“狮子”与“动物”和“森林”之间的关系的差别,它们反映了两类关系的主要不同之处。

(1)“狮子”与“动物”是上下义的关系,“狮子”是“动物”的下义义位,对于上下义关系,可以说“狮子是一种动物”,即两个义位之间的这种关系可以用谓词“IS-A”来描述。

(2)“狮子”与“森林”是部分-整体的关系,“狮子”是“森林”的一部分,对于部分-整体关系,可以说“狮子是森林的一部分”,即两个义位之间的这种关系可以用谓词“IS-PART-OF”来描述。

同样,整体-部分关系也在很多消歧过程中起作用。例如,在句子

I saw a boy with my eyes .

中,my eyes 是 I 的一个组成部分,而不可能是 a boy 具有的,因此,不可能作为 a boy 的修饰成分。

3.同义关系

同义关系是指两个义位表达的意义是相同的,例如:计算机和电脑。但是或许没有任何两个词会是真正的完全同义。通常所说的同义是在认知意义下的同义,即两个词在认知意义下具有相同的指称。

4.反义关系

两个义位 A 和 B 是反义关系,如果“X 是 A”能表明“X 不是 B”。例如:“X 是高的”能表明“X 不是矮的”,那么义位“高”和“矮”就是互为反义的。两个互为反义的义位可能有两种情况。

一种情况是在两个反义义位之间不存在中间状态,它们是非此即彼的,即如果“X 不是 A”,那就能断定“X 就是 B”。例如,“男”和“女”就是这样的反义关系。

另外一种情况是在两个互为反义的义位之间还有中间状态,它们不是非此即彼的。这时,如果“X 不是 A”并不能表明“X 就是 B”。例如,“X 不是高的”并不能就断定“X 是矮的”,有可能既不“高”也不“矮”。

5.包含关系

有些义位包含了另外一些义位所指的对象。例如“父母”包含了“父亲”和“母亲”,“兄弟”包含了“哥哥”和“弟弟”等等。

上面只是列举了几种较为常见的义位之间的关系,义位关系的种类是十分丰富的。通过义位关系可以构成另一个在进行语义分析时十分重要的概念——语义场,因为对于不同义位间关系的研究,通常需要限定在某个语义场中才能进行并产生有意义的

结论。

4.2 语 义 场

前面介绍了不同义位之间的各种关系,通常,任何义位都会与其他的义位通过某一种或几种关系关联起来。这种由几个相互关联的义位构成的语义系统称为语义场。

语义场这一术语是德国学者伊普森(G .Ipsen)于 1924 年提出的;30 年代初,另一位德国学者特里尔(J .Trier)提出了系统的语义场理论。一种语言的所有义位的集合是该语言最大的语义场,这个最大的语义场还可以分成较小一些的子场,子场又可以继续分成更小的场,这样一层层分下去,会在某个时候得到不能再分的最小语义场。

语义场的确定具有本体论性质,也即是说,一个语义场的组成并不是随意的,而是要求场内的各个义位之间是互相联系、互相制约和互相规定的,这种义位之间的相互关系不是词语本身内在的,而是由其所指的外部世界中的关系决定的。如"师傅、徒弟"两个义位构成一个语义场,它们二者共同规定了现实世界中的一种关系。而单独的"师傅"或者"师傅、儿子"两个义位集合都不是语义场,因为前者只包含一个不能自我规定的义位,"师傅"这个义位如果没有"徒弟"这个义位的话就难以得到说明,这二者是互相规定的;而后者包含的两个义位并不是互相联系、互相制约的,二者没有必然的联系,对这样两个义位进行对比得不到有意义的结果。因此在义位分析前应该确定合适的语义场,尤其以能确定适当的最小语义场为最好。因为语义场越小,就越能明确义位之间的关系。例如:"上、下"是一个语义场,"上、下、左、右"也是一个语义场,而在"上、下"这个更小的语义场中就更容易明确上、下之间的关系。

由于义位相互关联的方式不同,因此,构成的语义场也有多种。语义场的种类也分别与义位的各种关系相对应,贾彦德在其《汉语语义学》中给出了几类常见的语义场。

1 .分类义场

分类义场是一种最常见的义场,在这种义场中的各个义位构成了对某种对象的类别化分。最简单的分类义场可以只包含两个义位,称为二元义场,如"中医、西医"是对医疗方式的两分,"城市、乡村"是对地区类别的两分等等;也可以是多元的,如三元的"陆军、海军、空军",四元的"太平洋、大西洋、印度洋、北冰洋",五元的"非洲、亚洲、欧洲、美洲、大洋洲",以及更多元的情况。

分类义场实际上与义位的上下义关系紧密相关,例如,五元的"非洲、亚洲、欧洲、美洲、大洋洲"义场中的每一个义位都是与"五大洲"这个义位具有上下义关系,即这五个义位是同一个义位的下义义位。通常可以这样来判断几个义位是否构成了一个合适的分类义场。

前面介绍了多个上下义关系可以构成一个层次体系,因此分类语义场也可以是层次的,大类之下分小类,小类之下又可以分出更小的类。

2 .部分义场

通常,部分义场是对象的各个组成成分所构成的语义系统,义场中的各个义位能共同组成一个整体。例如:"头、颈、躯干、四肢"是一个语义场,四个义位共同组成了一个动物体。

3.顺序义场

顺序义场是由几个有顺序关系的义位组成的,各个义位之间存在一定的次序。例如,“星期一、星期二、星期三、星期四、星期五、星期六、星期日”就是由一周的几天组成,而这几天是按时间顺序进行的。这种顺序关系不局限于时间,如“优、良、及格、不及格”就是关于等级次序的顺序义场。

4.关系义场

关系义场中的义位表现了不同的角色,这些角色是相互依赖而存在的。例如“教师、学生”语义场中,包含两个角色,这两个角色是互相定义、互相依赖而存在的,构成一个自足的语义系统。

5.反义义场

反义义场通常只包含两个义位,它们是相互对立的关系,而没有中间的过渡意义。例如“男人、女人”是非此即彼的。

6.两极义场

两极义场与反义义场很类似,通常也只包含两个义位,二者是相互对立的意义的两极。但与反义义场不同的是,在意义的两极之间存在中间的过渡状态,例如两极义场“穷、富”,中间存在着穷富过渡状态。因此,也就具有可比较性,例如在“大、小”这个义场中,可以进行3比2大、1比2小等等的比较。

7.部分否定

部分否定与反义义场、两极义场都有相似之处,场内的诸义位处在某种相互否定的状态之中,但是,这种否定不是完全对立的。简单的部分否定义场是二元的,其中一个义项是另一个义项的部分否定,例如“必然、可能”,“必然”的完全否定是“不可能”,“可能”只是它的部分否定。部分否定义场也可以是多元的,例如,三元的有“前进、停止、倒退”以及四元的“热、温、凉、冷”等等。

8.同义义场

同义义场中的义位其基本义都是相同或相近的,差别只在于附加义。例如,“警告、正告”、“掩饰、粉饰”等等。通常并不把具有完全相同意义的词即等义词构成的系统看成是一个语义场,例如“西红柿、番茄”,二者完全同义,并不构成一个有意义的义场。

9.枝干义场

枝干义场通常包含一个总的义位,又包含一个或多个特殊的义位,例如“打、拍、捶”,“打”是一个总的义位,而“拍”和“捶”是两种打的特殊方式(一个是用手掌打,一个是用拳头打)。

10.描绘义场

描绘义场中的义位通常是反映某些性质、行为或状态的,它们之间的差别体现在感情色彩、形象上,例如“湿漉漉、湿淋淋”、“白茫茫、白皑皑、白花花、白蒙蒙、白晃晃”等等。

可以看到,通过把一些义位放在相同的语义场中,可以描述不同义位间的一些关系。但是,要深入揭示语义场中各个义位之间关系的实质,揭示为什么一些义位和另一些义位之间存在特定的语义关系,还需要对义位进行进一步的分析。

由于义位本身的数量是十分巨大的,不可能对这些义位逐个进行分析。为了分析这些十分多样的语义关系,通常的一种方法是对义位进行抽象、概括,得到较一般化的概念,

这个一般化之后的概念通常称为原语。例如:“走、跑、跳、爬”几个义位一般化为“移动”,“移动”就是一个原语概念,它能表征几个义位中的某个共同特征,从而揭示几个义位间之所以存在关系的实质。

但是,在这种一般化的过程中,如果抽象程度不高,就会导致原语数量过多,那么由义位到原语的抽象就意义不大。比如,极端情况,每一个义位对应一个原语,那么原语就完全失去了其存在的意义。而如果原语抽象程度太高,这时虽然原语数量少,但容易导致在由义位概念化为原语时丢失的信息过多。比如,在极端情况,所有动词对应一个原语ACT,那么不同动词之间的差别就完全得不到体现,这样的抽象不比句法理论中的句法范畴 V 更有意义。并且,一个明显的问题是当利用原语表示的结构来生成句子时,其可能生成的句子数量是巨大的。

目前有两类提出原语概念的途径,一类是基于语义特征的方法;另一类是基于原型的方法。

4.3 语义特征

基于语义特征的方法其关键在于找到合适的有限数量的语义特征,所有义位都能利用这些语义特征进行组合而得到。这种方法的本质是对义位进行适当的分解。义素分析法是一种常见的对义位进行特征分解的方法。(在后面,把语义特征和义素混合使用。)

义素是比义位更基本的语义单位,是义位的组成成分。一个义位通常由多个义素组合而成。例如,哥哥这个词只有一个义项,组成该义位的义素包括人、亲属、同胞、年长和男性等。

早在 20 世纪 40 年代,丹麦的结构主义语言学家叶姆斯列夫(L .Hjelmslev)就提出了义素分析法的设想。50 年代,美国人类学家朗斯伯里(F .G .Lounsbury)和古德内夫(W .H .Goodenough)受到雅可布逊(R Jakobson)在音位学中提出的区别性特征分析方法的启示,在研究亲属词的含义时提出了义素分析法。但义素分析法引起语言学家的重视还是在 60 年代以后,美国语言学家卡兹(J .Katz)和福多(J .A .Fodor)在其解释语义学理论中,为了提供语义特征而引入了义素分析法。

义素分析法就是通过对不同义位的对比,找出它们所包含的义素的方法。这里遇到的第一个问题就是用来进行对比的不同义位如何选择,这就要用到前面介绍的语义场,只有对处在相同的最小语义场(至少也是较小语义场)中的义位进行义素分析才可能得到有意义的结论;否则,如果把“桌子”和“精神”不是处于较小语义场中的两个概念进行义素分析是得不到有意义的结果的,即分解不出本质共同的义素。

在确定了语义场之后,就可以进行有意义的义位对比,对义位进行特征分解,把该义位用更基本的意义单位来表示,发现其中的义素。例如,对“男人、女人”这个义场,首先把这两个义位进行分解,可得“男人 = 人 + 男性”,而“女人 = 人 + 女性”。在两个义位的分解式中,“人”是两个义位都共有的一个特征,可以算一个义素;另一个特征二者是不相同的,但是稍微注意就可以发现,实际上是对于“性别”这一特征的两个取值,这样就可以得到另一个特征“性别”。“男人、女人”这两个义位的分解也就可以用义素来表示 (如表 4-1

所示)。

表 4-1　义位的分解(1)

义位 \ 特征	人	性别
男 人	是	男
女 人	是	女

这样,“男人、女人”这两个义位就可以用两个义素“人”和“性别”来完全描述。而这两个义素又可以作为其他义位的分析单元。例如,在分析“男孩、女孩”这两个义位时,人、性别仍然是两个基本的义素,这样就可以通过共同的义素来揭示不同义位的差别所在,以及导致这种差别更精细的原因。

基于义素分析方法提取特征原语方法的吸引人之处就在于它提供了分析上下义关系、同义关系等各种关系的基础,明确在义位间之所以存在这些关系的更深入的原因。例如,通过义素分析,可以明确“鳏夫”和“男人”为什么是上下义关系,导致其出现这种关系的关键所在。如表 4-2 所示。

表 4-2　义位的分解(2)

义位 \ 特征	人	成年	性别	婚姻状况
鳏 夫	是	是	男	未婚
男 人	是	是	男	不定(未婚或已婚)

从表中可以看到,“鳏夫”之所以是“男人”的下义位,主要原因在于“婚姻状况”这个特征上,“男人”涵盖了“鳏夫”的取值范围。

当然,这种方法也存在明显的问题。

(1) 对于很多词难以进行义素分析,主要存在于特征的选取和说明上,例如“茶杯”和“茶缸”、“椅子”和“沙发”、“水果”和“苹果”等等,其差别从本体论上来说是十分明显的,但对于它们的区别性特征的选取和表示还不是十分明确。有时,为了说明这种差别,需要引入更多的特征,这样导致特征数目的迅速增加,而这并不是所希望的。

(2) 有些概念似乎缺少共同的属性可以进行抽象,但它们通常又被认为是处在某个共同的语义场中的,例如“篮球、足球、羽毛球、毽球……”,对这种情况的分析,后面会介绍另外一种理论来对它们进行解释,即“家族相似性”。

(3) 许多义位之间的边界是模糊的,例如“赤、橙、黄、绿、青、蓝、紫”,颜色之间的区别是渐变的,没有一个“是/否”的边界。

依据义素分析得到原语系统的方式奠基于这样一个信念:义位是由多个不同的义素共同组成的,而这一点在认知科学中有长期的心理学支持,比如在一些心理学的语义记忆模型中,包括层次网络模型、集理论模型和特征比较模型,都认为词或概念是由一些语义特征来表征的。而在概念结构的研究中,心理学家们也在 20 世纪 70 年代提出了类似的特征表说。特征表说认为,概念或概念的表征是由两个因素构成的:①概念的定义性特征,即一类个体具有的共同的有关属性(如果把一个义位看成是一个概念的话,一个定义性特征就是一个义素);②诸定义性特征之间的关系,即整合这些特征的规则。这可以用下述方程来表示:

$$C = R(X, Y, \cdots)$$

其中,C 为概念;X,Y,…为一类个体或行为具有的共同的定义性特征;R 为整合这些特征的规则,这些规则又称为概念规则,它们确定诸定义性特征的关系。例如,人工概念“红色圆形”的定义性特征为“红色”、“圆形”,两者之间的关系为“和”。

杉克(R .Schank) 在其概念相依理论中提出了一些关于行为的原语,这些行为原语可以通过基于义素分析的方法得来。这 11 个原语分为五组:

第一组为 MOVE, PROPEL, INGEST, EXPEL 和 GRASP,这五个原语均反映人类执行的物理行为,其中 MOVE 表示的是人身体某部分的行为;PROPEL 表示的是由人执行而导致某个对象发生位置变化的行为;INGEST 表示的行为是把某些东西放入某人的体内,主要用于描述吃和喝; EXPEL 与 INGEST 相反,是表示生物把东西从体内放到体外;GRASP 用来表示人握取物品的行为。

第二组 ATRANS 和 PTRANS 为两个表示状态改变的原语,二者之间的差别在于,PTRANS 本质上表示物理对象的位置变化,可以有对象、方向和工具等概念格;而 ATRANS 表示对象和生物之间抽象关系(如所属关系)的改变,无需有物理位置上的变化。

第三组的两个原语 SPEAK 和 ATTEND 为通信类,是表示与外部世界交流信息的行为原语。其中 SPEAK 表示向外界发送信息的行为;而 ATTEND 与 SPEAK 相反,它表示从外部获取信息。

第四组 MTRANS 和 MBUILD 为抽象精神行为。MTRANS 表示精神状态的变化,MBUILD 与思考有关。

此外,还设置了一个 DO 行为,作为在其他未预见到的情况下使用。当行为不能归结为上述一些行为原语时,可使用此原语。

进一步,杉克提出了概念句法规则。概念句法规则有两个方面的作用,一方面可以用来描述概念结构及其要素。

杉克认为,为了构成行为概念的完整意义结构,除了需要行为本身,还必须有另一个要素的参加,他称之为角色。有几种不同的角色:行为者(ACTOR)、对象(OBJECT)、方向(DIRECTION)、工具(INSTRUMENT)等等;可以扮演这些角色的又都是一些其他类型的概念,例如:PP(物理对象概念,例如生物、非生物)、LOC (地点概念,例如行为发生所在的物理地点)、T (时间,可以是一点的时间,也可以是一段时间)、PA (PP 的属性,通常一个 PP 可以由多个 PA 来共同描述)。

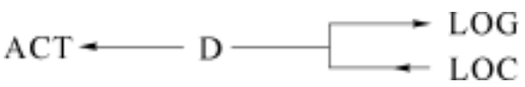
下面是几条概念句法规则,分别用来为一个行为(用 ACT 表示)指定不同的角色。

PP ACT

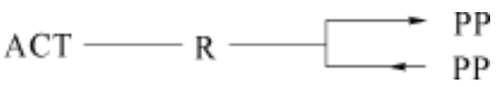
表示一些 PP 可以发出行为 MOVE。

ACT←OBJECT-PP

表示 MOVE 有对象 OBJECT,而 OBJECT 是一种 PP。



表示 ACT 有方向性。



表示 ACT 有承受者,承受者为 PP。

表示 ACT 有工具。

另一方面,利用概念句法规则,可以基于上述几组原语来形成更为复杂的义位。这时,概念句法规则相当于概念结构特征表说中的整合特征的规则。例如,可以用如图 4-4 所示的规则表示“吃”这个概念。

图 4-4　描述“吃”的概念句法

从图中可以看到,“吃”是由 INGEST 和 PTRANS 两个原语共同构成的。图中的 PP1 和 PP2 分别为两个物理对象,其中 PP1 为“吃”行为的发出者,而 PP2 为“吃”行为的对象。同时,“吃”的过程是通过把对象移到嘴里而实现的,因此,这一过程 (由 PTRANS 原语来描述的) 整个可以看成是 INGEST 的工具。

除了杉克的行为原语系统之外,还有许多原语系统都是基于义素分析的,有的以语义分类系统的形式出现。

但是,作为对义位进行义素分析的心理学基础,特征表说存在明显的弱点,它通常只能较好地解释人工概念和一些具有简单逻辑组合的概念,对于一些常见的自然概念,难以进行简单语义特征的组合。例如,“水果”等概念,就难以用几个特征来完全刻画。为此,在认知心理学中,还提出了一个关于概念结构的学说称为原型说,这在下一节中介绍。

4.4　原　　型

原型说的主要代表 Rosch(罗莎)认为,概念主要是以原型即它的最佳实例表征出来的。人们主要是从能最好地说明一个概念的实例来理解该概念的。例如,在思维活动中涉及鸟的概念时,通常会想到鸽子,而不大会想到企鹅或鸵鸟。这说明鸽子和企鹅不能在同等程度上表征鸟的概念,但是企鹅无疑也属于鸟类。因此,人们对一个概念的理解不仅包含着原型,而且也包含维量。Rosch 把这个维量称作范畴成员代表性的程度(degree of category membership),它表明同类个体的容许的变异性,也即其他个体偏离原型的容许距

离。例如,企鹅偏离鸽子的程度在容许距离内,因此也属于鸟类;而狮子则偏离太多,因而不属于鸟类。Rosch 认为概念就是由这两个因素共同构成的:①原型或最佳实例;②范畴成员代表性的程度。这两个因素紧密地结合在一起,而原型起着核心的作用。

这种学说有许多心理学真实性,例如在幼儿的语言练习的过程中,幼儿总是先倾向于用一种水果例如苹果来称呼所有的其他类似的水果;用球来称呼所有圆形的东西等等。

这种方法可以解决一些在基于特征表的义素分析中难以解决的问题。

(1) 有些义位用特征表的方法定义并不完整,例如,上述对狮子的定义,准确地说,只是对一个典型的狮子的描述,如在定义中说狮子身长约 3 米,如果严格按定义来判定,则很多幼狮就都不是狮子了。诸如此类,对于基于特征表说对自然概念的定义很容易找出一些这样的问题。其原因就在于,用特征表的方法进行定义实际上定义的是一个典型实例。

(2) 有些概念似乎缺少共同的属性可以进行抽象,但它们通常又被认为是处在某个共同的语义场中的,例如"篮球、足球、羽毛球、毽球……",这种情况,由维特根斯坦称为"家族相似性",即在一个家族中,爷爷与父亲的鼻子很像,父亲和儿子的嘴型很像,儿子又和妈妈的眼睛很像,等等,各个个体间只在某些特征上相似,而没有共同的相似点。

(3) 许多义位之间的模糊边界可以用原型的转移来解释,例如"赤、橙、黄、绿、青、蓝、紫",颜色之间的区别是由于原型色的渐变导致的。

原型的观点使我们可以从另外一个方面来得到词汇的层次结构,如图 4-5 所示。

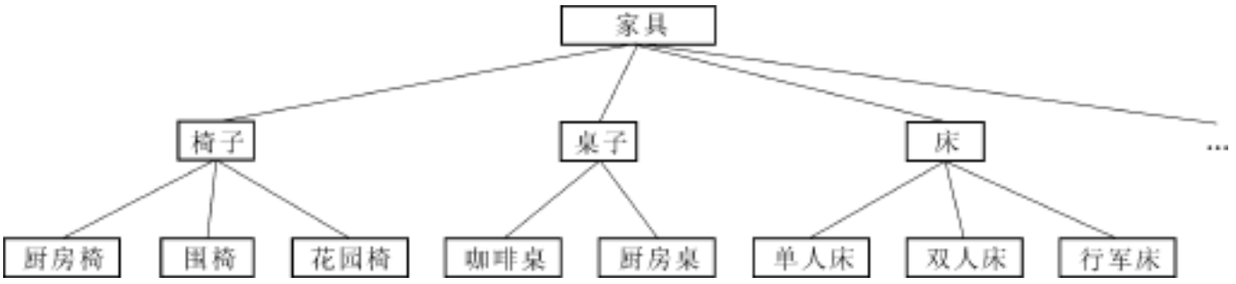


图 4-5　从原型观点得到的一个层次结构

一个典型的基于原型说的原语系统是 Wierzbicka 提出的原语系统,其原语均为英语单词(实例),而不是抽象的概念。这个原语系统包括如下的 16 个类别,每个类别分别有几个英语单词。

substantives: I, YOU, SOMEONE, SOMETHING, PEOPLE

determiners: THIS, THE SAME, OTHER, SOME*

quantifiers: ONE, TWO, MANY(MUCH), ALL, MORE*

mental predicates: THINK, KNOW, WANT, FEEL, SEE*, HEAR*

speech: SAY

actions and events: DO, HAPPEN

evaluators: GOOD, BAD

descriptors: BIG, SMALL

time: WHEN, BEFORE, AFTER, A LONG TIME*, A SHORT TIME*, NOW*

space: WHERE, UNDER, ABOVE, FAR*, NEAR*, SIDE*, INSIDE*, HERE*

partonomy and taxonomy: PART (OF), KIND (OF)

metapredicates: NOT, CAN, VERY

interclausal linkers: IF, BECAUSE, LIKE

movement, existence, life: MOVE, THERE IS, LIVE

imagination and possibility*: IF, WOULD, CAN, MAYBE

WORD*

这个原型系统中,带 * 号的词或类是 Wierzbicka 与他的同事第二批增加的,他声称,第一批原语已得到了广泛的测试。下面选取其中的几个考察其选取原语的依据。

I 和 YOU:目前为止,还没有发现任何一种语言中没有类似 I 和 YOU 的指代词汇的,学术文献中关于"没有人称代词"的语言到目前为止还只是个猜想。很多语种中,说话人都会用一些其他的词来代替 I 和 YOU,例如,在汉语中有称自己为"不才",称对方为"足下"等,但其概念上都是指"我"或"你"。

到目前为止,任何企图定义 I 和 YOU 的研究都没有成功。例如,曾有学者试图用 speaker 来定义 I,但发现 speaker 有比 I 更大的语义复杂度。虽然 speaker 通常称自己为 I,但 speaker 并一定是 I。例如句子:I don't like the speaker。显然,speaker 指的不是 I。进一步,I 不一定要和说话密切相关,比如在思想中的 I。

THINK 作为表示思维活动的一个典型实例,其他如推理、论证都可以建立在此原语之上。

利用原型方法同样也可以对语义关系进行解释,但是从目前的应用来看,其较少用于计算实现上。实际应用中的原语通常是基于特征分解的。

4.5 词义选择

在前面的几节里,介绍了一些语义关系,以及分析这些语义关系的两种途径。同时,在介绍上下义关系时也已经提到,这种关系可以用来帮助消除词汇意义选择时存在的不确定性。通常把这种能够帮助确定意义合法组合的语义关系称为选择约束条件。前面提到的一些基本的语义关系就是一些选择约束条件,下面介绍更为复杂一些的选择约束条件。

4.5.1 论旨角色

1968 年,菲尔摩(C J .Fillmore)发表了一篇颇有影响、题为《格辩》的论文,在文中他提出了格语法。

在格语法中,菲尔摩提出利用句子中动词周围的名词性成分与动词的语义组合关系来形成表达句子意义的格结构,这给句子意义的研究提供了一个新途径。格语法在计算机上的分析效率也比较高,受到了自然语言处理研究者的欢迎,尤其在自动机器翻译方面得到了广泛的应用。

菲尔摩的"格"与传统语法中的"格"十分不同。

在传统语法中,格指的是在某些语言中,同一名词的不同形式(如主格、宾格、所有格、

与格等等)。不同的动词在支配这些名词时,需要采用不同的格,例如,在拉丁语中,动词 vielere(看)的宾语必须是名词的宾格(如 hominem 人)、meminisse(记住)的宾语是所有格(如 hominis 人)等等;名词出现在不同的位置时,也应该使用不同的格,如在做主语时用主格。可以看到,这些不同的格通常是词的各种曲折变化形式,在拉丁语、德语、俄语等语种中都存在,虽然其他的一些语言(如英语)的名词没有这种曲折变化(英语中有这些复杂曲折变化的仅限于代词),但是从更为一般的观点来看,前置词或后置词(如英语中的 to, by 等为前置词,汉语中有一些后接成分如儿、性等)可以被认为同格在具有格曲折变化的语言中一样,具有同样的语义和句法功能。显然,在这种意义下,格因语种的不同而有较大的差异。

菲尔摩的"格"则具有其他的意义,它把格作为语言深层结构的范畴,指的是施事、地点以及动作的工具、目的等概念(目前,为了区分两种格,研究人员通常把这种格称为论旨角色,本文由于不涉及到传统用法中的格,因此,仍使用简单的名称——格)。这些概念可以对具有十分不同的语法结构的语言进行统一的描述。这种描述能够实现也在一定程度上说明语言的语义组合关系具有某种与具体语种无关的普遍性,同时,也与通常人工智能中的知识表示方案具有一致性。

菲尔摩最初列出的格系统包括六个格,它们是:

Agentive 格(施事格):一个句子成分被认为是施事格,如果该成分所代表的对象是句子主动词所表现的事件、行为或状态等的主动发起者。例如,如下的句子:

Tom broke the windows .

中,Tom 是动词 broke 的施事格,即 Tom 是 broke 这一行为的发起者。

Instrumental 格(工具格):一个句子成分被认为是工具格,如果该成分所代表的对象是句子主动词所表现的事件、行为中使用的工具。例如,如下的句子:

Tom broke the windows with a ball .

中,a ball 是动词 broke 的工具格,即 a ball 是施事 Tom 发起 broke 这一行为的工具。

Dative 格(与格):一个句子成分被认为是与格,如果该成分所代表的对象是句子主动词所表现的事件、行为的参与者。例如,如下的句子:

Tom give me a ball .

中,me 是动词 give 的与格。

其他三个为 Factitive 格、Locative 格和 Objective 格。

Fillmore 的格也被称为语义角色、深层格等,Fillmore 的格表现了句子动词和名词的语义组合关系。在后面,将把这种关系推广到更一般的语义网络的情况。

利用格关系可以定义一些选择约束条件,例如,对于词 read 取义位"READ1 = 阅读"时,可以定义两个约束条件:

(AGENT READ1 PERSON)——READ1 的 Agentive 格必须为"人"

(OBJEC READ1 READ-OBJ)——READ1 的 Objective 格必须为"可阅读的东西"

再看如下的例子:

The dishwasher read the article .

这里 dishwasher 有两个不同的义位:"洗碗机"和"洗碗工";而 article 有"文章"、"冠词"等不同的义位。利用上述对 read 的两个约束条件,首先可以确定 dishwasher 应该是"人";而 ar-

ticle 应该是“可阅读的东西”。单凭这个约束还不能确定义位的选择,进一步还要用前面介绍的一般的义位关系。这里需要考察上下义关系,显然 dishwasher 的两个义位中“洗碗工”是“人”的下义位,而“洗碗机”不是,因此,首先可以确定 dishwasher 应选择的义位;而 article 的义位“文章”是“可阅读的东西”的下义位,“冠词”则不是,因此可确定 article 的义位选择。这样就可以消除 dishwasher 和 article 的词义选择歧义。

在上述过程中,并没有说明为什么可以断定 dishwasher 就是 read 的 Agentive 格,而 article 就是 read 的 Objective 格,这是句子的格结构,通常是从句子中的句法结构映射得到的。这里介绍一个简单的映射方案,这种映射方案是以动词为中心的。首先需要为每个动词设计一个格框架,每一个格框架都由规定的几种格(例如上述的六种)组成,每一个格都需要标记为必须的、可选的或者禁止的。先通过一个例子来介绍这个方案,对于下面的句子:

他用刀划破了衣服。

它的句法结构为

(他((用刀(划破了))(衣服)))。

通过句法结构可以确认句子的主动词:划破;再确认主动词周边的名词短语有:他(主语),刀(介词宾语),衣服(宾语)。格分析就将在此基础上来确定这些周边的名词短语将分别成为主动词的哪个格。

主动词“划破”的格框架需要在词典中预先给出,设为

(可选的,可选的, 可选的, 可选的, 可选的, 可选的)

下面就可以进行格指派。

首先依据主动词的结构来判定句子是主动句式还是被动句式,然后依据如下的规则。

规则 1:如果是主动句,则检查主动词的格框架。如果它需要 Agentive 格,则句子的主语很可能就能充当 Agentive 格,而直接宾语可能是 Objective 格,间接宾语可能是 Dative 格。

规则 2:如果是被动句,则有所不同,主语可能充当 Objective 格和 Dative 格,这要依据进一步的信息来确定,而 Agentive 格可以依据介词的出现来判定,这在下一步说明。

其次,由于介词是重要的格表标志,需要单独考察。通常,在被动句中,由 by 引导的名词短语可能作为 Agentive 格。而一般,在具体语言中特定的介词可能标识一个特定的格。比如:

被、by——Agentive

用、with、by——Instrumental

把、to——Dative

在、于、on、in、under——Locative

因为介词“用”可以作为工具格的标识,则介词短语“用刀”可以辨识为工具格。然后反向利用上述规则 1 和规则 2 来辨识句子的主语和宾语是什么格。在主动句中,如果主动词需要 Agentive 格,则主语很可能就能充当 Agentive 格,直接宾语可能是 Objective 格,间接宾语可能是 Dative 格,这样,在上句中就可以断定“他”充当 Agentive 格,而“衣服”作为 Objective 格。在被动句中有所不同,但可以有类似的规则。

实际上,上述规则并不是充分的,因为经过上述的过程后,可能为一些名词短语确定的位置是错误的,还有一些名词短语的位置仍不能确定。

通常,为了进一步确认格结构,还需要更多的语义知识的帮助。比如,除了确定格框架中各个格位(格框架中确定位置的格)的格标(指明每个格是必须的、可选的或禁止的三种值之一)之外,还对可以充当某个格的名词短语进行相应的语义范畴约束。例如下句中:

球砸破了玻璃。

主动词“砸破”的格框架为:

(可选的,可选的,可选的,可选的,可选的,可选的)

有两个名词短语:“球”和“玻璃”。如果按照上面的规则,“球”是 Agentive 格,而“玻璃”是 Objective 格,但实际上,“球”并不是“砸破”动作的施动者,而是该动作的工具。为了能得出正确的格分析结果,就必须除了确定格框架中各个格位的格标之外,还对可以充当某个格的名词短语进行相应的语义范畴约束。例如:充当主动词“砸破”的 Agentive 格的主名词应该是表示有行为能力的生物,充当 Objective 格的主名词应该是非生物的有破这种状态的物品,充当 Instrumental 格的主名词应该是可用来砸的物品等等这样一些常识知识。具有这样的一些知识后,就可以进行比较正确的格结构确认。如上面的句子中,“球”虽然处在主语的位置,但是它不是具有主动动作能力的,因此不能作为 Agentive 格;而由于其具有可用来砸的特性,所以可以确定为 Instrumental 格;“玻璃”作为 Objective 格在语义上也是合适的。

从上面的分析可以看到,利用格语法进行句子分析,需要对每个动词建立适当的格框架,同时对于每一个格位,都应该定义可以作为此格位的语义分类范畴,这个范畴越细致,那么在应用中就会越有效。这也说明句子格结构的确定与词义选择是相互补充的。

许多研究人员对菲尔摩的格系统进行了扩展,例如中国人民大学林杏光等在日本 CICC 组织的亚洲五国多语种机器翻译系统的研究中,提出了一种分层的格系统。如图 4-6 所示。

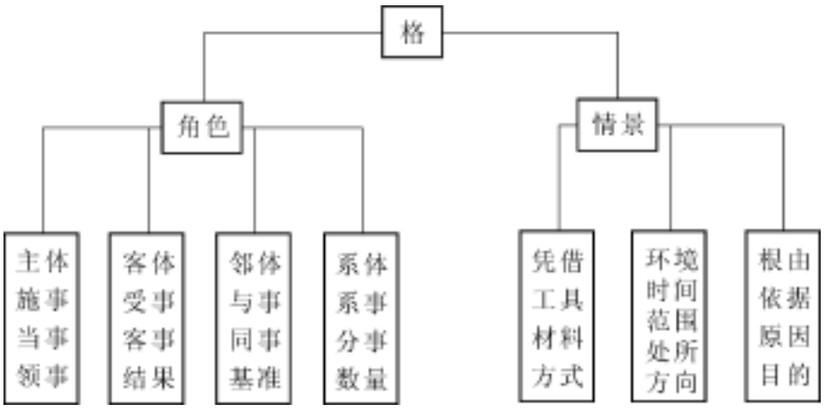


图 4-6　一个分层的格系统

4.5.2　语义网络

义位关系、格关系都可以统一在语义网络下得到描述,并进行推理。例如,表示“鸽子”与“动物”之间的上下义关系,如图 4-7 所示。

图中两个结点框分别是两个概念义,连接两个概念的有向弧上标识的是两个概念之间的一种关系。在图中,表示鸽子与动物的关系是种属关系。还可以表示整体-部分关

系,如图 4-8 所示。

这种关系反映了概念之间的整体部分关系。

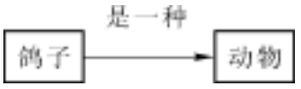


图 4-7 “是一种”关系

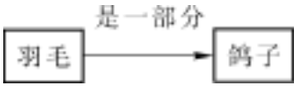


图 4-8 “是一部分”关系

属性关系,如图 4-9 所示。

这种关系反映了概念之间的属性关系。

表示格关系,比如上述例子中 Instrumental 格、Objective 格表示两种关系,而“球”、“砸破”、“玻璃”分别表示三个概念,则句子可以用图 4-10 所示的语义网络来表示。

图 4-9 属性关系

图 4-10 格关系的语义网络表示

从这个语义网络中,可以得到一个句子的意义结构,即“球砸破玻璃”。而这个意义结构要进一步实现句子“球砸破了玻璃”,还需要加上对于整个句子起作用的 Modality,即该句子的时态是“完成”时。为了在语义网络中完整体现句子,可以把各个 Modality 定义为关系,而对于一种 Modality 的具体实现定义为概念。例如,时态是一种关系,而“现在”时态、“完成”时态、“进行”时态等均为概念。这样,我们把上述意义结构的语义网络进一步表示为如图 4-11 所示的结构。

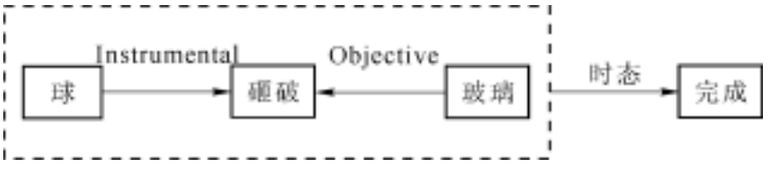


图 4-11 格语法的图表示

图中的虚线框表示内部是一个组合概念,它是由一些概念和关系组合成的一个语义网络来表达的,这个概念与“完成”之间是通过时态这一关系关联起来的,或称此概念在时态关系中取得“完成”值。图 4-11 表示了一个完整的句子,通常也将它称之为格语法的图表示。

小　　结

本章首先介绍了词汇语义的基本意义单元——义位,考察了在进行义位分析时需要关联的意义环境——语义场。随后介绍了基于语义特征的义素分析方法,分析了语义特征的认知心理学背景,从而引出了其相对的原型说,并介绍了其在原语分析中的应用。最后介绍了利用选择约束进行词义选择的途径。

第五章　句义分析

在解决词义(小结构)分析之后,如何分析句子(大结构)意义。大结构是无穷的,小结构(词)是有限的,一种解决办法是利用小结构的意义来指派大结构的意义,这种方式称为组合。它解决大结构的语义是如何依赖小结构的语义而得到的(复杂结构的语义是如何依赖简单结构的语义而得到的),其关键是意义的组合性,这种组合性包括两个方面:

(1) 句义(短语结构)依赖于组成它的词(小单位)的意义,例如:

Tom admires Sandy .

