

信 号 与 信 息 处 理 丛 书

计算机自然语言处理

王晓龙 关 毅 等 编著

清华大学出版社

信号与信息处理丛书

计算机自然语言处理

王晓龙 关 毅 等 编著

清华大学出版社
北京



内容简介

计算机自然语言处理技术在我国现代化及信息化建设中起着越来越重要的作用,我国政府已经将它列入“国家中长期科学技术发展纲领”。近年来,语言处理技术,特别是基于国际互联网的中文语言处理技术正在引起我国广大科技工作者的高度重视。

本书既全面阐述了中文语言处理技术的特殊规律,又借鉴了国内外学者在计算语言学领域里的最新成就,还包括了作者的实践经验和体会。

本书可以作为计算机相关专业研究生的专业课教材,也可供相关专业高年级大学生和从事自然语言处理技术研究和应用的科技人员参考。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

本书防伪标签采用清华大学核研院专有核径迹膜防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

计算机自然语言处理/王晓龙,关毅等编著. —北京:清华大学出版社,2005.4
(信号与信息处理丛书)

ISBN 7-302-10089-6

I. 计… II. ①王… ②关… III. 自然语言处理 IV. TP391

中国版本图书馆 CIP 数据核字(2004)第 130469 号

出版者:清华大学出版社

<http://www.tup.com.cn>

社总机:010-62770175

组稿编辑:陈国新

文稿编辑:马幸兆

版式设计:肖米

印刷者:北京市清华园胶印厂

装订者:三河市李旗庄少明装订厂

发行者:新华书店总店北京发行所

开本:185×260 印张:11.5 字数:260千字

版次:2005年4月第1版 2005年4月第1次印刷

书号:ISBN 7-302-10089-6/TP·1035

印数:1~3000

定价:23.00 元

地址:北京清华大学学研大厦

邮编:100084

客户服务:010-62776969



《信号与信息处理丛书》编委会

主 编 李衍达

编 委(排名不分先后)

王宏禹 张贤达 李衍达 何振亚

迟惠生 保 铮 侯朝焕 袁保宗

阎平凡 谭铁牛

责任编辑 陈国新



丛书出版说明

FOREWORD

信号与信息处理可以说是信息技术中的核心部分。随着信息科学与技术的飞速发展,随着信息技术深入到各个领域而得到广泛的应用,信号与信息处理也作为前沿技术而发生着重大的变化。编辑出版“信号与信息处理丛书”正是为了反映这种变化,为了加速培养这方面的人才,也为了进一步推动这一领域的发展。本丛书的内容力求能反映信号与信息处理技术的前沿内容,具有高的学术意义与应用价值。入选的书稿可以是创作的专著,也可以是高水平的译作。

这套丛书不仅适合于作研究生教学参考之用,也可作为高校教师与有关领域研究人员学习与工作的参考书。

从历史来看,真正影响着生活的是不断增长的知识与技术的积累和经反复探索所形成的观念。相信这套丛书的出版,会增加正在成长中的信号与信息处理技术的积累,而它对生活的作用则是显而易见的。

李衍达

2004年8月24日



前言

FOREWORD

计算机自然语言处理是用计算机通过可计算的方法对自然语言的各级语言单位(字、词、语句、篇章等)进行转换、传输、存储、分析等加工处理的科学,是一门与语言学、计算机科学、数学、心理学、信息论、声学等相联系的交叉性学科。国际互联网技术的飞速发展,极大地推动了信息处理技术的发展,也为信息处理技术不断提出新的需求。语言作为信息的载体,语言处理技术已经日益成为全球信息化和我国社会与经济发展的重要支撑性技术。

本书全面阐述了自然语言处理技术的基本原理和实用方法,全书共分为基础、原理和应用3个篇章。第1章是概论;第2、第3章构成本书的基础篇,论述了自然语言处理技术的数学基础和中文语言处理特有的自动分词技术;第4、第5章构成本书的原理篇,分别论述了基于统计和基于语言学规则的语言处理技术的基本原理;第6~9章构成本书的应用篇,论述了在音字转换、自动文摘、信息检索、手写体识别等应用领域中的实用语言处理方法。

参加本书编写工作的有王晓龙、关毅(第1章、第2章、第3章、第8章)、刘秉权(第6章)、林磊(第9章)、陈清才(第2章、第7章)、刘远超(第7章)、赵岩(第3章、第5章)、赵健(第4章、第8章)、肖镜辉(第4章),全书由王晓龙、林磊等进行了统编和审校。由于编著者水平有限,错误和疏漏在所难免,敬请读者批评指正。

作 者

2004年3月



目 录

CONTENTS

第 1 章 引言	1
第 2 章 数学基础	7
2.1 初等概率理论	7
2.1.1 基本概念	7
2.1.2 条件概率与独立	9
2.1.3 全概率公式与贝叶斯公式	10
2.1.4 随机变量	12
2.1.5 多维随机变量	13
2.1.6 数学期望与方差	15
2.1.7 常用分布	16
2.2 信息论基础	18
2.2.1 信息熵	18
2.2.2 联合熵和条件熵	20
2.2.3 互信息	20
2.2.4 相关熵	21
2.2.5 语言与熵	22
2.2.6 噪声信道模型	23
2.3 粗糙集	25
2.3.1 信息系统	25
2.3.2 不可分辨关系	25
2.3.3 集合近似	26
2.3.4 约简	27
2.3.5 属性依从	28
2.3.6 决策规则合成	29
2.4 小结	29
第 3 章 汉语自动分词技术	31
3.1 引言	31
3.2 分词规范	33

3.3 常用的分词方法.....	35
3.3.1 正向最大匹配分词	35
3.3.2 反向最大匹配分词	35
3.3.3 基于统计的词网格分词	36
3.4 歧义的分类和识别.....	36
3.4.1 歧义的分类	36
3.4.2 歧义的抽取和消歧	37
3.5 新词的识别.....	39
3.5.1 统计构词能力	40
3.5.2 汉字构词模式	40
3.5.3 未登录词识别算法	41
3.6 关于分词的若干统计结果.....	41
3.7 语言单位的统计分布规律(Zipf 定律).....	42
3.8 小结.....	44
第 4 章 基于数学统计的语言模型	47
4.1 统计语言模型概述.....	47
4.2 现有的主要统计语言模型.....	48
4.2.1 上下文无关模型	48
4.2.2 N 元文法模型	49
4.2.3 N-POS 模型	50
4.2.4 基于决策树的语言模型	51
4.2.5 动态、自适应、基于缓存的语言模型	51
4.3 数据平滑技术.....	52
4.3.1 数据平滑算法的评价标准	53
4.3.2 常见平滑方法	53
4.4 隐马尔科夫模型.....	57
4.4.1 随机过程	57
4.4.2 马尔科夫链和马尔科夫性	57
4.4.3 马尔科夫模型	58
4.4.4 隐马尔科夫模型	58
4.5 最大熵模型.....	62
4.5.1 模型介绍	62
4.5.2 模型评价	64
4.5.3 最大熵语言建模	64
4.6 小结.....	65

第 5 章 基于语言理解的处理方法	69
5.1 引言	69
5.2 常用的基于语言理解的分类标注体系	70
5.2.1 词性分类体系	70
5.2.2 词义分类体系	72
5.3 常用的基于语言理解的语法理论	74
5.3.1 常用的语法理论	75
5.3.2 浅层语法分析技术	82
5.4 语料库多级加工	84
5.4.1 语料库的多级加工	85
5.4.2 分词	86
5.4.3 词性标注	86
5.4.4 词性标注的 HMM 模型	88
5.4.5 Viterbi 词性标注算法	89
5.4.6 语法分析	90
5.4.7 概率上下文无关文法	93
5.4.8 语料库的应用	95
5.5 小结	96
第 6 章 音字转换技术	99
6.1 引言	99
6.2 声音语句输入	100
6.2.1 声声音语句输入的提出	100
6.2.2 声声音语句的推理	101
6.2.3 声声音语句输入的系统实现	102
6.3 汉字智能拼音键盘输入	103
6.4 拼音输入的多种表达形式	104
6.4.1 拼音助学和提示输入	104
6.4.2 简拼快速输入	105
6.4.3 用户自定义简拼	105
6.4.4 模糊拼音输入	105
6.4.5 面向数字键盘的数字拼音输入	105
6.5 拼音预处理	106
6.5.1 拼音流的切分	106
6.5.2 拼音纠错	108
6.6 音字转换的实现方法	109
6.6.1 基于理解的方法	109

6.6.2 基于语用统计的方法.....	109
6.6.3 基于模板匹配的方法.....	110
6.6.4 基于上下文关联的音字转换.....	110
6.7 小结	111
第7章 自动文摘技术.....	113
7.1 引言	113
7.2 文本的内部表示方法	115
7.3 基于浅层分析的文摘技术	116
7.3.1 建立特征库.....	117
7.3.2 文摘句抽取.....	119
7.4 基于实体分析的文摘技术	120
7.4.1 特征提取.....	120
7.4.2 文摘抽取.....	122
7.5 基于话语结构的文摘技术	122
7.5.1 基于词汇衔接的文摘方法.....	123
7.5.2 基于话语树的文摘方法.....	124
7.6 文摘系统评测方法	126
7.7 关键词自动抽取	127
7.8 小结	129
第8章 信息检索技术.....	131
8.1 信息检索综述	131
8.1.1 信息检索的定义与术语.....	131
8.1.2 信息检索系统.....	132
8.1.3 信息检索系统的评价.....	134
8.1.4 信息检索简史.....	136
8.2 信息检索的统计模型	137
8.2.1 基于统计的信息检索模型.....	137
8.2.2 布尔模型.....	138
8.2.3 向量空间模型.....	139
8.2.4 概率模型.....	142
8.3 信息检索中的自然语言处理方法	143
8.4 文本自动分类技术	146
8.4.1 问题的提出.....	146
8.4.2 分类预处理.....	146
8.4.3 向量空间简化方法.....	147
8.4.4 分类方法.....	149

8.5 小结	154
第9章 文字识别技术.....	157
9.1 引言	157
9.2 联机手写体汉字识别的国内外研究概况	158
9.2.1 国外研究概况.....	158
9.2.2 国内研究概况.....	159
9.3 联机手写体汉字识别方法综述	160
9.3.1 基于统计的识别方法.....	160
9.3.2 基于结构的识别方法.....	161
9.3.3 基于神经元网络的识别方法.....	162
9.3.4 基于机器学习的识别方法.....	162
9.4 典型联机手写体汉字识别系统	163
9.4.1 汉王中文手写体汉字识别系统.....	163
9.4.2 豪文中文手写体汉字识别系统.....	163
9.5 联机手写体汉字识别后处理系统	164
9.5.1 手写体汉字识别模型.....	164
9.5.2 $P(I S)$ 估计	165
9.5.3 $P(S)$ 估计	166
9.5.4 基于词网格的手写体汉字识别的语言学解码方法.....	166
9.5.5 联机手写体汉字识别后处理系统.....	167
9.6 小结	169



第1章

CHAPTER 1

引言

语言是音义结合的词汇和语法的体系,是人类最重要的交际、思维和传递信息的工具。语言随着社会的产生而产生,随着社会的发展而发展,经历了漫长而缓慢的发展过程,成为一种极其复杂的、特殊的、充满了灵活性和不确定性的社会现象。在人类逐步进入信息化社会的今天,语言文字信息的计算机自动处理水平和处理量已成为衡量一个国家是否步入信息社会的重要标准之一。我国政府一直将中文语言处理技术这一学科作为高技术产业化重点领域。从 20 世纪 80 年代开始至今,中文语言处理技术在字处理、词处理等领域均取得了重要的进展,获得了一系列实用化的成果,不仅使中文这一世界最古老的文字之一顺利地搭上了信息时代的火车,而且在文字识别、语音识别、机器翻译等语言处理技术方面与西文相比毫不逊色,在排版印刷等应用方面达到了世界领先的水平。

所谓计算机自然语言处理,是用计算机通过可计算的方法对自然语言的各级语言单位(字、词、语句、篇章等等)所进行的转换、传输、存储、分析等加工处理。本书主要讨论中文语言处理的基本理论和实用方法。这里所说的中文,广义上是指汉语和我国少数民族的语言文字,狭义上是特指汉语文字,包括以文本、图像、声音等形式存在的汉语口语和书面语。本书采用狭义的定义。中文语言处理通常是指以计算机为工具,采用可计算的方法对中文信息所进行的自动加工处理。从技术路线上,可以分为基于统计的语言处理技术和基于语言学规则的语言处理技术两大类。前者从大规模真实语料库中获得各级语言单位上的统计信息,并依据较低级语言单位上的统计信息,用相关的统计推论技术计算较高级语言单位上的统计信息;后者通过对语言学知识的形式化,形式化规则的算法化,以及算法实现等步骤将语言学知识转化为计算机可以处理的形式。按语言处理技术处理对象的不同,语言处理技术可以划分为字处理技术、词处理技术、语句处理技术、篇章处理技术等。按照语言处理技术的应用领域,语言处理技术可以划分为应用基础技术、应用技术两大类。本书主要介绍在词、语句、篇章等语言结构单位上引入语言学规则的统计语言处理方法的基本原理和应用。

自然语言处理技术是一门与语言学、计算机科学、数学、心理学、信息论、声学相联系的交叉性学科,与自然科学和社会科学的许多主要学科都有千丝万缕的联系,其中,又与语言学、计算机科学和数学的关系最为密切。在更加细微的层面上,与自然语言处理技术密切相关的学科有计算语言学、智能化人机接口、自然语言理解,等等。其中,计算语言学是现代语言学的一大分支,它是用计算机理解、生成和处理自然语言,即它的研究范围不仅涵盖语言信息的处理,还包括语言信息的理解和生成。智能化人机接口侧重于语言信息处理的应用研究,即运用语言处理技术改善人机交互的方式、手段和途径。自然语言理解则是人工智能的一个分支,其研究重点侧重于对经过深度加工处理的语言信息的理解,相当于语言处理技术在较高级语言单位上的应用基础研究。

从 20 世纪 50 年代初俄汉机器翻译系统诞生算起,中文语言处理技术的发展历史已经有 50 多年了。从 20 世纪 80 年代初期开始,随着计算机技术的普及和发展,中文语言处理技术在应用需求的推动下进入了一个快速发展的时期,在字处理、词处理、句处理、篇章处理等技术方面取得了一系列基础研究和应用研究的标志性成果。

在字处理技术方面,提出了信息处理用的汉字机内码。汉字机内码定义了汉字在计算机内部的存储方式,目前有中国国家标准 GB2312—80;港澳台地区普遍使用的台湾《通用汉字标准交换码》,其地区标准号为 CNS11643;国家信息产业部和质量技术监督局发布的《信息技术和信息交换用汉字编码字符集、基本集的扩充》,其国家标准号为 GB18030—2000;中日韩三国根据 UCS 标准共同制定的《CJK 统一汉字编码字符集》,其国际标准号为 ISO/IEC10646,我国的国家标准号为 GB13000—90。

汉字的输入码(或称汉字外码)提供了汉字输入的途径。目前主要的汉字输入码有五笔字型、拼音码等。通过建立输入码与汉字机内码的一一对应关系,使汉字输入到计算机中。

汉字字形库(或称汉字字形码、汉字发生器编码)存放如汉字的宋体、黑体、楷体等各种字体的点阵或曲线矢量字形信息,通过专门的处理程序把要输出的汉字转换成对应的汉字字形后在显示器、打印机上输出。汉字库中还包含了汉字放大、缩小、斜体、粗体等字体变化的信息。

汉字处理的应用技术主要包括汉字排版(如北大方正的激光照排系统)、印刷体汉字识别和联机手写汉字识别技术,对这些技术目前已经出台了一些标准,例如中国电子技术标准化研究所、北京汉王科技有限公司等单位起草的《联机手写汉字识别技术要求与测试规程》。

在词处理技术方面,词是自然语言中最小的有意义的构成单位,是自然语言处理中最基本的研究对象,也是其他研究的先行和基础。汉语不同于印欧语,它不把空格作为词的分隔标志,所以制定分词规范非常必要。目前有国家制定的《信息处理用现代汉语分词规范即自动分词方法》^[1](中华人民共和国国家标准 GB13715),其主导的判别标准就是“使用频繁,结合紧密”。当然,在实际应用时,还要考虑文化、语感等多种因素才能得到正确的分词结果。对此,山西大学的刘开瑛^[2]等、北京语言文化大学的宋柔^[3]等学者进行了长期的研究,取得了很多研究成果。

词处理主要包括分词、词性标注、词义消歧三项内容。常用的分词方法包括正向最大

匹配、反向最大匹配以及基于词网格的统计方法。分词的主要难点在于歧义消解和新词识别。由于汉语本身的复杂性,目前这两个问题并没有得到根本性的解决。清华大学孙茂松、黄昌宁^[4,5],东北大学姚天顺等学者对此均进行了深入研究,取得了长足的进展。词性标注常用的方法是基于隐马尔科夫模型的词性标注方法。常用的词性标注方法包括基于词典知识库的方法,还有一些常用的基于统计的分类方法,包括贝叶斯方法和最大熵模型。分词和词性标注是所有中文语言处理应用技术的基础,广泛地应用于机器翻译、信息检索等各个领域。由中国科学院刘群、白硕等开发的 ICTCLAS 分词和词性标注系统,以及由哈尔滨工业大学自然语言处理研究室开发的 ICSU 分词和词性标注系统均取得了较高的分词和词性标注精度。

在语句处理方面,汉语语句处理技术是近年来中文信息处理研究领域的一个热点,也是一个难点。它是建立在汉字编码、汉语词语切分、汉语词法分析等基础上的一项技术,同时也是汉语篇章理解的基础。汉语的语句级处理技术主要包括句法分析、语句的语义分析等内容。吴蔚天^[6]等在设计汉英机器翻译系统 Sino Trans 时提出的汉语完全语法树模型,虽然在一定程度上推动了用计算机分析汉语句法结构研究的发展,但在直接发掘汉语语言知识,揭示汉语的语言成分组合规律方面并没有多大进展。由于汉语语句处理是以语句作为研究对象,因此无论是在汉语语法研究还是汉语计算的数学模型上都存在相当大的难度,目前主要针对汉语短语展开研究,采用的方法主要是基于语法规则的方法、基于数学统计的方法,以及规则与统计相结合的方法。《现代汉语语法信息词典详解》^[7]以朱德熙^[8]先生提出的词组本位语法体系作为设置各项语法范畴的理论基础,马希文^[9]的《从计算语言学角度看语法研究》,冯志伟^[10]的《计算语言学对理论语言学的挑战》和他提出的潜在歧义结构论,白硕^[11]的《语言学知识的计算机辅助发现》,罗振声^[12]等对汉语句型的自动分析和分布统计的研究,以及北京大学计算语言学研究所的俞士汶、詹卫东^[13]在基于规则的汉语短语分析方面所做的研究工作,都代表了基于规则的汉语句法分析的主流。在汉语语句语义的理解方面,值得关注的是国内学者黄曾阳^[14]积多年研究心得,提出面向整个自然语言理解的理论框架——概念层次网络理论(HNC),对传统的基于句法知识的语言表述及处理模式提出了挑战,代之以语义表达为基础,以对汉语进行理解。近年来,随着国外对统计语言学研究的兴起,国内的学者也针对汉语的统计语言模型展开了大量的研究。清华大学的黄昌宁^[15],哈尔滨工业大学的王晓龙^[16],赵铁军^[17],微软亚洲研究院的自然语言理解小组,中国科学院计算研究所的白硕、刘群^[18]等学者和机构对汉语的统计语言模型进行了深入的研究和探讨,对汉语的 N-gram 模型、HMM 模型进行了完善,并应用到了音字转换、机器翻译、句法分析等研究方面,取得了丰硕的成果。随着对汉语语句分析技术研究的不断深入,一些系统也随之投入到了现实应用当中,其中最引人瞩目的就是哈尔滨工业大学王晓龙等学者与微软公司联合开发的微软拼音输入法(MSPY),这种语句级的输入法较以前的单词级的输入法在性能上有了很大的提高。此外,还有黑马中文自动校对系统、安徽中科大讯飞信息科技有限公司开发的 TTS(text to speech)系统、Word 中的中文自动更正功能等,都采用了汉语语句或者短语处理技术。

在篇章处理技术方面,当前对于篇章级的分析主要集中于研究一篇文章的话语结构,即进行话语结构分析。话语结构分析主要是指跨越语句本身的多个语句、段落之间在结

构或语义上的相互关系的分析。话语结构分析通常包括两种类型：一种是基于语法结构的衔接性(cohesion)分析，另一种是基于语义之间的连贯性分析(coherence)。前者主要是指在结构与形式上的衔接，而后者则指语义上的连贯性。前者讨论的范围比较趋向于文本呈现出来的表面结构如何彼此串联，例如，文本中间是如何运用适当的连接词或副词来串联句子，或者在文法层次上，句子和句子之间是如何依赖同样的主题词以及类似的句法结构来串联彼此；后者趋向于语义层面上更抽象的一致性，即文本实体之间是否基于相同主题进行讨论。在实际的自然语言处理中很难将它们区分开，因为判断语法层的连接需要通过人脑中的认知模型来完成，而对人脑中认知模型的判断又是以文本中的语法结构为基础的。也就是说，人们在判断文本实体间的语义连贯性的时候往往会展开实体之间的语法衔接关系，而作者在撰写文章的时候也往往会借助形式上的语法衔接关系来反映语义上的连贯性，所以很难将这两者区分开来。

作者在撰写文章的时候，总会赋予一篇文章一定的语法或者语义结构。通过对这种结构的分析，可以帮助我们更好地理解全文的内容，从而更准确地完成相应的自然语言处理任务。当前话语结构分析主要应用于自动文摘领域，如哈尔滨工业大学的王开铸^[19]等开发的 HIT—863 系统、王晓龙等开发的 InsunAbs^[7]系统、上海交通大学的王永成等开发的“OA”系统，都不同程度地进行了文章的话语结构分析。此外，在清华大学的罗振声、北京邮电大学的钟义信等的自动文摘研究中，话语结构分析也是一个重要的研究内容。除了应用于文摘系统，话语结构分析也可应用于信息抽取研究，如东北大学的姚天顺^[20]等关于信息抽取研究中所进行的语段层分析。

经过 20 余年的艰苦奋斗，在语言学家、计算机专家的共同努力下，中文语言处理技术从无到有，取得了非常丰硕的成果。当前，在基础研究方面，中文语言处理技术正在跨越汉语自动语法分析的难关。在应用研究方面，基于国际互联网的语言处理技术，如文本分类、信息提取、自动问答、基于内容的信息检索等正在成为新的研究热点。但是，在国际互联网技术飞速发展，中文网页信息急剧膨胀，以及中文已经成为仅次于英语的世界第二大网页信息语种的今天，中文语言处理的发展速度仍然缓慢，特别是随着各项基础研究和应用研究的纵深发展，许多长期积累的困难和问题日益突出，已经成为中文语言处理技术继续发展的障碍。这些困难和问题主要表现在：长期以来，汉语语言学研究基本上是面向汉语教学的，能够直接面向计算机的形式化语言学的研究成果数量较少，而且缺乏统一的标准；中文语言处理研究力量分散，存在着低水平重复的现象；科学公正的评测机制尚未建立起来。中文语言处理技术的发展期待着语言学家和计算机专家打破门户之见，协同一致，兼收并蓄，早日实现资源和成果共享，为进一步提高我国的语言处理水平而共同奋斗。

本书作者长期从事中文语言处理的研究工作，20 多年来，在课题组全体成员的共同努力下，在语句输入、语料库加工、自动文摘、信息检索等领域做了一些探索性的工作，本书就是对这些工作的总结。我们希望通过本书，使读者在掌握中文语言处理技术的基本原理和主要方法的同时，认识到研究这门科学的高度复杂性和艰巨性，并且希望越来越多的人加入到这一艰苦、繁琐而又充满情趣的科学的研究队伍中来。

参考文献

- 1 刘源等.信息处理用现代汉语分词规范即自动分词方法.北京:清华大学出版社,广西:广西科学技术出版社,1994
- 2 刘开瑛.现代汉语自动分词系统中几个问题的讨论.计算机开发与应用,1998
- 3 宋柔.关于分词规范的探讨.语言文字应用,1997(3)
- 4 孙茂松,黄昌宁,邹嘉彦等.利用汉字二元语法关系解决汉语自动分词中的交集型歧义.计算机研究与发展,1997,34(5):332~339
- 5 黄昌宁.中文信息处理中的分词问题.语言文字应用,1997(1):71~78
- 6 吴蔚天,罗建林.汉语计算语言学——汉语形式语法和形式分析.北京:电子工业出版社,1994:83,155~164
- 7 俞士汶.现代汉语语法信息词典详解.北京:清华大学出版社,1996
- 8 朱德熙.语法问答.北京:商务印书馆,1993
- 9 马希文.从计算语言学角度看语法研究.国外语言学,1989(3)
- 10 冯志伟.计算语言学对理论语言学的挑战.语言文字应用,1992(1)
- 11 白硕.语言学知识的计算机辅助发现.北京:科学出版社,1995
- 12 罗振声,郑碧霞.汉语句型自动分析和分布统计算法与策略的研究.中文信息学报,1994(2)
- 13 詹卫东.面向中文信息处理的现代汉语短语结构规则研究.博士论文,1999
- 14 黄曾阳.HNC(概念层次网络)理论.北京:清华大学出版社,1998
- 15 黄昌宁.关于处理大规模真实文本的谈话.语言文字应用,1993(2)
- 16 关毅,王晓龙,张凯.基于统计与规则相结合的汉语计算语言模型及其在语音识别中的应用.高技术通讯,1998,8(4):16~20
- 17 赵铁军等.机器翻译原理.哈尔滨:哈尔滨工业大学出版社,2000
- 18 刘群,詹卫东,常宝宝等.一个汉英机器翻译系统的计算模型与语言模型.见:吴泉源,钱跃良主编.智能计算机接口与应用进展.北京:电子工业出版社,1997,253~258
- 19 刘挺,王开铸.基于篇章多级依存结构的自动文摘研究.计算机研究与发展,1999,36(4)
- 20 姚天顺.自然语言理解——一种让机器懂得人类语言的研究.北京:清华大学出版社,2002



第2章

CHAPTER 2

数学基础

本章简要介绍为理解本书后续内容所必须掌握的数学基础知识,包括概率论、信息论、粗糙集等方面的基本理论。为使计算机处理中文语言,必须首先将语言信息加以形式化,使之能以严密规整的数学形式表现出来,这一过程通常称之为语言的模型化。本章描述了在语言模型化中广为应用的数学工具,包括作为基于统计的中文语言处理技术的理论基础的概率理论和信息论,以及粗糙集理论等。由于受篇幅限制,在此仅列出要点,更加详细的内容请读者参阅本章末列出的参考文献。

语言处理技术中常用的数学基础知识还包括集合论、函数与关系、微积分等,在此我们假定读者已经掌握了这些背景知识。对概率论和信息论比较熟悉的读者可以跳过本章,直接进入后续章节的学习,如果遇到难以理解的公式,再查阅本章的相关内容以作参考。我们建议读者通读本章的内容,了解全书中统一使用的数学符号的含义,以便更好地理解本书其余章节的内容。

2.1 初等概率理论

基于统计的语言处理技术已经成为语言处理技术的主流。统计语言处理的目的在于以自然语言为处理对象进行统计推导,它包括两个步骤:首先,收集自然语言词汇(或者其他语言单位)的分布情况,即统计语言单位出现的概率;其次,根据这些分布情况进行统计推导。因此,初等概率理论是统计语言处理的数学基础之一。

2.1.1 基本概念

概率论(probability theory)是研究随机现象的数学分支。所谓随机现象是指这样的一类现象,当人们观察它时,所得到的观察结果不是确定的,而是许许多多可能结果中的一种,例如,抛一枚质量均匀的硬币,可能出现正面朝上或反面朝上两种结果。随机现象的实现和对它的观察称为随机试验(random

experiment 或者 trial)。随机试验的每一可能结果称为一个基本事件(elementary event),一个或一组基本事件又通称为随机事件(random event)或简称事件(event)。事件的概率(probability)则是衡量该事件发生的可能性的量度。虽然在一次随机试验中发生某个事件是带有偶然性的,但那些可以在相同条件下大量重复的随机试验往往呈现出明显的数量规律性。人们在长期实践中已逐步觉察到某些这样的规律性,并在实践中加以应用。例如连续多次掷一枚均匀的硬币,出现正面的频率(出现次数与投掷次数之比)随着投掷次数的增加逐渐稳定于 $1/2$ 。这种在大量重复试验或观察中所呈现出的固有规律性,称之为随机现象的统计规律性(statistical rule)。

随机试验的所有可能结果或者全体基本事件构成的集合称为样本空间(sample space),记为 Ω 。随机变量(random variable) X 是定义于 Ω 上的函数,即对每一基本事件 $\omega \in \Omega$,有一数值 $X(\omega)$ 与之对应。随机变量的取值可以是离散的,也可以是连续的。在语言处理的应用中,通常可以将语言单位(如词汇)视为一个离散型的随机变量。概率的统计定义如下:

首先引入频率的概念,它描述了事件出现的频繁程度。

频率定义:若事件 A 在相同条件下进行的 n 次试验中出现了 r 次,则称

$$W_n(A) = \frac{r}{n} \quad (2-1)$$

为事件 A 在 n 次试验中出现的频率(frequency),称 r 为事件 A 在 n 次试验中出现的频数(frequence)。如果随着试验次数 n 的增大,事件 A 出现的频率 $W_n(A)$ 总在区间 $[0, 1]$ 上的某个数字 p 附近摆动,那么定义事件 A 的概率为

$$P(A) = p \quad (2-2)$$

概率的这种定义称为概率的统计定义(statistical definition of probability),由此确定的概率称为统计概率(statistical probability)。除统计概率外,还有古典概率、几何概率等多种定义方式,将这些定义方式加以综合,建立公理化体系,给出概率的数学定义如下:

定义 设随机试验的样本空间为 Ω ,如果对于每一事件 $A \subseteq \Omega$,总有一实数 $P(A)$ 与之对应,此实值函数 $P(A)$ 满足如下公理:

公理 1(非负性) 对于任一事件 A ,有

$$0 \leq P(A) \leq 1 \quad (2-3)$$

公理 2(规范性)

$$P(\Omega) = 1, P(\emptyset) = 0 \quad (2-4)$$

公理 3(完全可加性) 对于任意有限个两两互斥的事件 A_1, A_2, \dots, A_n ,有

$$P(A_1 \cup A_2 \cup \dots \cup A_n) = P(A_1) + P(A_2) + \dots + P(A_n) \quad (2-5)$$

则称 $P(A)$ 为事件 A 的概率,这个定义称为概率的公理化定义(axiomatic definition of probability)。公理 3 又称为概率的加法定理(addition formula)。

由概率的公理化定义可以得到如下推论:

$$(1) P(B - A) = P(B) - P(A \cap B)$$

$$(2) A \subseteq B \Rightarrow P(A) \leq P(B)$$

$$(3) P(\bar{A}) = 1 - P(A)$$

$$(4) P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

其中, $B - A$ 标记事件 B 与事件 A 的差事件(differential event), 表示“事件 B 出现而事件 A 不出现”;

$A \cap B$ 标记事件 A 与事件 B 的积事件(product event), 简记为 AB , 表示“事件 A 和事件 B 同时发生”;

$A \subseteq B$ 标记事件 A 是事件 B 的子事件(sub-event), 表示“事件 A 发生必然导致事件 B 发生”;

\bar{A} 标记事件 A 的逆事件(inverse event), 表示“ \bar{A} 与 A 必有一个出现, 而不能同时出现”, 即 $\bar{A} = \Omega - A$;

$A \cup B$ 标记事件 A 与事件 B 的和事件(additive event), 表示“事件 A 和事件 B 至少有一个出现”。

例 1 现代汉语字频统计。该统计由北京航空航天大学和国家语言文字工作委员会于 1985 年完成。从 1977 年至 1982 年间社会科学和自然科学的规模为 1108 万余字的语料中利用计算机进行统计得到汉字的字频, 前 20 个最高频汉字如表 2-1 所示。

表 2-1 现代汉语字频统计结果: 前 20 个最高频汉字及其频率

汉字	频率	汉字	频率	汉字	频率	汉字	频率
的	0.040 855	了	0.008 470	中	0.006 012	国	0.005 406
一	0.013 994	有	0.008 356	大	0.005 857	我	0.005 172
是	0.011 758	和	0.007 297	为	0.005 720	以	0.005 117
在	0.010 175	人	0.006 821	上	0.005 705	要	0.004 824
不	0.009 034	这	0.006 557	个	0.005 488	他	0.004 685

2.1.2 条件概率与独立

在实际问题中, 除了要知道事件 A 的概率 $P(A)$ 外, 往往还要知道在事件 B 出现条件下事件 A 出现的概率, 称这种概率为事件 B 出现条件下事件 A 的条件概率(conditional probability), 记为 $P(A|B)$ 。当预测一个事件的出现时, 如果已经具备一些关于该事件的信息或者知识, 就可以使用条件概率来反映这种情况。不考虑先决条件(信息或者知识)而得到的该事件的概率, 通常称为该事件的先验概率(prior probability)。在具备该事件出现的信息或者知识的条件下得到的该事件的概率, 通常称为该事件的后验概率(posterior probability)。例如, 当预测“大学”一词出现的概率时, 如果已经知道出现在它前面的两个词是“哈尔滨”和“工业”, “大学”一词出现的概率就大大增加了。在事件 B 已知出现的条件下, 事件 A 出现的条件概率记为 $P(A|B)$, 由下式计算:

$$P(A | B) = \frac{P(A \cap B)}{P(B)} \quad (2-6)$$

在公式(2-6)中, 当 $B = \Omega$ 时, $P(A\Omega) = P(A)$, $P(\Omega) = 1$, 于是, $P(A|\Omega) = P(A)$, 也即条件概率转换为无条件概率。可见, 可以把一般概率看作是必然事件出现条件下的条件概率。

由公式(2-6)立即可得

$$P(AB) = P(B)P(A|B) = P(A)P(B|A) \quad (2-7)$$

公式(2-7)称为概率的乘法定理(production rule)。乘法定理易于推广到有限多个事件联合出现的情形,设有 n 个事件 A_1, A_2, \dots, A_n ,则有

$$P(A_1 \cap A_2 \cap \dots \cap A_n) = P(A_1)P(A_2 | A_1)P(A_3 | A_1 \cap A_2) \dots P(A_n | \bigcap_{i=1}^{n-1} A_i) \quad (2-8)$$

公式(2-8)又称为链规则(chain rule),在统计语言处理技术中有着广泛的应用,是构造统计语言模型的理论基础之一。

两个事件 A 和 B 是独立(independent)的,当且仅当

$$P(A \cap B) = P(A)P(B) \quad (2-9)$$

否则,称事件 A 和 B 是依赖(dependent)的。类似地,称事件 A 和事件 B 在事件 C 发生的条件下相互独立,当且仅当

$$P(A \cap B | C) = P(A | C)P(B | C) \quad (2-10)$$

2.1.3 全概率公式与贝叶斯公式

在计算某复杂事件出现的概率时,如果直接计算事件的概率比较困难,可以将该事件划分为若干彼此独立的简单事件之和,再应用概率的加法定理和乘法定理计算该事件的概率。

定义 满足如下条件的一组事件 B_1, B_2, \dots, B_n ,称为样本空间 Ω 的一个划分(partition):

$$(1) B_i B_j = \emptyset, i \neq j; i, j = 1, 2, \dots, n$$

$$(2) B_1 \cup B_2 \cup \dots \cup B_n = \Omega$$

定理 如果 B_1, B_2, \dots, B_n 构成样本空间 Ω 的一个划分,且 $P(B_i) > 0 (i = 1, 2, \dots, n)$, 则对任一事件 A ,有

$$\begin{aligned} P(A) &= P(B_1)P(A | B_1) + P(B_2)P(A | B_2) + \dots + P(B_n)P(A | B_n) \\ &= \sum_{i=1}^n P(B_i) \cdot P(A | B_i) \end{aligned} \quad (2-11)$$

公式(2-11)称为全概率公式(breakdown law)。

与全概率公式密切相关的另一个重要公式称为贝叶斯公式(Bayesian formula),它在统计语言处理中占据着举足轻重的地位(在后续章节中将经常看到它的应用)。当直接计算条件概率 $P(A|B)$ 比较困难,而 $P(B|A)$ 已知或者容易计算时,可以用贝叶斯公式通过 $P(B|A)$ 来计算 $P(A|B)$ 。

由条件概率公式(2-7)得

$$P(B | A) = \frac{P(AB)}{P(A)} = \frac{P(A | B)P(B)}{P(A)} \quad (2-12)$$

公式(2-12)就是贝叶斯公式。显然,

$$\arg \max_B P(B | A) = \arg \max_B \frac{P(A | B)P(B)}{P(A)} = \arg \max_B P(A | B)P(B) \quad (2-13)$$

在公式(2-13)中, $\arg \max_x f(x)$ 表示使 $f(x)$ 取最大值的 x , 公式(2-13)是贝叶斯公式在统计语言处理中主要的应用形式之一。贝叶斯公式更为一般和严格的说明由贝叶斯定理(Bayes theorem)给出。

贝叶斯定理 如果存在一组事件 B_i 是事件 A 的划分, 换句话说, 事件 A 能且只能与两两不相交的事件 B_1, B_2, \dots, B_n 之一同时发生, 即 $A \subseteq B_1 \cup B_2 \cup \dots \cup B_n$, 且当 $i \neq j$ 时,

$B_i \cap B_j = \emptyset$, 那么, 由全概率公式(2-11), $P(A) = \sum_{i=1}^n P(A | B_i)P(B_i)$, 于是,

$$P(B_j | A) = \frac{P(A | B_j)P(B_j)}{\sum_{i=1}^n P(A | B_i)P(B_i)} \quad (2-14)$$

例 2 将 A, B, C 3 个字符之一输入信道, 输出为原字符的概率为 a , 而输出为其他字符的概率都是 $(1-a)/2$, 现将字符串 AAAA, BBBB, CCCC 之一输入信道, 如图 2-1 所示。输入 AAAA, BBBB, CCCC 的概率分别是 p_1, p_2, p_3 ($p_1 + p_2 + p_3 = 1$), 已知输出为 ABCA, 问输入为 AAAA 的概率是多少? (设信道传输每个字母的工作是独立的)



图 2-1 噪声信道示意图

解 (请注意, 下述推导中 A, B, C 是指字符, 而非事件。 $P(I=AAAA | O=ABCA)$ 是指已知输出为 ABCA 时, 输入为 AAAA 的概率; $P(O=ABCA | I=AAAA)$ 是指已知输入为 AAAA 时, 输出为 ABCA 的概率; $P(AAAA)$ 是指输入为 AAAA 的概率; $P(B|A)$ 是指输入为 A 时, 输出为 B 的概率。)

根据贝叶斯公式,

$$P(I=AAAA | O=ABCA) =$$

$$\frac{P(O=ABCA | I=AAAA)P(AAAA)}{P(O=ABCA | I=AAAA)P(AAAA) + P(O=ABCA | I=BBBB)P(BBBB) + P(O=ABCA | I=CCCC)P(CCCC)}$$

由于信道传输每个字符的工作是独立的, 因此,

$$\begin{aligned} P(O=ABCA | I=AAAA) &= P(A | A)P(B | A)P(C | A)P(A | A) \\ &= a \cdot \frac{(1-a)}{2} \cdot \frac{(1-a)}{2} \cdot a \end{aligned}$$

类似地有

$$P(O=ABCA | I=BBBB) = \frac{(1-a)}{2} \cdot a \cdot \frac{(1-a)}{2} \cdot \frac{(1-a)}{2}$$

$$P(O=ABCA | I=CCCC) = \frac{(1-a)}{2} \cdot \frac{(1-a)}{2} \cdot a \cdot \frac{(1-a)}{2}$$

已知 $P(AAAA) = p_1, P(BBBB) = p_2, P(CCCC) = p_3$, 且 $p_1 + p_2 + p_3 = 1$, 于是,

$$P(I = AAAA \mid O = ABCA)$$

$$\begin{aligned} &= \frac{a \cdot \frac{(1-a)}{2} \cdot \frac{(1-a)}{2} \cdot a \cdot p_1}{a \cdot \frac{(1-a)}{2} \cdot \frac{(1-a)}{2} \cdot a \cdot p_1 + \frac{(1-a)}{2} \cdot a \cdot \frac{(1-a)}{2} \cdot p_2 + \frac{(1-a)}{2} \cdot \frac{(1-a)}{2} \cdot a \cdot \frac{(1-a)}{2} \cdot p_3} \\ &= \frac{2ap_1}{a(2p_1 - p_2 - p_3) + p_2 + p_3} \\ &= \frac{2ap_1}{(3a-1)p_1 + 1-a} \end{aligned}$$

在应用贝叶斯定理时,首先需要确定先验概率 $P(B_i)$ 。在通常情况下,可根据贝叶斯假设来确定。

贝叶斯假设 如果没有任何以往的知识来帮助确定 $P(B_i)$,可以采用均匀分布来作为其分布。这也是贝叶斯方法的缺陷之一。

2.1.4 随机变量

许多随机试验的结果与数值有关,要么可以直接用数值来描述,要么可以转换为等价的数值形式。一般情况下,对于样本空间 Ω 中的每一个样本点 e ,都对应着一个实数 $\xi(e)$,这个随着实验结果而变化的实数 $\xi(e)$ 通常称之为随机变量(random variable, RV)。

定义 设 E 是随机试验,它的样本空间是 $\Omega = \{e\}$,如果对于每一个 $e \in \Omega$,都有一个实数 $\xi(e)$ 与之对应,就得到一个定义在 Ω 上的单值实值函数 $\xi = \xi(e)$,称之为随机变量。

随机变量通常用大写英文字母 X, Y, Z 或者希腊字母 ξ, η, ζ 表示。

随机变量根据其取值不同分为离散型随机变量(discrete random variable)和连续型随机变量(continuous random variable)。

定义 如果随机变量 ξ 只能取有限个值或可列个值,则称它为离散型随机变量。

设离散型随机变量 ξ 的所有可能取值为 $x_k (k=1, 2, \dots)$, ξ 取各个可能值的概率为

$$P(\xi = x_k) = p_k, k = 1, 2, \dots \quad (2-15)$$

由概率的定义, p_k 满足如下两个条件:

$$(1) p_k \geq 0, k = 1, 2, \dots$$

$$(2) \sum_{k=1}^{\infty} p_k = 1$$

称(2-15)为离散型随机变量 ξ 的概率分布(probability distribution),或者分布律(distribution law)。分布律可用表 2-2 的形式来表示。

表 2-2 离散型随机变量的分布律

ξ	x_1	x_2	x_3	...	x_i	...
p_k	p_1	p_2	p_3	...	p_i	...

还有一类随机变量不是离散型的,即其可能的取值不能一一列举出来,为给出它的定义,首先给出分布函数的定义,如下:

定义 设 ξ 为随机变量, x 是任意实数,则函数 $F(x) = P(\xi < x)$ 称为 ξ 的分布函数

(distribution function)。对于任意的实数 $x_1, x_2, x_1 < x_2$, 有

$$P(x_1 \leq \xi < x_2) = P(\xi < x_2) - P(\xi < x_1) = F(x_2) - F(x_1) \quad (2-16)$$

因此,若已知 ξ 的分布函数,就可知道 ξ 落在任意区间 $[x_1, x_2]$ 上的概率。分布函数完整地描述了随机变量的统计规律性。

连续型随机变量可由概率分布函数定义如下:

定义 设 $F(x)$ 是随机变量 ξ 的概率分布函数,若存在非负函数 $f(x)$,使对任意实数 x 有

$$F(x) = \int_{-\infty}^x f(t) dt \quad (2-17)$$

则称 ξ 为连续型随机变量,其中函数 $f(x)$ 称为 ξ 的概率密度函数 (probability density function, PDF 或者 probability mass function, PMF), 简称为概率密度 (probability density)。

比较一下离散型随机变量的概率分布和连续型随机变量的概率密度函数,可以帮助我们理解概念。

对于离散型随机变量, $f(x) = P(\xi = x)$, 其中 $f(x)$ 是频率函数;

对于连续型随机变量, $f(x) = F'(x) = \frac{d}{dx}F(x)$, $f(x)$ 是概率密度函数。

对于离散型随机变量, $P(\xi \leq x) = \sum_{x_i \leq x} f(x_i)$;

对于连续型随机变量, $P(\xi \leq x) = F(x) = \int_{-\infty}^x f(t) dt$ 。

对于离散型随机变量, $P(\Omega) = \sum_{x \in \Omega} f(x) = 1$;

对于连续型随机变量, $P(\Omega) = \int_{-\infty}^{+\infty} f(x) dx = 1$ 。

因此,也可以将离散型随机变量的概率分布和连续型随机变量的概率密度函数统称为概率密度函数。

随机变量是离散型随机变量或者连续型随机变量的充要条件是:

(1) 随机变量 ξ 是离散型随机变量,当且仅当 Ω 是有限集合或可数集合。

(2) 随机变量 ξ 是连续型随机变量,当且仅当其分布函数 F 连续且除有限个点之外处处连续可导。

离散型随机变量 X 具有概率函数 $f(x)$ 或者连续型随机变量具有概率密度函数 $f(x)$, 记为 $X \sim f(x)$ 。

2.1.5 多维随机变量

随机试验的结果经常需要同时用两个或者两个以上的随机变量来描述,由若干个随机变量组成的有序数组称为多维随机变量。本小节介绍多维随机变量的有关概念和定理,为简明起见,在此仅以二维随机变量为例,多维随机变量的情形易于由二维随机变量推知。

定义 设 E 是一个随机试验, 它的样本空间 $\Omega = \{e\}$, $\xi = \xi(e)$ 和 $\eta = \eta(e)$ 是定义在 Ω 上的随机变量, 由它们构成的有序数组 (ξ, η) 称为二维随机变量 (two-dimensional random variable)。二维随机变量 (ξ, η) 分布函数的定义如下:

定义 设 (ξ, η) 是二维随机变量, 对于任意实数 x 和 y , 二元函数 $F(x, y) = P((\xi < x) \cap (\eta < y)) = P(\xi < x, \eta < y)$ 称为二维随机变量 (ξ, η) 的分布函数, 或称为随机变量 ξ 和 η 的联合分布函数 (joint distribution function)。

与前面讨论过的一维随机变量类似, 二维随机变量也分为二维离散型随机变量 (two-dimensional discrete random variable) 和二维连续型随机变量 (two-dimensional continuous random variable)。

定义 若二维随机变量 (ξ, η) 的所有可能取值为有限个或可数的多个数组, 则称 (ξ, η) 为二维离散型随机变量。

设二维离散型随机变量 (ξ, η) 的所有可能取值为 $(x_i, y_j), i, j = 1, 2, \dots$, 记 $P(\xi = x_i, \eta = y_j) = p_{ij}, i, j = 1, 2, \dots$, 则由概率的定义, 有

$$\sum_{i=1}^{\infty} \sum_{j=1}^{\infty} p_{ij} = 1, \quad 0 \leq p_{ij} \leq 1$$

称 $P(\xi = x_i, \eta = y_j) = p_{ij}, i, j = 1, 2, \dots$ 为二维离散型随机变量 (ξ, η) 的概率分布或者分布律, 或称随机变量 ξ 和 η 的联合分布律 (joint distribution law)。

与一维连续型随机变量类似, 对于二维随机变量 (ξ, η) 的分布函数 $F(x, y)$, 如果存在非负函数 $f(x, y)$, 使得

$$F(x, y) = \iint_D f(x, y) d\sigma = \int_{-\infty}^x \int_{-\infty}^y f(x, y) dx dy \quad (2-18)$$

则称 (ξ, η) 为二维连续型随机变量, 函数 $f(x, y)$ 称为二维随机变量 (ξ, η) 的分布密度或称为随机变量 ξ 和 η 的联合分布密度。

二维随机变量 (ξ, η) 作为一个整体, 具有分布函数 $F(x, y)$, 而 ξ 和 η 都是随机变量, 也都有各自的分布函数, 将它们分别记为 $F_{\xi}(x)$ 和 $F_{\eta}(y)$, 分别称为二维随机变量 (ξ, η) 关于 ξ 和关于 η 的边缘分布函数 (marginal distribution function)。

对于二维离散型随机变量, 设二维离散型随机变量 (ξ, η) 的分布律为 $P(\xi = x_i, \eta = y_j) = p_{ij}$, 则二维离散型随机变量 (ξ, η) 关于 ξ 的边缘分布律为

$$P(\xi = x_i) = \sum_{j=1}^{\infty} p_{ij} \quad (2-19)$$

同理, 二维离散型随机变量 (ξ, η) 关于 η 的边缘分布律为

$$P(\eta = y_j) = \sum_{i=1}^{\infty} p_{ij} \quad (2-20)$$

由边缘分布函数的定义, 离散型随机变量的边缘分布函数为

$$F_{\xi}(x) = P(\xi < x, \eta < +\infty) = \sum_{x_i < x} \sum_{j=1}^{\infty} p_{ij} \quad (2-21)$$

$$F_{\eta}(y) = P(\xi < +\infty, \eta < y) = \sum_{x=1}^{+\infty} \sum_{y_j < y} p_{ij} \quad (2-22)$$

对于二维连续型随机变量,设二维连续型随机变量 (ξ, η) 的分布密度为 $f(x, y)$,则

$$F_{\xi}(x) = P(\xi < x, \eta < +\infty) = \int_{-\infty}^x \left[\int_{-\infty}^{+\infty} f(x, y) dy \right] dx \quad (2-23)$$

$$F_{\eta}(y) = P(\xi < +\infty, \eta < y) = \int_{-\infty}^y \left[\int_{-\infty}^{+\infty} f(x, y) dx \right] dy \quad (2-24)$$

并记 $f_{\xi}(x) = \int_{-\infty}^{+\infty} f(x, y) dy, f_{\eta}(y) = \int_{-\infty}^{+\infty} f(x, y) dx$ 为二维随机变量 (ξ, η) 的边缘分布密度。

定义 设 $F(x, y)$ 及 $F_{\xi}(x), F_{\eta}(y)$ 分别是二维随机变量 (ξ, η) 的分布函数及边缘分布函数。若对于所有的 x, y 有 $P(\xi < x, \eta < y) = P(\xi < x) \cdot P(\eta < y)$,即

$$F(x, y) = F_{\xi}(x) \cdot F_{\eta}(y) \quad (2-25)$$

则称随机变量 ξ 和 η 是相互独立的。

对于离散型随机变量 (ξ, η) ,有

$$P(\xi = x_i, \eta = y_j) = P(\xi = x_i)P(\eta = y_j) = p_i p_j \quad (2-26)$$

对于连续型随机变量 (ξ, η) ,在分布密度 $f(x, y)$ 的连续点处有

$$f(x, y) = f_{\xi}(x) \cdot f_{\eta}(y) \quad (2-27)$$

2.1.6 数学期望与方差

数学期望(expectation)和方差(variance)是随机变量的两个重要的数值特征,数学期望反映了随机变量的平均取值,方差则反映了随机变量取值的分散程度。

定义 设离散型随机变量 ξ 的分布律为

$$P(\xi = x_k) = p_k, k = 1, 2, \dots$$

若级数 $\sum_{k=1}^{\infty} x_k p_k$ 绝对收敛,则称级数 $\sum_{k=1}^{\infty} x_k p_k$ 为随机变量 ξ 的数学期望,记为 $E(\xi)$ 或 μ ,即

$$E(\xi) = \sum_{k=1}^{\infty} x_k p_k \quad (2-28)$$

设连续型随机变量 ξ 的概率密度为 $f(x)$,若积分 $\int_{-\infty}^{+\infty} xf(x) dx$ 绝对收敛,则称 $\int_{-\infty}^{+\infty} xf(x) dx$ 的值为随机变量 ξ 的数学期望,记为 $E(\xi)$ 或者 μ ,即

$$E(\xi) = \int_{-\infty}^{+\infty} xf(x) dx \quad (2-29)$$

数学期望简称期望,又称均值(mean),可以将数学期望视为随机变量取值的,以取这些值的频率为权重的加权平均。

对随机变量 ξ 的函数 $\eta = g(\xi)$ 的数学期望 $E(\eta)$,不必知道 η 的分布,只需知道 ξ 的分布即可。因为有如下定理:

定理 设 η 是随机变量 ξ 的函数: $\eta = g(\xi)$,且 g 是连续函数。

(1) ξ 是离散型随机变量, 其分布律为 $p_k = p(\xi=x_k), k=1, 2, \dots$, 若级数 $\sum_{k=1}^{\infty} g(x_k)p_k$ 绝对收敛, 则

$$E(\eta) = E[g(\xi)] = \sum_{k=1}^{\infty} g(x_k)p_k \quad (2-30)$$

(2) ξ 是连续型随机变量, 它的概率密度为 $f(x)$, 若 $\int_{-\infty}^{+\infty} g(x)f(x)dx$ 绝对收敛, 则

$$E(\eta) = E[g(\eta)] = \int_{-\infty}^{+\infty} g(x)f(x)dx \quad (2-31)$$

随机变量的数学期望(假设数学期望存在)的性质如下:

- (1) 设 C 是常数, 则有 $E(C)=C$;
- (2) 设 ξ 是随机变量, C 是常数, 则 $E(C\xi)=CE(\xi)$;
- (3) 设 ξ, η 是随机变量, 则有 $E(\xi+\eta)=E(\xi)+E(\eta)$;
- (4) 设 ξ, η 是相对独立的随机变量, 则有 $E(\xi, \eta)=E(\xi) \cdot E(\eta)$ 。

方差是衡量一个随机变量的取值离散程度的数值特征, 它的定义如下:

定义 设 ξ 是一个随机变量, 若 $E(\xi-E\xi)^2$ 存在, 则称它为随机变量 ξ 的方差, 并记为 $D(\xi)$ 或者 $\text{var}(\xi)$ 或者 σ^2 , 即

$$D(\xi) = E[\xi - E(\xi)]^2 \quad (2-32)$$

对于离散型随机变量,

$$D(\xi) = \sum_{x \in \Omega} (x - \mu)^2 \cdot f(x)$$

对于连续型随机变量,

$$D(\xi) = \int_{-\infty}^{+\infty} (x - \mu)^2 f(x) dx$$

方差的平方根称为标准差 (standard deviation), 是另一个经常使用的随机变量的数值特征, 记为 σ 。

方差与数学期望存在如下关系:

$$D(\xi) = E(\xi^2) - (E\xi)^2 \quad (2-33)$$

方差的一些性质如下:

- (1) 设 C 是常数, 则 $D(C)=0$ 。
- (2) 设 ξ 是随机变量, C 是常数, 则有 $D(C\xi)=C^2 D(\xi)$ 。
- (3) 设 ξ, η 是两个相互独立的随机变量, 则有 $D(\xi+\eta)=D(\xi)+D(\eta)$ 。
- (4) $D(\xi)=0$ 的充要条件是 $P(\xi=C)=1$ 。

2.1.7 常用分布

本小节介绍在计算语言学研究中经常使用的概率分布。当构造语言的概率模型时, 如果已知某种语言现象服从某种特定的概率分布, 则只需确定该概率分布的几个数字特征(如期望、方差等), 就能完整地描述这类语言现象。在计算语言学研究中常用的分布包

括二项分布(binomial distribution)、泊松分布(Poisson distribution)、正态分布(normal distribution)等。

1. 二项分布

离散型随机变量 ξ 只可能取两个值, 即当事件 A 不出现时, $\xi=0$; 当事件 A 出现时, $\xi=1$, 且 $P(\xi=1)=p$, $P(\xi=0)=1-p$, $0 < p < 1$ 。这种只有两个可能结果的随机试验称为伯努利试验(Bernoulli trials)。

重复进行 n 次独立的伯努利试验, 这里“重复”是指在每次试验中事件 $A(\bar{A})$ 出现的概率保持不变, 这种试验称为 n 重伯努利试验。

在 n 重伯努利试验中, 设事件 A 出现的概率为 p ($0 < p < 1$), 以 ξ 表示 n 次试验中事件 A 出现的次数, ξ 的可能取值为 $0, 1, 2, \dots, n$ 。令 $P(\xi=k)=P_k(n, p)$, 则

$$P_k(n, p) = \binom{n}{k} p^k (1-p)^{n-k}, \quad k = 0, 1, \dots, n \quad (2-34)$$

其中, $\binom{n}{k} = \frac{n!}{(n-k)! k!}$, $0 \leq k \leq n$ 。称 ξ 服从参数为 n 和 p 的二项分布, 记为 $\xi \sim B(n, p)$ 。

二项分布的数学期望是 np , 方差是 $np(1-p)$ 。

二项分布在统计语言处理中应用广泛, 例如, 在英语语料库中, 包含单词“the”的语句占语料库中语句总数的比例近似地服从二项分布; 另外, 某英语动词在英语语料库中作为及物动词的出现也近似地服从二项分布等。

2. 泊松分布

离散型随机变量 ξ 的取值范围为 $k=0, 1, 2, \dots$, 且

$$p_k = P(\xi = k) = \frac{\lambda^k}{k!} e^{-\lambda} \quad (2-35)$$

其中, $k=0, 1, 2, \dots, \lambda > 0$ 。称随机变量 ξ 为服从参数 λ 的泊松分布, 记为 $\xi \sim P(\lambda)$ 。

泊松分布的数学期望是 λ , 方差是 λ 。

在某一固定大小的范围(或者时间段)内, 某种特定类型事件的分布, 例如, 在一个篇幅内出现的打字错误, 在一页内的某个词的分布等都近似地服从泊松分布。

泊松分布与二项分布的关系由泊松定理给出。

泊松定理 设有一列二项分布 $\{B(n, p_n)\}$, 其中参数列 $\{p_n\}$ 满足 $\lim_{n \rightarrow \infty} np_n = \lambda > 0$, 则对任意非负整数 k , 有

$$\lim_{n \rightarrow \infty} p_k(n, p_n) = e^{-\lambda} \frac{\lambda^k}{k!}$$

3. 正态分布

连续型随机变量最重要的概率分布是正态分布, 又称高斯分布(Gaussian distribution)。

连续型随机变量具有概率密度函数

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (2-36)$$

其中, $\sigma > 0$, μ 为常数。称连续型随机变量为服从参数 μ 和 σ 的正态分布, 记为 $\xi \sim N(\mu, \sigma^2)$ 。特别当 $\mu=0, \sigma=1$ 时, 称为标准正态分布 (standard normal distribution), 记为 $\xi \sim N(0, 1)$ 。

正态分布的数学期望是 μ , 方差是 σ^2 。

正态分布是自然界中最常见的一种分布, 例如, 人的身高、体重等随机变量都近似地服从正态分布, 汉字的笔画数与该笔画所对应的汉字的个数符合正态分布, 等等。在语言处理技术中, 正态分布经常用在文本自动分类技术中。另外, 其他的许多分布 (例如二项分布) 可以用正态分布来近似。特别是在数理统计中, 某些用于推断的统计量都是在正态分布条件下导出的, 这就是正态分布在概率论和数理统计中占据重要位置的原因。

2.2 信息论基础

信息论是由香农 (C. E. Shannon, 1916—2001) 于 20 世纪 40 年代末创立的关于信息通信的数学理论。40 年代末, 随着通信技术的飞速发展, 人们迫切地需要信号传输的理论模型。香农研究了通过一条噪声信道 (例如一条电话线) 的输出信号来恢复输入信号的问题, 在 1948 年发表了著名的论文《通信的数学理论》, 宣告了信息论的诞生。在这篇论文中, 他选择概率论作为数学工具, 提出了用“不确定性的量度”来计算信息量的数学公式, 为信息论奠定了理论基础。在香农为信息量确定名称的时候, 数学家冯·诺伊曼 (J. L. von Neumann, 1903—1957) 建议称为熵 (entropy), 理由是不确定性函数在统计力学中已经用在熵下面了, 香农接受了这个建议。在热力学中, 熵是物质系统状态的一个函数, 它表示微观粒子之间无规则的排列程度, 即表示系统混乱的程度, 维纳 (N. Wiener, 1894—1964) 说: “信息量的概念非常自然地从属于统计学的一个古典概念——熵。正如一个系统中的信息量是它的组织化程度的度量, 一个系统的熵就是它的无组织程度的度量; 信息量正好是熵的负数”。由于信息论中的熵与热力学中的熵无关, 所以为加以区分, 我国将信息领域里的不确定性的量度称为信息熵。

语言是人类最重要的交际工具, 也是社会上传递信息的工具, 是信息的载体。因此, 语言处理技术中的许多应用, 都大量地运用或借鉴了信息论的基本理论与量度。信息论是语言处理的又一数学基础之一。

2.2.1 信息熵

对于信息接收者而言, 信息通信的目的在于获取信息, 也就是说, 收信人事先对通信的内容并不确知, 而通过得到信息消除了这种不确定性。因此, 不确定的程度是信息量的一个量度。不确定性越大, 信息量就越大; 反之则越小。

抛一枚硬币, 或者掷一个骰子, 结果都具有不确定性, 但它们的不确定性各不相同。抛一枚硬币的结果只有 2 个, 其不确定性比可能有 6 个结果的掷骰子要小。用 H 表示一次随机实验的不确定性, H 与随机实验结果的个数有关, 换句话说, 是随机实验结果的个数的一个函数, 于是有 $H = f(k)$, k 表示随机实验可能的结果数。对于抛硬币和掷骰子, 它们的不确定性分别是:

$$H(\text{抛一次硬币}) = f(2)$$

$$H(\text{掷一次骰子}) = f(6)$$

再考虑随机实验“抛一枚硬币同时掷一次骰子”的不确定性。其可能的结果共有 12 种，因此，

$$H(\text{抛一枚硬币同时掷一次骰子}) = f(12)$$

如果规定“抛一枚硬币同时掷一次骰子”的不确定性分别是抛一次硬币的不确定性与掷一次骰子的不确定性之和，即

$$H(\text{抛一枚硬币同时掷一次骰子}) = H(\text{抛一枚硬币}) + H(\text{掷一次骰子})$$

于是有

$$f(12) = f(2) + f(6)$$

最后，如果某随机实验结果惟一，那么它就没有不确定性了，即 $f(1)=0$ 。

可以证明，满足上述条件的函数只能是对数函数 \log ，即不确定性 H 与随机实验的可能结果的个数 k 满足如下关系：

$$H = \log k \quad (2-37)$$

在信息处理领域，公式(2-37)中对数的底通常取 2，相应的计量单位称为比特(bit)，以 10 为底的对数称为哈特利(hartley)，以 e 为底的对数称为纳特(nat)。掷骰子的各种结果概率都相同，而在信息通信中，并不是每个字出现的概率都相同，而是具有特定的概率分布，令 $p(x)$ 为一个在有限字符集 Ω 上取值的随机变量 X 的概率密度函数：

$$p(x) = P(X = x), \quad x \in \Omega$$

随机变量的信息熵是该随机变量的平均不确定性程度：

$$H(p) = H(X) = - \sum_{x \in \Omega} p(x) \text{pb} p(x) = \sum_{x \in \Omega} p(x) \text{pb} \frac{1}{p(x)} \quad (2-38)$$