与句子

Tom hates Sandy .

的意义差别就是由于构成句子的词的不同。

(2) 也依赖于词(小单位)的结合方式,例如:

Tom admires Sandy .

与句子

Sandy admires Tom .

两个句子的构成词是一样的,句义的差别来源于它们结合方式的不同。

为此可以提出组合性原理。整个句子的意义是如下两个方面的函数:

每个部分的意义

部分与部分间的组合方式

组合性原理是赋予句子意义的一种可行方式,但也有例外。如:

惯用语:kick the bucket = die

隐喻、反语

首语重复法:Sam and Kim came in and [they sat down]

依赖上下文的语言使用:He did it there, like that, yesterday

句法结构与语义结构不匹配:

[[Every man] [loves [a woman]]]

[For All x man(x) [Some y woman(y) [loves(x, y)]]]

I enjoyed the meal (= eating)

I enjoyed the film (= watching)

I enjoyed the book (= reading)

为此,下面首先定义一种句子的表示方法,定义在这种表示下的意义;然后进行句子意义的分析。

5.1 逻辑表示

考察句子:

Do you know what gate you are going to ? (5-1-1)

这个句子在不同的上下文中可以有不同的理解,其一可以是:

你的登机口是几号 ?

另外也可以是问:

你知道你在哪个登机口登机吗 ?

回忆上一章列出的一个例句:

every boy loves a dog . (5-1-2)

这个句子也是有歧义的。但是句子 (5-1-1) 和 (5-1-2) 的歧义有着本质的区别。句子 (5-1-2) 中的歧义是由对于句子本身的分析得到的;而句子 (5-1-1) 的上述歧义是由于不同的上下文所导致的,其逐字解释的意义其实是明确的。本书的句义分析主要解决句子 (5-1-2) 中的这种歧义;而对句子 (5-1-1) 的歧义分析需要交给基于上下文的推理来进行歧义求解,本书不涉及。

为此,首先要定义一种形式化的语言,使得用这种语言定义的句义是上下文无关的。本章介绍的逻辑形式就是一种上下文无关的形式化句义表示,由句子分析得到逻辑形式的过程称为语义解释。而上下文相关的分析是建立在语义解释之后,把逻辑形式映射到一种知识表示语言,进而进行推理理解,这部分称为上下文解释,主要由人工智能的相关技术来完成。本章主要集中在语义解释的问题上。下面首先介绍一个表示句子语义的逻辑形式语言,并解释在这种表示下,其语义的实质是什么;而后,将介绍如何获得句子的逻辑形式。

考察如下的句子:

He is tall and thin . (5-1-3)

任何懂得英语的人都能从上述句子得到用下面的句子表述的结论:

He is tall . (5-1-4)

显然,句子 (5-1-4) 是句子 (5-1-3) 的必然结论,即如果句子 (5-1-3) 为真,那么句子 (5-1-4) 也为真。这时,我们关心的是两个句子之间的某种关系,而这种关系取决于句子的真值,这种真值就是本章关心的句义。

通常,如果没有特定的情景,谈论一个任意描述的真假值是没有意义的。例如,在上面的句子(5-1-3)中,如果不知道 He 是指代什么,那么是不可能断定其真值的。进一步,即使给定了一个情景,如果这个情景与描述没有关联,那么对于真值判定也是没有帮助的。例如,对于如下描述:

Tom love Mary . (5-1-5)

如果给出的情景是有关风扇用途的介绍,那么这个情景对于决定描述(5-1-5)的真值仍然是没有意义的,就像没有指定任何情景一样。

情景必须与描述存在某种联系,特定的描述只有在特定的情景里才能确定其真值。

通常用词表来建立起这种联系。因为要处理的自然语言描述在后面都将用逻辑表示语言来描述,因而,在后面将把描述直接称为逻辑表示,而情景将称为模型。我们的最终目的是定义一种逻辑表示语言,并展示逻辑表示语言如何在一个模型中得到赋值,而赋值过程就能告诉我们一个逻辑表示在此模型下是真还是假。

首先来介绍联系逻辑表示和模型的词表。下面是一个词表的例子:

{(LOVE, 2),
(CUSTOMER, 1),
(ROBBER, 1),
(MA, 0),
(WEI, 0),
(HONG, 0),
(PIAO, 0) }

这个词表包含了两类很重要的信息,一类是词表告知了语言描述可能涉及到哪些内容,从上面的词表可以看到,有可能要谈论的话题是关于四个对象的一种关系和两种属性。(LOVE, 2)表示 LOVE 是一种二元关系,即存在于两个实体对象间;而(CUSTOMER, 1)和(ROBBER, 1)都是表示一元关系,即实体的属性;(MA, 0)、(WEI, 0)、(HONG, 0)和(PIAO, 0)表示四个常量。另一类重要信息是词表也告诉了我们在谈论这些内容时应如何使用这些词,例如,使用 LOVE 时应有两个常量,而使用 ROBBER 时只需一个常量的参与。

总之,词表提供了定义模型所需要的全部信息。下面,就以上面给出的词表为例,进一步讨论如何基于词表来建立模型。

一个模型 M 包括两个方面的信息:其一是要研究的对象实体集合 D,这个集合就是模型的语义值域;其二是要有一个映射函数 F,为词表中的每一个符号指派合适的语义值,这个函数称为解释函数。即

$$M = (D, F)$$

显然,D,F 的定义不同,就会生成完全不同的模型。当然,D,F 的定义不是完全随意的,在定义 F 时,应该把词表中的常量映射成 D 中的语义值,而把词表中表示 $n(n>0)$元关系的词映射为 D 中的 n 元组或 n 元组的集合。例如,下面是一种可能的定义:

D = {d1, d2, d3, d4}
F(MA) = d1
F(HONG) = d2
F(WEI) = d3
F(PIAO) = d4
F(CUSTOMER) = {d1, d3}
F(ROBBER) = {d2, d4}
F(LOVE) = {(d4, d2), (d3, d1)}

在这个模型中,d1 称为 MA;d2 称为 HONG;d3 称为 WEI;d4 称为 PIAO;MA 和 WEI 是 CUSTOMER;而 HONG 和 PIAO 是 ROBBER;PIAO 对 HONG,WEI 对 MA 分别存在 LOVE 的关系。

同样的 D,如果解释函数不同,模型也就不同了,如:

D = {d1, d2, d3, d4}

F(MA) = d2

F(HONG) = d1

F(WEI) = d4

F(PIAO) = d3

F(CUSTOMER) = {d1, d2, d4}

F(ROBBER) = {d3}

F(LOVE) = {(d3, d4) }

在这个模型中,d1 称为 HONG;d2 称为 MA;d3 称为 PIAO;d4 称为 WEI。只有 PIAO 是 ROBBER,其余均是 CUSTOMER,而且只有 WEI 对 PIAO 存在 LOVE 的关系。

值得指出的是,上述的两个模型都比较特殊,即每一个 D 中的实体分别对应于词表中不同的一个常量,即都有唯一的一个不同的名称,这种模型称为精确模型(exact model)。并非所有的模型都会是精确模型。例如,在上面的 D 中增加两个实体,得到新的模型:

D = {d1, d2, d3, d4, d5, d6}

F(MA) = d2

F(HONG) = d1

F(WEI) = d4

F(PIAO) = d3

F(CUSTOMER) = {d1, d2, d4, d5}

F(ROBBER) = {d3, d6}

F(LOVE) = {(d3, d4) , (d6, d5)}

其中,有两个实体没有被词表中的常量命名,未命名的 d5 是 CUSTOMER,未命名的 d6 是 ROBBER,而未命名的 ROBBER 对未命名的 CUSTOMER 存在 LOVE 的关系。这样的模型也是合法的。与此类似的另一端是,一个语义实体也可以有一个以上的常量名称与之对应。

以上是基于一个给定的词表建立模型,下面介绍基于词表建立逻辑表示语言。基于词表的逻辑表示语言是建立在以下几个因素的基础上的。

(1) 词表中的所有符号都是语言的非逻辑符号;

(2) 可数无限个元素的集合:{ x, y, z, …};

(3) 联结词非、实质蕴涵、析取、合取;

(4) 两个量词:全称量词和存在量词;

(5) 圆括号( )。

第(2)~(5)项是规定逻辑形式语言的,而不同逻辑形式之间的差别主要体现在非逻辑符号(词表)的选择上。

假设已经选定了一些词表。把它与其他因素结合起来是由逻辑形式语言的句法来定义的。

首先定义逻辑项,包括逻辑常项和逻辑变项。逻辑语言与词表的一般对应关系是,逻辑常项对应专有名词短语,而变项对应于代词。

其次是谓词,谓词与词表中的关系相对应。常项与谓词的结合形成原子式:如果 R

是 n 元关系，$\tau_1, \tau_2, \cdots, \tau_n$ 为 n 个逻辑常项，则 $R(\tau_1, \tau_2, \cdots, \tau_n)$ 为一个原子式。例如仍以上面的词表为例：

$$LOVE(PIAO, HONG)$$

就是一个原子。在原子式的基础上，可以构造更为复杂的描述。下面归纳定义一般的逻辑形式，称为合式公式(well formed formulas)。

(1) 所有的原子式是合式公式；

(2) 如果 α 和 β 都是合式公式，那么 $\neg\alpha$，$\alpha \wedge \beta$，$\alpha \vee \beta$ 和 $\alpha \rightarrow \beta$ 都是合式公式；

(3) 如果 α 是合式公式，x 是变项，则 $\exists x\alpha$ 和 $\forall x\alpha$ 都是合式公式，其中 α 称为该合式公式的母式；

(4) 此外再没有合式公式。

在后面，通常把合式公式简称为公式。粗略地，公式中的“$\neg$”，“$\wedge$”，“$\vee$”和“$\rightarrow$”分别相当于自然语言中的“不是”、“并且”、“或”和“如果……那么”，而 $\exists x\alpha$ 和 $\forall x\alpha$ 分别相当于自然语言的“有一些”和“所有”。

下面定义自由变量和受限变量。考察如下公式：

$$\neg(CUSTOMER(x) \vee \forall x(ROBBER(x) \wedge \forall y PERSON(y)))$$

其中第一个出现的 x 是自由变量；第二和第三个出现的 x 是受限变量，它们受到第一个出现的量词 $\forall$ 的限制；第一和第二个出现的 y 是受限变量，它们受到第二个出现的量词 $\forall$ 的限制。

下面给出自由变量和受限变量完整的定义：

(1) 在原子公式中出现的任何变量都是自由变量；

(2) 在原子公式中出现的任何变量都不是受限变量；

(3) 如果在 α 或 β 中出现的任何变量都是自由变量，那么在 $\neg\alpha$，$\alpha \wedge \beta$，$\alpha \vee \beta$ 和 $\alpha \rightarrow \beta$ 中出现的这些变量也都是自由变量；

(4) 如果在 α 或 β 中出现的任何变量都是受限变量，那么在 $\neg\alpha$，$\alpha \wedge \beta$，$\alpha \vee \beta$ 和 $\alpha \rightarrow \beta$ 中出现的这些变量也都是受限变量，并且对于任何的 y，这些变量在 $\exists y\alpha$ 和 $\forall y\alpha$ 中也都是受限的；

(5) 在公式 $\exists y\alpha$ 和 $\forall y\alpha$ 中，紧接着第一个量词出现的 y 是受限变量；

(6) 如果变量 x 在公式 α 中是自由的，那么对于任何与 x 不同的变量 y，x 仍然是 $\exists y\alpha$ 和 $\forall y\alpha$ 中的自由变量；而在 $\exists x\alpha$ 和 $\forall x\alpha$ 中，x 是受限变量。

句子：不存在任何自由变量的合式公式称为句子。

在不至于误解的情况下，在后面的讨论中将采用尽量少的括号，例如，公式

$$(CUSTOMER(VINCENT) \vee ROBBER(PUMPKIN))$$

简写为

$$CUSTOMER(VINCENT) \vee ROBBER(PUMPKIN)$$

在上述的逻辑形式中，谓词通常是词汇本身，这不利于一般化，为此可以对谓词进行抽象化。上一章所述的论旨角色就是一种方案，例如：$Agentive(x, y)$ 表示 x 是 y 的 Agentive。

以上基于词表建立了逻辑公式作为句义的描述工具。这样，到目前为止，在词表的基础上建立了作为语言情景的模型以及作为描述工具的逻辑公式。这样就完成了建立自然

语言语义的两个预备工作。

下面,定义要用的语义。

5.2 模型论语义

为定义语义,首先定义如下真值。

对于给定的词表,一个句子和模型之间是否存在一种叫真值的关系,如果存在,则句子为真;如果不存在,则句子为假。检查一个句子在给定的词表模型下是否为真通常是容易的,但如何为任意句子的这种关系给出一个精确的定义需要进一步的讨论。

再定义真值指派。

给定模型 M = (D, F),对 M 中变量的真值指派是一个从逻辑变量集到 D 的函数 g。

真值指派类似于给自然语言以上下文,作为情景的模型提供这种上下文。这样以模型作为中介,可以为每个句子建立真值指派,不同的指派导致不同的真值。

通常,是用二值模型,在 D 中包含两个实体:0 和 1,或者称为假和真,统称为真值。对于不同的真值指派,一个句子可能得到不同的真值。两个不同的句子,如果其中有相同的变项,那么,就可以利用句子间这种可能的真值之间的关系(通过变量的真值指派而获得的)来解释句子之间的推理关系。

下面首先是建立这种理论的一个准则。

真值条件准则:设 S1 是 S2 的必然结论,那么若一种理论把 S2 指派为真,则它也必把 S1 指派为真。

模型论语义就试图为自然语言的句子建立一个满足上述准则的真值指派机制。

考察下面的一个例子,句子:

$$\text{Tina is tall and thin .} \tag{5-2-1}$$

$$\text{Tina is thin .} \tag{5-2-2}$$

显然,直观上句子(5-2-2)是(5-2-1)的必然结论。下面,从模型论语义的角度来阐明这个结论。设模型为 M = (D, F), D 非空, d1 为 D 中的元素, F(Tina) = d1,即 Tina 记 D 中的元素 d1,词 tall 和 thin 分别记 D 的两个非空子集 S1 和 S2, F(tall) = S1, F(thin) = S2,这些指定可以是任意的。而 is 和 and 是代表明确定义的两个关系, and 代表一个二元关系,是一个交集函数, F(and) = 任意两个 D 的子集的交集;而 is 代表一个一元关系,是一个成员函数,它的值是这样定义的:设 x 是 D 中的一个元素, A 是 D 的子集,如果 x 是 A 中的元素,则函数值为真,否则值为假。有了上面的假设,就可以进行下面的分析了。

由上面的假设可知,句子(5-2-1)为真当且仅当

$$d1 \in S1 \cap S2 \tag{5-2-3}$$

句子(5-2-2)为真当且仅当

$$d1 \in S2 \tag{5-2-4}$$

显然,式(5-2-3)成立则式(5-2-4)必然成立,因此可知:句子(5-2-2)是句子(5-2-1)的必然结论。

以上建立了句子的意义,但如何能得到句子的语义结构是下面的任务。通常,获得句

子的语义结构有三种途径。一种途径是把语义作为一个句法特征,这样,在进行句法分析的同时,就得到句子的句义。在这种方式中,进行语义分析的依据是内含在句法中的,一个句法规则就包含了一个语义规则,语义分析是由句法驱动的。另一类方法是在句子的句法分析结果之上进行句义分析。最后一类是直接由句子来分析句义。在这三种方式中,句法信息逐渐减少。本章分别介绍这三类语义分析技术。

5.3 句法驱动的语义分析

5.3.1 语义组合性

上下文无关语法的一个典型特点是其组合性,即大的语法单元是由较小的语法单元构成的,句法结构的树形表示是对句法组合性的最好诠释:根结点(句法分析中的最大语法单元:句子)由下一级子结点(短语结构)构成,一级子结点又是由更下一级、更小的短语结构组成,如此,一直分解到最后是最小的语法分析单元(词汇)。句法分析的过程就是不断搜索句子中句法组合性的过程,而句义分析要能和语法分析同时进行。一个重要的要求就是句义的组合性,即句子中某个成分的意义可以唯一地由其组成单元的意义来导出。

看如下的例子,句子:

Tom hit Sue . (5-3-1)

的逻辑形式可以表示为:

(HIT1 E1 (NAME T1“Tom”) (NAME S1“Sue”)) (5-3-2)

同时该句子的句法结构是:

(S(NP(Tom) VP(hit Sue))) (5-3-3)

为满足组合性要求,应该能分别写出 Tom 和 hit Sue 两个部分的逻辑表示,Tom 部分不去考虑,对于 VP 部分,一个自然的想法是表示为

(HIT1 E1 (NAME T1“Sue”)) (5-3-4)

这个表示的一个明显的问题是,它在实现与其他句子成分的意义组合时,没有依据(相应的在句法中,它必须前面为 NP,共同组成一个 S)。这对于句义的组合性分析是不足的。为此,需要对逻辑表示进行扩展,一个克服该问题的重要技术是所谓的 λ-演算。

例如,在上面的例子中,可以利用 λ-演算来为 VP 部分提供一个表示形式(称为 λ-表达式):

(λx(HIT1 E1 x (NAME S1“Sue”))) (5-3-5)

式(5-3-5) 和式(5-3-4)的差别在引入了额外的一个变元 x,通过指明变元,明确了 VP 还需要和另外一个成分结合才能形成完整的句义。这样就完整地体现了 VP 的组合性要求,为下一步实现意义的组合奠定了形式基础。这样,句子(5-3-1)的逻辑形式在 λ-演算下可扩展为

((λx(HIT1 E1 x (NAME S1“Sue”))) (NAME T1“Tom”)) (5-3-6)

命题为真当且仅当(NAME T1“Tom”) 满足谓词 (λx (HIT1 E1 x (NAME T1“Sue”))),也即(5-3-2)为真。

命题(5-3-2)是由 λ-表达式(5-3-5)应用到变元 (NAME T1“Tom”)而获得的。这一过程也称为 λ-还原。

λ-表达式在处理语义组合时是十分有效的。它可以用来处理十分复杂的语言现象,下面举两个例子。

例 5-1　两个具有不同句法结构的动词短语的联结使用,如下面的句子:

Sue laughs and opens the door . (5-3-7)

可以先分别为 laughs 和 opens the door 建立 λ-表达式,就可以很容易地建立句义表示,对于 laughs,有

(λ*x* (LAUGHS1 E1 *x*))

对于 opens the door,有

(λ*y* (OPENS1 E2 *y* (OBJ o1“the door”)))

当两个 VP 有相同的动作主体,组合在一起时,有 *x* = *y*,组合后的 λ-表达式为

(λ*x* (& (LAUGHS1 E1 *x*) (OPENS1 E2 *x* (OBJ o1“the door”))))

这恰好是句子(5-3-7)的 VP 部分的语义表示。和 NP 部分结合,并进行 λ-还原后就完整地表示了句子(5-3-7)的意义:

(& (LAUGHS1 E1 (NAME s1“Sue”)
(OPENS1 E2 (NAME s1“Sue”) (OBJ o1“the door”))))

例 5-2　介词短语,在名词短语 The man in the store 中的介词短语 in the store 可以不用给其单独的意义,但是,这种介词短语也会出现在句子

The man is in the store .

中。因此,能够表示介词短语的意义并用在不同的句法结构中,对于组合语义是十分有价值的,同时,也有利于反映句子中的内在语义联系。如果把短语 in the store 用 λ-表达式表示为

(λ*x* (IN-LOC1 *x* (LOC o1“the store”)))

则名词短语 The man in the store 的逻辑形式为

((the man m1) (IN-LOC m1 (LOC o1“the store”)))

句子 The man is in the store 的逻辑形式为

(IN-LOC1 (the man) (IN-LOC m1 (LOC o1“the store”)))

以上两个例子反映了应用 λ-演算进行语义组合性处理的一些优势,同时可以看到,利用 λ-演算可以建立对短语结构的意义表示,这对于下面利用短语结构语法的句法分析来驱动语义分析具有关键的作用。关于 λ-演算的进一步应用可以参见相关文献。

但需要指出的是,语义的组合性有一些需要特别处理的难题。

(1) 句法成分和语义成分通常是不一致的,也即是说:句法结构的成分分解结果与语义结构分解的结果不是一一对应的。例如,句子:

Tom loves every dog .

句法分析把该句子分解成三级短语组合结构:

((Tom) (loves (every dog))) .

但是,用逻辑形式来表述其语义,则为

(EVERY *d*: (DOG1 *d*) (LOVES1 E1 (NAME j1“Tom”) *d*))

可以看到,在句法结构和逻辑形式之间并没有一一对应的关系,在句法结构中,(every dog)是作为动词短语(loves (every dog))的子成分,而在逻辑形式中就看不到这一点,反而是(EVERY *d*: (DOG1 *d*)…)似乎要把动词(LOVES1 E1…)作为其中的一部分。更有甚者,every 和 dog 在句法结构中是紧密结合的短语,而在逻辑形式中,它们是完全分开的,一个在谓词外面作为量词,而另一个作为谓词的参量。这就使人很难把 every dog 作为一个意义单元来对待。一个处理方法是引入无范围(unscoped)逻辑形式,此时有:

(LOVES1 E1: (NAME j1"Tom") < EVERY *d* DOG1 >)

这时,逻辑形式与句法形式就比较相近了。

(2) 习语很难用组合性来处理,经典的例子:

Tom kicked the bucket .

意为

Tom died

这与句子的组成部分 kicked 以及 bucket 没有任何关系,句子的意义不可能由其组成成分来组合而成。一种解决方案是为 kick the bucket 指定一个意义 die,而不使用它的组合义。

5.3.2 句法驱动的语义分析

有了上述为短语结构建立的组合语义表示,语义分析就可以方便地在句法分析的驱动下进行。为此需要一些扩展,主要的扩展是在每一个词条和语法规则中增加一个特征,这个特征就是反映语义的,通常用 SEM 来表示这个特征。下面给出一个带有语义特征的小的词汇集以及具有语义特征的规则集,并利用它们来对句子进行语义分析。词条 Tom,在没有语义特征时,可以是:

$$\text{Tom} = \begin{bmatrix} \text{POS} & \text{N} \\ \text{AGR} & \text{3s} \end{bmatrix}$$

在增加语义特征后,可扩展为

$$\text{Tom} = \begin{bmatrix} \text{POS} & & \text{N} \\ \text{AGR} & & \text{3s} \\ \text{SEM} & \text{NAME} & \text{t1} \quad \text{"Tom"} \end{bmatrix}$$

其中,(NAME t1"Tom")为词条 Tom 的语义特征。当前主要考察语义特征,因此忽略其他特征。写成括号表达式的形式为

Tom(SEM (NAME t1"Tom"))

其他几个:

see (SEM SEE1)

the (SEM THE)

dog (SEM DOG1)

而语法规则

S→NP VP

在增加语义特征后的形式为

S(SEM(? semvp ? semnp))→(NP SEM ? semnp)(VP SEM ? semvp) (5-3-8)

其中的语义特征就是对句法规则在语义上的约束。再给出其他几个:

VP(SEM(λ *a*1(? semv ? v *a*1 ? semnp)))→(V SEM ? semv)(NP SEM ? semnp) (5-3-9)

NP(SEM(NAME ? semname))→(NAME SEM ? semname) (5-3-10)

NP(SEM< ? semart (? semcnp) >)→(ART SEM ? semart)(CNP SEM ? semcnp) (5-3-11)

CNP(SEM ? semn)→(N SEM ? semn) (5-3-12)

下面用上述的词和规则来分析句子,我们集中在语义特征上:

Tom saw the dog .

首先读入 Tom,由规则 (5-3-10) 得到一个 NP,其语义特征如下:

NP1(SEM(NAME t1“Tom”)

随后读入 saw,没有可应用的规则,继续 the;无可用规则,继续 dog。此时,由规则 (5-3-12)得到一个 CNP,其语义特征如下:

CNP(SEM DOG1)

该 CNP 与前面的 the 应用规则(5-3-11),得到另一个 NP:

NP2(SEM < THE DOG1 >)

NP2 与 saw 可应用规则(5-3-9),得到一个 VP,其语义特征如下:

(VP(SEM(λ *a*1(SEE1 *a*1 < THE DOG1 >))))

最终,VP 和 Tom 应用规则(5-3-8),得到句子的意义:

S(SEM(SEE1(NAME t1“Tom”) < THEDOG1 >))

由于这个句子的语义表示是与句法分析的过程同时获得的,因此具有与句法树相同的结构,如图 5-1 所示。

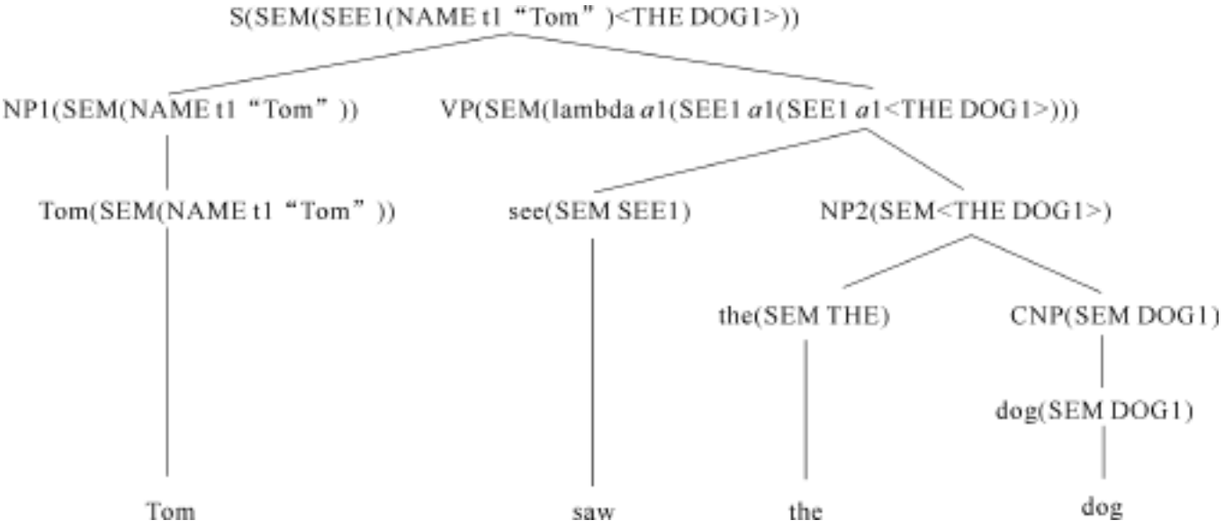


图 5-1 语义表示的树形式

图中,用单词 lambda 代替符号 λ。

可以看到,一个一般的基于特征的句法分析算法都可以经过简单的扩展 (主要是增加处理语义特征的部分) 就可以同时进行语义分析。当然,这种语义分析的能力与句法分析一样是建立在对词条中语义特征的描述以及语法中语义特征的规定之上的,同时语义的组合性还有许多问题需要去克服,在上面已经提到过两类语义的组合性的难题。

5.4 基于句法结构的语义分析

更为通常的语义分析方法不是像上一节那样与句法分析同时进行,而是建立在句法分析的结果之上,以句法分析的结果作为语义分析的输入。在这一类方法中,依据对句法分析结构的处理不同,又有各种变化。下面仅介绍一种基于语法关系的语义分析方法。

通常,利用(增强的)上下文无关语法,可以得到句子的特征结构,例如,对于句子

Jack bought a ticket . (5-4-1)

可以得到其特征结构:

(PRED BUYS1 SUBJ(NAME j2"Jack") OBJ < A t1 TICKET1 >) (5-4-2)

其中,PRED 表示句子的谓词;SUBJ 表示行为的主体;OBJ 表示行为的对象。

在基于语法关系的语义分析中,为了进行语义分析,并不需要句法分析的所有结果,而主要需要其中包含的某种语法关系,这种语法关系可以用三元组来表示。例如,如下的三元组:

< S PRED BUYS1 >

表示 BUYS1 是 S(句子)的谓词。

从特征结构到三元组的转换是容易的,例如上述的特征结构(5-4-2)可以直接转换成如下的三元组:

(S PRED BUYS1) (S SUBJ(NAME j2"Jack")) (S OBJ < A t1 TICKET1 >) (5-4-3)

其中包含三个三元组,说明了在句子中存在的几个关系,即

(S PRED BUYS1)表示 BUYS1 是 S 的谓词,(S SUBJ (NAME j2"Jack"))表示 Jack 是 S 的主语,(S OBJ < A t1 TICKET1 >)表示 a ticket 是 S 的宾语。表 5-1 列出了几个由句子到语法关系的对照表。

表 5-1 几个由句子到语法关系的对照表

句 子	语 法 关 系
Jill gave Jack a book .	(S PRED GIVES1) (TNS S PAST) (S SUBJ(NAME j1"Jill")) (S OBJ < A b1 BOOK1 >) (S OBJ (NAME j2"Jack"))
Jill gave a book to Jack .	(S PRED GIVES1) (TNS S PAST) (S SUBJ(NAME j1"Jill")) (GIVES1 OBJ < A b1 BOOK1 >) (GIVES1 TO (NAME j2"Jack"))
Jill thinks that Jack stole the book .	(S PRED THINKS1) (S SUBJ(NAME j1"Jill")) (S OBJ S1) (S1 PRED STEALS1) (S1 TNS PAST) (S1 SUBJ (NAME j2"Jack")) (S1 OBJ < THE b1 BOOK1 >)

注:表中 TNS 表示句子的时态。

在获得了语法关系的三元组表示后,语义分析的过程就是与语法分析完全独立了,语

义分析的任务就是把语法关系映射到逻辑形式。由于语法关系已经包含一些语义信息，因此到逻辑形式的映射通常是直接的，如表 5-2 所示。

表 5-2　语法关系到逻辑形式的映射

语　法　关　系	逻　辑　形　式
(< VERB > AT < LOC >)	(AT-LOC < VERB > < LOC >)
(< VAR > PRED < PRED >)	(< PRED > < VAR >)
(< ACTION-VERB > SUBJ < ANIMATE >)	(AGENT < ACTION-VERB > < ANIMATE >)
(< ACTION-VERB > OBJ < PHYSOBJ >)	(THEME < ACTION-VERB > < PHYSOBJ >)
(< GIVE-VERB > OBJ < ANIMATE >)	(TO-POSS < GIVE-VERB > < ANIMATE >)

上面的语法关系列是要匹配的部分，例如第一行中，先匹配 < VERB > ，任何的具有类型 VERB 的词都可以匹配上；然后是 AT，只有具有类型 AT 的词能匹配上；最后是匹配第三个类型 < LOC > 。所有的都匹配成功后就可以产生一个 SEM 结构，这个 SEM 结构表现了一个 AT-LOC 关系，有两个参数 < VERB > 和 < LOC > 。如果在一个句子中的每一个语法关系都得到了成功的匹配，就可以组合起来最终得到句子的语义解释。

例如，句子：

Jack bought a ticket .

该句子中包含如下三个语法关系：

(1) (S PRED BUYS1)

(2) (S SUBJ (NAME j1“ Jack ”))

(3) (S OBJ < A t1 TICKET1 >)

第一个三元组可以匹配表 5-2 中第二行的映射，得到逻辑形式：

(BUYS1 S)

第二个三元组可以匹配表 5-2 中第三行的映射，得到逻辑形式：

(AGENT S (NAME j1“ Jack ”))

第三个三元组可以匹配表 5-2 中第四行的映射，得到逻辑形式：

(THEME S < A t1 TICKET1 >)

这三个逻辑形式组合起来就可以得到如下句子的逻辑形式：

(BUYS1 S [AGENT S (NAME j1“ Jack ”)] [THEME < A t1 TICKET1 >])

类似地，对于复杂的句子：

Jill thinks that Jack stole the book .