根据数学期望的定义，信息熵的另一种解释是函数 $\text{pb} \frac{1}{p(x)}$ 的数学期望，即

$$H(X) = E\left(\text{pb} \frac{1}{p(x)}\right) \quad (2-39)$$

例 3 考察英语中特定字母出现的频率。

当观察字母的个数较少时，频率有较大幅度的随机波动，但当观察数目增大时，频率即呈现出稳定性，有人统计了 438 023 个字母，得到如表 2-3 所示的数据。

表 2-3 特定英语字母的出现频率

字母	频率	字母	频率	字母	频率
E	0.1268	L	0.0394	P	0.0186
T	0.0978	D	0.0389	B	0.0156
A	0.0788	U	0.0280	V	0.0102
O	0.0776	C	0.0268	K	0.0060
I	0.0707	F	0.0256	X	0.0016
N	0.0706	M	0.0244	J	0.0010
S	0.0634	W	0.0214	Q	0.0009
R	0.0594	Y	0.0202	Z	0.0006
H	0.0573	G	0.0187		

根据表 2-3 的数据,由公式(2-38)计算可知,每收到一个英文信号消除信息接收者的信息的不确定程度 H 是 4.1606bit。中文有 6000 多个常用字,经中国科学家冯志伟等人测算,汉字的信息熵随着汉字个数的增加而增加,当汉字的个数达到 12 366 个时,汉字的信息熵值为 9.65bit,因此,汉字机内码必须用两个字节才能表示一个汉字。

汉字是当今世界上信息量最大的文字符号系统,这一特点使中文信息处理在信息编码、存储和传输等方面具有一定的特殊性,与其他语言处理相比较而言,中文处于不利的地位。

2.2.2 联合熵和条件熵

两个离散型随机变量 X 和 Y 的联合熵是确定两个随机变量的值所需的平均信息量,设 $X, Y \sim p(x, y)$, 它们的联合熵由下式给出:

$$H(X, Y) = - \sum_{x \in \Omega_x} \sum_{y \in \Omega_y} p(x, y) \log p(x, y) \quad (2-40)$$

给定另一个离散型随机变量 X ,一个离散型随机变量 Y 的条件熵是在已知随机变量 X 的取值的条件下,确定 Y 的取值所需的平均信息量,设 $X, Y \sim p(x, y)$, Y 的条件熵由下式给出:

$$\begin{aligned} H(Y | X) &= \sum_{x \in \Omega_x} p(x) H(Y | X = x) \\ &= \sum_{x \in \Omega_x} p(x) \left[- \sum_{y \in \Omega_y} p(y | x) \log p(y | x) \right] \\ &= - \sum_{x \in \Omega_x} \sum_{y \in \Omega_y} p(x, y) \log p(y | x) \end{aligned} \quad (2-41)$$

在熵值的计算中,也有一个类似于公式(2-8)的链规则:

$$H(X, Y) = H(X) + H(Y | X) \quad (2-42)$$

$$H(X_1, X_2, \dots, X_n) = H(X_1) + H(X_2 | X_1) + \dots + H(X_n | X_1, \dots, X_{n-1}) \quad (2-43)$$

公式(2-43)称为熵的链规则。

2.2.3 互信息

由熵的链规则可知,

$$H(X, Y) = H(X) + H(Y | X) = H(Y) + H(X | Y)$$

于是,

$$H(X) - H(X | Y) = H(Y) - H(Y | X)$$

这一差值称为随机变量 X 和 Y 的互信息(mutual information),记作 $I(X; Y)$ 。两个随机变量的互信息可以解释为知道一个随机变量的取值后对另一个随机变量的不确定性的减少量,或者一个随机变量包含的另一个随机变量的信息量。

两个随机变量的互信息是非负、对称的量度,可以用于衡量两个随机变量的依赖程度(或者独立性),当两个随机变量独立时,它们的互信息刚好为 0,互信息的取值越大,表明两个随机变量地依赖程度越高。经简单推导可知,

$$I(X;Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \quad (2-44)$$

由于 $H(X|X)=0$, $H(X)=H(X)-H(X|X)=I(X;X)$, 因此, 信息熵又称为自信息 (self-information)。互信息与熵的关系由图 2-2 给出。

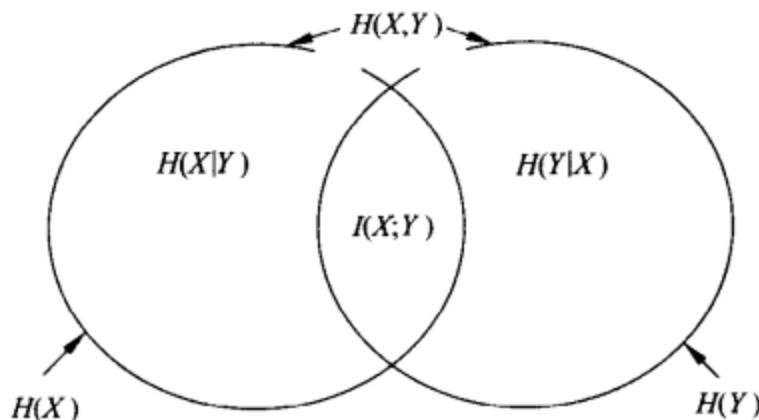


图 2-2 互信息与熵关系图

随机变量 X 和 Y 在随机变量 Z 已知的条件下的条件互信息由下式给出:

$$I(X;Y|Z) = I((X;Y)|Z) = H(X|Z) - H(X|Y,Z) \quad (2-45)$$

互信息的链规则由下式给出:

$$\begin{aligned} I(X_1;Y) &= I(X_1;Y) + \cdots + I(X_n;Y|X_1, \dots, X_{n-1}) \\ &= \sum_{i=1}^n I(X_i;Y|X_1, \dots, X_{i-1}) \end{aligned} \quad (2-46)$$

在研究自然语言词汇的搭配关系中, 经常使用“互信息”作为描述两个单词之间关联程度大小的量度。这里的“互信息”是指点与点的互信息, 它的计算方法由下式给出:

$$I(x,y) = \log \frac{p(x,y)}{p(x)p(y)} \quad (2-47)$$

其中, $p(x)$ 和 $p(y)$ 分别是随机变量 x 和 y 的概率, $p(x,y)$ 是二元搭配 (x,y) 出现的概率。

当 x 和 y 表示单词时, 互信息 $I(x,y)$ 比较了二元单词对 (x,y) 构成的固定搭配与仅仅是单词 x 或 y 的偶然出现的可能性, 如果两个单词的互信息 $I(x,y)>0$, 则说明它们存在着比较紧密的二元搭配关系, $I(x,y)$ 越大, 这种搭配关系越紧密; 如果 $I(x,y)=0$, 则说明它们的同时出现仅属偶然; 如果 $I(x,y)<0$, 说明这两个单词之间不存在二元搭配关系。

互信息广泛地应用于统计语言处理, 例如词的自动分类、语义排歧等。

2.2.4 相关熵

任意给定两个概率密度函数 $p(x)$ 和 $q(x)$, 它们的相关熵 (relative entropy) 由下式给出:

$$D(p \parallel q) = \sum_{x \in \Omega_x} p(x) \log \frac{p(x)}{q(x)} \quad (2-48)$$

规定 $0 \log \frac{0}{q} = 0$; $p \log \frac{p}{0} = \infty$ 。

相关熵又称为 Kullback-Leibler 距离 (Kullback-Leibler divergence), 是衡量在同一事件空间的两个概率分布的差异的一个量度。如果用数学期望来描述, 则,

$$D(p \parallel q) = E_p \left(\log \frac{p(X)}{q(X)} \right) \quad (2-49)$$

相关熵这一量度是非负的, 且 $D(p \parallel q) = 0$, 当且仅当 $p = q$ 。值得注意的是, 相关熵不满足对称性, 即 $\exists p, q, D(p \parallel q) \neq D(q \parallel p)$ 。相关熵也不满足三角不等式, 即 $\exists p, q, r, D(p \parallel q) + D(q \parallel r) \leq D(p \parallel r)$, 因此相关熵这一量度不是可度量的量 (metric), 尽管如此, 相关熵仍然是衡量概率分布之间的相近程度的重要的量度。

互信息与相关熵的关系由下式给出:

$$I(X;Y) = D(p(x,y) \parallel p(x)p(y)) \quad (2-50)$$

2.2.5 语言与熵

熵是不确定性的量度。因此对事物了解得越多, 它的熵就越小。对于构造语言的统计模型而言, 如果一个语言模型更加精确地描述了语言的结构, 那么它的熵应该越低。我们能够使用熵作为衡量语言模型的质量的参数。

1. 语言的熵

如果假定语言是由 n 个记号 (字符或者词汇) 构成的序列 $L = (X_i)$, 则语言信息所包含的信息量与信息的长度有关。对于一条长度为 n 的信息, 我们更加关心的是平均每个字符或者词汇的信息量。平均每个字符或者词汇的熵称为熵率 (entropy rate), 熵率的计算公式由公式 (2-51) 给出:

$$H_{\text{rate}} = \frac{1}{n} H(H_{1:n}) = -\frac{1}{n} \sum_{x_{1:n}} p(x_{1:n}) \log p(x_{1:n}) \quad (2-51)$$

可以用熵率来定义该语言的熵 (entropy of language):

$$H_{\text{rate}}(L) = \lim_{n \rightarrow \infty} \frac{1}{n} H(X_1, X_2, \dots, X_n) \quad (2-52)$$

由于 $p(x_{1:n})$ 未知, 故仍然无法从公式 (2-52) 中直接计算语言的熵。但是, 通过考察在大规模真实语料中语句的概率分布, 可以给出 p 的估计 m , 这个过程称为构造语言的模型或者语言的模型化。根据相关熵的定义可知, 语言的真实概率分布 p 与其模型 m 的相关熵 $D(p \parallel m)$ 越小, 模型 m 与 p 符合得越好, 模型 m 也就越精确。然而, 由于 p 未知, 所以仍然无法直接计算 $D(p \parallel m)$ 。为了能够评价模型 m 的精确性, 只好求助于一个新的量度, 即交叉熵。

2. 交叉熵

设任一随机变量 X , 其真实的概率密度函数为 $p(x)$ 。对这一随机变量, $p(x)$ 与另外一个概率密度函数 q (通常是 p 的一个模型) 的交叉熵 (cross entropy) 由下式给出:

$$\begin{aligned}
 H(X, q) &= H(X) + D(p \parallel q) \\
 &= -\sum_x p(x) \log q(x) \\
 &= E_p \left(\log \frac{1}{q(x)} \right)
 \end{aligned} \tag{2-53}$$

根据这个定义,任一种语言 $L = (X_i)$,其概率密度函数 $p(x)$ 与它的概率模型的交叉熵为

$$H(L, m) = -\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{x_{1:n}} p(x_{1:n}) \log m(x_{1:n}) \tag{2-54}$$

由于 $p(x)$ 未知,所以无法直接计算 $H(L, m)$ 的值。如果假设语言是各态可遍历的随机过程,那么可以证明,

$$H(L, m) = -\lim_{n \rightarrow \infty} \frac{1}{n} \log m(x_{1:n}) \tag{2-55}$$

对于足够大的 n ,公式(2-55)可以近似为

$$H(L, m) \approx -\frac{1}{n} \log m(x_{1:n}) \tag{2-56}$$

公式(2-56)给出了衡量一个语言统计模型的优劣的定量评价方法。由公式(2-56)和公式(2-53)可见, $H(X)$ 为一固定值, $H(L, m)$ 越小, 表明 $D(p \parallel m)$ 越小, 也就是说 m 与 p 符合得好, 模型 m 也就越精确。在构造语言的统计模型时, 可以使用该模型的交叉熵这一量度定量地评价这一模型, 并向着使 $H(L, m)$ 减小的方向努力改进模型。

在语音识别领域经常使用迷惑度(perplexity)来衡量统计模型的优劣, 它是一个与交叉熵等价的量度, 一个模型的迷惑度与交叉熵的关系由下式给出:

$$\text{perplexity}(x_{1:n}, m) = 2^{H(x_{1:n}, m)} \tag{2-57}$$

2.2.6 噪声信道模型

统计语言模型广泛地应用于语音识别、文本校错、机器翻译、词性标注、中文键盘输入等各个应用领域。本小节用信息论中的信源-信道模型来说明它在各类应用中的作用, 并说明如何将语音识别及其他的一些应用纳入这一模型框架之中。

语音识别器的任务阐明如下: 一个声学信号 A 对应于一个语句, 一个语音识别器需找到其对应的可能性最大的语言文本 T , 即

$$T = \arg \max_T P(T | A) \tag{2-58}$$

但是, 直接构造 $P(T | A)$ 的精确模型通常十分困难^[3], 于是, 根据贝叶斯公式可得到如下的公式:

$$T = \arg \max_T \frac{P(T) P(A | T)}{P(A)} = \arg \max_T P(T) P(A | T) \tag{2-59}$$

公式(2-59)中的概率分布 $P(T)$ 称为语言模型, 它描述了任一语句属于某一种语言的可能性的大小。概率分布 $P(A | T)$ 称为声学模型, 它描述了一个语句 T 对应于声学信号 A 的可能性。语言模型 $P(T)$ 就是自然语言的概率模型。

信息论中的信源-信道模型描述了发自信息源经过一个噪声信道信息恢复的问题,

它的定义如下：假设有一信息源模型 $P(I)$ ，有一噪声信道模型 $P(O|I)$ （表示在输入信息为 I 时，输出信息为 O 的可能性。显然，一个理想信道（无噪声信道），对于任意 $O=I$ ， $P(O|I)=1$ ，否则 $P(O|I)=0$ ），现要求通过在信道的另一端的输出 O 来恢复信源发出的原始信息 I ，即需找到

$$I = \arg \max_I (P(I|O)) = \arg \max_I \frac{P(I)P(O|I)}{P(O)} = \arg \max_I P(I)P(O|I) \quad (2-60)$$

将公式(2-60)与公式(2-59)相比较可以看出，信息源对应于以概率 $P(T)$ 生成的语句文本，噪声信道对应于以概率分布 $P(A|T)$ 将语句文本转换成的声音信号。语音识别的目的就是将通过噪声信道输出的声音信号恢复为其原始的语句文本。

信源-信道模型是一个非常典型的范例，只需修改信道模型的定义，就可以将语言模型在许多领域的应用纳入到这一框架之中。再以手写体文字输入为例，在这一应用中，噪声信道对应于将文本转换成图像数据，由如下公式所示：

$$T = \arg \max_T P(T)P(\text{image} | T) \quad (2-61)$$

在文本校错应用中，噪声信道对应于打字员将正确原文 T 打成含有错误的文档 T_n ，那么，

$$T = \arg \max_T P(T)P(T_n | T) \quad (2-62)$$

在机器翻译领域里，噪声信道对应于翻译器将一种语言的文本 T 翻译成另一种语言的文本 T_f ，于是，

$$T = \arg \max_T P(T)P(T_f | T) \quad (2-63)$$

在中文键盘输入应用中，噪声信道对应于打字员将汉字文档 T 以某种汉字编码 T_e 打出来，那么，

$$T = \arg \max_T P(T)P(T_e | T) \quad (2-64)$$

上述的各种应用场合都需要从已知噪声信道的输出来恢复原来的文本 T ，而噪声信道的输出可以是图像数据，有拼写错误的文本，某种语言书写的文档，或者是汉字文档的某种汉字编码。

通过改变信源模型，能够继续扩展信源-信道模型的应用领域。在词性标注中，如果把信源-信道模型中的信源定义为对语句生成相应的词性标注序列 T_{POS} ，把信道定义为将词性序列 T_{POS} 转换成相应的文本 T ，那么，

$$T_{\text{POS}} = \arg \max_{T_{\text{POS}}} P(T_{\text{POS}})P(T | T_{\text{POS}}) \quad (2-65)$$

可见，这一应用的目的就是依据噪声信道的输出文本来恢复其原来的词性序列 T_{POS} 。而构造模型 $P(T_{\text{POS}})$ 的方法与构造语言模型 $P(T)$ 的方法一致。

值得注意的是，这些应用具有一个共同的特点，就是都需要构造一个源语言模型，即 $P(T)$ 或者 $P(T_{\text{POS}})$ 。正是由于这一共性的重要性，这一工作才从各自的应用中独立出来自成一体，本书称之为语言模型化（language modeling）工作。语言模型化这一术语通常是指像 $P(T)$ 那样的源语言概率模型；本书使用语言模型（language model）这一术语将语言模型化这一术语加以泛化，它不仅包括像 $P(T)$ 那样的源语言概率模型，而且也包括

像信道模型、语言学语法那样的更一般的语言模型。

2.3 粗糙集

在 20 世纪 80 年代早期, Z. Pawlak 提出了用粗糙集(rough set)技术来处理结构化数据的分类问题^[1,2]。此后粗糙集理论作为一种新的处理模糊和不确定性知识的数学工具, 被广泛应用于机器学习、决策分析、过程控制、模式识别及数据挖掘等领域。为叙述方便, 后文将粗糙集简称为粗集。

在 Pawlak 的粗集模型中^[1], 主要是以不可分辨关系(indiscernibility relation)为基础, 通过概念的上近似、下近似及属性约简, 来解决带有模糊或者不确定性的结构化数据的分类以及规则获取等问题。该方法除了增加集合的上近似、下近似等算子外, 并没有超出传统的集合论, 每个集合元素与相应的集合之间仍然是一个二值关系, 即一个元素要么属于该集合、要么不属于该集合, 即算子的运算结果是原集合的确子集(crisp set)。粗集与模糊集都是为刻画自然界中的模糊与不确定性而提出的关于集合的理论, 这两种理论之间具有很强的互补性, 粗集方法在提供了对传统概念的近似描述的同时也提供了对模糊概念的描述能力。因为模糊方式往往能使概念的表述更加简洁、自然, 所以对模糊概念的描述能力, 对于简化 KDD 中的概念、规则, 以及模式的表示具有重要价值。当然, 粗集的这种能力主要是通过粗集的成员函数(member function)来获得的, 如果要想进一步增加粗集对模糊与不确定问题的描述能力, 则往往需要通过对 Pawlak 的粗集模型进行泛化、扩展或者结合其他软件计算工具来完成。

2.3.1 信息系统

在粗集理论中, 论域内的数据是采用信息表或者信息系统的形式来存储的, 一个信息系统可以用以下四元组 $I = (U, A, V_a, f_a)_{a \in A}$ 来定义, 这里,

- (1) U 是所有个体(object)的非空有限集合;
- (2) A 为属性的非空有限集合;
- (3) 对于任意一个属性 $a \in A$, 有
 - 一个属性值的集合 V_a ;
 - 一个信息函数 $f_a: U \rightarrow V_a$ 。

这里信息函数 f_a 的作用是, 当给定一个个体时, 可以通过该函数来确定给定个体中每个属性的值。给定个体 x 及属性 $p \in A$, $f_p(x) \in V_p$ 表示属性 p 在个体 x 中的取值, 简记为 $p(x)$, 对于属性子集 $P \subseteq A$, $P(x) = \{p(x) : p \in P\}$ 。

在通常情况下, 属性集合 A 可以分成两部分, 即条件属性子集 C 与决策属性子集 D , $C \cap D = \emptyset$, $A = C \cup D$ 。把区分了条件子集和决策子集的信息表称为决策表(decision table)。

2.3.2 不可分辨关系

不可分辨关系是粗集理论的基础, 为了定义不可分辨关系, 首先给出等效关系的

定义：

定义 令 I 为一个信息系统,称在集合 U 上的一个关系 θ 为等价关系 (equivalence relation),当且仅当对任意个体 $u, v, w \in U$, θ 满足

- (1) 自反性, $u\theta u$;
- (2) 传递性, 如果 $u\theta v, v\theta w$, 则有 $u\theta w$;
- (3) 对称性, 如果 $u\theta v$, 则有 $v\theta u$, $\forall u, v \in U$ 。

而 Pawlak 模型中的不可分辨关系定义如下。

定义 令 I 为一个信息系统, $P \subseteq A$ 为一个属性子集, 称二元关系 IND_P 为个体集 U 上的一个 P -不可分辨关系 (P -indiscernibility relation), 定义: 对任意两个个体 $u, v \in U$, $u \cdot \text{IND}_P \cdot v$ 当且仅当 $p(u) = p(v)$, $\forall p \in P$ 。

显然, IND_P 是一个等价关系。直观地看, $u \cdot \text{IND}_P \cdot v$ 表示, 如果仅仅借助于属性集 P 所包含的特征, 则无法区分两个个体 $u, v \in U$ 。 $u \text{IND}_P v$ 也可记为 $\text{IND}_P(u, v)$, 同时, 在不发生混淆的情况下, 可用一元算子 $\text{IND}_P(u)$ 来表示一个个体子集 $X \subseteq U$, X 中的每个个体 $x \in X$ 皆满足 $\text{IND}_P(x, u)$ 。

虽然在 Pawlak 的粗集模型^[1]中要求不可分辨关系是一个等价关系,但实际上这一要求并不是必需的。Slowinski 等认为,除了自反性以外等价关系中的其他两种性质对于不可分辨关系而言并不是必需的,并在他们的模型中引入了相似关系以定义不可分辨关系,当然也可以在保持等价关系的约束条件下对不可分辨关系进行扩展。

2.3.3 集合近似

借助上节定义的不可分辨关系,首先给出论域 U 划分的概念。

定义 令 I 为一个信息系统,对于 U 上的一个不可分辨关系 θ , 定义 U 上的划分 H_θ 为: 对于两个个体 $x, y \in U$, 当且仅当 $x\theta y$, 它们才属于划分 H_θ 的同一个类(称为等价类),一个个体 x 的等价类可以表示为 $\theta(x) = \{y \in U : x\theta y\}$, 划分 H_θ 对应的等价类的集合用 U/θ 来表示。

根据上述定义,可以通过对论域 U 进行划分来获得 U 中满足特定条件的个体子集,其中我们最关心的是,在 U 中找出给定属性集的值满足一定条件的子集。由于论域中的概念是通过对属性集的值进行约束来定义的,因而这实际上是在论域中寻找所有隶属于某一概念的个体的集合,而从另一方面来看,也可以通过指定论域中的特定个体子集来定义一个概念。通过对个体子集中属性值的特点进行研究,可以总结出该子集所描述的概念的特征,然后再借助这一特征对论域中或者是对没有出现在论域中的个体是否隶属于这一概念进行判别。然而由于构造信息系统的数据集及属性集有限,当面对实际问题时,常常会有我们所关注的某些概念无法用论域内的个体子集来准确描述的情况。为此,需要寻找一种能够对概念进行近似描述的工具,这一点和模糊集中描述一个个体与一个集合的关系是有所区别的。而 Pawlak 模型中的集合近似的概念正是为了满足这一需要而提出的,下面给出集合近似的形式化定义。

定义 对于一个个体的子集 $X \subseteq U$ 及属性集 $P \subseteq A$, X 的 P -下近似 (P -lower approximation) 及 P -上近似 (P -upper approximation) 分别定义为

$$\underline{X}_P \stackrel{\text{def}}{=} Y\{\text{IND}_P(x) : \text{IND}_P(x) \subseteq X\} \quad (2-66)$$

$$\overline{X}_P \stackrel{\text{def}}{=} Y\{\text{IND}_P(x) : \text{IND}_P(x) \cap X \neq \emptyset\} \quad (2-67)$$

而集合

$$BN_P(X) = \overline{X}_P - \underline{X}_P \quad (2-68)$$

称为集合 X 的 P -边界 (P -boundary)。如果 $BN_P(X) = \emptyset$, 则称集合 X 相对于 P 是清晰的 (crisp); 否则称集合 X 对于 P 是粗糙的 (rough)。如果 X 对于 P 是粗糙的, 则仅靠属性集 P 无法准确刻画出集合 X 所表述的概念, 只能通过式(2-67)和式(2-68)给出一个近似的描述。

虽然集合的上近似与下近似是对该集合所描述概念的一种近似刻画, 但上近似与下近似本身仍然是传统集合, 个体与近似集合之间的关系并没有超出传统集合的二值关系。为了增强粗集对模糊性的处理能力, Mohamed 通过 α 粗集模型扩展了 Pawlak 模型中的近似集合的概念, 从而每个近似集合都由原来的传统集合扩展成了模糊集。

2.3.4 约简

粗集的一个重要的作用是, 在保留原有信息系统的分类和概念表示能力不变的情况下, 尽可能对信息系统中的冗余信息进行约简。这种约简体现在两方面, 一方面是基于 2.3.2 节给出的不可分辨关系, 由于在借助不可分辨关系将一个信息系统进行划分后, 可以用每个个体子集中的任意一个个体来代替该子集进行存储, 因此实现了对原有信息系统的约简。这是一种基于个体的纵向约简。另一方面则是基于属性的约简^[3], 在这一约简过程中, 只保留那些能够保证原有的不可分辨关系 (以及集合近似) 不被改变的属性, 并将其他属性从信息系统中移除。虽然这两种约简经常同时出现在一个应用过程中, 但第一种约简比较简单直观, 所以这里主要研究后一种约简情况, 如果没有特别说明, 本节的约简算法都是指基于属性的约简。

定义 给定一个信息系统 $I = (U, A, V_a, f_a)$, 称属性子集 B 为 I 的一个约简 (reduct), 当且仅当 B 满足:

- ① 对于 I 中的任意一个个体 $\forall x \in U$, 有 $\text{IND}_B(x) = \text{IND}_A(x)$;
- ② $|B|$ 为所有满足上一条件的属性子集中最小的, 这里 $|B|$ 为集合 B 的势。

一个信息系统的约简可能不止一个, 信息系统 I 的所有约简的交集称为 I 的核 (core), 显然, 核中的所有属性都是信息系统保持原有概念表述能力不变时所必需的特征。

计算一个信息系统的约简是一个 NP-Hard 问题。为了解决这个问题, Skowron 等人提出了基于辨识函数 (discernibility function) 与分辨矩阵 (discernibility matrix) 的约简算法^[4], Shan & Ziarko 等人则在分辨矩阵的基础上提出了约简与核的增量求解方法^[5]。由于分辨矩阵在很多约简算法中都具有重要地位, 下面将给出分辨矩阵的概念。

令 $I = (U, A, V_a, f_a)$ 为一个包含 n 个个体的信息系统, I 的分辨矩阵为一个 $n \times n$ 的对称矩阵, 矩阵中的每个元素 e_{ij} 为

$$e_{ij} = \{a \in A \mid a(x_i) \neq a(x_j)\} \quad i, j = 1, \dots, n \quad (2-69)$$

这里, x_i 为 U 中的第 i 个个体。显然, 根据式(2-69), 分辨矩阵的元素为一个属性子集, 该子集包含了两个个体 x_i 和 x_j 中取值不同的所有属性。

令 I 为一个信息系统, I 的分辨函数 f_I 定义为 m 个布尔变量 a_1^*, \dots, a_m^* (对应于属性 a_1, \dots, a_m) 的函数:

$$f_I(a_1^*, \dots, a_m^*) = \bigwedge \{ \bigvee e_{ij}^* \mid 1 \leq i \leq n, e_{ij} \neq 0 \} \quad (2-70)$$

这里, 符号“ \wedge ”和“ \vee ”分别表示析取与合取。分辨函数 f_I 的主蕴含的集合决定了信息系统 I 的所有约简。这里蕴含指的是由多个文字(变量或变量的非)采用合取符号连接起来所构成的合适公式, 主蕴含指的是最小蕴含。在式(2-70)中, 如果保持变量 j 等于一个常数 k , 即合取运算只遍历分辨矩阵中的第 k 列, 则可以得到 k 相关分辨函数, 该函数的所有主蕴含决定了 I 的所有的 k 相关约简, 这些约简表示了在信息系统中分辨第 k 个个体所需的最少信息。

在有导师学习问题中(在训练集中, 给出了每个个体在决策分类中所属的类别), 我们的目标是在给出条件属性集的值的情况下判别一个新个体决策属性 d 的取值(对新的个体进行分类)。显然, 由于受到知识库空间需求的限制, 通常要求判别时所需的条件属性集为最小。

如上所述, 在一个决策系统中, 通常将属性集分成条件属性集 C 与决策属性集 D 两部分。考虑决策属性集只包含一个决策属性 d 的情况, 即 $A=C \cup d$, 这是在一个决策系统中最常见的情况, 则集合 $d(U)=\{k \mid d(x)=k, x \in U\}$ 的势称为 d 的秩(rank), 记为 $r(d)$ 。借助决策属性 d 可得到论域 U 的一个划分 $CLASS_I(d)=\{X_1^d, \dots, X_{r(d)}^d\}$, 称该划分为信息系统 I 中有决策 d 确定的个体的分类, 集合 X_i^d 称为 I 的第 i 个决策类。对于任何 $u \in U$, 用 $X_I(u)$ 表示决策类 $\{x \in U \mid d(x)=d(u)\}$ 。

令 I 为一个决策信息系统, $A=C \cup d$, I 中的决策是一个函数 $\partial_I: U \rightarrow V_d$, 定义为 $\partial_I(x)=\{i \mid \exists x' \in U x' \text{ IND}_{Cx}, d(x)=i\}$ 。如果对任意个体 $x \in U$, 有 $|\partial_I(x)|=1$, 则称 I 为一致的(consistent)或确定的(deterministic), 否则称 I 为非一致的(inconsistent)或非确定的(non-deterministic)。

由于决策信息系统(或称决策系统)中决策属性具有特殊的意义, 所以给出如下定义: 令 I 为一个一致决策系统, $A=C \cup d$, $M_I=(e_{ij})$ 为 I 的分辨矩阵, 可以构造一个新的矩阵 $M_I^d=(e_{ij}^d)$, 这里假设如果 $d(x_i)=d(x_j)$, 则 $e_{ij}^d=\phi$, 否则 $e_{ij}^d=e_{ij}-\{d\}$ 。矩阵 M_I^d 称为 I 的决策相关分辨矩阵(decision-relative discernibility matrix)。同样, 可以按与构造分辨函数相同的方法在 M_I^d 的基础上构造一个决策相关分辨函数 f_I^d 。

2.3.5 属性依从

数据分析中的一个重要问题是挖掘属性之间的依从关系。直观地, 令 I 为一个信息系统, 属性子集 $P, Q \in A$ 被称为是函数依从的(function dependent), 当且仅当对任意个体 $\forall u, v \in U$, 有 $P(u)=P(v) \Rightarrow Q(u)=Q(v)$, 记为 $P \rightarrow Q$ 。否则, 称 Q 以依从度 μ 部分依从于 P , $P(\mu) \rightarrow Q$, 这里 μ 称为依从度(degree of the dependency), 定义为

$$\mu(P, D) = |\text{POS}_P(Q)| / |U| \quad (2-71)$$

其中, $\text{POS}_P(Q)=\sum_{x \in U/Q} X_P$, 称为划分 U/Q 相对于 P 的正域。不难看出, 如果属性集 Q 函数

数依赖于 P , 则有 $\text{IND}_P \subseteq \text{IND}_Q$, 这意味着由属性 P 来决定的论域的划分比由 Q 来决定的论域的划分具有更精细的粒度。

2.3.6 决策规则合成

当获得了决策系统的约简以后, 可以通过将决策表中的属性值代入约简中相应属性来构造出最小决策规则集。

令 I 为一个决策系统, $V = Y\{V_a | a \in C\}YV_d$, 定义在集合 $B \subseteq CY\{d\}$ 及 V 上的原子公式 $a = v, a \in C, v \in V_a$ 。令集合 $\mathcal{F}(B, V)$ 为覆盖了集合 B 和 V 上的所有原子公式的最小公式集合, 集合中每个公式由原子公式以及命题联接词“ \wedge ”(合取)、“ \vee ”(析取)及“ \neg ”(非)组成。令 $\varphi \in \mathcal{F}(B, V)$, $\|\varphi_I\|$ 代表 φ 在决策表 I 中的语义, 即 U 中所有满足 φ 的个体集合。这些集合定义如下:

如果 φ 的形式为 $a = v$, 则 $\|\varphi_I\| = \{x \in U | a(x) = v\}$, $\|(\varphi \wedge \varphi')_I\| = \|\varphi_I\| \cap \|\varphi'_I\|$, $\|(\varphi \vee \varphi')_I\| = \|\varphi_I\| \cup \|\varphi'_I\|$, $\|\neg \varphi_I\| = U - \|\varphi_I\|$ 。

集合 $\mathcal{F}(B, V)$ 称为 I 的条件公式(conditional formulae), 决策系统 I 中的一个决策规则定义为形如 $\varphi \Rightarrow d = v$ 的表达式, 这里 $\varphi \in \mathcal{F}(B, V), v \in V_d$ 且 $\|\varphi_I\| \neq \emptyset$ 。合适公式 φ 和 $d = v$ 分别称为决策规则 $\varphi \Rightarrow d = v$ 的前趋项(predecessor)与后继项(successor), 决策规则 $\varphi \Rightarrow d = v$ 为真的充要条件为 $\|\varphi_I\| \subseteq \|(d = v)_I\|$ 。这里 $\|\varphi_I\|$ 为所有与决策规则匹配的个体的集合, 个体子集 $\|\varphi_I\| \cap \|(d = v)_I\|$ 称为决策规则的支持集(supporting set)。

从决策系统(决策表)中建立规则库是粗集技术的一个非常重要的应用, 这里需要解决的主要问题是, 决定哪些属性应该被包含在规则的条件属性集中。虽然可以通过约简来计算最小决策规则, 但这一方法所获得的结果往往可能出现过度适应(over-fit)现象, 而且因规则粒度过细而导致对未知个体的判别能力降低。为了解决这个问题, 通常可以通过构造更短的规则来提高规则集的鲁棒性(但是以适度降低规则集对于决策系统中的已知个体的决策性能为代价的)。

2.4 小结

本章概述了统计语言处理技术的数学基础, 其中, 概率论是统计语言模型的理论基础; 信息论广泛地应用于统计语言处理的各项应用和语言模型的评价中; 而粗糙集作为一种处理不确定性知识的数学工具, 广泛地运用于机器学习、模式识别、数据挖掘等领域。

参考文献

- 1 Chris Manning & H. Schütze. Foundations of Statistical Natural Language Processing. MIT Press, 1999
- 2 朱翼隽, 陶永德, 王文初. 概率论. 北京: 机械工业出版社, 1998
- 3 国家语言文字工作委员会, 国家标准局编. 现代汉语字频统计表. 1992
- 4 冯志伟. 自然语言的计算机处理. 上海: 上海外语教育出版社, 1996

计算机自然语言处理

- 5 中国大百科全书总编委员会《数学》编辑委员会. 中国大百科全书—数学卷. 北京: 中国大百科全书出版社, 1988
- 6 齐玉霞. 英汉数学词典. 北京: 科学出版社, 1984
- 7 C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 1948, 27: 379~423, 623~656
- 8 维纳著, 钟韧译. 维纳著作选. 上海: 上海译文出版社, 1978
- 9 Z. Pawlak, Rough Sets. *International Journal of Computer and Information Sciences*, 1982, 11: 341~356
- 10 W. Marek, Z. Pawlak. Rough Sets and Information Systems. *Fundamenta Informaticae*, 1984, 17: 105~115
- 11 Xiaohua Hu, Nick Cerone. Mining Knowledge Rules from Databases. *Proceedings of the Twelfth International Conference on Data Engineering*. New Orleans, USA, 1996: 96 ~105
- 12 A. Skowron, C. Rauszer. The Discernibility Matrices and Functions in Information Systems. In: R. Slowinski, (Ed.). *Intelligent Decision Support-Handbook of Applications and Advances of the Rough Sets Theory*. Dordrecht: Kluwer Academic Publishers, 1992
- 13 N. Shan, W. Ziarko. Data-based Acquisition and Incremental Modification of Classification Rules. *Computational Intelligence*, 1995, 11: 357~370



汉语自动分词技术

词是自然语言中最小的有意义的构成单位。汉语文本是基于单字的文本，汉语的书面表达方式以汉字作为最小单位，词与词之间没有明显的界限标志，因此，分词是汉语文本分析处理中首先要解决的问题之一。本章将简要介绍中文分词中的主要问题，包括汉语分词的规范、歧义的分类和识别、新词的识别等问题。同时还将介绍目前几种常见的分词方法：正向最大匹配分词方法、反向最大匹配分词方法、全切分词网格分词方法等。最后介绍有关分词的比较常用的统计规律。

3.1 引言

由于汉语的书写习惯，汉语句子中词与词之间的边界标志是隐含的，对于大多数汉语处理系统来讲，第一步要识别这些隐含的词语边界，即添加明显词语边界标志，使得所形成的词串反映句子的本意。这个过程就是分词。

分词问题是汉语计算语言学的一个古老课题。相关研究始于 20 世纪 80 年代初，迄今已有 20 多年的发展历史，研究出了很多各具特色的方法，从简单的模式匹配、基于规则的方法^[1~4]到基于统计的方法^[5~10]。虽然这些方法大大推动了汉语分词研究的发展，但在实际应用中仍然存在汉语词义模糊、构词模式自由、汉语词典覆盖能力有限，以及汉语语料库资源缺乏等诸多制约因素。在 1998 年，由国家 863 计划智能机主题专家组等单位组织的现代汉语分词系统评测中，最高的分词正确率只有 87.42%。

如何面向大规模开放应用是汉语分词研究亟待解决的主要问题。在处理大规模开放文本时，汉语分词系统还将面临以下困难：

(1) 如何识别未登录词。由于不存在绝对完备的词典，虽然一般的词典都能覆盖大多数的词语，但有相当一部分的词语不可能穷尽地收入系统词典中，这些词语称为未登录词或新词。常见的未登录词有如下几类：

- ① 专有名词，包括中文人名(如“朱镕基总理”)、地名(如“綦江县”)、机构名

称(如“杭州娃哈哈集团公司”)、外国译名(如“克林顿总统”)、时间词(如“1979年5月26日”);

- ② 重叠词,如“高高兴兴”、“研究研究”;
- ③ 派生词,如“一次性用品”;
- ④ 与领域相关的术语,(如“互联网”。

一个鲁棒的分词系统必须具备识别这些未登录词的能力。虽然迄今出现了很多识别未登录词的方法,包括规则方法^[2]和统计方法^[11],但由于汉字组词自由,缺少像英语那样的显性识别信息(如大写字母)等原因,目前大多数识别方法只能进行小规模应用或只能识别某些类型的新词,如中文人名、地名等。

(2) 如何廉价高效地获取分词规则是汉语分词系统设计中不可忽视的问题之一。一方面,目前还没有一个可以利用的大规模的汉语分词语料,而人工加工大规模的分词语料是耗费很大的工作;另一方面,任一汉字对之间都可能是一个词语边界,而且分词直接面对的是词,参数空间巨大,目前还没有适用于分词的完全有效的无指导参数学习方法。近来一些学者试图采用近似的方法来解决知识获取难题。Sproat^[10]于1996年提出一种利用串频(string frequency, SF)近似地逼近词频的方法,但该方法易产生短词较多的切分结果。Nagata于1997年^[9]提出了一种基于最大匹配串频的改进方法(longest match string frequency method, LSF),该法采用最长匹配的串频近似估计相应的词频,虽然能避免分词结果中短词的大量出现,但该方法容易产生这样的数据稀疏:当词 w_1 是词 w_2 的一个子串,而在训练语料中 w_1 总是作为 w_2 的子串出现时, w_1 的词频 $P_r(w_1)=0$ 。此外,孙茂松等^[8]还提出一种基于汉字互信息的无字典分词方法。

(3) 所谓词语边界歧义,指的是对于一个给定的汉语句子或汉字串,有多种词语边界划分形式。汉语词语边界歧义包括组合歧义和交叉歧义^[12,13]。

根据在大规模语料中的分析结果,交叉歧义还可细分为真歧义和伪歧义^[12]。真歧义指存在两种或两种以上的可实现的切分形式,如句子“必须/加强/企业/中/国有/资产/的/管理/”和“中国/有/能力/解决/香港/问题/”中的字段“中国有”是一种真歧义;而伪歧义一般只有一种正确的切分形式,如“建设/有”、“中国/人民”、“各/地方”、“本/地区”等。

在这些歧义中,伪歧义字段的切分结果是上下文无关的,一般仅依据字段内部的信息,如词频或字间互信息就可正确切分伪歧义字段,而真歧义字段或组合歧义字段的结果依赖于它所处的上下文环境,因而常常需要更多的信息,特别是上下文信息,才能正确处理真歧义字段。

(4) 实时性问题。大多数分词系统只注重分词准确率,而忽视了速度。有些应用系统,如机助翻译系统,对实时性能要求较高,要求分析算法对输入句子能做出迅速准确的处理。而对于给定的输入句子,其可能的切分词串数量与句子长度成指数关系。现已证明,最坏情况下的穷举搜索算法实际并不可行。贪心算法虽然能避免组合爆炸,但不能保证输出结果最佳。可见,分词算法的效率在实时性应用系统中的地位非常重要。

以上问题将在下文进一步讨论。

3.2 分词规范

在大多数表音文字中,词是由传统确定的,一般来说不存在分词规范问题。汉语的书写则以汉字为单位,是一种缺少严格意义的形态变化的表意文字语言,没有明显的形态界限可以作为分词标志,因而汉语存在特有的分词问题。由于汉语分词存在许多理论和技术问题,语言学界虽经数十年的努力,但至今尚未制订出一套认识一致的系统、完整和可行的分词标准。而语言处理技术的发展对分词标准的需求越来越迫切。制订“信息处理用现代汉语分词规范”的目的就是要克服存在的种种理论和技术困难,为汉语信息处理提出一整套通用、实用、科学及系统的分词规则。它将对汉语信息处理的规范化,对各种汉语信息处理系统之间的兼容性起到重要作用,进而还有可能对汉语言的规范化起到促进作用。

词是自然语言的一种客观存在。不管人们是否研究词,是否对词进行分类,它总是存在于语言之中。词是各民族的人民在长期使用中约定俗成的,不是由专家确定的。拼音文字由于书写过程必须分词连写,以空格为分词标记,故对词的认识是一致的。汉语则不同,虽然词客观存在于汉语中,每个人都在使用词,但是书写过程中并不分词连写,于是对词组和词、单字语素和单字词的划分因人而异,甚至因时而异。汉语信息计算机处理现在需要制订统一的分词标准^[14]。

《信息处理用现代汉语分词规范及自动分词方法》^[4]一书中所介绍的词应具有结合紧密、实用频繁的特点。下面是具体的分词标准实例。

(1) 结合紧密、使用稳定的二字或三字词,例如:

发展、可爱、红旗、对不起、自行车、青霉素。

(2) 四字成语一律为分词单位。例如:

胸有成竹、欣欣向荣。

四字词或结合紧密,使用稳定的四字词组,如

社会主义、春夏秋冬、由此可见。

(3) 五字和五字以上的谚语、格言等,分开后如不违背原有组合的意义,应予以切分,例如:

时间/就/是/生命/

失败/是/成功/之/母。

(4) 结合紧密,使用稳定的词组则不予切分,例如:

不管三七二十一。

(5) 惯用语和有转义的词或词组,在转义的语言环境下,一律为分词单位,例如:

妇女能顶/半边天/

他真小气,像个/铁公鸡/。

(6) 略语一律为分词单位,例如:

科技、奥运会、工农业。

(7) 分词单位加形成儿化音的“儿”,例如:

花儿、悄悄儿、玩儿。

(8) 阿拉伯数字等,仍保留原有形式,例如:

1234 7890

(9) 现代汉语中其他语言的汉字音译外来词,不予切分,例如:

巧克力、吉普。

(10) 不同的语言环境中的同形异构现象,按照具体语言环境的语义进行切分,例如:

把/手/抬起来

这个/把手/是木制的。

动词分词规范简介如下:

(1) 动词前的否定副词一律单独切分,例如:

不/写、不/能、没/研究、未/完成。

(2) 用肯定加否定的形式表示疑问的动词词组一律切分,不完整的则不予切分,例如:

说/没/说、看/不/看、相信/不/相信。

(3) 动宾结构的词,例如:

开会、跳舞、解决/吃饭/问题、孩子该/念书/了。

(4) 结合不紧密或有众多与之相同结构词组的动宾词组一律切分,例如:

吃/鱼、学/滑冰。

写/信(写/文章、写/论文、写/书……)

(5) 动宾结构的词或词组,若中间插入其他成分,则应予切分,例如:

吃/两/顿/饭、跳/新疆/舞。

(6) 动补结构的二字词或结合紧密、使用稳定的二字动补词组,不予切分,例如:

打倒、提高、加长、做好。

(7) “2+1”,1 或“1+2”结构的动补词组一律切分,例如:

整理/好、说/清楚、解释/清楚、打/得/倒、提/不/高。

(8) 偏正结构的词及结合紧密的词不予切分,例如:

胡闹、瞎说、死记。

(9) 复合趋向动词一律为分词单位,例如:

出去、进来。

当插入“得”、“不”时应予切分,例如:

出/得/去、进/不/来。

(10) 动词与趋向动词结合的词组一律切分,例如:

寄/来、跑/出去。

(11) 多字动词无连词并列,一律切分,例如:

调查/研究、宣传/鼓动。

3.3 常用的分词方法

常用的分词方法一共有3种：正向最大匹配分词、反向最大匹配分词、全切分词网格分词。下面分别予以介绍。

3.3.1 正向最大匹配分词

正向最大匹配方法(forward maximum matching method, FMM)的基本思想是：假设自动分词词典中的最长词条所含汉字个数为 I ，则取被处理材料当前字符串序数中的 I 个字作为匹配字段，查找分词词典。若词典中有这样的一个 I 字词，则匹配成功，匹配字段作为一个词被切分出来；如果词典中找不到这样的一个 I 字词，则匹配失败。匹配字段去掉最后一个汉字，剩下的字符作为新的匹配字段，进行新的匹配，如此进行下去，直至切分成功为止。亦即完成一轮匹配切分出一个词，然后再按上面的步骤进行下去，直到切分出所有的词为止。

例如短语“计算机科学和工程”，假设词典中最长的词为7字，则先取“计算机科学和工”为匹配字段，查找分词词典以匹配这个字段。由于词典中没有该词，故匹配失败。去掉最后一个汉字成为“计算机科学和”作为新的匹配字段，查找分词词典以匹配这个字段，同样匹配失败。再取“计算机科学”作为新的匹配字段进行查找，假定词典中有“计算机科学”一词，则匹配成功，切分出第一个词“计算机科学”。用同样的方法还可以切分出第二、第三个词……

根据梁南元^[15]的统计，FMM方法的错误切分率为1/169。因为用这种方法进行分词的时候，对上文提到的交叉歧义和组合歧义没有什么好的办法。因为对组合歧义来说，通常它都会作为一个分词单位，如“市场中国有企业才能发展”这个例句中，按照正向最大匹配分词方法，切分结果为“市场/中国/有/企业/才能/发展/”，可以看到，在这个例句中，有两个分词错误，分别为“中国/有”这个交叉歧义和“才能”这个分词的组合歧义。目前，FMM方法虽然作为一种基本的方法已被确定下来，但是由于它的错误切分率比较大，故一般不单独使用，而是与其他方法配合使用。

3.3.2 反向最大匹配分词

与FMM方法相对应的方法是反向最大匹配分词方法(backward maximum matching method, BMM)。它的分词过程与FMM方法相同，不过它是从句子(或文章)的末尾开始处理，每次匹配不成功时去掉的是最前面的一个汉字。BMM方法的精度要高一些，它的错误切分率为1/245。

例如“计算机科学和工程”，首先取“计算机科学和工程”作为匹配字段来查找分词词典，由于词典中没有该词，故匹配失败。去掉最前面的一个汉字，即取“算机科学和工程”作为新的匹配字段进行查找，同样匹配失败，如此进行下去，最后取“工程”作为匹配字段来查找分词词典，由于分词词典中有“工程”一词，则匹配成功，切分出第一个词“工程”。

在“市场中国有企业才能发展”这个例句中，按照反向最大匹配分词方法，切分结果为

“市场/中/国有/企业/才能/发展/”,可以看到在这个例句中有一个分词错误,那就是“才能”这个分词的组合歧义,应该切分为“才/能”。目前,BMM方法也作为一种基本的方法被确定下来了。

从正向最大匹配分词方法和反向最大匹配分词方法可以看出,配合使用这两种方法可以识别出分词中的交叉歧义,但是对组合歧义用两种方法却不能切分出来。

3.3.3 基于统计的词网格分词

词网格分词方法是基于统计的方法,它具有比较高的分词正确率和较好的可扩充性。可以通过加入相应的统计信息来扩展不同的功能。

词网格分词的第一步是选择词网格构造。利用词典匹配,列举输入句子所有可能的切分词语,并以词网格形式保存。实际上,词网格是一个有向无环图(directly acyclic graph,DAG),它蕴含了输入句子所有可能的切分,其中的每一条路径代表一种切分。图3-1表示的是字串“中国人民生活”的切分词网格,它包含了16种不同的切分情况。

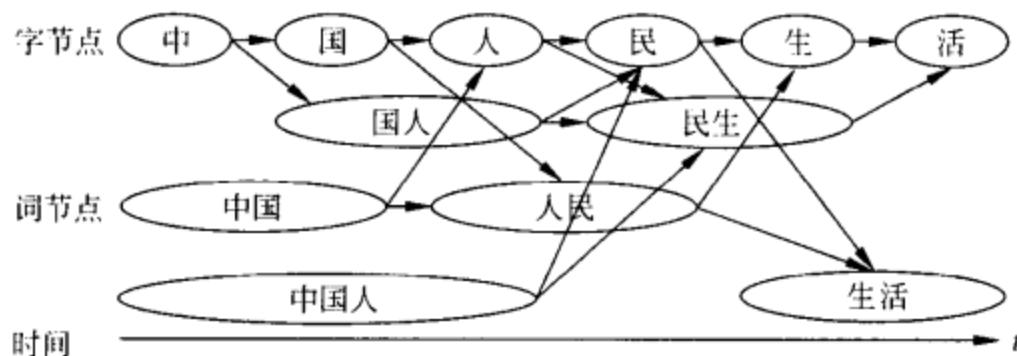


图 3-1 字串“中国人民生活”的切分词网格

第二步,选择计算词网格中的每一条路径的权值。权值通过计算图中每一个节点(每一个词)的一元统计概率和节点之间的二元统计概率的相关信息得到,然后根据图搜索算法在图中找到一条权值最优的路径,对应的路径即为最后的分词结果。

3.4 歧义的分类和识别

3.4.1 歧义的分类

歧义切分现象是自动分词中不可避免的现象,是自动分词中一个比较棘手的问题。对歧义切分字段的处理能力,严重影响到汉语自动分词系统的分词精度。实践表明,只用机械匹配进行分词,其精度最高不超过 $1/100 \sim 1/150^{[16]}$ 。

分词歧义分为组合歧义和交叉歧义两种,如下定义所示。

定义 组合歧义和交叉歧义:设有一汉字串 $C=c_1c_2\cdots c_n$,如果 C 构成词,且存在整数 $i_1, i_2, \dots, i_m, j_1, j_2, \dots, j_m (m \geq 2)$,满足:

- ① $w_1 = c_{i_1} \cdots c_{j_1}, w_2 = c_{i_2} \cdots c_{j_2}, \dots, w_m = c_{i_m} \cdots c_{j_m}$ 分别构成词;
- ② $1 = i_1 \leq j_1 < i_2 \leq j_2 < i_3 \leq j_3 < \cdots < i_{m-1} \leq j_{m-1} < i_m \leq j_m = n$;

则 C 为组合型歧义字段。如果 C 不构成词,且存在整数 $i_1, i_2, \dots, i_m, j_1, j_2, \dots, j_m (m \geq 2)$,满足:

① $w_1 = c_{i_1} \cdots c_{j_1}$, $w_2 = c_{i_2} \cdots c_{j_2}$, ..., $w_m = c_{i_m} \cdots c_{j_m}$ 分别构成词, 且 C 中不存在包含 w_1 , w_2 , ..., w_m 的词;

② $1 = i_1 < i_2 \leq j_1 < j_2, i_2 < i_3 \leq j_2 < j_3, \dots, i_{m-1} < i_m \leq j_{m-1} < j_m = n$;

则 C 为交叉型歧义字段。

例如, 句子“以/我/个人/的/名义/”和“他/一/个/人/在家/”的“个人”是一个组合歧义字段, 句子“从/小学/到/中学/”和“从小/学/计算机/”中, 字串“从小学”则存在交叉歧义。

从分词的结果上看, 歧义切分字段还可以分为如下两类:

第一类, 具有确定分法的歧义切分字段;

第二类, 具有不确定分法的歧义切分字段。

这里说的确定与不确定, 当然是对要进行分词的特定短句来说的。

例如在句子“在这种情况下工作是不可想象的”中的交集字段“下工作”, 只能惟一地切分成“下/工作”。再如“把手举起来”中组合歧义字段“把手”只能切分成“把/手”。因而, 在这个短句中它们都具有固定的切分形式。属于第一类情况。又如“研究生会采取行动”中的歧义的字段“研究生会”, 即可以切分成“研究生/会”, 又可以切分成“研究生会/”。这两种切分方法在语法上、语义上在此短句上都是正确的, 仅根据此短句不能确定哪种分词形式是正确的, 因而它属于第二类歧义切分字段。

很容易列举出属于第二类的其他歧义字段的例子: “乒乓球拍卖完了”, “美国会采取行动”等。从这类例子不难看出, 它们本身就是汉语言中的歧义问题, 解决这类歧义必须依据上下文语义信息。对歧义组合切分字段进行分类的目的是把由于计算机分词所产生的特有的歧义现象和自然语言本身所固有的歧义区分开来, 使我们能够排除干扰, 更好地处理计算机分词所产生的特有的歧义现象。

要在自动分词阶段解决第二类歧义切分字段, 就必须增加对上下文语义信息的处理, 即增加语义理解处理。这无疑对自动分词的效率有很大的影响(时间上和空间上), 但目前无法实现。若是在自动分词以后的处理阶段, 例如在短语理解、句子理解和篇章理解阶段, 对分词阶段没解决的歧义字段进行处理, 则会起到事半功倍的效果。统计表明, 第二类歧义字段只占整个歧义字段总数的 1/30 以下, 因此不必在分词阶段花费巨大的开销处理它们。

3.4.2 歧义的抽取和消歧

对歧义的识别分为两种, 一种是对交叉歧义的抽取, 一种是对组合歧义的抽取。首先介绍利用最大匹配和交叉歧义定义来抽取语料中的潜在交叉歧义字段。设有句子 $C = c_1 c_2 \cdots c_n$, $W = w_1 w_2 \cdots w_m$ 为相应的基于 FMM 法的粗切分词串, $w_k = x_1 x_2 \cdots x_{l_k}$ 和 $w_{k+1} = y_1 y_2 \cdots y_{l_{k+1}}$ 分别为 W 中的两个相邻词语, 根据交叉歧义定义, 如果 $w_k w_{k+1}$ 存在交叉歧义, 则字段 $x_1 x_2 \cdots x_{l_k} y_1 y_2 \cdots y_{l_{k+1}}$ 必定存在词语 $w' = x_i x_{i+1} \cdots x_{l_k} y_1 y_2 \cdots y_j$, 满足 $2 \leq i \leq l_k, 1 \leq j < l_{k+1}$ 。交叉歧义抽取算法具体描述如下:

输入 输入 FMM 切分句子 $W = w_1 w_2 \cdots w_m$ 。

输出 句子 W 包含的所有交叉歧义字段 AmbFragSet。

算法

```

初始化: AmbFrag $\leftarrow \emptyset$ ;
k  $\leftarrow 1$ 
while k < m-1
    AmbWordNum  $\leftarrow 0$  //交叉歧义字段包含的词数
    设  $w_k = x_1 x_2 \cdots x_{l_k}$ ,  $w_{k+1} = y_1 y_2 \cdots y_{l_{k+1}}$ 
    for i = 2 to  $l_k$ 
        for j = 1 to  $l_{k+1}$ 
            if LookupDictionary ( $x_i x_{i+1} \cdots x_{l_k} y_1 y_2 \cdots y_j$ ) = TRUE
                AmbWordNum  $\leftarrow$  AmbWordNum + 1
                k  $\leftarrow$  k + AmbWordNum
                break to 3
            else
                if AmbWordNum > 0 then AmbFragSet  $\leftarrow w_k - AmbWordNum \cdots w_k$ 
                k  $\leftarrow$  k + 1
            endif
        endfor
    endfor
endwhile

```

例如：输入正向分词结果“市场/中国/有/企业”算法依次处理 w_k 和 w_{k+1} ，当处理到“中国/有”时发现“中/国有”两词也存在于词典中，所以，“中国/有”为一交叉歧义。

下面介绍相应的组合歧义抽取算法。

汉语字词界限模糊，几乎每个汉字都可以成为词语，因此从理论上讲，任何一个汉语多字词就是一个潜在的组合歧义字段。换句话讲，给定词典中的多字词是一个理论上的组合歧义候选词集，但实际上真有意义的组合歧义字段数量很少，因此必须加以区分，以降低工作量。

设有词语 $w=w_1w_2$ ，其中 w_1, w_2 为 w 所包含的子字串，若 w 是一潜在的组合歧义词语，则根据组合歧义定义有 $|w|>1$ ，且 $\exists w_1, w_2 \in D$ 。设 $P_r(w)$, $P_r(w_1)$ 和 $P_r(w_2)$ 分别为 w, w_1 和 w_2 在语料中出现的概率，则存在

$$0 \leqslant score(w) = \frac{P_r(w)}{(P_r(w) + P_r(w_1))(P_r(w) + P_r(w_2))} \leqslant \frac{1}{P_r(w)} \quad (3-1)$$

若 $score(w)=0$ 或 $score(w)=1/P_r(w)$ ，则可认为 w 在所讨论的语料中不存在组合歧义。在最大匹配分词中， w 一般被分为一个词，因此，如果 w 满足条件：

① $P_r(w)>0$ ；

② $score(w)P_r(w) \leqslant \epsilon$ ($0<\epsilon<1$ ，根据实验确定， ϵ 越大，抽取精度越低，召回率越高)；

则可认为 w 在所讨论的语料中是一潜在的组合歧义词语。

下面介绍一种交叉歧义的处理策略^[12]。对于交叉歧义中伪歧义型高频最大交集型歧义切分字段，可以把它们的正确(惟一)切分形式预先记录在一张表中，其歧义消解通过简单的查表来完成。本质上，这是一个基于记忆的学习模型，其特点体现在以下几个方面：

(1) 适用范围广,前 4 279 个伪歧义型高频最大交集型歧义切分字段的覆盖率高达 53.35%。通过前面的讨论,已得出高频最大交集型歧义切分字段关于领域相当稳定这一结论。

(2) 对伪歧义型最大交集型歧义切分字段可实现无错误切分。考虑一个伪歧义的例子“提高人民生活水平”。如果按常规程序来处理,则首先需进行全切分(这样可在理论上彻底避免切分盲点),然后利用规则法或 n 元语法模型求解。但是,全切分往往产生膨胀的候选解空间(连接两个字间位置的弧指示该弧所跨越的汉字串构成一个词,从起始位置到结束位置的一条弧路径对应一种可能切分),要从如此庞大的解空间约束到一个正确解绝非易事,而方法本身存在的局限性又导致规则法或 n 元语法模型在某些情形下会陷入鞭长莫及、束手无策的境地(不妨设想一下,假使要求人完成同样的任务,需要综合运用多少知识!),虽经复杂计算、费尽周折,但仍然可能发生错判。基于记忆的模型则保证了切分结果的准确无误。

(3) 时间、空间代价小。存储一张入口项数量不足 5 000 的相关类型的数据表,所需内存仅约 100KB,且查表过程就是全部求解过程,避开了纷繁的计算,因此速度很快(平均 12 次二分查找即可定位)。

(4) 为基于实例的排歧计算提供了典型的实例集。比如,数据表中已将“今天下午”及其解“今天/下午”记录在案。如果遇到了数据表之外的另一个最大交集型歧义切分字段“今天上午”,则通过相似度比较,可以参照“今天下午”的解,推测出“今天上午”的解很可能是“今天/上午”。在统计分析的支持下,制定一个基于记忆的伪歧义型高频最大交集型歧义切分字段来进行消解,这个策略虽然似乎简单得不能再简单,但实验证明它行之有效。按照歧义切分字段消解策略实现一个预处理模块可以明显改善实用型非受限汉语自动分词系统的排歧能力。而隐寓于其中的,涉及智能实现机制的一些哲学思想也值得我们再三品味。

对组合歧义的处理通常需要上下文信息^[17]。排歧的基本模型采纳 WSD 中广泛使用的向量空间法。设字串 W_0 为组合歧义切分字段,句子 S_i 包含 W_0 ,将 W_0 左右各 d 个词所形成的窗口看作是 W_0 的上下文,窗口中每一个不同的词均为 W_0 的特征词,特征词关于句子的集合为 W_0 的特征词集。一般地, W_0 的切分形式有两种,即“合”和“分”两种情况。于是,特征词集中的每个特征词 W_j 又被依次赋予分别与两种不同切分形式相对应的两个权值,从而构成了特征矩阵。特征矩阵中权值列向量的维数和权值的大小因处理的方法而异。特征分矩阵和特征合矩阵可根据适当规模训练集自动获得。

任意输入一含 W_0 的句子,其排歧过程同上,即首先从该句中提取出 W_0 的上下文,继而得到 W_0 关于这个句子的特征矩阵,然后分别计算这个特征矩阵与分矩阵、合矩阵之间的距离,取较近距离所对应的下脚标作为切分结果。

3.5 新词的识别

新词的识别就是对未登录词的识别,所谓未登录词是指系统词典中没有收录的词。例如,汉语中的数量词(1250)、时间词(1998 年 8 月)、人名(赵健)、地名(宽甸县)、机构名

(松雷百货)等,没有办法把这些词全部收录到词典中去,但是这些词经常在局部文本中大量出现,所以识别这些新词是分词系统的一个重要任务。

中文的人名通常是由姓氏和名字两部分构成。姓氏是相对封闭的,可以作为人名识别的一个明显的左开端。人名通常可以通过基于统计的方法来识别。方法就是通过计算一个候选人名的概率值。如果该概率值大于某个值,那么就是一个人名,反之就不是一个人名。

地名的识别和人名的识别有些不同。有些地名有明显的右边界,例如,伊春市、五常县、松峰山等。通常通过地名的特点、用字规律、用词规律、构词规律和地名的上下文规律等,实现在真实文本中中国地名的自动识别。

下面介绍一种基于构词力的新词识别方法。通过统计发现,绝大多数的未登录词常被切分为单字词,因此新词识别在某种程度上就是一个在适当的地方和范围内融合这些连续的单字词的过程。Yao Yuan 博士^[11]曾提出基于 head-middle-tail 结构的构词模式和基于构词能力的未登录词识别方法,将新词识别看作是词语边界识别的后处理过程,应用简单叠加原理,提出了一种汉字构词能力、构词模式、字结合点和 N-gram 模型相融合的未登录词识别算法。

3.5.1 统计构词能力

从理论上讲,虽然任何一个汉字均可成为多字词的成员,但每个汉字构成多字词的能力并不一样。汉语中存在许多的功能字(function character),如“的”、“了”,这些字的频度一般较高,而且常常单独存在,构成多字词的能力较弱。设 WFP(c)为汉字 c 的构词能力(word formation power, WFP),则

$$WFP(c) = \frac{\text{count(含 } c \text{ 的多字词)}}{\text{count}(c)} \quad (3-2)$$

式(3-2)实际上反映了单个汉字构成多字词的概率大小。那么,从汉字构词能力角度,某一字串 $C=c_1c_2\cdots c_n$ 构成词语 w 的概率可表示为

$$P_{WFP}(w) = \begin{cases} 1 - WFP(c), & |C| = 1 (w \text{ 是单字词}) \\ \prod_{c_i \in w} WFP(c_i), & |C| > 1 (w \text{ 是多字词}) \end{cases} \quad (3-3)$$

3.5.2 汉字构词模式

对于任一汉字 c ,它在构成多字词时有如下 3 种模式:

- ① 词首(head) c 是多字词的首字,以 H 表示;
- ② 词尾(tail) c 位于多字词的尾部,以 T 表示;
- ③ 词中(middle) c 位于多字词的中部,以 M 表示。设 ptn(c)表示汉字 c 在某一多字词的模式,则

$$P_r(ptn(c) | c) = \frac{\text{count}(ptn(c))}{\text{count}(c \text{ 位于多字词})} \quad (3-4)$$

其中, $P_r(ptn(c) | c)$ 表示汉字 c 在某一多字词中取模式 ptn(c)时的条件概率。显然对于同一汉字 c ,存在 $\sum_{ptn(c)=H,M,T} P_r(ptn(c) | c) = 1$ 。

基于上述模式,任意一个多字词可能是如下两种结构之一: head+tail 和 head+middle+middle+…+tail。于是,多字词 $w=c_1c_2\cdots c_l$ 的构词模式概率为

$$P_{\text{pttn}}(w) = \prod_{i=1}^l P_r(\text{pttn}(c_i) \mid c_i) \quad (3-5)$$

3.5.3 未登录词识别算法

虽然未登录词在切分句子中常常表现为单字词,但并不是说可以自由地将任意的单字词融合成新的多字词,它不仅与新词内部各汉字的构词能力、构词模式和字对间的亲和力有关,而且与该新词所处的上下文有关。下面分别采用构词能力、字结合点模型和词语 N -gram 模型来描述这些特征信息,并采用线性插值方法将它们融合在一起。

设有词串 $W=w_L W_{\text{single}} w_R$,其中 $W_{\text{single}}=w_1 w_2 \cdots w_n$ 为一单字词串; w_L, w_R 分别为该串的左、右邻多字词,未登录词识别的任务就是对 W_{single} 重新进行词语边界划分,使新得到的词串更能反映句子的意思。设 $\hat{W}_{\text{new}}=x_1 x_2 \cdots x_m$ 为 $w_1 w_2 \cdots w_n$ 最可能的新的切分词串,则

$$\begin{aligned} \hat{W}_{\text{new}} &= \arg \max_{W_{\text{new}}} \{ P_{\text{WFP}}(W_{\text{new}}) + P_{\text{pttn}}(W_{\text{new}}) + P_{\text{CJM}}(W_{\text{new}}) + P_{\text{bigram}}(W_{\text{new}}) \} \\ &= \arg \max_{W_{\text{new}}} \{ \sum_{i=1, \dots, n} (P_{\text{WFP}}(x_i) + P_{\text{pttn}}(x_i) + P_{\text{CJM}}(x_i) + P_r(x_i \mid x_{i-1})) \} \\ &\quad \sum_{w_U} \text{count}(x_{i-1} w_U) \end{aligned} \quad (3-6)$$

式中, $x_0 = w_L$;当 x_i 为新词时, $P_r(x_i \mid x_{i-1}) = \frac{\text{count}(x_{i-1} w_U)}{\text{count}(x_{i-1})}$ (w_U 代表训练语料中的所有新词)。

作为词语边界识别系统的后处理,未登录词识别算法的核心就是对分词算法输出中的某一单字词串重新进行词语边界划分,从中找到一个满足式(3-6)的新的词串。未登录词识别算法的具体描述如下:

- (1) 从输入切分句子 $W=w_1 w_2 \cdots w_N$ 中寻找单字词串 $W_{\text{single}} (|W_{\text{single}}| > 1)$ 。
- (2) 如果 W_{single} 存在重叠词形式,如 $AABB, ABAB, ABB$,则将重叠部分组合成一个新词,转到步骤(4)。
- (3) 构造 W_{single} 的所有可能词语的边界划分,并利用分词算法从中搜索满足式(3-6)的最佳划分 \hat{W}_{new} 来替代 W_{single} 。
- (4) 检查输入句子,寻找下一个单字词串 W_{single} ,如果存在,则重复步骤(2)~(3);否则,输出结果,结束。

3.6 关于分词的若干统计结果

从 1993—1996 年《人民日报》中随机抽取约 19 251 887 个汉字的不同领域不同长度的文本,并采用分词算法对这些文本进行加工,形成了一个含有 12 348 010 词的训练语料。此外,对训练语料中的歧义字段和未登录词的分布情况进行了全部或抽样调查,调查结果分别见表 3-1、表 3-2 和表 3-3。

从这些数据中可以看出这样几个规律：

(1) 交叉歧义、组合歧义和词典未登录词加在一起约占训练语料总词次的 8.12%，这说明若不考虑其他因素，如词定义的不一致的影响，最简单的最大匹配方法的分词精度的下界应在 8% 左右。

(2) 训练语料中共出现 275 937 频次、41 394 种的交叉歧义字段，其中只有 218 种真歧义，约占歧义总数的 0.5%，其余全部为伪歧义。进一步的统计显示，歧义字段的平均长度是 2.37 词，平均约每 100 个词出现 5.30 交叉歧义字段，这说明交叉歧义是影响汉语分词性能的重要因素；另外，表 3-2 还表明频率最高的 771 种歧义字段约覆盖训练语料中 48.87% 的歧义。

(3) 组合歧义的消解虽然很困难，但其出现的概率很小，只占全部词次的 0.36%。

(4) 训练语料中出现的未登录词约占总词次的 2.46%，这说明未登录词是影响汉语词语边界划分准确率的重要因素之一；另外 5 394 个未登录词的抽样调查（见表 3-3）显示，人名和地名，特别是中文人名和地名，是未登录词的重要来源，分别占到 47.66% 和 14.83%，因此若能解决人名和地名的识别问题，近 80% 的未登录词问题就能迎刃而解。

表 3-1 训练语料中歧义和未登录词的分布

歧义	种数	频次	占总词次百分比/%	面向 FMM 切分的纠错规则数
交叉歧义	41 394	275 937	5.30	14 892
组合歧义	411	44 642	0.36	2 512
未登录词	23 932	303 531	2.46	23 932

表 3-2 交叉歧义字段随频度的分布

频度	≥ 200	≥ 100	≥ 50	≥ 20	≥ 10	≥ 5	≥ 2	≥ 1
歧义种数	138	325	771	1 954	3 892	7 721	20 667	41 394
总频度	78 779	104 800	134 859	170 939	197 013	221 761	255 209	275 937
覆盖度/%	28.55	37.98	48.87	61.95	71.40	80.37	92.49	100.00

表 3-3 未登录词分布

未登录词	人名		地名		机构名	重叠词	其他	合计
	中国	外国	中国	外国				
数量	2 571	565	800	292	231	74	861	5 394
百分比/%	47.66	10.47	14.83	5.41	4.28	1.37	15.96	100.00

3.7 语言单位的统计分布规律(Zipf 定律)

本节介绍一个适用于各种不同语言、各级语言单位的统计分布规律，它对于统计语言模型的建立具有重要的指导意义。

许多自然现象都遵循一定的统计规律。G. K. Zipf 发现,像工资收入的分布情况,英文单词词频和物种的分布情况等现象中,存在着一个普遍性的原理,他提出了一个定律来说明这一类现象,后人称之为 Zipf 定律,其实质是一个多项式衰减函数。Zipf 定律给出了英语文本中英语单词频度与词序之间的关系。下面简要讨论。

Zipf 定律是美国哈佛大学的语言学家 G. K. Zipf(1902—1950)发现的^[19]。他发现,在大量英文文本中对单词进行计数,并从最高频到最低频进行排序(w_1, w_2, \dots, w_n),那么其频度(f_1, f_2, \dots, f_n)近似地服从 Zipf 定律^[19],即 $f_n \propto \frac{1}{n}$,随后又由 Mandelbrot^[20]给出了更佳的近似 $f_n \propto n^{-1.05}$ 。自 Zipf 定律被发现以来,许多研究者对英语从音素级到语句级的不同结构层次,进行了统计研究^[21],给出了相应的 Zipf 定律。Zipf 定律对于自然语言所理解的各个领域,尤其对建立自然语言的统计模型具有重要的指导意义。

自 Zipf 定律被发现以来,许多语言学家、统计学家对英文的各个层次结构进行了统计研究,人们发现,英文的音素级、单词级、词性级、语句级、二元对(bigram)、三元对(trigram),甚至名词短语的词性序列、介词短语的词性序列等各个层次结构,都近似地遵循各自不同形式的 Zipf 定律。

那么,在现代汉语中各个层次的语言单位中,存在着怎样的词频-词序关系呢?我们在建立现代汉语计算语言模型的实践中通过实验发现,在现代汉语的字、词二元对等层次结构上,同样存在 Zipf 形式的词频-词序关系^[18]。发生这种现象的原因可能是由于人们在为书写文档选择词汇的时候,一般遵循如下两个原则:

- ① 关联原则,人们一般会使用他们先前使用过的词汇。
- ② 模仿原则,人们总是模仿和借鉴自己或其他人已有的作品。

研究大规模真实文本中的词频-词序关系,对建立现代汉语计算语言模型具有重要的意义。词频-词序关系表明,在大规模的真实文本中,少量高频的语言单位占有很高的比例。假如现代汉语中现存的词汇共有 100 000 个,而常用的词汇仅有 2 000 个,也就是说,仅仅考虑频度最高的 2 000 个词汇,那么这 2 000 词汇占统计语料中总词汇量的比例 Ratio 可近似地计算如下:

根据 Zipf 定律, $f(r) \cdot r = C(\text{常数}), r=1, 2, \dots,$

$$\begin{aligned} \text{Ratio} &= \frac{f(1) + f(2) + \dots + f(2000)}{f(1) + f(2) + \dots + f(100000)} = \frac{C(1 + 1/2 + 1/3 + \dots + 1/2000)}{C(1 + 1/2 + 1/3 + \dots + 1/100000)} \\ &= \frac{1 + 1/2 + \dots + 1/2000}{1 + 1/2 + \dots + 1/100000} \end{aligned} \quad (3-7)$$

由于 $\lim_{n \rightarrow \infty} \sum_{r=1}^n \frac{1}{r} - \ln n = E$ ($E = 0.5772$, 称为欧拉常数), 故

$$\text{Ratio} \approx \frac{\ln(2000)}{\ln(100000)} \approx 66\% \quad (3-8)$$

即占总词汇量 2% 的词汇的总出现次数占统计语料中的总词汇量的 66%。在对现代汉语的结构特征的研究中,可以将注意力集中在汉语中高频的词汇和结构中,它们已经覆盖了语料库中相当大的一部分。

Zipf 定律说明,仅仅基于词汇的分布情况,只能得到极少数高频词汇的有意义的统计

信息,而对于我们所感兴趣的绝大多数词汇或其他语言结构单位来说,很难得到什么有意义的结论。由此可以断定,许多语言单位的出现频率是极低的,而且还存在着极多合法的语言单位或者语言单位的搭配,由于语言单位未在用于训练语言模型的语料库中出现,因而难以分清到底这是因为在语法意义上非法,还是仅仅因为未在用于训练的语料库中出现。为解决这个数据稀疏问题,人们采用了许多办法,如扩大训练语料库的规模,采用一些数据平滑技术等。但是,语料库规模的扩大仅能起到十分有限的作用。

3.8 小结

本章讲述了中文自然语言处理中的最基本的处理步骤——分词和与其相关的统计规律。在分词部分主要介绍了分词的规范,分词的基本方法和分词中常见的问题。统计规律主要介绍了与分词相关的统计结果和 Zipf 规律。

参考文献

- 1 Wang Xiaolong, Wang Kaizhu, et al. The Problem of Separating Characters into Fewest Words and Its Algorithms. Chinese Science Bulletin (English edition), 1989, 34(22): 1924~1928
- 2 Yeh Ching-Long and Lee Hsi-Jian. Rule-Based Word Identification for Mandarin Chinese Sentences—A Unification Approach. Computer Processing of Chinese & Oriental Languages, 1991, 5(2): 97~117
- 3 梁南元,郑延斌.汉语分词模型和汉语分词系统 PC-CWSS. Communications of COLIPS, 1991, 1(1): 51~55
- 4 刘源.信息处理用现代汉语分词规范及自动分词方法.北京:清华大学出版社,广西:广西科技出版社,1994
- 5 Chiang Tung-Hui, Chang Jing-Shin, Lin Ming-Yu, and Su Keh-Yih. Statistical Models for Word Segmentation and Unknown Word Resolution. In Proceedings of ROCLING-V, R. O. C. Computational Linguistics Conferences, Taiwan, 1992. 123~146
- 6 高军,陈锡先.基于变长方法汉语语料自动分词. Communications of COLIPS, 1996, 6(2): 87~94
- 7 刘挺,吴岩,王开铸.串频统计和词形匹配相结合的汉语自动分词系统. 中文信息学报, 1998, 12(1): 17~22
- 8 Sun Mao-Song, Shen Da-Yang and Benjamin K Tsou. Chinese Word Segmentation without using Lexicon and Hand-crafted Training Data. Proceedings of COLING-ACL'98. Montreal, Canada, 1998. 1265~1271
- 9 Masaaki Nagata. A Self-Organizing Japanese Word Segmenter using Heuristic Word Identification and Re-estimation. In Proceedings of the Fifth Workshop on Very Large Corpora. Beijing, China, 1997. 203~215
- 10 Richard Sproat, Chilin Shih, William Gale, and Nancy Chang. A Stochastic Finite-State Word Segmentation Algorithm for Chinese. Computational Linguistics, 1996, 22(3): 377~404
- 11 Yao Yuan. Statistics Based Approaches towards Chinese Language Processing. Ph. D. thesis. National University of Singapore, 1997

- 12 孙茂松,左正平,邹嘉彦. 高频最大交集型歧义切分字段在汉语自动分词中的作用. 中文信息学报, 1999, 13(1): 27~34
- 13 Covering Ambiguity Resolution in Chinese Word Segmentation Based on Contextual Information COLING 2002, 2000. 598~604
- 14 刘源. 信息处理用现代汉语分词规范及自动分词方法. 北京: 清华大学出版社, 1994
- 15 梁南元. 书面汉语的自动分词与另一个自动分词系统 CDWS. 中国汉字信息处理系统学术会议, 桂林, 1983
- 16 第三次全国自然语言理解研讨会论文集. 太原: 1989
- 17 肖云, 孙茂松. 利用上下文信息解决汉语自动分词中的组合型歧义. 计算机工程与应用, 2001(19)
- 18 关毅, 王晓龙, 张凯. 现代汉语计算语言模型中语言单位的频度-词序关系. 中文信息学报, 1999, 13(2): 8~15
- 19 G. K. Zipf. Human Behavior and the Principle of Least Effort. Addison-Wesley, 1949
- 20 B. Mandelbrot. An informational theory of the statistical structure of languages. in Communication Theory, ed. W. Jackson (Betterworth, 1953), pp. 486~502
- 21 G. A. Miller, E. B. Newman. Tests of a statistical explanation of the rank-frequency relation for words in written English. American Journal of Psychology, 1958, 71, 209~218



第4章

CHAPTER 4

基于数学统计的语言模型

语言模型(language model)是描述自然语言内在规律的数学模型,构造语言模型是计算语言学的核心。在实践中,语言模型广泛地用于语言识别、手写体文字识别、机器翻译、键盘输入、信息检索等研究领域。

语言模型可分为传统的文法型语言模型和基于统计的语言模型。文法型语言模型是人工编制的语言学文法,文法规则来源于语言学家掌握的语言学知识,但这种语言模型不能处理大规模真实文本。为满足这一需求,基于统计的语言模型应运而生。这种语言模型通常是概率模型,计算机借助于统计语言模型的概率参数,可以估计出自然语言中每个句子出现的可能性,而不是简单地判断该句子是否符合文法。本章将对常用的统计语言模型,包括 N 元文法模型(N -gram model)、隐马尔科夫模型(hidden markov model, HMM)、最大熵模型(maximum entropy model)分别进行介绍,并给出如何去评价一个语言模型的方法。

4.1 统计语言模型概述

一个基于统计的计算语言模型以概率分布的形式描述了任意语句(字符串) s 属于某种语言集合的可能性。例如, $P(\text{他/认真/学习}) \approx 0.02$, $P(\text{他/认真/读书}) \approx 0.03$, $P(\text{他/认真/坏}) \approx 0$ 等。这里并不要求语句 s 在语法上是完备的,该模型需对任意的语句 s 都给出一个概率值。

假定词是一个句子的最小结构单位,并假设一个语句 s 由词 w_1, w_2, \dots, w_n 组成,那么,不失一般性, $P(s)$ 可由下式计算:

$$\begin{aligned} P(s) &= P(w_1)P(w_2 | w_1)P(w_3 | w_1w_2)\cdots P(w_n | w_1w_2\cdots w_{n-1}) \\ &= \prod_{i=1}^n P(w_i | w_1\cdots w_{i-1}) \end{aligned} \quad (4-1)$$

例如,对于二元模型而言, $P(w_i | w_1w_2\cdots w_{i-1}) = P(w_i | w_{i-1})$ ($1 \leq i \leq n$),式(4-1)为

$$P(s) = \prod_{i=1}^n P(w_i | w_{i-1}) \quad (4-2)$$

为保证式(4-1)和式(4-2)的完备性,必须引进一个起始词 w_0 ,并且假定 $P(w_1 | w_0) = P(w_1)$ 。式(4-2)和式(4-2)满足 $\sum_s P(s) = 1$ 。

概率 $P(w_i | w_{i-1})$ 可以使用许多方法估算,一种被广泛采用的称为最大似然度估计(maximum likelihood estimation, MLE)的方法用如下公式估算:

$$P(w_i | w_{i-1}) = \frac{\text{count}(w_{i-1} w_i)}{\sum_{w_i} \text{count}(w_{i-1} w_i)} \quad (4-3)$$

式(4-3)中, $\text{count}(w_{i-1} w_i)$ 为词对 $w_{i-1} w_i$ 在训练语料库中出现的次数。用于估算基于统计的计算语言模型中的概率分布的训练语料库文本称为训练数据(training data)。根据训练数据估算 $P(w_i | w_{i-1})$ 这类概率分布的过程称为训练。

前文曾经提到大量的低频词,无论训练数据的规模如何扩大,其出现频率仍旧很低,甚至根本不出现。如果采用 MLE 估算它们的概率分布,将出现大量的 $P(w_i | w_{i-1}) = 0$,从而导致 $P(s) = 0$ 的情况。这种情况大大削弱了该模型的描述能力。数据平滑技术(data smoothing)能够解决这一类问题,它通过调整 $P(w_i | w_{i-1})$ 这类概率分布的取值方法,避免 $P(s) = 0$ 这类情况的出现,达到使 $P(s)$ 的计算更加精确的目的,我们将在 4.3 节对其进行详细介绍。

以上简要介绍了基于统计的计算语言模型的一些基本概念和术语。下面介绍现有的常用统计语言模型和数据平滑技术。

4.2 现有的主要统计语言模型

本节将综述现有的主要统计语言模型。参照前文对语言模型作用的定义,根据对上下文空间不同的划分方法,将现有的语言模型分为 5 大类,并进行讨论。

4.2.1 上下文无关模型

上下文无关模型仅仅考虑当前词本身的概率,而不考虑该词所对应的上下文环境,这是一种最为简单,易于实现,而没有多大实际应用价值的统计语言模型。在这里列出,仅仅是作为其他更为复杂但实用的计算语言模型的一个基础。

一种最简单的上下文无关模型,是将所有的词视为具有相同的概率分布的随机变量的统计模型。设给定的词表为 V ,那么一个词的概率由下式给出:

$$P(w_i = w | c) = \frac{1}{|V|} \quad (4-4)$$

其中, c 是词的上下文,这个模型无需估计任何词的概率,因而也就不需要任何训练数据。如果根据在训练文本中词出现的频度估算词的概率,而不考虑该词的上下文环境,就可以给出一个能够更为精细反映词的统计特征的计算语言模型:

$$P(w_i = w | c) = P(w_i = w) \quad (4-5)$$

采用 MLE, $P(w_i = w)$ 由下式近似地给出:

$$P(w_i = w) = \frac{N_w}{N}$$

其中, N_w 表示词 w 在训练文本中出现的总次数, N 为训练文本的总词数。这类模型十分简单, 仅需估算出 $|V|$ 个词的概率。在实际应用中, 它被应用到一些商用语音识别系统中^[1]。因为它是 N -gram 模型当 $N=1$ 时的特殊情形, 所以它也被称为一元文法统计模型(unigram model)。

一元文法模型的优点是它仅仅需要非常少的训练数据, 它的缺点是没有考虑到上下文信息, 统计信息不充分, 因而在实际应用中系统精确度往往不高。

4.2.2 N 元文法模型

语言模型仅考虑了词本身的概率分布, 并没有考虑该词的上下文环境。从本小节开始, 后文将考察依赖于上下文环境的词的概率分布情况, 由此构成依赖于上下文环境的统计计算语言模型。

首先考察 N 元文法模型的一种简单情况, 即 $N=2$ 。此时的语言模型又称为 Bigram 语言模型。当读者看到一个句子的片段“他正在认真”时, 显然这并不是一个完整的句子。那么, 什么样的词可以接续在该片段的后面以构成一个完整的句子呢? 一种方法是通过查看“认真”一词, 来确定下一个词, 可以是“学习、工作、思考”等, 而不能是“美丽、我、中国”等。由此可知, 在自然语言中, 存在着后一个词的出现条件地依赖于前一个词的出现的现象。因此把一个词的上下文定义为该词的前一个词会对下一个词的出现具有较强的约束力, 尽管这一定义仅仅包含了上下文极少的一部分。语料库语言学研究表明, 许多词前面的词对于该词的出现具有很强的预测能力, 对于像英语这样有严格语序的语言更是如此。汉语的语序较英语灵活得多, 但这种约束关系依然存在。反映了这种约束关系的语言模型在实际的应用中非常有效。

如下的公式, 反映了这种条件依赖(或约束)关系:

$$P(w_i = w | c) = P(w_i = w | w_{i-1})$$

如果假定在一个语句中第 i 个词出现的概率条件地依赖于它前面的 $N-1$ 个词, 即将一个词的上下文定义为该词前面出现的 $N-1$ 个词, 这样的语言模型就叫做 N 元文法统计模型(N -gram), 如下式所示:

$$P(w_i = w | c) = P(w_i = w | w_{i-N+1}^{i-1}) \quad (4-6)$$

在 N 元文法统计模型中, 对于系统词表中的每一个 $N-1$ 元组都有一个概率分布与其相对应。由于共有 $|V|^{N-1}$ 个不同的 $N-1$ 元组, 因此这类统计模型有 $|V|^{N-1}$ 个不同的概率分布。对于每一个分布, 又必须估算 N 个参数, 因此这类统计模型共需估算出 $|V|^N$ 个参数。例如, 若 $|V|=10000$, $N=3$, 则必须计算出 10^{12} 个参数。对于大词表应用而言(例如 $N>10000$), 由于参数空间随着 N 的增大呈指数上升, 因此 N 不能取得太大, 一般取 2 或 3^[2]。

N -gram 模型的优点在于它包含了前 $N-1$ 个词所能提供的全部信息, 这些信息对于当前词的出现具有很强的约束力。它的缺点在于需要相当规模的训练文本来确定模型的参数。当 N 较大时, 模型的参数空间过大。

文献 3 指出: 自从十几年前在大词表语言识别系统中首次使用 Trigram 以来, 直到现在, Trigram 统计模型仍旧是在实际应用中表现最佳的语言模型, 并且成为许多其他语

言模型的重要组成部分。

4.2.3 N-POS 模型

N -gram 模型的主要缺点是它需要大规模的文本训练。另一方面,它假定一个语句中第 i 个词出现的概率,条件地依赖于它前面的 $N-1$ 个词,而事实并非如此,许多词出现的概率是条件地依赖于它前面词的语法功能。于是人们设想,将词按照其语法功能进行分类,由这些词类决定下一个词出现的概率。这样的词类称为词性(part of speech, POS),而相应的语言模型称为 N -POS 模型。

令 $G = \{g_1, g_2, \dots, g_t\}$ 为词类的集合。令 $g(w)$ 表示词 w 的词类,用 $g(w_{i_1}^{i_2})$, $1 \leq i_1 \leq i_2 \leq n$, 表示 $g(w_{i_1}), g(w_{i_1+1}), \dots, g(w_{i_2})$ 。

在 N -POS 模型中,一个词出现的概率条件地依赖于前 $N-1$ 个词的词类,因而,上下文定义为该词前面 $N-1$ 个词的词类。由如下公式所示:

$$P(w_i = w | c) = P(w_i = w | g(w_{i-N+1}^{i-1})) \quad (4-7)$$

这种模型有 $|G|^{N-1}$ 个不同的概率分布,共需估算出 $|G|^{N-1} \cdot |V|$ 个参数。例如,若 $|G| = 200$, $|V| = 10000$, $N = 3$,那么这种模型(Tri-POS 模型)共需估算出 8×10^{10} 个参数,其参数空间较同样规模的 N -gram 模型(Trigram 模型)减小了很多。

进一步,如果假设一个词的词类出现的概率条件地依赖于前 $N-1$ 个词的词类,而该词本身的概率依赖于该词所属的词类,就可以得到如下公式:

$$P(w_i = w | c) = P(g(w_i) | g(w_{i-N+1}^{i-1})) \cdot P(w_i = w | g(w_i)) \quad (4-8)$$

这种模型并没有改变上下文的定义,但是它只需估算出 $|G|^N + |G| \cdot |V|$ 个参数。例如,对于同样的 $|G|$ 和 $|V|$,它只需估算出 10^8 个参数,其参数空间进一步大大地减小了。

上述模型^[4]要求词类各不相交,但是,在自然语言中,广泛地存在着一词多类的现象。例如,“学习”一词可以是动词,也可以是名词。因此,词“学习”出现的概率应该是它作为一个名词的概率加上它作为一个动词的概率。针对这种情况,考虑到一个词的所有所属词类,可以给出如下公式:

$$P(w_i = w | c) = \sum_{g_j \in G} P(g(w_i) = g_j | g(w_{i-N+1}^{i-1})) \cdot P(w_i = w | g(w_i) = g_j) \quad (4-9)$$

N -POS 模型与 N -gram 模型有着密切的关系。考察 N -POS 模型的极端情况,即整个模型只有一个词类与每一个词都有一个词类的情况类似。如果 N -POS 模型只有一个词类,那么前 $N-1$ 个词类没有提供任何上下文信息,于是 N -POS 模型退化为 Unigram 模型;如果每一个词都有一个各不相同的词类,显然,这样的 N -POS 模型等价于 N -gram 模型。因此, N -gram 模型与 N -POS 模型因词类数量的不同而相互转化的关系如图 4-1 所示。

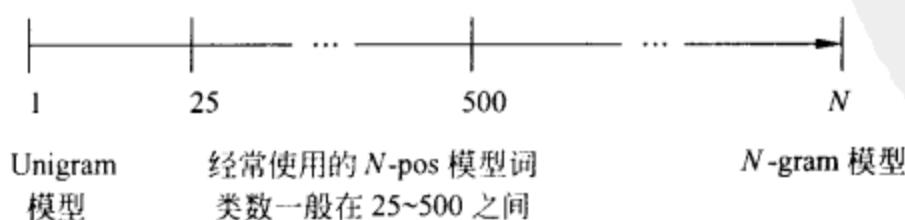


图 4-1 N -gram 模型与 N -POS 模型的关系

N -POS 模型的优点在于它需要的训练数据比 N -gram 模型少得多，并且模型的参数空间也要小得多。它的缺点是词的概率分布依赖于词性而非词本身，显然按照词类划分词的概率分布不如词本身的划分更加精细。因此，在实际应用（如语音识别）中，这类语言模型一般难以达到 N -gram 模型的精度^[5]。

4.2.4 基于决策树的语言模型

在前述的语言模型中，为建模所需估算的概率分布是固定的，与训练数据无关。例如，在 Tri-POS 模型中，每一个二元词类对都有一个概率分布，需估算出 $|G|^2$ 个概率分布。尽管存在许多二元词类对的概率分布相似的情况，但仍然要逐一枚举每一个二元词类对，并计算其概率分布。例如，在英语中，〔动词，冠词〕和〔介词，冠词〕的概率分布情况相近，却需要对这两种分别计算它们的概率分布情况，这不仅是参数空间上的浪费，而且相近的概率分布对提高语言模型的表现能力毫无助益。

统计决策树技术可以解决这一问题。近年来，统计决策树成功地应用在统计计算语言处理技术的许多领域^[6]。构造统计决策树的方法及统计决策树的有关细节可参见有关文献^[7]。本小节仅对统计决策树在构造语言模型技术中的应用作一简要描述。

在构造统计计算语言模型的应用中，一棵统计决策树包括所有的概率分布以及根据当前上下文查询其分布的机制。在这样的统计决策树中，通常包括两种类型的节点：中间节点和叶节点。每一个叶节点包括惟一的概率分布，而每一个中间节点包括关于上下文的一个提问。对于当前词的上下文，查询从根节点开始，由对根节点提问的不同回答进入子节点，直至叶节点。从而得到当前词上下文的分布信息。

为构造一棵统计决策树，必须预先定义一个关于上下文信息的问题集和一个评价问题优劣的函数。例如，可以参考统计决策树高度最小（查询所经过的中间节点最少、查询速度最快）等因素来构造这样的函数。首先，选取使评价函数值为最优的问题，构造一个根节点。然后递归地构造它的子节点，直到满足某种结束条件时构造过程终止。训练数据包括在给定上下文中出现的一些词，例如在训练语料库中出现一个词 w_i ，它的前一个词为 w_{i-1} ，这样的一个数据点为 $(w_{i-1}, w_i) = (\text{今天}, \text{是})$ 。在关于上下文信息的问题集中都是这样的提问：“它前一个词的词性是 g_j 吗？”这里 g_j 可以是任何词性标号。每一个这样的数据通过回答根节点上的提问而进入子节点，并回答子节点的提问进入下一级子节点，直到叶节点为止。在叶节点中，根据训练数据的频度来计算满足相应上下文条件的概率分布情况。

前述的所有语言模型，包括 Unigram, N -gram, N -POS 等，归根结底，都可以用统计决策树的形式表示出来。因此，统计决策树模型是一种更加通用的语言模型。它的优点在于分布树不是预先固定好的，而是根据在训练语料库中的实际情况确定。它的缺点是构造统计决策树的时空消耗非常大。

4.2.5 动态、自适应、基于缓存的语言模型

在前述的所有语言模型中，概率分布都是预先从训练语料库中估算好的，在语言模型的应用过程中，并不改变这些数据。因此，这样的语言模型可以称做静态语言模型^[8]。但

是,在自然语言中经常出现这样的现象。某些在文本中通常很少出现的词,在某一局部文本中突然大量地出现。文献 9 描述了因语料库中文本类型的不同而使词的频度分布不同的现象。能够根据词在局部文本中的出现情况,动态地调整语言模型中的概率分布数据的语言模型称为动态的、自适应的或者基于缓存的语言模型。

动态语言模型这一概念首先是由 Kuhn^[10] 提出的,在文献 11 中,这一概念得到了发展,并在文献 12 中得到了验证。由于这些方法都有一些相似之处,下文中仅以一种方法进行详细说明。

在文献 7 中, N 个最近出现过的词 w_{i-N}^{i-1} 存于一个缓存中,作为独立的训练数据。根据这些数据,可以估算出独立的 Unigram, Bigram, Trigram 等频度数据。通过某种平滑算法将这些频度数据结合起来,可以估算出一个单独的动态 Trigram 数据,用 $P_{\text{dyn}}(w_i | w_{i-2}^{i-1})$ 表示。对于在缓存中出现过的所有词,都可计算出一个动态 Trigram 数据。它们与静态语言模型中的频度分布数据 $P_{\text{sta}}(w_i | w_{i-2}^{i-1})$ 通过线性插值结合在一起,形成一个混合的动态自适应的模型,如下式所示:

$$P_{\text{com}}(w_i | w_{i-2}^{i-1}) = \lambda P_{\text{dyn}}(w_i | w_{i-2}^{i-1}) + (1 - \lambda) P_{\text{sta}}(w_i | w_{i-2}^{i-1}) \quad (4-10)$$

其中, $0 < \lambda < 1$,它的值可以通过 forward-backward 算法估算出来^[13],根据静态 Trigram 模型的规模以及 N 取值的不同, N 一般取值在 0.07~0.28 之间。

这种混合模型可以有效地避免数据稀疏问题,同时可以提高原静态模型的表现能力。在文献 7 中,一个孤立语音识别系统,经短文本测试错误率降低了 5%,经长文本测试错误率降低了 24%。

4.3 数据平滑技术

前文曾经提到,由于语言模型的训练文本 T 的规模及其分布存在着一定的局限性和片面性,许多合理的语言搭配现象没有出现在 T 中。例如,一个词串 w_{i-N+1}^i 没有出现在训练文本 T 中,根据公式(4-6),该词串对应的上下文条件概率 $P(w_i | w_{i-N+1}^{i-1}) = 0$,从而导致该词串所在的语句 S 的出现概率 $P(S) = 0$ 。这种情况通常称做数据稀疏问题(data sparseness)或零概率问题。对于一个给定的词典 V ,基于词的 N -gram 模型的模型参数空间为 $|V|^N$ 。显然,随着 N -gram 模型的元数(N 值)的增长,该模型的参数空间呈指数性增长,其面临的数据稀疏问题也越来越严重,极大地影响了模型的语言描述能力。根据 Zipf 统计定律^[14]可知,对于大量的低频词来说,无论训练语料库的规模如何扩大,其出现频度依然很低或根本不出现,无法获得其足够的统计特性提供给可靠概率估计。显然,单纯地扩大训练文本 T 的规模,不仅受到计算机时空复杂性的限制,而且对于数据稀疏问题所起的作用也十分有限。由于数据稀疏问题严重地制约着统计语言模型的性能,因此,用于解决该问题的数据平滑技术成了统计语言模型的研究热点。该技术对于采用最大似然估计得到的词的上下文概率分布作进一步调整,以确保统计语言模型中的每个概率参数均不为零,同时使模型中的概率分布更加趋向合理、均匀。

尽管数据平滑技术是构造统计语言模型的核心技术,但文献中缺乏对现有各种数据平滑方法的系统评价,只有 Nadas^[15~17],Katz^[18],Church^[19] 和 Peto^[20] 等在固定规模的语

料库上对极少数的数据平滑算法进行了比较。这种情况使得研究人员很难从众多的数据平滑技术中选择适合自己的方法。

4.3.1 数据平滑算法的评价标准

某种数据平滑算法的优劣,可以根据经过该平滑算法处理后的语言模型的性能度量出来。前文曾经提到,语言模型的性能一般根据它在测试文本上的交叉熵(cross entropy)或迷惑度(perplexity)进行度量^[21](见2.2.5节)。对于一个给定的语言模型 M 和一个测试文本 T , T 由句子序列 $(t_1 t_2 \dots t_{l_T})$ 组成,包含的总词数为 N_T ,令模型 M 采用第 m 类数据平滑技术之后得到的概率分布为 P_m 。第 m 类数据平滑技术的性能则可以通过计算模型 M 在 T 上的交叉熵 Entropy_m 进行评价。 Entropy_m 的计算公式如下:

$$\text{Entropy}_m = -\frac{1}{N_T} \sum_{i=1}^{l_T} \ln P_m(t_i) \quad (4-11)$$

在熵的基础上,可以计算出语言模型的迷惑度, $\text{Perplexity}_m = 2^{\text{Entropy}_m}$,作为数据平滑算法的另一种性能评价标准。语言模型的熵值或迷惑度值越小,说明该平滑算法的性能越好。

有时,研究人员也可以采用模型 M 的概率参数直接计算测试文本 T 的生成概率(对数形式),并用此概率度量语言模型的数据平滑算法的性能,具体公式如下:

$$F_M = \sum_{i=1}^{l_T} \ln P_m(t_i) \quad (4-12)$$

测试文本的生成概率越大,说明模型和测试集的符合程度越高,对应的数据平滑算法的性能越好。显然,上述3种数据平滑的度量方法是相互等价的。

4.3.2 常见平滑方法

1. 加法平滑

Lidstone, Johnson 和 Jeffreys 等提出了一种简单易行的数据平滑方法^[22],称为加法平滑(additive smoothing)。它的基本思想是:为了避免零概率问题,将 N -gram 模型中每个 N 元对的出现次数加上一个常数 δ ($0 < \delta \leq 1$),则相应的 N -gram 模型参数 $P_{\text{add}}(w_i | w_{i-n+1}^{i-1})$ 计算如下:

$$P_{\text{add}}(w_i | w_{i-n+1}^{i-1}) = \frac{c(w_{i-n+1}^{i-1}) + \delta}{\sum_{w_i} c(w_{i-n+1}^{i-1}) + \delta |V|} \quad (4-13)$$

Lidstone 建议 $\delta=1$ 。Gale 认为这种平滑技术的性能一般来说较差^[23]。

2. Good-Turing 平滑

Good-Turing 估计是许多数据平滑技术的核心^[24]。Good-Turing 估计的基本思想是:对于 N -gram 模型中出现 r 次的 N 元对 w_{i-n+1}^{i-1} ,根据 Good-Turing 估计公式,该 N 元对的出现次数为 r^* :

$$r^* = (r+1) \frac{n_{r+1}}{n_r} \quad (4-14)$$

其中, n_r 表示 N -gram 的训练集中实际出现 r 次的 N 元对的个数。那么, 对于 N -gram 中出现次数为 r 的 N 元对 w_{i-n+1}^{i-1} 的出现概率为

$$P_{\text{GT}}(w_{i-n+1}^{i-1}) = \frac{r^*}{\sum_{r=0}^{\infty} r^*} \quad (4-15)$$

由于 n_r 不能为零, Gale 和 Sampson 为此于 1995 年提出了一种用于 n_r 的平滑算法^[25]。Good-Turing 估计公式中缺乏利用低元模型对高元模型进行线性插值的思想, 它通常不单独使用, 而是作为其他平滑算法中的一个计算工具。

3. 线性插值平滑

线性插值平滑(linear interpolation smoothing)方法通常也称为 Jelinek-Mercer 平滑^[26]。该数据平滑技术主要利用低元 N -gram 模型对高元 N -gram 模型进行线性插值。Jelinek 和 Mercer 在 1980 年首先提出了这种数据平滑算法的思想, Brown 在 1992 年给出了线性插值的平滑公式:

$$P_{\text{int_erp}}(w_i | w_{i-n+1}^{i-1}) = \lambda_{w_{i-n+1}^{i-1}} \cdot P_{\text{ML}}(w_i | w_{i-n+1}^{i-1}) + (1 - \lambda_{w_{i-n+1}^{i-1}}) \cdot P_{\text{int_erp}}(w_i | w_{i-n+2}^{i-1}) \quad (4-16)$$

N -gram 模型可以递归地定义为由最大似然估计原则得到的 N -gram 模型和 $(N-1)$ -gram 模型的线性插值。为了结束以上的递归定义, 可以令 Unigram 模型为最大似然估计模型, 或者令 0-gram 模型为一个均匀分布模型 $P_{0\text{-ph}}(w_i) = \frac{1}{|V|}$ 。

对于插值系数 $\lambda_{w_{i-n+1}^{i-1}}$ 的估计, 一般可以采用 Baum-Welch 算法估计出来^[27]。其基本思想是: 使用经过数据平滑的模型概率参数, 计算一个测试集 H 的对数似然概率 $\log P(H)$, 当 $\log P(H)$ 为极大值时, 对应的 $\lambda_{w_{i-n+1}^{i-1}}$ 为最优值。因此可以求解 $\log P(H)$ 对应于每个 $\lambda_{w_{i-n+1}^{i-1}}$ 的偏导数, 令 $\frac{\partial \log P(H)}{\partial \lambda_{w_{i-n+1}^{i-1}}} = 0$ 。通过对该方程求解, 可以得到 $\lambda_{w_{i-n+1}^{i-1}}$ 的迭代计算公式:

$$\begin{aligned} \lambda'_{w_{i-n+1}^{i-1}} &= \frac{1}{c(w_{i-n+1}^{i-1})} \cdot \sum_{w_i} c(w_i) \\ &\cdot \frac{\lambda_{w_{i-n+1}^{i-1}} \cdot P_{\text{ML}}(w_i | w_{i-n+1}^{i-1})}{\lambda_{w_{i-n+1}^{i-1}} \cdot P_{\text{ML}}(w_i | w_{i-n+1}^{i-1}) + (1 - \lambda_{w_{i-n+1}^{i-1}}) \cdot P_{\text{int_erp}}(w_i | w_{i-n+2}^{i-1})} \end{aligned}$$

其中, $c(w_i)$ 代表词串 w_i 在测试集中出现的次数。由于需要计算的 $\lambda_{w_{i-n+1}^{i-1}}$ 参数众多, Jelinek 和 Mercer 建议对 $\lambda_{w_{i-n+1}^{i-1}}$ 参数空间进行桶式分类, 属于同一类的所有 $\lambda_{w_{i-n+1}^{i-1}}$ 被设置为相同值。Baul 进一步提出根据 $c(w_{i-n+1}^{i-1})$ 对 $\lambda_{w_{i-n+1}^{i-1}}$ 进行分类^[1], 属于同一类的 $\lambda_{w_{i-n+1}^{i-1}}$ 设置为相同值。 $c(w_{i-n+1}^{i-1})$ 越大, 设置 $\lambda_{w_{i-n+1}^{i-1}}$ 值越大。

4. 回退式平滑

Katz 提出的一种回退式数据平滑(backing-off smoothing)算法^[18], 该数据平滑算法的主要思想是: 当一个 N 元对 w_{i-n+1}^{i-1} 的出现次数 $c(w_{i-n+1}^{i-1})$ 足够大时, $P_{\text{ML}}(w_i | w_{i-n+1}^{i-1})$ 是 w_{i-n+1}^{i-1} 可靠的概率估计; 而当 $c(w_{i-n+1}^{i-1})$ 不是足够大时, 采用 Good-Turing 估计对其进行

平滑,将其部分概率折扣给未出现的 N 元对;当 $c(w_{i-n+1}^i) = 0$ 时,模型回退到低元模型,按照 $P_{\text{katz}}(w_i | w_{i-n+2}^{i-1})$ 比例来分配折扣给未出现的 N 元对的概率。综合上述思想,回退式平滑算法的平滑公式为

$$P_{\text{katz}}(w_i | w_{i-n+1}^{i-1}) = \begin{cases} P_{\text{ML}}(w_i | w_{i-n+1}^{i-1}), & c(w_{i-n+1}^i) \geq k \\ \alpha \cdot P_{\text{GT}}(w_i | w_{i-n+1}^{i-1}), & 1 \leq c(w_{i-n+1}^i) < k \\ \beta \cdot P_{\text{katz}}(w_i | w_{i-n+2}^{i-1}), & c(w_{i-n+1}^i) = 0 \end{cases} \quad (4-17)$$

其中, $P_{\text{ML}}(w_i | w_{i-n+1}^{i-1})$ 为最大似然估计模型。 $P_{\text{GT}}(w_i | w_{i-n+1}^{i-1})$ 为 Good-Turing 概率估计。常数 k 为一个常量,Katz 建议 $k = 5$ 。参数 α 和 β 保证模型参数概率的归一化约束条件,即 $\sum_{w_i} P_{\text{katz}}(w_i | w_{i-n+1}^{i-1}) = 1$ 。为了结束公式(4-17)的递归定义,可以令 Unigram 模型为最大似然估计模型,或令 0 阶模型为均匀分布模型。

令 $r = c(w_{i-n+1}^i)$, $n_r = \sum_{w_i} \delta(c(w_{i-n+1}^i), r)$, $N = \sum_{w_i} c(w_{i-n+1}^i) \circ P_{\text{GT}}(w_{i-n+1}^i)$ 的计算

公式如下:

$$P_{\text{GT}}(w_{i-n+1}^i) = \frac{n_{r+1}}{n_r} \cdot \frac{r+1}{N} \quad (4-18)$$

而 $P_{\text{ML}}(w_{i-n+1}^i)$ 采用最大似然估计原则进行估计:

$$P_{\text{ML}}(w_{i-n+1}^i) = \frac{r}{N} \quad (4-19)$$

根据 Good-Turing 估计公式,被折扣给所有未出现的 N 元对的次数之和等于 n_1 。显然, $\sum_{r>0} r \cdot n_r = N - n_1$ 。根据回退式平滑算法的定义,可以得出下面的公式:

$$\sum_{r=K}^{\infty} r \cdot n_r + \alpha \cdot \sum_{r=1}^{K-1} (r+1) \cdot n_{r+1} = N - n_1 = \sum_{r=2}^{\infty} r \cdot n_r \quad (4-20)$$

求解(4-20),可以得到 α 的值:

$$\alpha = \frac{\sum_{r=2}^{K-1} r \cdot n_r}{\sum_{r=2}^K r \cdot n_r} \quad (4-21)$$

将式(4-17)代入 $\sum_{w_i} P_{\text{katz}}(w_i | w_{i-n+1}^{i-1}) = 1$ 等式中,可得

$$\beta \cdot \sum_{w_i \in S_0} P_{\text{katz}}(w_i | w_{i-n+2}^{i-1}) = 1 - \alpha \cdot \sum_{w_i \in S_K} P_{\text{GT}}(w_i | w_{i-n+1}^{i-1}) - \sum_{w_i \in S_K} P_{\text{ML}}(w_i | w_{i-n+1}^{i-1}) \quad (4-22)$$

在式(4-22)中,

$$S_0 = \{w_i | c(w_{i-n+1}^i) = 0\}, S_K = \{w_i | 0 < c(w_{i-n+1}^i) < K\}, S_K = \{w_i | c(w_{i-n+1}^i) \geq K\}$$

将式(4-21)代入式(4-22)中,可以求出 β :

$$\beta = \frac{1 - \alpha \cdot \sum_{w_i \in S_K} P_{\text{GT}}(w_i | w_{i-n+1}^{i-1}) - \sum_{w_i \in S_K} P_{\text{ML}}(w_i | w_{i-n+1}^{i-1})}{\sum_{w_i \in S_0} P_{\text{katz}}(w_i | w_{i-n+2}^{i-1})} \quad (4-23)$$

显然,与线性插值平滑算法相比,回退式数据平滑算法的参数较少,而且可以直接确定,无需通过某种迭代重估算法反复训练,因此它的实现更为方便。

5. Kneser-Ney 平滑

Kneser 和 Ney 于 1995 年在绝对折扣平滑方法的基础上提出了一种新的平滑算法,一般称之为 Kneser-Ney 平滑^[28]。其基本思想是:当使用低元模型对高元模型进行插值平滑时,低元模型在混合模型中的影响较大,因此必须进行特殊处理。例如,在英语中,“Francisco”一词仅出现在“San”之后。由于频度 $c(\text{Francisco})$ 很大,概率 $P(\text{Francisco})$ 较高,所以如果采用绝对折扣平滑方法,则对于任意不等于“San”的上下文 h , $P(\text{Francisco}/h)$ 也会被赋予很高的概率,这显然是不合理的。因此在 Kneser-Ney 平滑中,他们认为 Unigram 概率 $p(w_i)$ 应与其不同的前向邻接词的数量成正比,而不是与其频度 $c(w_i)$ 成正比,即

$$P_{\text{KN}}(w_i) = \frac{N_{1+}(\cdot w_i)}{N_{1+}(\cdot \cdot)} \quad (4-24)$$

位置符号“·”代表在训练语料库中出现在词串 w_{i-n+1}^{i-1} 之后的任意一个词。其中

$$\begin{aligned} N_{1+}(\cdot w_i) &= |\{w_{i-1} \mid c(w_{i-1} w_i) > 0\}| \\ N_{1+}(\cdot \cdot) &= |\{(w_{i-1}, w_i) \mid c(w_{i-1} w_i) > 0\}| \end{aligned}$$

将公式(4-24)推广到高阶模型($N \geq 3$)中,可以得到

$$P_{\text{KN}}(w_i \mid w_{i-n+2}^{i-1}) = \frac{N_{1+}(\cdot w_{i-n+2}^i)}{N_{1+}(\cdot w_{i-n+2}^{i-1} \cdot)} \quad (4-25)$$

其中,

$$\begin{aligned} N_{1+}(\cdot w_{i-n+2}^i) &= |\{w_{i-n+1} \mid c(w_{i-n+1}^i) > 0\}| \\ N_{1+}(\cdot w_{i-n+2}^{i-1} \cdot) &= |\{(w_{i-n+1}, w_i) \mid c(w_{i-n+1}^i) > 0\}| \end{aligned}$$

综上所述,Kneser-Ney 平滑公式被定义为

$$\begin{aligned} P_{\text{KN}}(w_i \mid w_{i-n+1}^{i-1}) &= \frac{\max\{c(w_{i-n+1}^i) - D, 0\}}{\sum_{w_i} c(w_{i-n+1}^i)} + \\ &\quad \frac{D}{c(w_{i-n+1}^i)} \cdot N_{1+}(w_{i-n+1}^{i-1} \cdot) \cdot P_{\text{KN}}(w_i \mid w_{i-n+2}^{i-1}) \quad (4-26) \end{aligned}$$

其中, $P_{\text{KN}}(w_i \mid w_{i-n+2}^{i-1})$ 由公式(4-25)计算,而 D 的值由下式计算:

$$D = \frac{n_1}{n_1 + 2n_2} \quad (4-27)$$

$N_{1+}(w_{i-n+1}^{i-1} \cdot)$ 的定义如下:

$$N_{1+}(w_{i-n+1}^{i-1} \cdot) = |\{w_i \mid c(w_{i-n+1}^{i-1} w_i) > 0\}| \quad (4-28)$$

6. Witten-Bell 平滑

Witten-Bell 平滑算法是 Jelinek-Mercer 线性插值平滑算法的一个特例^[29]。该平滑算法与一般的线性插值平滑算法的唯一不同之处是线性插值中的插值系数 $\lambda_{w_{i-n+1}^{i-1}}$ 的设置方式。一般的线性插值平滑算法采用 Baum-Welch 重估算法训练 $\lambda_{w_{i-n+1}^{i-1}}$;而 Witten-

Bell 平滑算法采用如下的公式计算 $\lambda_{w_{i-n+1}^{i-1}}$:

$$1 - \lambda_{w_{i-n+1}^{i-1}} = \frac{N_{1+}(w_{i-n+1}^{i-1} \cdot)}{N_{1+}(w_{i-n+1}^{i-1} \cdot) + \sum_{w_i} c(w_{i-n+1}^i)} \quad (4-29)$$

其中, 符号 $N_{1+}(w_{i-n+1}^{i-1} \cdot)$ 的定义同式(4-28), $N_{1+}(\cdot)$ 表示括号里处于位置“ \cdot ”且出现次数大于零的词的个数。将 $\lambda_{w_{i-n+1}^{i-1}}$ 的值代入插值公式, 可以得到 Witten-Bell 平滑公式:

$$P_{WB}(w_i | w_{i-n+1}^{i-1}) = \frac{c(w_{i-n+1}^i) + N_{1+}(w_{i-n+1}^{i-1} \cdot) \cdot P_{WB}(w_i | w_{i-n+2}^{i-1})}{\sum_{w_i} c(w_{i-n+1}^i) + N_{1+}(w_{i-n+1}^{i-1} \cdot)} \quad (4-30)$$

4.4 隐马尔科夫模型

隐马尔科夫模型(hidden Markov model, HMM)^[30]于 20 世纪 70 年代在语音识别领域取得巨大成功, 之后被广泛应用到自然语言处理的各个领域, 成为基于统计的自然语言处理的重要方法, 是上世纪统计自然语言处理领域的重要成果之一。本节将介绍隐马尔科夫模型和它的 3 个基本问题, 以及经典算法。

4.4.1 随机过程

随机过程(stochastic process)是在同一样本空间 Ω 下的一个随机变量序列 $\epsilon_1, \epsilon_2, \epsilon_3, \dots, \epsilon_n$ 。这些变量可能出现的结果称为这个随机过程的状态。

例 1 以掷骰子为例, 不断掷出的骰子所形成的序列就可以看作一个随机过程。

例 2 某一时刻某商场内购物的人数可看作是一个随机事件, 可以用一个随机变量表示; 在某段时间商场内购物的人数情况可以用一列随机变量来表示, 可以看作是一个随机过程。

随机过程按照时间的连续与否可以分为连续随机过程(continuous stochastic process)和离散随机过程(discrete stochastic process)。上面的例 1 就是离散随机过程, 例 2 则是连续随机过程。以后用到的都是离散随机过程。

离散随机过程完全可以用以下两个条件刻画:

- ① 初始时刻的任意状态的初始概率为 $P(e_1 = x_j)$ 。
- ② 此后任意时刻的条件概率为

$$P(e_{t+1} = x_{i_{t+1}} | e_1 = x_{i_1}, \dots, e_t = x_{i_t})$$

其中 $t=1, 2, \dots$ 。

4.4.2 马尔科夫链和马尔科夫性

马尔科夫链(Markov chain)是一种特殊的离散随机过程, 这种随机过程的下一时刻的状态完全由当前时刻的状态来决定, 而与从前的状态无关, 即“未来”只与“现在”有关, 而与“过去”无关。用数学语言来表示就是

$$P(e_{t+1} = x_{i_{t+1}} | e_1 = x_{i_1}, \dots, e_t = x_{i_t}) = P(e_{t+1} = x_{i_{t+1}} | e_t = x_{i_t}) \quad (4-31)$$

称具有以上性质的随机过程为马尔科夫过程, 具有马尔科夫性(Markov property)。如果

随机过程的马尔科夫性与时间 t 无关, 则称这个随机过程是齐次(homogeneous)马尔科夫过程。以后讨论的都是齐次马尔科夫过程。

概率转移矩阵(stochastic matrix)是描述马尔科夫链的重要工具。给定一个具有 n 个状态 s_1, s_2, \dots, s_n 的马尔科夫链, 设从状态 s_i 到 s_j 的概率为 p_{ij} , 即 $p_{ij} = P(e_{t+1} = s_j | e_t = s_i)$, 则概率转移矩阵定义为

$$\mathbf{P} = \begin{bmatrix} p_{11} & \cdots & p_{1n} \\ \vdots & & \vdots \\ p_{n1} & \cdots & p_{nn} \end{bmatrix}, \quad p_{ij} \geq 0, \quad \sum_{j=1}^n p_{ij} = 1, \quad i = 1, \dots, n \quad (4-32)$$

通过概率转移矩阵可以直观地看到从状态 s_i 到状态 s_j 的概率 p_{ij} , 方便地计算出一个马尔科夫链的概率。也可以通过概率转移矩阵算出从状态 s_i 出发经过 n 个时刻到达状态 s_j 的概率 $p_{ij}^{(n)}$, 算法如下:

$$\mathbf{P}^n = \begin{bmatrix} p_{11}^n & \cdots & p_{1n}^n \\ \vdots & & \vdots \\ p_{n1}^n & \cdots & p_{nn}^n \end{bmatrix}, \quad p_{ij}^n \geq 0, \quad \sum_{j=1}^n p_{ij}^n = 1, \quad i = 1, \dots, n \quad (4-33)$$

很容易验证它的正确性, 这里不再给出证明。

4.4.3 马尔科夫模型

实际的马尔科夫模型是在马尔科夫链的基础上发展起来的。因为现实世界十分复杂, 实际问题往往不能直接转换成马尔科夫链来处理, 观察到的事件也不能与马尔科夫链的状态一一对应, 而是通过某种概率分布来与状态保持联系。这种实际事件与状态由某种概率分布来联系的马尔科夫链就是马尔科夫模型。实际上, 马尔科夫模型是一个双重随机过程。其中的一重随机过程是描述基本的状态转移, 而另一重随机过程是描述状态与观察值之间的对应关系。

下面我们形式地定义马尔科夫模型。

马尔科夫模型是一个五元组 $\mathbf{M} = \{\Omega, \Sigma, P, A, \Theta\}$, 其中 $\Omega = \{s_1, s_2, \dots, s_n\}$ 是所有状态的集合; $\Sigma = \{\sigma_1, \sigma_2, \dots, \sigma_m\}$ 是所有观察序列的集合; $P = \{p_{ij}\}_{n \times n}$ 并且 $p_{ij} = P(e_{t+1} = s_j | e_t = s_i)$ 是概率转移矩阵; $A = \{a_{ij}\}_{n \times m}$ 称为发射概率矩阵, 其中 $a_{ij} = P(\eta_t = \sigma_j | e_t = s_i)$ 是在状态 s_i 观察到 σ_j 的概率; $\Theta = \{\pi_1, \pi_2, \dots, \pi_n\}$ 是初始概率向量, 其中 π_i 是在状态 i 的初始概率。

4.4.4 隐马尔科夫模型

当马尔科夫模型中的“状态”对于外界来说是不可见的时候, 就转换成了隐马尔科夫模型。其实马尔科夫模型和隐马尔科夫模型的数学模型都是相同的, 只是因为现实问题的情况不同, 对模型的限定条件不同以及要解决的问题不同而分为两个模型。

隐马尔科夫模型有 3 个基本问题要解决:

- (1) 对于给定的观察序列, 它出现的概率是多大。
- (2) 对于给定的观察序列, 在状态序列未知的情况下, 根据现有的隐马尔科夫模型, 它最有可能的隐含状态序列是什么。
- (3) 给定观察序列, 怎样调整现在的 HMM 的模型参数, 使得此观察序列的概率最

大,即最好地“解释”了这一观察序列,这是 HMM 的训练问题。

对于以上 3 个问题,衍生出了 5 种算法。这 5 种算法都是动态规划算法。这些基本问题与它们的解决算法,都是从实际中得来的,与数学模型本身的联系不是十分密切。实际上,历史的发展恰恰是先提出了这些问题,又提出了相应的算法,最后才归结到统一的马尔科夫数学模型中去的。下面将逐一介绍这些问题和算法。

问题(1) 以上提到的隐马尔科夫模型的第一个问题,是对于一个给定的观察序列 $\mathbf{O} = \{\sigma_{k_1}, \sigma_{k_2}, \dots, \sigma_{k_T}\}$, 如果设产生它的状态序列是 $\mathbf{S} = \{s_{i_1}, s_{i_2}, \dots, s_{i_T}\}$, 求 $P(\mathbf{O})$ 是多少。注意,这时候的状态序列可以有多条。

可以这样计算:

$$P(\mathbf{O} | \mathbf{S}) = \prod_{t=1}^T P(\eta_t = \sigma_{k_t} | e_t = s_{i_t}) = \prod_{t=1}^T a_{i_t k_t} \quad (4-34)$$

$$P(\mathbf{S}) = \pi_{i_1} \cdot \prod_{t=2}^T p_{i_{t-1} i_t} \quad (4-35)$$

$$\begin{aligned} P(\mathbf{O} \cap \mathbf{S}) &= P(\mathbf{O} | \mathbf{S}) \cdot P(\mathbf{S}) = \left(\prod_{t=1}^T a_{i_t k_t} \right) \cdot \left(\pi_{i_1} \cdot \prod_{t=2}^T p_{i_{t-1} i_t} \right) \\ &= (a_{i_1 k_1} \cdot \pi_{i_1}) \cdot \prod_{t=2}^T p_{i_{t-1} i_t} \cdot a_{i_t k_t} \end{aligned} \quad (4-36)$$

将所有的状态序列产生的观察值的概率加起来,就是这个观察序列在这个隐马尔科夫模型下的概率:

$$P(\mathbf{O}) = \sum_{\mathbf{S}} P(\mathbf{O} \cap \mathbf{S}) = \sum_{s_{i_1}, \dots, s_{i_T}} (a_{i_1 k_1} \cdot \pi_{i_1}) \cdot \prod_{t=2}^T p_{i_{t-1} i_t} \cdot a_{i_t k_t}$$

这是一个非常直观的想法,但是这种算法的复杂度太高,需要 $O(2Tn^T)$ 的时间复杂度。当 $n=5, T=100$ 时,计算量达到 1072 指数型的复杂度,一般是不能应用在工程实际中的,何况还有复杂度更低的算法。下面介绍 forward 算法、backward 算法以及 forward-backward 算法。

对于 forward 算法,定义前向变量

$$\alpha_t(i) = P(O \leq t; e_t = s_i) = P(\eta_1 = \sigma_{k_1}, K, \eta_t = \sigma_{k_t}; e_t = s_i)$$