通过句法分析可以得到如表 5-3 所示的几个语法关系以及相应的语义形式。

表 5-3　几个语法关系及其相应的语义形式

语　法　关　系	逻　辑　形　式
(S PRED THINKS1)	(THINKS1 S)
(S SUBJ (NAME j1“ Jill ”)	(EXPERIENCE S(NAME j1“ Jill ”))
(S OBJ S1)	(THEME S S1)
(S1 PRED STEALS1)	(STEALS1 S1)
(S1 TNS PAST)	(PAST S1)
(S1 SUBJ (NAME j2“ JACK ”))	(AGENT S1(NAME j2“ Jack ”))
(S1 OBJ < THE b1 BOOK1 >)	(THEME S1 < THE b1 BOOK1 >)

合并这些逻辑表示可以得到整个句子的逻辑形式:

```
(THINKS1 S[EXPERIENCE (NAME j1“Jill”)]
         [THEME (STEALS1 S1
         [AGENT (NAME j2“Jack”)]
         [THEME S1 <THE b1 BOOK1>])
         ])
```

可以看到,在基于语法关系的语义分析方法中,语法关系成为了句法处理和语义解释间的一个十分方便的接口,它使我们可以把这两个过程分别独立进行,语义映射不再像在前面的方法中那样依附于语法规则,而可以独立构造复杂的语义解释规则。

5.5 基于语义语法的语义分析

在为特定的应用在特定领域中构造自然语言应用系统时,通常可以利用一些很强的约束技术来提高句法、语义分析的性能。例如,虽然一般的自然语言句子结构需要非常复杂的语法系统才能够得到比较完备的覆盖,但是在特定应用中,人们可能只会用到自然语言中非常小的一部分结构,而且句子结构中的每个成分可能都有十分明确的语义约束。例如,在飞机航班数据库的查询中,通常会有如下的名词短语结构:

the flight to Chicago
the 8 o'clock flight
the first flight out
flight 457 to Chicago

为处理这些名词短语,一般需要下面一些规则(括号中为相应的例子):

NP→DET CNP	(the flight)
CNP→N	(flight)
CNP→CNP PP	(flight to Chicago)
CNP→N PART	(flight out)
CNP→PRE-MOD CNP	(8 o'clock flight)
NP→N NUMB	(flight 457)

但是在这个特定的领域和应用情况下,如下的例子:

the city to Chicago
the 8 o'clock city
city 567
the first city out

虽然是合乎上述语法规则的,但并不可能出现。一般情况下,可通过选择性约束和特征来限制可能的句法结构。但是在特定领域,问题可以简化处理,可以通过语义属性来指定词汇类别,由于在特定领域,这种类别数目是可能穷尽的。例如,在上述飞机航班数据库的查询问题中,主要存在的几类名词包括表示飞机、城市、票等等,如果分别记为:FLIGHT-N,CITY-N,TICKET-N,则前面的一般性语法规则可以重新表述为

FLIGHT-NP→DET FLIGHT-CNP (the flight)

FLIGHT-CNP→FLIGHT-N	(flight)
FLIGHT-CNP→FLIGHT-CNP FLIGHT-DEST	(flight to Chicago)
FLIGHT-DEST→prep CITY-NP	(to Chicago)
FLIGHT-CNP→FLIGHT-CNP FLIGHT-SOUR	(flight from Chicago)
FLIGHT-SOUR→prep CITY-NP	(from Chicago)
FLIGHT-CNP→FLIGHT-N PART	(flight out)
FLIGHT-CNP→PRE-MOD FLIGHT-CNP	(8 o'clock flight)
FLIGHT-NP→FLIGHT-N NUMB	(flight 457)
CITY-NP→CITY-NAME	(Boston)
CITY-NP→DET CITY-CNP	(the city)
CITY-CNP→CITY-N	(city)

TIME-QUERY→When does FLIGHT-NP FLIGHT-VP

(when does the flight to Chicago leave ?)

这种以领域内语义范畴为结构单元的语法称为语义语法。

依据语义语法可以进行句子的语义分析,其分析是和通常的句法分析相同的,但是需要每个词的语义类别信息。分析结果也可以用树形结构来表示,例如,对于句子

when does the flight to Chicago leave ?

有语义树结构如图 5-2 所示。

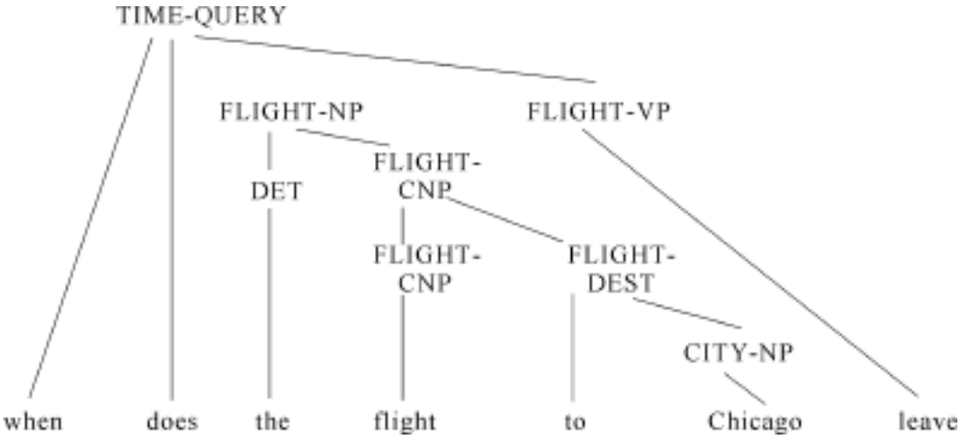


图 5-2 语义树结构

利用图中所示的语义树结构,就可以很容易地分解出句子的语义成分,从而辨别出所要查询的是什么。

利用语义语法可以快速地开发在特定领域针对特定任务的应用系统,但是其构造的语义语法基本上不能推广到其他的领域,新的领域需要构造新的语义语法。而通常的句法分析就不会受到如此强烈的领域约束。

5.6 语义驱动的句法分析

前面介绍的几种语义分析都是要建立在句法分析的基础上,在一个极端是语义作为一个特征发生在句法分析的同时,无需单独存在,有的在句法分析完成之后,而基于语义

语法的分析是用语义信息改造语法规则。本节介绍另一种极端,即完全抛开语法(除了基本的词的形态分析),直接对句子进行语义分析。当这种语义分析结束时,也同时就获得了句子的结构,因此称之为语义驱动的句法分析。

在这类系统中,语义信息主要储存在词典中,其中包括静态词的不同的可能意义,动词、形容词的格框架信息,还包括一些过程性的知识,如:消除词汇歧义的操作、语义结构组合的操作等等。本节介绍一个这样的系统:词典中的语义信息以模式-行为规则的形式存储,在读入一个词时,就进行模式匹配,一旦匹配上一条规则,就按规定的行为做相应的操作。例如,在词典中 book 一词可能包含如下信息:

```
BOOK .1
    < ANIMATE > " book "→(RESERVING * [AGENT 1])
BOOK .2
    < RESERVING > < TRANSPORT > →1 ∧ (RESERVING * [THEME 2])
BOOK .3
    < RESERVING > < ANIMATE > →1 ∧ (RESERVING * [BENEFICIARY 2])
BOOK .4
    < RESERVING > " for " < ANIMATE > →1 ∧ (RESERVING * [BENEFICIARY 2])
S .end
    < ANYTHING > " ."→ POP
```

可以看到,这是几条产生式规则,当左边匹配成功后,就用右边的项替换掉左边。规则中的 * 表示一个变量,数字 1,2 等代表左边的第 1,2 项。这个词典中 book 有 5 条规则,前面 4 条规定了 4 个模式的处理,最后 1 个表示测试句子是否结束,即遇到一个" ."符号后,一个句子就结束了。

下面以分析句子

John booked me a flight to Chicago .

来阐述如何使用词典中的那些规则。

设已经分析得到了第一项:

(NAME j1 HUMAN" John ") (5-6-1)

下一步读入单词 booked 为第二项,经过形态分析可得 book,查找词典,首先可以匹配上第一条规则 BOOK .1,第一项可以匹配上 < ANIMATE >,因此执行第一个规则规定的替换操作,用下面的项来替换前面的第一和第二项,得到当前项为

(RESERVING r1 [AGENT (NAME j1 HUMAN" John ")]) (5-6-2)

出现了保留项(RESERVING),因此还要继续使用,其中的 r1 表示规则中的变量 *。读入下一个词作为第二项,设单词 me 有结构

(PRO m1 HUMAN" me ") (5-6-3)

此时可以成功匹配规则 BOOK .3,替换后可得到新的当前项:

```
(RESERVING r1 [AGENT (NAME j1 HUMAN" John ")]
                [BENEFICIARY (PRO m1 HUMAN" me ")])
```

(5-6-4)

继续下面的词,读入 a 作为第二项,此时在 book 中没有规则可以匹配,因此要进行匹配位置的位移,把第二项作为当前匹配位置来分析后面的部分。首先给出后面几个词的

规则,对于 a 的规则有:

ART .1

“a”→INDEF1

对于 flight 有:

FLIGHT .1

< DET > “flight”→ < 1 * (FLIGHT *) >

FLIGHT .2

< FLIGHT > “to” < LOC > → < ? a ? f (FLIGHT ? f[TO-LOC 3]) >

对于 Chicago 有:

CHICAGO .1

“Chicago”→(NAME c1“Chicago”)

此外,还有关于名词短语结束的两个规则:

NP .end1

“ .”→POP

NP .end2

< verb > →POP

这样,由于第一项(5-6-4)不能进一步与 a 匹配,保存(5-6-4)到 Buffer1,把当前匹配位置后移到 a,开辟另外一个 Buffer2,保存:

“a”→INDEF1

继续取下一项 flight,可以匹配 FLIGHT .1,Buffer2 改变为

< INDEF1 f1 (FLIGHT *) >

其中的 f1 表示规则中的变量 *,以区分前面 Buffer1 中的变量名。继续取下一项 to,没有匹配的规则,开辟新的 Buffer3 保存 to,取下一个词 Chicago,可以匹配规则CHICAGO .1,保存到 Buffer4 为

(NAME c1“Chicago”)

Buffer2,Buffer3 和 Buffer4 可以共同匹配规则 FLIGHT .2,合并到 Buffer2 中,得到:

< INDEF1 f1 (FLIGHT f1 [TO-LOC (NAME c1“Chicago”)]) > (5-6-5)

继续读取下一个词,为“ .”,匹配规则 NP .end1,这标识已得到一个完整的 NP,同时也可以匹配规则 S .end,这标识了整个句子的结束。这时对句子的分析还分别放在两个 Buffer 中:Buffer1 中的(5-6-4)和 Buffer2 中的(5-6-5),二者可以继续匹配规则 BOOK 2 的右边,替换后得到一个表达式:

(RESERVING r1 [AGENT (NAME j1 HUMAN“John”)]
[BENEFICIARY (PRO m1 HUMAN“me”)]
[THEME < INDEF1 f1 FLIGHT
(TO-LOC f1 (NAME c1“Chicago”)) >]) (5-6-6)

这就是整个句子的分析结果。

可以看到,基于语义驱动的句法分析是依赖词典中对每个词构造的规则,不同的词有不同的规则,因此难以抓住语言现象的共性。另外,对于复杂句子结构也难以仅仅利用对词的构造的规则进行分析,因为词的规则通常只在小范围发生作用。

小　　结

本章介绍了对于句子意义进行分析的两个方面——意义表示和分析方法，句子的意义表示主要介绍了基于逻辑形式的表示方案，并介绍了建立在此基础上的模型论语义；还介绍了几种不同的句义分析方法，从 5 .3～5 .6 节的几种方法中，越来越多地直接面向语义，而语法的使用越来越少。

第六章　语 言 模 型

在前面的几章中,介绍了一些基于规则的语言分析方法。在这些方法中,句子的分析和生成都是基于一些给定的规则进行推理而实现的。这些规则系统被乔姆斯基称为是人的内在语言能力,人们就是通过运用这些规则系统来进行语言活动的。但是,这种人的内在语言能力还没有得到完全证实,到目前为止人们还没有看到一个能完整地表现人类语言能力的规则系统,能看到的只是大量的语言材料(通常称为语料)。因此,一个很自然的想法就是,能否利用已有的语料(通常称为训练语料)对新产生的语言进行分析,或者利用已有的语料来帮助生成一些新的与已有语言不冲突的语言呢?由于已有语料是非常大量的,对这些语料的使用利用统计方法就显得非常自然了。这种把统计推理的方法运用到语言处理中,就是所谓的统计语言处理。把统计方法运用到语言处理中,尤其是在语音识别领域,已经取得了相当好的效果。从本章开始,简单介绍一些统计语言处理技术。主要从两个层次进行介绍,其一是在词层,这时不考虑任何的句子结构特点,只有词和词的关系,利用词的某种关联来进行统计推理;其二是在句子层,这时,通过在句子的结构上附加统计信息来进行统计推理。本章及第七、八章,主要涉及第一个层次,第九章涉及第二个层次。

本章首先在 6.1 节中从统计信息处理的角度看语言交流中的信息量问题,这样也就引出了 6.2 节中的 n 元语言模型。随后介绍对语言模型的一些改进工作。最后,介绍语言模型的质量评估指标。

本章名为语言模型,只是因为通常较多地把本章介绍的 n 元语言模型简称为语言模型。实际上,广义地说,语言模型可以用来指模拟语言生成和处理的任何技术方法。这样,可以简单地把语言模型分为语法语言模型,即前几章介绍的基于规则的语言处理方法,以及后面要介绍的统计语言模型。

6.1　语言与信息量

众所周知,语言的功能是在人与人之间实现信息传输,信息传输中信息量是一个十分重要的指标。如何衡量一个句子包含了多少信息呢?首先从统计的观点看一下一般的信息传输,然后再回到这个问题。

通常,信息传输的两端一个是信息发送者(信息源),另一个是信息的接收者(接收器)。信息源不断发射一个包含在确定集合 V 中的字符,字符的发送遵从一定的统计规

则。这些字符流形成消息,发送给接收器,接收器对于接收的字符序列具有一定的先验知识(这里,不考虑信息在传输过程中的问题,假定字符序列的传输过程是没有干扰的,接收的序列与发送序列完全相同)。这样,信息源通过发送字符序列来给接收器提供一定的信息,当发送的字符是统计独立并且同分布时,所发送的消息所携带的平均信息量定义为

$$I = \mathrm{lb}|V| \tag{6-1-1}$$

在发送的字符是独立但非同分布的情形下,用 w_i 表示 V 中的第 i 个字符。设该字符被信息源发送出去的概率是 $P(w_i)$,则字符流消息所携带的平均信息量是

$$H = -\frac{1}{|V|}\left[\sum_{i=1}^{|V|} P(w_i)\,\mathrm{lb}\,P(w_i)\right] \tag{6-1-2}$$

H 被称为熵。进一步,如果一个信息源所产生的字符为非独立的,那么,一个字符流消息 $W = w_1 w_2 \cdots w_N$ 的信息量为

$$H = -\sum P(w_1, w_2, \cdots, w_N)\,\mathrm{lb}\,P(w_1, w_2, \cdots, w_N) \tag{6-1-3}$$

式中,$P(w_1, w_2, \cdots, w_N)$ 为字符流消息 $W = w_1 w_2 \cdots w_N$ 被信息源发送出去的概率。

如果把这个模型用于描述人与人的语言交流,信息源和接收器就是人,字符是一种语言中的单词,V 是这种语言的词汇表。显然,单词之间一般是非独立的,通常后面的一个单词会受到前一个甚至几个单词的约束。这样,一个句子所包含的信息量就可以用式(6-1-3)来描述。

从式(6-1-3)中可以显然看出,一个词串 $W = w_1 w_2 \cdots w_N$ 的信息量是由组成该词串的各个单词 $W = w_1 w_2 \cdots w_N$ 的联合概率 $P(w_1, w_2, \cdots, w_N)$ 来决定的。从语言的角度来看,$P(w_1, w_2, \cdots, w_N)$ 就是某种语言按次序产生出词串 $w_1 w_2 \cdots w_N$ 的概率,这个概率的大小反映了这个词串在该语言中的使用情况,大的概率表明该词串经常在一起使用,而小的概率则表明该词串不常在一起使用。从这种叙述中也可以看到从统计的角度看语言现象与前述规则方法的不同,在规则方法中,我们不使用表示频度的副词“常,不常”等,而是该句子“合”或“不合”语法。

$P(w_1, w_2, \cdots, w_N)$ 通常是不可知的,需要统计估计。下面,把这个概念进行一下变换,从而引出在词汇层的统计语言模型。

6.2 N-Gram 模型

把 $P(w_1, w_2, \cdots, w_N)$ 分解成条件概率的形式:

$$P(W) = P(w_1, w_2, \cdots, w_N) = P(w_1)\prod_{i=2}^{N} P(w_i \mid w_1, w_2, \cdots, w_{i-1}) \tag{6-2-1}$$

式中的右边包含两个部分,$P(w_1)$ 为 w_1 的先验概率,这可以通过对大量语料中该词出现的频率简单统计而获得;第二部分是条件概率 $P(w_i \mid w_1, w_2, \cdots, w_{i-1})$,已知前面 $i-1$ 个单词为 $(w_1, w_2, \cdots, w_{i-1})$ 时,下一个词为 w_i 的概率。从另一个方面来看,计算这个概率

$\mathrm{lb}|V| = \log_2|V|$。

就是进行统计预测,即已知前面若干个词,预测下一个词可能是什么。为了使这种预测能够实现,通常需要一个假设,即某一个词出现的概率只依赖于它之前出现的 $i-1$ 个单词,这个假设即为马尔可夫假设。满足这个假设的模型称为 $i-1$ 阶马尔可夫模型;而在语言模型里,称之为 i 元模型。从语言学的角度来看,这个假设是可用的,例如,看如下的一个词串:

我吃了一个红_____。

在词“红”的后面可能会是什么词呢?它受词“红”的制约,因此不太可能是“香蕉”,因为,香蕉通常并不是红的,即概率 P(香蕉/红)的值很小。它也受到前面的“一个”的制约,因此不太可能是“糖水”,因为通常糖水不说“一个”,即概率 P(糖水/一个,红)的值很小。它还受到更前面的“吃”的制约,概率 P(桌子/吃了,一个,红)的值很小,因此不太可能是“桌子”。而是“苹果”的可能性就比较大。

利用这个例子,还要说明两个问题。

其一,上述例子中最后一个词出现的可能性大小依赖前面的 4 个词。即在 i 元模型中,$i=5$,模型称为 5 元模型。类似地,如果只依赖前面的 1 个词,就是 2 元模型;依赖前面的 2 个词,就是 3 元模型。

其二,很显然,利用前面较多的词来选择后面要出现的词,与只利用前面一个词来选择后面的词相比,准确性要高。但是,这种准确性的提高需要付出计算量的代价。在上例中,为了进行可能性的比较,还需要知道一些条件概率。在 2 元模型中,需要知道所有 2 个词之间的 $P(\cdot/\cdot)$;而在 3 元模型中,需要知道所有 3 个词之间的 $P(\cdot/\cdot,\cdot)$……以此类推。若假设某种语言中有词 1 000 个,则在几种模型中需要知道的条件概率的数目如表 6-1 所示。

表 6-1　语言模型中的参数估计数量

模　　型	需要知道的条件概率的数量
1 元模型	$1\ 000^2$ = 100 万
2 元模型	$1\ 000^3$ = 10 亿
3 元模型	$1\ 000^4$ = 10 000 亿
……	……

这些条件概率是模型需要利用已有语料估计的参数。从表中可以看到,为了得到更高的准确性,从 2 元模型到 3 元模型需要增加很多的参数估计量,在词表更大时,这种增幅更大。除了估计量的大幅增加,高阶模型的参数估计问题也比低阶模型的要复杂,从而降低估计值的可靠性,这反而会对预测的性能起反面的影响。

事实上,从表 6-1 可以看到,即使对于低阶模型,当词表较大时,需要的估计量也是十分大的。并且,在利用一定的语料进行估计时,仍然会出现数据稀疏等影响参数估计性能的问题。我们将在后面的几节详细考察这些问题。下面简单介绍表 6-1 中提出的几个低阶模型。

(1) 1 元模型

在 1 元模型中,一个单词的概率跟其前面的单词无关,即每个单词的出现都是独立的。此时,式(6-2-1)简化为

$$P(W) = P(w_1, w_2, \cdots, w_N) = \prod_{i=1}^{N} P(w_i) \tag{6-2-2}$$

由于这个模型过分简化,很少被单独使用。而它经常与 2 元语法和 3 元语法一起使

用,以实现平滑功能。

(2) 2 元模型

在 2 元模型中,某个单词在句子中出现的概率假设为仅与在它前面出现的那个单词有关。在这样的假设下,式(6-2-1)简化为下式:

$$P(W) = P(w_1, w_2, \cdots, w_N) = P(w_1)\prod_{i=2}^{N} P(w_i \mid w_{i-1}) \tag{6-2-3}$$

其中,词 w_i 出现的概率仅与 w_{i-1} 有关,由概率 $P(w_i \mid w_{i-1})$ 决定。概率 $P(w_i \mid w_{i-1})$ 的估计在最简单的情况下,可以如下计算:

如果词对 (w_{i-1}, w_i) 在训练语料中出现过,则 $P(w_i \mid w_{i-1})$ 的值为 1,否则为 0。这种简化的模型称为词对模型。而通常采用相对频率来估计这个条件概率(下节介绍)。

(3) 3 元模型

在 3 元模型中,某个单词在句子中出现的概率假设为与在它前面出现的两个单词都有关。在这样的假设下,式(6-2-1)简化为下式:

$$P(W) = P(w_1, w_2, \cdots, w_N) = P(w_1)P(w_2 \mid w_1)\prod_{i=3}^{N} P(w_i \mid w_{i-1}, w_{i-2}) \tag{6-2-4}$$

其中,词 w_i 出现的概率与 w_{i-1} 和 w_{i-2} 有关,由概率 $P(w_i \mid w_{i-1}, w_{i-2})$ 决定。

目前,2 元模型和 3 元模型使用得最多。

6.3 参数估计与平滑

本节介绍 n 元模型中条件概率的估计。利用语料数据中词汇同现的相对频率即可以得到条件概率的极大似然估计,如下式:

$$P(w_i \mid w_1, w_2, \cdots, w_{i-1}) = \frac{N(w_1, w_2, \cdots, w_{i-1}, w_i)}{N(w_1, w_2, \cdots, w_{i-1})} \tag{6-3-1}$$

其中,$N(w_1, w_2, \cdots, w_{i-1}, w_i)$ 是在训练语料中词串 $w_1 w_2 \cdots w_{i-1} w_i$ 出现的频次。

对于 1 元模型,每个单词的出现概率由下式计算:

$$P(w_i) = \frac{N(w_i)}{\sum_{j=1}^{|V|} N(w_j)} \tag{6-3-2}$$

对于 2 元模型,则由下式计算:

$$P(w_i \mid w_j) = \frac{N(w_i, w_j)}{\sum_{i=1}^{N} N(w_i, w_j)} \tag{6-3-3}$$

从上面的估计式中可以看到,那些没有在训练语料中出现的词串其估计量为 0。这样,在利用式(6-3-1)计算包含该子串的某个词串的概率时,整个词串的出现概率也为 0,即使该词串中包含其他具有较高概率的子串。从实际的语言现象来看,这是一个很严重的缺陷。因为对于任何一种语言,通常只有部分常用词经常使用,这些词在一般的训练语料中出现的频次都比较高;而另外有大量的是不常用的词,这些词同时出现的情况会更少。因此,对于一个确定的训练语料,即使规模相当大,也会有大量的词串没有同时出现,

这就不可避免地会出现大量的估计值为 0 的条件概率(并且,出现频次不为 0 但也比较低的那些词串用上述极大似然方法来估计也不好),这就是所谓的数据稀疏问题。

已有很多的研究表明,数据稀疏问题是十分严重的。例如,1983 年,Bahal 从英语的 IBM 激光专利文献语料库中抽取了 150 万词的语料作为训练语料进行 3 元模型的参数估计后,再用于来自相同语料库的语料分析,发现有 23% 的 3 元词串没有在训练语料中出现过。这样的训练语料规模从现在来看的确是小了,因此,人们希望通过增加语料的规模来解决数据稀疏问题。增加语料的规模显然能增加新的词串,对于解决数据稀疏问题是有帮助的。但是,实践证明,仅仅通过简单地增加语料规模是不可能完全克服这一问题的。Essen 和 Steinbiss 在 1992 年把含百万词的 LOB 语料的 75% 用来作训练语料,25% 作测试用,结果发现在测试语料中有 12% 的 2 元词串没有在训练语料中出现过。Brown 和 Dellapietra 等在 1992 年使用含 366 百万英语词的样本训练后,在新的样本材料中仍发现了 14 .7% 的新 3 元词串。事实上,无论多么大的语料,它也只是包含有限的语言现象,包含了有限的词串同现情况,而语言的生成能力是无限的。尤其对于那些出现频率低的词(这些词在各种语言中都占词表的很大部分),不可能在有限的语料中把它们的所有同现情况都收集到。为此,人们在扩大语料规模的同时,也从估计方法本身入手采用了一些避免 0 概率的方法,这些方法通称为统计平滑技术。这些方法多是基于这样的原则:适当减少训练语料中出现了的词串的概率,而把减少的那部分概率赋给在训练语料中没有出现的词串。从词的条件概率曲线来看,这样的操作通常会使曲线更加平滑一些。平滑因此而得名。

下式称为 Laplace 平滑,是最早采用的平滑技术:

$$P(w_i \mid w_1, w_2, \cdots, w_{i-1}) = \frac{N(w_1, w_2, \cdots, w_i) + 1}{N(w_1, w_2, \cdots, w_{i-1}) + B} \qquad (6\text{-}3\text{-}4)$$

式中,在极大似然估计的基础上分子加了 1,这就能保证即使词串 $w_1 \cdots w_i$ 没有在训练语料中出现,相应的条件概率也不会为 0; B 为词表大小相当的量。

分析表明,估计式(6-3-4)虽然只为未出现词串加了 1 次,但给未出现词串增加的总概率太多了。Church 和 Gale 在 1991 年报告了一个把 46 .5% 的概率空间分配给了未出现词串的实验结果。为此,一个更常采用的方法是 Lidstone 平滑,如下式:

$$P(w_i \mid w_1, w_2, \cdots, w_{i-1}) = \frac{N(w_1, w_2, \cdots, w_i) + \lambda}{N(w_1, w_2, \cdots, w_{i-1}) + B\lambda} \qquad (6\text{-}3\text{-}5)$$

式中,λ 是取值 0～1 之间的参数(λ = 0 时回到极大似然估计,λ = 1 时回到了 Laplace 平滑)。通过调整 λ,估计式(6-3-5)可以有效缓解式(6-3-4)的问题。但是,实际应用中,λ 的确定还没有什么好的办法。此外,若令

$$\mu = \frac{N(w_1, w_2, \cdots, w_{i-1})}{N(w_1, w_2, \cdots, w_{i-1}) + B\lambda}$$

则式(6-3-5)可变为下式:

$$P(w_i \mid w_1, w_2, \cdots, w_{i-1}) = \mu \frac{N(w_1, w_2, \cdots, w_i)}{N(w_1, w_2, \cdots, w_{i-1})} + (1 - \mu)\frac{1}{B} \qquad (6\text{-}3\text{-}6)$$

式(6-3-6)表明 Lidstone 估计与极大似然估计具有线性关系。下面介绍其他的一些平滑技术。

6.3.1 Good-Turing 平滑

Good 在 1953 年为回答 Turing 的问题而提出的基于相对频次的估计，在 n 元语法的平滑中一般不能直接使用。因为它是随机变量服从二项分布得到的结果。但是在大语料样本情况下，词的出现可以近似使用二项分布，所导出的平滑估计在使用中结果较好。以下来导出在 n 元模型中常用的 Good-Turing 平滑。

设 $W_1, W_2, \cdots, W_n$ 是在训练语料中出现的所有不同的 n 元语法的词串。(若对于 2 元语法，就是 2 个词长的词串，后面简称 2 元词串。以此类推。)设 n 元词串 W_i 在语料中出现的频次为 $c(W_i)$；N 为语料中所有 n 元词串出现的总频次，即等于 $c(W_i)$ 的总和。令 p_i 表示 W_i 的出现概率。我们的目标是对 p_i 进行估计，这可以利用 $c(W_i)$ 来进行。设 W_i 在训练语料中出现了 r 次，则有

$$E(p_i \mid c(W_i) = r) = \sum_{k=1}^{n} P(i = k \mid c(W_i) = r)\, p_k \tag{6-3-7}$$

又设在语料中出现且仅出现 r 次的 n 元词串的数目为 n_r，则

$$E_N(n_r) = \sum_{i=1}^{n} P(c(W_i) = r) = \sum_{i=1}^{n} \binom{N}{r} p_i^r (1 - p_i)^{N-r}$$

对于(6-3-7)中的 $P(i = k \mid c(W_i) = r)$ 有

$$P(i = k \mid c(W_i) = r) = \frac{P(c(W_k) = r)}{\sum_{l=1}^{n} P(c(W_l) = r)} = \frac{\binom{N}{r} p_k^r (1 - p_k)^{N-r}}{\sum_{l=1}^{n} \binom{N}{r} p_l^r (1 - p_l)^{N-r}}$$

$$= \frac{p_k^r (1 - p_k)^{N-r}}{\sum_{l=1}^{n} p_l^r (1 - p_l)^{N-r}} \tag{6-3-8}$$

将式(6-3-8)代入式(6-3-7)得到

$$E(p_i \mid c(W_i) = r) = \frac{\sum_{k=1}^{n} p_k^{r+1} (1 - p_k)^{N-r}}{\sum_{l=1}^{n} p_l^r (1 - p_l)^{N-r}} = \frac{(r+1)(N+1) \sum_{k=1}^{n} p_k^{r+1} (1 - p_k)^{N-r}}{(N+1)(r+1) \sum_{l=1}^{n} p_l^r (1 - p_l)^{N-r}}$$

$$= \frac{(r+1) \dfrac{(N+1)!}{(r+1)!(N-r)!} \sum_{k=1}^{n} p_k^{r+1} (1 - p_k)^{N-r}}{(N+1) \dfrac{N!}{r!(N-r)!} \sum_{l=1}^{n} p_l^r (1 - p_l)^{N-r}}$$

$$= \frac{(r+1) \sum_{k=1}^{n} \binom{N+1}{r+1} p_k^{r+1} (1 - p_k)^{N-r}}{(N+1) \sum_{l=1}^{n} \binom{N}{r} p_l^r (1 - p_l)^{N-r}}$$

$$= \frac{(r+1) E_{N+1}(n_{r+1})}{(N+1) E_N(n_r)}$$

若取 $E_N(n_r) \approx n_r$，$\dfrac{N}{N+1} E_{N+1}(n_{r+1}) \approx n_{r+1}$，则有

$$E(p_i \mid c(W_i) = r) \approx \frac{(r+1)n_{r+1}}{Nn_r}$$

通常令 $r^* = \frac{(r+1)n_{r+1}}{n_r}$，而 N 为标准化因子，得到 p_i 的 Good-Turing 估计式为

$$p_i \approx \frac{r^*}{N}$$

最后，把剩余的概率平均分配到 $r=0$ 的那些 n 元词串。这样，一个完整的用于 n 元模型的 Good-Turing 平滑可以定义为：

对于任何一个 n 元词串，如果 $c(W) = r > 0$ 时

$$P(W) = \frac{\hat{r}^*}{N} \tag{6-3-9}$$

否则，即 $c(W) = r = 0$ 时

$$P(W) = \frac{1 - \sum_{r=1}^{\infty} n_r \frac{\hat{r}^*}{N}}{n_0} \approx \frac{n_1}{n_0 N} \tag{6-3-10}$$

可以看到，和简单的极大似然估计方法相比较，n 元词串出现的绝对频次 r 被修改值 $r^* = \frac{(r+1)n_{r+1}}{n_r}$ 所替换。

进一步，Gale 和 Sampson 还提出了利用光滑曲线 $n_r = ar^b$ 来得到 n_r，该曲线的参数可以按已知的 (n_r, r) 数据对采用回归方法得到。

6.3.2 插值平滑

上述的几种平滑方法都是基于对绝对频次进行某种修正来平滑低频次词串的概率，这使得无论低频次的词串是什么，它们的地位都是一样的（因为对于它们而言，修正估计所依据的信息是相同的，那就是它们具有相同的低出现频次），因此估计值是一样的。而通常，我们有理由希望，对于两个具有相同低频次的词串 W 和 V，如果 W 包含的子串比 V 包含的子串具有更高的出现频次的话，那么，W 在未来语料中出现的概率应该比 V 要大。这种希望的实现显然需要在估计高阶语言模型的参数时使用低阶语言模型的参数，而使用插值技术可以完成这种任务。下面，不失一般性，以 3 元模型为例，说明插值平滑技术。参数估计式：

$$P(w_n \mid w_{n-2}, w_{n-1}) = \lambda_1 P(w_n) + \lambda_2 P(w_n \mid w_{n-1}) + \lambda_3 P(w_n \mid w_{n-2}, w_{n-1}) \tag{6-3-11}$$

其中的权值 λ_1，λ_2 和 λ_3 满足

$$0 \leq \lambda_i \leq 1 \qquad i = 1,2,3$$

且

$$\lambda_1 + \lambda_2 + \lambda_3 = 1$$

可以看到，式(6-3-11)中，在 λ_1，λ_2 均不为零时，对 3 元模型参数的估计综合利用了 1 元模型和 2 元模型的信息。λ_1，λ_2 的取值代表了 1 元模型和 2 元模型的值对 3 元模型参数估计的重要度。这几个权值可以在计算前自行决定，也可以利用自动（如 EM 算法等）方法来选择。

式(6-3-11)是一个线性插值，权值 λ_1，λ_2 和 λ_3 一旦决定，就与出现的词或词串无关了。这可以更进一步的扩展为

$$\begin{aligned} & P(w_n \mid w_{n-2}, w_{n-1}) \\ = \ & \lambda_1(w_n) P(w_n) + \lambda_2(w_{n-1}) P(w_n \mid w_{n-1}) + \lambda_3(w_{n-2}, w_{n-1}) P(w_n \mid w_{n-2}, w_{n-1}) \end{aligned} \tag{6-3-12}$$

其中的权值为 $\lambda_1(w_n)$，$\lambda_2(w_{n-1})$ 和 $\lambda_3(w_{n-2}, w_{n-1})$，它们的取值与词或词串有关。当然也须满足

$$0 \leq \lambda_i \leq 1 \qquad i = 1,2,3$$

且

$$\lambda_1 + \lambda_2 + \lambda_3 = 1$$

这样，就可以更灵活地利用历史信息。如果当前的估计准确(相关的数据较多时)，可以把历史信息的权值调低；相反的情况时，就可以增加历史信息的权值。这个灵活性是固定权值的线性方法所不具备的，当然，这将不可避免地增加计算量。

在上述插值估计式中，低阶模型参数的获得也可以有不同的方法。可以直接用极大似然估计，如 Jelinek-Mercer 平滑(这和前述基本类似，不再说明)；也可以用前面介绍的 Good-Turing 估计，如 Katz 平滑。以 2 元语法为例，一种简单的 Katz 平滑如下式所示：

$$P(w_2 \mid w_1) = \begin{cases} P_{GT}(w_2 \mid w_1) & c(w_1, w_2) > 0 \\ \alpha(w_1) P(w_2) & c(w_1, w_2) = 0 \end{cases} \tag{6-3-13}$$

当 $c(w_1, w_2) = 0$ 时，2 元语法的估计是通过回退到 1 元语法而得到的。

6.4 基于词聚类的语言模型

上述的平滑技术是解决数据稀疏问题的一个重要方面。另一方面，从前面参数估计量的表 6-1 中已经看到，模型参数的估计量是与词表的数量直接相关的。模型参数数目的下降与词表数目的下降成指数关系，减少词表，对于降低参数估计的计算量十分有益。而更为重要的是，这对于解决数据稀疏问题也非常有帮助。因为，对于确定规模的一个训练语料库，如果所涉及的词表规模较小，那么，同等条件下，其中 0 出现频次的词串与大词表时相比当然要少。这即是说，数据稀疏问题要小。目前，词表的减少通常从两个方面来入手，一方面是直接减少词汇表中的词汇，只考察包含一部分词汇的语言；另一个方面是本节要介绍的词表压缩，即不是减少词汇的数量，而是通过词聚类来形成新的压缩词表。