是在 t 时刻,到达状态 s_i 时所观察到的 $O \leq t$ 的概率。

$$\text{初始: } \alpha_1(i) = P(\eta_1 = \sigma_{k_1}; e_1 = s_i) = a_{i_1} \cdot \pi_i, i = 1, \dots, n \quad (4-37)$$

$$\text{递归: } \alpha_{t+1}(j) = \left[\sum_{i=1}^n \alpha_t(i) \cdot p_{ij} \right] \cdot a_{j k_{t+1}}, t = 1, \dots, T-1; j = 1, \dots, n \quad (4-38)$$

$$\text{终止: } P(\mathbf{O}) = \sum_{i=1}^n P(\eta_1 = \sigma_{k_1}, \dots, \eta_T = \sigma_{k_T}; e_t = s_i) = \sum_{i=1}^n \alpha_T(i) \quad (4-39)$$

以上算法的复杂度是 $O(n^2 T)$ 。

backward 算法和 forward 算法非常类似,定义后向变量 $\beta_t(t) = P(O > t | e_t = s_i) = P(\eta_{t+1} = \sigma_{k_{t+1}}, \dots, \eta_T = \sigma_{k_T} | e_t = s_i)$ 。可以看出,backward 算法的计算方向与 forward 算法的计算方向是相反的。

初始: $\beta_T(t) = 1, t = 1, \dots, n$

$$\text{递归: } \beta_t(i) = \sum_{j=1}^n p_{ij} \cdot a_{jk_{t+1}} \cdot \beta_{t+1}(j), t = 1, \dots, T-1; i = 1, \dots, n \quad (4-40)$$

$$\begin{aligned} \text{终止: } P(\mathbf{O}) &= \sum_{i=1}^n P(\eta_1 = \sigma_{k_1}, e_1 = s_i) \cdot P(\eta_2 = \sigma_{k_2}, \dots, \eta_T = \sigma_{k_T} | e_1 = s_i) \\ &= \sum_{i=1}^n a_{ik_1} \cdot \pi_i \cdot \beta_1(i) \end{aligned} \quad (4-41)$$

backward 的算法复杂度同样是 $O(n^2 T)$ 。

将以上介绍的 forward 算法和 backward 算法结合起来, 就是 forward-backward 算法:

$$\begin{aligned} P(\mathbf{O}) &= \sum_{t=1}^n P(\mathbf{O}; e_t = s_i) = \sum_{i=1}^n P(\mathbf{O} \leq t; e_t = s_i) \cdot P(\mathbf{O} > t | \mathbf{O} \leq t; e_t = s_i) \\ &= \sum_{i=1}^n P(\mathbf{O} \leq t; e_t = s_i) \cdot P(\mathbf{O} > t | e_t = s_i) = \sum_{i=1}^n \alpha_t(i) \cdot \beta_t(i) \end{aligned} \quad (4-42)$$

这个算法的时间复杂度仍然是 $O(n^2 T)$ 。

问题(2) 第二个问题是, 给定某个观察序列, 根据现有的隐马尔科夫模型找到一条概率最大的状态序列。

一种自然的想法是, 对于 $t, 1 \leq t \leq T+1$, 找出状态 s_t , 使 $P(s_t | \mathbf{O})$ 的值达到最大。按照这种想法, 定义

$$\gamma_t(t) = P(s_t = i | \mathbf{O}) = \frac{P(s_t = i, \mathbf{O})}{P(\mathbf{O})} = \frac{\alpha_t(i) \beta_t(t)}{\sum_{j=1}^N \alpha_j(t) \beta_j(t)} \quad (4-43)$$

然后取状态 s_t 满足条件

$$s_t = \arg \max_{1 \leq i \leq N} \gamma_t(i), \quad 1 \leq t \leq T+1 \quad (4-44)$$

但是这样做有弊端: 这种做法没有考虑状态之间的联系, 而只是寻找特定时刻的最佳状态, 找出的状态序列并不能保证是全局最优的。下面的 Viterbi 算法能够保证找出全局最优的状态序列。Viterbi 算法的基本原理如下。

在 Viterbi 算法中, 定义辅助变量

$$\begin{aligned} \delta_t(i) &= \max_{s \leq t-1} P(s \leq t-1, e_t = s_i; \mathbf{O} \leq t) \\ &= \max_{s_1, \dots, s_{t-1}} P(e_1 = s_1, \dots, e_{t-1} = s_{t-1}, e_t = s_i; \mathbf{O} \leq t) \end{aligned} \quad (4-45)$$

这个概率表示的是从 $1 \sim t$ 时刻, 以状态 s_i 结束时所观察到的发射序列的最大概率。我们的目的就是要找出当 t 是结束时刻时, 满足这个最大概率的状态序列。因此要记录对应于 $\delta_t(i)$ 的状态序列, 只需要记录 s_i 的前一个状态就可以了, 其他的状态可以由 $\delta_t(i-1)$ 递归得到。用变量 $\psi_t(j)$ 来记录 s_i 的前一个状态。

下面给出 Viterbi 算法的形式化描述:

$$(1) \text{ 初始: } \delta_1(i) = \pi_i \cdot a_{ik_1}, \quad i = 1, \dots, n \quad (4-46)$$

$$(2) \text{ 递归: } \delta_t(j) = [\max_i \delta_{t-1}(i) \cdot p_{ij}] \cdot a_{jk_t}, \quad t = 2, \dots, T; j = 1, \dots, n \quad (4-47)$$

$$\psi_t(j) = \arg \max_{1 \leq i \leq n} (\delta_{t-1}(i) \cdot p_{ij}), \quad t = 2, \dots, T; j = 1, \dots, n \quad (4-48)$$

这里与 forward 算法、backward 算法不同的是递归的时候取最大值,而不是取乘积。并且 Viterbi 算法是对两个变量进行递归,一个变量记录概率的最大值,是衡量状态序列的标准;另一个变量记录状态路径,是要找的结果。

$$(3) \text{ 终止: } P_{\text{final}} = \max_i \delta_T(i)$$

这是在最佳状态序列观察到的发射序列的概率值。

$$s_{k_T} = \arg \max_{1 \leq i \leq n} \delta_T(i)$$

这是得到的在最佳状态 T 时刻的状态值。下面通过对变量 $\psi_t(i)$ 的回溯,找出 $1 \sim T-1$ 时刻最佳状态的状态值。如果说算法前面是在辛勤耕耘,那么下面就要“收获”了。

$$(4) \text{ 回溯: } s_{k_t} = \psi_{t-1}(s_{k_{t+1}}), \quad t=1, \dots, T-1 \quad (4-49)$$

可以证明,Viterbi 算法的时间复杂度是 $O(n^2 T)$ 。

Viterbi 算法实质上是在一个有向图中找一条最佳的路径,这是隐马尔科夫模型中用得最多的一个算法,也是很通用的一个算法。很多时候人们并不很了解隐马尔科夫模型,却能够用 Viterbi 算法来解决实际问题。从历史发展的角度来看也是这样,并不是人们一开始想到用隐马尔科夫模型,而是在解决各种各样问题的时候产生出了 forward 算法、backward 算法、Viterbi 算法以及将要介绍的 Baum-Welch 算法,然后人们经过思考,总结,才将这些算法归结到统一的隐马尔科夫模型中去。我们在学习中,也要把重点放在应用上,先做到能够灵活应用,然后再细细体味这个模型的“精美”之处,这也是所谓的站在前人的肩膀上,能够更加深入地把握这个模型。

问题(3) 通常,在解决实际问题时,没有现成的隐马尔科夫模型可供使用,需要构造一个专门的隐马尔科夫模型。这时候我们所面临的问题常常是,手边有一些已经标记好的训练语料,还有一大部分未知的测试语料,因此要构建一个数学模型以对标记好的语料给出最好的解释,然后将这个模型应用到未知的测试语料中去。这就需要我们根据现有的语料来构造隐马尔科夫模型。

如果手中的语料既标记了观察序列又标记了状态序列(在实际应用中根据问题的不同,观察序列和状态序列所对应的概念也不同),那么可以根据隐马尔科夫模型中各个参数的定义直接统计出这些参数。但这样的语料很难得到,或者需要大量的人力去标记,或者需要资金购买。如果既没有人力,也没有资金,那不是什么都做不了么?通常我们手中的语料能够提供观察序列的信息,结合 Baum-Welch 算法就估计出隐马尔科夫模型中的各个参数。

对 Baum-Welch 算法定义辅助变量:

$$\epsilon_t(i, j) = P(e_t = s_i, e_{t+1} = s_j | \mathbf{O}) = \frac{P(\mathbf{O}; e_t = s_i, e_{t+1} = s_j)}{P(\mathbf{O})} \quad (4-50)$$

由定义式可以看出, $\epsilon_t(i, j)$ 的含义是,在已知观察序列的情况下, t 和 $t+1$ 时刻的状态分别是 i 和 j 的概率。根据前几节对 $\alpha_t(i)$ 和 $\beta_{t+1}(j)$ 的定义,可以这样计算 $\epsilon_t(i, j)$:

$$\epsilon_t(i, j) = \frac{\alpha_t(i) \cdot p_{ij} \cdot a_{jk_{t+1}} \cdot \beta_{t+1}(j)}{P(\mathbf{O})} = \frac{\alpha_t(i) \cdot p_{ij} \cdot a_{jk_{t+1}} \cdot \beta_{t+1}(j)}{\sum_{i,j} \alpha_t(i) \cdot p_{ij} \cdot a_{jk_{t+1}} \cdot \beta_{t+1}(j)} \quad (4-51)$$

根据问题(2)中定义的 $\gamma_i(t) = P(s_t = i | \mathbf{O})$,不难发现 $\gamma_i(t)$ 和 $\epsilon_t(i, j)$ 之间的联系:

$$\gamma_t(i) = P(s_t = i \mid \mathbf{O}) = \sum_{j=1}^n P(e_t = s_i, e_{t+1} = s_j \mid \mathbf{O}) = \sum_{j=1}^n \epsilon_t(i, j) \quad (4-52)$$

有了以上对 $\epsilon_t(i, j)$ 和 $\gamma_t(i)$ 的定义, 可以给出隐马尔科夫模型的各个参数的估计公式:

$$\bar{\pi}_i = \gamma_1(i) \quad (4-53)$$

$$\bar{p}_{ij} = \frac{\sum_{t=1}^{T-1} \epsilon_t(i, j)}{\sum_{t=1}^T \gamma_t(i)} \quad (4-54)$$

$$\bar{a}_{ij} = \frac{\sum_{t=1, \sigma_{k_t}=s_j}^T \gamma_t(i)}{\sum_{t=1}^T \gamma_t(i)} \quad (4-55)$$

式中, $\gamma_1(i)$ 表示初始状态是 s_i 的概率, $\sum_{t=1}^{T-1} \epsilon_t(i, j)$ 表示从状态 s_i 转移到状态 s_j 的概率, $\sum_{t=1}^{T-1} \gamma_t(i)$ 表示从状态 s_i 转移到其他状态的概率, $\sum_{t=1, \sigma_{k_t}=s_j}^T \gamma_t(i)$ 表示在状态 s_i 观察到 σ_j 的概率。

一般的做法是, 先随机地给隐马尔科夫模型的参数赋值, 再用 Viterbi 算法找出现有模型下的最优状态序列, 然后代入以上的公式重新估计隐马尔科夫模型的参数。这样的过程可以根据实际需要进行多次。不过 Baum-Welch 算法的收敛速度很快, 过度地应用这些公式会产生很多副作用。

Baum-Welch 算法能对于已经观察到的序列给出很好的解释, 但是对于那些没有观察到的序列, 不一定能给出好的解释, 这是 Baum-Welch 算法的缺点之一。可见, 如果有标注得很好的语料, 应尽量用隐马尔科夫模型中各个参数的定义来构建模型, 万不得已时, 再采用 Baum-Welch 算法。

隐马尔科夫模型有多种定义方式, 以上介绍的是最容易理解的一种。在实际应用中, 应该根据实际需要, 选用最适合的模型。

4.5 最大熵模型

4.5.1 模型介绍

最大熵方法最先由 Jaynes 于 20 世纪 50 年代提出的^[31], 后又有 Berger 等将其应用到了自然语言处理中^[32]。很多学者已经应用最大熵模型解决了自然语言处理中的文本分类^[33]、名实体识别^[34]、词性标注^[35]等问题。DellaPietra^[36]第一次将最大熵方法用于语言建模。

最大熵方法的基本思想是, 在满足系统当前提供的所有条件下寻求分布最均匀的模型, 即熵最大的模型。最大熵模型的主要原理为^[37]: 令 $x = x_1, x_2, \dots, x_n$ 代表由 n 个随机

变量组成的一个随机变量序列, $f_i(x)$ 为对应于第 i 个特征的约束函数, 通常采用索引函数作为一个特征的约束函数, 即对于第 i 个特征, $f_i(x)=1$, 否则 $f_i(x)=0$ 。从而最大熵模型需要解决的主要问题可以描述为寻找一个合适的概率分布 $P(x)$, 使其满足如下条件:

对于约束 $K_i, i=1, 2, \dots, m, P(x)$ 满足:

$$\sum_x P(x) f_i(x) = K_i \quad (4-56)$$

在所有满足上述条件的概率分布中, $P(x)$ 所对应的熵最大, 即

$$P^*(x) = \arg \max_{P(x)} H(P(x)) = \arg \max_{P(x)} \left[-\sum_x P(x) \log P(x) \right] \quad (4-57)$$

为了使问题有一个合理的解, 要求同时满足式(4-56)提出的所有约束的概率分布是存在的, 即各个约束应该相容, 同时还必须加上第 0 个约束函数, 即对所有 x , 有

$$f_0(x) = 1 \quad (4-58)$$

为了寻找满足上述条件的概率分布, 引入一个差别函数 $D(P \parallel Q)^{[38]}$:

$$D(P \parallel Q) \equiv \sum_x P(x) \log \frac{P(x)}{Q(x)} \quad (4-59)$$

其中 $Q(x)$ 为一个已知的概率分布函数。如果令 $Q(x)$ 表示均匀分布, 则上式等价于 $-H(P(x))+c$, 其中 c 为一个常数。求解熵最大的解转换为求解 $D(P \parallel Q)$ 最小的解。式(4-59)的最小值可以通过引入非确定性拉格朗日乘子^[39] 来求解, 即对所有的随机变量 x , 使

$$\frac{\partial [D(P \parallel Q)] - \sum_i \lambda_i [\sum_{x_i} P(x^i) f_i(x^i) - k_i]}{\partial [P(x)]} = 0$$

从而对所有 x 有

$$\log \left[\frac{P(x)}{Q(x)} \right] + 1 = \sum_i \lambda_i f_i(x) + \lambda_0$$

对上式两边取指数, 并作适当变换后可以得到所求的解, 为

$$P(x) = Q(x) [\exp \lambda_0] [\exp \{ \sum_i \lambda_i f_i(x) \}] \quad (4-60)$$

这里待定拉格朗日乘子 λ_i 的取值必须满足式(4-57)和式(4-58)给定的约束条件, 即

$$[\exp \lambda_0] \sum_x Q(x) [\exp \{ \sum_i \lambda_i f_i(x) \}] f_j(x) = K_j, \quad j = 0, 1, \dots, m \quad (4-61)$$

在最大熵建模过程中, 所要解决的两个主要问题是约束的确定及参数 λ_i 的求解。

由于不同的问题域会提出不同的约束, 因此前一个问题与特定的应用密切相关的, 这是在将最大熵模型应用到不同领域的时候人们需要解决的一个主要问题。对于参数 λ_i 的求解, 最常用的是迭代规整算法(GIS)^[40], 算法描述如下:

设定 λ_i 的初值, $i=1, 2, \dots, m$,

for $j=0$ to m do

保持 $\lambda_i, i \neq j$ 不变, 寻找满足第 j 个约束的 λ_j 的最优值 λ_j^* , 令 $\lambda_j = \lambda_j^*$
end for

如果所有的约束条件都得到满足, 则迭代终止; 否则, 返回。

4.5.2 模型评价

最大熵原理和 GIS 算法有以下主要的优点：

(1) 最大熵模型简单并且直观。它只是通过添加主要的约束来融合多种信息，没有其他的前提假设条件。

(2) 最大熵模型具有通用性。对任何事件空间的任何子集的概率估计都可以使用最大熵原理。

(3) 任何现存的语言模型的知识都可以被添加到最大熵模型。

(4) GIS 算法是一个逐步适应的过程，新的约束条件可以被随时添加到模型当中。

(5) 在现存的约束条件下，最大熵模型肯定存在唯一的解。

最大熵原理以及 GIS 算法有以下主要缺点：

(1) GIS 算法的计算复杂性太高。

(2) 虽然 GIS 算法一定能收敛，可得到模型的解，但是无法确定 GIS 算法迭代次数的理论上限。

有时候给模型添加一些与训练数据不一致的约束，会显得很有用。例如可以选择 Good-Turing 折扣方法，在这种情况下，最大熵模型就会失去与最大似然原理的一致性，更重要的是，这种不一致性会导致不能保证模型理论解的存在性、唯一性和收敛性。

4.5.3 最大熵语言建模

这一节介绍如何用最大熵模型来整合多种知识源来建立语言模型。

1. N-gram 模型整合

传统的 N -gram 语言模型中的知识，通常是 Unigram, Bigram, Trigram 等的最大似然估计。在用最大熵模型融合时，被替换为具有同样信息的 Unigram, Bigram, Trigram 约束，具体描述如下：

对 Unigram w_1 而言，其约束特征函数的形式如下：

$$f_{w_1}(h, w) = \begin{cases} 1, & w = w_1 \\ 0, & \text{其他} \end{cases} \quad (4-62)$$

该特征的数学期望 K_{w_1} 为

$$E_{\tilde{P}} f_{w_1} = \frac{1}{N} \sum f_{w_1}(h, w) \quad (4-63)$$

相应的约束条件为

$$E_P f_{w_1} = \sum_{h, w} \tilde{P}(h) P(w | h) f_{w_1}(h, w) = E_{\tilde{P}} F_{w_1} \quad (4-64)$$

其中， $\tilde{P}()$ 为经验概率。

对 Bigram 而言，其特征函数、数学期望以及约束条件与 Unigram 的形式基本上一样，定义如下：

$$f_{[w_1, w_2]}(h, w) = \begin{cases} 1, & w = w_2 \text{ 且 } h \text{ 以 } w_1 \text{ 为结尾} (h \text{ ends in } w_1) \\ 0, & \text{其他} \end{cases} \quad (4-65)$$

$$E_{\tilde{P}} f_{[w_1, w_2]} = \frac{1}{N} \sum f_{[w_1, w_2]}(h, w) \quad (4-66)$$

$$E_p f_{[w_1, w_2]} = \sum_{h, w} \tilde{P}(h) P(w | h) f_{[w_1, w_2]}(h, w) = E_{\tilde{P}} f_{[w_1, w_2]} \quad (4-67)$$

Trigram 的特征函数、数学期望和约束条件,读者可自行写出,在此不赘述了。

2. Triger 整合

为了在模型中融合一个触发对 $A \rightarrow B$ 的信息,定义触发对 $A \rightarrow B$ 的特征函数如下:

$$f_{A \rightarrow B}(h, w) = \begin{cases} 1, & A \in h, w = B \\ 0, & \text{其他} \end{cases} \quad (4-68)$$

设 $E_{\tilde{P}} f_{A \rightarrow B}$ 为上述特征的经验期望,则其约束条件为

$$\sum_h \tilde{P}(h) \sum_w P(w | h) f_{A \rightarrow B}(h, w) = E_{\tilde{P}} f_{A \rightarrow B} \quad (4-69)$$

触发对 $A \rightarrow B$ 可以根据互信息来求得,详见参考文献 41。

与线性插值及回退算法相比,最大熵模型的主要优点是可以将所有不同的概率分布以约束的形式提出,然后通过最大熵模型求解一个满足所有约束条件的解。由于不同的知识源在建模过程中就已经集成到了一起,而且在建模过程中考虑了不同知识源之间的相互作用,所以保证了所求得的概率分布的一致性。但这也增加了建模的复杂性,因而在面对大规模语料库的时候,必须考虑如何进行算法优化及特征提取等问题。

4.6 小结

本章介绍了基于统计语言模型的定义与常用的统计语言模型,其中包括上下文无关文法、N-Gram 模型、N-POS 模型、决策树模型、基于缓存的自适应模型,并对以上模型中所存在的数据稀疏问题进行了详细讨论,给出了常用的数据平滑算法。此外,对目前应用非常广泛的隐马尔科夫模型和最大熵模型进行了详细讨论,描述了隐马尔科夫模型中三个常用算法及最大熵模型中的迭代训练算法。与其他融合算法相比,最大熵模型的主要优点是可以将所有不同的概率分布以约束的形式提出,然后通过最大熵模型来求解一个满足所有约束条件的解。

参考文献

- 1 Frederick Jelinek. The development of an experimental discrete dictation recogniser. Proceedings of the I. E. E. E. 1985, 73(11)
- 2 R. Kneser, V. Steinbiss. On the dynamic adaptation of stochastic language models. In Proceedings of International Conference on Acoustics, Speech and Signal Processing, 586~589. Minneapolis, Minnesota, USA, 1993
- 3 F. Jelinek. Up from trigram! the struggle for improved language models. In European Conference on Speech Communication and Technology, 1037~1040, 1991
- 4 P. Dumouchel, V. Gupta, M. Lennig, and P. Mermelstein. Three probabilistic language models for

①计算机自然语言处理

- a large-vocabulary speech recognizer. In Proceedings of International Conference on Acoustics, Speech and Signal Processing, 513~516, 1989
- 5 J. P. Ueberla. Analysing a simple language model—some general conclusions for language models for speech recognition. Computer, Speech and Language, 1994
 - 6 E. Black, F. Jelinek, J. Lafferty, D. Magerman, R. Mercer, and S. Roukos. Towards history-based grammars: Using richer models for probabilistic parsing. In Proceedings of DARPA Speech and Natural Language Workshop, 134~139, Harriman, New York, 1992
 - 7 L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. Classification and Regression Trees. Wadsworth, Inc., Belmont CA, 1984
 - 8 F. Jelinek, B. Merialdo, S. Roukos, and M. Strauss. A dynamic language model for speech recognition. In Proceedings of DARPA Speech and Natural Language Workshop, 1991, 293~295
 - 9 S. Johansson. Some observations on word frequencies in three corpora of present day English texts. International Review of Applied Linguistics in Language Teaching, 1985, 67~68: 117~126
 - 10 R. Kuhn. Speech recognition and the frequency of recently used words: A modified markov model for natural language. In International Conference on Computational Linguistics, 348 ~ 350. Budapest, August 1988
 - 11 R. Kuhn and R. De Mori. Corrections to a cache-based natural language model for speech recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1992, 14: 691~692
 - 12 R. Kneser and V. Steinbiss. On the dynamic adaptation of stochastic language models. In Proceedings of International Conference on Acoustics, Speech and Signal Processing, 586~589, Minneapolis, Minnesota, USA, 1993
 - 13 Lalit R. Bahl, Frederick Jelinek, and Robert L. Mercer. A maximum likelihood approach to continuous speech recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1983, PAMI-5(2): 179~190
 - 14 G. Zipf. The Psycho-Biology of Language. Houghton. Mifflin.
 - 15 A. Natas. Optimal Solution of A Training Problem in Speech Recognition. In IEEE Transactions on Acoustics, Speech and Signal Processing, 1983, ASSP-33: 326~329
 - 16 A. Natas. Estimation of Probabilities in the Language Model of the IBM Speech Recognition System. IEEE Transactions on Acoustics, Speech and Signal Processing, 1984, ASSP-32(4): 859~861
 - 17 A. Natas. On Turing's Formula for Word Probabilities. IEEE Transactions on Acoustics, Speech and Signal Processing. 1985, 32: 1414~1416
 - 18 S. M. Katz. Estimation of Probabilities from Sparse Data for the Language Model Component of A Speech Recognizer. IEEE Transactions on Acoustics, Speech and Signal Processing. 1987, ASSP-35(3): 400~401
 - 19 K. W. Church and W. A. Gale. A Comparison of the Enhanced Good-Turing and Deleted Estimation Methods for Estimating Probabilities of English Bigrams. Computer Speech and Language, 1991, 5: 19~54
 - 20 MacKay, J. C. David and L. C. Peto. A Hierarchical Dirichlet Language Model. Natural Language Engineering, 1995, 1(3): 1~19
 - 21 Stanley F. Chen. Building Probabilistic Models for Natural Language. In: PhD thesis, the Subject of Computer Science. Harvard University Cambridge Massachusetts, May 1996

- 22 H. Jeffreys. *Theory of Probability*. Oxford, Second Edition; Clarendon Press, 1948
- 23 W. A. Gale and K. W. Church. Estimation Procedures for Language Context: Poor Estimates Are Worse Than None. *Proceedings in Computational Statistics, Ninth Symposium*. Yugoslavia: Dubrovnik, September 1990, 69~74,
- 24 I. J. Good. The Population Frequencies of Species and the Estimation of Population Parameters. *Biometrika*, 1953, 40(3 and 4): 237~264
- 25 W. A. Gale, Geoffrey Sampson. Good-Turing Frequency Estimation without Tears. *Journal of Quantitative Linguistics*, 1995, 2(3): 15~19
- 26 F. Jelinek and R. L. Mercer. Interpolated Estimation of Markov Source Parameters from Sparse Data. In *Proceedings of the Workshop on Pattern Recognition in Practice*. Amsterdam, Netherlands: North-Holland, May 1980
- 27 A. B. Poritz. Hidden Markov Models: A Guided Tour. *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, New York Hilton, New York City, April 1988, 1: 7~13
- 28 R. Kneser, H. Ney. Improved Backing-off for m-gram Language Modeling. In *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing*, Vo. 1: 1995, 1: 49~52
- 29 I. H. Witten and T. C. Bell. The Zero-Frequency Problem: Estimating the Probabilities of Novel Events in Adaptive Text Compression. *IEEE Transactions on Information Theory*, July 1991, 37(4): 1085~1094
- 30 Brigitte Krenn, Christer Samuelsson. *The Linguist's Guide to Statistics*. 1997
- 31 E. T. Jaynes. Information Theory and Statistical Mechanics. *Physics Reviews*, 1957, 106: 620~630
- 32 A. L. Berger, S. A. Della Pietra, V. J. Della Pietra. A Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics*, 1996, 22(1): 39~72
- 33 Kamal Nigam, John Lafferty, et al. Using maximum entropy for text classification. In proceedings of the IJCAI-99 workshop on information filtering, Stockholm, SE, 1999
- 34 Andrew Borthwick. A maximum entropy approach to named entity recognition. Ph. D dissertation New York University, September, 1999
- 35 Adwait Ratnaparkhi. A maximum entropy model for part-of-speech tagging. In proceedings of conference on empirical method in natural language processing, university of Pennsylvania, 1996
- 36 Stephen Della Pietra, Vincent Della Pietra, Robert Mercer and Salim Roukos. Adaptive Language Modeling Using Minimum Discriminant Estimation. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, 633~636, San Francisco, March 1992 Also published in *Proceedings of the DARPA Workshop on Speech and Natural Language*, Morgan Kaufmann, 103~106, February 1992
- 37 Frederick Jelinek. *Statistical Methods for Speech Recognition*. London: The MIT Press, 1997
- 38 S. Kullback. *Information Theory and Statistics*. New York: Wiley, 1959
- 39 F. B. Hildebrand. *Methods of Applied Mathematics*. Prentice-Hall, Englewood Cliffs, NJ, 1952
- 40 J. N. Darroch, D. Ratcliff. Generalized Iterative Scaling for Log-linear Models. *The Annals of Mathematical Statistics*, 1972, 43: 1470~1480
- 41 D. Magerman, M. Marcus. Parsing A Natural Language Using Mutual Information Statistic. In *Proceedings of the eighth National Conference on Artificial Intelligence*, 1990

基于语言理解的处理方法

本章将简要介绍基于语言理解的处理方法中的主要问题,包括词法分析和句法分析。借助于现有的词法分析和句法分析技术,阐述了语料库自动多级加工的方法、意义和应用。大规模语料库加工是基于统计的计算语言学的基础,同时也为后续应用篇介绍的相关技术提供统计数据基础。

5.1 引言

为了使计算机能够真正地理解语言,必须以某种语言模型描述自然语言的规律。常用的语言模型分为两种:一种是以基于知识的方法为代表的理性主义方法,该方法以语言学理论为基础,强调语言学家对语言现象的认识,采用非歧义的规则形式描述或解释歧义行为或歧义特性。另一种是以基于语料库的统计分析为基础的经验主义方法,该方法更注重用数学方法,从能代表自然语言规律的大规模真实文本中发现知识,抽取语言现象或统计规律。本章的第 5.2 节将介绍几种常见的基于语言理解的标注体系和句法分析方法,这些体系大多是语言学家提出并制定的,属于理性主义方法。5.3 节介绍的语料库多级加工技术则是经验主义处理方法的基础。

所有计算语言的方法必须依托于一定的语言学资源,而不同的语言学资源往往带有不同的标注集。这些资源的不同完全来源于语言学家对于语言现象的不同理解。5.2 节中将主要介绍目前在中文信息处理中比较通用的分类方法和标准,那些标准都是语言学家依据这种分类方法建立的一些语言学资源,包括北京大学的《现代汉语语法信息词典》^[1]、《同义词词林》^[2]、“知网(Hownet)”^[3]等。其中《现代汉语语法信息词典》和“知网”为机器可读的辞典。

句法分析是语言学理论和自然语言应用的一个重要桥梁。一个实用、完备、准确的句法分析是计算机真正理解自然语言的基础,而自然语言中存在的大量歧义现象是对句法分析的巨大挑战。随着语料库语言学的发展,句法分析的研究方向正在转向大覆盖度的分析。5.3.1 节将介绍基本的乔姆斯基(Chomsky)

文法体系,中文句法分析中常用的文法和算法,如概率上下文无关文法(probabilistic context free grammar)、上下文依存文法等,还将介绍目前比较热门的研究方向,如浅层句法分析(shallow parsing)或叫做组块分析(chunking)。

随着计算语言学界从理性主义向经验主义的转移,语料库语言学和语料库的自动大规模加工越来越受到人们的重视。1990年8月在赫尔辛基举行了第13届国际计算语言学大会(Coling'90),会前讲座的主题是“处理大规模真实文本的理论、方法和工具”,它确定了计算语言学今后一个时期的主要战略目标。英语语料库的建设起步较早,相继出现了不同规模和加工深度的语料库,如20世纪60年代的百万词次的Brown语料库、70年代的千万词次的Birmingham语料库及90年代几百万词次的Penn树库,目前研究的重点已从词性标注向句法标注、语义标注方向转移。这些语料库及其加工方法极大地推动了英语的自然语言处理的发展。相比之下,汉语语料库的研究相对欠缺,无论是语料规模、加工深度,还是加工方法、工具和基于语料库的知识获取等语料库的语言学理论,都落后于英语,这在一定程度上制约了汉语自然语言处理的发展。5.4节中将主要介绍中文语料的获取,自动加工以及加工后语料的一些具体应用。

5.2 常用的基于语言理解的分类标注体系

在基于语言理解的处理方法中,无论是词性标注还是词义标注,都可以被认为是一个分类的问题,例如是把“红”分到名词词性类中,还是分到形容词词性类中;把“打”分到“买”的语义类中,还是分到“打斗”的语义类中等。可见,分类体系对于基于语言理解问题是至关重要的,也是一个基础性的工作。

5.2.1 词性分类体系

语言学界认为,划分词类的依据有形态标准、意义标准和分布标准^[5]3种。对于汉语来说,形态标准和意义标准都是行不通的,所以只能根据词在句法结构里所担当的语法功能,也就是分布标准进行分类。同时,还要考虑标注集的3个确定原则:标准性、兼容性和扩展性。标准性是指尽量采纳当前已经成为各种语言的词性标准或正在成为词性标准的分类体系和标记符号;兼容性是指尽量使标注集的表示与已有的标注集的表示相互转化;扩展性是指对未解决的遗留问题或是未来可能的技术发展方向进行充分考虑,以便扩充和修改,并使扩充和修改对系统的整体影响代价最小。

词的词性分类是以语言学界的分类为基础的,可分为名词、时间词、处所词、方位词、动词、形容词、状态词、区别词、数词、量词、代词、介词、副词、连词、助词、语气词、象声词、叹词、前缀、后缀、成语、简称、习用语等。

目前常见的比较有影响的词性标注集有“八五”汉语语料库词性标记集所给出的24大类词类和北京大学计算语言学研究所的《现代汉语语法信息词典》关于39个词类的定义(如表5-1所示)。此外还有各个科研院所自行定义的词性分类体系,如哈尔滨工业大学机器翻译教研室^[4]的42个词性标记的词性标注集和清华大学^[5]的111个词性标记的词性标注集。应该说明的是,这些标注集在大类上差别不大,有的只是对某个大类进行了

更细微的语法划分。《现代汉语语法信息词典》是机读词典,北京大学在这套词性标注集的基础上,对人民日报语料进行了一年的人工标注。本文将以《现代汉语语法信息词典》标注集为例进行介绍。这里需要强调的是,词性标注集目前还没有国家制定的标准,不同的词性分类体系基本上来源于不同语言学家对语言现象的认识。在计算语言学不同的领域,人们对词性依赖的程度不同,处理的精度不同,所以对词性分类的粒度的定义也不同。在不同的应用中应该定义适用于自己系统的词性标注集。

表 5-1 北京大学计算语言研究所定义的汉语词性标注集

Ag	形语素	形容词性语素。形容词代码为 a, 语素代码 g 前面置 A
A	形容词	取英语形容词 adjective 的第 1 个字母
Ad	副形词	直接作状语的形容词。形容词代码 a 和副词代码 d 一起使用
An	名形词	具有名词功能的形容词。形容词代码 a 和名词代码 n 一起使用
B	区别词	取汉字“别”的声母
C	连词	取英语连词 conjunction 的第 1 个字母
Dg	副语素	副词性语素。副词代码为 d, 语素代码 g 前面置以 D
D	副词	取 adverb 的第 2 个字母, 因其第 1 个字母已用于形容词
E	叹词	取英语叹词 exclamation 的第 1 个字母
F	方位词	取汉字“方”的声母
G	语素	大多数语素都能作为合成词的“词根”, 取汉字“根”的声母
H	前接成分	取英语 head 的第 1 个字母
I	成语	取英语成语 idiom 的第 1 个字母
J	简称略语	取汉字“简”的声母
K	后接成分	
L	习用语	习用语尚未成为成语, 有点“临时性”的意思, 取“临”的声母
M	数词	取英语 numeral 的第 3 个字母, n, u 已有他用
Ng	名语素	名词性语素。名词代码为 n, 语素代码 g 前面置以 N
N	名词	取英语名词 noun 的第 1 个字母
Nr	人名	名词代码 n 和“人(ren)”的声母一起使用
Ns	地名	名词代码 n 和处所词代码 s 一起使用
Nt	机构团体	“团”的声母为 t, 名词代码 n 和 t 并在一起
Nz	其他专名	“专”的声母的第 1 个字母为 z, 名词代码 n 和 z 一起使用
O	拟声词	取英语拟声词 onomatopoeia 的第 1 个字母
P	介词	取英语介词 prepositional 的第 1 个字母
Q	量词	取英语 quantity 的第 1 个字母
R	代词	取英语代词 pronoun 的第 2 个字母, 因 p 已用于介词
S	处所词	取英语 space 的第 1 个字母
Tg	时语素	时间词性语素。时间词代码为 t, 在语素的代码 g 前面置以 T
T	时间词	取英语 time 的第 1 个字母

续表

U	助词	取英语助词 auxiliary 的第 2 个字母, 因 a 已用于形容词
Vg	动语素	动词性语素。动词代码为 v, 在语素的代码 g 前面置以 V
V	动词	取英语动词 verb 的第一个字母
Vd	副动词	直接作状语的动词。动词和副词的代码并在一起
Vn	名动词	指具有名词功能的动词。动词和名词的代码并在一起
W	标点符号	
X	非语素字	非语素字只是一个符号, 字母 x 常用于代表未知数、符号
Y	语气词	取汉字“语”的声母
Z	状态词	取汉字“状”的声母的前一个字母

5.2.2 词义分类体系

在语言处理技术中, 词义分析占有非常重要的地位。传统的句法分析技术仅仅着眼于语言的结构分析, 而理解语言关键在于理解语言所表达的意义。对于汉语这种缺乏语法形态的语言来说, 词义知识就更加重要了。在介绍词义分类体系前, 首先要搞清楚什么是词义。目前对词义还没有一个准确的定义。一般来说, 词义可定义为在一定的语言环境中所阐明的内容^[6]。想要描述一种词语所表述的意义, 常见的方式有 3 种: 一是同义词分类的方法。这种方法出现的时间最早, 所基于的概念也最简单, 一个词的词义完全可以用属于同一个集合中的其他词的词义来表示。二是基于语义成分分析的词汇语义学 (componential lexical semantics) (也可译为义素分析法)。这种方式把一个词的意义分析为更小概念原子的组合。不过, 定义一套概念原子却非易事。三是许多认知心理学家和计算语言学家所提出的基于“网”的形式描述词语意义的方式。随着这方面研究的增多, 越来越多的人意识到, 除了利用语义成分(义素分析法)表示语义, 还有利用关系来表示语义的基于关系的词汇语义学 (relational lexical semantics), 而且后者有可能替代前者。

首先, 介绍词义表示的第一种方法, 即同义词分类的方法。这里以《同义词词林》的词义分类体系为例子进行介绍。《同义词词林》的词义分类体系分为大、中、小 3 级, 共分 12 个大类、94 个中类、1428 个小类。其中每个大类以大写英文字母表示, 中类以小写字母表示, 小类则以阿拉伯数字表示。12 个大类包括: 人(A)、物(B)、时间与空间(C)、抽象事物(D)、特征(E)、动作(F)、心理活动(G)、活动(H)、现象与状态(I)、关联(J)、助语(K)和敬语(L)。《同义词词林》提供了一个汉语词义的分类体系, 但不足之处是很多常用词没有被收录。

词义表示的第二种方法是义素分析法。这里以董振东先生的《知网》为例进行介绍。《知网》通过对约 6000 个汉字进行的考察和分析抽取了一千多个义原。义原是知网中最基本的, 意义不能再分割的最小单位, 将其作为解释知识词典的基本要素, 其他的词条全都由这些义原来定义。如《知网》中医生的定义为 DEF = human | 人, # occupation | 职

位, * cure|医治, medical|医, 其中, 人、职位、医治、医等就是用来定义语义的4个原子义原, 分别表示为医生是人, 与职位相关, 是医治的施事者。

词义表示的第三种方法是利用“网”的形式来描述词语的意义。词汇所表示的概念相互之间存在着联系, 彼此构成了一个知识网络, 因此词义词典在整体上就是一个词义网络或者是知识网络。这种思想最早来源于WordNet^[7], 它是一个在线义类词典(online thesaurus)。WordNet词库的基础是它的同义词集合。当然, WordNet所包含的信息远不止此, 它还给出同义词集合(synonym set, 或synset)之间关系的许多详细信息。它的名词和动词都是分层级组织词语之间的语义关系, 在名词中, 有上下位关系(hyper-hyponymy), 在动词中有下位(tropomy)关系, 动词中的继承(entailment)关系有些类似名词中的整体部分(meronymy)关系, 等等。

依据WordNet的这种思想, “知网”中也定义了相应的关系, 但这种关系只存在于义原中, 并没有定义词与词的关系。这些关系主要包括:

- ① 上下位关系;
- ② 同义关系;
- ③ 反义关系;
- ④ 对义关系;
- ⑤ 部件-整体关系;
- ⑥ 属性-宿主关系;
- ⑦ 材料-成品关系;
- ⑧ 施事/经验者/关系主体-事件关系;
- ⑨ 受事/内容/领属物等-事件关系;
- ⑩ 工具-事件关系;
- ⑪ 场所-事件关系;
- ⑫ 时间-事件关系;
- ⑬ 值-属性关系;
- ⑭ 实体-值关系;
- ⑮ 事件-角色关系;
- ⑯ 相关关系等。

这些关系在知网中通过KDM语言来描述, 知网的KDM语言中使用了一些特殊符号表示不同的关系(见表5-2)。

表5-2 KDM语言的关系标识符号说明

符号	表达意义	例子
!	表示某一属性为一种敏感的属性	gas 气→! odor 气味
#	表示与其相关	sufferFrom 罹患→# medical 医
\$	“事件-内容(受事, 对象, 领有物)”关系	tool 用具→use 利用
%	“部件-整体”关系	room 房间→% house 房屋

续表

符号	表达意义	例 子
*	“事件-工具(施事)”关系	landVehicle 车→ * vehicleGo 驶
+	“隐性角色-事件”关系	document 文书→ + write 写
?	“材料-成品”关系	tree 树→? material 材料
(“概念-概念词”关系	安大略湖→(US 美国)
=	标注动态复杂概念的特殊动态角色	挨→ = manner(regular 定期)
{	动态角色的格标记	按→{accordingto}
&	表示指向	安→& electricity 电
@	表示空间或时间	鞍→@ sit 坐蹲

目前比较常用的汉语知识词典的知网不仅采用了义素分析法,同时采用了网络表示方法。比起 WordNet 来说,它并没有定义同义词集合,但两个词的相似度却可以通过义原的重合度和网络表示中的各种距离计算得到。目前已有很多关于知网的研究,读者可到知网的网站上查询相关的文献。

5.3 常用的基于语言理解的语法理论

为了确切地描述一种语言,人们自然首先要问:什么是语言?有人定义语言为某一字母表上符号串(句子)的集合。这种定义很不全面,因为它没有为句子提供一种结构性的描述,也没有提供一种能准确地确定什么是该语言中的句子而什么不是的方法。自然语言是无穷的,而且相当复杂。如果要用计算机来表示、理解、生成语言,就是要将语言进行形式化。所谓的形式化是指用一定的范畴和作用在这些范畴上的规则来描述它们。用有限的范畴和规则来描述和生成无限的语言,并且这些范畴和规则都是可形式化的,就是说可以通过数理符号来表示和推理,而不是存在于语言学家头脑中的知识。随着语言学家对语言的认识不断深入以及基于计算机工具对语言理解的需要,语言学家和计算机工作者共同合作,创建了很多语法理论以及基于这些理论的算法。其中比较有代表性的是基于语言知识认知的功能主义语法,如乔姆斯基转换生成语法(TG)、特尼埃尔的依存语法(配价语法,DG)、菲尔格的格语法、韩礼德代表的系统功能语法、Langacker 倡导的认知语法等。另一类是基于语言知识表达的形式主义句法,如短语结构语法(PSG)、扩充转移网络(ATN)、支配约束理论(GB)、功能合一语法(FUG)、词汇功能语法(LFG)、中心词驱动的短语结构语法(HPSG)、广义短语结构语法(GPSG)、范畴语法(CG)、链接语法(LG)、树邻接语法(TAG)等。如果从语法所基于的基础来看,可以分为两类,一类是基于语法范畴的语法,另一类是基于词的语法,如图 5-1 所示。

这里需要强调一下,一般的语法理论基本上来源于外国语言学家所进行的研究,但翻译成中文时有不同的译法,即有些中文资料翻译成语法,有些中文资料翻译成句法,还有些中文资料翻译成文法,3 个词基本上指同一个概念。在本书中,为了不与已有的命名产生冲突,我们尽量采用与目前流行的相一致的叫法,本书统一为“语法”。

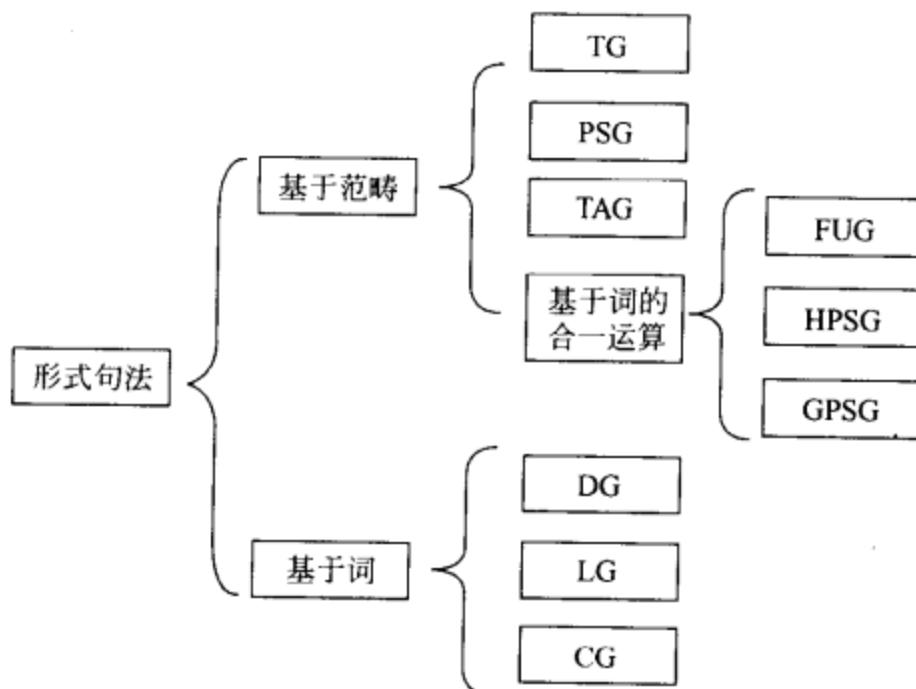


图 5-1 常用语法的分类体系

5.3.1 常用的语法理论

1. 短语结构语法

下面介绍一些形式化的语法,这些语法的特点是侧重于语言知识形式化的表示。20世纪50年代后期,美国语言学Chomsky提出了短语结构语法(phrase structure grammar,PSG),这是第一次提出的关于语言和语法的数学模型。

语法是用来精确且无歧义地描述语言构成方式的。语法描述语言的时候不考虑语言的含义。短语结构语法可表示成如(T, V, P, S)的四元组,其中 T 为终结符号集,是基本符号,不需要做进一步的定义; V 为非终结符号集,是需要定义的语法范畴,专门用于描述语法,它不能出现在最终生成的句子中; $S \in V$ 为文法的开始符号; P 为产生式规则集,均为非空的有限集,每条产生式规则集有如下的形式:

$$\alpha \rightarrow \beta$$

其中, $\alpha \in (TYV)^+$, $\beta \in (TYU)^*$,且 $\alpha \neq \beta$, $(TYU)^*$ 表示由 (TYU) 中的符号所构成的全部符号串的集合, $(TYU)^+$ 表示 $(TYU)^*$ 中除空符号以外的一切符号串的集合。

通常,用PSG处理自然语言时,它区分歧义的能力很差,所以需要对短语结构的形式体系增加某些约束。对这个问题,Chomsky提出了一种语法分类体系,包含4类语法,分别称之为0型语法、1型语法、2型语法和3型语法。

各类文法的定义以及它们之间的关系

(1) 无约束短语结构语法(0型语法)

若 P 中任一产生式都是这样的形式:

$$\alpha \rightarrow \beta \text{ 且 } \alpha \in (TYV)^+ \quad \beta \in (TYU)^*$$

其中,对 α, β 不加其他任何限制,则称 P 为0型语法或短语结构语法(简记为PSG)。

0型语法所描述的语言称之为0型语言。0型语法是Chomsky语法体系中生成能力

最强的一种形式。使用 0 型语法不足以描述自然语言,而且对程序设计语言的描述而言又太一般化,所以需要一定的约束。

(2) 上下文有关语法(1 型语法)

若一个 0 型语法 G 中所有产生式都具有如下的形式:

$$aAb \rightarrow a\beta b$$

其中, $a, b \in (TYU)^*$, $A \in V$, $\beta \in (TYU)^+$ 。之所以如此命名,是因为如果 A 前面出现符号 a ,后面紧接着符号 b ,则 A 可以重写为 β 。它的重写规则依赖于上下文。

(3) 上下文无关语法(2 型语法)

若在 1 型语法中所有的产生式有如下形式:

$$A \rightarrow \beta$$

其中, $A \in V$, $\beta \in (TYU)^+$, 每条产生式的左侧必须是一个单独的非终结符。规则应用时不依赖于 A 所处的上下文,所以叫做上下文无关语法(CFG)。

(4) 正则语法(3 型语法)

若 2 型语法中仅含有如下的产生式:

$$A \rightarrow aB, A \rightarrow a$$

其中, $A, B \in V$, $a \in T$, 则称为右线性语法。

若在 2 型语法中仅含如下的产生式:

$$A \rightarrow Ba, A \rightarrow a$$

其中, $A, B \in V$, $a \in T$, 则称语法为左线性语法。

一般地,把左线性语法和右线性语法统称为 3 型语法或者正则语法。一般的正则语法都可以用一个有向状态转移图(finite-state transition diagram)来表示,所以它又叫做有限状态语法。它所表述的语言可以用有限自动机来识别,一般用于描述程序设计语言的单词结构,不适宜用于描述自然语言。具体的应用可以参考相关资料或书籍。

从上述的介绍中可以看出:语法的型号越高,对规则附加的限制就越多,语法的生成能力就越弱。它们之间的关系是:0 型语法 \supseteq 1 型语法 \supseteq 2 型语法 \supseteq 3 型语法。

短语结构语法的好处在于可以方便地对一个句子进行结构的描述,但是由于它没有引入语义的成分,在生成正确句子的同时,也可能生成错误的句子。如“牛吃草”和“草吃牛”,在语法结构上都是一样的,短语结构语法没有办法加以区别。要解决这个问题,必须引入更精确的语义定义和描述,如定义“吃”的主语必须是一个动物等。而要引入这些定义必须要在词的基础上,对词这个语法范畴进行更多的定义。这样就引入了复杂特征集以及基于复杂特征集的合一运算。同时随着计算语言学的兴起,它们也都非常重视系统的逻辑结构和计算机上的可表示性。

短语结构语法的扩展。

(1) 广义短语结构语法(GPSG)

广义短语结构语法是 1985 年盖之达(Gerald Gazdar)等提出的。广义短语结构句法有两个特点:

① 只有一个句法对象,即谓短语结构;只有一个句法描写平面,即一般所说的表层结构。GPSG 从生成语法最早的短语结构语法出发,对短语结构语法进行扩充,使扩充后的

语法依然是短语结构语法,所以叫做广义短语结构语法。

② 语法和语义并重,在建立语法的同时,试图揭示出语法和语义之间的相互关系,并把这当作一种语法理论的中心目标。

③ GPSG 更致力于语法普遍性的探索。

④ GPSG 是一种高度严格的形式化方法^[8],它的总体结构如图 5-2 所示。

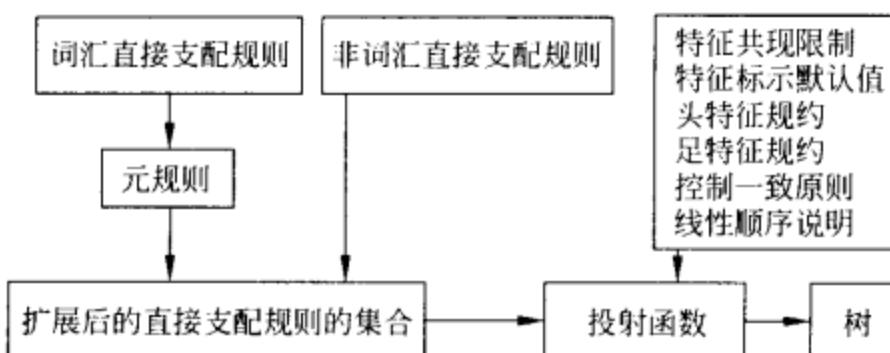


图 5-2 广义短语结构语法总体示意图

(2) 功能合一语法

功能合一语法是 1985 年美国计算语言学家马丁·凯依(Martin Kay)提出的,他采用了功能描述和合一运算的方法,所以叫做功能合一语法。简单地说,复杂特征集就是对事物从多方面进行描述,这些多方面特征的描述构成了一个复杂特征集。在功能合一语法中,复杂特征集叫做功能描述(functional description, FD)。一个功能描述由一组描述元组成,而描述元主要是一些带值的属性,叫做属性-值对。如下例:

类型 = 动词	类型 = 动词
词语 = run	数 = 单数
时态 = 现在	人称 = 3

合一运算是对复杂特征集进行运算的方法。在短语结构语法中,一般使用转换,如一个标记为 NP 的语法范畴转换为一个 N,一个标记为 N 的语法范畴转换成一个具体的名词等。当采用复杂特征集时,转换算法不再适用,于是采用了合一运算方法。合一就是把若干个功能描述合并成一个单独的功能描述,例如上例两个功能描述合并后,得到下面的功能描述:

类型 = 动词
词语 = run
时态 = 现在
数 = 单数
人称 = 3

合一运算不同于求并运算。在合一运算中,如果两个功能描述有相同的属性,但它们的值不相等,就说明这两个功能是不相容的,合一运算就会失败。功能合一语法把复杂特征集全面、系统地应用到语言描写中,具体说,就是它在词条定义、句法规则、语义规则和句子的描述中,都使用了复杂特征集,这是功能合一语法的最大特点^[9]。

(3) 词汇功能语法(LFG)

词汇功能语法是美国语言学家布莱斯南(J. Bresnan)和卡普兰(R. M. Kaplan)提出的,它对自然语言的计算机分析和生成产生了巨大的影响。与转换生成语法相同,词汇功能语法也有两个语法层次,即成分结构和功能结构,但在实现方法上具有明显的不同。在转换生成语法中,句子的深层结构和表层结构都表示为语法短语结构,论旨角色(thematic role)和表层结构之间的转换是建立在深层的语法短语结构上的。而LFG是一种非转换语法理论,它以无序的语法功能作为语法理论的基础。在LFG中语法-语义分析过程中,所依据的语言知识主要存储在机器词典的词汇项中,整个过程是词汇驱动的。LFG采用复杂特征集作为信息表示的基本手段,以合一运算作为语法-语义分析的基本算法。由成分结构到功能结构的映射均用合一这种简单的运算方式来完成。一般包括以下几个步骤:首先由语法规则和词法规则生成句子的成分结构,然后由成分结构求出功能描述,再由功能描述式构造功能结构。有合格的功能描述不一定能生成合乎语法的功能结构。从功能描述生成功能结构是通过合一运算和功能合格条件检查来实现的。功能合格条件由三部分组成:

① 惟一性(uniqueness) 在任何功能结构中,一个属性至多只能有一个值。惟一性检查体现在合一运算中。合一成功,则满足惟一性;合一失败,则不满足惟一性。

② 完备性(completeness) 当且仅当一个功能结构包含了它的谓词应该管辖的所有语法功能时,该功能结构是局部完备的;当且仅当一个功能结构内所有的子功能结构都是局部完备时该功能描述是完备的。

③ 一致性(coherence) 当且仅当一个功能结构所包含的可被管辖的语法功能都被一个局部谓词所管辖时,该功能结构是局部一致的;当且仅当一个功能结构内的所有子功能结构都是局部一致时,该功能结构是一致的。

(4) 中心语驱动短语结构语法(HPSG)

中心语驱动短语结构语法是波兰德于1984年在广义短语结构文法的基础上提出的,它基本上继承了广义短语结构语法的原则,是一种基于约束的语法理论。但是它同时又吸收了词汇功能语法、范畴语法的优点,强调了词汇在语言构成中的重要地位,强调中心语在语法分析中的作用,使整个语法系统由中心语驱动^[10]。HPSG把符号看作含有音系、句法、语义、话语及短语结构信息的结构性复合体,并采用分类特征结构(sorted feature structures)来描写。它具有如下特性:①分类性,即每个特征机构都有一个分类,表明该特征结构所描写的语言客体的类型;②特征合适性,即一个特征结构里面可以出现什么样的特征取决于该特征结构的分类,特征必须适合特征结构的分类;③特征完整性,适合一个特征结构的所有特征都必须出现在该特征结构中;④分类穷尽性,所有特征的值都必须是最具体的值分类。HPSG中的规则和语法普遍性原则都是对特征结构的制约,这些制约主要有词汇规定、语法普遍原则和直接支配格式^[11]。

2. 基于认知的语法

基于认知的语法理论侧重于从语言事实中发现范畴,建立规则。所有的这些语法理论无一不是语言学家或哲学家对语言现象的思考和认识。

(1) 依存语法(配价语法, GB)

依存语法是由法国语言学家 L. Tesniere 在其 1959 年所著的《结构句法基础》中提出的^[12]。它是基于这样一个事实：那就是不管是哪个学派的语法学说，都不得不正视句子中词与词之间的语法、语义联系这一事实。因此，反映动词对名词性成分的支配能力的“价”这一概念，是任何一种语法理论也回避不了的。当然，在理论处理上可以做出不同的安排。比如，依存语法把它放在非常核心的地位；格语法和系统功能语法把它置于比较重要的地位；而转换生成语法(GT)等形式语法则把它当作一种记载在词库的动词词条之下的词汇、语法特征，认为一个动词所必有的论元构成了动词的论元结构(argument struct)，基础句式(原子句)是动词的论元结构的一个投影。在此基础上建立起了投射原则、论旨原则。而这些原则都是 GB 理论的原则系统^[13]。综上所述，依存语法是表述词与词之间的一种最基本的联系。

在一般应用中，都是以动词为一个句子的中心来支配其他的成分，而它本身不受其他任何成分的支配，所有的受支配成分都以某种依存关系从属于支配成分，如图 5-3 所示。

Robinson J. J 在 1970 年提出了依存关系的 4 大公理，为依存关系奠定了基础，这 4 条公理^[14]是：

- ① 一个句子中只有一个成分是独立的，不受任何成分支配；
- ② 其他成分直接依存于另外的某一个成分；
- ③ 任何一个成分都不能直接依存于两个或两个以上的其他成分；
- ④ 如果 A 成分直接依存于 B 成分，而 C 成分在句中位于 A 成分和 B 成分之间，则 C 或者直接依存于 A，或者直接依存于 B，或者直接依存于 A 和 B 之间的某一成分。

依存语法的主要特点是：表示方法简洁，易懂。

依存语法生成的语法树不含非终结符，一个具有 N 个词的句子，生成依存语法树只有 N 个节点和 $N-1$ 条边。而利用短语结构语法得到的语法树由于含有非终结符，节点数大大超过 N ，大大超过 $N-1$ 条边。例如，图 5-3 所示的句子用结构语法表示时如图 5-4 所示。

依存语法关注的不是简单的顺序的同现概率。而是注重词与词的语义依存关系，例如，“一枝可爱的红色的花”。根据依存语法，可以直接提取出“一枝”和“花”的依存关系。如果按照简单顺序的同现概率，则提出“一枝”和“可爱”的同现概率了。

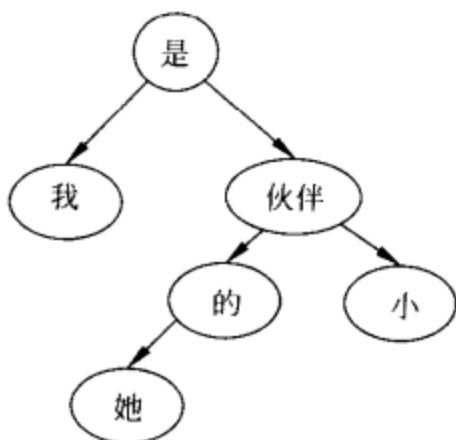


图 5-3 句子的依存语法表示

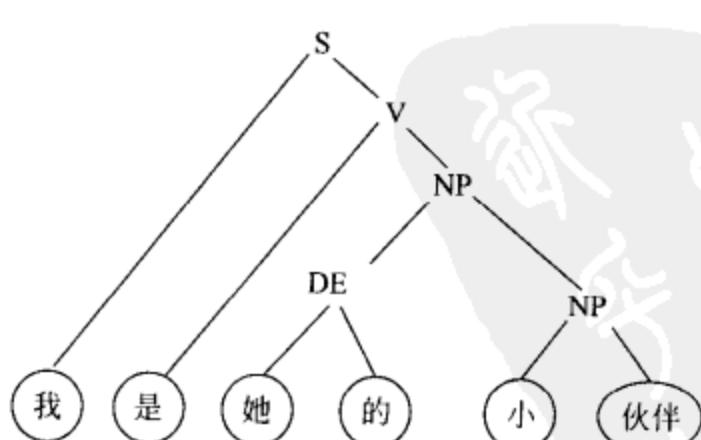


图 5-4 短语结构表示

(2) 格语法(case grammar)

格语法是 C. J. Fillmore 于 20 世纪 60 年代提出的。这里提出的格不是表层的语法格,而是深层的语义格,是一切语言的普遍现象^[15],如下面的例句:

Tom swam from one side of the river to the other side

动词 swam 与一个施事格 A:(Tom),一个源点格 S:(from one side of the river)以及一个目标格 G:(to the other side)并用。句子可以用深层框架的形式表示为[+…ASG]。Fillmore 从英语出发,最初提出 6 个格:施事、工具、与格、使成、处所、客体等。格语法偏重语义,但是忽略了词、短语之间的语法关系,这是由于语法的侧重点不同的缘故。短语结构语法属于语法平面,而格语法更加侧重语义平面。这两个平面都是语言分析所离不开的。

鲁川、林杏光两位先生根据汉语的特点,把格语法的说法改为格关系,并且提出了格关系的层次性,并根据这种理论,编写出了《动词大辞典》,这是格语法理论在汉语领域的一个应用成果。

(3) 系统功能语法

韩礼德提出的系统功能语法把语言看作一个与人类社会密切相关的职能体系,对这一体系的描述和解释都依赖于这个体系之外的语言的社会职能^[16]。系统功能语法的主要特点是外部性,而不是从语言体系本身去解释语言。韩礼德的系统功能语法由系统语法和功能语法两部分组成。前者说明语言系统表现为选择关系,后者阐明语言的 3 大功能,即概念功能、人际功能和语篇功能等^[17]。系统功能语法有 6 个核心思想,分别为纯理功能的思想、系统的思想、层次的思想、功能的思想、语境的思想和近似的思想^[18]。语言本身是个意义系统,语言的功能首先体现在语义功能上。

(4) 认识语法理论

认知语法理论是罗纳德·兰格克(Langacker Ronald)自 1976 年以来发展并阐述的一种语言机构理论。它以追求概括性为首要目标,力图找出一些基本的认知原则,对语言不同层次、不同方面存在的并行现象做出统一的解释,以收到以简化繁之效^[19]。它不是以数理符号形式表示的语言,这种表示一般可以认为是一种描述机制,也可以认为是一种生成机制,但是真实的语言却不是靠这些形式化的描述生成的。同样,语法也不是生成的,而是与人的认知能力有密切的联系,认知语法从认知的角度研究句法的理论。

以上简要介绍了 4 种语法,这 4 种语法有一个共同的目的,就是从不同的角度来认识语言的本质。由于角度的不同,提出的范畴和规则也不尽相同。前两个语法都是基于词汇的语法理论。对于词汇的功能,一种是用词与词之间的关系来描述的,另一种是用词本身的格来描述的。其实对于格是通过考察词在句子中所起的作用来定义的,而这些作用也都是从词与词之间的关系中提出来的。所以这两种方法都是基于对词汇的认识,都没有过多地考虑语言的结构,而是对语义进行了首要的考虑,都是从语言的内部来表示的,都是基于词汇这个范畴的。而后两种方法侧重于从语言的外部来认识语言,即一种从语言的外部功能来认识语言,另一种从人的认知能力来认识语言。这两种方法都有比较浓重的哲学色彩,不太适合直接在计算机上使用。

下面要介绍的语法侧重于语言的表示,有比较好的形式化表示,可以在计算机上比较方便地使用。

3. 其他语法

以上4种语法都是对短语结构语法的扩展，下面介绍3种与短语结构语法没有多大关系的语法。

(1) 扩充转移网络语法

扩充转移网络语法(ATN)是美国哈佛大学人工智能专家伍兹(W. Woods)提出的。它的出现对自然语言处理领域产生了重要的影响，并成功地应用于有限领域的问答系统中。最初的转移网络可以追溯到有限状态转移图(finite state transition diagram, FSTD)。FSTD由若干有限的状态以及从一个状态到另一个状态的弧组成。但是FSTD只能识别正则语言，所以用于表达自然语言是远远不够的。如果对FSTD赋予一种递归机制，就可以提高它的识别能力。扩充后的FSTD称为递归转移网络(recursive transition network, RTN)。可以证明，RTN具有识别上下文无关语言的能力，相当于一个上下文无关语法。RTN存在两点明显的局限性：

① 它只是一个识别器，而不是一个分析器，即只能指出输入句子是否合法，而不能进一步产生对句子的结构分析；
② 不能充分揭示句子成分之间的某种依赖性。

鉴于以上原因，ATN对RTN进行了三方面的扩充：

① 寄存器组，用来存放分析过程中产生的，关于句子或成分的结构信息。
② 测试，弧上除了用词类、语法结构等作为标记以外，还允许附加任意的测试。只有当这些测试满足条件后，该弧才能通过。
③ 动作，弧上还可以附加某些动作，当弧被通过时，动作便被执行。

ATN具有比较强的生成能力(相当于0型语法)但是它是过程性的，不是描述性的，同时，ATN对语法的过分依赖，限制了它处理某些合乎语义但不完全合乎语法的话语的能力^[20]。

(2) 树邻接语法

树邻接语法(tree adjoining grammar, TAG)是Joshi在1975年提出的。它是处于上下文无关语法和上下文有关语法之间的一种语法表示形式。他提出这种语法的原因是因为上下文过于简单，不足以限制一些不合法的语言现象；但是上下文有关语法的分析算法过于复杂，不适合实际应用。直观地说，在树邻接语法中，句子的结构也是用树形式来表示的，它的一种基本的操作是剪插。剪插是将树中的某一枝剪断，然后把另外一棵特殊的子树插入到树中被剪的位置，再将剪下的那一枝移接到子树的一个特定位置。除了剪插操作外，树邻接语法的形式化体系还包括两种原子树，一种是初始树，一种是辅助树。初始树的根节点都标以S，而叶节点都是终结符，或者是带有替换标志的非终结符。辅助树中还包括一个特殊的叶节点，这个特殊的叶节点与其根节点具有同样的标号。这个叶节点称为脚节点^[21]。为了把词汇信息引入树邻接语法，必须附加上词汇项的语法结构集合，并且要一个或一组用于组合结构的操作。也就是说在初始树和附加树中添加词汇锚点，并引入合一运算。

(3) 链语法

链语法(link grammar)由CMU计算机学院的Daniel Sleator、哥伦比亚大学音乐系

的 Davy Temperley 共同在 1991 年提出的。一部链语法就是一个单词的集合，其中的每个单词后面记录着各自的链接要求(linking requirement)。这些链接要求可以通过一系列链接子表达式(formula of connectors)指定。一个由单词组成的串(word sequence)，如果在单词之间存在满足下列条件的链(link)或者说，能够在单词之间画出一些链，并且这些链满足下面的条件，则这个单词串是链语法所定义的语言中的句子。首先这些链满足其中所有单词的链接要求。其次满足下面 3 条元规则：平面性(planarity)，这些链之间互相不交叉；连通性(connectivity)，这些链足以把所有的单词链在一起；顺序性(ordering)，公式中较左边的链接子必须与距离单词较近的单词连接。反之，公式中较右边的链接子必须与距离单词较远的单词连接。

D. Sleator 和 D. Temperley 提出了一个基于动态规划(dynamic programming)的分析算法或识别算法。算法以一种自顶向下的方式建立句子的链接集。自顶向下在这里的含义是，算法总是先建立较大的链，然后在这些链的范围内建立范围较小的链。

链语法的主要特点是：链语法是词汇主义(lexicalism)的语法系统，对于任何一个单词，词典中都详尽地描述了这个单词在句子中如何使用。所有词汇主义的语法系统的优点是：构造大型语法系统很容易，一个单词的描述只对包含这个单词的句子的分析产生影响。同时，允许逐步地建立一个大型语法系统。进而对于特殊的不规则的词语描述也很方便，这些词语描述在词典中单独占据一个条目。实际经验告诉我们，语言中任何现象都可以分为典型的和非典型的。描述典型规律的规则很容易，而描述非典型的规则则要麻烦得多。链语法用单词间的关系描述代替了成分描述。一般认为，词汇主义的语法系统易于用来构造统计模型，单词间的关系描述也使收集词汇之间的统计信息变得容易。

5.3.2 浅层语法分析技术

浅层语法分析(shallow parsing)也叫组块分析(chunking)，或者称为局部语法分析(partial parsing)。这种分析技术的提出源于下面的这种情况，自然语言处理层次可分为词、短语和句子 3 个不同的层次单位。而上述的语法理论可直接从词生成语法分析结果，并没有通过短语这个层次。这就不可避免地有大量的歧义问题，这些问题不仅会降低算法的运行速率，而且常常因此得不到正确的结果。实践证明，有必要将组块分析与语法分析分开处理，使基本的组块分析能够象词性标注一样取得较好的正确率，从而为后续的语法分析打下坚实的基础。

常见的浅层语法分析主要有两类：一类是基于统计的，另外一类是基于规则的^[22]。基于统计的方法主要有：

- ① 基于隐马尔科夫模型的方法；
- ② 互信息的方法；
- ③ Φ^2 统计方法；
- ④ 基于中心词依存概率的方法。

隐马尔科夫模型(HMM)是从语音识别中发展起来的一种统计技术。Church 进一步把 HMM 用于识别英语中简单的非递归的名词短语，他把短语边界识别问题划为在词类标记对之间插入 NP 的左边界“[”和 NP 的右边界“]”的问题。一对词类标记之间只有

[,],][,空等4种情况,最后一种情况可以进一步分为在NP内和在NP外两种情况。首先在有词性标注的语料中人工或半自动标注NP边界,以此作为训练数据;然后统计出任意一对词类标记之间出现以上5种状态的概率。统计得到的概率就成为短语边界标注的根据。这实际上把短语边界识别问题变成了一个与词性标注类似的问题。

互信息(mutual information)是信息论中的一个概念,它用来度量一个消息中两个信号之间的相互依赖程度。二元互信息是两个事件的概率函数:

$$MI(X, Y) = \lg \frac{P(X, Y)}{P(X) \cdot P(Y)} \quad (5-1)$$

把词类序列看成一个随机事件,这样就可以计算一对词类标记之间的互信息。如果X和Y在一起出现的机会多于它们随机出现的机会,则 $P(X, Y) >> P(X) \cdot P(Y)$,即 $MI(X, Y) >> 0$;如果X和Y是随机分布的,则 $P(X, Y) \approx P(X) \cdot P(Y)$,即 $MI(X, Y) \approx 0$,如果X和Y是互补分布的,则互信息 $MI(X, Y) << 0$ 。互信息越高,X和Y组成短语的可能性就越大。为了确定句子中短语的边界,需要更多的上下文,Magerman和Marcus提出了广义互信息的概念^[23],具体推导可参看参考文献。

用 χ^2 检验的方法来确定短语的边界是由Chen和Lee提出来的^[24],其主要的方法如下:对于两个词 W_1 和 W_2 ,可以建立如表5-3所示的联立表。

在表5-3中,a表示串 W_1, W_2 出现的次数,b表示不在 W_1, W_2 中的 W_1 出现的次数,c表示不在 W_1, W_2 中的 W_2 出现的次数,d表示既不是 W_1 又不是 W_2 的词的次数。 $a+b$ 是 W_1 的出现次数, $a+c$ 是 W_2 的出现次数, $c+d$ 是非 W_1 的总词次, $b+d$ 是非 W_2 的总词次, $N=a+b+c+d$ 表示语料库中的总词次。根据如表5-3所示的联立表。 χ^2 统计量定义如下:

$$\chi^2 = \frac{(a \cdot d - b \cdot c)^2}{(a+b)(a+c)(b+d)(c+d)} \quad (5-2)$$

当 $a=0$ 时, χ^2 近于0,即当 W_1, W_2 从不同时出现时, χ^2 取极小值。当 $b=c=0$ 时, $\chi^2=1$,即当 W_1, W_2 总是同时出现时, χ^2 取极大值。 χ^2 越大,说明 W_1, W_2 同时出现的机会越多,相反,则说明 W_1, W_2 同时出现的机会越少。

表5-3 词 W_1 和 W_2 联立表

	W_2	$!W_2$	Σ
W_1	a	b	$a+b$
$!W_1$	c	d	$c+d$
Σ	$a+c$	$b+d$	$a+b+c+d$

Collins提出了一种基于分析树中中心词之间依存概率的统计分析算法,该方法的要点是,把分析树归结为一个非递归的基本名词短语集合及依存关系的集合。在这些依存关系中,基本名词短语除了中心词外其他词都被忽略,所以依存关系就是基本名词短语的中心词与其他词之间的依存关系,依存概率可以通过树库中的统计得到,分析算法是一个自底向上的线图分析器,可利用动态规划来查找训练数据中所有的依存关系空间。这种方法是以大规模树库为基础的,而且以完全语法分析为目标;以基本名词短语识别为前

提,利用具体词之间的依存概率,把短语归结为中心词。这个研究思路对于部分语法分析也是很有借鉴意义的。

基于规则的方法就是根据人工书写的或半自动获取的语法规则标注出短语的边界和短语的类型。根据标注策略的不同可以把规则方法分为两种:①增加句法标记法;②删除句法标记法。规则的使用相对简单,但是规则的获取却是比较困难的。一般规则都是人工书写的,但是人工书写和总结这些上下文限制规则是十分困难的,所以下面介绍一些从语料库中自动获取这些语法规则的方法。在一种方法是 Eric Brill 提出的基于转换的错误驱动的学习方法^[25],另一种方法是 Cardie Pierce 提出的基于实例的方法。

(1) 基于转换的错误驱动的学习方法

① 初始标注。把训练语料中所有的基本名词短语的标记去掉,用一个简单的初始标注程序标注出训练集中可能的基本名词短语。把这个结果作为系统的底线。

② 形成候选规则集。在每个初始标注错误的地方,规则模板用来生成候选规则,规则的条件就是词的上下文环境,动作就是改正错误标记所要做的动作。

③ 获取规则。把候选规则集中的每条规则分别运用于初始标注的结果,选出得分最高的规则。把这条规则运用于初始标注的结果作为下一轮循环的基础,并把这条规则作为规则序列中的第一条规则输出。重复以上过程,直到得分最高的规则的得分为 0 或低于某个值为止。

(2) 基于实例的方法

基于实例的方法是把标注好的短语信息的语料库分为两个部分,一部分用于训练,另一部分用于剪枝。首先从训练的语料中得到一组名词短语的组成模式规则;然后把得到的这些规则应用到剪枝的语料中,对这些规则进行打分。比如,如果一个规则识别出一个正确短语得 1 分,识别出一个错误短语减 1 分,则根据每条规则的总得分情况对规则进行删减,去掉那些得分少的规则。

5.4 语料库多级加工

语料库语言学(corpus linguistics)是以语料库为基本知识源来研究自然语言规律的学科,其中语料库加工的理论、方法和工具及基于语料库的知识获取是语料库语言学研究的主要内容。语料库(corpus)是按照一定的原则组织在一起的真实的自然语言数据(包括书面语和口语)集合,主要用于研究自然语言的规律,特别是统计语言学模型的训练以及相关系统的评价和测试。

国外的语料库研究起步早。1961 年,世界上第一个大规模电子语料库——布朗语料库(Brown corpus, BC)的出现,标志着语料库语言学的诞生。BC 包含 10^6 词的美国英语,并在 1979 年采用 TAGGIT 结合人工校对方法进行了词性标注。20 世纪 70 年代初,英国 Lancaster 大学、挪威 Oslo 大学和 Bergen 大学联合开发了 LOB(London-Oslo-Bergen corpus)。LOB 含有相同数量的英国英语,并也在 1979 年采用 CLAWS 系统进行了词性标注。近年来,随着基于语料库的统计方法在语音识别领域的巨大成功和计算语言学方法论的战略转移,各国政府、科研机构和一些大公司纷纷介入语料库研究,语料库

语言学作为计算语言学的一个独立分支学科,越来越为世人所瞩目,并得到迅速发展。语料库建设已从单一的研究行为转向有计划的工程行为,如美国的 ACI/DCI 计划、欧共体的 ECI 计划等,并向更大规模、更深的加工层次发展,相继产生一批规模大、有影响的不同层次和类型的语料库,如 Penn 树库。

汉语语料库的建设起步较晚,直到进入 20 世纪 90 年代,国内的一些研究机构才开始根据各自不同的用途建立了相关的语料库,如清华大学按系统原则收集 5000 万汉字的原始语料库、中国台湾中央研究院的 200 万词次的带有词性标注的汉语平衡语料库等。虽然汉语的语料库建设近来有了长足的发展,但同英语相比,在标注体系的标准化、系统化、加工的深度以及资源的开放性等方面还存在着明显的不足,这已在一定程度上成为汉语自然语言处理的制约因素之一。特别值得一提的是,近年来在国内外相继出现了专门研究汉语树库的组织,如中国台湾中央研究院的中文信息处理组 (Chinese knowledge information processing group, CKIP)、马里兰大学 CLIP 实验室的汉字森林 (Chinese forest)、美国宾夕法尼亚大学 XTAG 计划,其中对汉语语料库特别是汉语树库的研究还在继续。

5.4.1 语料库的多级加工

所谓语料库标注或加工就是对电子语料(包括书面语和口语)进行不同层次的语言学分析,并添加相应的“显性”解释性语言学信息的过程^[4]。语料库是否能成为一个名副其实的语言知识库,不仅与其本身规模、分布的多样性和均衡性有关,更重要的是与其加工的深度层次有关。通常只有经过不同层次加工的“熟”语料才可真正用于自然语言处理系统的训练、测试和评价。对于未经加工的生语料 (raw corpora),由于语言学信息是“隐性”存在,所以所能直接获取的是非常初级的有关字或词的统计知识,只有在对语料库进行一层一层的标注之后,使语料库由“生”变“熟”,使孕育在生语料中“隐性”语言学知识“显性”化,方能从语料库中真正获取更多、更有用的语言学知识。层次越高,所能获取的语言学知识越丰富,加工难度越大。

与不同层次的自然语言分析相对应,语料库的加工主要包括词性标注、句法标注、语义标注、言语标注和语用标注等,由于汉语书写的特殊性,汉语的语料加工还包括分词。如表 5-4 所示,目前大多数的语料库加工集中在词性标注上。句法标注正处于一个迅速发展时期,而其他层次的语料加工,如语义、言语等,还处于起步阶段。

表 5-4 英语和汉语语料库标注情况对比

语言层	英语标注情况	汉语标注情况
词性 (part of speech)	基本实用化	迅速发展
句法 (syntactic)	迅速发展	大多在发展
语义 (semantic)	存在一些, 大多在发展	起步
言语 (discourse)	很少, 起步	无
语用 (pragmatic)	很少, 起步	无

目前,语料库的加工方式主要有 3 种方式,即人工、自动和人机结合的半自动方式。由于自然语言处理技术的限制,自动加工方式会给语料库标注带来一些错误,而人工加工

方式非常昂贵,需要大量的人力资源。相比之下,半自动加工方式则兼顾两者的优点。人机结合的方式有两种:一种是由计算机自动选择语料库中需要人干预的自动加工方式不能解决的部分,从而减少人的工作^[25];另一种方式则是先由计算机对加工的语料进行自动加工,然后由人工校对^[4]。

歧义消解研究与语料库的多级加工关系紧密,如图 5-5 所示。一方面,高性能的歧义消解技术是实现语料库加工自动化的关键。由于自然语言的歧义问题具有普遍性,所以在每个语言层次的语料库加工中几乎都要碰到相应的歧义问题。从某种意义上讲,语料库的多级加工实际上是一个面向真实文本的自然语言多级歧义消解过程。另一方面,语料库特别是经过加工的语料库又为歧义消解提供了资源支持。大量的实验表明,基于语料库的方法对于歧义消解非常有效,而这类方法的效果在很大程度上依赖于训练语料库的规模、分布和加工深度。语料库加工的层次越高,所能提供的语言学信息越丰富,越有利于歧义消解水平的提高。

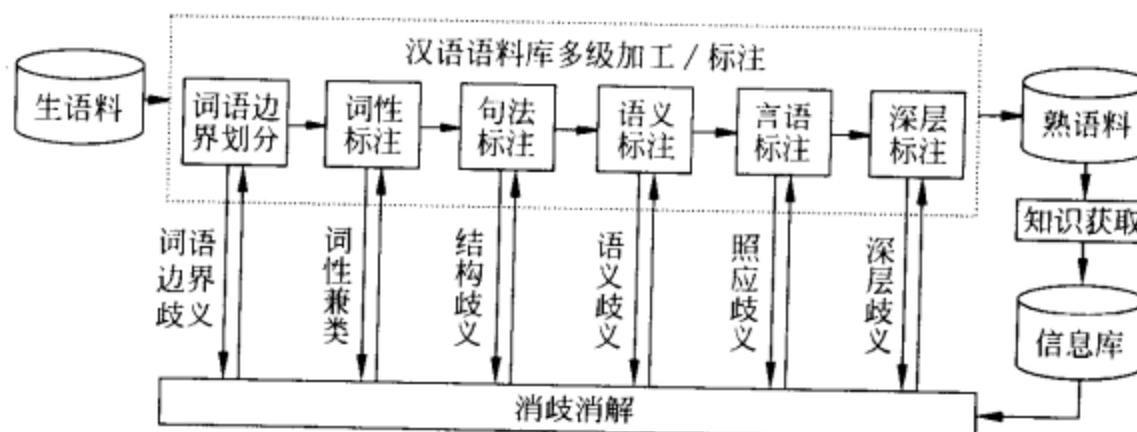


图 5-5 语料库多级加工与歧义消解的关系

5.4.2 分词

词是自然语言理解中有意义的最小构成单位,但是汉语文本是基于字的,词与词之间没有显性的界限标志,因此词语边界识别是汉语语法分析中首先要解决的问题之一。汉语自动分词就是把没有分割标记的汉语字串转换到符合语言实际的词串。目前困扰分词的两个主要问题就是歧义消解和未登录词的识别问题。由于本书前文已经对分词作了详细的介绍,这里就不再赘述。但有一点应该说明,汉语自动分词是所有中文信息处理的基础,同时也是语料库自动加工的基础。它的准确性也会直接影响到语料库加工的所有后续步骤,包括词性标注和句法分析层次。

5.4.3 词性标注

词性标注的任务就是根据一个词在某个特定句子中的上下文,为这个词标注正确的词性。其实质是研究词语所表现的语法功能的聚合关系,它要解决的主要问题是词性歧义(词性兼类)和未登录词词性的确定问题。词性标注是自然语言处理中的一个基本问题,因为在汉语中广泛存在着词语的词性多于一个的歧义现象。这种现象也叫做词性歧义或叫做词性兼类。例如“说”这个词在“哥白尼说”和“哥白尼的日心说”中分别为动词词

性和名词词性。

词性兼类问题在汉语中很突出,据不完全统计^[26,27],常见的汉语词性兼类现象有几十种,这些兼类现象具有以下分布特征:

- ① 在汉语词汇中,兼类词的数量不多,约占总词条数的 5%~11%。
- ② 兼类词的实际使用频率很高,约占总词次的 40%~45%。也就是说,越是常用的词,其词性兼类现象越严重。
- ③ 兼类现象分布不均。在孙茂松等^[17]的统计中,仅动名兼类就占全部兼类现象的 49.8%;在张民^[18]的统计中动名兼类和形副兼类就占全部 113 种兼类现象的 62.5%。

所以说,词性标注的正确率对整个语料库加工系统是比较重要的,同时它的正确率直接影响到文本的后续工作,因为词义消歧和句法分析都以经过词性标注的句子为基础。

目前词性标注方法主要有以下 3 种。

(1) 基于规则的方法

基于规则的方法利用语言学家手工建立的内省的规则来进行词性消歧,例如 TAGGIT 系统^[28]。TAGGIT 采用了 3300 个上下文框架规则(context frame rules)和 86 种词性标记,并用于标注布朗语料库,其标注准确率为 77%。Brill^[29]1992 年提出一个基于转换规则的词性标注方法,获得 95%~96% 的正确率,第一次使规则方法性能可以与统计方法媲美。Voutilainen(1994)^[30]则采用约束语法(constrain grammars)方法进行词性标注,获得 99.3% 的准确率,这一结果是迄今报道的关于词性标注的最好的结果。尽管语言学方法能取得很好的消歧结果,但考虑到用语言学方法获得一个好的模型所要付出的大量人力,把语言学方法看作一种非常耗时的方法。

(2) 基于统计的方法

基于统计的方法是最常使用的一种词性标注方法。对于给定的输入词串,基于统计的方法先确定其所有可能的词性串,然后对它们分别打分,并选择得分最高的词性串作为最佳的输出。常见的统计词性标注方法有基于频度的方法^[31]、基于 N 元模型的方法^[32]和基于隐马尔可夫模型(HMM)的方法。^[33]其中,HMM 结合 Viterbi 算法的词性标注方法最常见。

为了减少训练所需的手工标注语料的规模,Cutting 等^[34]在其开发的 Xerox 系统中采用了 Baum-Welch 算法来无指导地估计词性标注的 HMM 参数。传统的 HMM 方法只能处理固定长度的上下文信息。为提高消歧性能,Jung 等^[35]将最大熵模型应用于词性标注。该方法的显著特点是它可以融合不同阶的 N-gram 信息、长距离 N-gram 和其他有关词法的统计信息。

(3) 混合方法

混合方法是一种统计与规则相结合的方法,最典型的是 Lancaster 大学的 CLAWS (constituent-likelihood automatic word-tagging system)^[36]系统。CLAWS 系统将一些有关英语多字词、成语的规则引入 HMM,第一个版本完成于 1980—1983 年间,主要面向 LOB 语料库,包含 135 个标记。该系统从 CLAWS4 开始用于亿词级的英国国家语料库(British national corpus, BNC)的加工。最新版本是 1996 年的 CLAWS17。CLAWS 的准确率为 96%~97%。

在上述 3 种方法中, 基于统计的词性标注方法应用得最为普遍, 是目前词性标注的主流方法。词性标注是目前自然语言处理中比较成功的领域之一, 尤其是英语的词性标注, 其准确率一般可达 95%~97%, 已基本走向实用化。

汉语有关词性标注的研究起步较晚, 但近来发展很快。同英语相比, 汉语不仅缺少像英语那样对词性标注, 特别是对确定未登录词词性十分有用的丰富的形态信息, 如大写、词尾变化、词缀等; 而且汉语的词序相对英语自由, 汉语的词性与语法成分之间不像印欧语那样存在简单的一一对应关系。因而汉语词性标注的兼类问题和未登录词词性的识别问题更为突出。在同等条件下对英语词性标注比较有效的模型用在汉语中效果会有所下降。例如 HMM 方法在词性标注, 特别是词序相对严格的英语词性标注中获得极大成功, 是目前词性标注中广泛采用的一种方法。但该方法是建立在兼类词性的发射概率和转移概率独立假设基础上, 其效能依赖于目标词性的概率分布, 某个兼类词的候选词性(如汉语中的名词动词兼类词)的概率分布接近时, 这种方法常常产生错误的标注结果。

从概率角度上看, 词性标注问题可描述为: 给定一个含有词语序列 $w_1, w_2 \dots, w_m$ 的输入句子 W , 确定最有可能的词性标记序列 $T = t_1, t_2, \dots, t_m$, 使得条件概率 $P(T|W)$ 最大。 $P(T|W)$ 目前还很难直接估计, 一般采用贝叶斯原理进行转换, 即

$$P(T|W) = \frac{P(T)P(W|T)}{P(W)} \quad (5-3)$$

由于词性标注中, W 是给定的, $P(W)$ 不依赖于 T , 因此在计算 $P(T|W)$ 时, 可以不考虑 $P(W)$, 同时应用联合概率公式 $P(A, B) = P(A)P(B|A)$, 有

$$P(T|W) = P(T)P(W|T) = P(W, T) \quad (5-4)$$

进一步对式(5-4)应用概率乘法公式, 有

$$\begin{aligned} P(T|W) &= P(w_1, \dots, w_m | t_1, \dots, t_m) = \prod_{i=1, \dots, m} P(w_i, t_i | w_{1, \dots, i-1}, t_{1, \dots, i-1}) \\ &= \prod_{i=1, \dots, m} P(w_i | w_{1, \dots, i-1}, t_{1, \dots, i}) \cdot P(t_i | w_{1, \dots, i-1}, t_{1, \dots, i-1}) \end{aligned} \quad (5-5)$$

式中, $w_{1, \dots, i} = w_1, w_2, \dots, w_i$, $t_{1, \dots, i} = t_1, t_2, \dots, t_i$, $1 \leq i \leq m$ 。

公式(5-5)给出了不作任何简化假设的理想化的词性标注的概率模型。词性标注的任务就是从输入句子 W 的所有可能的词性标记序列 $\{T\}$ 中寻找一个最佳的标记序列 \hat{T} , 使得公式(5-5)的值最大为 W , 即

$$\begin{aligned} \hat{T} &= \arg \max_T P(T|W) \\ &= \arg \max_T \prod_{i=1, \dots, m} P(w_i | w_{1, \dots, i-1}, t_{1, \dots, i}) \times P(t_i | w_{1, \dots, i-1}, t_{1, \dots, i-1}) \end{aligned} \quad (5-6)$$

5.4.4 词性标注的 HMM 模型

模型(5-5)虽然反映理想状态下有关词性标注的概率分布, 但由于需要估计的参数空间太大, 该模型实际上不可计算的。因此, 在实际的词性标注系统中一般需要对该模型进行相应的简化, 以减少模型的参数空间。常用的简化方法是隐马尔可夫模型, 即在公式(5-5)所示的理想模型中引入一些独立性假设(即通常所说的马尔可夫假设), 这些假设包括:

(1) 词性标记 t_i 的出现只依赖于有限的前 $N-1$ 个词性标记, 即 N -POS 模型

$$P(t_1 | w_{1,\dots,i-1}, t_{1,\dots,i-1}) \approx P(t_i | t_{1,\dots,i-1}) \approx P(t_i | t_{i-N+1}, t_{i-N+2}, \dots, t_{i-1})$$

(2) 一个词语的出现不依赖于其前面的任何词语, 只依赖于前面的词性标记, 并进一步假设词语 w_i 的出现只依赖于词性标记 t_i , 即

$$P(w_i | w_{1,\dots,i-1}, t_{1,\dots,i}) \approx P(w_i | t_{i,\dots,i}) \approx P(w_i | t_i)$$

经过上述假设, 由式(5-5)可以得到一个 $N-1$ 阶词性标注的隐马尔可夫模型:

$$P(T | W) \approx \prod_{i=1,\dots,m} P(t_i | t_{i-N+1}, t_{i-N+2}, \dots, t_{i-1}) P(w_i | t_i) \quad (5-7)$$

式中, $P(w_i | t_i)$ 称为发射概率(emission probability)或词汇概率(lexical probability); 参数 $P(t_i | t_{i-N+1}, t_{i-N+2}, \dots, t_{i-1})$ 称为状态转移概率(transition probability)或上下文概率(contextual probability)。

5.4.5 Viterbi 词性标注算法

在 HMM 框架的词性标注中, 标注算法的主要任务就是为给定的观察值序列(即输入词串 W), 找到一个最佳的状态序列(词性标记串 \hat{T})。可以采用 Viterbi 算法实现这一过程。

给定词语序列 w_1, w_2, \dots, w_m , 设 $N_j (1 \leq j \leq m)$ 为词语 w_j 可能的词性标记数, 数组 $\text{SeqScore}(i, j)$ 为局部最优函数, 它记录从首词到第 j 词的第 i 个候选词性 t_j^i 的最佳词性标记序列的概率分值, 数组 $\text{BackPtr}(i, j)$ 则指向第 $j-1$ 词的在局部最佳标记序列的词性, 则基于 bi-gram 模型的 Viterbi 词性标注算法可具体描述如下。

输入 词语序列 $W = w_1, w_2, \dots, w_m$ 。

输出 最佳的词性标记序列 $\hat{T} = \hat{t}_1, \hat{t}_2, \dots, \hat{t}_m$ 。

算法

① 初始化:

for $i = 1$ to N_1

$\text{SeqScore}(i, 1) = P(w_1 | t_1^i) \cdot P(t_1^i | \phi)$

$\text{BackPtr}(i, 1) = 0$

② 递归过程:

for $j = 2$ to m

for $i = 1$ to N_j

$\text{SeqScore}(i, j) = \max_{1 \leq k \leq N_{j-1}} \{\text{SeqScore}(k, j-1) \cdot P(w_j | t_j^i) \cdot P(t_j^i | t_{j-1}^k)\}$

$\text{BackPtr}(i, j) = \hat{k} (\hat{k} \text{ 为上式取最大值时的 } k)$

③ 最佳状态序列求解:

$\hat{t}_m = \max_{t_m^i} \text{SeqScore}(i, m)$

for $i = n-1$ to 1

$\hat{t}_i = \text{BackPtr}(\hat{t}_{i+1}, i+1)$

5.4.6 语法分析

从理论上讲,语法分析要同时研究句子中蕴含的组合关系和聚合关系。根据索绪尔(Saussure)的语言理论,自然语言中存在着两种重要的关系,即组合关系(syntagmatic relations)和聚合关系(paradigmatic relations)。组合关系指语言单位组合的方式,而聚合关系表示语言单位间的相似关系或相似度。一个完整的语法分析涵盖词法和语法结构两个层次的自然语言分析,它既要研究词在形态上(词形)的组合关系以及各词在语法功能和意义上的聚合关系(即词法分析),又要研究各句子成分在结构上的组合关系和在语法功能上的聚合关系(语法标注)。因此,语法分析是一个自益问题(bootstrapping problem)。下面对目前常用的两种语法分析进行讨论,它们分别是上下文无关语法分析和依存语法分析。在自然语言理解中,上下文无关语法由于具有实用的分析算法而得到广泛应用。具体定义在上面已经讨论过了。下面给出具体的实例。假设有以下语法:

- | | | |
|---------------------------|---------------------------|--------------------------------|
| ① $S \rightarrow NP\ VP$ | ② $NP \rightarrow ART\ N$ | ③ $NP \rightarrow ART\ ADJ\ N$ |
| ④ $NP \rightarrow ADJ\ N$ | ⑤ $VP \rightarrow V$ | ⑥ $VP \rightarrow V\ NP$ |

常用的语法分析算法从方向上可以分为从上到下的语法分析和从下到上的语法分析两种。在从上到下的分析算法(图 5-6)中,算法开始于 S 符号,然后根据语法规则的左部符号来扩展右部符号,直到整个句子分析完成。在从下到上的分析算法中,算法开始于句子中的词,然后利用语法规则从语法规则中的右部符号来规约左部符号,直到规约到它只包含符号 S。

从上到下的算法中包含一个可能状态的线性表(possibilities list)。线性表中的第一个元素是当前状态(current state),其他状态叫做备份状态(backup states),每一个状态都是一个 \langle 符号列表,词位置 \rangle 对。算法描述如下:

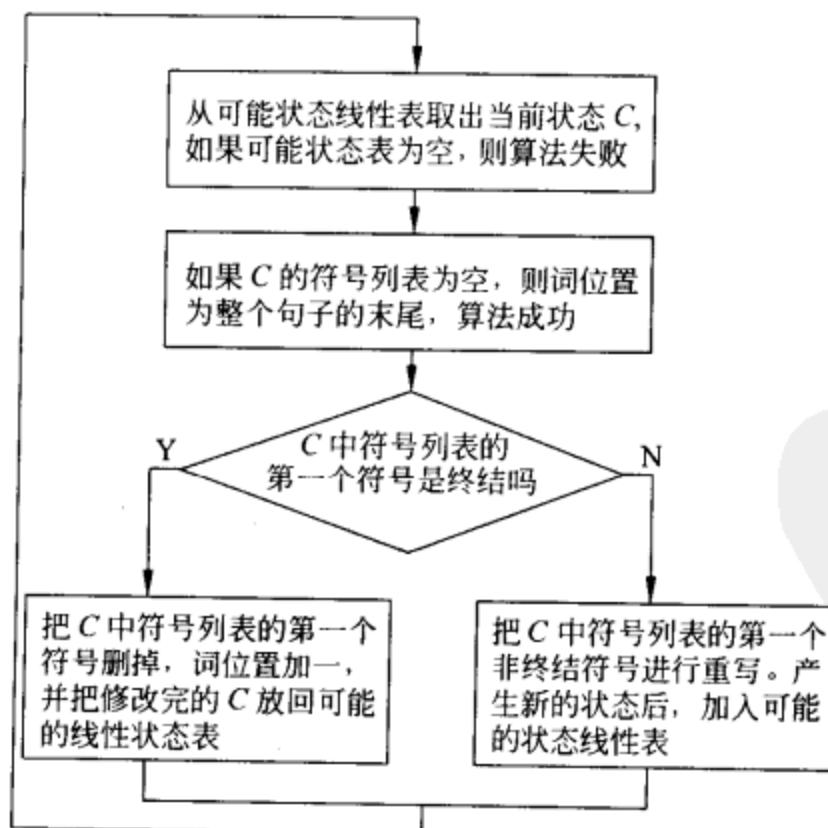


图 5-6 从上到下的句法分析方法

下面对句子“The old man cried”进行从上到下的语法分析(见表 5-5)。

the: ART old: ADJ, N man: N, V cried: V

其实,可以把语法分析看成一个人工智能中的搜索问题,对一个先深搜索,可能状态线性表是一个堆栈,按照后入先出原则;对一个先宽搜索,可能状态线性表是一个队列,按照先入先出原则。

表 5-5 从上到下句法分析的步骤

步骤	当前状态(current State)	备份状态(Backup states)
1	((S)1)	
2	((NP VP)1)	((ART ADJ N VP)1)
3	((ARTN VP)1)	
4	((N VP)2)	((ART ADJ N VP)1)
5	((VP)3)	((ART ADJ N VP)1)
6	((V)3)	((V NP)3) ((ART ADJ N VP)1)
7	((O)4)	((V NP)3) ((ART ADJ N VP)1)
8	((V NP)3)	((ART ADJ N VP)1)
9	((NP)4)	((ART ADJ N VP)1)
10	((ART N)4)	((ART ADJ N)4) ((ART ADJ N VP)1)
11	((ART ADJ N)4)	((ART ADJ N VP)1)
12	((ART ADJ N VP)1)	
13	((ADJ N VP)2)	
14	((N VP)3)	((V NP)4)
15	((VP)4)	
16	((V)4)	
17	((O)5)	

在做从下到上句法分析时,对句子从左到右进行扫描,将扫描得到的一系列词性范畴与语法规则右边进行匹配。如果想同时进行规约,则可规约成句法规则的左边。例如:对语法规则 $NP \rightarrow ART\ ADJ\ N$,可把 $ART\ ADJ\ N$ 规约成 NP 。整个分析过程就是一个不断扫描、归约、查找的过程。但是,这种简单的方法却付出很大的代价,因为需尝试很多次相同的匹配。通过一个叫线图的数据结构可以保存局部的分析结果,这样就避免了相同的结构被反复归约。线图是算法的关键数据结构,它保存了分析过程中到目前为止推导出来的成分,同时还保存了部分匹配的规则,我们把这部分匹配的规则叫做活动弧。另外把用来保存新的已归约成分的线性表称为代理表。自底向上的线图分析算法如下。

重复下面的步骤直到输入为空:

- ① 如果代理表为空,则取输入中的下一个词的词性,将其加入代理表中;
- ② 从代理表中选择一个成分;

③ 对于一个形如 $X \rightarrow \cdot CX_1 \cdots X_N$ 的规则, 增加一个从位置 $P_1 \sim P_2$ 的形式为 $X \rightarrow C \cdot X_1 \cdots X_N$ 的活动弧;

④ 使用弧扩展算法将 C 加入到线图中。

弧扩展算法表述如下: (在 Chart 表中从位置 $P_1 \sim P_2$ 增加成分 C)

① 把成分 C 从代理表中移出, 插入到线图的 P_1 到 P_2 的位置;

② 对于位置 $P_0 \sim P_1$ 的形式为 $X \rightarrow X_1 \cdots \cdot C \cdots X_N$ 的活动弧, 从位置 $P_0 \sim P_2$ 增加一个新的活动弧 $X \rightarrow X_1 \cdots C \cdots X_N$;

③ 对于位置 $P_0 \sim P_1$ 的形式为 $X \rightarrow X_1 \cdots X_N \cdot C$ 的活动弧, 从位置 $P_0 \sim P_2$ 将 X 作为新成分加入代理表中。

在算法分析过程中, 像上面介绍的从上到下算法介绍的那样, 也可以采用深度优先或者是宽度优先的搜索方法, 采用哪种方法取决于代理表是堆栈实现还是队列实现。下面给出一个实例^[37]。

我们分析的句子是 The large can hold the water, 其中各词的词性是

the: ART large: ADJ can: N/AUX/V hold: V water: N/V

语法如下所示:

(1) 读入 the 成分, ART1 被放到代理表中;

从代理表中取出 ART1(the 从 1 到 2);

在线图中加入 ART1;

在线图中增加活动弧, NP → ART · ADJ N 从 1 到 2;

在线图中增加活动弧, NP → ART · N 从 1 到 2。

(2) 读入下一个词 large, 成分是 ADJ2 被放到代理表中;

从代理表中取出 ADJ2(large 从 2 到 3);

在线图中加入 ART1;

在线图中增加活动弧, NP → ADJ · N 从 2 到 3;

在线图中增加活动弧, NP → ART ADJ · N 从 2 到 3;

第二个活动弧是对上一个活动弧的延伸, 若利用了弧扩展算法。

(3) 读入下一个词 can, 成分是 N1, AUX1 V1 被放到代理表中;

从代理表中取出 N1(can 从 3 到 4);

在弧扩展算法中产生了两个 NP, 第一个 NP 从 1 到 4, 是由规则 2 构造的, 第二个 NP 从 2 到 4, 是由规则 4 构造的; 现在的 NP 在代理表的顶端。

(4) 从代理表中取出 NP1, (the large can 从 1 到 4);

增加活动弧, S → NP · VP 从 1 到 4;

代理表中取出 NP2(large can 从 2 到 4);

增加活动弧, S → NP · VP 从 2 到 4;

到此为止, 线图的结构如图 5-7 所示。

在图中并没有显示活动弧, 而所有的活动弧如图 5-7 所示都是存在的。如果生成的 S 成分能够覆盖整个句子, 则说明分析成功(如图 5-8 中所示的 S1)。

如果不使用线图保存局部的分析结果, 则语法分析的时间复杂度将是 C^n , 而如果使用线

图,则时间复杂度将会是 $k \cdot n^3$ 。其中 n 是被分析句子的长度。从这里可以看出,基于线图的语法分析算法是很高效的。

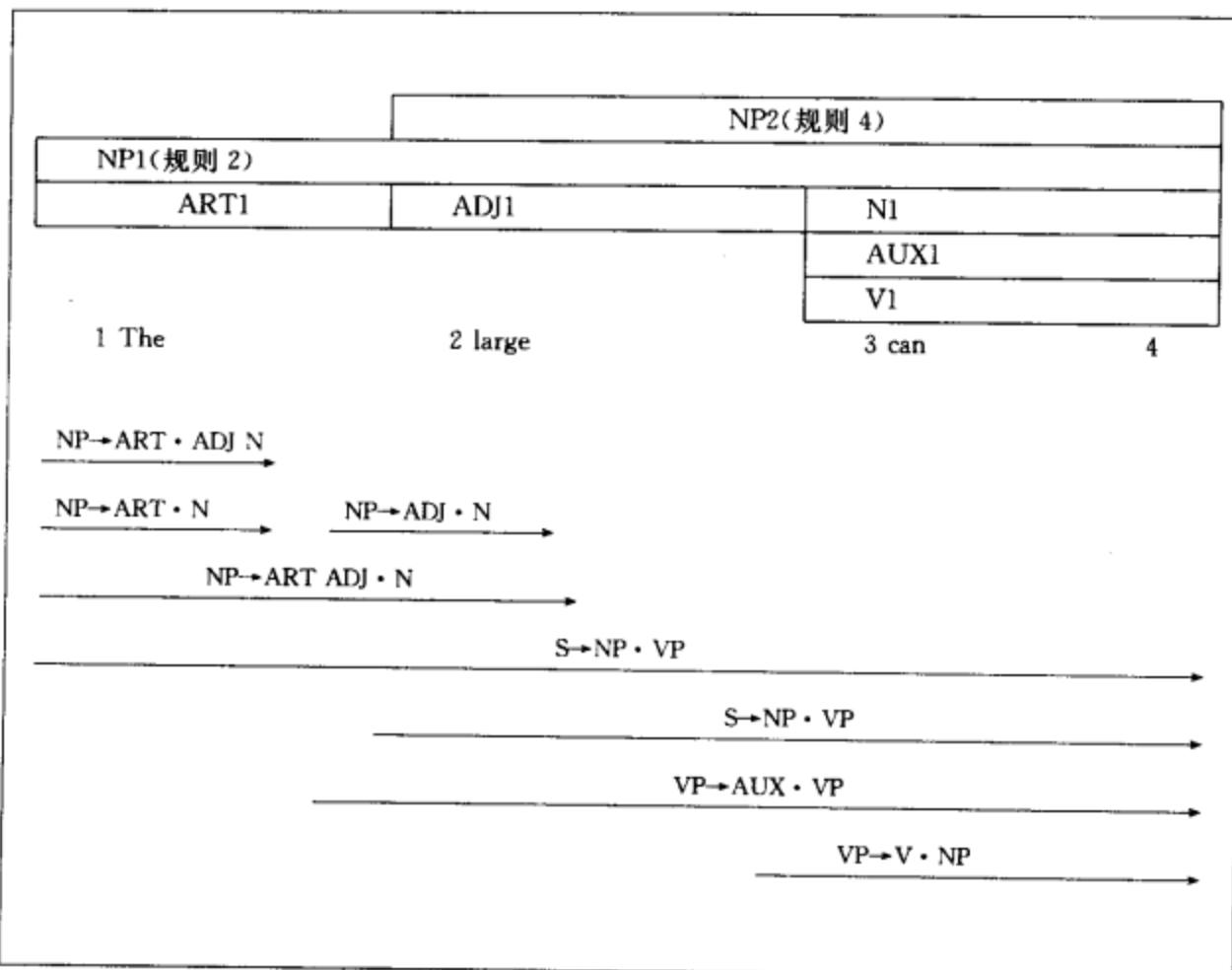


图 5-7 图分析中的局部结果线图

S1(规则 1 由 NP1 和 VP2 组成)							
S2(规则 1 由 NP2 和 VP 组成)							
		VP3(规则 5 由 AUX1 和 VP2 组成)					
NP2(规则 4)		VP2(规则 5)					
NP1(规则 2)				VP1(规则 6)			
		N1	N2		NP3(规则 3)		
		V1	V2	V3		V4	
ART1	ADJ1	AUX1	AUX2	N3	ART2	N4	

图 5-8 最后的结果线图

5.4.7 概率上下文无关语法

通过把概率引入语法分析,可以在以下 3 个方面使用概率:第一是可以利用概率加速语法分析,第二是可以用概率在语法分析的不同结果中选择正确结果,第三是通过概率进行句子确定^[38]。通过把概率值赋给每一个语法重写规则来进行基于概率的语法分析。语法规则的概率值来自于对已经进行了语法分析后的树库。树库中语法规则的统计如表 5-6 所示。

表 5-6 树库中语法规则的统计

规则	规则左部计数	规则计数	概率
S→NP VP	200	200	1
VP→V NP PP	300	200	0.666
VP→V NP	300	100	0.333
NP→N N	400	100	0.25
NP→NP PP	400	300	0.75

一个概率上下文无关语法(probabilistic/stochastic context-Free grammar, PCFG/SCFG)是一个四元组

$$G = \langle \Sigma_N^+, \Sigma_T^+, S, R \rangle$$

其中, $\Sigma_N^+ = \{N^1, N^2, \dots, N^{|\Sigma_N^+|}\}$ 是一个非终结符集合; $\Sigma_T^+ = \{t^1, t^2, \dots, t^{|\Sigma_T^+|}\}$ 是一个终结符集合; S 是一个起始符号, $S \in \Sigma_N^+$; $R = \{r^1, r^2, \dots, r^{|R|}\}$ 是一组产生式, 对于每个产生式 $r^i \in R$ 有

$$(r^i : N^j \rightarrow \xi^k, P_r(r^i))$$

其中, $\xi^k \in (\Sigma_T^+ \cup \Sigma_N^+)^*$ 产生式概率 $P_r(r^i) = P_r(N^j \rightarrow \xi^k | N^j)$, 表示句子推导过程中非终结符 N^j 被字符串 ξ^k 所替换的概率, 它一般满足

$$P_r(N^j \rightarrow \xi^k | N^j) > 0, \text{ 且 } \forall N^j \in \Sigma_N^+, \sum_{\xi^k} P_r(N^j \rightarrow \xi^k | N^j) = 1.$$

由以上可知, PCFG 实际上是 CFG 和概率的组合, 因此基于 PCFG 的语法分析结果取决于 CFG 规则及其统计特性。设某个句子 S 有 λ 个可能的语法分析树, 以 $\pi(S) = \{\pi_1, \pi_2, \dots, \pi_\lambda\}$ 表示, 那么, 句子 S 的第 i ($1 \leq i \leq \lambda$) 棵分析树 π_i 的概率为该分析树所用到的所有规则的概率的乘积, 即

$$P_r(\pi_i) = \prod_j P_r(r_j) \quad (5-8)$$

而句子 S 的概率则为它所包含的所有可能的分析树的概率之和, 即

$$P_r(S) = \sum_{i=1}^{\lambda} P_r(\pi_i) \quad (5-9)$$

于是, 语法分析的目的就是从输入句子所有可能的分析树中寻找最佳的分析树 $\hat{\pi} = \arg \max_{1 \leq i \leq \lambda} P_r(\pi_i)$ 。实际上, 概率 $P_r(\pi_i)$ 和 $P_r(S)$ 是语法分析的两个重要方面。前者反映了句子 S 的第 i ($1 \leq i \leq \lambda$) 棵候选分析树 π_i 的可能性, $P_r(\pi_i)$ 越大, 说明 π_i 在语法或语义上越符合句子 S 的本意; 后者则反映句子 S 合乎语法的程度。可见, 与 CFG 相比, PCFG 不仅具有定量描述句法歧义的能力, 而且可以描述不符合语法的句子。

一个应用概率语法的明显例子就是, 可以对上面提到的线图算法的代理表进行改进, 上例中的搜索过程属于盲目搜索, 没有任何启发信息, 但是可以对代理表中的待分析成分依据语法规则的概率进行排序, 具体地可以用优先队列来实现代理表。这种语法分析方法叫做 Best First Parsing。

基于概率的语法分析可以加快语法分析的速度, 但是却不能大幅度提高语法分析的正确率。例如, 用基于概率的语法对本输入“VNPN”进行语法分析时, 可以得到两种结构

(如图 5-9 所示)。

第 1 棵句法分析树的概率是 $P(T_1) = 0.666$, T_1 代表如图 5-9(a)所示的树;

第 2 棵句法分析树的概率是 $P(T_2) = 0.333 \times 0.75 = 0.25$, T_2 , 表示如图 5-9(b)所示的树。

从以上两个结果可以看出,当看到 V, NP, PP 这种结构时,首先把 V 和 NP 合成为 VP, 然后再把 PP 与 VP 进行关联, 当处理 eat cake with spoon 时, 这样做可以, 但当处理 eat cake with milk 时, 这样做是完全错误的。造成这种错误的原因是基于概率的语法分析没有引入词汇信息。如果把词汇信息引入语法分析, 就可以成功地解决这些问题。目前这类语法分析已成为研究的热点。

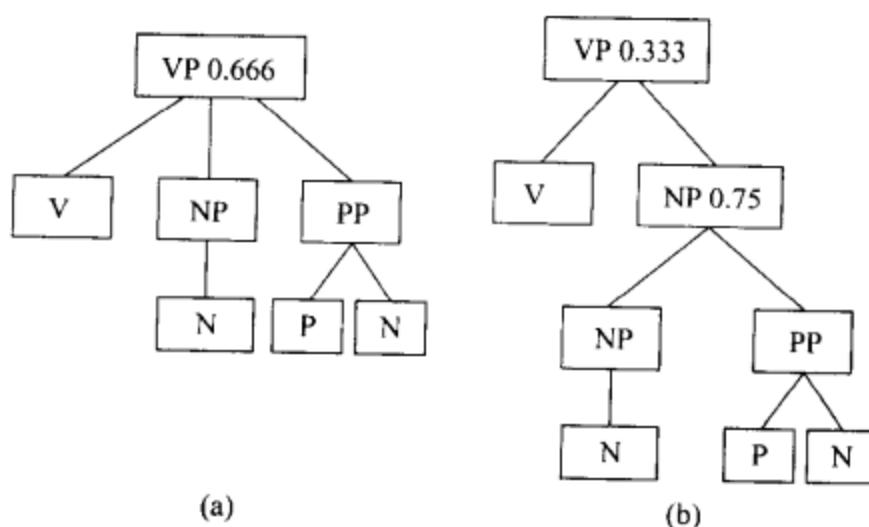


图 5-9 相同输入的不同语法分析结果

5.4.8 语料库的应用

语料库的应用基本上可以分为两部分:一部分是经过标注和预处理的语料库,该语料库可以为所有的基于统计的自然语言理解提供统计的数据资源;另外一部分就是语料库自动加工技术,如分词、词性标注、语法分析等自动加工技术是具体应用的基础。目前,基于统计的方法已经广泛用于机器翻译、信息提取、信息检索、文本分类、问答系统、自动文摘、词典编撰等各个方面。可以说,语料库是基于统计的自然语言处理技术的基础。

没有任何标注的语料库叫做原始语料库,也叫作生语料库。对于中文及这样的语料库,可以对汉语的字频进行相关的统计,也可以对字频进行二元同时出现概率的统计,这些统计数据对于汉字的拼音输入、汉字分词系统、文本纠错等都有很现实的应用意义。对于已经完成分词的语料,可以对其进行词一级的二元、三元同时出现概率的统计。这些统计概率反过来可以为汉语分词系统提供数据方面的支持,同时也可用在词典编撰、短语识别等方面的研究上。对于已经标注了词性信息的语料,可以统计出词性的二元同现概率以及同一词性下不同词的发射概率,而这两个概率又是基于隐马尔科夫模型的词性标注系统的两个主要资源。如果没有这两方面的数据,就没有办法进行词性标注。对于标注了层次短语结构的树库,可以统计出规则的应用次数,这样就可以把上下文无关语法变成概率上下文无关语法,而概率上下文无关语法明显优于上下文无关语法。

语料库自动加工技术主要包括分词、词性标注、短语识别、句法分析等层次的加工技

术。对不同的应用,有不同的加工深度要求,对于信息检索,分词是必不可少的;对于机器翻译,翻译又必须加工到语法分析层次。但无论是什么应用系统,都需要一定的语料库自动加工技术,这是非常明显的。

5.5 小结

本章主要讲述了常用的语法体系和基于短语结构语法的语料库多级加工技术。这两部分内容是彼此相关的。语法体系是语料库多级加工的理论基础,而加工好的语料库是最终的可用产品。汉语的自动语法分析,是汉语自然语言处理的核心难题,解决这个核心难题,依赖于适合汉语的语法体系,同时也依赖于成熟、丰富的汉语树库。只有有了完备、正确、实时、自动的带有语义分析的语法分析系统,我们才能真正地说:“计算机理解了语言”。

参考文献

- 1 俞士汶等. 现代汉语语法信息词典详解. 北京: 清华大学出版社, 1998
- 2 梅家驹等. 同义词词林. 上海: 上海辞书出版社, 1983
- 3 张民, 李生, 赵铁军. 统计与规则并举的汉语词性自动标注算法. 软件学报, 1998, 9(2): 134~138
- 4 白拴虎. 基于统计的汉语语料库词性自动标注的研究与实现. 清华大学硕士学位论文
- 5 赵铁军等. 机器翻译原理. 哈尔滨: 哈尔滨工业大学出版社, 2001
- 6 G. A. Miller, R. Beckwith, C. Tellbaum, D. Gross and K. Miller. Introduction to WordNet: an on-line lexical database. CSL43. Cognitive Science Laboratory, Princeton University, Princeton, NJ, 1993
- 7 张卫国. 广义短语结构语法述略. 语言文字应用, 1996(1)
- 8 苗传江, 张庆旭等. 功能合一语法. 语言文字应用, 1995(3)
- 9 Walt Detmar Meurers. On Implementing an HPSG theory, 1994
- 10 尤爱莉. 中心语驱动短语结构语法评价. 外语学刊, 2000(2)
- 11 冯志伟. 特思尼耶尔从属关系语法. 国外语言学, 1983(1)
- 12 袁毓林. 汉语动词的配价研究. 江西教育出版社, 1998
- 13 周明, 黄昌宁. 面向语料库标注的汉语依存体系的探讨. 中文信息学报, 1994, 8(3)
- 14 王玲玲. 格语法及其在汉语研究中的应用, “信息处理用语言理论讲话”第三讲. 语言文字应用, 1994(04)
- 15 庞素芬. 系统功能语法与格语法的比较. 语言文字应用, 1994(04)
- 16 苗兴伟. 系统功能语法与转换生成语法对比刍议. 外语研究, 1998(3)
- 17 胡壮麟, 朱永生等. 系统功能语法概论. 湖南教育出版社, 1989
- 18 沈家煊. 认知语法的概括性. 外语教学与研究, 2000(01)
- 19 孙茂松. 扩充转移网络语法. 语言文字应用, 1995(1)
- 20 翁富良, 王野翊. 计算语言学导论. 中国社会科学出版社, 1998
- 21 孙宏林, 俞士汶. 浅层句法分析方法概述
- 22 Magerman, D. M. Marcus. Parsing a natural language using mutual information statistics. In Proceedings of AAAI'90, 1990: 984~989

第5章 基于语言理解的处理方法

- 23 Chen, Hsin-His. Lee. Development of a partially bracketed corpus with part-of-speech information only. In Proceeding of the 3rd Workshop on Very Large Corpora, 1995:162~172
- 24 Brill Eric. Unsupervised learning of Disambiguation Rules for part speech tagging. In Proceedings of the 3rd Workshop on Very Large Corpora, 1995:1~13
- 25 刘小虎. 英汉机器翻译中词义消歧方法的研究. [博士学位论文]. 哈尔滨: 哈尔滨工业大学
- 26 刘开瑛, 王小鹏. 现代汉语句子分析中的歧义现象及处理策略. 见: 1990 年中国人工智能学术论文集(CJCAI90), 1990. 311~321
- 27 B. B. Green, G. M. Rubin. Automated Grammatical Tagging of English. Technical report, Brown University. 1971
- 28 Bernard Merialdo. Tagging English Text with a Probabilistic Model. Computational Linguistics, 1994, 20(2): 1~29
- 29 A. Voutilainen. Three Studies of Grammar-based Surface Parsing of Unrestricted English Text. Publication No. 24, University of Helsinki, 1994
- 30 I. Marshall. Choice of Grammatical Word-Class without Global Syntactic Analysis: Tagging Words in the LOB Corpus. Computers and Humanities. 1983, 17: 139~150
- 31 Kenneth Ward Church. A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. In Proceedings of 2nd Conference on Applied Natural Language Processing. Austin, Texas., 1988: 136~143
- 32 J. M. Kupiec. Probabilistic Models of Short and Long Distance Word Dependencies in Running Texts. In Proceedings of the (1989) DARPA Speech and Natural Language Workshop. Philadelphia, 1989: 290~295
- 33 D. Cutting, J. Kupiec, J. Pederson, P. Sibun. A Practical Part -of-speech Tagger. In Proceedings of 3rd Conference on Applied Natural Language Processing. Italy: Treno, 1992: 133~140
- 34 Sung-Young Jung, Young C. Park, Key-Sun Choi and Youngwhan Kim. Markov Random Field Based English Part-of-Speech Tagging System. In Proceedings of COLING '96. Copenhagen, Denmark, 1996: 236~242
- 35 G. Leech, R. Garside, M. Bryant. CLAWS4: The Tagging of the British National Corpus. In Proceedings of COLING'94. Kyoto, Japan. 1994
- 36 James Allen. Natural Language Understanding. The Benjamin/Cummings Publishing Company. Inc.
- 37 Christopher D. Manning. Hinrich Schutze . Foundations of Statistical Natural Language Processing. The MIT Press



音字转换技术

音字转换是汉字智能拼音键盘输入和汉语连续语音识别中的关键问题。本章将介绍音字转换问题的应用背景，并讨论拼音输入的各种表达形式、拼音音节切分、拼音(语音)纠错以及音字转换方法等主要问题。

6.1 引言

根据输入设备的不同，文字输入方法分为语音输入、手写输入和键盘输入3类。其中键盘输入方法目前得到最广泛的应用，另外两种方法为计算机用户提供了更人性化的输入方式，在某些特殊场合也具备特别的用途，但由于技术上的限制，其应用受到很多限制。对汉语的计算机处理而言，汉字的键盘输入也是个重要的问题。

在诸多汉语的输入方法中，语音识别和拼音键盘输入是两个典型的方法，它们面临一个共同的问题：音字转换。这也是本章的讨论内容。

汉语的发音以音节为单位，音节是语音识别的基本单位，也是拼音键盘输入的基本单位。每个音节通常对应多个汉字，在以单字或词为单位进行语音输入或拼音键盘输入时，给定输入音节，常对应多个同音字或同音词，需要用户进一步选择确定，因而输入速度非常慢。如果以较长的语段或语句为单位来输入，则字与字之间、词与词之间存在的语言上的约束关系，可以协助确定合适的同音字或词，有效加快输入速度，对拼音键盘输入而言，可有效缩短汉字输入的平均码长。

音字转换技术就是用计算机这个工具，采用人工智能、统计学、语言学等方法，充分利用各种语言单位的上下文关系，处理汉语语音音节串或拼音串到相应汉字串的自动转换。这里语音音节与汉语拼音实质上是相同的，但在不同场合表示形式不同，处理方法有所侧重和不同。

本章首先详细介绍汉语的声音语句输入和智能拼音键盘输入这两个背景问题；然后主要以智能拼音键盘输入为背景讨论音字转换系统面临的几个主要问题：拼音的多种表达形式，拼音音节的自动切分、拼音(语音)纠错、音字转换方

法等。

目前国内外在面向语音识别等大规模真实文本处理的语言学研究中,已形成了以统计方法为主要内容的语言模型研究分支,其内容在前面章节中有详细论述,其方法基本适用于本章的音字转换。

6.2 声语音句输入

6.2.1 声语音句输入的提出

声语音句输入是人机通信最自然最理想的方式,在国内外均被作为高技术重要课题加以研究^[1-4]。从声音到文字的输入过程可分为两个阶段:一是语音识别阶段,把自然的声音信号转换为机器可以处理的数字表达的音节形式(或拼音形式);二是语音理解阶段或音字转换阶段,把音节转换为汉字形式。

汉语的语音识别单位分为单音节(字识别)和多音节(词识别)两种。对于通用的汉语文字系统来说,引人注目的是单音节语音识别技术,这是因为

① 汉语是单音节语言。在口语中,词与词之间很少有清晰的停顿,主要是单音节之间有间隔,单音节输入较易为人们接受。

② 汉语中音节数只有 400 个,考虑音调也只有 1 200 个,词是由音节组成。只要正确输入上述音节,这些音节就能构成可以说是无限的词组和语句。而当做语音输入训练时,只训练单音节是比较容易的,而在大词汇量(如几万个)情况下,训练词的输入是难以让用户接受的。

③ 单音节输入比多音节输入变化范围小,对系统资源的要求较低。

在语音理解阶段语音识别处理有 3 种形式:字处理、词处理和语句处理。

(1) 字处理

字处理的典型形式是,把语音识别器给出的和输入音节相近的几个音所包含的近音字在计算机屏幕上显示出来,让用户选择所需的汉字。这种形式的特点是实现简单,对系统资源要求不高。但由于语音理解的工作基本上留给用户,作为屏幕提示的候选汉字太多,因此查找困难,用户输入速度慢,眼睛容易疲劳,不适于较长文本的输入。

(2) 词处理

词处理的典型形式是,根据用户读入音节停顿时间的不同来确定词的长短,再将词与系统词库进行近音匹配,把近音词在屏幕上显示出来让用户选择。其特点是,系统排除了不太可能的候选词,留给用户选择的范围较小,选择工作比较容易、迅速。但朗读时字词停顿的时间不能相同或相近,读词的时间停顿较长,节奏不好掌握,读音的速度相对较慢,用户读音时还要思考输入文本的语义,确认是字还是词,脑力劳动强度较大。

(3) 语句处理

语句处理明显优于字、词处理形式,这里因为

① 从操作心理学上看,操作人员倾向于按有一定意义的短语或句子为单位进行短时记忆。

② 从信息论角度讲,汉字的多维熵要少于一维熵,语句输入法比字、词输入法需要较少的输入信息。这样对语音识别器精度的要求较低,这一点非常重要,因为语音识别正确率达到较高程度以后,再提高一点很困难。