所谓词聚类就是根据需要把具有相似特点的词分为若干类，这些类成为一个新的词表，每一类都是新词表中的一个词。在计算频次时，同一类中的所有原词表中的词均视为新词表中的同一个词。设词表 V，聚类 π 把 V 中的词映到 C，$|V| \geq |C|$。对 " $w_i \in V$ 都存在 $c_i \in C$，(这里，为了描述的方便，V 和 C 中的元素用了相同的指标，但是通常 V 到 C 的映射是多到 1 的，并不会一一对应，下式也是如此。)使得 $\pi(w_i) = c_j$。并且对 $1 \leq k \leq n$，满足

$$P(w_k \mid w_1, \cdots, w_{k-1}) = P(w_k \mid c_k) P(c_k \mid c_1, \cdots, c_{k-1})$$

这样,由于$|V| \geq |C|$,所以语言模型的参数规模降低了。但是,这种参数规模的下降是有损失的。正如 Brown 所述,“星期五”和“星期四”可以归入一类,但是,二者之间在使用上还是有差别的。而一旦归入一类后,其间的差别将不可见。一个好的聚类,应该使信息损失在某种度量下最小。而一个好的基于词聚类的语言模型的质量,就取决于聚类的质量。由于聚类问题本身包含很多复杂的研究内容,这里不作介绍。

6.5 语言模型的评估

显然,一个语言模型的性能(质量)可以通过其在语言处理系统的最终表现来评估,比如用于词预测功能时,预测的错误率越低则模型越好。但是,通常,由于完整的语言处理系统涉及到的语言处理任务较多,各种处理之间相互影响,具有很大的复杂性。因此,常采用其他一些度量来评估语言模型,这主要是一些信息论中关于熵的量。语言模型的复杂度 PP(Perplexity)是其中经常使用的一个量。下面介绍复杂度。

前面已经知道,一段文本的熵如式(6-1-2)所示,其中 $P(w_i)$为 w_i 的真正的分布,在所有 w_i 独立同分布时,熵取得式(6-1-1)所示最大值 $\mathrm{lb}|V|$, $H \leq \mathrm{lb}|V|$。但 $P(w_i)$为未知的,只能用语言模型的估计值 $P(w_i|h_i)$来代替 $P(w_i)$。

为定义复杂度,先如下式定义对数概率 LP(Logprob):

$$LP = -\frac{1}{N}\sum_{i=1}^{N} \mathrm{lb}\ P(w_i|w_{i-1}) \tag{6-4-1}$$

则复杂度 PP 如下式所定义:

$$PP = 2^{LP} = 2^{-\frac{1}{N}\sum_{i=1}^{N} \mathrm{lb}\ P(w_i|w_{i-1})} \tag{6-4-2}$$

复杂度的含义,粗略地说,是对模型选择下一个词的范围大小的度量。例如,对于一个语音识别系统,复杂度表示的就是识别器每次将在多大的一个词集合中选择下一个词。显然,复杂度越大,识别器的识别难度就越大。复杂度比简单地用词表大小衡量识别难度要更为可靠。

从复杂度的定义可以看到,语言模型的复杂度依赖于用于评估它的语言数据。在训练语料上具有小的复杂度只表明语言模型对训练语料具有好的逼近能力,但是,目前并不能够保证在测试集上一定有小的值。如果在训练集上复杂度很小,但是在测试集上较大,说明语言模型的推广能力差,并称语言模型被过训练。一般,越是复杂的语言模型,由于逼近能力强,比较容易出现过训练的问题。

反之,由于复杂度依赖于语言数据,也可以利用同一个语言模型在不同测试集上的复杂度来评估语言数据的复杂度。

对于两种语言模型的比较,很显然,为保证不同语言模型比较的客观性,就应该要求两个模型必须在相同的训练集上训练,在相同的测试集合上测试。

小　结

本章开始介绍统计语言处理技术。n 元模型是一种被广泛使用的统计语言模型,模

型抛开已有的语言学规则,完全建立在对语料进行统计推理的基础上,因此,语料质量和规模对其参数估计都有重要影响。鉴于目前的研究状况,没有对语料本身的问题作介绍。本章首先通过语言中的信息量来引入 n 元模型,然后就模型中的数据稀疏问题介绍了一些平滑技术。这些技术中,有的方法已在使用中被证明是有效的,如简单的 katz 平滑;而有的还需要更多的验证或改进,如大规模的基于词聚类的语言模型。最后介绍了语言模型评估中的复杂度的概念。

第七章　隐马尔科夫模型

隐马尔科夫模型(HMM: Hidden Markov Model)被公认为是语音识别领域中最成功的统计模型之一,词性标注是隐马尔科夫模型在自然语言处理中的另一个成功应用。目前,其各种变形广为使用。通过假设一个“隐藏”结构的存在,隐马尔科夫模型能够把简单的词序信息和更高层的语言信息联系起来,以完成诸如词性标注这样的任务。

隐马尔科夫模型是建立在马尔科夫模型的基础之上的。因此,在进入隐马尔科夫模型之前,先在 7.1 节介绍马尔科夫模型。由于语言现象均为离散时间和离散状态的,所以,这里只限于介绍马尔科夫链。在 7.2 节定义隐马尔科夫模型。在 7.3 节解决隐马尔科夫模型的三个基本问题。7.4 节介绍用隐马尔科夫模型建模词性标注任务。

7.1　马尔科夫模型

首先定义马尔科夫链。马尔科夫链可以形式地描述如下:

随机序列 X,在 t 时刻的状态记为 q_t, q_t 在有限状态集合 $S=(s_1, s_2, \cdots, s_N)$中取值。如果 X 在 $t+k$ 时刻的状态 $q_{t+k}=s_i \in S$ 与其在 t 时刻以前所处的状态无关,即满足下式:

$$P(q_{t+k}=s_i \mid q_1, q_2, \cdots, q_t, \cdots, q_{t+k-1}) = P(q_{t+k}=s_i \mid q_t, \cdots, q_{t+k-1}) \qquad (7\text{-}1\text{-}1)$$

则 X 称为 k 阶马尔科夫链, P 称为转移概率。

若式(7-1-1)等号后面的概率与时间 t 无关,即对 $\forall\ m \neq t$,满足下式:

$$P(q_{t+k}=s_i \mid q_t, \cdots, q_{t+k-1}) = P(q_{m+k}=s_i \mid q_m, \cdots, q_{m+k-1}) \qquad (7\text{-}1\text{-}2)$$

则马尔科夫链是时不变或平稳的。

对于 1 阶平稳马尔科夫链,定义

$$a_{ij} = P(q_{t+1}=s_j \mid q_t = s_i) \qquad (7\text{-}1\text{-}3)$$

为转移概率。a_{ij}满足:

$$\forall\ i, j,\ a_{ij} > 0 \text{ 且 } \forall\ i, \sum_{j=1}^{N} a_{ij} = 1$$

矩阵 $A=(a_{ij})_{N\times N}$为状态转移矩阵。

除此之外,要完整描述一个 1 阶马尔科夫链,还应说明每一个状态的初始概率。

$$\pi_i = P(q_1 = s_i) \qquad i = 1, 2, \cdots, N$$

在马尔科夫链中,一个很重要的问题是状态序列 $X=(q_1, q_2, \cdots, q_T)$的联合分布。以 1 阶马尔科夫链为例,状态序列 $X=(q_1, q_2, \cdots, q_T)$的联合概率可以如下计算:

$$P(q_1, q_2, \cdots, q_T) = P(q_1)P(q_2 \mid q_1)P(q_3 \mid q_1, q_2)\cdots P(q_T \mid q_1, q_2, \cdots, q_{T-1})$$
$$= P(q_1)P(q_2 \mid q_1)P(q_3 \mid q_2)\cdots P(q_T \mid q_{T-1}) \quad (7\text{-}1\text{-}4)$$

7.2 隐马尔科夫模型的描述

在上述马尔科夫链中,随机序列的输出值(观察值)即为其状态,这样,随机序列的状态是可直接观察到的,因此,也称为可观察马尔科夫模型。在许多实际问题中,随机序列的内部状态与其输出并不相同,例如下面叙述的碗中取球问题。

有 N 个已编号的碗,每个碗中装有确定数量的各种着色球,所有的球共有 M 种颜色。一个人甲每次首先随机选定一个碗,再从这个碗中取出一个着色球,把该球的颜色告诉另一个人乙;对乙而言,能观察到的只是一个着色球的颜色序列,而不知道每个球是从哪个碗中取出的。这里存在两个随机过程,一个是碗的编号序列;另一个是观察到的着色球的颜色序列。显然,碗的编号序列与颜色序列存在某种联系,如何完整地描述整个系统,只依靠可观察马尔科夫模型不行,而需要用所谓的隐马尔科夫模型来刻画。在语言处理中,也有一些类似的任务,实践已经证明,用隐马尔科夫模型可以较好地进行建模,这在引入隐马尔科夫模型后介绍。

下面在 1 阶马尔科夫链的基础上说明隐马尔科夫模型。一个隐马尔科夫模型可以由如下几个要素构成。

(1) 模型的状态。设状态集合为 $S = \{s_1, s_2, \cdots, s_N\}$,时刻 t 时所处的状态为 $q_t \in S$。状态之间可以互相转移。

(2) 描述状态之间如何进行转移的状态转移矩阵 $A = (a_{ij})_{N\times N}$, a_{ij} 如式(7-1-3)所描述。

(3) 模型的观察值。设观察值集合为 $V = \{v_1, v_2, \cdots, v_M\}$,当 t 时刻的状态转移完成的同时,模型都产生一个可观察输出 $o_t \in V$。

(4) 描述产生输出的概率分布矩阵 $B = (b_{ij})_{N\times M}$。其中,

$$b_{ij} = b_i(j) = b_i(v_j) = P(o_t = v_j \mid q_t = s_i) \qquad 1 \leq i \leq N, 1 \leq j \leq M \quad (7\text{-}2\text{-}1)$$

表示 t 时刻状态为 s_i 时输出为 v_j 的概率。

(5) 模型的初始状态分布。设为 $\pi = \{\pi_1, \pi_2, \cdots, \pi_N\}$,其中,

$$\pi_i = P(q_1 = s_i) \qquad 1 \leq i \leq N$$

这样,一个隐马尔科夫模型可以由五元组(S, A, V, B, π)完整描述。但实际上,A, B 中包含对 S, V 的说明。因此,通常用

$$\lambda = \{A, B, \pi\}$$

来记一组完备的隐马尔科夫模型参数。

图 7-1 是一个具有三个状态的隐马尔科夫模型。图中显示的一个状态序列为$(s_1, s_1, s_2, s_2, s_3)$,其输出的观察序列为$(o_1, o_2, o_3, o_4, o_5)$。

在隐马尔科夫模型中,由于模型中对外表现出来的是观察向量序列 $O = \{o_1, \cdots\}$,内部状态序列 $Q = \{q_1, \cdots\}$不能直接观察得到,因此而称为“隐”马尔科夫模型。

在上述给定的模型框架下,为使隐马尔科夫模型能够用于解决实际问题,首先需要解决三个基本问题,它们是:

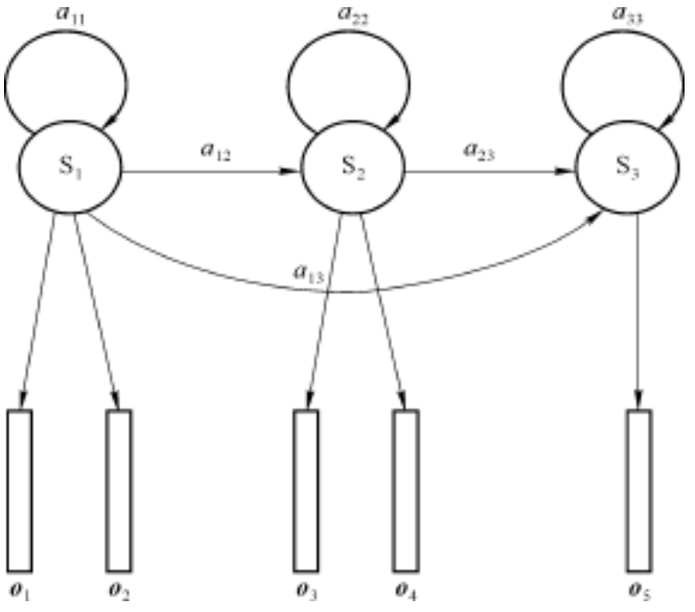


图 7-1　具有三个状态的隐马尔科夫模型及其观察向量

问题一:给定观察向量序列 $O=(o_1, o_2, \cdots, o_T)$和隐马尔科夫模型 $\lambda=(A, B, \pi)$,如何计算由该模型产生该观察序列的概率 $P(O|\lambda)$。

问题二:给定观察向量序列 $O=(o_1, o_2, \cdots, o_T)$和隐马尔科夫模型 $\lambda=(A, B, \pi)$,如何获取在某种意义下最优的内部状态序列 $Q=(q_1, q_2, \cdots, q_T)$。

问题三:如何选择(或调整)模型的参数 λ,使得在该模型下产生观察序列 O 的概率 $P(O|\lambda)$最大。

从另一个角度去看问题一,可以看到它实际上是一个评估问题,即计算给定的模型和观察序列的匹配程度。这个观点十分有用,当用几个模型去竞争匹配给定的观察序列时,问题一的求解使我们可以从中选出一个最合适的模型。问题二也被称为解码问题,即根据给定的模型和观察序列,寻找最有可能生成这个观察序列的内部状态。问题二的求解也会在问题三中接触到。问题三是训练问题,即在给定一些观察序列作为样本的条件下优化模型参数,使得模型能够最佳地描述这些观察序列。可见,问题三是所有隐马尔科夫模型应用的基础。如果不能解决训练问题,就根本无法得到隐马尔科夫模型,也就无所谓问题一和问题二了。

7.3　隐马尔科夫模型基本问题的解决

本节分别给出隐马尔科夫模型中上述三个问题的数学求解方法。

7.3.1　解决第一个基本问题

问题一为计算由给定模型 λ 生成某一观察序列$(o_1 o_2 \cdots o_t)$的概率 $P(O|\lambda)$。为此,先定义前向变量 $\alpha_t(i)$:

$$\alpha_t(i) = P(o_1, o_2, \cdots, o_t, q_t = s_i \mid \lambda) \tag{7-3-1}$$

前向变量表示的是在给定模型 λ 下，时刻 1 至时刻 t 产生的观察序列为 $(o_1\ o_2 \cdots o_t)$ 且 t 时刻系统状态为 s_i 的概率。可以如下迭代求解 $\alpha_t(i)$。

(1) 初始化

$$\alpha_1(i) = \pi_i b_i(o_1) \qquad 1 \leq i \leq N$$

(2) 迭代

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i)\, a_{ij}\right] b_j(o_{t+1})$$

$$1 \leq j \leq N, 1 \leq t \leq T-1$$

(3) 终止

$$P(O \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i)$$

图7-2为前向变量迭代求解过程的示意图。可以看到，在 $t+1$ 时刻，第 j 个状态 s_j 可以由 t 时刻的 N 个状态转移而至。由于 $\alpha_t(i)$ 是时刻 t 时处于第 i 个状态 s_i 和产生观察序列 $(o_1\ o_2 \cdots o_t)$ 的联合概率，故 $\alpha_t(i)\ a_{ij}$ 是时刻 $t+1$ 时由第 i 个状态 s_i 转移至第 j 个状态 s_j 和产生观察序列 $(o_1\ o_2 \cdots o_t)$ 的联合概率。i 从 1 取到 N，这 N 个乘积加在一起，就获得了时刻 $t+1$ 时处于第 j 个状态 s_j 和产生观察序列 $(o_1\ o_2 \cdots o_t)$ 的联合概率。然后乘以在第 j 个状态产生观察向量 o_{t+1} 的概率，就获得了 $\alpha_{t+1}(j)$。

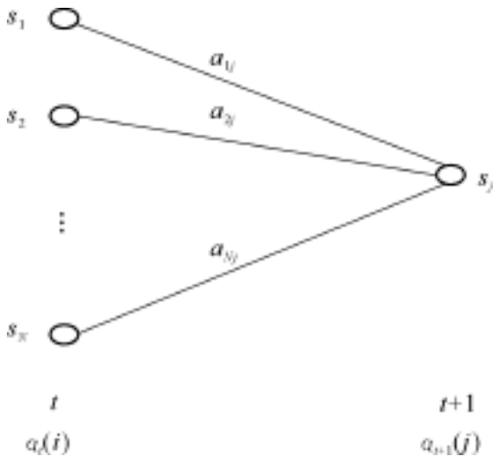


图 7-2　前向变量迭代求解过程示意

可以类似地定义反向变量 $\beta_t(i)$：

$$\beta_t(i) = P(o_{t+1}, \cdots, o_T \mid q_t = s_i, \lambda) \tag{7-3-2}$$

即在给定模型 λ 且 t 时刻状态为 s_i 的条件下，自时刻 $t+1$ 至时刻 T 产生观察序列 $(o_{t+1} \cdots o_T)$ 的概率。

反向变量同样可以如下迭代计算。

(1) 初始化

$$\beta_T(i) = 1 \qquad 1 \leq i \leq N$$

(2) 迭代

$$\beta_t(i) = \sum_{j=1}^{N} \beta_{t+1}(j)\, a_{ij} b_j(o_{t+1})$$

$$1 \leq i \leq N,\ t = T-1, T-2, \cdots, 1$$

尽管在问题一的求解中只需要计算前向变量，但反向变量在问题三的求解中需要使用。

7.3.2　解决第二个基本问题

问题二和问题一有所不同，问题一能够给出一个确切的解；而问题二却可能有一些不同的解。问题就在于问题二叙述中所说的“在某种意义下”最优是什么含义，不同的最优

标准得到的结果可能是不同的。

一个可能的标准是这样的:在给定的观察序列 O 下,Q 是由每一个时刻 t 最有可能处于的状态 q_t 所构成的。该优化标准使状态序列 Q 中正确状态的期望数量最大化。为了求解这种标准下的问题二,定义变量

$$\gamma_t(i) = P(q_t = s_i \mid O, \lambda)$$

即在给定模型 λ 和观察序列 O 的条件下,系统在 t 时刻状态为 s_i 的条件概率。利用前面所提及的前向变量和反向变量,这个条件概率可以表示为

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(O \mid \lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N}\alpha_t(j)\beta_t(j)}$$

这是因为 $\alpha_t(i)$ 是 t 时刻状态为 s_i 且产生观察序列 $(o_1 o_2 \cdots o_t)$ 的联合概率;$\beta_t(i)$ 为 t 时刻状态为 s_i 的条件下,自时刻 $t+1$ 至时刻 T 产生观察序列 $(o_{t+1} o_{t+2} \cdots o_T)$ 的概率;而归一化因子 $P(O \mid \lambda) = \sum_{j=1}^{N}\alpha_t(j)\beta_t(j)$ 使得变量 $\gamma_t(i)$ 满足

$$\sum_{i=1}^{N}\gamma_t(i) = 1$$

利用变量 $\gamma_t(i)$,可以求解得每一时刻 t 时最有可能的状态为

$$q_t = \underset{i=1}{\overset{N}{\arg\max}}[\gamma_t(i)]$$

尽管以上求解每一时刻最有可能状态的解法使状态序列中正确状态的期望数得到了最大化,但最终获得的状态序列可能存在着一些问题。例如,在隐马尔科夫模型中,某些状态之间的转移概率可能为 0(如图 7-1 所示的隐马尔科夫模型中的 a_{21}, a_{31}, a_{32}),这样一来,某些“最优”内部状态序列在概率上是不可能的。例如图 7-1 所示的隐马尔科夫模型中,由于 $a_{21}=0$,因此内部状态序列 $Q=(s_1, s_2, s_1, s_2, s_3)$ 是不可能出现的。

之所以出现上述的情况,是由于这个最优标准仅仅考虑了在每一个孤立时刻位于某一状态的可能,而没有考虑整个内部状态序列是否存在出现的可能性。

最常用的问题二的解是针对上述最优定义的一个修改。此时,最优内部状态序列被定义为在该观察序列 O 的条件下,最可能的内部状态序列(一个完整的最优路经)。即最优内部状态序列 Q 为

$$\begin{aligned} Q &= \arg\max P(Q \mid \lambda, O) = \arg\max[P(Q \mid \lambda, O)P(O)] \\ &= \arg\max P(Q, O \mid \lambda) \end{aligned} \tag{7-3-3}$$

对此问题的求解可以用基于动态规划思想的 Viterbi 算法。

为了求解在观察序列 $O=(o_1, o_2, \cdots, o_T)$ 的条件下,最优的内部状态序列 $Q=(q_1, q_2, \cdots, q_T)$,定义变量:

$$\delta_t(i) = \max_{q_1, q_2, \cdots, q_{t-1}} P(q_1, \cdots, q_t = s_i, o_1, \cdots, o_t \mid \lambda) \tag{7-3-4}$$

该变量意为在 t 时刻,沿着一条路径抵达状态 s_i,并生成观察序列 $(o_1 o_2 \cdots o_t)$ 的最大概率。利用迭代,有

$$\delta_{t+1}(j) = \left[\max_{i=1}^{N}\delta_t(i)a_{ij}\right]b_j(o_{t+1})$$

为了能够得到最优的状态序列,在求解过程中,对每一个时刻和状态,需要保留使得上式中最大化条件得以满足的上一时刻的状态。完整的算法如下所述。

(1) 初始化

$$\delta_1(i) = \pi_i b_i(o_1) \qquad 1 \leq i \leq N$$

$$\Psi_1(i) = 0 \qquad 1 \leq i \leq N$$

(2) 迭代

$$\delta_{t+1}(j) = \left[\max_{i=1}^{N} \delta_t(i)\, a_{ij}\right] b_j(o_{t+1}) \qquad 1 \leq j \leq N, 1 \leq t \leq T-1$$

$$\Psi_{t+1}(j) = \left[\operatorname{argmax}_{i=1}^{N} \delta_t(i)\, a_{ij}\right] b_j(o_{t+1}) \qquad 1 \leq j \leq N, 1 \leq t \leq T-1$$

(3) 终止

$$P^* = \max_{i=1}^{N}[\delta_T(i)]$$

$$q_T^* = \operatorname{argmax}_{i=1}^{N}[\delta_T(i)]$$

(4) 回溯

$$q_t^* = \Psi_{t+1}(q_{t+1}^*) \qquad t = T-1, T-2, \cdots, 1$$

除了回溯的步骤之外,问题二的解和问题一的解是类似的,主要的不同是在迭代过程中,求和的步骤变为最大化。事实上,如果认为每一个观察序列都是由一个与它最相关的内部状态序列生成的,那么在问题一的解中,求和步骤也可以近似地用最大化代替,即

(1) 初始化

$$\delta_1(i) = \pi_i b_i(o_1) \qquad 1 \leq i \leq N$$

(2) 迭代

$$\delta_{t+1}(j) = \left[\max_{i=1}^{N} \delta_t(i)\, a_{ij}\right] b_j(o_{t+1}) \qquad 1 \leq j \leq N, 1 \leq t \leq T-1$$

(3) 终止

$$P(O|\lambda) \approx \max_Q P(O, Q|\lambda) = \max_{i=1}^{N}[\delta_T(i)]$$

7.3.3 解决第三个基本问题

问题三是模型的参数估计问题,即依据一些观察序列,估计一组隐马尔科夫模型的参数(A, B, π),使得在该参数模型下,产生这些观察序列的概率最大化。到目前为止,训练问题没有已知的解析解法。事实上,在给出一些观察序列作为训练数据之后,不存在最佳的计算模型参数的方法。通常使用诸如 Baum-Welch 法(等价于 Estimation-Maximization 法)、梯度下降法等迭代的方法,将模型参数 $\lambda = (A, B, \pi)$ 调整至 $P(O|\lambda)$ 的局部极值。这是一个参数重估的迭代过程。为了便于描述,首先定义 $\xi_t(i, j)$ 为在给定模型 $\lambda = (A, B, \pi)$ 和观察序列 O 的条件下,在 t 时刻状态为 s_i 且 $t+1$ 时刻状态为 s_j 的概率,即

$$\xi_t(i, j) = P(q_t = s_i, q_{t+1} = s_j | O, \lambda) \tag{7-3-5}$$

如图 7-3 所示,依据前向变量和反向变量的定义,可以将 $\xi_t(i, j)$ 写为以下形式:

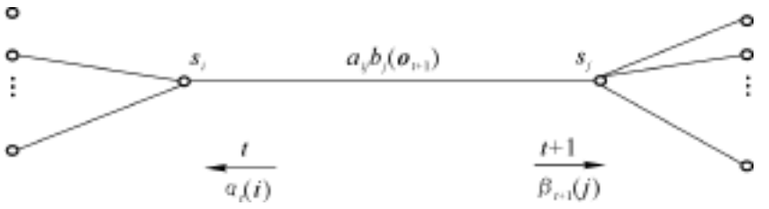


图 7-3　t 时刻位于状态 s_i 且 $t+1$ 时刻位于状态 s_j 的联合事件

$$\xi_t(i,j)=\frac{\alpha_t(i)\,a_{ij}b_j(o_{t+1})\,\beta_{t+1}(j)}{P(O\mid\lambda)}=\frac{\alpha_t(i)\,a_{ij}b_j(o_{t+1})\,\beta_{t+1}(j)}{\sum_{k=1}^{N}\sum_{l=1}^{N}\alpha_t(k)\,a_{kl}b_l(o_{t+1})\,\beta_{t+1}(l)}$$

式中，分子即为 $P(q_t=s_i,\ q_{t+1}=s_j,\ O\mid\lambda)$，除以分母($P(O\mid\lambda)$)后，归一化条件得以满足。

此前已定义了 $\gamma_t(i)$为在给定模型参数 λ 和观察序列 O 的条件下，时刻 t 位于状态 s_i 的条件概率，现在可以通过将 $\xi_t(i,j)$对 j 求和把两者联系起来，即

$$\gamma_t(i)=\sum_{j=1}^{N}\xi_t(i,j)$$

如果将 $\gamma_t(i)$对下标 t 求和，将可以得到在观察序列 O 下状态 s_i 的期望出现次数；如果在求和过程中除去 $t=T$ 这一项，就得到了在观察序列 O 下，由状态 s_i 转移到其他状态的期望次数。类似地，将 $\gamma_t(i)$对下标 t 自 1～$T-1$ 求和，就可以得到在观察序列 O 下，由状态 s_i 转移到状态 s_j 的期望次数。即

$$\sum_{t=1}^{T-1}\gamma_t(i)=\text{由状态 }s_i\text{ 转移出的期望次数}$$

$$\sum_{t=1}^{T-1}\xi_t(i,j)=\text{由状态 }s_i\text{ 转移至 }s_j\text{ 的期望次数}$$

利用以上所描述的公式和概念，可以给出如下一组隐马尔科夫模型的参数重估公式：

$$\bar{\pi}_i=\text{在时刻 }t=1\text{ 时位于状态 }s_i\text{ 的期望次数}=\gamma_t(1)\qquad(7\text{-}3\text{-}6a)$$

$$\bar{a}_{ij}=\frac{\text{由状态 }s_i\text{ 转移至状态 }s_j\text{ 的期望次数}}{\text{由状态 }s_i\text{ 转移出的期望次数}}=\frac{\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T-1}\gamma_t(i)}\qquad(7\text{-}3\text{-}6b)$$

$$\bar{b}_j(v_k)=\frac{\text{由状态 }s_j\text{ 输出观察向量 }v_k\text{ 的期望次数}}{\text{位于状态 }s_j\text{ 的期望次数}}=\frac{\sum_{1\le t\le T,\ s.t.\ o_t=v_k}\gamma_t(i)}{\sum_{t=1}^{T}\gamma_t(i)}\qquad(7\text{-}3\text{-}6c)$$

从一个初始模型 $\lambda=(A,\ B,\ \pi)$开始，可以利用上面的一组重估公式得到新的模型 $\bar{\lambda}=(\bar{A},\ \bar{B},\ \bar{\pi})$来代替原模型。如此不断迭代，Baum 等人证明了，新的模型的 $P(O\mid\lambda)$将不断变大，直到抵达局部的极值点。最终获得的隐马尔科夫模型被称为极大似然模型，该模型使产生观察序列 O 的概率最大化。

解决了上述三个基本问题，隐马尔科夫模型就可以用于解决实际问题。以下介绍用隐马尔科夫模型来对词性标注任务进行建模。

7.4 词性标注

词性是词的一个重要属性。常用的词性有名词、动词等等。表面上看,词性是按照它们的意义来分类的,例如那些指代人、地点、事件等的词常为名词,而指代动作的词常为动词。这种根据意义的划分很容易找到反例,例如"战争"与"打仗"两个词的意义是类似的,但是"战争"通常是作为名词用,而"打仗"通常作为动词用。目前,更常使用的词性的定义是通过词的同现属性来进行的。例如,朱德熙在《句法讲义》中认为名词的语法特点是:①可以受数量词修饰;②不受副词修饰。

某些词只有一种词性,这种词无论出现在文本的什么位置,其词性都相同,如"我们"总是代词。而有一些词有两种甚至两种以上的词性,这些词在文本的不同位置有不同的词性取值。例如,词"希望",在句子"大家希望天是蓝色的。"中为动词;而在短语"未来的希望"中为名词。为词标明其在上下文中的词性就是所谓的词性标注。

词性标注是自然语言处理的重要一步。可以认为,使用经过词性标注的语料是提高当前自然语言处理系统精度和实用性的一个比较好的"折衷点"。和生语料相比,它可以提供更丰富的信息;而尽管语法分析树可以提供更多的信息,但是,就当前的句法分析水平而言,精度远远不能够达到要求,并且句法分析器的构建耗费也比较高。

词性标注可以用在机器翻译、信息抽取、信息检索,以及更高层的语法处理中。在机器翻译(本段机器翻译的相关术语可以参见第九章"机器翻译")中,源语言中一个单词翻译到目标语言中一个单词的概率跟源语言单词的词性密切相关。例如,英语单词"hide"作为名词时汉语意思为"皮",作为动词时汉语意思为"隐藏"。因此,一旦知道"hide"的词性,就可以很容易进行翻译了。在前面介绍基于语法规则的句法分析技术中,进行句法分析的一个必要的前提是要有句子中出现的每个词的词性信息。在句法分析过程中,如果某个词有几个不同的词性,则在语法分析中必须利用某些方法选择其中的一个,这是句法分析中一个十分耗费计算量的部分。如果能在进行句法分析之前就为每一个词分配唯一一个正确的词性,就可以大大提高句法分析的效率。

词性标注的难点主要在于:①某些单词有多个词性,在不同的上下文中取到不同的值,因此,对这些词标注需要考察其所在的上下文;②新词的出现,由于新词没有在词典中出现,不知道其词性,需要特别的技术来处理。

下面用隐马尔科夫模型对词性标注的任务建模。

词性标注就是寻找一个词性序列 $T=\{t_1,\cdots,t_n\}$,使得它对于单词序列 $W=\{w_1,\cdots,w_n\}$是最优的。其中,t_i 为 w_i 的词性,$t_i\in S_t$,$i=1,\cdots,n$,S_i 为单词所有可能的词性的集合,通常也称为标注集。

如果假设词性序列是一个马尔科夫链,这个马尔科夫链在每次进行状态转移时都产生一个单词,具体产生哪个单词由其所处的状态决定。这样,可以很容易把词性标注和上述的隐马尔科夫模型联系起来。词性序列 $T=\{t_1,\cdots,t_n\}$对应于模型的状态序列,而标注集对应于状态集,词性之间的转移对应于模型的状态转移。假设词性序列的马尔科夫链是1阶的,即每个词的词性都只依赖前面一个词性而决定,因此有状态转移概率

$P(t_i \mid t_j)$，这对应于模型中的状态转移概率 a_{ij}。而单词为模型的观察值，不同的词性状态对不同的单词有不同的输出概率。设词性 t_i 产生单词 w_i 的概率是 $P(w_i \mid t_i)$，它对应于模型中描述产生输出的概率 b_{ij}。图 7-4 为一个用隐马尔科夫模型描述的简单句子及其词性序列。

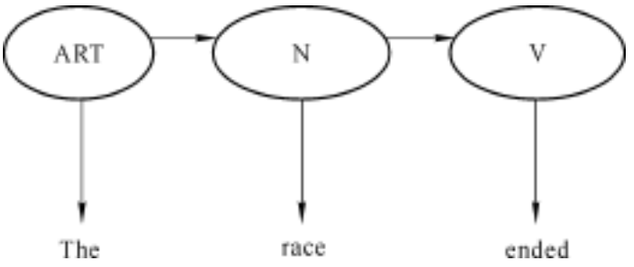


图 7-4　一个句子的隐马尔科夫模型描述

词性标注的任务可以等价于求解下式：

$$\arg\max_T P(T \mid W) \tag{7-4-1}$$

即对于特定的 W，寻找最可能的 T，这即是隐马尔科夫模型的基本问题二。当然，在此之前，需要利用已标注语料获得模型参数，即解决隐马尔科夫模型的基本问题三。

利用隐马尔科夫模型进行词性标注已成为统计方法在自然语言处理中成功运用的范例。

小　　结

本章介绍了马尔科夫模型，主要是隐马尔科夫模型及其在词性标注中的应用。隐马尔科夫模型是建立在马尔科夫模型的基础上的。通过假设一个“隐藏”结构的存在，隐马尔科夫模型能够把简单的词序信息和更高层的语言信息联系起来，以完成诸如词性标注这样的任务。

第八章　概率上下文无关语法

从本章的题目,可以看出其内容与前述上下文无关语法的关系。在第三章,是通过引入特征结构对一般的上下文语法进行了增强,而在本章,则是从另一个角度扩展上下文无关语法。本章是这样安排的:8.1 节介绍概率上下文语法的基本概念;8.2 节介绍概率上下文语法的基本算法;8.3 节对概率上下文无关语法基本假设进行一些讨论;最后是小结。

8.1　概率上下文无关语法的基本概念

为了介绍概率上下文无关语法(PCFG: Probability Content Free Grammar),首先需要对本章要使用的记号做一些说明,如表 8-1 所示。

表 8-1　本章使用的记号说明

记　　号	说　　明
N^i	非终结符号
w^i	终结符号,即词
T	句法分析树
$W_{1m} = w_1 w_2 \cdots w_m$	待分析的句子,w_i 是组成句子的词
$w_{ab} = w_a w_{a+1} \cdots w_b$	句子的一个子串,句子为其特例。因此,在后面 W_{1m}和 w_{1m}可以等价使用
$N^j \quad w_a w_{a+1} \cdots w_b$	N^j 推导出子串 w_{ab},或 N^j 支配子串 w_{ab}
N^j_{ab}	表示 N^j 支配句子中从位置 a 开始到位置 b 结束的子串
$P(W_{1m})$	待分析串 W_{1m}的概率

概率上下文无关语法是上下文无关语法的一种扩展,一个概率上下文无关语法是一个四元组:

$$\text{PCFG } G = (V_N, V_T, N^S, P)$$

其中,V_N 是非终结符号的集合,$V_N = \{N^1, N^2, \cdots, N^i, \cdots, N^n, N^S\}$;$V_T$ 是终结符号的集合,$V_T = \{w_1, w_2, \cdots, w_i, \cdots, w_V\}$;$N^S$ 是语法的开始符号;P 是一组带有概率信息的产生式集合,每条产生式形如$[N^i \to \zeta^j, P(N^i \to \zeta^j)]$,$\zeta^j$ 是终结符号和非终结符号组成的符号串,$P(N^i \to \zeta^j)$ 是产生式的概率,并且有

$$\sum_j P(N^i \rightarrow \zeta^j) = 1$$

例 8-1　下面是一组带有概率信息的产生式规则:

S→NP VP	1.0
NP→NP PP	0.4
PP→P NP	1.0
VP→VP PP	0.3
VP→V NP	0.7
NP→astronomers	0.1
NP→ears	0.18
NP→saw	0.04
P→with	1.0
NP→stars	0.18
NP→telescopes	0.1
V→saw	1.0
ART→a	1.0

例 8-1 中的概率上下文无关语法的规则可以分作两类,一部分规则的右端出现的是某种语言(英语)中的词汇,这些规则可以称为词汇规则;另一部分规则的右端出现的是词汇类别或短语类别符号,常称为语法规则。

如前所述,语法的作用在于帮助进行句子的句法分析,概率上下文无关语法也不例外。依据概率上下文无关语法进行句法分析,首先还是要得到该句子的句法分析树,这与前述利用上下文无关语法分析句子是类似的。但是,为了能使用附带了概率值的规则进行句法分析,需要首先做如下的几个假设。

(1) 位置无关性假设

子结点概率与子结点所管辖的字符串在句子中的位置无关,即

$$\forall k, P(N^j_{k(k+c)} \rightarrow \zeta)$$

相同。

下面的例子说明假设(1)的含义。对于句子:

$$_1\text{A}\ \ _2\text{boy}\ \ _3\text{saw}\ \ _4\text{a}\ \ _5\text{cat}\ . \quad (8\text{-}1\text{-}1)$$

有句法结构树如图 8-1 所示。

在句子(8-1-1)的位置 1,有一个 ART1→a,在位置 4 也有一个 ART2→a,可看成是结点 ART 处在句子的不同位置,但只要它们管辖的结点都是相同的(即用了相同的词汇规则),则

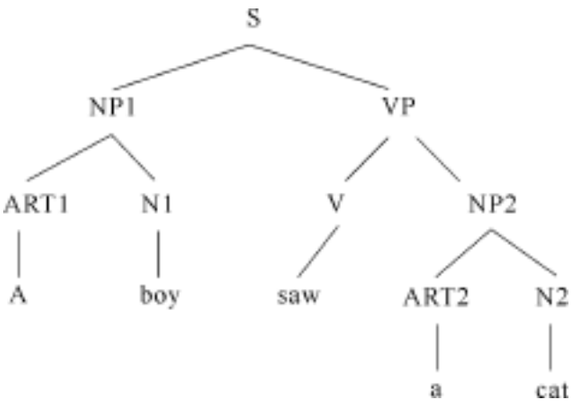


图 8-1　A boy saw a cat 的句法树

每个 ART 结点的概率均相同。这即是说 ART 只与其管辖的词 a 有关,而与该词在句子中所处的位置无关。对于句法树中出现的两个 NP 也是一样的情况,由于两个 NP 均管辖相同的串(ART N),可看成是结点 NP 处在句子的不同位置,它们的概率依据假设(1)也是相同的。

(2) 上下文无关性假设

子结点概率与不受子结点管辖的其他符号串无关,即

$$P(N_{kl}^{j} \rightarrow \zeta \mid \text{从 } k \text{ 到 } l \text{ 之外的其他词}) = P(N_{kl}^{j} \rightarrow \zeta)$$

例如,在句子(8-1-1)中,如果把 saw 换成 bought,ART,NP 等结点的概率值保持不变。该假设是上下文无关假设在概率上的体现,即不仅重写规则是上下文无关的,而且重写规则的概率也是上下文无关的。

(3) 祖先结点无关性假设

子结点概率与导出该结点的所有祖先结点无关,即

$$P(N_{kl}^{j} \rightarrow \zeta \mid \text{结点 } N_{kl}^{j} \text{ 的任何祖先结点}) = P(N_{kl}^{j} \rightarrow \zeta)$$

例如在图 8-1 中,N1 结点的概率就与 S 等祖先结点的概率无关。

有了这三个假设,概率上下文无关语法就不仅继承了语法本身(无附带概率值时)的上下文无关,还使得概率值也能够上下文无关地相同使用。这样就可以利用概率上下文无关语法对句子进行句法分析。首先,利用前述通常的上下文无关语法的句法分析算法,得到句子的句法分析树;然后,为每个结点附带上一个概率值,在上述三个假设下,每个结点的概率值就是对该结点进行进一步重写所使用的规则后面附带的概率。图 8-2 所示为利用概率上下文语法对句子

Astronomers saw stars with ears . (8-1-2)

进行句法分析所得到的句法分析树。由于该句子具有二义,因此图中得到了两棵不同的句法树。

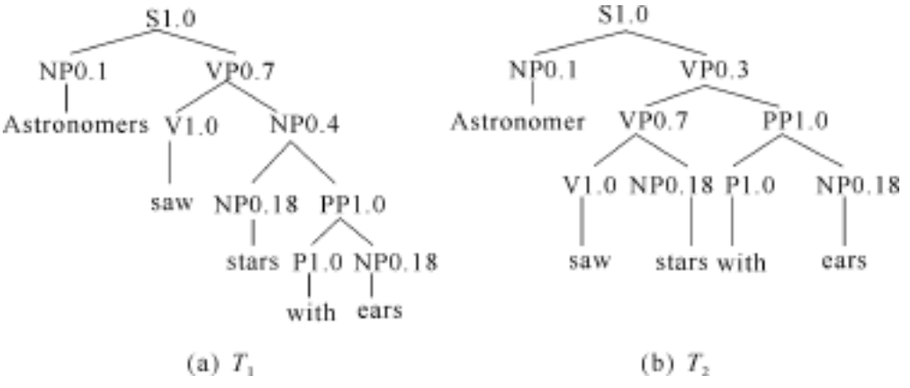


图 8-2 句子 Astronomers saw stars with ears 的两棵分析树

可以看到,与利用上下文无关语法分析句子不同的是,利用概率上下文无关语法得到的句法分析树中每个子树结点上都有一个与之相联系的概率值。加入这个概率值有什么意义呢?一般而言,利用上下文无关语法是希望找出句子的结构,也就是句子的分析树,这棵分析树是进一步分析句子意义的一个基础,从这个角度看,概率上下文无关语法并没有带来任何好处。因为利用概率上下文无关语法能够得到的句子结构,同样可以利用上下文无关语法得到,那么为什么要引入概率上下文无关语法呢?

根据对语言分析,以及句法分析算法的了解,可知自然语言中结构歧义是很严重的,

尤其是分析算法仅仅基于一部上下文无关语法进行时。通常,一个稍长的句子可能会拥有许多棵不同的句法分析树,那么哪一棵分析树是正确的呢?这一直是困扰自然语言句法分析研究的一个关键问题,即句法排歧问题。例如,对句子(8-1-2)的分析得到了图 8-2 中的两棵分析树,那么哪一棵分析树是正确的呢?概率上下文无关语法提供了一条解决问题的途径,即利用附在每条规则后的概率值给每棵分析树计算出一个概率值,利用这个概率值来作为评价分析树的依据,拥有最大概率值的分析树就是最可能的分析树。所以引入概率上下文无关语法的一个最为明显的好处在于句法排歧,因为当一个句子拥有不止一棵分析树时,可以利用分析树的概率值来对所有的分析树进行排序。为此,首先要定义概率上下文无关语法分析下的一棵句法分析树的概率是什么。

概率上下文无关语法分析下的一棵句法分析树的概率通常定义为句法分析树中所用到的产生式规则概率的乘积。

例如,图 8-2 中的两棵句法分析树,可分别计算其概率值为

$$P(T_1) = 1.0 \times 0.1 \times 0.7 \times 1.0 \times 0.4 \times 0.18 \times 1.0 \times 1.0 \times 0.18$$
$$= 0.000\,907\,2$$
$$P(T_2) = 1.0 \times 0.1 \times 0.3 \times 0.7 \times 1.0 \times 0.18 \times 1.0 \times 1.0 \times 0.18$$
$$= 0.000\,680\,4$$

按照上述利用概率值来进行句法排歧的想法,上述例句中的两棵分析树 T_1 的概率是 0.000 907 2,T_2 的概率是 0.000 680 4,因此,T_1 的概率值大,T_1 更可能是正确的分析树。这个结论恰好和我们的语感相符,如果不考虑上下文,通常也倾向于认为分析树 T_1 所表示的句子是正确的。

进而,还可以用下式来计算一个句子的概率:

$$P(W_{1m}) = \sum_{T} P(W_{1m}, T) \tag{8-1-3}$$

其中,求和是对句子 W_{1m} 的所有可能的句法分析树进行。则句子(8-1-2)的概率为

$$P(W_{15}) = P(T_1) + P(T_2) = 0.001\,587\,6$$

可以看到,通过引入概率,概率上下文无关语法的确有助于排除句法歧义。但是,这种排除句法歧义的能力,是通过对一般上下文无关规则附加一个概率值而获得的,这是否意味着概率上下文无关语法是把一般上下文无关语法句法排歧的困难转嫁到规则获取上了呢?至少从表面上看是这样的,因为获得一部概率上下文无关语法意味着要同时获得一部语法和一组概率,而对于一般上下文无关语法则只要获得一部语法,因此似乎前者的难度要比后者大。如果在语法完全由人工总结获得时,情况可能是这样的。但是通常即使是没有概率的语法其编纂也是一个非常艰苦而耗时的工作,当语法规则的数量变得多起来时,规则之间的一致性很难保证。因而目前随着语料语言学的发展,一种在自然语言工程中看来更为可行的办法是通过大规模语料自动学习语法规则,这通常称之为语法归纳。经验表明,一方面,概率上下文无关语法的确比无概率的上下文无关语法更适合语法归纳,实现起来更为简单;另一方面,概率规则还带来了无概率规则所没有的而在自然语言工程中特别需要的柔性处理能力。下面分别来说明这两个方面。

为了阐述第一个方面,需要首先了解一般的上下文无关语法规则是怎样通过语法归纳获得的。通常在这些语法归纳方法中,学习的素材分为两个部分,一个部分是正例训练

集,由正确合法的句子构成;另一个部分是反例训练集,其中的句子是不合法的句子。训练程序通过对比知道什么是合法的结构,什么是错误的结构,进而归纳出语法规则。因而用这样的方法学习语法规则,必须同时准备正例训练集和反例训练集。

问题是可以假定语料库中所有句子都是合法的句子,把它们作为正例训练集,然而,反例训练集却不易得到。另外这种需要反例训练集的学习方法似乎也没有很强烈的认知基础。儿童学习语言往往无需这样的反例训练集,父母们很少指出儿童的哪句话不合乎语法。在引入概率上下文无关语法后,学习问题就是如何得到一部带有概率的语法,使得正例训练集中的句子的概率最大,因此无需反例训练集。

另一方面,无概率的上下文无关规则意味着所建立的语法规则应该是永远成立的,而这种结论是难以从对有限语料的学习归纳中获得。因为语言的创造性,即使有再大的学习语料,也不能保证某个语法规则在后面的使用中没有例外,总会有某个新句子的语法描述会超出已确定的语法系统的规定。而通过增加概率,一个规则只保证以某个概率成立,只要样本充分大,这个概率就会较准确。

由于概率的存在,可以容许某些“不合法”的句子存在,这就为语言分析带来了柔性。没有概率的规则只带来合法与不合法两个选择。如果作为不合法的句子简单拒绝,这在遇到真实语料时经常是寸步难行的,实践证明,这会使得语言处理无法真正走向实用。而如果把这些句子作为合法的句子加以接受,则本身已经使得规则失去了意义。

而引入概率可以使上述两难境地得到一定程度的缓和。语法可以被分作两个部分,一个部分接受那些通常被认为是合法的句子,并给解释这些合法句子的规则以较高的概率值;另一部分则是用来解释“不合法”的句子,但这些规则的概率相对而言都比较小。从而语法不但能处理合法的句子,也可以处理“不合法”的句子,因为“不合法”句子的概率较小,语法仍然可以有效地区分合法与不合法。这样,虽然概率上下文无关语法并没有解决人们为什么会使用“不合法”句子的问题,但是带来了传统语法所不具有的柔性处理能力,也就是具有一定的容错能力。这种能力,对于语言处理系统走向实用是非常关键的。

因此,与表面上看来学习一部概率上下文无关语法意味着要同时学习一部语法和一组概率,学习的负担要大于仅仅学习一部上下文无关语法相反,本质上,概率的引入把严格的归纳学习转化为具有统计意义的归纳学习,反而降低了语法归纳的难度。

尽管如此,语法归纳问题仍是一个比较困难的问题,基于概率上下文无关语法的语法学习也存在很多需要解决的问题,典型的问题是学习过程的收敛性。有关基于概率上下文无关语法的语法学习表现出,在学习开始时,如果参数选择不同,学习结果会完全不同。

8.2 概率上下文无关语法的基本算法

运用概率上下文无关语法,主要是解决三个基本问题。

(1) 给定一部概率上下文无关语法G,如何计算句子 W_{1m}的概率?即计算 $P(W_{1m}|G)$的问题。

(2) 给定一部概率上下文无关语法 G 以及句子 W_{1m},最为可能的分析树是什么,即计算$\arg\max_T P(T|W_{1m},G)$的问题。

(3) 如何为语法规则选择概率,使得训练句子的概率最大?即计算$\arg\max_G P(W_{1m}|G)$的问题。

为了便于说明问题,以下讨论中,上下文无关语法均限制为乔姆斯基范式,即语法规则具有下面的形式:

$$N^i \to N^j N^k$$

$$N^i \to w_j$$

即产生式右端或者只有两个非终结符号,或者只有一个终结符号。可以证明任何一部上下文无关语法都有其等价的乔姆斯基范式。

为了解答上述三个问题,需要首先定义向外变量和向内变量。

向外变量

$$\alpha_j(p,q) = P(w_{1(p-1)}, N^j_{pq}, w_{(q+1)m}|G)$$

向内变量

$$\beta_j(p,q) = P(w_{pq}|N^j_{pq}, G)$$

向内变量$\beta_j(p,q)$为非终结符号N^j推导出词串$w_{pq} = w_p w_{p+1}\cdots w_q$的概率;向外变量$\alpha_j(p,q)$则为由语法的开始符号$N^s$推导出句子$w_1 w_2\cdots w_{p-1} N^j_{pq} w_{q+1}\cdots w_m$的概率。向内变量和向外变量的含义可以参看图8-3。

1.解决第一个基本问题

计算一个句子的概率,最为直接的方法是依据前面给出的计算句子概率的定义式(8-1-3),通过计算出这个句子所有可能的分析树的概率,然后对它们求和。然而这是一个效率很低的办法,因为当规则数量较多时,一个句子往往有很多分析树,因此需要考虑利用其他的方法,由于有

$$P(W_{1m}|G) = P(N^s \quad W_{1m}|G) = P(W_{1m}|N^l_{1m}, G) = \beta_l(1,m)$$

则可以通过下面的向内算法,利用向内变量归纳计算出一个子串的概率。

(1) 初始化

$$\beta_j(k,k) = P(w^k|N^j_{kk}, G) = P(N^j \to w^k|G)$$

(2) 归纳计算$\beta_j(p,q)$,其中$p<q$

因为限制语法为乔姆斯基范式,因此第一条使用的产生式必为

$$N^j \to N^u N^v$$

子串w_{pq}一定在某个位置d被分成两个部分,使得N^u支配子串w_{pd},而N^v支配子串$w_{(d+1)q}$,如图8-4所示。

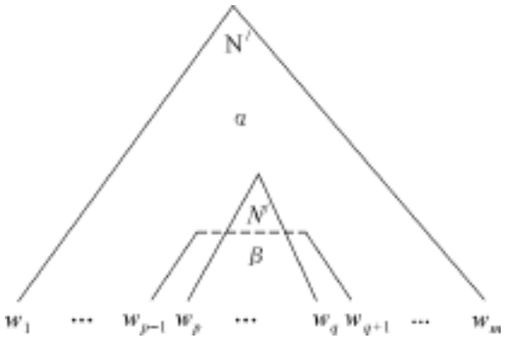


图8-3 向内变量和向外变量

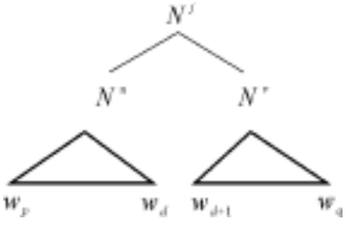


图8-4 向内算法图解

因此,对所有 $j, 1 \leq p < q \leq m$,则有

$$\begin{aligned}
\beta_j(p,q) &= P(w_{pq} \mid N^j_{pq}, G) \\
&= \sum_{u,v} \sum_{d=p}^{q-1} P(w_{pd}, N^u_{pd}, w_{(d+1)q}, N^v_{(d+1)q} \mid N^j_{pq}, G) \\
&= \sum_{u,v} \sum_{d=p}^{q-1} P(N^u_{pd}, N^v_{(d+1)q} \mid N^j_{pq}, G) \cdot P(w_{pd} \mid N^u_{pd}, N^v_{(d+1)q}, N^j_{pq}, G) \\
&\quad \cdot P(w_{(d+1)q} \mid N^u_{pd}, N^v_{(d+1)q}, N^j_{pq}, w_{pd}, G) \\
&= \sum_{u,v} \sum_{d=p}^{q-1} P(N^u_{pd}, N^v_{(d+1)q} \mid N^j_{pq}, G) \cdot P(w_{pd} \mid N^u_{pd}, G) \\
&\quad \cdot P(w_{(d+1)q} \mid N^v_{(d+1)q}, G) \\
&= \sum_{u,v} \sum_{d=p}^{q-1} P(N^j \rightarrow N^u N^v) \beta_u(p,d) \beta_v(d+1, q)
\end{aligned}$$

上述推导中使用了乘法公式以及概率上下文无关语法的独立性假设。利用这个公式,就可以以一种自底向上的方法归纳计算出句子的概率 $\beta_1(1, m)$。

例 8-2　利用向内算法,计算句子(8-1-2)的概率。

解: 计算过程可以用表 8-2 所示的上三角矩阵来表示,单元格(p, q)内为向内概率$\beta_i(p, q)$,空白单元格表示相应的向内概率为 0,计算顺序为沿对角线方向,然后逐步移向右上角。

表 8-2　利用向内算法,计算句子(8-1-2)的概率

	1	2	3	4	5
1	$\beta_{NP} = 0.1$		$\beta_S = 0.01$		$\beta_S = 0.001$
2		$\beta_{NP} = 0.04$ $\beta_V = 1.0$	$\beta_{VP} = 0.126$		$\beta_{VP} = 0.015\,876$
3			$\beta_{NP} = 0.18$		$\beta_{NP} = 0.012\,96$
4				$\beta_P = 1$	$\beta_{PP} = 0.18$
5					$\beta_{NP} = 0.18$
	Astronomers	Saw	stars	with	ears

也可以利用向外变量计算句子的概率,相应的算法称为向外算法。首先注意到,对于 $1 \leq k \leq m$,有

$$\begin{aligned}
P(W_{1m} \mid G) &= \sum_j P(w_{1(k-1)}, w_k, w_{(k+1)m}, N^j_{kk} \mid G) \\
&= \sum_j P(w_{1(k-1)}, N^j_{kk}, w_{(k+1)m} \mid G) \cdot P(w_k \mid P(w_{1(k-1)}, N^j_{kk}, w_{(k+1)m}, G) \\
&= \sum_j \alpha_j(k,k) P(N^j \rightarrow w_k)
\end{aligned}$$

则向外算法可以描述如下。

(1) 初始化

$$\alpha_1(1, m) = 1$$

$$\alpha_j(1, m) = 0 \quad (\text{当 } j \neq 1 \text{ 时})$$

(2) 归纳计算 $\alpha_j(p, q)$

因为仅仅考虑乔姆斯基范式的情形，在这种情形下，N^j_{pq} 可能是某个父结点的左子女，也可能是某个父结点的右子女，算法要考虑这两种情形。对这两种情形的求和作为向外概率值。这两种情形可以分别用图 8-5 和 8-6 来表示。

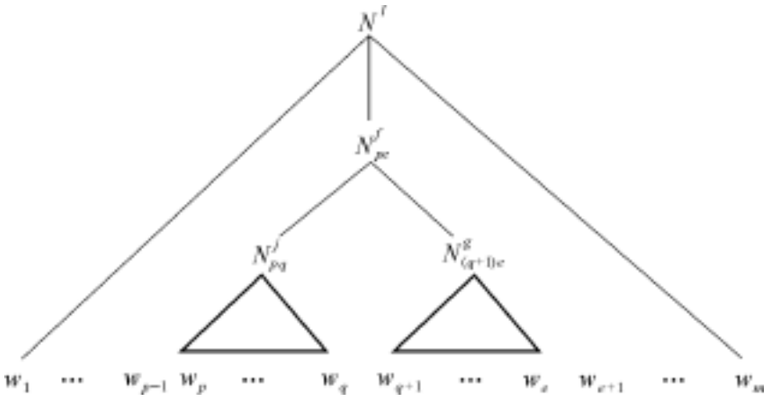


图 8-5　N^j_{pq} 是父结点的左子女

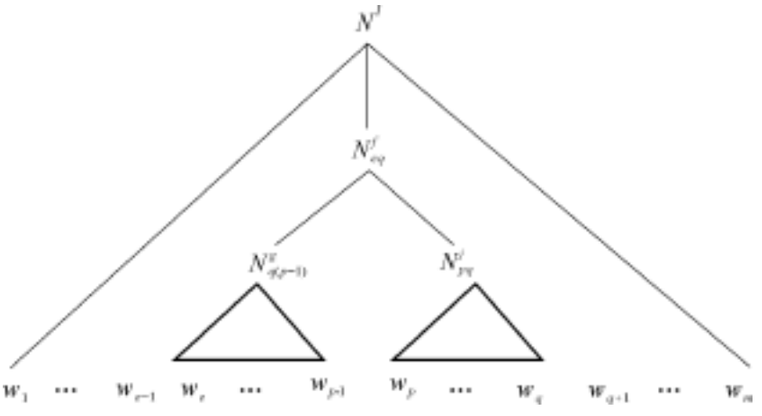


图 8-6　N^j_{pq} 是父结点的右子女

$$
\begin{aligned}
&\alpha_j(p, q)\\
&= \Big[\sum_{f, g \neq j} \sum_{e=q+1}^{m} P(w_{1(p-1)}, w_{(q+1)m}, N^f_{pe}, N^j_{pq}, N^g_{(q+1)e}) \Big]\\
&\quad + \Big[\sum_{f, g} \sum_{e=1}^{p-1} P(w_{1(p-1)}, w_{(q+1)m}, N^f_{eq}, N^g_{q(p-1)}, N^j_{pq}) \Big]\\
&= \Big[\sum_{f, g \neq j} \sum_{e=q+1}^{m} P(w_{1(p-1)}, w_{(e+1)m}, N^f_{pe}) P(N^j_{pq}, N^g_{(q+1)e} \mid N^f_{pe}) P(w_{(q+1)e} \mid N^g_{(q+1)e}) \Big]\\
&\quad + \Big[\sum_{f, g} \sum_{e=1}^{p-1} P(w_{1(e-1)}, w_{(q+1)m}, N^f_{eq}) P(N^g_{e(p-1)}, N^j_{pq} \mid N^f_{eq}) P(w_{e(p-1)} \mid N^g_{e(p-1)}) \Big]\\
&= \Big[\sum_{f, g \neq j} \sum_{e=q+1}^{m} \alpha_f(p, e) P(N^f \to N^j N^g) \beta_g(q+1, e) \Big]\\
&\quad + \Big[\sum_{f, g} \sum_{e=1}^{p-1} \alpha_f(e, q) P(N^f \to N^g N^j) \beta_g(e, p-1) \Big]
\end{aligned}
$$

为了防止当 $X \to N^j N^j$ 时引起重复计算，故上述求和中第一部分限制 $g \neq j$。同时，还应当注意到和向内变量不同，递归计算向外变量时，要用到向内变量。

2．解决第二个基本问题

概率上下文无关语法的第二个基本问题是在给定语法 G 和句子 W_{1m} 的前提下，如何

有效找出最为可能的分析树。这可以通过 Viterbi 算法求得。为了介绍该算法,首先引入变量 $\delta_i(p,q)$,定义该变量为子树 N^j_{pq} 的最大概率,即所有 N^i 支配 w_{pq} 的子树中概率最大的子树,则 Viterbi 算法可描述如下。

(1) 初始化

$$\delta_i(p,q) = P(N^i \to w^p)$$

(2) 归纳计算

$$\delta_i(p,q) = \max_{\substack{1\le j,k\le n \\ p\le r<q}} P(N^i \to N^j N^k)\,\delta_j(p,r)\,\delta_k(r+1,q)$$

$$\Psi_i(p,q) = \operatorname*{argmax}_{(j,k,r)} P(N^i \to N^j N^k)\,\delta_j(p,r)\,\delta_k(r+1,q)$$

(3) 归纳终止

$$P(\hat{T}) = \delta_1(1,m)$$

(4) 按照下面的步骤构造最为可能的分析树,即 $\hat{T}$。

① $\hat{T}$ 的根结点为 N^s_{1m},因为 N^s 是语法的开始符号。

② 若 N^i_{pq} 是 $\hat{T}$ 的一个内部结点且 $\Psi_i(p,q)=(j,k,r)$,则 N^i_{pq} 的左儿子结点是 N^j_{pr},N^i_{pq} 的右儿子结点是 $N^k_{(r+1)q}$。

3.解决第三个基本问题

概率上下文无关语法的最后一个问题是训练问题。假定事先设定了语法的终结符号、非终结符号以及语法规则的集合,训练的过程即为为每个规则赋概率的过程。

对于一个特定规则 $N^j \to \zeta$ 而言,其概率可以通过下面的公式计算:

$$\hat{P}(N^j \to \zeta) = \frac{C(N^j \to \zeta)}{\sum_{\gamma} C(N^j \to \gamma)}$$

这里,$C(\cdot)$表示某条规则使用的次数。

如果事先有一个足够大的已经分析过的语料库,问题比较容易解决,可以直接统计计算。根据最大似然估计的原则,一个好的语法(参数),应当是使得训练数据取最大概率的语法(参数)。

然而在很多情况下,并不存在大规模已经分析过的训练语料库。这里使用一种 EM (Expectation Maximize)算法,其基本思想是根据最初的一组参数,反复迭代求精,求出一组局部最优参数。由于算法同时需要使用向前变量和向后变量,故一般称为向前向后算法。

根据向前向后变量的定义有

$$\begin{aligned}\alpha_j(p,q)\beta_j(p,q) &= P(N^l \ w_{1m}, N^j \ w_{pq} \mid G)\\ &= P(N^l \ w_{1m} \mid G)\,P(N^j \ w_{pq} \mid N^l \ w_{1m}, G)\end{aligned}$$

令 $\pi = P(N^l \ w_{1m})$,则

$$P(N^j \ w_{pq} \mid N^l \ w_{1m}, G) = \frac{\alpha_j(p,q)\beta_j(p,q)}{\pi} \qquad (8\text{-}2\text{-}1)$$

N^j 在推导过程中使用次数的期望估计为

$$\sum_{p=1}^{m}\sum_{q=p}^{m}\frac{\alpha_j(p,q)\beta_j(p,q)}{\pi}$$

根据式(8-2-1)以及向内变量的迭代计算公式,可以得到

$$P(N^j \to N^r N^s \; w_{pq} \mid N^l \; w_{1m}, G) = \frac{\sum_{d=p}^{q-1} \alpha_j(p,q) P(N^j \to N^r N^s) \beta_r(p,d) \beta_s(d+1,q)}{\pi}$$

则规则 $N^j \to N^r N^s$使用次数的期望为

$$\frac{\sum_{p=1}^{m-1} \sum_{q=p+1}^{m} \sum_{d=p}^{q-1} \alpha_j(p,q) P(N^j \to N^r N^s) \beta_r(p,d) \beta_s(d+1,q)}{\pi}$$

规则 $N^j \to N^r N^s$的概率可以估计为

$$P(N^j \to N^r N^s) = \frac{N^j \to N^r N^s \text{使用次数的期望}}{N^j \text{使用次数的期望}}$$

即

$$\hat{P}(N^j \to N^r N^s) = \frac{\sum_{p=1}^{m-1} \sum_{q=p+1}^{m} \sum_{d=p}^{q-1} \alpha_j(p,q) P(N^j \to N^r N^s) \beta_r(p,d) \beta_s(d+1,q)}{\sum_{p=1}^{m} \sum_{q=p}^{m} \alpha_j(p,q) \beta_j(p,q)}$$

同样,对于 $N^j \to w^k$类的规则,有

$$P(N^j \to w^k \mid N^l \; w_{1m}, G) = \frac{\sum_{h=1}^{m} \alpha_j(h,h) P(N^j \to w^k, w^h = w^k)}{\pi}$$

$$= \frac{\sum_{h=1}^{m} \alpha_j(h,h) P(w^h = w^k) \beta_j(h,h)}{\pi}$$

则

$$\hat{P}(N^j \to w^k) = \frac{\sum_{h=1}^{m} \alpha_j(h,h) P(w^h = w^k) \beta_j(h,h)}{\sum_{p=1}^{m} \sum_{q=p}^{m} \alpha_j(p,q) \beta_j(p,q)}$$

以上为只有一个训练句子的情况,如果训练集有多个训练句子,即

$$W = (W_1, W_2, \cdots, W_\omega)$$

其中,

$$W_i = w_{i,1} w_{i,2} \cdots w_{i,m}$$

令

$$f_i(p,q,j,r,s) = \frac{\sum_{d=p}^{q-1} \alpha_j(p,q) P(N^j \to N^r N^s) \beta_r(p,d) \beta_s(d+1,q)}{P(N^1 \; w^i \mid G)} \quad (8\text{-}2\text{-}2)$$

$$g_i(h,j,k) = \frac{\alpha_j(h,h) P(w^h = w^k) \beta_j(h,h)}{P(N^l \; w^i \mid G)} \quad (8\text{-}2\text{-}3)$$

$$h_i(p,q,j) = \frac{\alpha_j(p,q) \beta_j(p,q)}{P(N^l \; w^i \mid G)} \quad (8\text{-}2\text{-}4)$$

则

$$\hat{P}(N^j \rightarrow N^r N^s) = \frac{\sum_{i=1}^{\omega} \sum_{p=1}^{m_i - 1} \sum_{q=p+1}^{m_i} f_i(p, q, j, r, s)}{\sum_{i=1}^{\omega} \sum_{p=1}^{m_i} \sum_{q=p}^{m_i} h_i(p, q, j)} \tag{8-2-5}$$

$$\hat{P}(N^j \rightarrow w^k) = \frac{\sum_{i=1}^{\omega} \sum_{h=1}^{m_i} g_i(h, j, k)}{\sum_{i=1}^{\omega} \sum_{p=1}^{m_i} \sum_{q=p}^{m_i} h_i(p, q, j)} \tag{8-2-6}$$

一般,首先可以给规则任意确定一组概率,使得 $\sum_j P(N^i \rightarrow \zeta^j) = 1$; 然后利用训练集,计算上述式(8-2-2)、(8-2-3)和(8-2-4),进而利用式(8-2-5)和(8-2-6)计算得到一组更新的参数。如此继续,直到参数收敛于一个局部最优点。

从有关文献来看,尽管原则上可以从一个没有分析过的语料库用向内向外算法学习概率上下文无关语法,但实际上并非易事。向内向外算法仅仅保证收敛于一个局部最优点,而且对初始选定的参数初值非常敏感。例如 Charniak 进行过一组学习实验,对于同样的训练集合,每次采用不同的初始参数,进行了 300 次训练,结果得到了 300 个不同的结果。

8.3 概率上下文概率语法基本假设的问题

虽然如前所述,概率上下文语法通过对规则分配概率具有了句法排歧的能力,但是需要指出的是,已有的利用概率上下文概率语法进行句法排歧的实验表明,概率上下文无关语法由于其本身的限制,其在句法排歧方面的能力是相当有限的,有时候甚至是适得其反。下面从两个方面看,一个方面是与 n 元语法的比较;另一个方面是概率上下文语法本身用于不同句子的句法分析。

首先把概率上下文语法与 n 元语法做一个对比。像 n 元语法那样,概率上下文无关语法也为语言提供了一种概率模型,在这种模型中,语言由各个句子组成,语言的概率即为各个句子概率的乘积。然而,它和 n 元语法的侧重点不同,n 元语法侧重在词汇层,而概率上下文无关语法主要考虑句子的结构因素。

有时候用概率上下文无关语法能比 n 元语法更好地描述语言现象。例如对于词串:

I boy a am

利用概率上下文无关语法能判定它作为合法句子的概率很小,但是用 n 元语法就不能得到这个结论。因为 2 元语法和 3 元语法作为语言模型完全忽略了语言的结构因素。又比如对于下面的英文句子:

Fred watered his mother 's small garden .

3 元模型可能不能给出一个好的解释,因为概率 $P(\text{garden} \mid \text{mother 's small})$ 的值较小,但是如果认识到 garden 是短语 his mother 's small garden 的中心词(句子的结构信息),而利

用概率 $P(X = \text{garden} \mid X$是 water 宾语的中心词)来进行计算,结果会更为合理。并且这也会带来另外一个 3 元语法所不具有的好处,就是参数规模会得到控制。3 元语法要考虑所有可能的词汇共现关系,不管在这些共现中是否存在句法关系,因而参数规模庞大。如果仅考虑具有合法关系的词汇间的共现关系,参数规模就会大大缩小。

但是,在另一些语言现象的解释方面,利用概率上下文无关语法解释又不如利用 n 元语法。以英语词串"the green banana"和"the green time"为例,一个 3 元语法可以给词串"the green banana"赋以较高的概率(因为这 3 个词共现的可能性很大),而给词串"the green time"赋以较低的概率(因为 green 和 time 的共现可能性不大),这个解释非常符合我们的认识。然而利用概率上下文无关语法,情形就不同了。下面分别写出两个词串的句法分析树,如图 8-7 所示。

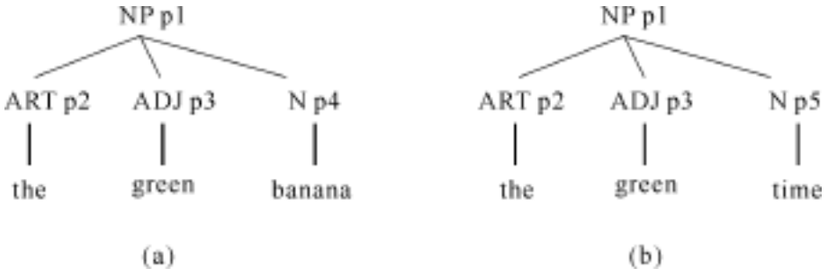


图 8-7　两个词串的句法树

它们之间的差别在于 banana 和 time 实现为名词时概率的差别(p5 和 p4 不同)。由于 time 作为名词出现的频次远远比 banana 要高,因此在相应于它们的 N 结点上所标的概率值 time 的要比 banana 的大(p5 > p4),除此之外均相同。这样,比较两个句法树的概率,可知词串"the green time"比"the green banana"要具可能性。显然这个结论非常不符合我们的认识。

其次,来看概率上下文无关语法对不同句子的分析。把句子 (8-1-2) 换成如下的句子:

Astronomers saw stars with telescopes .