因此单音节语句输入操作比较自然,对用户的限制较少,其与语音后处理技术相结合可以实现理想的语音识别系统。

6.2.2 声声音语句的推理

图 6-1 给出了声音语句输入的示意图。

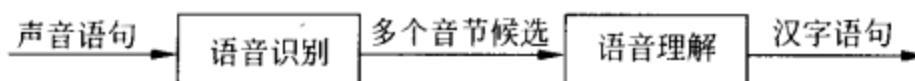


图 6-1 声声音语句输入分阶段处理示意图

语音识别部分采用模块匹配法或隐马尔科夫模型等方法把听觉信号转换为数字表达的音节形式。由于不同的人发音不同,即使同一个人在不同时刻、不同环境的发音也不同,再加上噪音等多种因素的影响,所以语句部分的输出通常不是惟一确定的。如果把作为候选的 M 个相似音节作为行,相应输入语句中的 N 个音节作为列,则语音识别部分的输出可表达为一个音节候选二维向量。图 6-2 为“今天是星期五”语音识别部分给出的音节候选向量的示例,其中,音节用汉语拼音表示,候选音节和输入音节的距离用数字表示,沿 Y 轴顺序排列。数字越小,相似度越好。声音语句的音节按输入的顺序沿 X 轴排列,音节数 $N=6$,候选音节数 $M=9$ 。

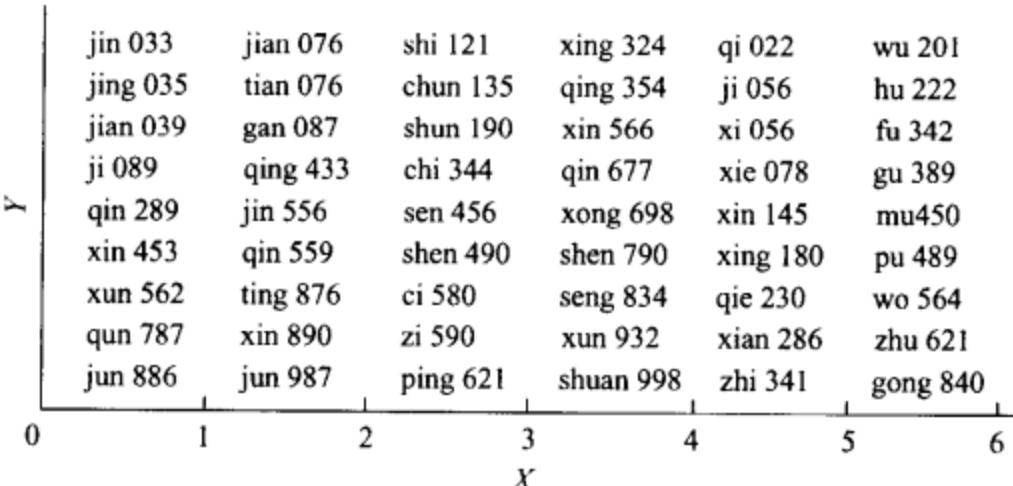


图 6-2 对“今天是星期五”语音识别给出的音节候选向量

语音理解的任务不仅是从音节候选向量中选择出正确的音节,更主要的是从音节中选出正确的汉字。如果把同音字作为一维向量(沿 Z 轴方向),则二维的音节候选向量可以转化为三维的音字候选向量。图 6-3 将图 6-2 的二维向量转化为三维向量。为清晰起见,只列出了前 3 个候选音节的情况。

人类在理解自然语言时,除了听觉信号外,还必须有其他知识(如音节、词、语法、语义、语境等知识),才能理解所听到的声音。对这些知识的组织、表达、获取以及使用,都是语音理解部分研究的内容。该部分模拟人们听写行为的思维过程,运用多种知识排除歧

义性，在音字候选向量中找到一条沿 X 轴方向从 1~N 的最佳路径作为输出。

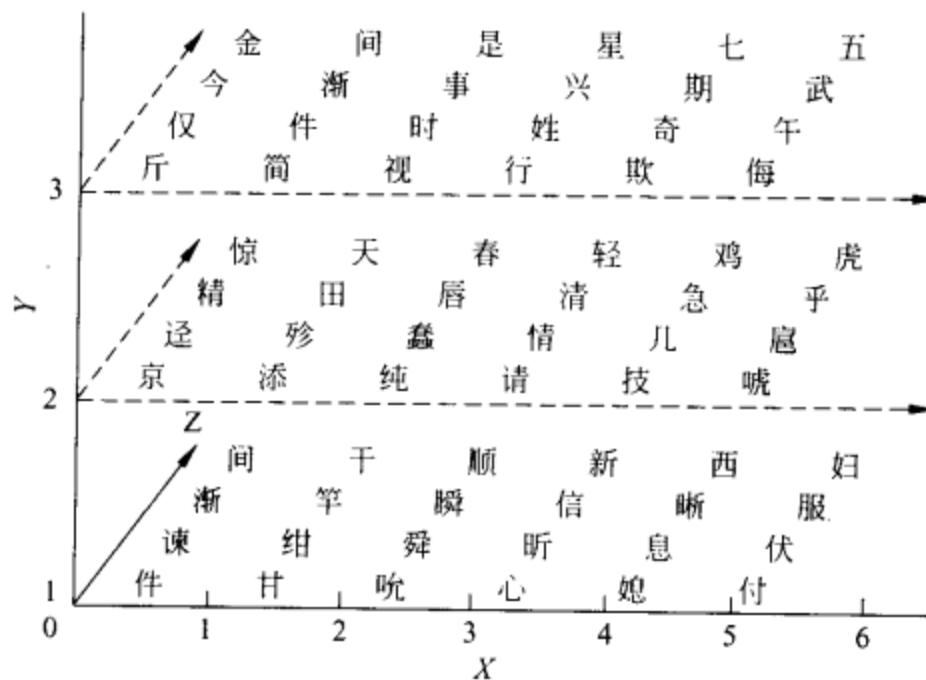


图 6-3 音字候选三维向量举例

6.2.3 声语音句输入的系统实现

声语音句输入系统的框图如图 6-4 所示，以下对几个主要模块给予说明。

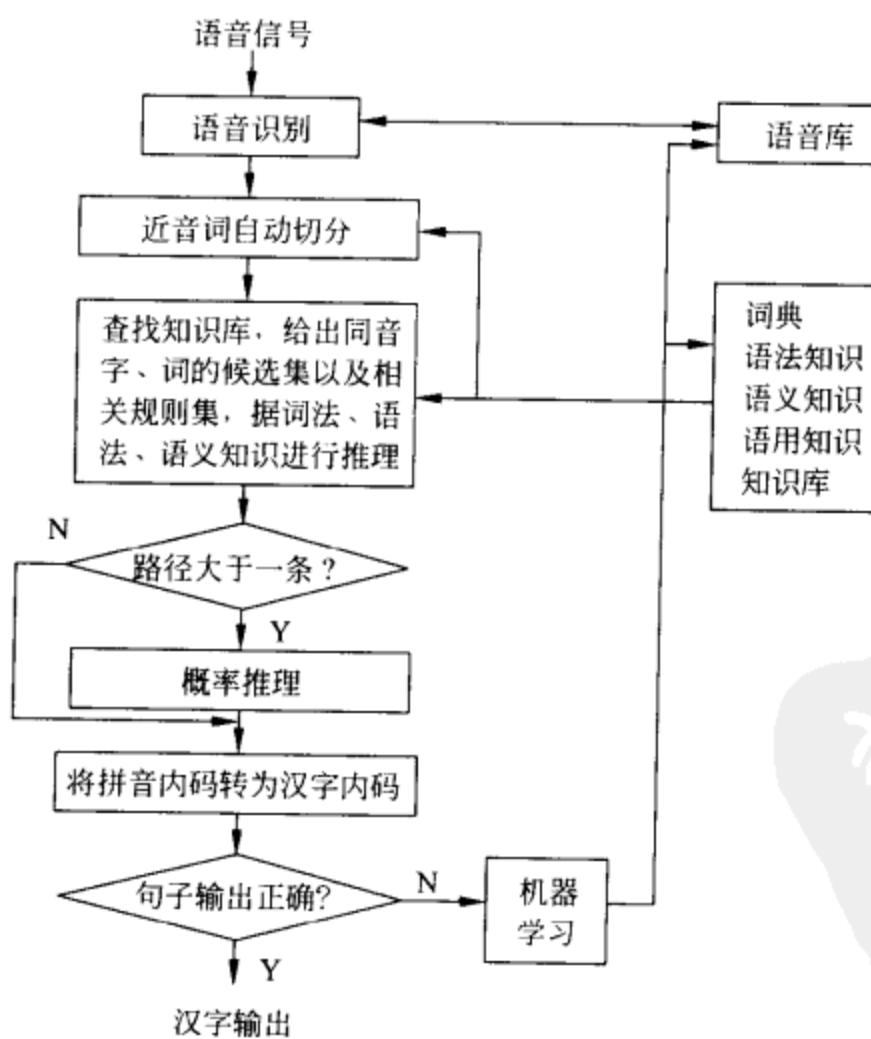


图 6-4 声语音句输入流程图

(1) 语音识别模块

该模块把声音输入转化为计算机可以处理的音节候选二维向量。

(2) 音词自动切分模块

由于书面语言中词与词之间没有空格,而人们说话时也极少在每个词之间给予停顿,所以以词为输入单位的转换方式很不自然。系统自动分词是必不可少的。

(3) 语义和语法推理模块

根据分词后的近音语句查找知识库,根据语义和语法规则进行自底向上的归约推理。

(4) 概率推理模块

如果在上个模块得到两个以上的候选结果,而又无法用其他知识或在缺少某些知识的情况下做进一步处理,则采用概率推理的方法进行歧义判断,从而选出一个最有希望的可能结果作为输出。

(5) 输出验证和机器学习模块

从理论上讲,语句级声音输入方法无法或很难达到 100% 正确。在将声音语句转换为汉字语句后,系统一般把输出结果显示在屏幕上,等待用户验证。

如果转换结果完全正确,则系统对转换机制及所用的知识加以肯定,即对转换该语句所使用的字、词、规则等的可能性或优先度进行小幅度增值处理,也可对其他知识进行小幅度减值处理,这称为自然记忆。

如果转换结果有误,则系统为用户提供一个实时修正转换错误的方法,即句内编辑。系统将对用户改正后的汉字语句进行词法、语法、语义等各方面的分析,且进行词和规则的自动生成、对相关字、词的可能性及优先度大幅度增(减)值处理等操作,对知识库进行较大的修改和更新,以便保证今后对该类语句声音输入的正确性,这称为强化记忆。

6.3 汉字智能拼音键盘输入

汉字键盘输入是汉字输入的主要方式。汉字键盘输入方法可以分为两类,一类是基于字形的输入法,一类是基于拼音的输入法。基于字形的输入法(形码)重码率低,平均码长较短,因而输入速度快,但难以掌握,只适用于专职打字员;拼音是汉字的标准音调符号,因其易学易用,成为最受欢迎的输入手段,97%以上的中文用户使用这种方法^[5],缺点是由于汉字多音字现象很普遍,因而重码率高,基于字、词的拼音输入往往速度较慢。根据上下文建立汉语语音模型,并在此基础上以句子为单位进行输入能很好地克服普通汉字输入法的弊端。目前,基于拼音的汉字智能语句输入法因其易学、快速、具有良好的人机接口等特点已成为广大计算机用户认同的主流输入法。

汉字智能拼音键盘输入的核心内容是拼音串到汉字串的自动转换,这个过程与汉语声音语句输入中的语音后处理部分本质上是一样的,简称音字转换。音字转换是将连续的拼音流自动转换成相应的文字流的过程。给定一个连续的拼音流,存在各种不同的切分方式,由此得到拼音的各种不同组合。每个特定拼音的切分对应一系列同音字或同音词,相邻的字或词间存在一个由前向后的连接关系,由此构成音字转换问题的状态空间(如图 6-5 所示)。音字转换即为在词网格中搜寻最优路径的过程。

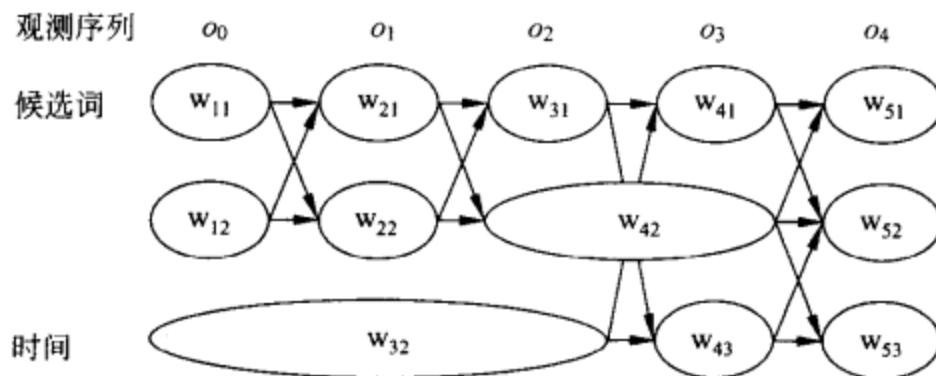


图 6-5 词网格示意图

6.4 拼音输入的多种表达形式

在汉语拼音中,音节是语音中最小的结构单位,也是人们可以自然地察觉到的最小语音单位。普通话的音节一般由声母、韵母、声调 3 部分构成。一般来说,一个汉字的读音就是一个音节,它由 1~4 个音素组成(音素是从音质角度划分的最小语音单位,包括元音和辅音)。

汉语拼音是一种国家标准,是中国语文教育的必修内容,因此掌握汉语拼音的人数众多,这也是拼音输入法易于掌握的原因。目前汉语拼音输入法的基本形式是采用汉语拼音原型,称之为“全拼”。全拼输入方式由于其易学性被广泛接受,但也存在一些弱点,比如这种输入方式表示一个音节的字母数量较多,使得汉字输入的平均码长较长;再比如这种拼音编码的码长不一致,在输入拼音的时候也为用户带来一些麻烦,拼音之间必须显式输入间隔,进一步增加了汉字输入的平均码长。这个问题可以通过拼音音节自动切分方法来解决,但这毕竟增加了系统的负担,另外,对全拼方式很多计算机用户并不完全熟悉,怎样更好地适应这类用户是一个重要问题。针对全拼输入方式的各种缺点,可以考虑对基本拼音的表达形式作适当变换,以适应不同用户的需要。下面将介绍除全拼输入方式之外的其他拼音输入方式。

6.4.1 拼音助学和提示输入

对于拼音不熟的用户,或者偶尔忘记某个拼音的拼法,或者想了解某个拼音有哪些同音字时,都可以使用提示方式了解。

1. 声母提示

声母有 23 个,另外可定义 5 个准声母包括 y, w 以及韵母音节的首字母 a, o, e, 也就是说,若音节中不含声母,则该音节第一个字母定义为声母。汉语中有 400 多个音节,1200 多个音调节,6000 多个汉字(GB2312—80),它们都可按声母分为 26 组。

用户输入声母,再输入提示键,即可熟悉以此声母为首的汉字各音节、双拼(简拼)字符或常用字。

2. 拼音提示

当输入一个完整的拼音音节后,再按提示键,可以了解该拼音所有的同音字以及相应的形码编码,可以再输入音调或形码键,以便形成音节、音调节或再根据形码确认汉字。

6.4.2 简拼快速输入

1. 拼音的压缩表达(简拼、双拼、三拼)

为减少拼音字母串的长度,可以为多字母声母和韵母定义简码,这样,每个拼音音节仅用至多3个简码就可表示,简化后的拼音形式称为简码。双拼和三拼是特殊的简拼形式,双拼即1个拼音由2个简码组成(声母+韵母),三拼即1个拼音由3个简码组成(声母+介母+韵母)。

各种简拼方式可以单独使用,也可与全拼形式混合使用,为用户提供更大的灵活性。输入法系统通常提供各种简拼形式供用户选用。

2. 快速语句输入

对简拼形式熟悉的用户可以采用快速语句输入方式。该方式的要点是用双拼表示拼音(每个音节只需用两个字母表示),目的是省略拼音间隔符号,这样绝大多数汉字可以用两键输入。这种方式和其他简拼方式都可选用音调。

6.4.3 用户自定义简拼

由于简拼形式没有国家标准,不同的汉字系统采用的简拼形式多种多样,一些用户对某种拼音缩写表达形式已成习惯,比较难适应其他方法,因而输入法系统提供用户自定义简拼功能,以利于用户自定义某种简拼形式。

6.4.4 模糊拼音输入

对于发音不准的用户,例如平卷舌音节c和ch,s和sh,z和zh,b和p,in和ing等分不清的用户,一下子改正发音很困难,可以使用模糊拼音输入方法。系统为用户提供模糊音定义的功能,根据用户定义,系统可自动处理拼音模糊问题。

6.4.5 面向数字键盘的数字拼音输入

使用标准计算机键盘输入汉字,直接按上述方式输入拼音串即可实现语句级拼音汉字输入。在手机等数字信息设备中,必须利用数字键盘(小键盘)输入信息。通过数字键盘输入汉语拼音,每个数字键对应多个拼音字母(如图6-6所示),因而增加了输入拼音的难度。这里把每个合法汉语拼音对应的数字串称为数字拼音,根据通常数字键盘的定义,用数字2~9表示。以数字拼音输入汉字,具体输入方式有两种:

① 数字拼音之间显式输入拼音间隔。若用0作为人工切分键,例如输入“哈尔滨是个美丽的城市”,则其相应的数字拼音串为“420370246074404306340540330243640744”。

这种方式由于需要人工切分数字拼音,因而增加了击键次数。

② 连续输入数字拼音,不用显式间隔。如上述数字串将“0”去掉结果为“423724674443634543324364744”。这种方式减少了击键次数,但需要自动确定拼音间隔,称之为数字拼音切分。

1	2 ABC	3 DEF
4 GHI	5 JKL	6 MNO
7 PQRS	8 TVU	9 WXYZ
*	0	#

图 6-6 数字键盘字符键的标准布局

具体地,数字拼音处理中的如下几个问题为数字键盘的拼音智能输入带来更大的难度:

① 一个未切分的数字串可以对应多种数字拼音的组合。根据实验结果,上述数字串可切分为 189 种合法的数字拼音串。如何确定最合理的切分结果是一个难题。

② 一个数字拼音通常对应多个合法的候选拼音,如数字串“226”对应“ban”,“can”两个合法拼音;而数字串“726”则对应“pan”,“pao”,“rao”,“san”,“sao”,“ran”等多达 6 个合法拼音。

以上两个问题使得一个数字串可能对应成千上万个标准拼音串。

③ 如果考虑简拼、模糊音等不同的拼音输入方式,则数字拼音的自动切分将更复杂,为音字转换带来更大的不确定性。

6.5 拼音预处理

拼音预处理包括拼音音节的自动断开和输入拼音的错误处理。

6.5.1 拼音流的切分

拼音流的切分就是确定拼音串中的各个音节。在语句级汉字输入方法中,用户输入一串拼音,而拼音串是由若干个有效拼音组成,每个有效拼音对应一个汉字。因此用户在输入过程中,为了得到正确的转换结果,输入拼音串时,应该在每输入完一个有效拼音后,就做一下断开,这样才能使转换模块的工作正确地进行。从操作心理来看,操作人员倾向于按有一定意义的短语或句子为单位进行短时记忆、持续输入。因此用户应该根据输入的内容连续输入拼音流,如果对拼音串进行人工切分,则容易打断人的连续思维,加重操作员的脑力劳动,因此对连续的拼音串进行自动的切分是有必要的。为简便起见,本书不讨论面向小键盘的数字拼音切分问题。

拼音音节的自动断开可通过事先在大规模拼音语料基础上建立统计模型,即通过拼音语言模型来实现。这种方法通过整句的上下文信息来确定切分结果,具体地,利用统计

信息来确定多个切分结果中的最大概率结果。这种方法能够达到很高的准确率,但在用户修改切分错误时会带来一些操作上的不便。在实践中,通过对汉语拼音构成的分析,我们发现,利用几条拼音边界切分的规则即可实现拼音音节的断开,少数切分歧义现象通过用户手工断开即可。

基于拼音规则的拼音切分纠错算法的基本思想:根据汉语拼音的规则,对用户输入的拼音串进行切分,采取边输入边切分的方式。

具体算法如下:

- ① 开始。
- ② 若当前输入的字母为 I,U,V 中的一个,则转⑫。
- ③ 判断当前拼音串是否为有效拼音,如果是有效拼音或有效拼音的一部分,则不需要切分,转⑬。
- ④ 判断最后输入的字母是否为声母,如果不是声母,则转⑥。
- ⑤ 若当前输入的字母是声母,且它前面的拼音串为有效的拼音,则在当前输入声母前断开拼音,转⑬,否则转⑫。
- ⑥ 判断拼音串的倒数第 2 个字母是否为 G,R,N 中的一个,如果不是,则转⑪。
- ⑦ 判断 G,R,N 前面的拼音是否为有效拼音,如果无效,则转⑩。
- ⑧ 若 G,R,N 前面的拼音为有效拼音,并且 G,R,N 和最后的字符组成的拼音为有效拼音或有效拼音的一部分,则在 G,R,N 前面断开,转⑬。
- ⑨ 若 G,R,N 前面的拼音为有效拼音,但 G,R,N 和最后的字符组成的拼音无效,则判断在 G,R,N 后面的断开是否有效;如果有效,则断开,转⑬,否则转⑫。
- ⑩ 若 G,R,N 前面的拼音为无效拼音,则判断在 G,R,N 后面的断开是否有效;如果成功,则断开,转⑬,否则转⑫。
- ⑪ 若最后输入的字母不是声母,并且倒数第 2 个字符也不是 G,R,N 中的一个,则判断在 G,R,N 后面的断开是否有效;如果成功,则断开,转⑬。
- ⑫ 若输入错误,当删除当前输入。
- ⑬ 结束。

算法要求输入的拼音串必须是一个有效的拼音、一个有效拼音的一部分或一个有效拼音后跟一个字母这三种形式中的一种。

算法需要有两个数据结构支持:一个是汉语拼音的声母表,保存汉语拼音中的所有声母,用于判断输入字符是否为声母。另一个是汉语拼音的有效拼音表,保存汉语拼音中的所有有效拼音以及它们的 ID,主要用于判断拼音串是否为有效拼音或有效拼音的一部分。在判断拼音串时,利用二分查找方法查找有效拼音表,给出当前拼音串的判断结果。

从上面算法可知,当拼音串中出现 G,R,N 时,容易发生切分歧义,因为 G,R,N 可以与前面的字符组成韵母,也可以与后面的零声母组成一个有效的拼音,因此要特殊处理。在某些情况下利用该算法会发生切分错误,需要人工干预。例如:用户想输入“西安”两个字,拼音为“XIAN”,因为它是一个有效的拼音,所以算法不对其进行切分,这时需要用户干预,对其进行人工切分。实践证明,需要人工切分的拼音出现的概率比较小,因此该算法提高了用户的输入速度,减轻了用户的负担。

6.5.2 拼音纠错

由于拼音序列是转换成汉字的基础和前提,拼音序列输入错误对汉字输入系统有直接的影响,它的正确性直接关系到转换结果。因此必须在拼音层次上检测和处理用户的输入错误,以保证汉字输入系统接受的是正确的拼音输入^[6]。

一般来说,用户在输入拼音串的过程中或多或少地会存在一些错误。引起输入错误的原因有以下几个方面:

(1) 与人们日常使用语言文字的习惯有关。通常人们都是用汉字进行书写,而不是用拼音来传递交流信息。由于很多人在日常生活中不是频繁使用拼音,因而对拼音不是特别熟练^[6]。

(2) 由于中国地域广阔,人口众多,不同地域的人的发音存在着一定的差异,存在着多种方言。例如南方人对卷舌与平舌、前鼻音与后鼻音的区分就不同于北方人,这也就是常说的“南方模糊音”^[6]。

(3) 用户真正关心的是转换后的汉字是否正确,因此一般情况下不对输入拼音的正确与否进行检查,因此要求系统必须自动检测拼音串的正确与否。

(4) 还有很重要的一点就是对键盘的熟练程度,这不但影响用户的输入速度,而且直接影响到用户输入的准确度。在拼音语句级输入中,如果用户输入拼音时出错,会影响到整个句子的转换结果。用户需要找到拼音出现错误的地方,并进行纠正,才能得到正确的转换结果,这样一来就增加了用户的工作量。因此为了减轻用户的负担,输入系统应该自动地在拼音层次上进行拼写检查和修正,这样可以提高输入速度和输入效率。

文献 6 以拼音语句输入法为基础,提出了在中文输入过程中对拼写的自动修改方法。该文方法通过对用户输入过程中所犯各种错误的分析,建立了一种有效可行的打字模型,并通过收集用户真实输入的数据,统计得到用户打字模型的参数。同时基于大量的中文文本,通过训练得到一个强大的中文语言模型,并与中文的打字模型相结合,采用类似语音识别的技术,修改用户输入过程中所犯的各种错误,得到最合适的汉字。拼写纠正不仅可以进行用户自适应处理,而且还适用于各种语言。

拼音纠错过程可以是一个明确的“预”处理过程,也可与后续音字转换过程结合起来进行。

文献 7 在语音识别系统中使用了拼音纠错技术。其纠错过程利用了音字转换的结果,具体过程如下:

- (1) 利用统计语言模型对输入拼音串进行初次转换,得到初次转换词串;
- (2) 对初次转换词串纠错,得到正确的候选拼音;
- (3) 对上一步得到的候选拼音再一次利用统计语言模型转换,得到最终词串。

拼音纠错技术具体的纠错方法采用了“词组匹配纠正法”,纠错分两步:首先找到错误拼音所在的位置,然后用“可信度”来衡量音字转换结果中汉字的正确程度。可信度越低,该汉字出错的可能性越大。可信度计算考虑以下几个因素:

- (1) 转换结果中单字词容易出错,多字词不易出错;
- (2) 某词与其前后相邻词的关系越小,其可信度越小,两个词的关系根据二者是否可

以按语法规则组成短语,或二者的词性关系,或二者构成的词串在语料中是否出现过来判断。

得到每个汉字的可信度后,假设可信度最小的汉字附近存在错误拼音,纠错就是纠正可信度最低的汉字及其相邻字所对应的拼音,进而产生一个候选的正确拼音序列,然后再一次应用统计方法进行音字转换。候选拼音的产生方式有两种:第一根据统计的二元词连接对,第二根据汉语的词组生成规则。当拼音正确率高于85%时,这种带纠错的音字转换方法可以提高音字转换的正确率。

6.6 音字转换的实现方法

语言建模尤其是统计语言建模技术是当前自然语言处理和计算语言学的主要研究对象,被广泛应用于面向大规模文本的真实应用中。统计语言建模技术就是针对大词表连续语音识别问题而提出的,目前的主要语言建模技术均可用于解决音字转换问题。本文仅针对汉字智能键盘输入问题对音字转换技术进行论述。关于这方面的论述有大量的文献报道^[8~19],其中文献[19]对当前主要汉字键盘智能输入处理软件的综述较好地反映了这个问题。目前成功应用的主要音字转换方法有以下几类。

6.6.1 基于理解的方法

基于理解的方法主要利用汉语语法知识来消化同音字、词,以及化解歧义分词。通常表述为计算机能够识别和处理的一系列固定搭配、公式和自定义规则。在学科分类中,该方法属于人工智能的自然语言理解分支。使用该方法的软件是最早出现,也是最理想化的智能输入软件。

该方法根据自动分词得到同音字、词的候选集,查找知识库得到相关的规则,再经过归约推理,得出转换结果。利用句内编辑实时修正转换错误和批量学习可以使得系统知识不断完善和充实,这就是自学习功能。

这种方法的优点是,由于采用了自行构造的“语法体系”,因而大体上能够包括最基本和较少歧义的汉语语法知识,因此系统正确率比较稳定。缺点是其语言的覆盖面较小,当输入语句的语法不规范时,不能做有效处理。另外该方法在建立知识库时,知识表达和知识获取均非常困难。

6.6.2 基于语用统计的方法

基于语用统计的方法主要利用语用统计的数据来消化同音字、词,以及化解歧义分词,在学科分类中属于统计学和运筹学范畴。

这种方法主要通过汉语中字与字或词与词之间的同时出现概率来完成汉语语用统计库的构造。主流的统计语言建模技术主要讨论的就是这种方法。由于概率信息空间十分巨大,所以统计信息库的存储通常采用各种索引和压缩技术。例如对基于词的N-gram模型,其N元概率信息可以采用图6-7所示的多级索引形式来表示。其中V代表词表,|V|表示词表规模,不同的词用整数编号,称为词ID(word ID),对于一级索引,词条的ID

号用位置信息表示。这种数据结构大大压缩了数据空间。

基于语用统计的方法通过概率计算求得词网格中的最佳路径。为减少状态空间的搜索时间,必须采用有效的搜索方法,比如采用动态规划的 Viterbi 算法或各种 A* 算法。

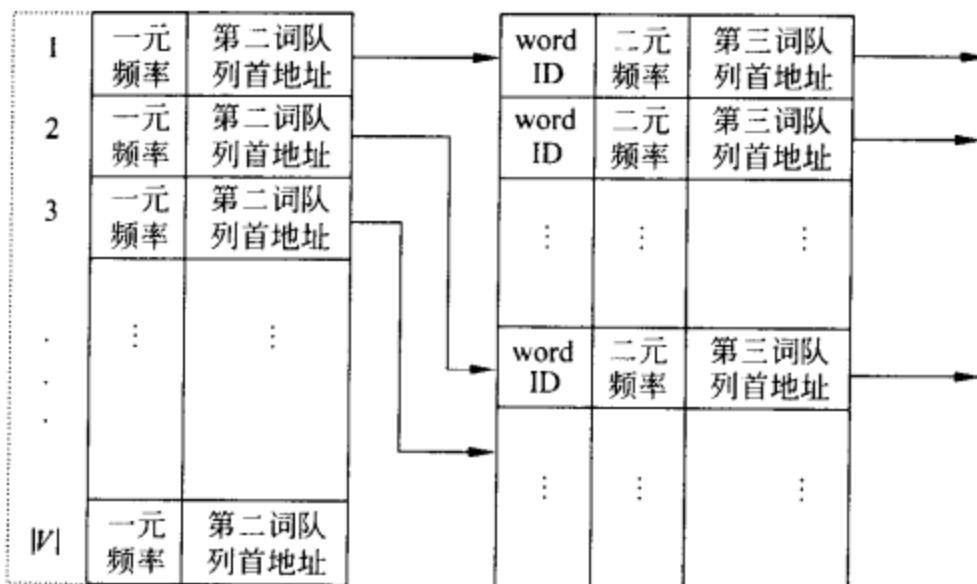


图 6-7 N -gram 语言模型数据结构

文献[14,17]以统计语言模型为基本框架,实现了基于理解和基于语用统计相结合的方法,在一定程度上,实现了两种方法的取长补短。

基于语用统计的方法适用于大规模真实文本的应用,是目前的主流方法。其优点是:对于已经进行过语用统计或具有相同类型的领域,系统的转换正确率比较高,语用统计具有一定的偏向性。对于某个用户而言,在使用过程中,语用统计库将会从最初的通用模型逐渐变为符合这个用户语用习惯的专用模型。

6.6.3 基于模板匹配的方法

基于模板匹配的方法寓汉语语法知识于巨量的短语串中,进而利用这些短语串来消化同音字、词,以及化解歧义分词。这种短语串通常称之为“模板词”。

这种系统通过模板词搜索引擎来完成汉语语法体系的组织任务。只有搜索巨量的语料,获取巨量的短语串,才有可能大体上包容汉语语法知识。

根据分词后的输入语句查找模板词库和语法规则库,然后进行匹配处理。如果匹配结果惟一,则不必再用概率推理;若存在两个以上的候选结果,则根据语法规则或概率推理做进一步判定,选出一个最有希望的可能结果作为输出。

该方法的优点是:对于已经搜索过模板词的或者具有相同类型的领域,系统的转换正确率比较高。但由于模板词数量巨大,因此对计算机存储空间的要求较高。

6.6.4 基于上下文关联的音字转换

前文基于语用统计的方法可认为是一种上下文字、词关联的实现方法。“青月亮汉字通智能输入软件”实现了一种基于模糊控制理论,利用上下文关联的语用环境来智能选择重码字词的方法。这在学科分类中属于自动控制分支非线性控制范畴。

音字转换过程根据字、词属性知识,语法知识和动态语用统计知识计算上下文关联重

码字、词的特征函数值,根据特征函数值确定最有可能的一个字、词。如果存在音字转换错误,则系统提供给用户错误修正机制,根据修正后的结果,系统机器学习机制自动更新知识库。这里将自然语言看成是一个模糊的集合,将汉字输入系统作为一个基于非线性控制范畴的模糊控制系统来对待,机器学习(或转换出现错误时的手工键选信号)相当于一个传感器,算法程序、汉语知识库和动态语用统计库可作为非线性调节器,以使得系统的键选率和平均码长逐渐趋于最优。

“青月亮汉字通智能输入软件”向上关联4个词语,向下关联1个词语,合计上下文关联5个词,这一调节机制涉及许多相互矛盾和相互牵制的受控参数,模糊集合的特征函数从[0,1]区间连续取值,可以较为准确地表现各种语言现象差异,获得比较好的结果。

值得一提的是,“青月亮汉字通智能输入软件”作为一种音码、形码和笔画码三位一体的通用智能输入处理平台,可为各种输入法增加上下文关联智能输入的后处理支持,增强了这些输入法的易学性和易用性。另外,这种输入法采用字段输入,不使用语句级输入,使语法规则简约化,易于知识表达,其优点是不但降低了键选率,还大大缓解了输入过程中“回头看”的问题。

6.7 小结

音字转换是汉字智能拼音键盘输入和汉语连续语音识别中的关键问题。本章首先介绍了这两个应用背景,之后讨论了拼音输入的各种形式、拼音音节切分、拼音(语音)纠错以及音字转换方法等主要问题。

参考文献

- 1 L. R. Baul, F. Jelinek, R. L. Mercer. A Maximum Likelihood Approach to Continuous Speech Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1983, 5(2): 179~190
- 2 F. Jelinek. Self-organized Language Modeling for Speech Recognition. (In): In Waibel and Lee, editors, *Readings in Speech Recognition*. San Mateo, California: Morgan Kaufmann, 1990: 450~506
- 3 黄泰翼. 人机语音通讯的发展及展望. 见:中国中文信息学会成立十周年学术报告会论文集. 北京:1991
- 4 方隶棠,吴文虎. 汉语语音识别的回顾、现状及展望. 见:中国中文信息学会成立十周年学术报告会论文集. 北京:1991
- 5 陈原主编. 汉语语言文字信息处理. 上海:上海教育出版社,1997
- 6 陈正,李开复. 拼写纠正正在拼音输入法中的应用. *计算机学报*, 2001, 24(7): 758~763
- 7 张瑞强,王作英,张建平. 带拼音纠错的汉语音字转换技术. *清华大学学报(自然科学版)*, 1997
- 8 王晓龙,王开铸. 声音语句输入的研究. *计算机学报*, 1994, 17(2): 96~103
- 9 万建成. 语音代码-汉字智能转换研究. *中文信息学报*, 1994, 8(2): 61~71
- 10 潘凌云,杨长生. 拼音、汉字计算机自动转换系统. *计算机学报*, 1990, 13(4): 271~275

- 11 关毅,王晓龙. 基于转移的音字转换纠错规则获取技术. 计算机研究与发展,1999,36(3): 268~273
- 12 倪小东,李人厚,余克艰等. 适用于信息设备的汉字输入法研究. 中文信息学报,2001,15(5):58~64
- 13 王轩,王晓龙. 大规模文本计算机音字相互转换技术的研究. 计算机研究与发展,1998,35(5): 417~421
- 14 王轩,王晓龙. 语音识别中统计与规则相结合的语言模型. 自动化学报, 1999,25(3): 309~315
- 15 章森等. 语句拼音-汉字转换的智能处理机制分析. 中文信息学报,1998(2)
- 16 张普. 智能化-汉字键盘输入法的最重要发展方向. 北京:中国标准出版社,1997(2)
- 17 蔡榕. 最优拼音汉字一次输入变换法及拼音汉字转换系统的实现. 见:第三届中文信息处理国际会议论文集. 1992
- 18 陈一凡,胡宣华. 汉字键盘输入技术与理论基础. 北京:清华大学出版社,1994
- 19 陈一凡,朱亮. 汉字键盘输入智能处理软件综述. 中文信息学报,2003,17(2): 60~65



自动文摘技术

自动文摘技术(auto-summarization)是自然语言处理的一个难点。当前,随着互联网的日益兴起,自动文摘技术引起了越来越多研究者的关注。本章将就自动文摘技术的发展过程和主要方法作一个简要介绍。

7.1 引言

文本摘要是指通过对全文信息进行处理,从中提取出最重要的内容,经过重组后生成比原文更简短、精练的文本(原文摘要)的过程。这一任务对人们来说并不陌生,摘要的历史其实远远长于计算机的历史,而摘要所涉及的体裁包括小说、新闻、科技论文等相当广泛的领域。即使到了今天,摘要也主要是由原文的作者或专业文摘员来完成。而随着互联网的普及,信息量爆炸性地增长,单单靠手工来完成对文本信息的摘要编撰工作除了无法应付互联网上不断涌现的海量信息外,还存在着以下缺点:首先,每个人对文本内容的理解必然受到其自身知识背景的影响,这使得人工撰写的摘要存在很大的主观性。不同的文摘员对同一篇文章所作的摘要往往差别很大,甚至同一个人对同一篇文章在不同时期做出的摘要也会有很大不同。其次,人工摘要也很难满足面向特定任务或者基于用户请求的文摘任务,比如在信息检索过程中不同的用户会用不同的关键字或自然语言指出他所特别关心的内容,这就需要在撰写文摘时必须根据用户请求,对包含特定信息的语句有所侧重,用户请求是很难预知的,显然由人来编写相关的摘要将无法满足用户对信息获取的实时性要求,这里还没考虑多用户同时提出请求的情况,这就是当前互联网环境下的真实情况。

互联网在提供了信息获取途径的同时却给人们留下了另一个难题,那就是如何在每天不断涌现的海量信息中更快速、准确地获得有用的信息。显然,通过阅读文摘而不是全文人们能够极大地提高获取信息的速度,更容易地获得有用的信息。因而对自动文摘技术的需求由过去的科技人员转向了普通用户,这也极大地鼓励了人们对自动文摘技术的研究热情。然而,自动文摘技术当前所取

得的研究成果还不像其他自然语言处理技术那样成熟并得到广泛应用,这主要是因为为一篇文章撰写文摘,首先必须真正“理解”原文的内容,这一任务对人来说是较容易完成的,但对计算机而言却显得异常困难。这就导致了自动文摘的质量,尤其是摘要的逻辑连贯性还很难与人工摘要相比。

一般地说,一个自动文摘系统可以分成:文本内部表示、文档分析、文摘提取、摘要生成4个部分,各部分的主要功能如下:

(1) 文档内部表示

文档内部表示主要是将用户输入的文档进行词语、语句、段落、章节等的划分,由于文档的结构特点,通常将划分后的结果用结构树的形式表示^[1]。

(2) 文档分析

文档分析是当前大多数自动文摘系统所做的主要工作,它主要对文档进行不同层次的分析,通常在这一阶段需要给出一个定量的标准,以衡量每一个文档基本单元(语句、段落或章节等)的重要程度,并根据分析结果将这一度量赋值给每个基本单元。

(3) 文摘提取

在这一部分首先需要确定文摘生成时用到的基本单元,即确定提取什么样的文档基本单元来生成摘要。现有技术的限制使得用低层次语言生成技术,如通过词语来生成语句,还很难达到令人满意的效果,因而往往选择较高层次的语言生成技术,如以单句^[2]、复句^[1,3]等为摘要提取的基本单元,其中大多数文摘系统选择语句(单句或复句)为提取的基本单元。然后根据给定的文摘比率(文摘比率=文摘长度/原文长度)提取出分析过程中得到的最重要的文摘基本组成单元。

(4) 摘要生成

摘要生成的主要任务是将文摘基本单元进行合成,并做进一步的加工,根据选取的组成文摘的基本单元的不同,这部分任务的难易程度也有所不同。例如,如果选择语句作为文摘的基本单元,则需要解决如指代消解(anaphora resolving)、关联词匹配等问题;如果以词语作为基本单元,则需要处理语句生成等问题。

以上我们将一个文摘系统根据处理过程分成了4部分,由于在现有技术下文摘生成部分往往只能做一些浅层的工作,而对于深层的工作语言生成等技术还无法实现,所以文档分析往往是现有文摘系统的核心部分,而主要的文摘技术也是针对这一部分任务而设计的。一个文摘系统所采用的文档分析技术可以分为以下3类^[4]。

(1) 浅层分析(surface-level)

浅层分析只对文档中蕴含的一些浅层的特征进行统计与分析,然后将其中的某些特征按特定的量化模型结合起来作为文档信息的量化度量,并据此选择出文档的核心内容。对于文档的浅层分析往往不需要复杂的文档内部表示,而只需要能够划分出文摘提取时的基本单元就可以了。这一分析方法涉及的主要特征有:

① 主题特征(thematic features) 指文档中出现的在统计意义上信息含量较高的词语,主要根据对短语的频度统计来进行分析。

② 位置特征(location) 指在文本、段落中的位置,例如新闻类文档,其第一段往往是对全文的概括性描述,而文档中每一段的段首或段尾也往往是对这一段主要内容的

描述。

③ 背景特征(background) 背景特征指在标题、子标题、文档首段或者用户查询中出现的词或短语。

④ 指示性词语和短语(cue words and phrases) 指文档内的一些特定的词语,如“总而言之”、“综上所述”、“本文”等,或者一些起着强调作用词语,如“特别地”、“重要的是”等,当然也包含一些特定领域的专有词汇,因为这些词汇往往对文档的中心主题有着很好的指示作用。

(2) 实体层分析(entity-level)

实体是指前文所述的组成文档的各个单元,如语句、段落等,每个语句或者每个段落皆可以称为一个实体。实现实体层的分析需要首先将文档转化成内部表示的形式,然后分化出文档的各个实体,并建立起文档实体间的相互关系。对文档实体及其相互关系建模,将有助于确定各个实体表述文档内容的作用。在实体层分析利用到的特征主要是各个实体之间的相互关系,包括:

- ① 相似关系(similarity);
- ② 相近关系(proximity),即文档各个单元之间的距离;
- ③ 同现关系(co-occurrence),指两个词语在同一个上下文中出现,当然,同现关系不一定只是指词语,也可以是词性同现或者词义同现等;
- ④ 基于辞典的关系,如上下位关系、同义关系、反义关系等;
- ⑤ 逻辑关系(logical relations);
- ⑥ 语法关系(syntactic relations),如语法分析树;
- ⑦ 语义表示关系(meaning representation-based relations),如基于断言-论点的关系。

(3) 话语层分析

话语层分析主要是对全文的宏观结构进行建模,这些结构包括:

- ① 文档格式,如超文本标记或者文档大纲;
- ② 用于区分文本主题的线索;
- ③ 文本的修辞结构(rhetorical structure),如议论文或者说明文的结构特征。

一个文摘系统往往不只涉及一种分析层次,更常见的情况是在一个文摘系统综合了多个层次、多种特征的分析与集成,以共同完成文档分析以及摘要提取任务。

本章的主要内容如下:在7.2节将讨论文档的内部表示方法,尽管不是所有的文档分析都需要事先对文本建立特殊的内部表示,但内部表示的建立将为以后文摘系统的深层分析打一个良好的基础。本章还将详细介绍各个层次的分析技术以及在典型文摘系统中的实现方法。最后一节描述自动文摘系统的评测方法。

7.2 文本的内部表示方法

显然,每个自动文摘系统由于采用的分析技术不同,其内部表示也不尽相同,我们不可能介绍每一种表示方法,在这里将主要介绍文献1中使用的一种比较通用的、根据文本

的自然结构来表示文档的方法,即文档结构树(structure tree of document)。

一篇文档按其自然结构可以划分为章节(section)、段落(paragraph)、复句(complex sentence)、分句(clause)、词语(word)等,它们之间从前往后依次存在包含关系,因而可以用树来表示,如图 7-1 所示。在图 7-1 中每个节点(根节点除外)由一个惟一的多层编号来表示,编号的层次由“.”分隔。由于除了章节以外的其他单元之间都会用明显的标记加以区分,如段落之间以段落标记进行区分,复句之间由表征复句结束的标点符号“。”、“?”、“!”等来进行分隔,因而可以很方便地确定相应层次的节点所包含的子节点。但文档的章节(也称意义段)之间则不一定具有明显的界限,所以文档表示中一个重要的工作就是进行文档的章节划分,关于文档章节的划分方法可以参见文献 5~8。

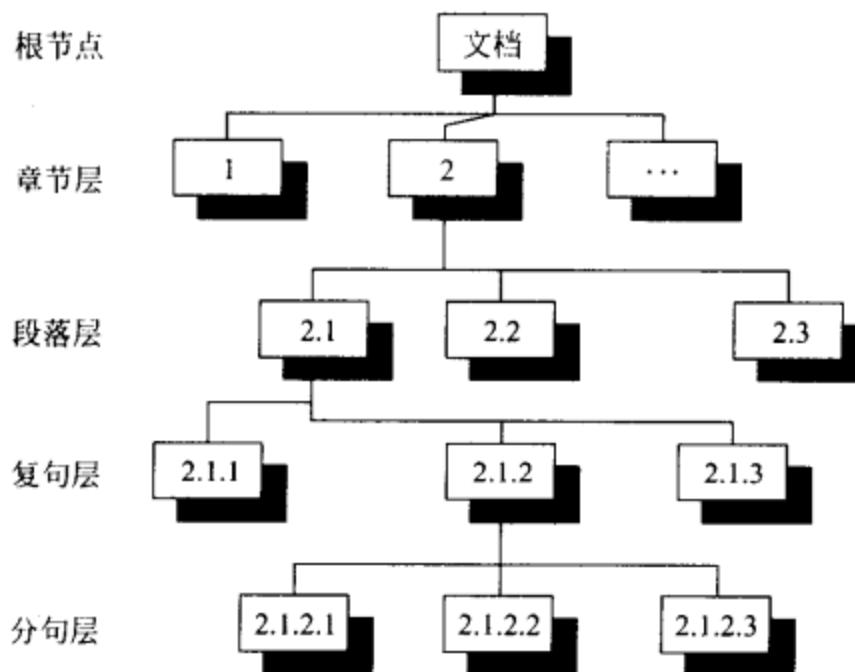


图 7-1 文章结构树

一篇文档除了根据自然的物理结构来进行表示外,在文档分析阶段往往会根据分析结果引入更深层次的表示,如话语结构树等,以反映文档各个基本单元之间的语法语义关系,7.5 节将对此进行详细描述。

7.3 基于浅层分析的文摘技术

如上所述,文本的浅层分析仅仅借助统计方法并只涉及对一些浅层的语言学知识的利用,这种方法并没有考虑到语言的语法、语义及语用知识,而这些知识是人们对文档内容进行理解的基础。因此,我们把基于浅层分析的文摘方法称为机械式文摘方法,而把基于实体层及话语层分析技术的文摘方法称为理解式文摘方法。浅层分析是在 20 世纪 50 年代中期出现的最早的自动文摘技术,60 年代以后人们将主要的精力用于研究基于理解的自动文摘技术,但从 90 年代以来,人们对自动文摘系统重新燃起的热情超过了对这两种技术的研究,尤其是基于浅层分析技术的文摘系统,由于其高效、简洁而得到广泛的应用。当然,在很多实际系统中往往综合机械式文摘和理解式文摘这两种分析技术来分析文档,能够在保证系统效率的同时提高文摘的质量,这样的技术称为复合文摘技术。

1958年Luhn在IBM704数据处理机上建立了有史以来第一个自动文摘系统，并于同年将其研究成果发表在IBM Journal上，首次提出了一整套的自动文摘方法，从而开创了自然语言处理的自动文摘领域。Luhn所采用的主要是用词频统计的语句加权方法从原文中抽取出主题语句来生成摘要。经过10年左右的发展，Edmundson则在Luhn的基础上综合运用了多种特征来进行语句的加权。Edmundson的研究涉及了对大多数浅层特征的使用，是一种比较典型的基于浅层分析的文摘方法，本节主要介绍Edmundson的文摘系统实现方法。

一个基于浅层分析的文摘系统的处理过程如图7-2所示，这里把整个过程分成两部分，即脱机处理和联机处理。脱机处理部分主要是手工建立自动文摘系统的训练语料库，并通过对训练语料库的统计来建立一个通用（指与特定文档无关）的特征库。联机处理主要完成三部分功能：首先根据输入文档以及通用特征库来抽取输入文档中所蕴含的特征信息，即文档特征库；然后根据文档特征库以及相应的权值计算方法来计算每个语句的权值；最后根据语句权值的大小，并考虑到文摘系统的连贯性来抽取文摘，生成摘要。

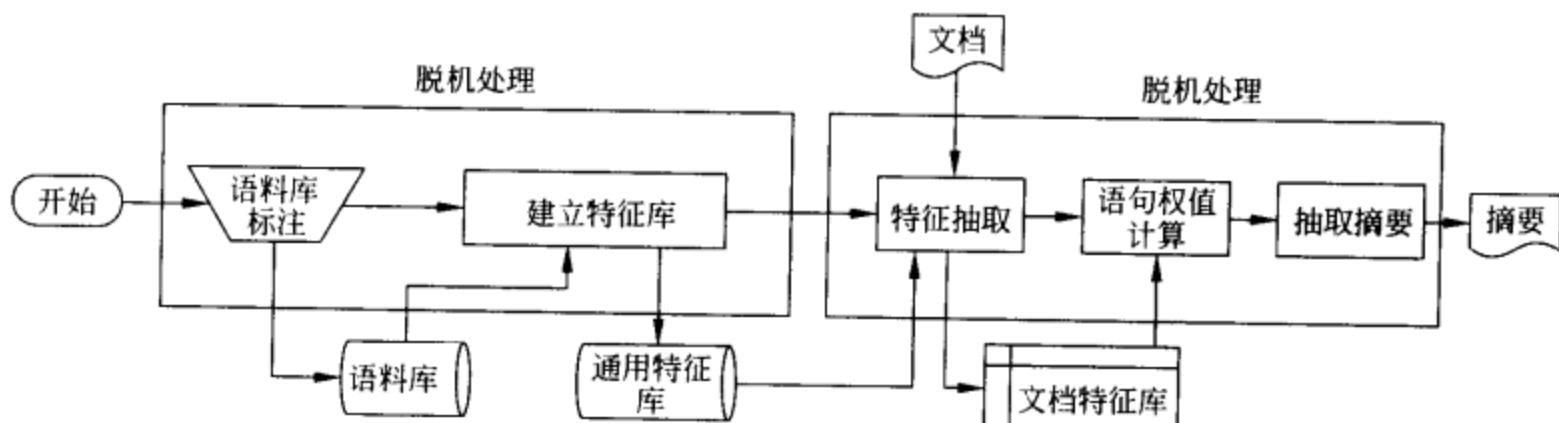


图7-2 基于浅层分析的自动文摘系统处理过程

7.3.1 建立特征库

为了建立一个通用特征库，Edmundson首先收集了目标领域内一定数量的文档来构成训练语料库，并对每一篇文本手工编写摘要；然后通过将手工编写的摘要与原始文档进行对比统计来建立通用特征库。

这里用到的特征主要指浅层特征，在Edmundson的算法中主要用到了以下4种特征信息。

1. 线索词词典

线索词词典由如下3个子词典构成：

- (1) 奖励词典(bonus dictionary)，如果一个语句中包含有该词典中的词或短语，则该语句作为文摘句的可能性就会增加，即可以获得正向加权；
- (2) 惩罚词典(stigma dictionary)，与奖励词典正好相反，包含该词典中词或短语的语句出现在文摘中的可能性将比不包含这样词语的语句更小，即该语句的权值会降低；
- (3) 无关词典(null dictionary)，该词典中的词条与语句的加权无关。

为了构造线索词词典,需要事先构造一个训练语料库,该训练语料库由 N 篇通常取自不同领域的文本文档组成,同时还需要为每篇文档准备相应的一篇或者多篇摘要。然后通过统计每个词语的如下信息来决定每个词语应该归入哪一个词典之中:

(1) 频度(frequency) 每个词语在语料库中出现的次数;

(2) 分布(dispersion) 在语料库中包含该词语的文档数目;

(3) 选择率(selection ratio) 词语出现在文摘库中的频度与出现在语料库中的频度的比值。

按如下原则构造候选线索词:

(1) 无关词候选,当词语的分布大于给定的域值 λ_d ,并且选择率在两个给定的域值 $\lambda_{st}, \lambda_{su}$ ($0 \leq \lambda_{st} < \lambda_{su}$) 之间的时候,该词语列入无关词候选列表;

(2) 奖励词候选,当词语的选择率大于 λ_{su} 时,列入该候选列表;

(3) 惩罚词候选,如果词语的选择率小于 λ_{st} ,则归入该候选列表。

除了以上三类候选列表以外,还有一类词语有可能对于文摘语句的选取产生影响,这类词语往往出现频度较低,但是其分布也很低,即只在较少的文档中出现,这类词语称为后备词(residue)。当一个词的分布小于给定域值,并且其选择率在 $\lambda_{st}, \lambda_{su}$ 之间时,可以列入该候选列表,该列表主要是在进一步调整其他候选词典的时候使用。

通常情况下需要对上述候选词列表结合语法或语用信息来进行进一步地调整,以获得最后的线索词典。目前还没有一个更加系统的方法来完成这一工作,所以需要根据实际应用提供的语言学资源决定对候选进行进一步筛选的方法。

2. 关键词词典

关键词词典的构造通常是采用词频统计来完成的,主要是基于这样的假设,即高频率的实词往往对于一篇文档所表述的内容起到比较重要的作用。事实上,关键词的抽取不仅涉及频度统计问题,对于汉语而言,还涉及分词、名实体识别等汉语自然语言处理的基本技术,以便识别出文档中的人名以及专有名词等,从而提高关键词抽取的准确率。这里仅介绍如何利用统计方法来建立一篇文档的关键词词典,即首先统计所有实词在一篇文档中的出现频度,并按频度的大小降序排列;然后按照给定的比率将特定数量的未出现在线索词典中的高频实词抽取出来,构成关键词词典,同时给每个关键词赋予一个权值,来表明该词语在抽取文摘句时的重要程度,也可以直接用该词语的频度作为每个关键词的权重。

3. 标题词库

标题词库包括标题以及各级子标题中出现的,同时又没有出现在无关词典中的实词。在多数情况下标题或者子标题中往往蕴含了一篇文档所描述的主要内容,所以标题中出现的词语对于文摘句的抽取有一定的参考意义。为了利用标题信息,同样需要给每个词语设置一个权值,只是该权值的设置与下一个浅层特征——位置特征,是密切相关的,其设置方法将在下面给出。

4. 位置特征

通常文章的作者都希望将一篇文章的主要内容尽早地呈现给读者,而在文章的结尾会再一次对全文内容作一个总结,这一特点在新闻类文章中体现得尤为明显。除了全文内容具有这一特点外,文本中每一意义段或者自然段往往也具有相似的特征。另外,在紧接着标题或者小标题后的语句往往更能反映全文或者小标题所要陈述的内容。Edmundson通过对100篇文章及其相应的人工摘要的统计也证实了上述特征,这一点主要体现在处于上述位置的语句中的词语选择率往往比位于文章中其他位置语句中的词语选择率高。根据这些特点,在进行文摘语句的抽取时,应该考虑到一个语句在文章或者段落中的位置特征,并将这种特征进行量化加权。

这里将语句的位置特征的权值分成了两部分,一部分称为标题词加权(heading weight),即首先根据训练语料库统计标题词在文摘句权值计算中的重要程度,对于文中一个特殊的句子,如果其包含的词曾经在标题中出现,则给其增加相应的标题词权重;另外一部分是位置加权,即对于某些特殊位置上的句子(如篇首句、篇尾句等)增加位置权值。

7.3.2 文摘句抽取

在7.3.1节建立了4个文本浅层特征库,每个特征库中的元素都被赋予了一个量化权值,以表明该元素在文摘抽取过程中的重要程度。文摘语句的抽取实际上就是通过这些特征来计算每个语句的权值,并根据用户给定的文摘比率来抽取出权值最大的前m个语句。由于涉及多个特征词典,所以每个语句的权值将按线性插值的方式进行累加:

$$W_s = a_1 C + a_2 K + a_3 T + a_4 L \quad (7-1)$$

式中, a_i 为每个特征类型的权重; C, K, T, L 分别表示用节所建立的4种特征对语句进行加权后得到的权值。

通常情况下,在文摘句被抽取出来以后,还需要做进一步的处理,比如进行指代消解,去除文摘句中内容重复或者相近的语句等,以便使建立的文摘具有较好的连贯性。

由上可见,在基于浅层分析文摘抽取方法中,最重要的工作是确定应该建立什么样的特征词典以及如何对词典中的每个元素进行加权。显然,为了获得更好的系统性能,需要一个具有相当规模的训练语料库,事实上,建立一个大规模的文摘语料库对于所有的文摘系统都起着相当重要的作用。

基于浅层分析文摘方法的一个主要特点是系统容易实现,并且系统的处理效率很高,适合于对大规模文档的在线处理。虽然这种方法只用到了文章的浅层特征,但却能获得良好的文摘性能,因而现在很多文摘系统都用到了浅层特征。

浅层分析的主要问题在于缺乏对文本内容的深层分析与理解,从而很难解决当前文摘系统所面临的主要问题,即如何保证文摘的逻辑连贯性。此外,如何准确地判断文本的中心主题,如何根据用户提交的关键词或者语句来抽取用户特别关心的内容等问题,仅靠文本的浅层特征也是很难解决的。

7.4 基于实体分析的文摘技术

这一节将介绍基于实体分析的文摘技术。正如 7.1 节所述,用于实体分析的特征范围非常广泛,而对如何使用这些特征则没有一个系统的方法。与 7.3 节一样,本节将通过一个基于实体分析的文摘系统^[9]来介绍实体分析的主要方法。

由于实体分析所用到的特征不再是简单的词频或位置信息,因而需要有更多的自然语言处理技术,如搭配关系的获取、名实体识别等的支持,特征提取算法要比浅层分析算法复杂。

7.4.1 特征提取

为了进行特征提取,首先介绍一个在特征提取中常用到的概念 TF · IDF,TF 指术语频度(term frequency),即术语在一篇文章中出现的次数。IDF 为文档频度的倒数(inverse document frequency),术语的文档频度指语料库中包含该术语的文章数目,显然一个术语的 IDF 是针对特定语料库的,同一个术语在不同语料库中有可能具有不同的 IDF 值,这里给出 TF · IDF 的一个比较常用的定义方法:

$$w_{ik} = \frac{\text{TF}_{ik} \cdot \log\left(\frac{N}{n_k}\right)}{\sqrt{\sum_j^t (\text{TF}_{ij})^2 \cdot \left(\log\left(\frac{N}{n_k}\right)\right)^2}} \quad (7-2)$$

其中, w_{ik} 是当第 k 个术语 T_k 相对于语料库中第 i 个文档 D_i 的 TF · TDF 权值, TF_{ik} 为 T_k 在 D_i 中出现的频度, N 是语料库中所有文档的数目, t 为所有术语的总数目, n_k 为给定语料库中包含有短语 T 的文档数目。

TF · IDF 权值还可以有其他的定义,事实上,式(7-2)的分子项也可以单独作为 TF · IDF 权值的一种定义。我们在第 8 章将进一步讨论这一量度。

1. 词语搭配库

7.3 节介绍了浅层分析中通常用到的对单个词语频度的统计,当然在浅层分析中也可以将式(7-2)定义的加权方法应用到对单个词语的分析中。但是由于每个词语在不同的语境中往往具有不同的语义,所以仅依赖单个词语出现频度的加权方法往往引入许多干扰,例如词语“打”在不同的上下文,如“有一打”、“打开水”、“打败”等中显然具有不同的语义,而简单的词频统计并不考虑这一点,在以上几种语境中的“打”都被认为是相同的,并且都被累加到“打”的词频中。为了解决这一问题,可以采用词语搭配来取代孤立的词语,即统计与计算每个词语搭配而不是单个词语的权值。

词语搭配关系可以分为语法上的搭配、习惯用法、事理上的搭配 3 种类型^[10]。由于我们通常采用统计的方法来寻找合理的搭配关系,所以不需要严格区分这种搭配关系到底是属于什么类型,而更关心搭配是否合理,即判断两个或多个词语是否构成搭配关系。为此可以借助第 2 章所介绍的互信息来作为判别准则,同时由于使用词语搭配的主要目

的是为每个词语提供相应的语境,以排除语义模糊性的干扰,所以这里主要考虑名词与名词之间的二元搭配关系。

综上所述,借助于统计方法构造一个名-名搭配的二元词语搭配库需要完成以下工作:

- (1) 准备大规模的原始语料库,一般语料库规模应该达到 100MB 以上;
- (2) 对原始语料库进行词性标注;
- (3) 统计并计算各个名词两两之间的互信息;
- (4) 根据存储空间以及处理时间的限制给定一个互信息的临界值,将互信息值超过该临界值的名词对加入词语搭配库。

当构造了一个词语搭配库以后,就可以将库中的每个词语搭配看成一个整体来计算其权重。

2. 名实体识别

名实体包括了人名、地名、机构名称等,是自动文摘技术中非常重要的特征,但是对于汉语而言,由于词语之间不存在明显的分界,如何识别词语边界已经成为一个比较难于解决的问题,而名实体的识别往往需要在汉语分词结果的基础上结合语法分析来完成。

研究表明,尽管特殊的人名对于判断一篇文章所涉及的领域有一定的帮助,比如像克林顿、丘吉尔等人名与政治具有密切关系,但一般而言,人名与文档所涉及的领域之间并没有明显的关系,而机构的名称则往往与特定的领域有关。

通过采用相应的名实体识别算法,可以建立一个包含人名、地名、机构名称等名实体的名实体特征库。

3. 语义词典

除了需要知道一个语义所属的语义类别外,通常还希望语义词典能够提供各个词语之间或者各个义素之间的语义关系,如同义、反义、上位、下位以及相似等。现在常用的语义词典有英语的 WordNet、汉语的 Hownet、同义词词林等。其中 WordNet 主要描述了词语之间的语义关系,而 Hownet 是通过描述各个义素(基本的语义单位)之间的关系来间接描述词语之间的语义关系,作为非商业性的研究工作,这两个词典都可以从网上免费获得。