利用概率上下文无关语法可以得出图 8-8 的两棵句法树。

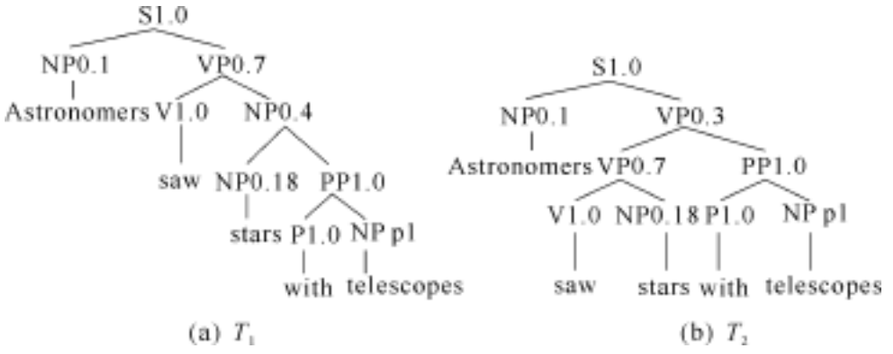


图 8-8　句子 Astronomers saw stars with telescopes 的两棵分析树

可以看到,图 8-8 中的两棵树与图 8-2 中的两棵树的结构分别是完全相同的。除了把 ears 换成 telescopes 以外,所有结点名称没有任何变化;每个结点的概率除了指向 telescopes 的 NP 结点概率均换成另一个相同值(设为 p1)之外也完全不变,于是结论也是不变的,即

第一棵树具有较大的概率,这就得出了和常识不符的结果。

出现上述这两个方面情况的原因,可直接归结到上述三个假设,尤其是概率的上下文无关性假设和祖先结点无关性假设。在第一种情况下,time 重写 N 的概率与出现在其周围的词是什么无关,而总是取其作为名词的概率;在第二种情况下,telescopes 和 stars 与 saw 在搭配上的差别也没有影响其他各个结点的概率。

而实际上,无论上述何种情况,在确定哪一棵分析树是正确的这个问题上,具体的词汇信息都对句子结构起着重要作用。

因此,至少可以期望,如果能把主要考虑句子结构的概率上下文无关语法在某种程度上词汇化,一定会产生一个更好的语言模型。这是对概率上下文无关语法进行改进的一个重要方向,有兴趣的读者可以参阅有关文献。

小　　结

本章介绍了概率上下文无关语法,对于基本的概率上下文无关语法,除了规则本身是上下文无关的,很重要的基本假设是规则的概率也是上下文无关的,这与规则的上下文无关假设一样带来两个完全不同方面的影响,一方面使得许多问题得以简化,但另一方面也给概率上下文语法带来很大的局限性。利用概率上下文无关语法进行句法分析要解决三个基本问题,前两个问题是直接面向解决句法分析的;而第三个问题是关于如何获得规则的概率,这是概率语法归纳的一部分,目前还没有很好的解决办法。

第九章　机 器 翻 译

机器翻译是相对于人的翻译而言的,其主要目的是试图利用计算机把一种自然语言(称为“源语言”)翻译为另外一种自然语言(称为“目标语言”)。机器翻译依据语言传播媒介的不同而可以分为文本机器翻译和语音机器翻译。

制造一种机器,让使用不同语言的人能够互相自由交流,一直以来是人类的一个梦想。随着国际互联网络的日益普及,在今天,人们已经比以往任何时候更容易获得各种信息;语言障碍的问题在新的时代又一次凸显出来,人们比以往任何时候都更迫切需要语言的自动翻译系统。然而机器翻译却是一个极为困难的研究课题,无论目前对它的需求多么迫切,现在全自动高质量的机器翻译系统仍未出现。

一个机器翻译系统的研制几乎可以覆盖自然语言处理技术的各个方面的内容,因此,本书把机器翻译作为自然语言处理技术的应用部分的唯一内容来介绍。

本章是这样安排的:9.1节概述,简单介绍目前机器翻译系统所采用的翻译策略;9.2节介绍一些基于规则翻译系统的基本知识和翻译流程,这里没有把焦点集中于具体的翻译系统,而是希望对基于规则的方法有一个总体的介绍;9.3节将对经验主义的机器翻译进行介绍,包括基于统计的机器翻译和基于实例的机器翻译;9.4节介绍对统计的机器翻译和基于实例的机器翻译都十分关键的一个技术——双语对齐;最后介绍当前机器翻译的应用需求和状况。

9.1　机器翻译概述

9.1.1　机器翻译的基本方法

经过不懈努力,学术界为解决机器翻译问题已经提出了种种策略和方法,尽管目前没有任何一种方法能实现机器翻译的完美理想,但这些在方法论方面的探索已经使得人们对机器翻译问题的认识更加深刻,而且这些方法也确实带动了不少不那么完美但尚可使用的产品问世。

从系统所采用的技术来看,机器翻译方法可以分为基于规则的方法和基于语料库的方法,因为在20世纪90年代以前,机器翻译方法的主流一直是基于规则的方法,所以今天的人们往往称之为传统的机器翻译方法。传统的机器翻译从总体模式上可以分为三类,直接翻译法、中间语言法以及转换法。

1．直接翻译法

直接翻译法有时也称为逐词翻译法，采用这种方法，一般总是针对某一个特定的语言对。这种方法实质上认为计算机不需要对源语言进行太多的分析就可以获得译文。目标语言可以通过对源语言逐词进行翻译获得。五六十年代的许多系统都是按这种方法设计的，这些系统按它们所结合的分析多少而有所不同，有的几乎没有对源语言进行任何的分析，因而也没有任何目标语言的重构工作；有的对源语言进行了较浅的分析，相应地也就有一部分目标语言的重构工作。这种方法对翻译过程的认识显然过于简单化，基本上属于一种过时的认识，现在已很少采用这种办法，原先按照这种模式设计的系统也逐渐改变了策略。

2．中间语言法

中间语言法认为，把源语言经过分析转换成一种对所有语言都适合的一种句法-语义表示是可能的，从这种表示可以生成任何一种目标语言，这种中间表示称为中间语言。在中间语言系统中，从源语言到目标语言的翻译过程经过两个完全独立的阶段。在第一个阶段中，源语言被完全分析转换成中间语言；而第二个阶段则根据中间语言生成目标语言。源语言分析只面向特定的源语言而不考虑任何目标语言；同样，目标语言生成只面向特定的目标语言而不考虑任何源语言，如图 9-1 所示。

图 9-1　中间语言系统

不同的中间语言系统可能采用不同的中间语言，有的是一种逻辑形式的语言；有的是一种类似自然语言的人工语言，例如荷兰的 DLT 计划采用世界语 Esperanto 作为中间语言。

中间语言法在进行多种语言互译时是非常有效的，它能把 $n(n-1)$个直接翻译模块减少为 $2n$ 个翻译模块。以 4 种语言之间的互译为例，采用中间语言方法只需要设计每种语言到中间语言的翻译模块，因而仅需要 8 个翻译模块；而采用直接翻译方法或其他翻译方法，则必须为任何两种语言之间建立互译模块，因而需要 12 个翻译模块，如图 9-2 所示(图中每一个箭头可以理解为一个单向翻译模块)。所以在设计多种语言互译的机器翻译系统时，这种方法在理论上是非常经济的。

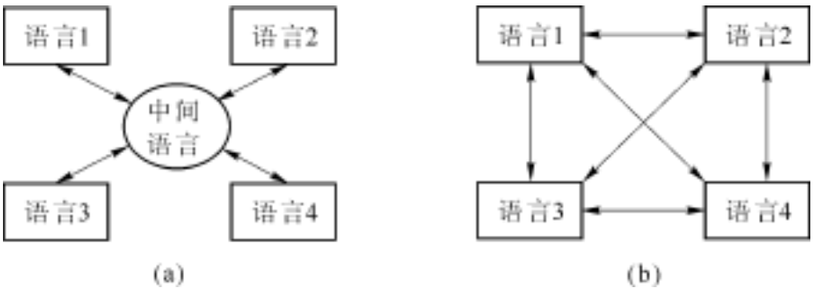


图 9-2　4 种语言之间互译

3．转换法

与中间语言法不同，转换法不是采用一种中间表示按两个阶段进行翻译，而是采用两

种内部表达并按三个阶段进行翻译,第一个阶段把源语言转换成源语言的内部表达;第二个阶段把源语言的内部表达转换成目标语言的内部表达;第三个阶段再根据目标语言的内部表达生成目标语言,如图 9-3 所示。

图 9-3　转换式系统

不同的转换系统按它们分析的深度和内部表达的抽象程度而有所不同,早期的系统分析较浅,分析结果只是一种表层的句法结构,转换就在这个层次上进行;现在的系统一般进行较深层次的分析,分析结果是一种句法-语义结构,相应的转换表达和转换规则也就比较抽象。值得指出的是,当今许多商品化系统都采用了这样的总体结构。这种方法目前仍然是最为成功的方法。

由上可以看出,所有的翻译方法都要对源语言进行分析,这种分析使得寻找和源语言表达等价的目标语言表达更为容易,所以从本质上讲,形形色色的机器翻译系统之所以不同,主要是各个系统对翻译所需要的分析(或理解)深度有不同的意见。直接翻译法认为,不需要深层次的源语言分析,在源语言句法结构未知的前提下就可以翻译;转换法认为,要进行翻译,就应该首先得到源语言的句法结构;而中间语言法则强调需要更为彻底的源语言分析。从预想的翻译结果来看,似乎是对源语言理解得越深刻,翻译结果应该越好,但实际情况似乎并不如此,分析深度增加必然同时伴随系统复杂性以及开发代价的增大,同时深层次的理解还受着当前分析技术的制约。因而,开发实用系统必须对系统复杂性和翻译质量进行平衡。如只希望得到能供译后编辑的译文,可以用相对较浅的源语言理解。如果要追求更高的译文质量,则不仅要进行句法分析,还要进行语义解释。

20 世纪 70 年代,为了摆脱机器翻译困境,受人工智能、知识工程发展的影响,人们提出了基于知识的机器翻译方法。基于知识的机器翻译方法强调成功翻译的前提是对源语言意义更为彻底的理解。从总体结构上讲,基于知识的系统属于中间语言系统,它和一般中间语言系统相比,主要区别在于基于知识的系统加入了关于世界知识的显式处理。系统不仅要包含各种语言知识资源,还需要建造对理解有益的各种本体知识库。

本质上,利用上述传统的方法建立机器翻译系统,都需要建立各类知识库,描述源语言和目标语言的词法、句法以及语义知识,甚至也要描述和语言知识无关的世界知识。然而这些知识库的描述和建立是极为困难的,为此一些研究人员致力于探索避开这些困难的机器翻译方法,从 80 年代中后期开始出现了基于语料库的机器翻译方法。这种反传统的方法排斥对语言进行深层次的分析,试图通过大规模收集互为译文的双语语料并基于这些语料进行双语翻译。在这种模式下,有两个方法分支比较引人注目,一种称为基于实例的机器翻译方法,认为可以通过在双语语料库中查找最为相似的翻译实例的方法来获得语言的翻译;另一分支称为基于统计的机器翻译,主张通过对大规模的双语语料进行统计,翻译可以基于双语之间的复杂共现和分布概率计算而获得。

考虑到这些方法背后的哲学背景,一些文献也把基于规则的机器翻译方法称为理性主义方法,而把基于语料库的翻译方法称为经验主义方法。由于基于知识的机器翻译方

法依靠人工智能为背景,因此有时也称为基于人工智能的机器翻译方法。

9.1.2 困难和对策

机器翻译的困难主要是由语言的歧义造成的。歧义现象是自然语言的显著特点,机器翻译不仅要研究一种语言内部已经相当棘手的歧义问题,而且还要考虑不同语种之间的复杂歧义现象。

语言单位无论从小到大都存在歧义,并且在语法、语义、语用每个层面上都有表现。

词汇一级,从句法层面上说,单词可能是兼类的,例如:英语中单词 work 可能是名词,也可能是动词;从意义层面上说,许多词汇是多义的,例如:英语单词 bank 可能表示"银行",也可能表示"河岸"。

短语结构一级,三个名词组合 NNN,既可以是(NN)N,如:cat food tin,也可以是 N(NN),例如:toy coffee grinder。

句子一级,著名的例子"The boy saw a girl with a telescope"就有两种合法的句法结构。从意义上说,如汉语"他刚刚做过外科手术"中,"他"是"大夫"还是"病人"就存在歧义。

语言之间的不同更是举不胜举,词汇一级,表达同一概念,有的语言笼统,有的则比较具体。例如各语种中的亲属称谓问题,汉语中要明确区分性别、大小、母系和父系;但另外一些语言则不做区分,如英语对"兄弟姐妹"不区分大小、对一些"社会关系"不区分母系还是父系,这样的话,要把英语中的亲属称谓翻译为正确的汉语表达,没有有效的上下文处理手段是非常困难的。不同语言的词汇在意义上也并非一一对应,而大多相互交叉重叠。

在句子一级,有的是 SVO 结构的语言,有的是 SOV 结构的语言,很难找出在表达上完全等价的结构。例如汉语中"述补结构"就很难对应英语中的某种结构。

研究人员已提出了一系列技术来解决这些歧义问题,虽然取得了一些进展,但歧义问题还远远没有从根本上得到解决。下面是目前一些机器翻译系统采用的主要技术和策略。

1.在限定的领域内进行翻译

这种方法一般也称为"子语言"法。这种方法不追求系统能在所有领域获得高质量译文,而只希望在翻译某一狭窄的专业领域的文本时获得高质量的译文。实际上,当今许多机器翻译系统都属于这一类型,这类系统的词典和规则无需覆盖本领域之外的语言现象。

TAUM-METEO 是这类系统中最为成功的一个例子。据报道,加拿大蒙特利尔大学开发的这个全自动机器翻译系统在把气象预报信息由英文译成法文时达到了百分之九十以上的准确率。该系统大约含 1 500 个词汇,其中大约半数是地名,涉及到的句法结构也只是英语句法结构中一个很小的子集,并且,很多具有多个义项的一般词汇在气象领域通常会有一个确定的意义,因此系统只需解决很少的歧义,准确率较高。

2.利用受限语言作为输入

这种方法一般称为"受控语言"法或"受限语言"法。这种方法通过在词汇、句法结构方面加以限制,以力图避免机译系统难以处理的语言现象。这种方法要求,交付系统翻译的文本必须遵从受控语言的规定,因而,翻译不满足受控语言规定的文本,事先要经过熟悉受限语言知识的人员改写。例如著名的跨国公司施乐公司,曾经制定了一种受限的英语——"多国规范英语",文档书写必须遵照这个受限语言的规定,然后进一步交付 SYS-

TRAN 系统(SYSTRAN 系统是在乔治顿大学开发的系统的基础上发展起来的一个商用系统)翻译为其他语种。受限语言一般对词汇、句子结构类型以及句子的长度都有一些严格的限制,尽量避免使用机器翻译系统不能很好处理的复杂语言表达。

3 .人机交互式机器翻译

这种方法是试图通过牺牲全自动,而达到获取较高译文质量的目的。基本想法是,在机器翻译存在困难时(例如机器不能准确判定某个多义词在当前环境下的意义时),通过人工干预导引的方法帮助机器进行解决,从而使得最终获得较高质量的译文。典型的人工干预包括译前编辑和译后编辑。译前编辑主要针对机器翻译系统难以处理的结构和词汇,对提交翻译的源语言文本手工进行一些标记和改写;译后编辑则对机器产生的质量不高的译文进行必要的润色,增加译文的可读性和可理解性。要产生可用于出版的译文,目前的机器翻译系统都必须依赖于译后编辑工作。

更为深入的人机交互式翻译研究追求的目标是允许用户在翻译的任何一个阶段都可以参与。这类研究可以根据人机交互发生的阶段分为:①交互式分析,用户帮助系统得出正确的源语言结构,尤其是复杂句子,对多义词进行排歧等。②交互式转换,用户参与选择与源语言结构等价的目标语言结构,排除不适当的转换。③交互式生成,用户协助产生流畅译文,用户在省略、指代、主题化等方面对生成提供指导。

实际上,很多系统并不单纯允许一种类型的交互,而是同时使用多种交互类型。交互式系统也称为人助机译系统。这类系统除解决翻译的技术问题外,还要大力改善用户界面,方便用户参与。

子语言、受控语言以及交互式翻译虽然提高了译文质量,但其带来的限制条件在许多应用场合并不满足,有时也并不必要。在许多情况下,译文并不是用于出版,例如,一个科技人员只是想浏览本领域的外文文献,选择自己感兴趣的文章,这时译文内容只要从总体上能够把握即可。许多应用场合对翻译效率要求很高,大量文献要在短时间内完成,这些情况下只能接受低质量的译文。

9 .1 3　机器翻译研究的发展历程

机器翻译是最早把数字计算机用于非数值处理的领域之一。20 世纪 40 年代计算机诞生后,机器翻译就成为一个颇有吸引力的研究课题。首先,机器翻译研究的动力来自现实世界的要求,随着科技发展,以各种文字为载体的信息越来越多,高效翻译成为一项迫切的任务,翻译自动化显然可以解决这方面的问题。其次,从技术上,人们观察到,既然翻译工作基本上可以由人有规律地完成,如果对这一过程加以分析和模拟,计算机也应该能完成这样的任务。在人工翻译过程中,查词典占去了相当可观的工作时间,联机词典的引入显然能有效改善这一状况。同时,第二次世界大战期间已有过成功应用的信息论、密码学技术似乎预示着这些技术也可以同样应用于翻译问题(当读到一句俄语句子,可以认为它实际上是用英语写成的,只不过是用一种奇怪的符号加了密而已)。所有这些因素,加之战后尤其是 50 年代后期美国需要大量翻译前苏联的科技信息,使得机器翻译研究得以在充分的经费支持下顺利开始。

1949 年,Rockefeller 基金会副总裁 Weaver 写了一份备忘录,并把 200 份这样的备忘录分寄给对机器翻译感兴趣的人。在这份备忘录里,他谈到了一些机器翻译的理论和方法

问题,其中有:语言单位的多种意义问题,语言的逻辑基础问题,密码学方法的应用问题。虽然不是所有的提议都有价值,但它确实引起了人们对机器翻译研究的兴趣。1948 年,由 Booth 和 Richens 领导的伦敦大学机器翻译小组还是世界上唯一的机器翻译研究机构,而自从 Weaver 发表了他的备忘录之后的两年里,美国出现了一大批从事机器翻译的研究机构,包括麻省理工学院、华盛顿大学、加州大学洛杉矶分校、兰德公司、美国国家标准局、乔治顿大学以及哈佛大学等。

有关机器翻译的主要概念、研究课题,如:形态分析、语法分析、译前编辑、译后编辑、同形多义词的歧义消解、语义的中间表示等在当时都已开始提出。1952 年,在麻省理工学院召开了第一次机器翻译学术会议;1954 年,乔治顿大学在 IBM 公司的协同下第一次公开演示了一个机器翻译系统。

乔治顿大学的这个机器翻译系统是一个俄英翻译系统,词典中含有 250 多个词,有 6 条俄语文法规则。在试验时,该系统把 50 个俄语句子(从化学文本中选出)翻译成英语,无需译后编辑,翻译质量非常好。试验进一步向公众和机译事业的赞助者证实了机器翻译是可行的,极大地促进了在世界范围内进行机译研究,中国、前苏联在这时也都开始了机器翻译的研究。

整个 20 世纪 50 年代以及 60 年代前期,机器翻译研究一直在积极进行并呈扩大趋势。并且,在机器翻译的理论要求刺激下,计算语言学作为一门学科诞生了。研究人员对机器翻译前景充满乐观,许多人认为,机器翻译的实现已为时不远,所缺乏的只是完全的语言学数据。但是好景不长,乔治顿机译系统的进一步研究却遇到了困难,随着规模的扩充,翻译质量明显下降,译文需要大量的译后编辑。

机器翻译的深入研究所揭示出的困难促使一些人开始进行反思,并导致一些人开始从根本上否定机器翻译研究。1959 年到 1960 年,著名机器翻译专家 Bar-Hillel 连续发表对当时机器翻译研究的批评性意见。他认为,全自动高质量翻译在当时是根本不可能实现的。机器译文被作为笑话频频引用,人们逐渐开始了对机器翻译研究的重新评价。美国国家科学院于 1964 年 4 月成立了自动语言处理咨询委员会(ALPAC: Automatic Language Processing Advisory Committee),对当时美国政府资助的机器翻译研究进行了调查,并于 1966 年出版了 ALPAC 报告。ALPAC 报告批评了当时的机器翻译研究,并建议大幅度削减对机器翻译的资助。

ALPAC 报告的批评是尖锐的,但其中有很多论点是正确的,尤其是报告中采用的评价机器翻译系统的准则广为以后的研究人员所接受。但是,ALPAC 报告引起的负面效果也是很大的,它导致了机器翻译事业的骤然停滞。到机器翻译研究的再一次全面复兴,其间竟跨过了十几年的时间。

在 ALPAC 报告发表后,全世界只有为数不多的几个小组还在继续从事机译研究,直到 70 年代中期,机器翻译研究一直处于低潮。

70 年代中期发生的一系列事件,促使机器翻译研究又一次开始复兴。加拿大蒙特利尔大学成功地开发出英法翻译系统 TAUM-METEO。该系统能成功地翻译天气预报信息,译文质量令人满意,并在 1977 年投入使用。1976 年,欧洲经济共同体决定购买机译系统 SYSTRAN 作为他们的翻译工具。受人工智能研究进展的影响,一些学者提出的基于知识的机器翻译方法似乎有希望引导机译研究走出困境。并且在这一时期,世界上也首次出

现了一批商品化的机器翻译系统,如 ALPS 系统、Weidner 系统。这些事件证明,机器翻译系统是有使用价值的,并且在领域受限的情况下,还是可以获得高质量的译文。

1982 年,欧共体开始实施一个庞大的机器翻译研究计划 Eurotra, Eurotra 计划试图实现欧共体所有官方语言之间的互译,包括:丹麦语、英语、法语、德语、荷兰语、葡萄牙语和西班牙语。日本于 1980 年在政府和业界的资助下,开展了英日、日英机器翻译 Mu 系统的研制。之后,又由通产省出面,组织了与亚洲四邻国合作研究日、汉、印尼、马来、泰五种语言的多语机器翻译 ODA 计划。欧共体在资助 Eurotra 计划的同时,还资助另一项多语机器翻译研究计划 DLT, 1984 年后该计划改由荷兰政府长期资助。这几个计划在研究规模和深度方面都是空前的。与此同时,世界各地也相继出现了为数不少的双语单向或双语双向机器翻译研制计划。机器翻译研究又一次迎来欣欣向荣的发展景象。

20 世纪 90 年代初,历时 10 年的 Eurotra、ODA 计划均告结束,但遗憾的是,这两个举世瞩目的大规模研制计划都未能取得预期结果;同时人们还注意到,商品化的机器翻译系统的实用程度仍然是相当有限的,日本著名的机器翻译专家长尾真(Makoto Nagao)说:总的来说,当前的系统还不十分完善,还不能让一般人很方便地使用。机器翻译研究人员又一次开始了对包括机器翻译方法在内的一系列问题的反思。一些人注意到,计算机性能的大幅度提高,以及目前可以广泛获得的联机语料,使得 60 年代放弃的经验方法实现的条件成熟了,因而转而采用统计方法处理机器翻译问题。经验主义的复兴引发了许多学者就机器翻译方法问题展开争论,机器翻译进入了一个多种方法并行、混合的时期。

9.2 基于规则的机器翻译

本节概略地介绍一些基于规则翻译系统的基本知识和翻译流程,没有把焦点集中于具体的翻译系统,而是希望对基于规则的方法有一个总体的介绍。

9.2.1 基于规则的机器翻译策略

基于规则的方法可以涵盖的机器翻译系统很广泛,不同的基于规则的系统共同点是,在这些系统中,都有一个用来表达语言学知识的符号系统,系统在规则的驱动下完成翻译任务。因而,按照这样的理解,直接的词对词的翻译、基于转换的翻译以及基于中间语言的翻译都可以归入基于规则的翻译方法。不同的翻译策略之所以不同,关键在于它们对所处理语言的分析深度有所不同。这种认识可以用图 9-4 来描述。

如果对语言不进行分析或进行很少的分析,仅把源语言和目标语言视为单词的线性序列,翻译在单词一级进行,这就是词对词的翻译,也就是图 9-4 最底层描述的情况。这种策略的问题是,不同语言的词汇之间并不存在一一对应关系,尤其对于差异巨大的语言对之间,像英语和汉语之间,有严格一一对应关系的词只占到一个很小的比例,一个词往往和另一种语言中的多个词对应,如何选择出正确的译词,在词对词翻译的架构下很难做到。另外,不同的语言,词序也往往差异很大,利用词对词的翻译,目标语言的词序也很难确定。

一个语言中的单词排列顺序往往和该种语言的句法关系密切,所以为了产生词序合

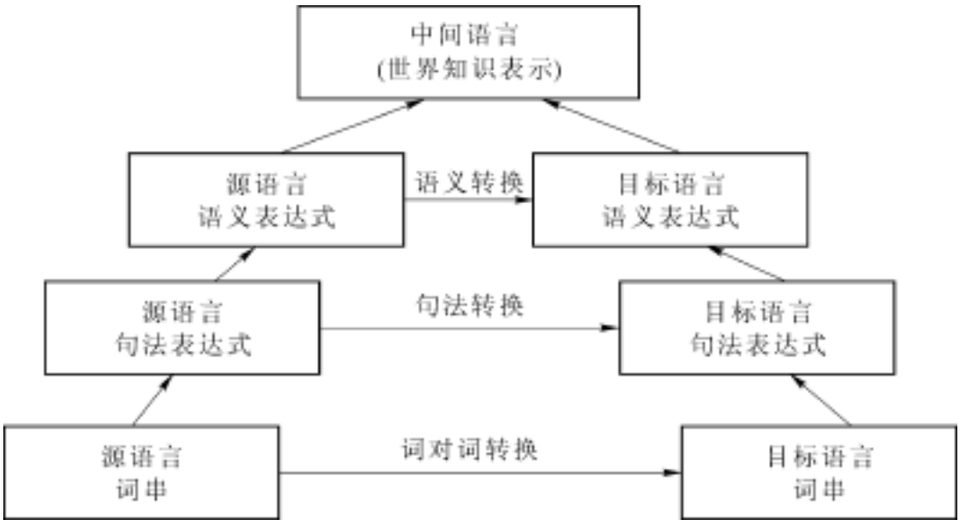


图 9-4　不同的基于规则的机器翻译策略

理的翻译,人们在词对词翻译的基础上更进一步。首先对源语言进行句法分析,得到源语言的句法结构,再根据源语言和目标语言的句法对应关系,得到目标语言的句法结构,在目标语言的句法指导下合理排列译词,这种策略一般可以称为一种基于句法的转换策略,在图 9-4 中,这种策略位于词对词策略的上边。

然而,词序排列合理并不说明翻译的正确,正如人们所熟知的那样,一个句法合理的句子未必是一个有意义的句子。词对词翻译中存在的多义词问题,在句法转换系统中还没有得到很好的解决。同时词序也并不单纯取决于句法,单词的意义往往对词序有着强烈的影响,对于包含这样单词的句子而言,单纯利用句法也往往得不到正确合理的词序。因此要翻译得更好,理论上必然要处理句子的意义问题,转换不能仅在句法一级进行,而且要对语言进行语义分析,因而在图 9-4 中位于句法转换策略上方的是基于语义的转换系统。

但是即使在语义一级进行转换,目标语言的生成仍然强烈地受制于源语言的结构,这往往导致产生的译文生硬不自然。要想译文不受源文的影响,并且能利用一些在翻译时必须的语言学之外的世界知识,进一步的考虑是取消“源语言中间结构”、“目标语言中间结构”,代之以一种独立于任何语言的中间结构,这就是中间语言策略的基本思想,一种大而全的翻译策略,这种策略位于图 9-4 中的最上部。

这里并不打算详尽介绍每一种策略,而主要立足于介绍基于转换的策略。实际上,从目前众多的机器翻译系统来看,完全的词对词策略的系统已经没有,彻底的采用中间语言策略的系统也十分少见,大部分是采用基于转换的策略。

9.2.2　翻译知识的描述和表达

开发一个基于规则的机器翻译系统,首先要做的是设计一个知识表示系统,将翻译过程中所有需要的知识以计算机可以操作的形式表述出来。知识表示系统设计的好坏会影响到系统对知识的表达能力,也会影响到系统能否有效操作这些知识。因而翻译知识的描述和表示是关系基于规则的翻译系统成败的一个关键问题。

一般而言,一个翻译过程需要下述一些知识的支撑:

(1) 源语言知识。系统利用源语言知识分析源语言句子,得到源语言句子的结构和意义。

(2) 目标语言知识。系统利用目标语言知识,产生可以接收的目标语言句子。

(3) 源语言到目标语言的对译知识。在基于转换的系统中,系统需要根据各种级别的对应关系来完成源语言到目标语言的转换。最基本的对译知识,是词之间的对译关系。

(4) 领域知识和世界知识。利用源语言知识、目标语言知识,在领域知识和世界知识的协助下,可以更好地完成对源语言的理解和目标语言的生成。

(5) 有关社会、文化和习俗的知识。在人类完成翻译的过程中,这些知识在准确翻译过程中也起着重要作用。但鉴于目前的处理水平,几乎没有机器翻译系统把该类知识纳入处理范围。因为人们目前并没有有效的方法把这些知识以一种机器可以操作的方式描述出来,所以下面也不再提及这类知识。

从目前发展状况看,基于规则的机器翻译系统所主要依赖的知识仍然是语言学知识,尤其是下面的三类知识:

(1) 形态知识。主要关注词的构成问题。印欧语言有丰富的形态变化。源语言的形态知识是源语言分析的重要依据。同时,为了生成可以接收的目标语言,就需要目标语言的形态知识做支撑。

(2) 句法知识。句法知识关注的是句子和短语的构成问题。短语如何由词构成,句子又是如何由短语构成。如果不是简单的词对词翻译,这些知识对源语言的分析和目标语言的生成也是很关键的。

(3) 语义知识。主要关注不同语言单位的意义问题。如何根据词的意义推导出短语的意义,再如何由短语的意义得到句子的意义。

在具体的机器翻译系统中,有关词的知识一般记录在词典中,源语言的形态知识、句法知识和语义知识记录在源语言词典中;目标语言的形态知识、句法知识和语义知识记录在目标语言词典中;词语之间的对译关系则记录在对译词典中。一般,源语言词典、目标语言词典称为单语词典;而对译词典称为双语词典。有些小型单向机器翻译系统可能并不区分单语词典和双语词典。对于一个机器翻译系统而言,词典可称得上是最重要并且包含最多信息的部件。词典的大小和质量直接影响翻译系统的译文质量和所能处理的文本范围。对一个词而言,其所有知识可以分为两个类型,一类信息记录了该词的内在特性;另一类信息则描述了一个词对其所在环境中其他词的要求和影响。前一类信息包括该词的词性,印欧语中的阴阳性以及人称数等方面的特性;后一类如一般语言学上所说的选择性限制信息(restrictional selection information)以及次范畴化信息(subcategorization information)。多数机器翻译系统采用一种属性-值对的方式来描述词典中词的各种信息。例如,英语单词“button”,可以用下面的属性-值对的方式描述其有关信息:

```
lex = button
cat = v
tense =
finite =
person =
number =
subcat = [SUB:sem-agent:np,OBJ:sem-patient:np]
sem-agent = human
```

sem_patient = clothing

上述词条描述了 button 是一个动词,主语是一个名词短语,宾语是一个名词短语。其中主语是 button 的施事,语义要求是 human;而宾语充当 button 的受事,语义要求是 clothing。词条中有的属性后面没有具体的属性值,一般取默认值,表示遵从一般规律。

当然也可以把有关译词的信息以同样的方式进行记录,但是这样做的缺点是,使得词典只能服务于一个单向翻译系统。为了使词典能用于双语系统,更好的办法是建立一个单独的双语词典,专门记录翻译信息。例如 button 作为动词时,译成中文为"扣";作为名词时,可能多译为"钮扣"。可见,双语词典并不仅仅是列出一系列双语词对,一般还应该描写一个源语言单词翻译成一个目标语言单词的条件。

对于一个句子或短语的分析,一般而言我们关心的是句子或短语的组成成分,以及句子或短语中各个成分的句法功能。各种分析方法和结果可以参见前面基于规则的句法分析中的相关章节。在得到源语言句子的句法结构之后,就要把它转换成目的句子的句法结构。

基于规则的机器翻译大体上是利用所谓的组合性原理来进行转换的,即一个句子的转换可以由句子的各个组成部分的转换合成,下面是一个简单的转换规则:

[HEAD:$HEAD, D-SUBJ:$SUBJECT, D-OBJ:$OBJECT]→

[HEAD:$H, D-SUBJ:$S, D-OBJ:$O]

该规则的含义是:"→"左面描述的是源语言句子的组成情况,中心成分 $HEAD、主语部分 $SUBJECT 以及宾语部分 $OBJECT;"→"右面描述了与之对应的目标语言句子的组成情况。目标语言也是三个部分,其中心成分是源语言中心成分 $HEAD 的译文 $H;主语部分是源语言主语部分 $SUBJECT 的翻译 $S;宾语部分是源语言宾语部分 $OBJECT 的翻译 $O。

转换规则的具体形式和分析的深度有很大关系,并且同一个源语言结构可以转换成多个目标语言结构,因而转换规则中还应该描写不同对应关系成立时的条件。

9.2.3 基于规则系统的基本翻译流程

下面举例描述一个转换式翻译系统的翻译流程。假如机器翻译系统完成的是从德语到英语的翻译,用户输入的句子为

Drehen Sie den Knopf eine Position zurück .

(1) 翻译系统首先把输入的德语句子分解成一个个单词,对每个单词进行形态分析,得到每个单词的词典形式或基本形。根据这些基本形查词典得到每个单词的词性。这里需要注意的是,如果源语言是德语等有形态变化的语言,一定要有一个形态分析器处理单词的形态问题。习惯上在电子词典中,对于具有规则形态变化的单词,词典中仅仅包含基本形态。在此例中除两个冠词以外,所有单词都是其词典形式。在本例中还有一个德语特有的问题要处理,也就是要识别出 Drehen …zurück 是一个可分动词,应合并为 zurück drehen 再查词典,这一般也可通过编写规则的方式来完成。

(2) 利用德语的句法分析器进行句法分析。例如句法分析器可以识别出 den Knopf 是一个名词短语,并且是 zurück drehen 的宾语成分,语义角色是受事。句法分析的结果应

该是一棵句法分析树。句法分析器是一个基于规则机器翻译系统的关键部件,尤其对于源语言和目标语言差异较大时,为了得到好的翻译结果,往往要进行很深入的分析。句法分析主要要解决的问题是歧义问题,如何从众多可能的分析结果中得到正确的分析结果,对于德语和英语这样比较接近的语言,不一定需要一个复杂的句法分析器。

(3) 得到句子的分析树后,系统进入转换阶段。这可以看作两个阶段,第一个阶段,利用双语词典找出每一个单词的英语译词,例如根据 den 的原形 der,查双语词典可以知道其译词应为 the;同样 eine 的译词应为 a。在词一级转换的过程中,要考虑到一词多义的问题,双语词典中不但要指明词和词的对译关系,还应指明对应关系成立的条件。在本例中,为了说明的简洁,假定双语词典的词条为如下的形式:

knopf {cat = n}→button{ cat = n}

eine{ cat = det}→ a{cat = det}

……

在得到所有英文译词后,就可以进入第二个阶段,进行结构的转换,本例英文结构和德文结构基本一致。需要注意的是,该句子是一个祈使句。德语的祈使句一般有主语,而英语祈使句则没有,为了使翻译结果自然,系统应根据句法分析器提供的信息,应用有关规则进行处理。例如可以利用下面的规则将其中的主语去掉:

X{cat = v,mood = imperative}Sie→X

转换结果是一个英文句法树,其对应的句子是

Turn back the button a position .

(4) 系统利用英文的形态生成器把英文单词的词典形式转换成正确的形态。例如动词根据其时态、人称添加词尾,名词如果是复数也要增加词尾等。本例中由于是祈使句,所以无需形态生成工作。

(5) 根据得到英文句法树,输出最终译文:

Turn back the button a position .

需要说明的是,为了说明的简洁,这里没有详细介绍每个阶段所涉及的细节,仅仅是列举了基于规则系统的几个主要工作过程。正如前面指出的那样,基于规则的翻译系统根据分析深度的不同各具特点,这里不可能列举每个系统的所有细节。但是,尽管这些系统形形色色,总体上都遵循着分析—转换—生成或分析—生成这样的阶段,这些必要的阶段在上述例子中都已经体现出来了。

9.3 经验主义及混合机器翻译方法

除了传统的基于规则的机器翻译方法之外,在上个世纪八九十年代分别出现了两种以经验主义为哲学背景的机器翻译方法,即基于统计的机器翻译方法和基于实例的机器翻译方法,本节介绍这两种方法以及混合这两种方法的机器翻译。

9.3.1 基于统计的机器翻译

用统计学方法解决机器翻译问题的想法也并非 90 年代的全新思想,早在 1949 年,

Weaver 就已经提出使用这种方法,由于乔姆斯基等人的批判,这种方法很快就被放弃了。很多坚持用统计方法的人认为,50 年代,经验方法遭到放弃的真正原因是缺乏高性能的计算机和联机语料。现在,计算机从计算速度、容量等各方面都有了很大幅度的提高,昔日大型计算机才能胜任的工作,在今天,由工作站或个人计算机就能够完成,同时也有了大量的联机语料;另外,七八十年代统计方法在语音自动识别、词典编纂领域的成功应用使研究人员对此类方法用于机器翻译充满期待。将统计方法用于机器翻译的研究最为突出的是 IBM Watson 研究中心的 Brown 等人的工作。他们认为:翻译问题可以看成是一个噪声信道问题,如图 9-5 所示。

图 9-5 机器翻译的噪声信道模型

可以认为,一种语言 S(信道意义上的输入,翻译意义上的目标语言)由于经过了一个噪音信道而发生了扭曲变形,从而在信道的另一端呈现为另外一种语言 T(信道意义上的输出,翻译意义上的源语言),翻译问题实际上就是如何根据观察到的 T,恢复最为可能的 S 的问题。

这种观点认为,一种语言中的任何一个句子都有可能是另外一种语言中某个句子的译文,只不过可能性有大有小,而取可能性最大的那个译文应该是风险最小的。因此,在实际操作中,就是要把可能性最大的译文找出来。

设 $P(S|T)$ 表示 S 译成 T 的概率,那么翻译问题就成为了在观察到 T 的前提下,寻找一个 S,使得 $P(S|T)$ 取最大值的问题,即

$$\hat{S} = \arg\max_S P(S|T)$$

利用贝叶斯公式,有

$$P(S|T) = \frac{P(S)P(T|S)}{P(T)}$$

因 $P(T)$ 和 S 无关,故有

$$\hat{S} = \arg\max_S P(S)P(T|S)$$

其中,$P(S)$ 称为语言 S 的语言模型;$P(T|S)$ 称为 S 到 T 的翻译模型。Brown 等认为,尽管因式 $P(S)$ 和 $P(T|S)$ 之间的相互作用十分复杂,但这两个因式还是各有其直观意义。翻译模型 $P(T|S)$ 可以考虑为根据观察到的 T 语言句中的单词选择 S 语言中相对应的单词;而语言模型 $P(S)$ 则给出了 S 语言中的单词在句中的顺序。

因此,在基于统计的翻译系统中要解决三个问题,一是如何计算语言模型 $P(S)$;二是如何计算翻译模型 $P(T|S)$;三是如何在所有可能的 S 中有效地搜索使 $P(S)P(S|T)$ 最大的 S。

先看 n 元语言模型,由 $S = s_1 s_2 \cdots s_n$,不失一般性,得到

$$P(S) = P(s_1 s_2 \cdots s_n) = P(s_1)P(s_2|s_1)\cdots P(s_n|s_1 s_2 \cdots s_{n-1})$$

其中,$P(s_i|s_1 s_2 \cdots s_{i-1})$,$i = 2,3,\cdots$,$n$ 表示在前面 $i-1$ 个词已确定分别为 $s_1 s_2 \cdots s_{i-1}$ 的前提下,第 i 个词为 s_i 的概率。尽管从语言学角度来讲,n 元模型过于简单化了,但是从语音识别等领域已使用的情况来看,它还是十分有效的。Brown 在他的论文中利用“包翻译(bag translation)”试验展示了 3 元模型的有效性。

对于翻译模型,由于涉及两种语言,模型简化较语言模型远远复杂,令 $S = s_1\ s_2 \cdots s_n$, $T = t_1\ t_2 \cdots t_m$,考虑 S 和 T 中单词的对应关系,会发现它们之间既可能是一一对应关系,也可能是多对一、一对多关系,甚至是零对一、一对零关系。Brown 等在他们的文章中,列举了如下的一些例子用以说明英语句子和法语句子中单词一级的复杂对齐关系:

The(1) proposal(2) will(3) not(4) now(5) be(6) implemented(7)

Les(1) propositions(2) ne(4) seront(3) pas(4) mises(7) en(7) application(7) maintenant (5)

其中,第一句英语句子中每个单词后的数字为该单词在该句出现的位置编号;而第二句法语单词后的数字表示与其对应的英语单词的位置编号。

Brown 等对翻译模型做出如下的简化:

$$P(T \mid s) = \prod_{i=1}^{n}\left[P(f_i \mid s_i) \cdot \prod_{j=1}^{f_i} P(t_j \mid s_i)\right] \cdot \prod_{i,j,l} P(i \mid j, l)$$

其中,$P(f_i \mid s_i)$表示 S 中单词 s_i 翻译成 T 中 f_i 个单词的概率,Brown 等形象地称其为繁殖概率(fertility probability)。从法语到英语的情况看,英语中 not 在法语中常用 ne…pas 来表示,即英语中的 not 对应于法语中的两个词,该词的繁殖率 $f = 2$,其繁殖概率记为 $P(2 \mid \text{not})$。模型要求针对任一单词 s 估计参数 $P(0 \mid s)$, $P(1 \mid s)$, …, $P(k \mid s)$,其中 k 为一个假设的上限,即该词最多可能对应的单词数目。

$P(t_j \mid s_i)$称为翻译概率(translation probability),表示单词 s_i 译成单词 t_j 的概率,如英语单词 dog 译为法语单词 chien 的概率可写为 $P(\text{chien} \mid \text{dog})$。

$P(i \mid j, l)$称为变形概率(distortion probability),用以描述翻译过程中造成的单词位置上的变化。Brown 等假定 $P(i \mid j, l)$仅依赖于 T 的长度 l, S 中单词的位置 j, T 中单词的位置 i。

则上述法语句子译为上述英语句子的概率可以通过下式计算:

$$\begin{aligned}
&P(1 \mid \text{The}) \cdot P(\text{les} \mid \text{the}) \\
&\cdot P(1 \mid \text{proposal}) \cdot P(\text{propositions} \mid \text{proposal}) \\
&\cdot P(1 \mid \text{will}) \cdot P(\text{seront} \mid \text{will}) \\
&\cdot P(2 \mid \text{not}) \cdot P(\text{ne} \mid \text{not}) \cdot P(\text{pas} \mid \text{not}) \\
&\cdot P(1 \mid \text{now}) \cdot P(\text{maintenant} \mid \text{now}) \\
&\cdot P(0 \mid \text{be}) \\
&\cdot P(3 \mid \text{implemented}) \cdot P(\text{mises} \mid \text{implemented}) \cdot P(\text{en} \mid \text{implemented}) \\
&\cdot P(\text{application} \mid \text{implemented}) \\
&\cdot P(1 \mid 1,9) \cdot P(2 \mid 2,9) \cdot P(3 \mid 4,9) \cdot P(4 \mid 3,9) \cdot P(5 \mid 4,9) \cdot P(6 \mid 7,9) \\
&\cdot P(7 \mid 7,9) \cdot P(8 \mid 7,9) \cdot P(9 \mid 5,9)
\end{aligned}$$

在得出简化模型后,接下来的工作就是利用实际语料进行参数估计。对于语言模型而言(以 2 元模型为例),需要利用 S 语言的语料估计概率 $P(s_i \mid s_{i-1})$,一般采用相对频率法(Relative Frequency)进行估计,即统计实际语料中单词 $s_{i-1}\ s_i$ 相邻出现的次数除以单词 s_{i-1} 出现的次数,即

$$P(s_i \mid s_{i-1}) = f(s_i \mid s_{i-1}) = \frac{f(s_{i-1}, s_i)}{f(s_{i-1})}$$

其中，$f(\cdot)$代表·在实际语料中的出现频次。Brown 等在法—英翻译试验中使用加拿大议会的会议文集 Hansards 作为训练语料(关于机器翻译用的语料及其下述的对齐问题，在下面一节专门介绍)。该文集同时使用英、法两种语言，Brown 等使用其中的英语部分的 57 万句训练 2 元语言模型。他们限定英语词汇为语料中最为常用的 1 000 个单词，其余的词均用未知词(unknown word)来代替。这样的话，估计总共约需 100 万个参数。

翻译模型中三类参数的估计，需要使用 S，T 两种语言的语料，同时需要已经对齐的语料。Brown 等的做法是首先使用他们提出的基于长度的句子对齐算法，对 Hansards 语料进行对齐，得到约 300 万对齐的句子对，从中选取 117 000 对句子用 EM(Expectation Maximize)算法训练翻译模型，他们限定法语词汇为语料中法语部分最为常用的且能完全覆盖所限英语词汇的 1 700 个单词，估计了大约 1 700 多万参数建立起翻译模型。

有了上述模型之后，翻译过程即为一个解码(decode)过程，对所有可能的 S 计算 $P(S)P(S|T)$，找出取值最大的 $\hat{S}$ 作为 T 的译文。但问题是 S 的数量巨大，Brown 等借鉴了语音识别中的“栈式搜索(stack search)”方法。栈式搜索的主要数据结构是表结构。表结构中存放着当前最有希望的部分对齐结果，算法不断循环，每次循环扩充一些最有希望的情况，直到表中包含一个得分明显高于其他结果的完整的对齐结果时结束。栈式搜索并不能保证得到最优解，因而仅是一种次优化算法(suboptimal search)，因此可能导致错误的翻译。

Brown 等人利用上述翻译进行了翻译试验，他们从 Hansard 语料中选取了 73 个法语句子(不包含在训练语料中)进行翻译，结果分为五类：第一类为“精确类(exact category)”，得到的英语句子和语料中的英语句子完全一样，这样的结果有 4 句；第二类为“替换类(alternate category)”，得到的英语句子和语料中的英语句子不完全一样，但意义相同，这样的情况有 18 句；第三类为“不同类(different category)”，得到的英语句子可以视为法语句子的翻译，但和语料中的英语句子意义不完全一样，这样的结果有 13 句；第四类是“错误类(wrong category)”，得到的是一个合法的英语句子，但不能视为法语句子的译文，这样的情况有 11 句；第五类是“不合乎语法类(ungrammatic category)”，得到的英语句子是错误的，这样的情况有 27 句。前三类是正确的翻译，占 48%。

基于统计的翻译方法在进行参数训练时，无论语言模型还是翻译模型都存在数据稀疏问题(即要翻译的内容在语料中没有出现过，从而导致相应的参数为零)。解决的办法是一方面加大语料的规模；另一方面采用“平滑(smoothing)”技术，利用一定的算法使得取值为零的参数取到适当的值。

同时翻译模型、语言模型在简化过程中也带来了一些缺陷，即在简化和可行之间存在一个权衡问题，上文翻译模型中一个明显的缺陷就是仅支持从 S 到 T 的一多对齐，但不支持从 T 到 S 的一多对齐。这些都说明这些模型都还需要进一步的改进。从一定角度来看，基于统计的翻译方法目前把语言视为一种无结构的单词串，如何把语言的形态结构考虑进去需要进一步的研究。

9.3.2 基于实例的机器翻译

基于实例的机器翻译(EBMT: Example-Based Machine Translation)方法的基本思想，是由日本著名机器翻译专家长尾真提出的。长尾真 1984 年发表的论文《A Framework of a Me-

chanical Translation between Japanese and English by Analogy Principle》可视为这一研究领域的起点。长尾真在文章开始探讨了外语初学者(日本人学习英语)翻译句子的基本过程,初学外语的人总是记忆最基本的英语句子和对应的日语句子,长尾真在文章中说:"第一步完全是大量的英、日对应的类似的句子和词汇的记忆训练,我们没有翻译理论来告诉学生,他得靠自己的直觉获得翻译机理,他得对比不同的英语句子和对应的日语句子,他得从大量例子猜测、推理句子的结构。"参照这个学习过程,我们考虑给我们的机器翻译系统一些例句及对应的译语。系统一定能够识别所给例句的相似和差异之处。"长尾真进一步阐述说:"关于翻译我们最基本的思想是:①人类不通过做深层语言学分析翻译句子。②人类的翻译过程:首先正确分解输入句子,分解成短语碎片(经常是格框架单元);接着,把这些短语碎片译成其他语言短语,最后把这些短语构成一个长句。每个短语碎片采用类比的原则进行翻译。"长尾真还提到,传统的基于深层次分析的方法在处理截然不同的语种时,结构转换往往很困难,他说:"所有的欧洲语言都有一定的共同基础,但对于两种差别较大的语言来说,如英语和日语,翻译则有大量的困难。有时完全相同的内容,结构完全不一样,仅通过详尽的句法分析不能获得这种翻译。如果我们找到每个词汇的对应译词,目标语句子的合成还是几乎不可能,而且不参看上下文,从众多的候选译词中选择正确的译词也非常困难。"因此,我们应利用一种由例子引导推理的机器翻译方法,或称之为基于类比原则的机器翻译。而利用这种方法的原因之一就是:详尽分析源语句子对结构完全不同的语言之间的翻译没有用处。为此,应当在词典中存储例句,还得有一个为给定句子找类似例句的机制。"这种方法的系统中知识增加很容易,仅需要增加新的词汇及作为用法的例句和它们的译语,不需要深层次的分析信息。"长尾真还认为:"语言学理论迅速反复变化,若干年后,有些模型可能被抛弃,与之相反,语言数据和使用法则会保持较长时期不变。"

这一方法的基本原理归纳起来非常简单,系统的主要知识源是双语对照的翻译实例库。实例库主要有两个字段,一个字段保存源语言句子;另一个字段保存该句对应的译文,每当输入一个源语言句子 S 时,系统利用 S 和实例库中的源语字段进行比较,找出其中和 S 最为相似的句子 S',并模拟 S '的译语 T '构成 S 的译语 T 然后输出。

这种方法有以下一些优点,从而吸引了不少研究人员。

(1) 系统维护容易,系统中知识以翻译实例和义类词典等形式存在,可以很容易地利用增加实例和词汇的方式扩充系统。

(2) 容易产生高质量的译文,尤其是利用了较大的翻译实例或和实例精确匹配时更是如此。

(3) 可以避免一些传统的基于规则机器翻译必须进行的深层次语言学分析。

基于实例的翻译方法主要有三个关键问题需要加以解决。

(1) 双语对齐问题。如上文所述,实例库中要存储源语言及其相应的目标语,要大规模扩充实例库,必须解决双语对齐问题。而且,在许多实例翻译的具体实现中,不仅要求句子对齐,更多的是要求词汇或短语一级的对齐(在下一节将集中介绍关于对齐的研究)。

(2) 一个有效的实例匹配检索机制,这个问题又涉及到三个方面。

首先,必须确定检索应该在哪一种级别上进行,是在句子一级(sentence-level),还是在亚句子(sub-sentence)一级。很多研究人员认为,基于实例翻译的潜力在于利用多个亚句

子级的实例碎片(fragment),这就要求匹配要在亚句子一级进行,同时实例库要求亚句子一级的对齐。但是利用的实例碎片越小,不但碎片的边界难于确定,还会引入较多歧义,使翻译质量下降。

其次,要建立一套相似度准则(similarity metric)。相似度准则主要用来确定两个句子或短语碎片是否相似。目前关于相似度准则的研究比较多,具体方法由于各个系统的设计思想不同而不尽相同。多数方法可以视为一种基于单词的方法(word-based approach),这种方法逐一比较两个句子(或亚句子)中各个相应单词的相似度,然后加以组合,形成句子(或亚句子)的相似度,最为常见的是使用"语义距离(semantic distance)"的概念。语义距离的计算一般以树形的义类词典(thesaurus)为基础,通过计算"最为具体的共同抽象(MSCA: the Most Specific Common Abstraction)"而获得。也有研究认为,不仅要比较两个单词在语义方面的相似度,而且还要考虑形态等方面的相似度。最后对各项相似度加权计算,Nirenburg 认为可以在形态、词类诸多方面加以考虑,形成不同的等价类指导匹配,他开列了一些要考虑的因素,还考虑了在实现时需要的背景知识和工具,如表 9-1 所示。

表 9-1　判别相似度时应加以考虑的因素

	相似度判别依据	背景知识和工具
1	精确匹配	
2	仅形态不相同	形态分析器及相应词典
3	形态不同,词类不同	形态分析器、词类标注程序、词典
4	指定意义上的同义词	同义词典或义类词典、语义标注程序及各种知识资源
5	所有意义上的同义词	义类词典或同义词典
6	指定意义上的上义位一样	义类词典、语义标注程序及各种知识资源
7	所有意义上的上义位一样	分层的词汇数据库或义类词典
8	指定意义上的下义位一样	义类词典、语义标注程序及各种知识资源
9	所有意义上的下义位一样	分层的词汇数据库或义类词典
10	指定意义上的反义词	反义词词典、语义标注程序以及相关知识资源
11	所有意义上的反义词	反义词词典
12	仅词类相同	词类标注程序及词典

除了基于单词的方法外,还有句法驱动(syntax driven)的方法、基于字符(character-based)的方法及混合方法。对这一问题提出的方法尽管很多,但目前的相似度计算还有很多不足。

最后,实例检索的效率问题。由于实例的数量通常十分巨大,为了保证翻译系统有合理的响应时间,检索效率就十分关键,除采用高效的索引机制外,一些研究人员还探索了引入并行机制。

(3) 如何根据检索到的实例生成输入源语的译文。由于基于实例的机器翻译不强调

对源语的分析,生成时往往缺乏必要的信息。目前有些系统采用的方法是把传统的机器翻译方法结合进来,但更多的方法是仅对相应实例的译文进行简单的修改,如进行一些词汇的替换、删除和插入工作。

9.3.3 混合的机器翻译方法

上面分别介绍了基于规则的理性主义方法和基于统计的经验主义方法。综观机器翻译的整个发展历程,和自然语言处理技术有着类似的情况。20 世纪 90 年代以前,机器翻译的主流方法一直是理性主义方法。50 年代统计方法虽然曾一度盛行,但很快被放弃了。随后由于机器翻译任务的艰巨,沿着理性主义方法并未取得突破性进展,60 年代更是一度停滞不前。70 年代发展了基于知识的方法,当时认为,对机器翻译事业而言,关键是建立基于知识的翻译系统,可事实远非如此。80 年代,几个耗资巨大的多国机器翻译研究计划也未能获得理想结果。因此,90 年代统计方法重新复苏引起了广泛关注,似乎凭借统计方法就能使得机器翻译研究从僵局中走出来。然而统计方法也并非无懈可击,因此,学术界围绕两种方法展开了激烈的争论,各自为自己的方法辩护。理性主义认为,统计模型对结构处理乏力且过于简单,因而有很大的局限性,乔姆斯基指出的统计方法的缺点依然存在,如:如何解决像主、谓一致(subject-verb agreement)这样的远距离制约(long-distance constraint)问题?如何解决形态问题?甚至有人指责统计翻译是"石头汤",意指统计翻译取得的有限成功仍然应归功于其在局部采用的语言学方法。经验主义方法则认为,传统方法不能彻底解决机器翻译问题,人工智能方法也不行,基于知识的方法曾被认为是解决机器翻译问题的关键方法,可是现在留给我们的仍然是一些写在纸上的合情合理的例子,如:" The Soldiers fired at the women and I saw several fall ."这里" several "指" women "而不指别的,并不能通过统计的办法或语言学规则的办法而得知。至今为止,还没有见到含有丰富世界知识的实用知识库(knowledge bank)出现。1991 年 DARPA 开始了机器翻译资助计划,一开始同时资助三个机器翻译系统。三个系统分别采用了不同的翻译方法,IBM 的 Brown 等的法英系统 CANDIDE,采用纯统计方法,支持人助机器翻译(HAMT: Human Assistant Machine Translation)和全自动(FAMT: Full Automatic Machine Translation)两种工作方式;卡耐基-梅隆大学机译中心、南加州大学信息科学研究所、新墨西哥州立大学计算研究实验室合作研制的人助机器翻译系统 PANGLOSS,采用基于知识的方法,主要完成从西班牙语到英语的翻译;Dragon System 公司的计算机辅助翻译系统 LINSTAT,采用统计方法和语言学方法混合的方法,辅助翻译人员完成从日语到英语的翻译。DARPA 每年举行一次评价,不同方法直面交锋,争论更趋激烈。

在一定程度上,争论双方似乎都企图打倒对方,未能正视对方的优势。但是一些学者包括参与争论的一些学者却慎重思考了这个问题,认为两类方法各有优缺点,在许多方面是互补的,应该在研究时把两种方法结合起来。

首先,两种方法孰优孰劣,无法断言,正如 Church 在《Statistical MT ! = Stone Soup》一文中所说的那样:" 谁知道谁更重要,语言学方法还是统计学方法?争论令人厌倦,两种方法都未取得重大进展,我们距离 Bar-Hillel 提出的终极目标 FAHQMT(Full Automatic High Quality Machine Translation)还有很长的路要走。也许统计方法向此目标迈进了一步,也许没有。"经验方法绝非一无是处," 经验方法产生了令人十分感兴趣的副产品:对齐程序。

对齐程序可以有效的用于翻译重用(translation reuse)”,许多应用场合,译文需要不定期或定期的更新,翻译重用使得人们很容易的只处理其中发生改变的部分。另外,“术语(teminology)问题对翻译人员而言一直是瓶颈问题,并行文本(parallel text)可以帮助翻译人员解决专业知识不足的问题,并行文本提供了大量的翻译实例,使得翻译人员很容易参考。”“所以,统计方法给我们提供一套十分有效的术语工具和重用工具,不像传统的机器翻译,这些工具不在人类做的更好的地方和人比赛(例如人可以用最容易的词汇和句法进行翻译),而是弥补人类的不足之处。”

其次,两类方法各有自己的应用范围,有些场合,质量问题无关紧要,但涉及不受限的领域,如吸收型的翻译任务就可以采用统计翻译;有些场合,如传播型的翻译任务,常常领域限定,要求较高的翻译质量,则可以采用传统的基于语言学的方法;还有一些场合则可采用二者混合的方法。

在一定程度上两类方法的优缺点是互补的,如表 9-2 所示。

表 9-2　规则方法与统计方法之比较

特　　点	基于符号的方法	统计方法(不含 EBMT)
健壮性、覆盖范围(robustness/ coverage)	不好	较好
质量、流畅性(quality/ fluency)	较好	不好
表示(representation)深度	很深	较浅

如果两种方法能有效地结合起来,则一定能有效地改善机器翻译的性能。事实上,在实际的机器翻译系统研制过程中,也体现出了这样的要求,正如 E. Hovy 所说:(对传统方法而言)如果你想建造一个非玩具的机器翻译系统(non-toy system),并希望它能健壮地处理以前未曾处理过的输入,最终你总是要包含一些统计处理模块。(对统计方法而言)如果,你希望你的系统的输出质量合理,你就得包含一些语言学的驱动知识和模块。为什么会这样呢?首先,对于纯统计系统而言,一个单词以“t”结尾,或一个单词是大写的,还是它伴随着“the”、“a”一起出现,这些都没有什么区别。但对于人来讲,很容易知道,大写的单词很重要,在许多语言中它们是专有名词,翻译规律和其他单词不同,如果一个单词前有冠词,则它是一个普通名词,它有数的变化,且影响动词形态。如果将这些抽象语言知识预先告知统计系统,显然可以改善系统性能。其次,对于传统方法而言,系统中需要有大量的词汇特征、语法范畴、甚至大量语义知识,这些知识必须实实在在地建立起来,这就需要大量人力、物力,编纂速度异常缓慢,唯一的途径是使知识获取自动化或半自动化,减少每一词条的信息量,利用机器获取词条或语法模式,这样短时间内就能使词典、语法规则达到相当规模和覆盖范围。

实际上,随着 DARPA 的几次评价之后,各参评系统在自己的立场上都有所松动,都开始采用了一些其他方法改善自己的系统,如 Nirenburg 提出了多引擎(Multi-engine)的概念并在 PANGLOSS 系统中予以具体实施。同时,DARPA 又资助了南加州大学的另一个机译系统 JPANGLOSS,这个系统也是既采用统计方法,也采用传统方法。机器翻译系统翻译方法呈现出一种混合趋势。下面简要介绍 PANGLOSS MarkⅢ所采用的多引擎体系。

最初,PANGLOSS 被设计为一个纯基于知识的系统,采用中间语言的体系结构。但后来在外部定期评价的压力下,越来越趋向于采用更为折衷的方案,这是因为,一定程度上

基于知识的系统必须依靠大规模知识库,但知识库的建造十分耗费时间。1992 年夏季的评价中,PANGLOSS Mark Ⅰ的表现并不好,这促使该课题组人员决定临时引入一些其他资源,使得译文质量在短期内有明显改观,用于应付评价;与此同时,继续坚持研制基于知识的系统的主部件,并在基于知识的系统的部件达到一定阶段时,将其他部件去掉。果然,PANGLOSS Mark Ⅱ评价结果优于 PANGLOSS Mark Ⅰ。可是,这种趋向混合方法的趋势并没有到此为止,他们的思想进一步向前发展,决定不再抛弃其他部件,而使多个翻译部件共存于一个系统中,这就是 PANGLOSS 目前的版本 PANGLOSS MarkⅢ。PANGLOSS MarkⅢ和其他任何翻译系统不同的是,该系统采用了不止一个翻译引擎,每个引擎都试图翻译整个或部分输入源语,由系统综合评价各引擎的输出,最后系统输出总体最好结果。这样显然有利于排除局限于具体方法的不足。目前系统有三个翻译引擎:一个基于知识的引擎,是该系统的一个主要引擎;一个基于实例的引擎;一个基于词汇转换的引擎。总体结构如图 9-6 所示。

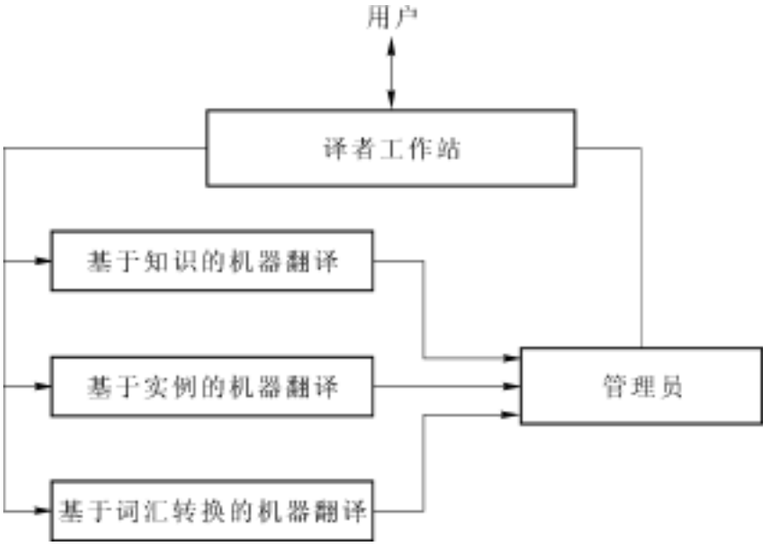


图 9-6　多引擎机器翻译方法

值得指出的是,多引擎翻译并非多种方法结合的唯一途径,也有一些更细粒度的结合方法。如在翻译系统中,某些部件采用基于规则的办法构造,而另外的一些部件则采用基于统计或基于实例的方法构造。

9.4 双语对齐

从上面的介绍可以看到,在基于统计和基于实例的翻译方法中,双语语料库的建设以及对双语语料的对齐(alignment)是一个关键的问题。基于统计的方法需要大规模双语语料,其翻译模型、语言模型参数的准确性直接依赖于语料的多少和对齐的质量;在基于实例的方法中,需要大规模双语实例库供比较,系统的翻译能力和准确性直接依赖于实例库的规模和实例对齐质量。

双语语料库(bilingual corpus)也叫并行语料库(parallel corpus),语料库中不仅收集源语言的语料,也必须收集它们的目标语言翻译。例如,一个汉英双语语料库不仅收集有文本

的汉语原文,还收集有文本的英语译文。因为同时要求有原文和译文,双语语料库的收集较单语语料库要困难一些。但在一些双语国家或多语文化中,存在有非常丰富的双语资源。例如在加拿大、瑞士和中国的香港。

同单语语料库需要加工一样,要想使得双语语料库发挥更大的作用,对双语语料库的加工是十分重要的。对双语语料库的加工包括对每种语言的文本进行断词、标注、分析等,这种加工同单语语料库的加工是类似的(最近,也有一些对双语同时进行某种加工的研究)。但是,对双语语料库的加工中最重要也是最直接服务于机器翻译的是双语对齐工作,这也是在单语语料加工中所不会遇到的。

简单地说,双语对齐指的是在两种语言文本的不同语言单位之间建立对应关系,也就是确定源语言文本中哪个(些)语言单位和目标语言文本中哪个(些)语言单位互为翻译关系。在互为译文的两种语言文本之间建立对应关系,可以有各种级别,最基本的是句子一级,其次可以是单词一级,还可以是短语一级。下面主要介绍在句子和词两个级别的对齐。

9.4.1 句子一级的对齐

为双语文本建立句子一级的对齐关系,就是要确定源语言文本中哪个(些)句子和目标语言文本中哪个(些)句子互为译文。例如下面的两段文字,一段为中文,另一段为英文,在其中插了几个编号。

①中国支持在平等参与、协商一致、求同存异、循序渐进的基础上,开展多层次、多渠道、多形式的地区安全对话与合作。②中国参加了东盟地区论坛、亚洲建立协作与建立信任措施会议、亚太安全合作理事会和东北亚合作对话会等活动,主张通过这些政府和民间讨论安全问题的重要渠道,增进各国的相互了解与信任,促进地区和平与稳定。

③China advocates regional-security dialogue and cooperation at different levels, through various channels and in different forms .④Such dialogue and cooperation should follow these principles: participation on an equal footing, reaching unanmimity through consultation, seeking common ground while reserving differences, and proceeding in an orderly way and step by step . ⑤China has participated in the ASEAN Regional Forum (ARF), Conference on Interaction and Confidence-Building Measures in Asia (CICA), Council on Security Cooperation in Asia and Pacific Regional (CSCAP), Northeast Asia Cooperation Dialogue (NEACD) and other activities, holding that all countries should further mutual understanding and trust by discussions on security issues through these important governmental and non-governmental channels, so as to promote regional peace and stability .