在上述语义关系中,我们比较关心的是同义词。一方面,由于在同一文档中,作者有可能用不同的词语来表达同一个意义,比如在本文中可能用“文本”、“文档”或者“文章”来表示同一个事物,显然在计算短语的 TF·IDF 加权时,把它们看作同一个词更合理一些。另一方面,在面向任务的自动文摘系统中,往往需要根据用户提出的要求来重点提取用户所关心的内容,比如用户可能要求根据给定的关键词集或者查询语句来生成文档的摘要,而文档中可能只含有关键词集合或者查询语句中词语的同义词,这时候显然需要对关键词集或者查询语句进行同义词扩展。

除了同义关系,语义相似关系在文摘系统中也具有非常重要的作用,这一点将在 7.5 节中进行介绍。

7.4.2 文摘抽取

基于实体分析的文摘抽取算法与基于浅层分析的文摘方法是相似的,即都是通过建立好的特征库来对每个语句进行加权,最后根据给定的文摘比率来抽取权值最大的 m 个语句,以构成本文摘语句集。这里可以用式(7-2)定义的加权公式来计算每个特征的权值,而一个语句的权值等于它所包含的所有特征的权值的加权平均。

由于一个文摘系统可能涉及多种特征,与浅层特征不同,每一个实体特征的获取往往需要较为复杂的算法,并且特征获取的准确率会受到当前自然语言处理技术的限制,所以并不是系统用到的特征越多,系统性能就越好,而是需要综合考虑系统生成的文摘的质量、系统的处理时间、当前可利用的语言学资源及自然语言处理技术等多方面的因素来进行特征的选择。此外,对于系统所选择的每种特征,也可以在加权平均的计算中赋予不同的权重。

无论是特征的选择还是特征权重的确定,都需要用一个评测标准来定量评测文摘系统的性能。由于文摘系统的定量评测非常困难,所以将在 7.6 节专门加以讨论。

7.5 基于话语结构的文摘技术

话语结构分析在文摘系统中具有非常重要的作用,这是因为作者在撰写文章的时候,总会赋予一篇文章一定的结构。同样,在进行文摘抽取的时候,应该尽可能地获取并在文摘中保持这种结构。本节所要关心的问题是如何给文章的话语结构建立一个合理的模型,以便在生成文摘的时候能够对话语结构信息加以利用。

话语结构的类型分为衔接性(cohesion)和连贯性(coherence)两种^[11],这是两个比较相似但又不同的概念。为了明确它们之间的区别,可以参照王寅所编的《简明语义学词典》^[12]来将它们分别称为结构衔接和语义衔接,显然前者主要是指结构与形式上的衔接,而后者指语义上的连贯性。前者讨论的范围比较趋向于文本呈现的表面结构如何彼此串连,例如文本之间如何运用适当的连接词或副词来串连句子,或者在文法层次上,句子和句子之间如何依赖同样的主题词以及类似的语法结构来串连彼此;而后者是语义层面上更抽象的一致性,即表示文本实体之间是否基于相同的话题进行讨论。在实际的自然语言处理中要将它们区分开很难,因为判断语法层的连接需要借助于人脑中的认知模型来完成,而脑中认知模型的判断又是以文本中的语法结构为基础的,也就是说,人们在判断文本实体间的语义连贯性时,往往会参照实体之间的语法衔接关系,而作者在撰写文章时,也往往会借助形式上的衔接来反映意义上的连贯性,所以很难将这两者区分开来。有关衔接性与连贯性的更多讨论,可参见文献 13。

不过,在自动文摘系统的分类上,衔接性的处理主要涉及指代、省略、关联关系以及词汇层关系,如反复(reiteration)、同义(synonymy)、上位(hypernymy)等。而连贯性的处理偏重于段落、复句、分句之间的宏观(macro-level)关系,这种关系可通过明显的线索词反映出来,如“虽然……但是……”、“综上所述”等,也可通过统计方法来获得文本实体内部的语义连贯性。

7.5.1 基于词汇衔接的文摘方法

衔接性可以分为词汇衔接(lexical cohesion)、指涉(reference)、代替(substitution)、省略(ellipsis)、连接关系(conjunction)等,其中词汇衔接以及连接关系在文摘的抽取中具有比较重要的作用。本节将通过一种基于词汇链(lexical chain)的文摘方法介绍如何借助于词汇衔接来进行文摘抽取^[14]。

在基于词汇链的文摘系统中,词汇衔接是通过相应的词汇链来描述的,一个词汇链是指从一篇文档中抽取出来的,具有某种相似或者相关特征的词语链表,比如一篇文档中所有语义相似的名词性词语可以构成一个词汇链。通常词汇链的构成可以分成以下3步:

- (1) 选择一个候选词汇集,或者称为特征词集。在通常情况下我们不会把一篇文档中的所有词语都加入相应的词汇链,而是根据需要对可以加入词汇链的词语加以限制。比如,若需要通过词汇链来分析文章的主题,则需要词汇链中的词语具有较强的主题相关性,而像副词、介词等对于确定文章的主题没有太大帮助的词,可以不予考虑。
- (2) 对于每个候选词,根据相应的判别准则来寻找相应的词汇链。
- (3) 如果这一词汇链存在,则将候选词插入该词汇链中;否则,根据需要创建一个或多个词汇链。

在第(2)步时,需要有一个给定的判别准则,事实上还没有一个标准的方法可用来决定什么样的判别准则是最好的,所以读者完全可以根据不同的应用环境设计出不同的判别准则。作为例子,这里所用到的判别准则是根据语义词典,如WordNet, Hownet等来计算语义相似度。

为了将候选词插入相应的词汇链,需要判断计算每个候选词与所有词汇链之间的距离。每当将一个词语插入一个词汇链以后,该词汇链的组成实际上已经改变了,这必然会影响其后词语插入的决策。实际应用中往往还会有一个词语插入多个词汇链的情况,比如按如下方式来定义插入过程:如果一个词语的语义与某一词汇链中词语的语义之间有关联,则将该词语插入这一词汇链中。由于多义词的存在,显然会出现一个词可以插入多个词汇链的情况,而插入不同的词汇链会直接影响到其后的与其相关的词语的插入。而到底选择哪一个词汇链插入才是最好的,则需要遍历所有可能的插入情况,并对结果加以比较后才能得出结论。这是一个动态聚类问题,考虑到算法的复杂度,要想求得全局最优的聚类结果显然非常困难。另外,随着词汇数量的增多,聚类算法的复杂度有可能呈指数级增加,为了避免这种情况,需要在算法中引入相应的候选剪切方法。

此外,还可以引入文本切分算法将文档切分成若干个意义段,并对每个意义段建立相应的词汇链,然后再将各个意义段中的词汇链进行合并,完成对全文词汇链的建立。

在建立了词汇链之后,另一个问题是怎么样借助于词汇链来完成自动文摘任务。由上面的词汇链建立过程可以看到,每个词汇链中所包含的词语实际上集中反映了文档的一个主题,而文摘往往是为描述一篇文章的主要内容而建立的,所以一种比较直接的方法是找到几条最重要的词汇链,并根据这几条词汇链来抽取文摘句,即将含有这重要词汇链中每个成员的第一个语句抽取出来,组成文摘句。但按这种方法建立的文摘质量并不是很好,因为一条词汇链中的许多词语往往描述同一个概念的不同方面,并不是所有词语都能

够很好地表达该概念,因而需要选择词汇链中有代表性的词语,并按这些词语来抽取文摘句。然而,对如何判断一个词语是否更能代表该词汇链所表述的概念并没有成熟的方法。Regina & Michael 在对比研究后认为,在文本中的出现频度低于词汇链中词语的平均出现频度的那些词语更能代表该词汇链所表述的概念,而文摘句抽取也应该在这些能够代表词汇链的词语上来选择。当然,对于汉语,应该在实际应用中通过实验来寻找更好的词汇选择方法。

那么该如何确定什么样的词汇链更重要呢?显然,对于词汇链的重要度,需要有一个定量的评测。对这一问题同样缺乏一个系统而成熟的方法。作为一种简单的方法,可以根据需要选择尽可能长的词汇链,即用词汇链的长度来作为评价一个词汇链重要程度的标准。此外,也可以考虑采用上节所定义的 TF·IDF 权值,即用一个词汇链的平均 TF·IDF 权值来判断词汇链的重要性,具体哪一种度量方法更适合,则需要通过实验来进行评测。

除了采用词语来建立词汇链以外,也可以引入 7.4 节中介绍的名实体识别等自然语言处理技术,并用术语来替代单个词语构造词汇链,从而获得更好的文摘性能。

7.5.2 基于话语树的文摘方法

7.5.1 节介绍了如何利用统计方法来获取文章的衔接结构,并在此基础上生成文摘。本节将介绍如何通过话语树分析器来描述文档的连贯性,以及怎样通过话语结构分析来建立文摘。

话语结构通常是采用话语树的形式来描述的。话语树中各个节点之间可以由两种关系来表示:对称关系(symmetric relation)与非对称关系(asymmetric relation)。对称关系涉及两个或多个节点,这些节点称为核节点(nucleus),通常认为在对称关系中的所有节点对于作者希望表达的信息具有同等重要程度;非对称关系只涉及两个节点,一个核节点、一个辅助节点(satellite),通常认为对于传递作者的意图而言,核节点起着比辅助节点更为重要的作用。在非对称关系中,辅助节点依赖于核节点,并且在不同的依赖关系中以不同的方式改变核节点所表述的内容。

由于每个辅助节点又可能是另一个关系中的核节点(但不可能是另一个关系中的辅助节点),反之,每个核节点也可能是另一个关系中的辅助节点,因而整个话语结构是一种层次结构。

1. 节点关系分类

话语结构分析的目的是确定各个文本单元所对应的节点之间的相互关系,可以把节点间的关系分成以下 11 种类型,即解释、并列、递进、对立、选择、充分、必要、让步、无条件、因果、转折等。其中每种关系是属于对称关系还是非对称关系需要根据语言学知识来判断,比如在一个复句中,可以将并列、选择、充分、必要等关系作为对称关系,而将其他几种关系作为非对称关系来对待。当然,简单地将这些关系分为对称还是非对称对于自动文摘系统而言是不够的,还需要知道在每个关系中两个节点之间的关联程度。为了便于使用可计算的方式,这种关联程度通常需要用一个数值来定量表示,我们把这些数值称为

依赖关系的权值。依赖关系的权值可以根据经验指定,也可以通过机器学习的方法确定。

2. 话语结构分析

在话语结构分析方面,Marcu 给出了一个比较完整的分析方法^[15],这一方法主要借助于关联词的匹配规则来完成。研究表明,大约 50%以上的分句中存在关联词,借助这些关联词可以较为准确而快速地进行规则匹配,并建立起相应的话语树。但是这里存在两个问题:第一,如何分析那些不含关联词的分句;第二,由于关联词本身具有歧义,比如有的副词可以作为关联词使用,因而需要进行词语的兼类分析,以判别该词语是作为关联词还是副词。

对于第一个问题,可以借助一些启发式规则来部分解决^[16],而词语兼类现象可以由词性标注算法进行判别。

下面给出一个话语结构树的例子:

[1.1]{社会科学尽管因为作为其研究对象的社会现象、社会事实总是与个人的主观意志有着一定的关联,};[1.2]{因而有着主观性的外观,};[1.3]{有着历史的殊异性,};[1.4]{但社会科学本质上是整体的,是客观的。};[2.1]{我们在强调社会科学的独立性和客观性的同时,};[2.2]{还应该看到社会科学作为一门学科的相对性,};[2.3]{看到社会科学与其他科学,};[2.4]{尤其是人文科学的联系。}。

图 7-3 给出了上述文本片断所对应的话语树,这里采用带箭头的弧线表示非对称关系,其中箭头所指为在这一关系中的核节点,而对称关系用两条折线来表示。由于每个节点只能有一个父节点,所以如果在非对称关系中的箭头指向多个核节点,则认为该非对称关系中的核节点为箭头所指节点中的第一个节点。

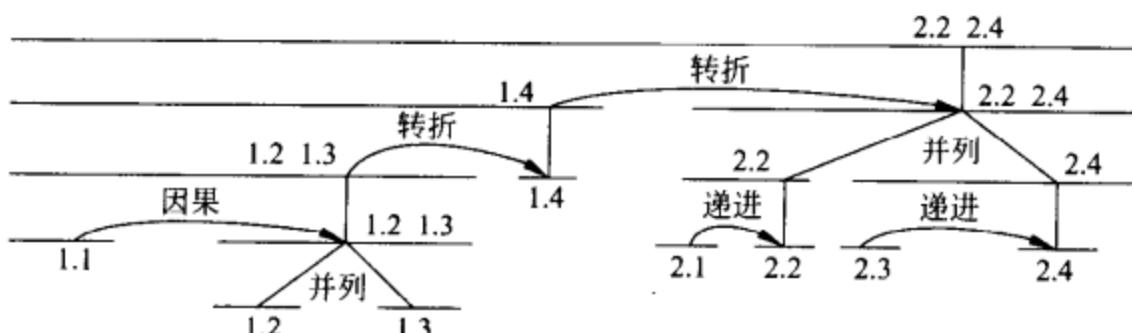


图 7-3 话语结构树

3. 基于话语结构树的文摘方法

话语结构树在给出了各个文本单元之间修辞关系的同时,也提供了建立文摘的重要信息,如上所述,非对称关系中的辅助节点往往对于传递作者的信息起辅助作用,因而不宜出现在文摘中。但是并不能简单地通过抽取核节点来构成本文摘,事实上有的辅助节点对于保证文摘的连贯性或者准确表达作者希望传达的信息具有重要的作用,我们可以借助于每种修辞关系的权值来表达这一信息。

在借助于图 7-3 所示的话语结构树生成文摘时,可以根据每个节点在第一次作为核节点时在树中所处的层次来建立一个偏序关系,并据此抽取文摘语句。这里定义根节点

为第 0 层节点,其直接子节点为第 1 层节点,依此类推。通过遍历结构树,可以得到所有节点重要程度的偏序关系

$$2.2, 2.4 > 1.4, 2.1, 2.3 > 1.2, 1.3 > 1.1$$

这里最重要的文本单元为 2.2 和 2.4,因而最短的文摘是由这两个单元组成的,即所生成的摘要应该为“我们应该看到社会科学作为一门学科的相对性,看到社会科学与人文科学的联系”。

上述方法实际上是假设所有非对称关系的权值都是 1,在很多情况下这一假设显然是不合理的,比如解说关系与因果关系同样为非对称关系,但是在解说关系中辅助节点与核节点之间的关系显然不如因果关系中这两个节点之间的关系密切,采用不同的权值表示这一差别能够更好地改善文摘系统的性能。当然,确定每种关系的权值需要额外的学习算法或者语言学知识。

7.6 文摘系统评测方法

文摘系统的评测一直是较难解决的问题,当前最可靠的方法仍然是依靠专家来人工评测。缺乏对文摘质量的客观评测,这对于文摘系统的开发、尤其是系统优化无疑是很大的阻碍,因而这方面的研究对于自动文摘系统也具有非常重要的意义。本节将主要介绍现有的文摘系统评测方法,包括人工评测以及自动评测。

人工评测可以通过语言学家阅读系统生成一定数量的摘要,并根据摘要的连贯性、流畅性、概括性等性能给每一篇摘要一个主观的评价,比如采用打分的方法,给出优、良、中、差等成绩,然后再根据所有文摘的平均性能评价文摘系统的整体性能。由于不同人对同一篇文本所作的摘要差异非常大,即不同人对于用什么样的语句更能反映原文的中心内容会有不同的看法,这种评测很难准确反映系统的性能。

另外一种人工评测方法是采用对比的方法。首先由多个语言学家对一定数量的文本人工撰写摘要(抽取式文摘),然后再由系统对相同的语料自动生成文摘。在进行对比评测时,将系统文摘与语言学家的文摘混合到一起,让第三方评测人员(即除了参与撰写摘要的人员以外的评测人员)对每篇文本的所有文摘根据文摘质量进行对比排序,然后统计系统生成的文摘在所有文摘中的排序结果,并根据这一结果来判断系统性能。由于有了比较标准,因而对文摘质量的评价会比第一种方法客观。但是,这两种评测方法都需要人工完成,还是会不可避免地受到人们的主观因素影响,更主要的是评测工作量大,很难将评测结果用于系统优化。

为了克服人工评测的缺点,需要建立一套完整的自动评测体系。然而,文摘本身的特点使得建立一套可靠的评测体系面临很大的困难。当前的自动评测方法主要是针对抽取式文摘,通过将系统生成的文摘与已有的人工摘要集进行字符串比较来完成。为了具有可比性,由语言学家建立的人工摘要集也必须是采用抽取的方式来建立的,即首先确定系统生成摘要时的基本单元;然后在人工建立摘要时,采用原文中对应的基本单元,而不能将小于基本单元的语言元素单独抽取出来重新组织,或者重新撰写原文所没有的语句。当然,如果能够引入语义相似度等度量来比较两个文本实体之间的语义相似性,而不是简

单地进行字符串匹配,上述限制也可以适当地放宽,比如在建立摘要时可以撰写与原有语句意义相近的语句等。

在自动评测时常用到的系统性能指标通常为召回率(recall)和准确率(precision),在采用简单的字符串匹配的情况下系统对每篇文章生成的摘要的召回率与准确率定义如下:

$$\epsilon_r = \frac{|R \cap A|}{|R|} \quad (7-3)$$

$$\epsilon_p = \frac{|P \cap A|}{|A|} \quad (7-4)$$

式中, A 为系统生成文摘所包含的基本文摘单元的集合; R 和 P 分别对应于人工建立的召回率基准评测集以及准确率基准评测集,这两个集合可能是相同的,即可直接将人工建立的摘要同时作为评测召回率与准确率的基准测试集,尤其在每篇文章都只对应一篇人工摘要时,常采用这一方法。而在一篇文章对应多篇人工摘要的情况下,这两个基准测试集有可能是不相同的,比如假设每篇人工摘要都具有100%的召回率和准确率,因为人工摘要通常是由专家建立的,而在现有技术下所建立的系统只要能达到任何一个专家建立文摘的水平就可以了,在这种情况下这一假设显然是合理的。为了满足这一条件,可以采用所有人工摘要的并集作为召回率的基准测试集,而将所有人工摘要的交集作为准确率的基准测试集,显然这两个集合是不一样的。

对于更为复杂的评测方式,例如采用语义相似度来进行文摘质量评测,其召回率与准确率的计算比上述两式要复杂一些,读者可根据不同的应用系统对上述两式进行调整,以获得相应的召回率与准确率计算方法。

虽然召回率与准确率能够反映系统不同方面的性能,但在很多情况下需要一个惟一的度量来评测文摘系统的性能,比如在进行系统参数的自动优化时,可以用 F 量度将这两个量统一起来^[17]:

$$F = 2\epsilon_r\epsilon_p / (\epsilon_r + \epsilon_p) \quad (7-5)$$

自动评测的主要优点是评测结果比较客观,不会受到主观因素的影响,同时评测效率远高于人工评测。其主要缺点是只能评测抽取式文摘,无法判断文摘的连贯性、流畅性等无法量化的指标。尽管如此,上述自动评测方法对于文摘系统的参数优化以及特征选择仍然具有重要的指导作用。

当然,要推动文摘系统的发展,进一步完善文摘系统的定量自动评测体系将是一个非常重要的课题。

7.7 关键词自动抽取

文本关键词和文本的标题、摘要一样,也提供了一种可以迅速了解全文信息的重要途径。另外关键词的实际应用还有:

- (1) 利用生成的关键词为大量的文档集合自动生成超级链接;
- (2) 通过将文中的所有关键词标出,以便提高阅读和理解的效率;

(3) 用于搜索引擎的返回结果定制……。目前搜索引擎的一个突出问题是返回的网页有相当一部分不是用户需要的。所以如果让用户同时看到网页的关键词,会在一定程度上帮助用户选择正确的链接。如 Keegle 就是这样的系统。

目前几乎所有的网页都没有关键词,大多数的科技文献需要作者手工标注关键词。与自动文摘相比,关键词抽取技术生成的结果不存在连贯性和可读性的问题,从而在一定程度上克服了由于指代消解等问题的研究精度低而对系统最终结果产生的影响。

和关键词抽取(keyword extraction)关系比较密切的一个术语是主题词抽取(keyword assignment)。主题词抽取是指从领域的主题词表中选择能够代表文章内容的词。而关键词抽取则允许出现自由词,即抽取的词可以不在主题词表中。

关键词自动抽取一般需要解决三方面的问题:

1. 候选关键词短语抽取

通常所说的关键词(keyword)有相当一部分是短语,所以称作关键词短语(keyphrase)要更加恰当一些。为了和传统说法兼容,后面凡是提到关键词的地方一般是指关键词短语。关键词短语比关键词包含的信息更加丰富,也更具有概括能力,如“关键词抽取”比“关键词”或者“抽取”概括性更全面。关键词短语一般由两个词或者三个词组成。有三种比较常用的获取候选关键词短语的方法:第一种方法从文中抽出的 N -gram 词对作为候选关键词短语;第二种方法抽取名词短语 chunk 作为候选关键词短语;第三种方法根据词性标注的匹配模式抽取词对,如 Anette hulth^[18]的工作,她总结了 56 个词性匹配模式用于辅助关键词短语自动抽取。研究表明,名词短语的 chunk 以及满足一定词性模式的规则更加适合进行关键词短语的自动抽取。而相比之下, N -gram 则表现出一定的武断性。

2. 词的重要性评价

统计表明作者手工标注的关键词有 65% 到 90% 可以在原文直接找到。这意味着,通过对文章关键词进行重要性评价,并直接从原文中选择关键词短语,基本上可以达到关键词抽取的目的。

很多研究将关键词的重要性评价看作一个有指导的机器学习问题,这一思想是首先由 Turney 提出的。评价词的重要性需要综合考虑相关特征。比较重要的特征有:词的全文频率、词的首次出现位置、词性等等。

在评价词的重要性时,还需要考虑词汇集聚现象。例如,一篇关于奥运天气的文章中,出现“气象”、“气温”、“天气”等词,但这些词的文本频率都较低。如果不考虑语义特征,会导致最后抽取的词没反映“天气”这一特征。因此可以通过采用本章前面提到的词汇链计算对文章进行词汇集聚分析,使每个链上比较重要的词的权重都得以增加。

3. 冗余消除

一般从文中抽取的关键词为 3~8 个,而一篇网络上文章的惟一词的个数平均为 300 左右。因此要求抽取的关键词要比较精炼,能够反映原文的主要内容。但是经过关键词

的重要性评价处理后,权值较高的词中可能存在冗余。同义词和某些存在较强语义相似度的词会造成冗余,例如“气象”和“天气”这样两个词就不适合同时出现在关键词中。另外,存在简略关系的词也容易产生冗余,如“奥林匹克委员会”和“奥委会”、“自然语言处理”和“NLP”等等。因此为了消除冗余现象,可以首先构造一个按照重要性程度排序的候选关键词集合,并对其进行冗余处理后输出最终的关键词。

7.8 小结

本章介绍了当前自动文摘技术中三种主要的分析方法,即浅层分析、实体分析以及话语结构分析,并简要介绍了自动文摘系统的评测方法。事实上,除了上述三种主要的文摘方法以外,还有一种基于特定框架或者模板的文摘技术,与前三种方法不同的是,这一方法通过引入领域相关的模板,将研究的重点转移到了摘要的生成上。由于在现有技术条件下,这一方法往往只能限于特定领域的摘要生成,所以本章没有进行介绍。但随着自然语言处理技术,特别是超大规模语言学知识库的发展,这一方法仍然具有很好的前景。此外,文摘系统的评测仍然是今后自动文摘研究领域需要进一步完善的重要课题。另外,多文档自动文摘、基于内容的多媒体信息摘要生成等文摘相关领域的研究也越来越引起广大研究者的兴趣,成为未来自动文摘研究的重要课题。

参考文献

- 1 刘挺,王开铸. 基于篇章多级依存结构的自动文摘研究. 计算机研究与发展,1999, 32(4): 479~488
- 2 沈洲,王永成,许一震,吴芳芳. 一种面向新闻文献的自动摘要系统的研究与实践. 计算机工程, 2000, 26(9): 70~72
- 3 Simone Teufel, Marc Moens. Sentence Extraction and Rhetorical Classification for Flexible Abstracts. In: Intelligent Text Summarization-Papers for AAAI Symposium. Menlo Park, California, USA: AAAI Press, 1998: 16~25
- 4 Inderjeet Mani, Mark T. Maybury (eds). Advances in Automatic Text Summarization. London: MIT Press, 1999
- 5 M. A. Hearst. TextTiling: segmenting text into multi-paragraph subtopic passages. Computational Linguistics, 1997, 23, 33~64
- 6 J. C. Reynar. An automatic method of finding topic boundaries. In: Proceedings of the 32nd Annual meeting of the Association for Computational Linguistics, Las Cruces, New Mexico, 1994: 331~333
- 7 S. Chan, B. K. T'sou. Segmentation of Chinese discourse in content-based information retrieval. In: RIAO'2000 Proceedings. Paris, 2000
- 8 Beeferman D., Berger A., Lafferty J. Statistical models for text segmentation. In: RIAO' 2000 Proceedings. Paris, 2000
- 9 C. Anone, M. E. Okurowski, J. Gorlinsky, B. Larsen. A Trainable Summarizer with Knowledge Acquired from Robust NLP Techniques. In: Advances in Automatic Text Summarization. Cambridge, MIT Press, 1999: 71~80
- 10 史玉柱. 语法论稿. 北京:北京语言学院出版社,1995(24~32)

- 11 M. Halliday, R. Hasan. *Cohesion in Text*. London: Longmans, 1996
- 12 王寅. 简明语义学词典. 山东: 山东人民出版社, 1993
- 13 刘郁青. 新闻写作连贯性之研究:[硕士论文]. 台湾国立政治大学新闻研究所, 2001
- 14 Regina Barzilay, Michael Elhadad. Using Lexical Chains for Text Summarization. In: Proceedings of the ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization. Madrid, Spain, 1997: 10~17
- 15 D. Marcu. The Rhetorical Parsing of Natural Language Texts. In: Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics. Madrid, Spain, 1997: 96~103
- 16 Simon H. Corston-Oliver. Beyond String Matching and Cue Phrases: Improving Efficiency and Coverage in Discourse Analysis. AAAI Symposium, Stanford, California, USA, 1998: 9~15
- 17 Van Rijsbergen, C. J. *Information retrieval* (second edition). Butterworths, London, 1979
- 18 Anette Hulth. Enhancing Linguistically Oriented Automatic Keyword Extraction. In: Proceedings of the Human Language Technology conference/North American Chapter of the Association for Computational Linguistics Annual Meeting (HLT/NAACL 2004). Boston, May 2004



第8章

CHAPTER 8

信息检索技术

本章介绍信息检索技术(Information Retrieval)的基本原理和方法。所谓信息检索,是指将信息按一定的方式组织和存储起来,并根据用户的查询字串,从表示信息的非结构化数据,特别是非结构化的文本数据中找到与查询字串相关信息的过程。信息检索这一称谓,最早是 1952 年由 Calvin N. Mooers 提出的,其原意包括海量信息的存储和查找两方面内容,现在这个概念通常更加侧重于信息的查找方面。信息检索按照信息源的不同可以分为光盘数据库检索、网络数据库检索和互联网信息检索三部分。由于计算机信息检索具有速度快,效率高,数据内容新,信息容量大等特点,已成为人们日常工作和生活中获取信息的主要手段之一。特别是近年来,随着国际互联网技术的普及和发展,信息检索技术与网络通信技术和语言处理技术彼此融合,互相借鉴,共同发展,成为当前十分活跃的研究热点之一。

8.1 信息检索综述

本小节介绍信息检索的基本概念和术语、信息检索系统的基本构成及评价、信息检索技术的发展史等。

8.1.1 信息检索的定义与术语

信息检索是指从非结构化的数据记录,特别是包含自由格式的自然语言文本的数据记录中获取与用户的信息需求相关的数据记录的系统、方法与过程。其他类型的数据,如图像、音频和视频数据等也是非结构化的,但是在当前信息检索的研究中,非结构化的数据记录通常特指自然语言文本数据记录,又称文档(document)。将大量非结构化的数据记录按照一定的方式组织和存储起来,构成的数据记录的集合称为信息检索中的数据全集(collection)。在此强调“非结构化”,是为了将信息检索与数据库数据检索加以区分,前者是从非结构化的数据记录中获取用户信息需求,后者是从结构化的数据记录中获取用户的信息需

求。自然语言的丰富性和二义性,使得信息检索在精确性和召回率等方面都比数据库数据检索低,其难度也更大。当然,信息检索系统的实现通常要借助于数据库管理系统。

给定一个数据全集,信息检索过程可以描述为根据用户特定的信息需求(information need),在数据全集中获取所有和仅有的与用户信息需求相关的文档,并将这些文档按照相关性(relevance)的大小由大到小地排列(rank)。用户特定的信息需求由查询(query)来表达,换句话说,查询是反映用户信息需求的字符串,这个字符串可以是关键字序列,也可以是一个布尔表达式,或者直接用自然语言表达的问句。相关性是信息检索中的一个重要的概念,是指信息检索结果符合用户信息需求的程度。显然,这是一个带有主观色彩的量度,它不仅与查询和数据全集有关,而且与信息检索结果的正确性、现实性和权威性有关,甚至与用户个人的主观判断有关。

8.1.2 信息检索系统

一个信息检索系统(information retrieval system)是一个能够对数据全集的数据记录进行存储、组织与维护,并根据用户查询获取相关信息的系统。一个典型的信息检索系统如图 8-1 所示。

在图 8-1 中,矩形文本框表示信息处理模块,椭圆形文本框表示中间数据或者输入输出信息,圆柱形文本框表示系统资源。一个信息检索系统由 8 个基本处理模块和 2 大系统资源组成,基本处理模块分别是: 用户接口模块、用户查询文本操作模块、文档文本操作模块、用户查询处理模块、索引构建模块、数据库管理模块、搜索模块、相关度排序模块等。两大系统资源包括整个系统公用的语义词典(thesaurus)和以数据库形式存放的数据全集。语义词典的内容为系统词汇及其语法语义信息,还包括停用词表(stoplist)和词形转换表等系统资源。下面分别介绍。

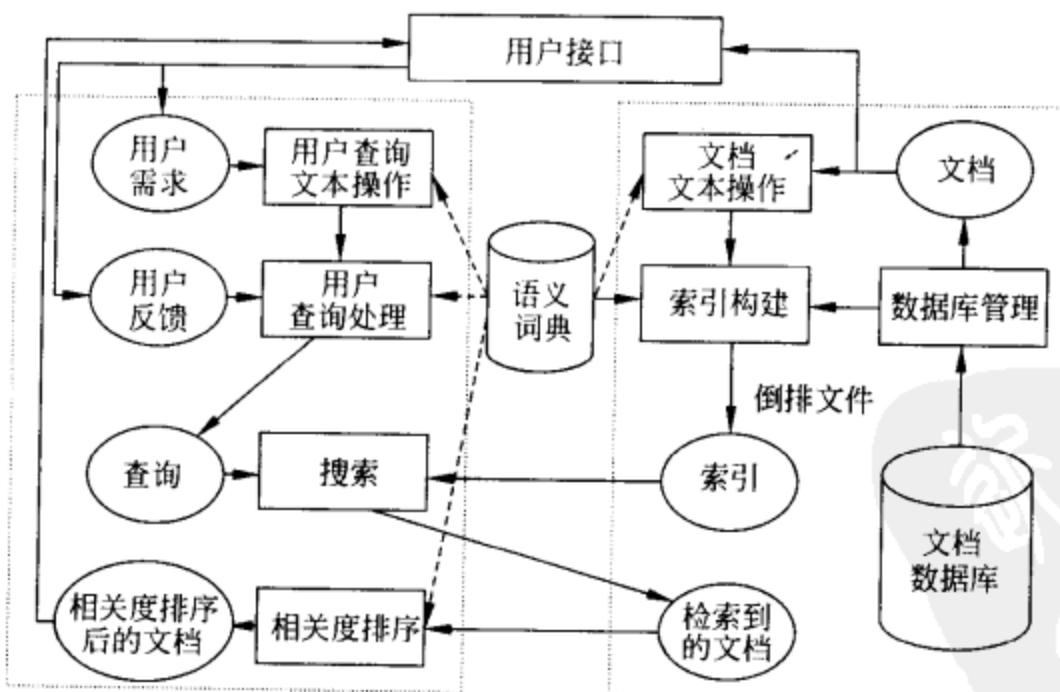


图 8-1 信息检索系统的系统结构

(1) 用户接口模块

用户接口模块的作用是与信息检索系统的用户交互信息。主要包括接受用户的查

询,根据用户对信息检索结果的反馈调整信息检索系统的有关参数,显示用户查询的结果等。

(2) 用户查询文本操作模块

用户查询文本操作模块对用户的查询字串进行过滤停用词(stop word)、词干抽取(stemming)等处理,并转换为机器内部的用户查询表示格式。停用词是指因不具有确切语义而未列入索引词表的高频词。由于停用词不具有实际意义,因此对信息检索结果没有影响,又因为它们大量地出现在文档中;所以如果不“停用”,而对其建立索引,则需消耗大量的时间与空间资源。因此信息检索系统都要收集停用词,建立一个停用词表,对在停用词表中列出的停用词,直接进行过滤处理。例如,英语中的“of, the, to”,汉语中的“的、在、啊”等都是停用词。考察一个词是否为停用词,除了看它的频度是否相对较高外,关键在于看它是否具有确切的语义。停用词通常包括连词、介词、代词等虚词,甚至包括像“干、做”等一些一词多义(语义不确切)的动词。所谓词干抽取,是指将一个词的各种不同的词形变化替换为称为该词词干(stem)的统一形式的过程。例如:将“engineering, engineered, engineer”全部替换为“engineer”。显然,这一过程仅对英语等具有同一词形多种变化特征的语言是必要的,对汉语这样的没有词形变化的语言则不需要。由于在相关度排序模块需对用户查询和索引文档进行逐词匹配,因此必须在匹配之前统一词形。

(3) 文档文本操作模块

文档文本操作模块对文档数据库中的文档进行停用词过滤、词干抽取等处理,并将文档转换为机器内部的文档表示格式,供建立索引模块处理。

(4) 用户查询处理模块

用户查询处理模块的作用是对用户查询的词汇进行同义词扩充,或者根据用户对信息检索的倾向性对查询的词汇进行转换处理。由于语言中大量存在着同一语义多种表达的现象,用户查询中的词汇也有多种不同的表达方法,如“电脑”的同义词有“电子计算机”、“计算机”等。通过对用户查询的词汇进行同义词扩充,用其他词汇表达的与用户查询词汇相关的文档才能被索引和抽取出来,并返回给用户,从而使系统具备了处理同一语义多种表达的能力。

(5) 索引构建模块

索引构建模块的作用是建立从词汇到该词汇出现的文档(编号)的倒排索引表(inverted index),从而对用户查询中的词汇进行快速定位。

(6) 数据库管理模块

数据库管理模块的作用是将文档以数据库的格式存储、管理、编辑和访问,通常采用基于SQL的开放式大型关系数据库管理系统,如Oracle, Sybase等。

(7) 搜索模块

搜索模块的作用是根据用户查询,借助倒排索引表和数据库管理模块从数据库中抽取出包含用户查询关键字的文档。

(8) 相关度排序模块

相关度排序模块逐一计算用户查询与搜索模块返回文档的相关度,最后将这些文档按照相关度由大到小的顺序排序。

整个信息检索系统可以分为检索子系统和信息存储管理子系统两大部分,如图 8-1 虚线框所示。左方的虚线框是检索子系统,它接受用户查询,对用户查询词汇进行停用词过滤和词干抽取等处理,然后进行同义词扩充等转换处理,转换为用户查询的机器内部格式(通常是经过同义词扩充的关键词序列)。图 8-1 右方的虚线框是信息存储管理子系统,通过由信息存储管理子系统预先建立的倒排索引表,找到包含这些关键字的所有文档,并将它们作为候选文档,然后逐一计算用户查询与候选文档的相关度,将候选文档按照相关度由大到小的顺序排序,并返回用户。信息存储管理子系统将数据全集文档存储于数据库中,并提供对这些文档的编辑、增删等操作;对文档进行停用词过滤、词干抽取等处理后生成的副本也存储于数据库中,并对文档的副本建立倒排文件供检索使用。

信息检索系统的终极目标是使满足系统用户信息需求的开销(overhead)最小。所谓开销,是指从用户向系统输入了一个查询开始,到他读到了包含他的信息需求的文档为止的全部时间。开销这一指标,由于包含了过多的主观或不可控因素,而难于成为衡量一个信息检索系统的定量指标。人们通常采用其他的量度来评价一个信息检索系统的优劣。

8.1.3 信息检索系统的评价

一个系统在实际应用中的时间和空间消耗是衡量一个系统优劣的重要指标。对于一个信息检索系统来说,响应用户查询的时间、建立倒排索引耗费的系统空间等都是评价该信息检索系统的定量标准。然而,这些标准都没有考虑到评价信息检索系统的一个核心性因素,即“相关性”。一个信息检索系统,无论其耗费多少时间和空间资源,归根结底是要满足用户信息需求,因此“相关性”是评价一个信息检索系统的核心所在。

两个最常用的基于相关性的系统评价指标分别是精确度(precision)和召回率(recall)。精确度是检索获取的相关数据记录个数与检索获得的所有数据记录个数的比值;或者给定一个检索获取的数据记录,它与用户查询相关的概率。召回率是检索获取的与用户查询相关的数据记录的个数与数据全集中所有与用户查询相关的数据记录个数的比值;或者一个相关数据记录能被检索系统获取的概率。二者间的关系如图 8-2 所示。对于任意给定的用户查询 q ,令 A 为信息检索系统获取的数据记录的集合,令 R 为数据全

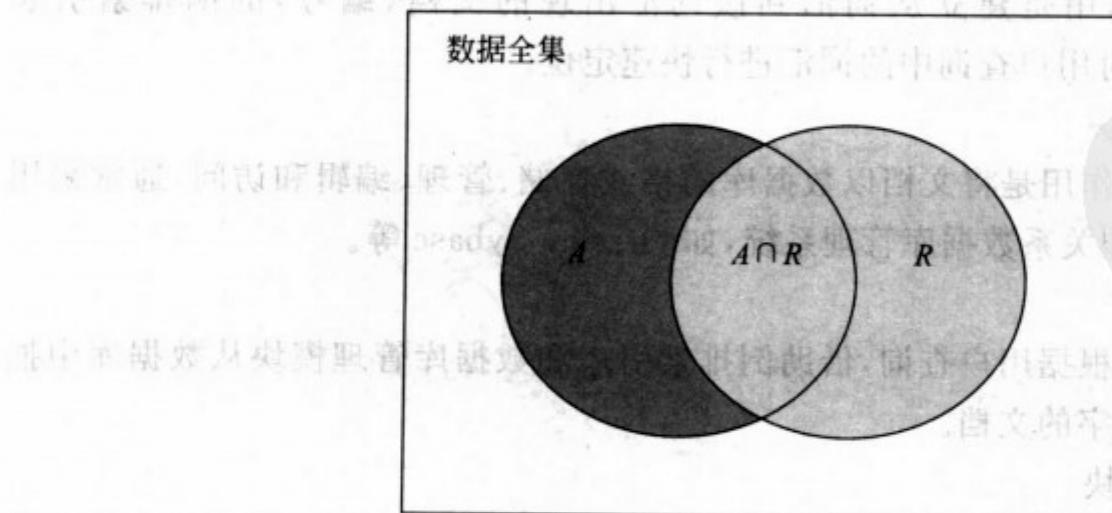


图 8-2 精确度和召回率示意图

集中所有与用户查询相关的数据记录的集合,于是有

$$\text{Precision} = \frac{|A \cap R|}{|A|} \quad (8-1)$$

$$\text{Recall} = \frac{|A \cap R|}{|R|} \quad (8-2)$$

精确度反映了系统能够返回与用户查询相关数据记录的能力,召回率反映了系统能够找到全部相关数据记录的能力。信息检索系统的精确度较召回率更易于估计,只需专家向检索系统输入查询信息,统计系统返回数据记录的个数和其中与查询相关的数据记录的个数,即可得出。对召回率的估计,只有小规模的数据全集可以由专家统计的方法计算,对于大规模的数据全集,例如搜索引擎对应的海量网络信息,只能通过与其他信息检索系统相比较的方法来估算。

人们经常使用精确度-召回率曲线(precision-recall curve)来定量分析一个信息检索系统的改进情况或者比较几个信息检索系统的优劣。在这种曲线的坐标系中,横轴表示召回率的取值范围(0%~100%),纵轴表示精确度的取值范围(从0%~100%),曲线上任意一点(r, p)表示系统在精确度为 p 时具有召回率 r ,如图8-3所示。

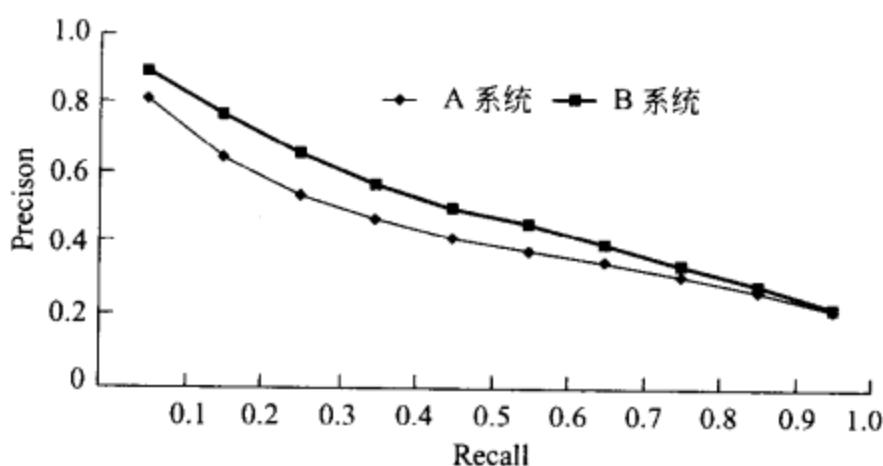


图8-3 精确度-召回率曲线

例如:设有一特定查询 q ,在数据全集中所有与该查询相关的文档为 $R=\{d_3, d_5, d_9, d_{25}, d_{39}, d_{44}, d_{56}, d_{71}, d_{89}, d_{123}\}$,信息检索结果按相关度由大到小顺序是(□标志与 q 相关的文档):

- ① $d_{123} \square$; ② d_{84} ; ③ $d_{56} \square$; ④ d_6 ; ⑤ d_8 ; ⑥ $d_{38} \square$; ⑦ d_{48} ; ⑧ d_{250} ; ⑨ d_{133} ; ⑩ $d_3 \square$; ⑪ d_9 ; ⑫ d_{511} ; ⑬ d_{129} ; ⑭ d_{187} ; ⑮ $d_{25} \square$ 。

可见,当召回率为10%时,精确度为100%;当召回率为20%时,精确度为67%(2/3);当召回率为30%时,精确度为50%。

如果使用 n 个查询测试系统,则可以统计得到平均精确度-召回率曲线。系统对 n 个查询的平均精确度由下式计算,对每个召回率 r ,有

$$\bar{P}(r) = \frac{1}{n} \sum_{i=1}^n P_i(r)$$

在精确度-召回率曲线的坐标系区域中,曲线越接近于左上方,说明系统检索到高相关性文档的能力越强,同时较多地遗漏了一些相关的文档;曲线越接近于右上方,说明系

统的精确度和召回率越高,系统在相关性方面的整体性能更佳;曲线越接近于右下方,说明系统检索到全部相关性文档的能力越强,但是同时也检索到较多无关的文档。例如,在图 8-3 中,B 系统较 A 系统在相关性方面的整体性能更佳。在改进一个信息检索系统时,如果改进后的曲线较改进前的曲线更趋向于右上方,则可以保留这样的改进。

许多用户对信息检索系统精确度要求较高,他们希望尽快查到相关的文档,而不把时间浪费在无关的文档上。另外一些用户则认为召回率更加重要,他们认为相关文档占检索返回的文档比例越高,系统效果越好。Van Rijsbergen 于 1979 年提出了 E (effectiveness)量度,将精确度和召回率结合起来^[4],并赋予不同的权重,成为一个统一的系统有效性的量度:

$$E \approx 1 - \frac{1}{\alpha \left(\frac{1}{P} \right) + (1 - \alpha) \frac{1}{R}} \quad (8-3)$$

式中, P 为系统精确度。 R 为系统召回率。 α 取值在 $0 \sim 1$ 之间, α 取值为 0,表示不考虑精确度,仅考虑召回率; α 取值为 0.5,表示精确度与召回率重要性相同; α 取值为 1,表示仅考虑精确度。对于数据全集规模较大,特别是搜索引擎一类基于国际互联网的信息检索系统来说,精确度往往是较召回率更加重要的系统评价量度。

8.1.4 信息检索简史

信息检索技术起源于人们为方便查询和访问,而将文献资料重新整理与分类的手工劳动。现代信息检索技术则是在计算机技术和通信技术发展的基础上建立起来的。它产生于 20 世纪 50 年代,经过长期缓慢的发展阶段,从 20 世纪 90 年代开始至今,随着国际互联网技术的诞生和发展进入了一个兴盛的时期。

1950 年,美国学者 Calvin N. Mooers 首创了“信息检索”这一术语,1958 年,美国学者 Luhn 提出了统计信息检索的基本理论和方法。1960 年,Marson 和 Kuhns 提出了信息检索的概率模型。1965 年,美国康奈尔大学的 Gerard Salton 教授及其学生创立了信息检索向量空间模型,他们也因此成为信息检索领域的世界顶尖学者。向量空间模型至今仍然是信息检索领域最为有效和广为应用的理论模型,Salton 教授及其学生开发的基于向量空间模型的 SMART 信息检索系统也沿用至今,成为信息检索系统的样板,甚至一些信息检索商业或实验系统的重要组成部分。1966 年,在 Cranfield 项目中,信息检索系统的评价方法被首次提出。1968 年,Rocchio 和 Salton 共同提出了查询扩展的方法。1972 年,Lockheed 公司推出了 DIALOG 系统,成为世界首例商用在线信息查询服务系统。在这一阶段,学者们的卓越工作为信息检索技术奠定了坚实的理论基础。进入 20 世纪 80 年代以后,信息检索技术的发展进入了一个相对沉寂的时期。这一时期的一些典型的理论成果包括模糊集、模糊推理、线性回归技术、通用向量空间模型等。

从 20 世纪 90 年代开始,随着各种廉价的数据存储设备陆续研发成功,信息存储技术有了革命性的进展。特别是 90 年代初期国际互联网技术的诞生和随之而来的网络信息的爆炸式增长,更使信息检索技术进入了一个崭新的发展时期。在这一时期,出现了代表性的理论成果,包括潜在语义索引技术、贝叶斯网络和神经网技术等。基于国际互联网的

大型搜索引擎在这一时期大量涌现出来,如 Google,infoseek,Lycos 等。

当前,计算机技术的不断进步和网络信息量爆炸性增长,对信息检索技术提出了越来越高的要求。信息检索技术正在深度和广度上得到迅猛发展:在深度上,信息检索技术使信息自动管理和自动加工,如自动抽词、自动索引、自动检索、自动文摘、自动分类、自动翻译等的能力得到提高;在广度方面,信息检索正由文本信息检索向全文文本、多媒体、跨语言等新型信息检索发展。另外,信息存储技术正在向着信息资源的网络化和分布化方向发展。在信息检索技术研究领域中,基于概念、超文本信息和多媒体信息检索技术的研究最为活跃,并已取得了突破性发展。国际互联网技术的发展给信息检索技术提供了空前广阔的舞台,而信息检索技术的发展为人们利用信息提供了更加方便快捷的手段。

8.2 信息检索的统计模型

本节将介绍现有的主要信息检索技术与相关研究成果。这些应用于信息检索领域的技术与方法可以粗略地划分为两大类:基于统计的方法和基于语义的方法。基于统计的方法主要根据用户查询与数据全集中数据的统计量度计算相关性。基于语义的方法对用户查询和数据全集中的数据进行一定程度的语法语义分析,换句话说,这类方法是在对用户查询和数据全集内容理解的基础上进行两者的相关性计算。从信息检索技术的发展来看,基于统计的方法长期占据着主流地位,因而也是本节介绍的重点。针对基于统计的方法的不足之处,人们将语言处理技术(如对文档和用户查询进行浅层次的语法语义分析)引入到信息检索领域,将其作为基于统计的方法的补充,以进一步提高信息检索系统的性能。

信息检索模型是对实际信息检索过程加以抽象而构造的信息检索的数学模型,是关于信息检索的各个主要处理阶段的形式化框架,其主要内容包括用数学工具表示的数据全集中的文档、用户信息需求、用户需求与文档的匹配、检索结果文档的相关性计算等。信息检索模型适用的数学工具包括集合、布尔代数、线性代数、概率、模糊集理论、神经网络、贝叶斯网络等。从信息检索模型推知的结论依赖于该模型对实际检索过程模拟的近似程度,一个信息检索模型与其具体实现有所不同。

一个信息检索模型 IRM 是一个三元组 $IRM=(D, Q, R)$, 其中 D 是文档的集合; Q 是用户需求的集合; $R: D \times Q \rightarrow \mathbf{R}$ 是集合 D 与 Q 的笛卡尔乘积到实数集 \mathbf{R} 的一个映射, 对每个用户查询 $q \in Q$, 每个文档 $d \in D$, 映射 R 将 (d, q) 映射为一个实数, 称为用户查询 q 与文档 d 的相关度。因此,一个信息检索模型必须确定文档的表示方式, 用户查询的表示方式以及用户查询与文档之间相关度计算的方法。

8.2.1 基于统计的信息检索模型

基于统计的信息检索模型包括布尔模型、扩展布尔模型、向量空间模型、概率模型等。在这类模型中, 文档被表示为关键词(keyword)的集合; 这一表示方式又称为文档的平面结构(flat structure); 关键词又称为索引词(index term), 指除停用词之外的代表文档内容的词汇, 大多数是名词。例如, 如果停用词中包括 {all, and, could, had, he, of, on,

than, that, the, to, with, their}, 则文档

Rumsfeld said today that he had ordered the military's regional commanders to rewrite all of their war plans to capitalize on precision weapons, better intelligence and speedier deployment

被表示为：

Rumsfeld, better, capitalize, commanders, deployment, intelligence, military's, ordered, plans, precision, regional, rewrite, said, speedier, their, today, war, weapons

当然,不同的词汇在表示文档内容时起的作用不同。一个词汇在表示文档内容时所起的作用的大小,用该词汇的重要性来表示,称为一个词汇的权重(weight)。令 N 表示整个信息检索系统中关键词的总数;令 k_i 表示一个关键词;令 d_j 表示一个文档;对每个 (k_i, d_j) ,令 $w_{i,j} \geq 0$ 表示关键词 k_i 在表示文档 d_j 时的权重,若 $w_{i,j} = 0$,表示关键词 k_i 不在文档 d_j 中出现,则文档 d_j 可以由一个 N 维向量,表示为 $\mathbf{d}_j = (w_{1,j}, w_{2,j}, \dots, w_{N,j})$ 。

统计信息检索模型通常假设出现在文档中的词汇彼此独立。另外,它还假设词汇在文档中的出现没有二义性,即一个词的词义由其词形惟一地确定。这些假设当然是不正确的,但它们大大简化了处理过程,提高了信息检索系统的实时性,且对系统的精度影响不大。

西文的信息检索系统都包含一个词干抽取的模块,以对词形进行归一化处理,由于中文没有词形变化,中文的信息检索系统没有这一模块。但是对于中文信息检索系统来说,由于中文的词与词之间没有空格,而任何的系统词表只能包括有限的词,因此经常会由于分词的错误而导致文档表示的内容不准确,从而使信息检索的结果也不精确。

8.2.2 布尔模型

在布尔模型中,文档中索引词的权重只有 0 和 1 两种取值,分别表示文档中包含该索引词和不包含该索引词。用户查询是由标准逻辑操作符 AND, OR, NOT 等将索引词连接起来构成布尔表达式。

用户查询与文档的相关度的计算方法是:对用户查询中的每个索引词 k_i ,构造一个文档集合 D_{k_i} ,使得该集合内的每一个文档都包含索引词 k_i : $D_{k_i} = \{d_j | w_{i,j} = 1\}$,将用户查询中布尔表达式中的操作符 AND, OR, NOT 替换为集合运算符 \cap , \cup , \neg ,于是用户查询中的布尔表达式转换为集合之间的操作。信息检索的返回结果是一个集合,在集合中的文档是相关文档,否则是无关文档。

例如:设关键词为 k_1, k_2, \dots, k_8 ,数据全集中的文档为 D_1, D_2, \dots, D_6 ,其中,

$D_1 = \{k_1, k_2, k_3, k_4, k_5\}$; $D_2 = \{k_1, k_2, k_3, k_4\}$; $D_3 = \{k_2, k_4, k_6, k_8\}$; $D_4 = \{k_1, k_3, k_5, k_7\}$; $D_5 = \{k_4, k_5, k_6, k_7, k_8\}$; $D_6 = \{k_1, k_2, k_3, k_4\}$ 。若用户查询为

$$k_1 \text{ AND}(k_2 \text{ OR } \text{NOT}(k_3))$$

则查询结果为

$$\{D_1, D_2, D_4, D_6\} \cap (\{D_1, D_2, D_3, D_6\} \cup \{D_3, D_5\}) = \{D_1, D_2, D_6\}$$

布尔模型的最大优点是机制简单,检索效率很高,因此在早期的商用信息检索系统中得到了普遍的应用。但是由于它的分类能力有限,仅能够将文档划分为相关和不相关两大类,而不能给出相关性大小的数值,因此经常会出现高度相关的文档排序靠后的现象。

8.2.3 向量空间模型

在向量空间模型中,令 N 表示整个信息检索系统中的关键词的总数,文档表示为由索引词的权重构成的 N 维向量: $\mathbf{d}_j = (w_{1,j}, w_{2,j}, \dots, w_{N,j})$, $w_{i,j}$ 为索引词 i 在文档 \mathbf{d}_j 中的权重,如果 $w_{i,j}$ 为 0,则表示该词在 \mathbf{d}_j 中不出现。用户查询也可表示为由用户查询中索引词的权重构成的 N 维向量: $\mathbf{q} = (w_{1,q}, w_{2,q}, \dots, w_{N,q})$, $w_{i,q}$ 为用户给定的索引词 i 的权重。

用户查询与文档相关度的计算方法是:对于用户查询 \mathbf{q} 和数据全集中的每个文档 \mathbf{d}_j ,计算用户查询向量与文档向量的相似度 $\text{Similarity}(\mathbf{q}, \mathbf{d}_j)$, Similarity 值越高,表示用户查询与文档的相关度越大。选择相似度大于某特定阈值的所有文档,或者具有较高相似度的前 n 个文档作为检索结果输出。

因此,向量空间模型的两个关键问题是索引词权重的计算以及用户查询与文档的相似度计算。

1. 权重的确定

索引词权重的计算有以下几种方法。

(1) 词频与倒文档频度方法

在索引词权重的计算中最为成功的和广为应用的方法称为“词频与倒文档频度”(term frequency * inverse document frequency, TF * IDF)方法。该方法将一个索引词在单个文档中的重要性和在整个数据全集中的重要性结合起来,成为一个统一的量度。

一个词在文档中出现的频度是该词重要性的标志之一。如果一个文档中索引词 A 出现的频度大于索引词 B 出现的频度,那么索引词 A 对于表达该文档内容的重要性应该比 B 大。因此,一个简单的确定一个词权重的方法是用该词在文档中的频度来标志其权重,令 $\text{freq}_{i,j}$ 表示索引词 k_i 在文档 d_j 中的频度, $w_{i,j} = \text{TF}_{i,j} = \text{freq}_{i,j}$, 显然,这一量度反映了一个词在特定文档中的局部统计特征。

然而,这一量度仍然不够完善,主要表现在词汇在某文档中的频度不能够完整地表征该词的重要性。从整个数据全集的角度来看,一个词如果出现的频度很高,那么该词对于表示某特定文档的内容以及对于用户查询区分相关文档和不相关文档的重要性就不会很大,例如,在关于计算机的数据全集中,“计算机”一词的频度很高,但它对表达文档内容的重要性应该比“硬盘”、“显示器”等词汇要小;反之,如果一个词在整个数据全集中出现的频度很低,则它应该是反映包含该词文档内容的重要词汇。因此,一个索引词的权重还应与该词所在文档的总数成反比或者近似反比的关系,这一分量反映了包含该索引词的文档区别于其他文档的程度,是一个索引词在整个数据全局中重要性的全局性统计特征,称为倒文档频度(Inverse Document Frequency, IDF),通常由下式计算:

$$\text{IDF}_i = \log\left(\frac{n}{n_i}\right) \quad (8-4)$$

其中, n 为数据全集中文档的总数, n_i 为包含索引词 i 的文档总数。函数 \log 的底可以取 10、自然对数 e 或者 2。

综上所述,一个索引词 k_i 在文档 d_j 中的权重 $w_{i,j}$ 由下式计算:

$$w_{i,j} = \text{TF}_{i,j} \cdot \text{IDF}_i \quad (8-5)$$

由公式(8-5)计算得出的词汇权重的值即为该词的 $\text{TF} \cdot \text{DIF}$ 值, 它说明一个在单个文档中频度很高, 而在整个数据全集中频度很低的词是更加重要的词。

例如: 一个数据全集共有文档 10000 个, 某索引词 A 在某一个文档中出现了 20 次, 在数据全集中, A 总共出现了 2000 次, 那么索引词 A 的 $\text{TF} \cdot \text{IDF}$ 值为 $20 \times \lg \frac{10000}{2000} = 13.98$ 。

但这一量度仍有不足之处, 主要表现在它没有考虑文档中索引词的总数, 例如, 一个在 100 个词构成的文档中出现 10 次的词, 应该较 1000 个词构成的文档中出现 20 次的词更加“重要”。为反映文档中索引词总数对权重的影响, 以 $\text{TF} \cdot \text{IDF}$ 量度为基础, 许多学者对这一量度提出了“正规化”(normalization)^[5] 的改进方法。

(2) 最大正规化法

针对 TF 的改进主要是将词频进行正规化处理, 将它映射为一个在区间 $[0, 1]$ 中的量。改进方法之一是将词频除以某个与包含该词文档的索引词总数相关的因子, 如文档中词的总数或者文档中具有最大频度的词的频度等, 即

$$\text{TF}_{i,j} = \frac{\text{freq}_{i,j}}{\max_k \{\text{freq}_{k,j}\}} \quad (8-6)$$

这类改进方法称为“最大正规化”(maximum normalization)法。

公式(8-6)的一个变体如公式(8-7)所示, 该公式将 TF 映射为一个在区间 $[0.5, 1]$ 上的值, 即

$$\text{TF}_{i,j} = \left(0.5 + 0.5 \times \frac{\text{freq}_{i,j}}{\max_k \{\text{freq}_{k,j}\}} \right) \quad (8-7)$$

公式(8-7)称为增量正规化词频(augmented normalized term frequency)。

使用文档中的最大词频作为分母的正规化方法有时效果并不理想, 特别是当文档中某词的词频很大, 而其他词词频相对较小的时候, 使用前述的方法会出现多数词汇 TF 值较小, 并且彼此差别不大的结果。

(3) 对数词频法

另一个经常使用的 TF 正规化方法称为“对数词频”(logarithmic term frequency)法, 该方法使用如下公式计算 TF 值:

$$\text{TF}_{i,j} = \log(\text{freq}_{i,j}) + 1 \quad (8-8)$$

对数词频法不使用文本长度或者最大词频这些正规化因子, 而是通过对数函数降低了词频对 TF 取值的影响, 从而减少了文档中少数高频词对权重计算的影响, 降低了低频词权重的取值, 而且减轻了文档长度的变化对这一取值的变化影响。

(4) 余弦正规化法

另一类正规化方法是通过整个文档向量的长度来实现。当一个文档向量构造完成后, 该向量的每一维都设定了对应索引词的 $\text{TF} \cdot \text{IDF}$ 值, 将这个向量的所有维上的这些值都除以该文档向量的欧氏长度, 即得到经过正规化的文档向量。一个向量的欧氏长度

是该向量上所有分量平方和的平方根。由于经过正规化后的向量具有单位长度,而且在每一维上的值恰好是该向量及其在这一维上相应坐标轴上投影的夹角的余弦值,因此,这种正规化方法又称为余弦正规化法(cosine normalization)。余弦正规化法解决了文档中少数高频词对其他词权值扰动过大这一问题。

2. 相似度计算

当用户查询和数据全集中的文档按照前面讨论过的方法向量化后,下一个步骤就是计算它们的相似度(similarity)。相似度是用户查询与文档相关性的量度。令 d 表示文档向量, q 表示用户查询向量, $\text{similarity}(d, q)$ 取值通常需满足如下条件:

非负性: $\text{similarity}(d, q) \geq 0$;

对称性: $\text{similarity}(d, q) = \text{similarity}(q, d)$;

若 $\text{similarity}(d, q) = 0$, 则表示文档 d 与用户查询 q 完全不相关;

$\text{similarity}(d, q)$ 值越大, 表示文档 d 与用户查询 q 相关性越大;

$\text{similarity}(d, q)$ 的取值范围通常经正规化处理为区间 $[0, 1]$ 上的一个值;

$\text{similarity}(d, q) = 1$, 当且仅当文档 d 与用户查询 q 完全相同。

给定一个用户查询 N 维向量 $q = (w_{1,q}, w_{2,q}, \dots, w_{N,q})$, 其每一维上的权值为 $w_{i,q}$ ($1 \leq i \leq N$); 文档 N 维向量 $d_j = (w_{1,j}, w_{2,j}, \dots, w_{N,j})$ 每一维上的权值为 $w_{i,j}$ ($1 \leq i \leq N$), 一个简单的相似度计算方法是计算这两个 N 维向量的内积, 即

$$\text{similarity}(q, d) = \sum_{i=1}^N w_{i,q} \cdot w_{i,j} \quad (8-9)$$

特别是当这两个 N 维向量都经过了余弦正规化法处理以后, 它们的内积恰好是两个向量夹角的余弦, 因此把这时的相似度量度称做余弦相似度(cosine similarity), 如公式(8-10)所示。

$$\text{similarity}(d_j, q) = \frac{\sum_{i=1}^N w_{i,j} \cdot w_{i,q}}{\sqrt{\sum_{i=1}^N (w_{i,j})^2} \cdot \sqrt{\sum_{i=1}^N (w_{i,q})^2}} \quad (8-10)$$