按照一般的认识,把句号作为句子的结尾标记,上述汉语部分含有两个句子,分别编号为句子①和②,英语部分则含有三个句子,编号为句子③、④和⑤。不难发现,汉语中句子①被翻译成英语中句子③和④,一个汉语句子同时对应两个英语句子。汉语句子②则和英语句子⑤互为译文关系。

不难想象,在两种语言的句子间建立对齐关系,对齐关系可能会有多种模式,最为常见的是源语言文本中一个句子和目标语言文本中一个句子对应(1-1 即 1 对 1),除此之

外,还可能有如下几种情况:

(1) 源语言中一个句子和目标语言中两个或多个句子对应(1-2 或 1-n);

(2) 源语言中两个或多个句子和目标语言中一个句子对应(2-1 或 n-1);

(3) 源语言中两个或多个句子和目标语言中两个或多个句子对应(2-2 或 n-n);

(4) 在有些情况下,翻译未必忠实于原文,常常会有省略不译的现象,也有为了使目标语读者更容易理解原文而增加解释性成分的现象发生,在这些情况下,也会有一种语言的文本的某个(些)句子在另外一种语言中没有句子与之对应的现象发生(0-1 和 1-0)。

目前,句子对齐的方法基本上可分为两类:基于长度(length-based)的对齐和基于单词(word-based)的对齐。在一定条件下,无论基于长度的方法还是基于单词的方法都达到了90%以上的准确率。从效率上说,基于长度的办法要快于基于单词的方法。

在基于长度的对齐方法中,有的以句子中单词数作为句子长度的度量,有的以句子中字符数作为句子长度的度量。下面仅对 Gale 等人的方法进行介绍。

基于长度对齐利用了一个非常简单的统计模型,主要基于以下事实:一种语言中长句翻译成另外一种语言后句子仍然是较长的;反之,短句翻译成另外一种语言后句子也仍是较短的。即互为翻译的两个句子在长度上高度相关。Gale 等人基于以上事实定义了句长的距离度量,然后利用动态规划的办法完成句子的对齐。Gale 等人的对齐算法的另一前提是互为译文的句子在各自文本中的顺序没有剧烈变化。

距离度量基于这样的假设,语言 L_1 中的每一个字符在语言 L_2 中所对应的字符数 c 是一个随机变量,而且该随机变量呈正态分布 $N(c, s^2)$,定义 $\delta = (l_2 - l_1 c)/\sqrt{l_1 s^2}$, l_1, l_2 分别为欲对齐的两句的长度(也是随机变量),则 δ 服从标准正态分布。在此基础上定义距离度量为 $-\log P(\text{match} \mid \delta)$。(由于一种语言中的一个句子可以对应另一种语言的一个句子,也可以对应多个句子,Gale 等考虑了 1-1, 1-0, 0-1, 2-1, 1-2, 2-2 六种情况,统一用 match 符号来表示。)Gale 等人对以上假设作了验证,并用 UBS(Union Bank of Switzerland 出版的经济报告,同时使用英、法、德三种语言)语料分别针对英德、英法两种情况估计了参数 c 和 s^2。结果在两种情况下,参数取值基本相等,因而他们认为参数取值可以认为是独立于语言的,其中, $c \approx 1$, $s^2 \approx 6.8$。Gale 认为对于任意两种欧洲语言该参数是合适的,但该参数对于英汉等语种并不合适,需要重新估计。对于概率 $P(\text{match} \mid \delta)$,可以利用贝叶斯公式转化为 $P(\delta \mid \text{match})\ P(\text{match})$,其中 $P(\text{match})$是常数,可以根据手工标记的语料统计出来;而 $P(\delta \mid \text{match})$又可通过下式进行估计:

$$P(\delta \mid \text{match}) = 2(1 - P(\delta))$$

其中

$$P(\delta) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\delta} e^{-z^2/2}\, dz$$

式中,z 为服从正态分布的随机变量,可通过查表或其他数值方法进行计算;δ 可由上面的定义进行计算。

根据以上的距离定义,考虑各种可能对齐情况下的距离,有:

$d(x_1, y_1; 0, 0)$　1-1 对齐的情况,句子 x_1 和句子 y_1 对齐时的距离值

$d(x_1, 0; 0, 0)$　　1-0 对齐的情况，句子 x_1 不对应任何句子时的距离值

$d(0, y_1; 0, 0)$　　0-1 对齐的情况，句子 y_1 不对应任何句子时的距离值

$d(x_1, y_1; x_2, 0)$　2-1 对齐的情况，句子 x_1, x_2 和句子 y_1 对齐时的距离值

$d(x_1, y_1; 0, y_2)$　1-2 对齐的情况，句子 x_1 和句子 y_1, y_2 对齐时的距离值

$d(x_1, y_1; x_2, y_2)$ 2-2 对齐的情况，句子 x_1, x_2 和句子 y_1, y_2 对齐时的距离值

然后利用动态规划算法通过计算两个文本的最小距离的办法确定两个文本的句子对齐情况，动态规划算法可总结为下面的递归等式

$$D(i, j) = \min \begin{cases} D(i, j-1) + d(0, t_j; 0, 0) \\ D(i-1, j) + d(s_i, 0; 0, 0) \\ D(i-1, j-1) + d(s_i, t_j; 0, 0) \\ D(i-1, j-2) + d(s_i, t_j; 0, t_{j-1}) \\ D(i-2, j-1) + d(s_i, t_j; s_{i-1}, 0) \\ D(i-2, j-2) + d(s_i, t_j; s_{i-1}, t_{i-1}) \end{cases}$$

其中，s_i，t_j（$i = 1, 2, \cdots, I; j = 1, 2, \cdots, J$）分别为两个文本中的句子。算法开始时 $D(i, j) = 0$。

Gale 等人利用该对齐算法主要对 UBS 和 Canadian Hansard 语料进行了对齐，对齐准确率达 96%，但其中一多、多一、多多对齐错误率较高，如 2-2 对齐错误率为 33%，由于不考虑 3-1，3-2，这些情况下的错误率均为 100%。然而由于 1-1 对齐所占比例很大，整体上准确率仍然很高。若选取距离值较大的 80% 的对齐结果，则对齐准确率可达99 .3%。同时 Gale 还认为，以字符为单位度量句长比单词好，这是因为字符数更多，较少具有不确定性。Gale 还验证，采用和语言有关的参数 c, s^2 对结果影响不大。

上述基于长度的对齐方法没有利用任何词汇信息，然而事实上往往可以从词汇对齐关系推导出句子的对齐关系，在基于长度对齐的方法中，如果两种语言的句子长度差别不大，那么利用词汇信息可以使对齐结果更加可靠。Martin Kay 和 Martin R scheisen 曾经提出一种利用部分词汇对齐信息推导句子对齐信息的方法，即利用词汇共现的统计特性来确定部分词汇的对齐关系，再利用句子和这些词之间的包含关系确定句子的对齐关系。

在基于长度对齐的基础上，也可以利用一些明显的词汇对齐关系改进对齐效果。例如在有的双语文本中，出现了大量的人名、地名、数字、日期，这些词之间的对应关系非常容易确定，对这些信息加以利用，就有利于对齐结果的改进，如下面的汉语文本和英语文本：

第三阶段为 1985 年至今。1995 年，中国粮食总产量达到 4 .666 亿吨，11 年间年均递增 1 .2%。这一时期，中国政府在继续发展粮食生产的同时，积极主动地进行农业生产结构调整，发展多种经营，食物多样化发展较快，猪牛羊肉、水产品、禽蛋、牛奶和水果产量分别达到 4 254 万吨、2 517 万吨、1 676 万吨、562 万吨和 4 211 万吨，比 1984 年分别增长 1 .8 倍、3 .1 倍、2 9 倍、1 .6 倍和 3 .3 倍。

The third phase (1985—present): In 1995 the country's grain output totaled 466 .6 million tons, increasing by an average of 1 .2 percent a year over the previous 11 years .While continuing to

develop grain production in this period, the Chinese government has initiated measures to readjust the structure of agricultural production and develop a diversified agricultural economy . At the same time rapid progress was achieved in the production of various other kinds of foodstuffs, with the output of meat (pork, beef and mutton), aquatic products, eggs, milk and fruit reaching 42 .54 million tons, 25 .17 million tons, 16 .76 million tons, 5 .62 million tons and 42 .11 million tons respectively, or 2 .8, 4 .1, 3 .9, 2 .6 and 4 .3 times the 1984 firgures, respectively .

根据上述文本中大量数字很容易确定汉语句子和英语句子的对齐关系,这种有明显对应关系的词一般称为“锚点(anchor)”。

9.4.2 词汇一级的对齐

为双语文本建立词汇一级的对齐关系,就是要确定源语言文本中哪个(些)词和目标语言文本中哪个(些)词有对应关系。词汇一级对应关系的确定难度远远大于在句子一级进行对齐。在上述句子对齐关系中,对齐句子不能相互交叉,也就是说满足下面一种顺序上的制约关系。如果源语言中句子 i 和目标语言句子 j 有对齐关系(i, j),那么一定不存在对齐关系(m, n)使得 $m > i$ 且 $n < j$,或 $m < i$ 且 $n > j$。在词汇一级,这个约束不再成立,尤其是目标语言和源语言差异比较大时,词汇交叉对齐的现象很常见。在词汇一级对齐中,源语言句子中一个词,可能对应目标语言一个词,也可能对应多个连续的词或不连续的词,甚至没有对应的词。

针对词汇一级的对齐,目前提出的主要方法有:基于统计的方法、基于词典的方法和混合的方法。在众多的研究中,统计模型居多数,统计方法中采用的模型也不尽相同,有的简单,例如基于共现频率模型的技术;有的复杂,例如基于概率翻译模型的技术。统计方法要求有足够的双语语料,非此不能获得可靠的统计数据和避免数据稀疏问题。目前词汇一级的对齐技术并不成熟,下面简单介绍一种基于词的关联度(association degree)的词汇对齐方法。

基于关联度的词汇对齐建立在句子对齐的基础上,基本思想是,如果一个词 s 和 t 有对应关系,那么在一个对齐的句对中,源语言部分出现了 s,则目标语言部分很可能出现 t。可以用一个 2×2 联立表来刻画一对词在双语语料中的分布情况,如表 9-3 所示。

表 9-3 词 s 和 t 在双语语料中的分布情况

	t	$\neg t$
s	a	b
$\neg s$	c	d

在单元格 a 表示的是在句子一级对齐的语料中,有多少句对同时含有源语言单词 s 和目标语言单词 t。单元格 b 表示有多少句对含有源语言单词 s,但不含目标语言单词 t。单元格 c 表示有多少句对不含源语言单词 s,但含有目标语言单词 t。而单元格 d 则表示既不含有源语言单词 s,也不含有目标语言单词 t 的句对数。按照这个定义,有下面的关系成立:

$$a = \text{freq}(s, t)$$

$$b = \text{freq}(s) - \text{freq}(s, t)$$

$$c = \text{freq}(t) - \text{freq}(s, t)$$

$$d = N - a - b - c$$

这里,freq(x)表示含有 x 的句对数; N 表示所有的句对总数。

例如,对一个英语-法语的句子对齐的双语语料库统计,英语单词 house 和法语单词 chambre 在对齐句对中的分布情况如表 9-4 所示。英语单词 house 和法语单词 communes 在对齐句对中的分布情况如表 9-5 所示。

表 9-4　英语单词 house 和法语单词 chambre 在对齐句对中的分布情况

	chambre 出现	chambre 不出现
house 出现	31 950	12 004
house 不出现	4 793	848 330

表 9-5　英语单词 house 和法语单词 communes 在对齐句对中的分布情况

	communes 出现	communes 不出现
house 出现	4 974	38 980
house 不出现	441	852 682

那么到底如何衡量两个词之间的关联度呢?一般的衡量方法为使用互信息。对上述两个例子计算互信息:

$$I(\text{house},\text{chambre}) = \text{lb}\frac{31\,950N}{(31\,950+12\,004)(31\,950+4\,793)} = 4.1$$

$$I(\text{house},\text{communes}) = \text{lb}\frac{4\,974N}{(4\,974+38\,980)(4\,974+441)} = 4.4$$

则有

$$I(\text{house},\text{chambre}) < I(\text{house},\text{communes})$$

这意味着 house 和 communes 关联度更强;然而事实并非如此,house 和 chambre 的关联度理应更强。因此采用 ^{2} 统计值来衡量两个词之间的关联度。^{2} 统计值定义如下:

$$^2 = \frac{(ad-bc)^2}{(a+b)(a+c)(b+d)(c+d)} \quad (0 \leq {}^2 \leq 1)$$

可以得到 house 和 chambre 的 ^{2} 统计值是 0.62;而 house 和 communes 的 ^{2} 统计值是 0.098。house 和 chambre 的关联度更强。

需要指出的是,利用这种基于关联度的方法发掘词汇对应关系并不能可靠对齐所有单词,对齐的召回率往往较低。

9.5　机器翻译系统的使用

正如前文指出的那样,机器翻译是一项非常困难的研究课题。目前市场上的机器翻译产品还远远不能令人满意,并且,在短时期内,也不太可能会有全自动高质量的机器翻译系统出现。但这并不能说明,目前的机器翻译技术一无是处,在一些应用场合,机器翻译系统确实可以成功满足人们的一些翻译需求。本节简单介绍目前见诸文献的有关机器翻译的使用,并简单介绍一些有关机器翻译的评价。

9.5.1　目前对机器翻译的需求

无论是利用机器进行翻译,还是由翻译专家提供翻译,翻译的服务对象也就是说翻译的用户不外乎有两个目标,一是希望把自己的信息传播出去,例如,翻译的用户可能是一

些跨国企业,为了在国门之外开拓市场,他们需要向使用不同语言的人群宣传自己的产品;为了使他们的产品能够被正确使用,他们也需要把他们的产品的使用说明和技术资料介绍给他们的外国客户。很显然,翻译的用户希望把自己的信息准确无误地传播出去,因此可以把这种翻译需求称之为一种以信息传播为目的的翻译需求。与此相反,另外一种翻译需求不以信息传播为目的,而是翻译的用户希望吸收不同语言的信息,希望了解以自己所不通晓的语言为载体的信息,如情报部门希望了解敌对国家的军事情报、经济情报;科技人员也要掌握其他国家在某个领域的研究进展。显然,翻译用户希望准确无误地吸收信息,这是一种以吸收信息为目的的翻译需求。

很难笼统地说,信息传播对翻译质量的要求高于信息吸收,因为传播和吸收实际上是信息流向的两极。也很难令人信服地说明,用户对他所阅读的翻译文本的质量毫不关心,在任何情况下都可以不理会其中的错讹之处。因而,联系到目前的机器翻译水平而言,无论对于信息传播还是对于信息吸收,机器翻译都不能完全胜任。但是,尽管如此,机器翻译在这两类翻译需求中都已经发现了自己的位置。

对于信息吸收型用户而言,例如科技工作者、情报工作者,他们往往面临的问题是有太多的文献资料需要浏览,而并不是所有文献都具有参考价值,因而自然而然就产生了这样一种需求,如何从众多的外语文献中发现最有价值的内容。在这种情形下,雇佣翻译专家翻译所有文献既不经济也不现实。因为,通常这些文献技术性很强,并非所有通晓某种外语的翻译人员都可以准确转译,只有既通晓外语又具备专业知识的专家型翻译人才才能胜任,这显然翻译费用是极其昂贵的,并且,这样的人才也不会很多。此外,这些文献往往具有很强的时效性,短时期内完成大量翻译往往是根本不可能的。机器翻译虽然做不到译文的信达雅,虽然纰漏百出,但译文一般而言可以传达文献的总体思想,这样的机器翻译完全可以起到一种协助翻译用户搜寻定位他们最感兴趣内容的作用。当用户要从大量外语文献中寻找某些信息时,首先他可以把所有文献交给一个机器翻译系统进行翻译。机器翻译有着翻译专家不可比拟的两个优势,一是廉价;二是高速。然后用户再从机器译文中选择出感兴趣的文献。对于选出来的文献,他可以再聘请翻译专家重新进行翻译,或聘请专业后编辑人员润色机器译文,从而产生高质量的译文。

对于信息传播型的用户而言(这类用户大多是一些跨国公司),他们需要把所有技术资料都准确翻译,不存在任何妥协的可能性。在这种需求下,机器翻译仍然可以发挥巨大的作用,机器可以协助翻译人员产生高质量译文。这种翻译任务往往有这样的特点,以跨国公司为例,技术文献的领域往往比较狭窄,无论企业的规模多大,产品的数量也往往有限,在这种情形下,如果采用一种受限语言来撰写技术资料,也不会丧失过多的表达能力;同时,领域比较狭窄,机器译文的质量也会相对较好。即使不采用受限语言的办法,也可以采用后编辑的方式提高译文质量,而且由于译文质量较好,后编辑的工作相对好做。可见,机器翻译在专业人员的干预下,也可以应用于以信息传播为目的的翻译任务。一般而言,这类翻译任务的另一个特点也使得机器翻译更好地发挥作用,因为这类翻译任务具有相当的重复性。可以想象,一个公司不太可能总在生产全新的产品,大多数情况下只是产品的更新升级,如由于技术的改进,一种产品某些性能指标有了提高,这样,产品的技术资料并不需要完全重写和翻译,往往只是改进的部分需要重写和翻译。而目前在机器翻译中普遍使用的译文记忆存储技术可以有效地发现哪些部分需要重新翻译,而哪些可以使

用原有的译文。另外,在术语数据库的支持下,在这种机器翻译需求中,机器翻译也表现出一些翻译人员不可比拟的优势,如可以保证译文中术语的一致性,而专业人员要做到这一点往往比较困难。

从上面的分析可以看出,目前机器翻译的应用价值决不体现在它可以取代翻译专家,相反,机器翻译价值体现在,它可以在一个完整的翻译过程中的部分环节中有所贡献;或者更为广义地说,它在一个信息流动过程中找到了自己的位置。同时也可以清楚地看到,机器翻译的价值体现在,它可以带来翻译生产率的提高和翻译成本的降低。如果机器翻译不能在这两个方面体现优势,即使翻译需求再强烈,翻译任务再紧迫,机器翻译也毫无应用价值。设想一下,如果一个跨国公司的大量技术手册,如果机器翻译加上人工后编辑的速度还比不上翻译专家的翻译速度,机器翻译加上人工后编辑所需成本还高于人工翻译的成本,那么还会不会有公司采用机器翻译。可以这样说,正是机器翻译在这两个方面的出色表现,才建立了目前对机器翻译的需求。

前面已述,国际互联网的高速发展加速了人们对机器翻译的需求,这表现在互联网拉近了国与国之间的距离,人们有更多的外语文献需要吸收,人们也有更多的信息需要传播给本国之外的人们。同时,因特网和信息技术的进步也建立了一些新的翻译需求,这些翻译需求,往往是翻译专家所不能胜任的,机器翻译又一次出现在这些地方。同样,在这些新的应用场合,机器翻译也不能提供完美的翻译,但也不是一无是处。这些新的应用场合有:

(1) 联机实时翻译

这些翻译任务包括:

① 万维网上的 Web 页面的翻译

目前万维网已经成为一个巨大的信息源,每天都可以看到不同的内容出现,这加大了获取各种信息的范围。然而,因特网上的绝大部分信息不是用其母语写的,要获取这些信息,需要翻译。而且在这些情况下,不会想到去聘请一个翻译专家来帮你了解有线新闻网(cnn)的新闻内容,而是需要实时地获取一些关键信息,这时一个机器翻译系统基本可以满足需求,基本可以从一篇新闻中获取关键元素,诸如时间、地点、主要人物等等。关键是机器翻译可以不中断浏览,可以像阅读母语那样实时地工作。

② 聊天室中对话的实时翻译

目前因特网上有大量的聊天室、各种专业论坛,为了让说不同语言的人能够利用这些论坛和聊天室进行交流,必须进行某种翻译,这是一种以实时交流为目的的翻译需求。在这种情形下,一般也不可能聘请翻译专家来完成,一方面翻译人员的介入费用昂贵;另一方面更为重要的是翻译人员的介入会破坏交流的实时性,交流的双方也不希望外人介入他们具有一定隐私性的谈话。从一定角度看,这种翻译需求和口语同声传译有很多共同点,有着不同于书面语翻译的一系列特点。

(2) 服务于信息存取系统的翻译需求

对这种翻译任务的典型需求来自于近年来产生的一些信息存取系统,主要是跨语言的信息检索系统。例如搜索引擎系统,用户在使用本国语言键入其检索需求后,不只是期望得到本语种的相关文档,同时也希望得到其他语言中的有关文档,这样这种检索表达式需要被翻译成其他语言,然后交给检索系统进行检索;在检索系统完成检索后,用户也希

望检索出来的非母语文献能够翻译成母语呈现出来。除信息检索系统外,类似的信息检索系统还有信息抽取系统,用户不仅希望能抽取本语种的相关信息,还希望抽取系统对文本的语种没有限制;同样,用户也希望抽取结果以母语方式呈现。总之,在这类翻译需求中,翻译专家的介入也不太可能,因为他们的介入同样会破坏信息存取系统的自动性、实时性等。

9.5.2 机器翻译的使用

1.政府和非商业目的的使用

在机器翻译研制的早期,机器翻译系统的用户多为国家和国际的政府机构以及军队,主要原因是计算机硬件设备费用昂贵。1970 年美国空军(US airforce)开始使用著名的机器翻译系统 SYSTRAN,目的在于将俄国军事方面的科学技术文献翻译成英语。据说部分翻译文献经过后编辑处理,但大部分译文输出都未做修改直接交给用户,而且据称技术报告翻译准确率达到 90%以上。现在美国国家空军情报中心(National Air Intelligence Center)接管了美国空军的翻译服务,为美国政府多个机构提供广泛的翻译服务,并且许多翻译不做译后编辑直接提交给用户。此外翻译也不再局限于从俄语到英语的翻译,SYSTRAN 现在可以处理更为广泛的语种,不仅能处理更多的欧洲语言,例如塞尔维亚语、克罗地亚语,也能处理包括汉语、日语以及朝鲜语在内的一系列东方语种。

在欧洲,语言问题一直是一个广泛受到关注的问题。欧洲联盟(European Commission)目前有 12 个成员国,几乎每个国家都使用自己的语言,这严重妨碍了欧洲联盟进行经济政治一体化的努力。每年欧盟在翻译方面的开支巨大,这也导致欧盟成为欧洲最大和最早的机器翻译用户;同时,欧盟也成为欧洲各国机器翻译研究的最大的资助单位。1976 年,欧盟的前身欧洲经济共同体开始采用 SYSTRAN 系统把英语翻译成法语。随着时间的推移,目前欧盟几乎在其所有成员国的语言翻译中都使用了机器翻译系统。机器翻译系统的引入为欧盟节省了大量的翻译费用。当然,这并不是说,欧盟已经无需雇佣翻译人员进行翻译;相反,在欧盟,翻译目前仍然主要是由翻译人员完成,尤其是法律法规方面的翻译,但机器翻译系统也在某些场合起着越来越重要的作用。为了机器翻译能更为广泛地使用,欧盟仍然积极资助机器翻译方面的研究工作。

2.企业和商业目的的使用

也有许多例子说明,长期使用机器翻译系统能给企业以及商业机构带来效益和利润。而最有名的例子是位于加拿大新不伦瑞克的 Lexi-Tech 公司,利用 Logos 系统进行技术文档的翻译。开始是把海军护卫舰维修手册翻译成法语;目前,该公司已经建立一个服务机构负责承接各种其他的翻译任务。Logos 系统在商业界的用户还有 Ericsson 公司、Osram 公司、Océ 技术公司、SAP 公司以及 Corel 公司。SYSTRAN 系统在商界也拥有众多大的客户,如福特汽车公司、通用汽车公司、Anospatiale、Berlitz、Xerox 公司等等。西门子公司的 METAL 德英翻译系统也拥有一些欧洲客户,例如:Boehringer Ingelheim, SAP, Philip 及瑞士联合银行等。

大公司翻译任务的特点是,一般而言领域狭窄而且固定,翻译量大,技术性强,所以对术语一致性要求较高;通常要求给出多种译文。因而针对这些特点,不同的公司采用不同的策略产生高质量的译文。针对术语问题,许多大公司在引入机器翻译的同时,也逐步建

立了自己的术语数据库,并使机器翻译系统和术语数据库协同工作。为了弥补机器译文质量不好的问题,有的公司采用了受限语言的思想,有的公司采用了后编辑的方式。

Xerox 公司曾经采用过受限语的办法来完成技术文档的翻译。正如前文提到过的那样,该公司曾经制定过一种规范语言,即多国规范英语,在 Xerox 公司内部,技术文献的作者必须用多国规范英语来编写文件。这时不仅某个术语的说法确定下来,而且怎样造句也确定下来。经过这样产生的文献再交给 SYSTRAN 系统去翻译。这种做法的好处是,排除了许多机器翻译系统难以处理的歧义,因而输出译文的质量会很好,并且可以更快地同时译成其他多个语种。这些优点得到了其他跨国公司的认同。“受限语言”用得越来越多,例如 Caterpillar 公司设计了他们自己独特的英文格式,以便使用 CMU 为其开发的基于知识的机器翻译系统。并且还出现了针对公司的特点,专门为各个公司开发以受限语言为输入的机器翻译系统的公司,例如纽约的 Smart 公司。Smart 公司主要客户有: Citicorp、Chase、福特公司和通用电器公司等。在这些系统中,每个都包含一个对英文文献‘正规化’的系统。这一部分非常重要,以至于真正的翻译过程被看作为它的‘副产品’。Smart 开发的翻译系统可以把英语译为法语、德语、希腊语、意大利语、日语和西班牙语等多个语种。Smart 公司曾经为加拿大就业部设计过一个翻译就业广告等信息的翻译系统。在欧洲,荷兰的 Cap Volmac 公司和比利时的 Lant 公司,也同样提供类似 Smart 公司的服务,用他们的受限语言软件为各种类型的客户建立专业翻译系统。

由于每个公司均有自己的特点和要求,购买的商用系统往往需要经过调整定制才能完成翻译任务,因而也有一些公司和单位正开发或委托开发他们自己的机器翻译系统,而不是购买通用系统。这类系统实际上是采用前面所说的通过限定领域的办法来提高译文质量的策略。该类定制系统的一个较早的例子是 TITUS 系统,系统翻译的对象是纺织行业文档的摘要,完成英语、德语、法语和西班牙语的互译,从 20 世纪 70 年代开始,该系统就一直在有关行业得到了应用。另一个成功的例子是泛美健康组织(Pan American Health Organization)开发的翻译卫生与健康方面的文献,可以在英语和西班牙语之间进行互译。值得注意的是,系统是由该组织员工自行研制开发的。尽管目前该系统已经拥有该组织之外的用户,并且朝着通用化的方向发展,但该系统的知识库仍然有着很强的领域倾向性。90 年代在芬兰出现的 Kielikone 系统是为 Nokia 公司开发的翻译工作站。这样的例子还有很多。

商业机构翻译任务的另一个特点是有很强的重复性。在这一方面最有代表性的工作是软件本地化(software localization)。软件业要求某个软件版本发布时,最好能同时发布该软件的面向各种语言的版本。翻译必须迅速完成,但从一个版本到另一个版本有很大部分的重复,所以很多系统采用翻译记忆或存储的办法来解决这一问题。例如 SAP、Lotus、Corel 等软件公司都利用机器翻译及其翻译存储技术解决软件本地化的工作。

3 . 面向个人的使用

在 20 世纪 80 年代早期,出现了面向个人用户的机器翻译系统。在日本出现了一大批机器翻译系统,这些机器翻译系统都有面向个人用户的版本。例如: NEC 公司的 PIVOT 系统,东芝公司的 ASTRANSAC 系统,日立公司的 HICATS 系统,OKI 公司的 PENSEE 系统,以及夏普公司的 DUET 系统。这些系统大都是进行日语和英语间的翻译。

在日本以外,也出现了一批面向个人计算机的机器翻译系统,例如 80 年代早期美国

Weidner 公司开发的 ALPS 系统。ALPS 系统主要目的是进行辅助翻译,提供有建立和访问术语资源的工具,但是系统也包含一个交互式翻译模块。ALPS 系统最初销售业绩很好,但根据该公司的有关资料显示,销售并没有取得进一步的成功,实际上该公司最终不再销售该产品。此后,Weidner 开始销售全自动的机器翻译系统。Weidner 公司的系统共有两个版本,其中一个是面向个人计算机的 MicroCat 系统,系统主要提供涉及英语、法语、德语和西班牙语的翻译服务。但该公司后来为日本 Bravis 公司所收购,随后 Bravis 认为面向个人计算机的机器翻译市场并不成熟,MicroCat 进一步转手。

到 80 年代末期,今天市场上能见到的大部分系统均已出现。这其中有美国 Linguistic Products 的 PC-Translator,从销售业绩来看还是很成功的。Globalink 公司的一系列系统提供从英语到法语、德语和西班牙语以及从这些语言到英语的翻译。后来,Globalink 公司同 MicroTac 公司合并。在 90 年代早期,Globalink 公司推出了著名的 Power Translator 系列产品,后又推出 Telegraph 系列。目前,Globalink 公司本身已经被 Lernaut & Hauspie 公司所收购。

面向大型用户的机器翻译销售商也都推出了他们面向小型或个人用户的系统,例如 SYSTRAN Pro 以及 SYSTRAN Classic 都是基于 Windows 的翻译系统。针对有些语言对,这两个系统的售价甚至低于 500 美金。著名的出版公司 Langenscheidt 开始出售 METAL 系统的个人版 Langenscheidt T1。IBM 和 von Rheinbaben & Busch 公司联合推出了 Personal Translator 系统。这些系统很多都是面向非专业翻译人员的。

从销售看,这些面向个人用户的系统是成功的。据估计,目前市场上有不下 1 000 种翻译软件,仅在北美洲至少有 6 000 家商店销售 Globalink 的产品。日本一家公司 Catena 的系统 Korya Eiwa 推出第一年就卖出了 100 000 多套。当然这些销售的成功并不完全证明这些卖出的系统都真正发挥了作用,但至少可以说明机器翻译的个人需求是巨大的。

4 . 在因特网上的使用

许多机器翻译销售商提供基于网络的翻译服务,例如 SYSTRAN, Logos, Globalink, Fujitsu 和 JICST(日本科学技术信息中心)以及 NEC 等等公司和机构。用户在使用这些翻译服务时,可以选择要求后编辑,这样用户提交的文本经过机器翻译系统处理后会由后编辑人员编辑,然后再返回用户。网上翻译服务一般采用典型的客户机-服务器软件结构。除了为正式客户提供翻译服务外,这些网上翻译服务也提供试用服务,以便于潜在机器翻译客户发现适合他们使用的机器翻译系统。

除了传统机器翻译制造商,目前甚至出现了专门以提供机器翻译服务为目标的公司。在比利时,LANT 就是一个这样的公司。1997 年,它开始提供多语种翻译服务,翻译诸如电子邮件、Web 页面以及邮件附件一类的电子文本。在新加坡,国立新加坡大学系统科学研究所也成立了一个这样的部门——机器翻译服务部(Machine Translation Service Unit),翻译的语种涉及中文、英文、马来文、日文以及朝鲜文等,并且提供译后编辑服务。

另外,目前也出现了一大批专门翻译 Web 页面的翻译系统。例如 AltaVista 提供法语、德语、西班牙语网页和英语网页之间的互译。提供中文到英文网页翻译服务的有 Readworld 网站。用户对这些翻译是否满意,这些服务是否很成功,很难轻易断言,仍然需要时间的验证。

正如前面提到的那样,网络技术的高速发展不但加大了现实世界对机器翻译的需求,

同时也开创了新的机器翻译的需求。20 世纪 90 年代后期,CompuServe 在网上论坛上开始尝试为论坛用户提供翻译服务。并且也看到一些网络门户网站以及搜索引擎也提供机器翻译服务,典型的提供翻译服务的搜索引擎如 Infoseek,它利用 SYSTRAN 系统提供对检索结果的翻译。

显然,许多公司都把网上机器翻译服务视为获得利润的新起点。几乎所有公司都不甘落后,均在这个新领域有比较详细周密的商业计划。Lernout & Houspie 就是一个典型的例子。该公司雄心勃勃,购买了多家翻译公司以及翻译系统,例如 METAL, Globalink, Neocor 等等,是成是败,人们拭目以待。

9.5.3 进一步的需求和展望

尽管连篇累牍地罗列了众多的机器翻译应用,但是不能不清醒地意识到,目前机器翻译系统所能提供的服务还很有限,没有人的参与,机器翻译仍然很难给出满意的服务。尤其对于某些翻译用户和某些使用场合,机器翻译系统要想在其中扮演一定角色,还需要不懈的努力和充足的投入。

对于一个自由职业的翻译工作者而言,他的翻译特点是翻译主题多变,因而需要一个在任何领域都能工作的通用系统,显然目前这是非常困难的。一些翻译人员尝试过为个人计算机开发的翻译软件,但很少有感到满意的,因为机器翻译后需要的人工编辑工作量很大,并不能带给他们经济上和时间上的效益。但这一需求,据目前看,短时期内仍然很难有胜任的产品问世。

对于个人用户而言,有些情形机器翻译也很难发挥作用。例如,对于一个不懂英语的人,如果希望在机器翻译系统的协助下进行英语写作,目前仍然是不可能的。一些研究利用文档模板类的思想协助用户创作外语商业信函等,但对其他文体,目前并没有什么好的策略和方法。

同样,对于网络技术带动的新的翻译需求,也就是把机器翻译技术和信息访问、信息提取和文摘软件结合在一起的尝试仍处在研究阶段,目前市场上还没有十分成功的商用产品。当前人们对这类跨语言应用兴趣越来越大,最吸引人的应用之一是"跨语言信息检索(cross-language information retrieval)",即允许用户用自己的语言搜索外语数据库的软件。在跨语言信息检索中,大多数工作集中于如何建立和操作合适的翻译词典,以便将查询词串与数据库文档中的词和词组匹配,目前有的搜索引擎已经开始提供跨语言检索服务(例如,Google),相信该领域在未来数年中会有更多发展。

未来还有一个应用是公众迫切需要的,就是口语的翻译。但从商业角度(甚至从研究角度)看,全自动口语(语音)翻译还是一件十分遥远的事。在 20 世纪 80 年代以前,由于语言识别技术和语音合成技术的制约,很少有人进行口语翻译的大规模研究。80 年代,语音识别和语音合成取得了较大的进展,口语翻译的研究条件似乎走向成熟。日本、德国的一些研究人员在口语翻译方面做了一些探索。作为翻译研究的一个新领域,口语或语音翻译无疑极富创新意义。由于口语翻译中有许多有别于书面语翻译的特殊困难,期望在这一领域短期内取得较大的进展,其可能性仍然是微乎其微的。有时候,现实和研究热情往往有很大的距离。

记得在机器翻译研究诞生初期,职业翻译人员曾经是机器翻译研究的一个阻力。但

当后来这些翻译家看到众多翻译产品后,基本上都放心释然了,机器翻译根本不可能对他们的职业造成任何威胁。而且,在未来很长时间内,专业翻译人员仍然无需杞人忧天。尤其是对于文学艺术方面的翻译,将继续是机器翻译的翻译禁区。在其他领域内,机器翻译和翻译人员将可能是各司其职,或相互配合工作。

小　　结

本章介绍了自然语言处理技术的一个重要应用领域——机器翻译。从系统所采用的翻译策略来看,介绍了机器翻译系统所采用的三种方式——直接翻译法、中间语言法和转换法,分析了它们各自的特点,目前,大部分的系统都是采用转换法;从所采用的技术来看,介绍了基于规则的理性主义方法、基于统计的和基于实例的经验主义翻译方法以及这两者的混合方法,学术界围绕这两种方法有许多激烈的争论。9.4 节介绍了双语对齐,之所以单独介绍,除了它对于基于统计的机器翻译和基于实例的机器翻译都是十分关键的技术之外,双语对齐对于翻译重用十分有价值,而这已不再只是某一种方法的问题,而是对于整个机器翻译的实用化都十分重要。

参 考 文 献

1 Allen J . Natural Language Understanding . 2nd ed . Redwood City: The Benjamin/ Cummings Publishing Company, Inc ., 1994

2 Brown P F, Della Pietra V J, Peter V D, et al . Class-based n-gram models of natural language . Computational Linguistics, 1992, 18(4):467～479

3 Brown P F, Della Pietra S A , Della Pietra V J, et al . The Mathematics of Statistical Machine Translation: Parameter Estimation . Computational Linguistics, 1993, 19(2):263～311

4 Christopher D Manning, Hinrich Schütze . Foundations of Statistical Natural Language Processing . London: The MIT Press, 1999

5 Harris M D . Introduction to Natural Language Processing . Virginia, Reston Publishing Company, Inc .,1985

6 Rabiner L R . A tutorial on hidden Markov models and selected applications in speech recognition . Proc . Of the IEEE, 1989, 77(2)

7 Wierzbicka A . Semantics . Oxford: Oxford University Press, 1996

8 **费尔迪南·德·索绪尔 . 普通语言学教程 . 中译本 . 北京:商务印书馆**, 1980

9 **冯志伟 .自然语言机器翻译新论, 北京:语文出版社**, 1994

10 **林杏光, 王玲玲, 孙德金 . 现代汉语动词大词典 . 北京:北京语言学院出版社**, 1994

11 **石纯一, 黄昌宁, 王家厥 . 人工智能原理 . 北京: 清华大学出版社**, 1993

12 **姚天顺, 等 .自然语言理解 .北京:清华大学出版社**, 1995

13 **叶蜚声, 徐通锵 .语言学纲要 .北京:北京大学出版社**, 1981

14 **朱德熙 .语法讲义 .北京:商务印书馆**, 1982