根据余弦函数的性质, 余弦相似度的取值范围恰好在 $[0, 1]$ 区间, 且满足相似度函数取值需满足的条件, 因此公式(8-10)成为向量空间模型最为常用的相似度计算方法。余弦相似度的主要缺点是: 对于一个用户查询, 包含索引词个数较多的长文档往往计算结果偏低。

在计算用户查询向量和文档向量的相似度时采用的计算方法, 除了内积及其正规化形式——余弦相似度之外, 还包括其他一些距离函数, 例如, 一组基于距离的量度如公式(8-11)所示:

$$L_p(D_1, D_2) = \left[\sum_i |D_{1i} - D_{2i}|^p \right]^{1/p} \quad (8-11)$$

式中, D_1 和 D_2 是向量, D_{1i} 和 D_{2i} 分别是 D_1 和 D_2 第 i 维的取值, 参数 p 是正整常数, 当 $p = 1$ 时, 这一量度称为街区距离(city block distance); 当 $p = 2$ 时, 这一量度称为欧氏距离(Euclidean distance); 当 $p = \infty$ 时, 这一量度称为最大方向距离(maximal direction distance)等。

另外,还有一些量度完全不考虑向量及其各维的取值情况,例如掷骰子系数(dice's coefficient)、Jaccard 系数(Jaccard's coefficient)等。

与布尔模型相比较,向量空间模型在实际系统中的表现更优,而且计算效率较高。因此,向量空间模型在信息检索系统中获得了广泛的应用。

8.2.4 概率模型

在概率模型中,文档和用户查询也被表示为索引词集合的形式,与前面讨论过的信息检索模型不同的是,概率模型通常采用索引词在文档中的统计分布等参量,计算任意文档 d 与给定用户查询 q 相关的概率 $P(q|d)$ 。

本节重点介绍一类典型的信息检索概率模型:贝叶斯推理网络模型,该推理网络模型提供了一种将不同来源的证据结合起来,以确定给定文档满足用户查询或者信息需求概率要求的自然的方法。

贝叶斯网络是一个描述随机变量之间因果关系的有向无环图。在一个贝叶斯网络中,节点表示随机变量,一条从父节点 Y 到子节点 X 的边表示两个随机变量的依赖关系,在贝叶斯网络中的任一节点 X 都附加了一系列条件概率 $P(X|Y_1, \dots, Y_n)$,该条件概率表示该节点与其父节点 Y_1, \dots, Y_n 依赖关系的强度,在贝叶斯网络中,一个节点仅条件依赖于它的父节点。

在应用于信息检索的贝叶斯网络中,节点是每个索引词 k_i ,数据全集中的文档 d_j 和用户查询 q 。边有两种类型:一类是由文档 d_j 指向索引词 k_i ,表示索引词 k_i 出现在文档 d_j 中;另一类是从索引词 k_i 指向用户查询 q ,表示索引词 k_i 出现在用户查询 q 中。整个贝叶斯网络也由此划分为 3 个基本层次,即文档层、索引词层和用户查询层,于是,计算用户查询与文档的相关度这一问题转化为由贝叶斯网络计算用户查询与文档的联合概率的问题,如图 8-4 所示。

可见:

$$\begin{aligned} P(q, d_j) &= \sum_{k_1, \dots, k_N} P((q, d_j) | k_1, \dots, k_N) \cdot P(k_1, \dots, k_N) = \sum_{k_1, \dots, k_N} P(q, d_j, k_1, \dots, k_N) \\ &= \sum_{k_1, \dots, k_N} P(q | (d_j, k_1, \dots, k_N)) \cdot P(d_j, k_1, \dots, k_N) \\ &= \sum_{k_1, \dots, k_N} P(q | (k_1, \dots, k_N)) \cdot P(k_1, \dots, k_N | d_j) \cdot P(d_j) \end{aligned}$$

又由索引词的独立假设

$$P(k_1, \dots, k_N | d_j) = \prod_{i|k_i=1} P(k_i | d_j) \cdot \prod_{i|k_i=0} (1 - P(k_i | d_j))$$

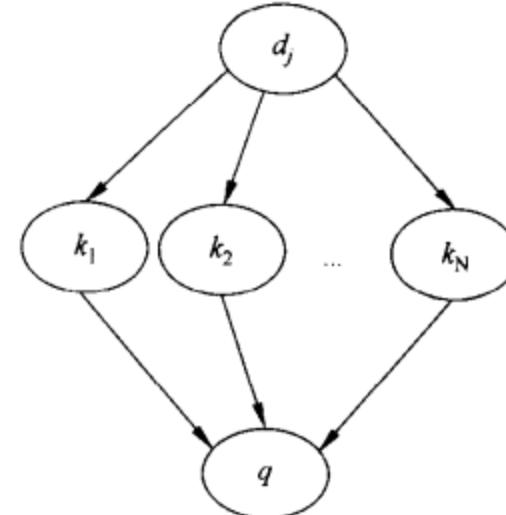


图 8-4 信息检索中的贝叶斯网络模型

于是有

$$P(q, d_j) = P(d_j) \cdot \sum_{k_1, \dots, k_N} P(q | k_1, \dots, k_N) \cdot \prod_{i | k_i=1} P(k_i | d_j) \cdot \prod_{i | k_i=0} (1 - P(k_i | d_j)) \quad (8-12)$$

因此,为计算 $P(q, d_j)$,必须已知先验概率 $P(d_j)$ 、条件概率 $P(k_i | d_j)$ 和后验概率 $P(q | k_1, \dots, k_N)$ 。其中,先验概率 $P(d_j)$ 为文档的概率, $P(k_i | d_j)$ 是索引词 k_i 与文档 d_j 相关的概率, $P(q | k_1, \dots, k_N)$ 是索引词 k_1, \dots, k_N 与用户查询相关的概率。

剩下的问题是如何计算这些概率。对于 $P(d_j)$,可有如下一些计算方法:

(1) 均匀分布法

$$P(d_j) = \frac{1}{n} \quad (8-13)$$

其中, n 为数据全集中文档的总数。

(2) 正规化法

$$P(d_j) = \frac{1}{|d_j|} \quad (8-14)$$

其中, $|d_j|$ 为文档向量 d_j 的长度。

对于 $P(k_i | d_j)$,有如下一些计算方法:

(1) 二值法

$$P(k_i | d_j) = \begin{cases} 1, & \text{如果索引词 } k_i \text{ 在文档 } d_j \text{ 中出现} \\ 0, & \text{否则} \end{cases} \quad (8-15)$$

(2) 权重法

$$P(k_i | d_j) = \frac{\text{freq}_{i,j}}{\max_k \{\text{freq}_{k,j}\}} \quad (8-16)$$

对于 $P(q | k_1, \dots, k_N)$,有如下一些计算方法:

(1) 二值法

$$P(q | k_1, \dots, k_N) = \begin{cases} 1, & \text{如果索引词 } k_1, \dots, k_N \text{ 在且仅在查询 } q \text{ 中出现} \\ 0, & \text{否则} \end{cases} \quad (8-17)$$

(2) 权重法

$$P(q | k_1, \dots, k_N) = \begin{cases} \text{某个基于 IDF 的函数} \\ 0 \end{cases} \quad (8-18)$$

8.3 信息检索中的自然语言处理方法

1. 自然语言处理方法

所谓信息检索中的自然语言处理方法,是指通过对文档中自然语言文本所做的语法语义分析,来提高信息检索精确度或者召回率的方法的统称。与基于统计的信息检索方法相比较,自然语言处理方法以对文档文本的语言结构分析和语义分析为特色,将信息处理的层次深入到了文档中文本的内容,而非仅依据文本中索引词的统计信息。由于信息

检索对实时性有较高的要求,不可能对网络文档进行全面完备的语法语义分析,因此这些自然语言处理方法通常只能停留在“浅层”(shallow)处理的层次上,例如,对文档中的名词短语进行自动识别和分类等。另外,由于自然语言处理方法中也大量运用了统计分析技术,因此在信息检索中应用的自然语言处理方法与统计方法并没有明显的界限,可以将自然语言处理方法视为前面讨论过的基于统计的信息检索技术的有益补充。

自然语言处理技术按照语言处理对象的不同语言单位可以划分为音韵(phonological)、词形(morphological)、词法(lexical)、语法(syntactic)、语义(semantic)、语篇(discourse)、语用(pragmatic)等不同的技术层次。许多传统的信息检索技术都采用了自然语言处理技术中的词形分析和词法分析方法。词形分析是指对西文中给定词汇的不同形式进行分析，并确定不同的词素成分，即前缀、词根和后缀。在信息检索系统的词干抽取模块用词形分析技术将用户查询和文档中的词汇还原成统一的形式。词法分析方法是指在词汇这一语言单位上的结构与词义分析方法，在传统的信息检索中应用也很普遍，例如，停用词表中词汇的选择等，都需要采用词法分析方法。近年来，信息检索技术越来越多地采用更高级别的语言处理技术，以进一步提高信息检索系统的表现，如专有名词的自动识别与分类、文档文本的自动词性标注等。

人们利用信息检索系统查询信息时,存在着这样的现象:同一语义可以有多种不同的表达方式,而不同用户使用相同的词汇进行查询的可能性微乎其微。为解决这一实际问题,信息检索的研究者们引入了“语义相关”的概念,计算词汇间的语义相似度(semantic similarity),并据此对用户的查询词汇进行语义相似词汇的扩展,或者改进用户查询与文档相似度的计算方法,本节重点介绍在这一研究方向上的最新进展。

在语言处理技术中,语义分析占有非常重要的地位。传统的语法分析技术仅仅着眼于语言的结构分析,而理解语言关键在于理解语言表达的意义。对于汉语这种缺乏语法形态的语言,语义知识就更加重要了。确定汉语词汇的语义,不仅在于标明一个词的语义属性,更为重要的是,需要确定该词汇与其他词汇的关系,正如董振东先生所言:“关系是词汇语义的灵魂。”在知网^[1]中,董先生将词汇之间的关系归纳为上下位关系、部分-整体关系、同义关系、反义关系、对义关系等十余种,词汇之间构成如图 8-5 所示的概念(词汇)语义网络。

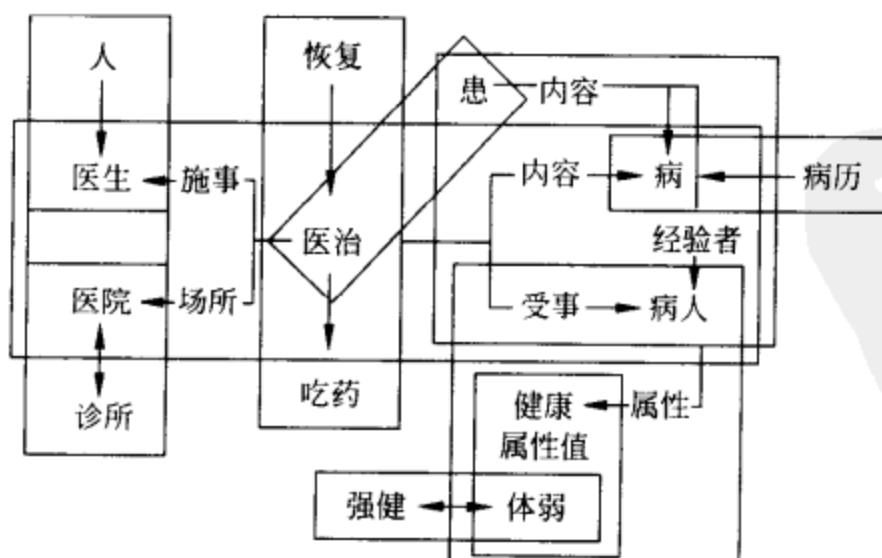


图 8-5 概念语义网络

语义相似度是将词汇间的种种不同的直接或间接语义关系映射为一个表示词汇间语义相关的紧密程度的数值。

词汇间语义相关度的计算方法大体上可以分成两类：一类是基于按照概念间结构层次关系组织的语义辞典的方法，这种方法建立这样一个假设的基础上，即当且仅当两个词汇在概念的结构层次网络图中存在一条通路（主要是上下位关系）时，它们具有一定的语义相关性。这主要是根据这类语言学资源中概念间的上下位关系和同义关系来计算的，如 Rada 等^[2] 和 Lee 等^[3] 通过计算在 Wordnet^[4] 中由词节点之间上下位关系构成的最短路径来计算词之间的语义相关度。

Resnik^[5] 根据在 Wordnet 中上下位关系网络中两个词的公共祖先节点的最大信息量来衡量两个词的语义相关度。在 Wordnet 中，一个概念 c 是一个由同义词构成的集合，概念 c 的概率由公式(8-19)给出：

$$p(c_p) = \frac{\text{freq}(c_p)}{N_p} \quad (8-19)$$

式中， N_p 是在该语料库中具有词性 p 的词汇总数， $\text{freq}(c_p)$ 是具有词性 p 的概念 c 在大规模语料库中出现的个数，它的计算方法如下：

$$\text{freq}(c_p) = \sum_{n \in \text{word}(c_p)} \text{count}(n) \quad (8-20)$$

公式(8-20)中， $\text{word}(c_p)$ 是在 Wordnet 中所有具有词性 p 的概念 c 包含的词汇，包括概念 c 本身以及概念 c 通过上下位关系相连接的所有后代概念节点。于是，任意两个词汇间的语义相似度由公式(8-21)给出：

$$\text{similarity}(w_{1p}, w_{2p}) = \max_{c_p \in \{x | \text{subsume}(x, w_{1p}) \wedge \text{subsume}(x, w_{2p})\}} [-\log p(c_p)] \quad (8-21)$$

式中， $\text{subsume}(x, w)$ 为词汇 w 的所有通过上下位关系相连接的祖先概念节点。

2. 基于统计的方法

基于统计的方法，是将词汇的上下文信息的概率分布作为词汇间语义相关度计算的参照，这类方法建立在两个词汇具有某种程度的语义相关当且仅当它们出现在相同的上下文中这一假设的基础上。如 Brown^[6] 的基于平均互信息的方法，Lillian Lee^[7] 的基于相关熵的方法等等。

3. 基于语义辞典的方法

基于语义辞典的方法通常依赖于比较完备的按照概念间结构层次关系组织的大型语义词典，如 Wordnet 等。20世纪 80 年代以来，我国也开展了汉语语义词典的研究与开发。如：董振东、董强等开发的《关系语义知识库(KBRS)》、清华大学黄昌宁等 1988 年进行的语义词典研究、“905”项目《信息处理用汉语语义词典》等。由于构造汉语语义词典是一件规模浩大的系统工程，目前国际上还没有像 Wordnet 那样严格按照概念结构层次关系组织得比较完备的汉语语义词典。另一方面，国内外对汉语语义分类体系的研究已有一些成果，如：梅家驹的《同义词词林》^[8]、林杏光的《简明汉语义类词典》、陈群秀与张普的《信息处理用现代汉语语义分类体系》等。但这些分类体系都是用将词汇纳入逐步细

化的语义类中的方法组织的,因此必须引入基于统计的定量分析方法,才能够对词汇间的语义相关性进行比较精确的度量。哈尔滨工业大学的关毅以中文语义词典 Hownet 为基础,采用基于向量的方法计算了汉语词汇间的语义相似度,并应用于具有自然语言接口的问一问搜索引擎中,大大提高了信息检索系统的精确度。

8.4 文本自动分类技术

文本自动分类技术(text automatic classification)的基本任务就是对一篇文档,根据其内容,从预先定义好的标记集中,找出一个或者多个最适合于该文档的标记。文本自动分类技术从开始出现到现在,经历了从基于规则到基于统计分类,再到规则和统计相结合的一个过程。在本节中,通过对文本分类问题的提出,对解决问题的主要步骤、主要方法、结果评测方法、文本分类的应用等内容的介绍,让读者对文本自动分类技术有一个总体的认识。最后给出在没有预先定义类别集合的情况下,对文本进行分类的自动聚类技术(automatic clustering)的简要介绍。

8.4.1 问题的提出

文本分类的问题可以这样描述:给一个决策矩阵中的每个元素赋值,并且在 $\{0,1\}$ 的实数范围内取值,如表 8-1 所示。

表 8-1 文本分类决策矩阵

	d_1	...	d_j	...	d_n
c_1	a_{11}	...	a_{1j}	...	a_{1n}
\vdots	\vdots		\vdots		\vdots
c_i	a_{i1}	...	a_{ij}	...	a_{in}
\vdots	\vdots		\vdots		\vdots
c_m	a_{m1}	...	a_{mj}	...	a_{mn}

表 8-1, $C=\{c_1, \dots, c_m\}$ 是预先定义好的类别集合或者标记集合; $D=\{d_1, \dots, d_n\}$ 是待分类文档集合; $a_{i,j}$ 是一个文档和类别标记的隶属度, $a_{i,j} \in [0,1]$ 。注意以下几点:

- ① 类别标记是一组符号。例如,大多数搜索引擎使用的类别体系,包括科学技术、社会文化、政治军事、医疗健康、体育健身等。
- ② 文档对类别的隶属度应该是基于文档的内容,而不是基于描述文档的元数据(metadata)(例如文档出版日期、文档类型等)。
- ③ $a_{i,j}$ 为一条件概率 $a_{i,j}=P(c_i|d_j)$, 如果 $a_{i,j}=1$, 表示第 j 个文档完全属于第 i 个类别(或者说完全相关); $a_{i,j}=0$, 表示文档 j 和类别 i 完全无关。

8.4.2 分类预处理

文本自动分类由以下几个主要步骤完成^[9],如图 8-6 所示。下面对这些步骤分别加以阐述:

(1) 文本预处理

文本预处理过程视具体的文档来源不同而有所不同,而且中文和英文的处理方法也有所不同。对中文文本而言,最常用的文本源就是 Internet 上的 HTML 页面。通常的预处理方法包括:

- 去除 HTML 的标记以及乱码;
- 汉语分词;
- 过滤文本中的停用词。

对英文文本而言,预处理过程不包括分词技术,但是要进行英文单词的词根化处理。

(2) 建立文本索引

现在的文本自动分类技术都是基于向量空间模型的,即把一篇文档表示为由词串组成的向量^[10]。对于文档集合,可以表示为坐标词和文档组成的矩阵 A (注意:在具体的实现中,可以对不同的类别,采用不同的坐标词)。 $A=(w_{i,j})$,其中 $w_{i,j}$ 表示词 i 在文档 j 中的权重。通常矩阵 A 是一个稀疏矩阵(如果用系统词典的全部词作为横坐标,则稀疏问题会更为严重)。后面部分会讨论如何解决稀疏问题。确定 $w_{i,j}$ 的大小可参见 8.2 节的讨论。

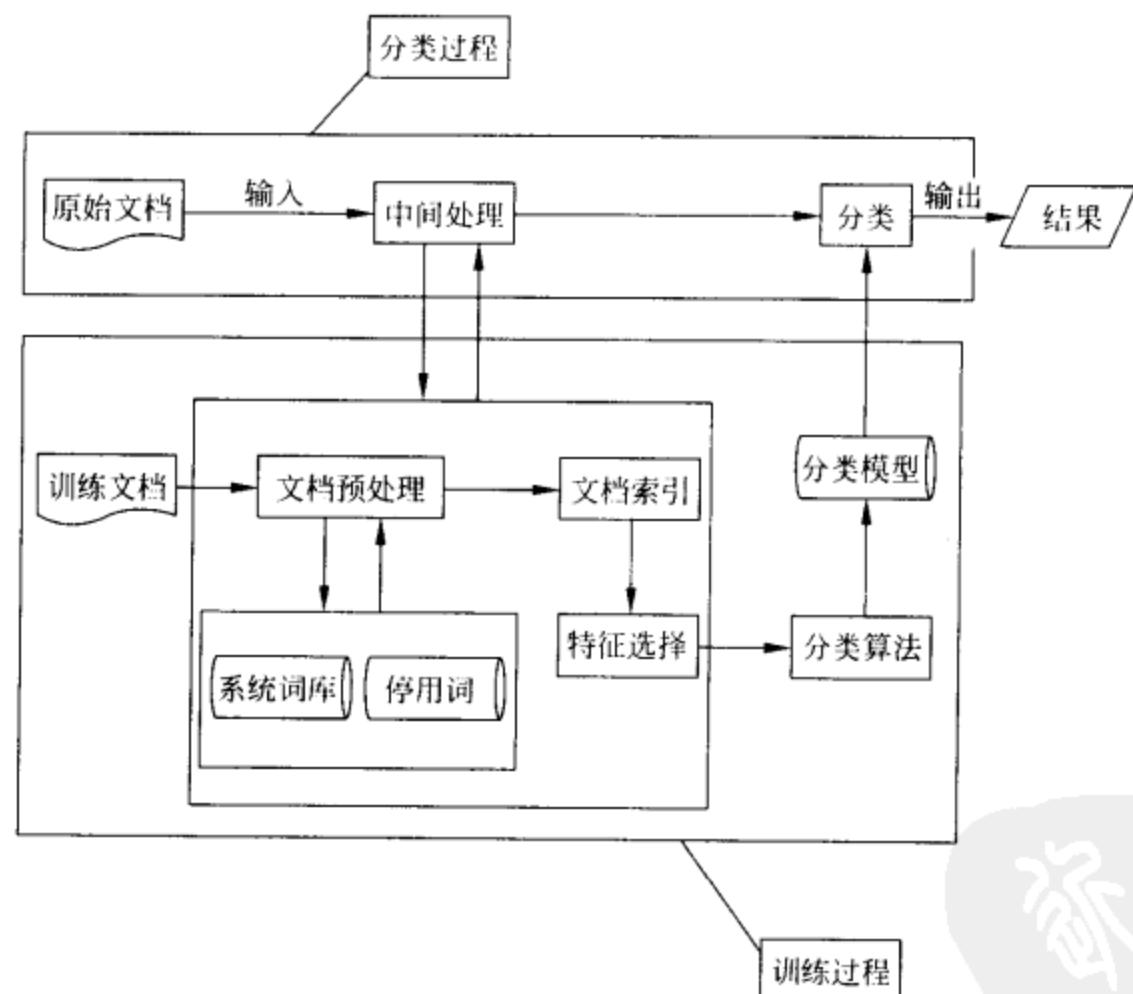


图 8-6 文本分类系统

8.4.3 向量空间简化方法

在向量空间模型中,如果使用所有在训练文档集合中出现的词语作为二维空间的横坐标,那么表示一个文档的向量维数可能会很大(有可能有几千个)。通常的分类方法不

能处理这种特征集,计算量太大,程序运行时间太长;而且没有规模足够大的训练集,用以保证分类结果的有效性。因此,通常都要采用一些方法,对向量空间进行约简。下面详细介绍这几种方法。

(1) 基于文档频率的特征选择

基于文档频率特征选择的方法是一种最简单的特征选择方法,即以在所有训练文档集合中出现的词作为候选特征,从中去除那些包含特征词的文档数小于某个既定阈值的特征。这种方法是在稀有特征对分类决策的信息贡献很小的情况下使用。

(2) 基于信息增益度(information gain)的特征选择

设 c_1, \dots, c_k 为预定义的类别集合,一个词语(特征)的信息增益度定义如下:

$$\begin{aligned} IG(w) = & -\sum_{j=1}^k P(c_j) \log P(c_j) + P(w) \sum_{j=1}^k P(c_j | w) \log P(c_j | w) \\ & + P(\bar{w}) \sum_{j=1}^k P(c_j | \bar{w}) \log P(c_j | \bar{w}) \end{aligned} \quad (8-22)$$

其中, $P(c_j)$ 为第 j 类别出现的概率; $P(c_j | w)$ 为包含特征词 w , 并且属于第 j 类文档的概率; $P(c_j | \bar{w})$ 为不包含特征词 w , 并且属于第 j 类文档的概率。在得到特征词 w 的信息增益度后, 将其与一个阈值进行比较, 如果 w 的信息增益度小于这个阈值, 就从特征集中删除该词。

(3) 基于 χ^2 分布的特征选择

χ^2 量度定义了特征词和类别之间的相关性, 即

$$\chi^2(w, c_i) = \frac{N \cdot (AD - CB)^2}{(A + C) \cdot (B + D) \cdot (A + B) \cdot (C + D)} \quad (8-23)$$

式中, N 代表全体文档; A 代表属于类别 c_i , 并且包含特征词 w 的文档; B 代表包含特征词 w , 但不属于类别 c_i 的文档; C 代表属于类别 c_i , 但不包含特征词 w 的文档; D 代表既不属于类别 c_i , 又不包含特征词 w 的文档。对一个特征词, 它的 χ^2 度量值定义如下:

$$\chi^2(w) = \sum_{i=1}^K P(c_i) \chi^2(w, c_i) \quad (8-24)$$

(4) 基于矩阵分解的特征选择

一篇文档中词语的使用模式有一个潜在结构, 通过使用矩阵运算的方法可以估计这种潜在的结构。潜在语义索引(latent semantic indexing, LSI)技术, 用矩阵的奇异值分解得到一个低维的特征向量^[11]。下面对这种方法给出详细的介绍:

假定一个由词语和文档组成的 $M \times N$ 维的矩阵 \mathbf{A} , 其中 M 为特征词的数目, N 为文档的数目, 则 \mathbf{A} 的奇异值分解定义如下:

$$\mathbf{A} = \mathbf{U} \mathbf{S} \mathbf{V}'$$

其中, \mathbf{U} 是 $M \times R$ 维的矩阵; \mathbf{V} 是 $R \times N$ 维的矩阵; \mathbf{S} 是一个由奇异值组成的 $R \times R$ 维的对角方阵, $R \leq \min(M, N)$ 是矩阵 \mathbf{A} 的秩。把 \mathbf{S} 矩阵对角线上的值按由大到小的顺序排序, 保留前 k 个最大的元素, 剩下的 $R - k$ 个元素置为零。把调整后的特征-文档矩阵记为 \mathbf{A}_k , 定义如下:

$$\mathbf{A}_k = \mathbf{U}_k \mathbf{S}_k \mathbf{V}'_k \quad (8-25)$$

其中, $\mathbf{S}_k (k \times k)$ 是去掉了 Σ 中所有为零的行和列后得到对角阵; $\mathbf{U}_k, \mathbf{V}_k$ 是矩阵 \mathbf{U}, \mathbf{V} 去掉了与 \mathbf{S} 中零元素对应的行和列后得到的矩阵。 \mathbf{A}_k 在一定程度上反映了 \mathbf{A} 的潜在结构。 k 越大, 失真越小, 但开销越大。 k 的选择是按实际问题的要求进行平衡的结果。利用矩阵 $\mathbf{U}_k, \mathbf{S}_k, \mathbf{V}_k$ 可以解决以下问题:

- ① 特征 i 和 j 有多相似。使用“正向”乘法, 即 $\mathbf{A}_k \cdot \mathbf{A}'_k = \mathbf{U}_k \mathbf{S}_k \mathbf{V}'_k \mathbf{V}_k \mathbf{S}_k \mathbf{U}'_k$, 其中 $\mathbf{V}'_k \cdot \mathbf{V}_k = I$, 结果矩阵中的元素 $a_{i,j}$ 表示特征 i 和特征 j 的相似度。
- ② 文档 i 和 j 有多相似。使用“逆向”乘法, $\mathbf{A}'_k \cdot \mathbf{A}_k = \mathbf{V}_k \mathbf{S}_k \mathbf{U}'_k \mathbf{U}_k \mathbf{S}_k \mathbf{V}'_k$, 其中 $\mathbf{U}'_k \cdot \mathbf{U}_k = I$, 结果矩阵中的元素 $a_{i,j}$ 表示文档 i 和文档 j 的相似度。
- ③ 特征 i 和文档 j 有多相关。 \mathbf{A}_k 中的元素正好表示了特征 i 和文档 j 的相关度。

8.4.4 分类方法

现在文本自动分类技术中采用的主要分类算法大都是基于统计和机器学习技术的, 因此这一节中将介绍目前主要的一些统计分类方法。文本分类是一个典型的有指导学习的过程。根据决策结果类别的数目, 可以把分类方法分为两类问题和多类问题。

- (1) 两类分类问题(binary classification)是指待分类对象的目标类别只有两类, 即“是”和“不是”;
- (2) 多类问题(multi-classification), 即类别主题多于 2 的分类问题, 也是实践中使用较多的分类, 它的结果往往是一个按相关度大小排序的类别集合。

两类问题和多类问题可以相互转化。下面介绍的这些方法大多适用于多分类问题, 而且是过去被广泛使用和验证过的。先定义几个变量:

- ① 分类测试集 $\{d_1, \dots, d_M\}$, 所有可能的类别集合 $\{c_1, \dots, c_K\}$;
- ② 分类训练集 $\{d_1, \dots, d_N\}$, 已经被确定的类别集合 $\{y_1, \dots, y_N\}$;
- ③ N_i 是训练集中属于类别 C_i 的文档数目。

1. Rocchio 算法

Rocchio 算法是文档分类的经典方法。其基本思想是为每一个类别 c_i 建立原型向量, 然后根据文档向量和类别原型向量的距离, 确定文档的类别^[12]。类别 i 的原型向量是通过计算属于该类别的所有文档向量的平均值而得到的。这种方法的特点是速度快, 但是精度较低。

2. Naïve Bayes 算法

Naïve Bayes 算法是一种概率方法^[13], 通过对训练数据的学习, 得到在一个文档出现的条件下类别 i 出现的条件概率, 用 Bayes 方法来估计这一概率:

$$P(c_i | d) = \frac{P(c_i) P(d | c_i)}{P(d)} \quad (8-26)$$

式中的分母对所有的类别都是相同的, 因此可以把上式简化为

$$P(c_i | d) = P(c_i) P(d | c_i) \quad (8-27)$$

$\hat{P}(c_i)$ 为 $P(c_i)$ 的估计, 可以用下面的方法得到:

$$\hat{P}(c_i) = \frac{N_i}{N}$$

Naïve Bayes 算法最根本的特点是：假设文档中词语的出现是相互独立的，可以得到下式：

$$P(d | c_i) = P(c_i) \prod_{j=1}^M P(d_j | c_i) \quad (8-28)$$

$\hat{P}(d_j | c_i)$ 为 $P(d_j | c_i)$ 的估计，通过下面的公式得到：

$$\hat{P}(d_j | c_i) = \frac{1 + N_{i,j}}{M + \sum_{k=1}^M N_{j,k}} \quad (8-29)$$

其中， $N_{j,i}$ 是训练集中词语(特征) j 在类别 c_i 中出现的次数。

这种方法其实是一种基于最小错误率的贝叶斯决策理论的分类方法，其他的贝叶斯决策方法，例如基于最小风险的决策方法，可详细参见参考资料 14。

3. 决策树算法

决策树是一个树型的结构，在内部节点上选用一个属性进行分割，每个分叉都是分割的一个部分，叶节点表示一个分布。这种算法的基本思想是用文档向量去匹配一个建立在训练集上的决策树，以决定与类别主题相关还是不相关。其关键就是如何建立决策树。建立决策树包括决策树生成和决策树剪枝两个过程，即开始时，数据都在根节点，然后递归地进行数据分片，生成一个大树，如图 8-7 所示；接下来进行修剪，以去掉噪声或者异常的数据，防止过适应(overfitting)的发生。下面分别讨论这两个过程。

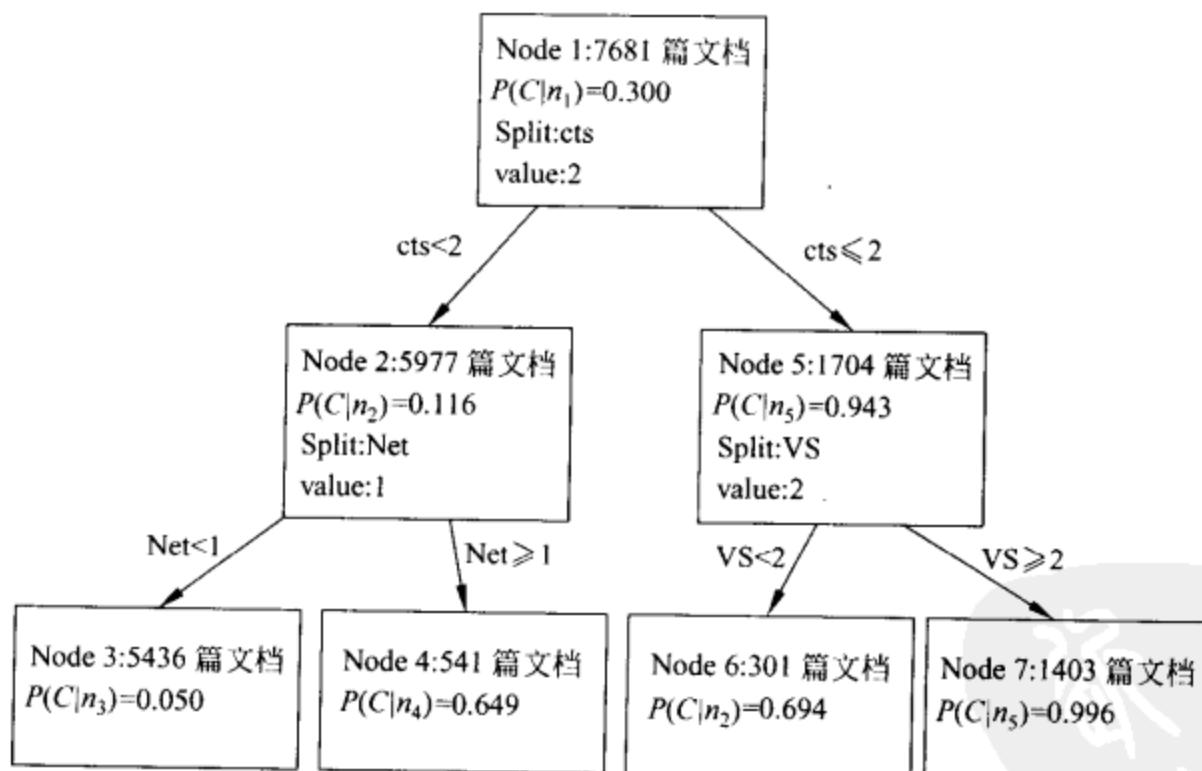


图 8-7 文本分类的决策树表

在生成一棵树的过程中，需要注意的重点是切分点的选择准则和递归循环的停止准则。建立在信息增益度(information gain)之上的 ID3 算法是现在经常使用的一种建树方法，算法描述如下^[15]：

Procedure BuildTree(S)

用数据集 S 初始化根节点 R

用根节点 R 初始化队列 Q

While Q is not Empty do {

 取出队列 Q 中的第一个节点 N

 if N 不纯(Pure) {

 for 每一个属性 A

 估计该节点在 A 上的信息增益

 选出最佳的属性, 将 N 分裂为 N_1, N_2

 }

}

其中信息增益度的定义如下:

$$\text{Gain}(A) = I(s_1, \dots, s_k) - E(A) \quad (8-30)$$

式中, A 为文档向量中的某个属性, 可能有 V 个取值, s_i 为某个节点上属于第 i 类文档的文档数目; $I(s_1, \dots, s_k)$ 为

$$I(s_1, \dots, s_k) = \sum_{j=1}^K P(c_j | t) \log P(c_j | t) \quad (8-31)$$

其中, $P(c_j | t)$ 是节点 t 上训练样本属于第 j 类文档的条件概率, 用下式估算得到:

$$P(c_j | t) = \frac{N_j(t)}{N(t)}$$

$N_j(t)$ 为 t 节点上属于第 j 类文档的数目, $N(t)$ 为 t 节点上文档的数目; $E(A)$ 为

$$E(A) = \sum [(s_{1,i} + \dots + s_{k,j})/S \cdot I(s_{1,i}, \dots, s_{k,j})] \quad (8-32)$$

利用公式(8-30), 对图 8-7 中的部分节点(1, 2, 5)计算其信息增益度, 得到如下结果(图中 Net, VS, cts 等表示被分类文档中出现的词):

① 节点 1 的熵: $P(C|n_1) = 0.300 \quad 0.611$

② 节点 2 的熵: $P(C|n_2) = 0.116 \quad 0.359$

③ 节点 5 的熵: $P(C|n_3) = 0.943 \quad 0.219$

④ 节点 2 和节点 5 的加权和: $\frac{5977}{7681} \times 0.359 + \frac{1704}{7681} \times 0.219 = 0.328$

⑤ 元素 cts 的信息增益度: $0.611 - 0.328 = 0.283$

停止分割数据的条件: 当一个节点上的数据都属于同一个类别, 或者没有属性可以再用于对数据进行分割。至此, 我们建立了一棵决策树。评价决策树的标准通常有以下几种:

(1) 有效性

最为直接的评价决策树在测试集上的性能的方法是将决策树在测试集上进行实际测试, 从而可以选择在测试集中表现最好的一棵决策树。但是这种方法等价在测试集上重新建立决策树, 这在大多数情况下是不现实的。一般采用训练集本身来估计训练算法的有效性。最简单的方法是用训练集的一部分(2/3)来训练决策树, 而用剩余的部分(1/3)检验其有效性。这种方法的缺点是: 减小训练集的空间, 增大过适应的可能性。

(2) 交叉有效性

把训练集 T 分为互不相交且大小相等的 k 个子集。对于任意子集 T_i , 用 $T - T_i$ 训练决策树, 之后用 T_i 进行测试, 得到错误率 e_i , 然后估计整个算法的错误率:

$$e = \frac{1}{k} \sum_{i=1}^k e_i \quad (8-33)$$

(3) 余一有效性

余一有效性方法与交叉有效性相似, 只是将每个 T_i 的大小设定为 1。假设训练集的规模为 n , 则把式(8-33)改为

$$e = \frac{1}{n} \sum_{i=1}^n e_i \quad (8-34)$$

(4) 复杂程度

显然, 决策树构造的过于复杂对用户来说将会是难以理解的, 在很大程度上使得分类树的构造没有意义, 所以应该在保证正确率的前提下尽量构造简单的决策树。

接下来看看如何对决策树进行修剪, 通常有两种剪枝标准:

(1) 最小描述长度原则(MDL)

思想: 最简单的解释最期望的。

方法: 对 Decision-Tree 进行二进位编码, 所需二进位最少编码的树即为“最佳剪枝树”。

(2) 期望错误率最小原则

思想: 选择期望错误率最小的子树进行剪枝。

方法: 对树中的内部节点计算其剪枝/不剪枝可能出现的期望错误率, 比较后加以取舍。

最常用的简化决策树方法有以下几种:

(1) 预先剪枝

在上述的建树算法中, 要求每个叶节点中的训练实例都属于同一类, 并将此作为算法终止条件。在这种情况下, 决策树的错误率为 0。然而在预先剪枝算法中, 并不使用这个标准, 而是在这个标准得到满足之前就停止继续扩展决策树。一种最为简单的方法就是在决策树达到一定高度的情况下就停止树的扩展, 这在某些情况下能得到比较好的效果。另一种更为通用的方法是计算每次扩展对整个系统性能的增益, 如果这个增益值小于某个阈值, 就不再扩展, 即使有些叶节点的实例不属于同一类, 算法也要终止。

(2) 后剪枝

在后剪枝算法中, 一个未经剪枝的决策树 T 作为输入, 然后输出一个被剪枝了的决策树 T' 。剪枝的过程是将一些子树删除, 而用叶节点来代替, 这个叶节点属于的类别用其所对应的子树中大多数训练实例所属的类别代替, 并且在相应叶节点上标出属这个类的训练实例所占的比例。显然, 经过剪枝的决策树 T' 对于训练集的错误率已经不再是 0, 但是由于用这种剪枝算法位于底层的子树将被优先剪枝, 而这些节点都只包含了很多的训练实例, 所以这种方法可以减少噪声对决策树构造的影响; 此外, 这种方法能够产生一组而不是一棵决策树, 为最终的结果提供了更多的选择。

(3) 增量树学习

在一般情况下,为了解决内存空间的问题利用增量树学习(incremental tree sizing)的方法,通过逐步增加训练实例增量式地构造、剪枝决策树。

4. k 近邻方法

k 近邻方法的基本原理很简单,即通过找出训练集中与待分类文档最相似的 k 篇文档,然后根据这 k 个文档的类别确定该文档的类别。常用的相似度度量标准是欧氏距离和余弦夹角。该方法不同于前面几种方法之处是,它是一种基于实例的“惰性”学习算法,即该方法不需要一个独立的模型训练过程^[16]。下面给出两类问题的 K-NN 算法描述:

- ① 输入待分类文档 y 和训练集 X ;
- ② 计算 y 和 X 中所有文档的相似度;
- ③ 找出 X 中和 y 最相似的 k 篇文档;
- ④ 设 n_1, n_2 分别是 k 篇文档中属于 C_1 和 C_2 类的文档数目,计算以下两个条件概率:

$$P(C_1 | y) = \frac{n_1}{n_1 + n_2} \quad P(C_2 | y) = \frac{n_2}{n_1 + n_2} \quad (8-35)$$

- ⑤ 如果 $P(C_1 | y) > P(C_2 | y)$, 则 y 属于 C_1 类, 否则属于 C_2 类。

5. 最大熵分类方法

最大熵模型是一种可以集成不同种类信息的分类模型,它把在训练集中的、与分类有关的数据描述为一系列的特征。这些特征在大多数情况下是一些信号函数,例如在文档自动分类中,所使用的特征函数的形式如下:

$$f_i(\mathbf{x}_j, c) = \begin{cases} 1, & \text{如果 } s_{i,j} > 0 \text{ 且 } c = 1 \\ 0, & \text{其他} \end{cases} \quad (8-36)$$

其中 $s_{i,j}$ 是词 i 在文档 j 中的权重。然后使用最大熵模型的变换形式——对数线性模型来整合这些特征。

$$P(\mathbf{x}, c) = \frac{1}{Z} \prod_{i=1}^K \alpha_i^{f_i(\mathbf{x}, c)} \quad (8-37)$$

式中, K 是模型中特征的规模, Z 是归一化用参数。为便于计算, 对式(8-37)两边取对数, 得到

$$\log P(\mathbf{x}, c) = -\log Z + \sum_{i=1}^K f_i(\mathbf{x}, c) \log \alpha_i \quad (8-38)$$

为求得式(8-37)中的一个最大熵分布 P^* , 且使得该分布满足下面的约束条件:

$$E_{P^*} f_i = E_{\bar{P}} f_i \quad (8-39)$$

求解过程是一个迭代循环过程,通常采用 GIS(generalized iterative scaling) 算法(见 4.5.1 节)。

在近来的基于最大熵方法的文档分类系统中,采用了实值形式的特征,而且提出了改进的 GIS 算法,参见参考文献 17。

以上介绍了常用的文本自动分类系统的主要步骤和方法。不同的方法有不同的优缺

点,在实际的系统实现中,应根据系统对精度、算法复杂度、算法实现难易程度的不同要求,选择不同的算法。另外,与自动分类技术相近的技术——自动聚类技术,也可以用于文本分类。自动聚类技术适用于系统没有预先定义类别集,训练数据也没有标注类别信息的情况,可直接由训练数据之间的相似性进行分组,是一种典型的无教师的机器学习问题。目前的文本聚类方法大致可以分为层次聚类法和平面聚类法。由于实际应用较多的是有导师学习的自动分类技术,因此在此对自动聚类技术不再详述。对此感兴趣的读者可参见参考文献 18。

8.5 小结

信息检索综合应用了自然语言处理与数据库等领域的知识,在社会生产的很多领域已经被广泛采用,例如知识管理系统、图书档案管理系统等,这些应用大都是基于关系数据库的,其相关技术已经比较成熟。近来随着互联网的迅速发展,对信息检索提出了更高的要求。现有的互联网搜索引擎都是基于关键字,然后通过关键字之间的逻辑运算,来完成一些高级检索。而对更高级的检索系统而言,像自然语言问答系统,其目标不仅是返回给用户所要查找信息所在的文本,而且要提取其中与用户查询相关的答案,以减少用户在文本中的查找时间,即要对网页信息内容进行检索,这就需要自然语言处理的一些基本技术的支持,例如词性标注、句法分析、名实体识别、语义分析等。而这些技术相对来说大都不是非常成熟。其次,信息检索就本质而言就是一个分类过程,尽管现有的分类算法大都能保证分类的精度和召回率,但是当把这些分类算法应用到海量数据集时,一些高精度的分类算法,由于其时间复杂性太高,会导致检索系统的整体性能下降。因此,要做一个高性能的信息检索系统,还需要在自然语言处理、文本分类、信息抽取等相关领域,做大量的研究工作。

参考文献

- 1 Rada R., Hafedh M., Bicknell E., and Blettner M. Development and application of a metric on semantic nets. *IEEE Transactions on System, Man, and Cybernetics*, 1989, 19(1):17~30
- 2 Lee J. H., Kim M. H., Lee Y. J. Information Retrieval based on conceptual distance in IS-A hierarchies. *Journal of Documentation*, 1993, 49(2): 188~207
- 3 Miller G., Beckwith R., Fellbaum C., Gross D. and Miller, K. Introduction to WordNet: An Online Lexical Database. *International Journal of Lexicography*, 1990, 3(4): 235~244
- 4 Resnik P. Using information content to evaluate semantic similarity in a taxonomy. In: *Proceedings of IJCAI*, 1995
- 5 P. Brown, S. Della Pietra, V. Della Pietra and R. Mercer. Word sense disambiguation using statistical methods. In *Proceedings of the 29th Meeting of the Association for Computational Linguistics (ACL-91)*, Berkley, C. A., 1991, 264~270
- 6 Lillian Lee. Similarity-Based Approaches to Natural Language Processing: [Ph. D. thesis] Harvard University Technical Report TR-11-97

- 7 梅家驹等. 同义词词林. 上海: 上海辞书出版社, 1983
- 8 Kjersti Aas, Line Eikvil. Text Categorization: A Survey. Technical Report. Norwegian Computing Center, 1999
- 9 G. Salton , M. J. McGill. An Introduction to Modern Information Retrieval. McGraw-Hill, 1993
- 10 M. W. Berry, S. T. Dumais and G. W. O'Brien. Using Linear Algebra for Intelligent Information Retrieval. SIAM Review, 1995, 37: 573~595
- 11 T. Joachims. A probabilistic Analysis of the Rocchio algorithm with TFIDF for Text Categorization. In Proc. of the Fourteenth International Conference on Machine Learning, 1997
- 12 Mitchell T. Machine Learning. McGraw Hill, 1996
- 13 边肇祺等. 模式识别. 北京: 清华大学出版社, 1998
- 14 Quinlan J. R. Introduction of Decision Tree. Machine Learning, 1986
- 15 S. Cost, S. Salzberg. A Weighted Nearest Neighbor Algorithm for Learning with Symbolic Features. Machine Learning, 1993, 10(1):57~78
- 16 Kamal Nigam, John Lafferty, et al. Using maximum entropy for text classification. In proceedings of the IJCAI-99 workshop on information filtering, Stockholm, SE, 1999
- 17 Michael Steinbach, George Karypis. A Comparison of Document Clustering Techniques: [Technical Report]. Department of Computer Science and Engineering, University of Minnesota, 1998
- 18 Ricardo Baeza-Yates & Berthier Ribeiro-Neto. Modern Information Retrieval. Addison-Wesley, New-York, 1999
- 19 Frakes W. B. and Baeza-Yates R. Information Retrieval: Data Structures & Algorithms. Prentice Hall, 1992
- 20 Greengrass E. Information Retrieval: A Survey. UMBC CADIP, 2000
- 21 van Rijsbergen C. J. Information Retrieval (2nd ed.). London: Butterworths, 1979
- 22 Lee J. H. Combining multiple evidence from different properties of weighting schemes. Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1995, 180~188
- 23 G. Salton, M. J. McGill. An Introduction to Modern Information Retrieval. McGraw-Hill, 1993



文字识别技术

本章首先介绍联机手写体文字识别技术的国内外概况、联机手写体文字识别基本的识别方法和典型的系统；然后探讨利用统计语言模型进行联机手写体文字识别后处理的方法，及其在手写体汉字识别中的重要性；进而描述手写体汉字识别的后处理数学模型、基于词网格的手写体汉字识别的语言学解码方法和最优语句候选的搜索算法；最后描述联机手写体汉字识别后处理系统的构成。

9.1 引言

电子计算机的出现和发展，使人类社会的经济、科学、技术发生了日新月异的变革。电子计算机信息量大，处理速度快，越来越广泛地被应用于信息处理。以电子计算机为基础手段的现代信息处理技术已成为信息社会的重要标志。

信息处理技术包括信息的输入、信息的处理和信息的输出 3 个方面。信息处理的一个重要内容就是文字信息处理。汉字是我国文化的宝贵遗产，是汉语的书写符号，是人们学习、工作和交流的主要手段和工具之一。汉字对中华民族的政治、经济和文化的发展有着不可磨灭的功绩，全世界使用汉字的人数之多是任何其他文字所不能及的。因此，汉字信息处理技术不仅在我国，而且在全世界都有着极为重要的实用价值和现实意义^[1~6]。

在汉字信息处理技术中，汉字输入一直是人们重视并有待于妥善解决的问题。因为汉字的结构决定了汉字不能像英文那样简单地通过键盘上二十多个字母键送入计算机。如何把汉字信息高速准确地输入到计算机中，成了汉字信息处理中最为关键的问题之一。

目前汉字的输入方式大致有两种方法：一种是非智能的输入方式，如各种编码输入法，整字大键盘输入法；另一种是智能拼音输入法，如汉字识别和语音输入。由于汉字字数多，字形复杂，非智能输入方式要求记忆复杂的编码，只有专业操作人员才能熟练掌握使用，非专业操作员使用起来很不方便，效率很难提高，因此必须使用智能化输入方式。

汉字识别是指计算机自动辨识印刷或手写体汉字的技术,依据识别对象及输入设备的不同,汉字识别可以分为印刷体汉字识别、脱机手写汉字识别和联机手写体汉字识别。

印刷体汉字识别是由计算机识别通过光电扫描仪输入的印刷体汉字的技术;脱机手写汉字识别是由计算机识别人写在纸上的手写体汉字的技术;联机手写体汉字识别则是由使用者在数字化仪上写入汉字,计算机实时接收汉字笔记,将其识别为标准内码的技术。这3种汉字识别技术依据不同特点,各自有着不同的应用领域,其中尤其以联机手写体汉字识别更具有实用价值。

随着计算机技术的应用与普及,人们已不满足于在办公室内使用计算机,而希望随时通过计算机与各地联系,处理信息。这一要求促使计算机向便携和小型化方向发展,出现了操作简单,易于携带的掌上电脑和个人数据助理(PDA)产品,尤其是笔操作的PDA产品,因为取消了键盘,采用了图形用户界面这一最接近人类自然习惯的用户接口,具有极好的市场前景。而目前影响这一产品发展的关键技术之一就是联机手写体汉字识别技术。

尽管目前联机汉字手写体识别技术已进入实用化阶段,许多科研机构推出了各自的联机汉字手写体识别系统。但是由于被识别的汉字之间的结构相似性以及文字图像上的噪声,严重制约着汉字识别的识别率的进一步提高,尤其随着汉字识别技术正在朝着连笔草书方面的发展,汉字识别面临的技术难度日益增大。解决方案之一便是利用人类能够自觉地使用词典知识和语言学知识,根据句子中的上下文信息对带有噪声的文字图像进行理解和识别。根据仿生学、认知科学原理,将语言学机制引入到汉字识别问题中,采用一个语言解码器(linguistic decoder)自动处理句子中的上下文信息,从而对文字图像进行某种程度的理解,以便自动修正识别错误,选择最优的候选识别字,从而提高手写汉字识别的正确率。

本章在介绍联机手写体的国内外概况、联机手写体基本的识别方法和典型的联机手写体的同时,将介绍手写体汉字识别的后处理系统,同时验证统计语言模型提高联机手写体汉字识别系统识别准确率的有效性。

9.2 联机手写体汉字识别的国内外研究概况

9.2.1 国外研究概况

字符识别的发展历史可以追溯到20世纪40年代,随着电子计算机的发明出现了现代意义上的光学字符识别系统(OCR)。50年代末期发明了电子图形输入板,电子图形输入板可以实时地将笔在图形板上的移动笔迹坐标数据输入计算机,因而可以用笔在电子图形板上写字的方法将汉字实时输入计算机。1966年美国麻省理工学院的J. Lu发表了题为“Real time Chinese handwriting recognition machine”的学位论文^[7]。其后美国的匹兹堡大学、IBM公司等均有研究成果发表,加拿大的大学也有学者从事这方面的研究^[8]。

20世纪70年代中期,在政府的支持下,日本开始了面向实用的汉字识别研究,并于1980年首先研制出可以识别2000个汉字,识别速度小于0.4s/字的联机汉字识别系统。随后日本各大公司和学校也相继推出了各自的手写汉字识别系统^[9~11]。

目前国际上从事字符识别文档理解方面研究的学术团体的主要代表有：

(1) 美国 Buffalo 大学的 CEDAR(Center of Excellence for Document Analysis and Recognition)。该中心成立于 1978 年,最初的研究方向是模式识别和人工智能。该中心已经推出了一些可识别邮件上手写的姓名、地址和邮政编码等方面的产品。目前主要的研究方向有表格处理(forms processing)、手写地址翻译(handwritten address interpretation)、智能字符识别(intelligent character recognition)、日语文本识别(japanese document recognition)、手写文本识别(handwritten text recognition)以及名片机、姓名地址阅读器等。

(2) 加拿大 Concordia 大学的 CENPARMI(Center for Pattern Recognition and Machine Intelligence)。该中心成立于 1988 年,目前主要的研究方向有手写及签字识别(handwriting recognition and signature verification)、图像分析及文本处理(image analysis and document processing)、多特征融合及分类技术(combination of different types of features and classification techniques)、专家系统及神经元网络(application of multiple expert systems and neural networks)、自然语言处理等(natural language processing)等。

(3) 荷兰的 NICI(Nijmegen Institute for Cognition and Information)。该研究所成立于 1990 年,目前主要的研究方向有联机手写字符数据库收集管理(on-line handwriting database collection project)、语音及联机字符识别(combining speech and on-line handwriting recognition)以及在脱机字符识别中应用联机知识(using knowledge of on-line handwriting production in off-line recognition)等项目。

另外,美国的 Berkeley 大学、Maryland 大学,英国的 Essex 大学、Nottingham 大学以及日本、韩国等地也有大量学者从事相关研究工作。

9.2.2 国内研究概况

国内的联机手写体汉字识别研究始于 20 世纪 70 年代末。1983 年,在武汉召开的中文信息学会第二届全国学术会议上,哈尔滨工业大学、江西冶金工程学院、上海仪器仪表研究所等单位提交了有关联机手写体识别的研究论文。此后,哈尔滨工业大学、总参通讯部、清华大学、西安交大等单位均有研究成果通过鉴定。

汉字识别技术的发展可以分为 3 个阶段。从 20 世纪 70 年代末到 1983 年,是联机手写体汉字识别技术的理论探索、方法研究阶段。这一阶段的研究成果大多以论文的形式出现。1984 年到 90 年代是联机手写体汉字识别实际系统的研究阶段。在这个阶段中研制出了几个初步实用的装置,其指标为识别字数 3755~6763 个,识别率在 85% 左右,识别速度基本上能跟上人书写的速度,书写时要求笔划数和笔划类型基本正确。20 世纪 90 年代之后,联机手写体汉字识别技术逐步走向实用阶段。在国家的大力支持和市场需求的促进下,初步可实用的手写体汉字识别系统陆续出现。目前国内从事字符识别方面研究的学术团体主要有清华大学、北京大学、中科院自动化所和哈尔滨工业大学。另外,我国台湾地区的汉字识别研究也硕果累累,主要研究产品有蒙怡中国笔、新颖中文手写、孙悟空中文手写等。

9.3 联机手写体汉字识别方法综述

本小节将简单概述常用的联机手写体汉字识别方法。

9.3.1 基于统计的识别方法

手写体汉字识别的系统方法是以统计为特征基础,以汉字的结构信息为辅助特征的识别方法。不同的方法往往采用不同的特征,但其最后的识别方法则是一致的,即都采用特征向量与模板匹配的方法。但是这类方法分辨相似字的能力较弱,因此在识别字集增大或者畸变幅度增大时,该类方法宜作为组分类方法使用。在早期的汉字识别研究中,人们往往用整字的统计特征来识别汉字。下面介绍其中主要的几种。

1. 续波形谱分析特征^[12]

人们书写时笔尖的运动轨迹可以看作是两个坐标采样的序列 $x(n), y(n), n=1, 2, \dots, N$, 对此序列做傅里叶变换,把展开式的低阶系数作为汉字识别的特征。其展开系数为

$$a_i = \frac{2}{N} \sum_{n=1}^N x(n) \cos\left(\frac{n_i}{N}\right) \quad (9-1)$$

$$b_i = \frac{2}{N} \sum_{n=1}^N y(n) \sin\left(\frac{n_i}{N}\right) \quad (9-2)$$

由于展开式中的低阶系数能够有效地反映曲线的大致变化,并且能够克服高频噪声,因此可以用来描述一个汉字:

$$L = (a_0, b_0, a_1, b_1, \dots, a_M, b_M) \quad (9-3)$$

其中 M 的大小根据情况而定。

2. 字曲线描述特征^[13,14]

将输入汉字中的前一笔划的结束点与下一笔划的起始点连接起来,将汉字看作是一条曲线,然后利用文献 14 中提供的算法将这条曲线表示为一个具有固定分段数的 freeman 链码序列,并以此序列来描述一个汉字。匹配时,将链码序列中的每一个元素与特征模板中的每一个元素作比较,产生相应的误差向量,再将误差向量与一个加权向量相乘,得到匹配误差值,以匹配误差最小的模板作为识别结果。

此外,在文献 15 中提出了一种在整字曲线描述基础上的动态规划算法,该算法也是将输入汉字前后笔划相连,把汉字看成是一条连续的曲线,然后将这一曲线按照其运笔方向提取出方向序列及长度序列,并将这些序列与字典中的标准序列作动态规划匹配,得出的距离最小者即为识别出的汉字。

以上介绍的基于整字的联机汉字识别特征及其匹配方法的共同特点是允许输入汉字的笔划之间有连笔,但是要求笔顺正确。另一方面,由于忽略了汉字的结构信息,因此当识别字域增大时,得不到满意的识别效果。

9.3.2 基于结构的识别方法

由于汉字是有结构的线段图形,汉字笔划之间有一定的相关性,无论如何变化,其相互位置不变,因而可以从结构方面入手进行汉字识别^[15~17]。

基于结构的识别方法需要先识别笔划,然后以这些笔划作为主要特征来识别汉字。常见的笔划基本类型有横、竖、撇、捺和点,这些基本笔划可以组成复杂笔划。笔划的识别方法主要有以下几种。

1. 按笔划的书写方向及其变化识别笔划

不同的联机识别系统对笔划有不同的定义,通常笔划可以分为基本笔划和复合笔划两类。基本笔划包括横、竖、撇、捺和点;复合笔划指的是由基本笔划组合构成的笔划。

在 X-Y 平面上进行分割,根据人书写横、竖、撇、捺的习惯,确定笔划的基本方向。基本笔划都是直线型笔划,输入一笔划的坐标串后,若判定该笔划不是复合笔划,则根据该笔划的起、终端点坐标计算出基本笔划类型。如果基本笔划包括点,则计算出笔划长度,当长度小于一个阈值时,该笔划即为点。

根据书写笔划时方向的变化,检测出笔划中的折点,根据折点的弯折方向和个数,就可以进行复合笔划的判断。

2. 动态规划识别笔划^[18,19]

动态规划识别笔划的方法是直接根据笔划上的线段方向,形成笔划方向码序列,然后用动态规划法来识别笔划。

设输入汉字为

$$X = x_1, x_2, \dots, x_n$$

标准模板为

$$G = g_1, g_2, \dots, g_m,$$

两个笔划方向码符号 x_i 与 g_i 之间的距离 $d(x_i, g_i)$ 定义为 $d(i, j)$:

$$d(i, j) = \min(|x_i - g_j|, 8 - |x_i - g_j|) \quad (9-4)$$

其中, x_i 与 g_i 是方向符号,其值为 0~7,表示 X-Y 平面上的 8 个方向。则输入汉字与标准模板之间的距离 $D(X, G)$ 可由动态规划法求出:

$$d(i, j) = \min[d'(i-1, j-1), d'(i, j-1), d'(i-1, j)] + d(i, j) \quad (9-5)$$

其中, $d'(i, j)$ 表示前串 x_i 与 g_i 之间的距离, $d'(1, 1) = d(1, 1)$ 。最后有

$$D(X, G) = d(n, m) \quad (9-6)$$

以上迭代过程实质上是以伸长或缩短某一方向码的长度来找到两个序列的最优匹配。这种方法的缺点是迭代速度慢,从而影响识别速度。

3. 模糊属性自动机识别笔划

文献[16]提出了一种识别笔划的模糊属性自动机,它利用笔划的方向和长度信息,借助模糊信息处理方法,以不变嵌入原理为着眼点,提出了一种模糊属性文法及其相应的模

糊属性自动机。这种方法的文法形式是有限状态文法,但语义规则中有上下文信息。一个输入笔划的方向码序列和方向码长度序列输入自动机后,得到的结果不是真假二值,而是这一笔划属于自动机所规定的笔划类的隶属度值。隶属度值最大的一类即为识别出的笔划类型。

经过笔划识别之后,输入文字被描述为一个笔划序列或笔段序列。汉字识别问题即转化为对输入笔划、笔段序列的识别问题。

由于汉字的组成原则是:笔段—笔划—部件—整字,所以根据识别出的笔划,笔段序列首先进行部件识别,然后识别出整个汉字。

9.3.3 基于神经元网络的识别方法

人工神经元网络方法也是研究手写汉字识别的常用工具。国际上比较著名的 Fukushima 模型就是一种专门用于视觉模式识别的模型,其中 Neocognition 模型模仿了人脑视觉系统的细胞感受理论,结构精美复杂。由于这种结构具有对位置、大小变化的容忍能力,所以更接近人脑^[20,21]。

此外,清华大学的吴佑寿等提出了一种改进的自组织聚类网络^[22],这种方法可以用于汉字的自组织聚类。中国科学院的刘迎建等提出了一种组块神经元网络模型^[23],该模型是由大量基本单元——组块神经元通过一定组织关系构成的复杂系统。

9.3.4 基于机器学习的识别方法

模式识别是一种智能行为,传统模式识别方法,即统计方法和结构方法由于本身在特征提取、模式描述方面的单调性和识别过程的简单性,造成了现有手写体字符识别系统水平不高,系统对用户限制较多等问题。随着人工智能技术在模式识别领域的不断渗透,几年前人们提出了基于知识的模式识别思想,并以此构造了一些实用的专家系统。传统识别与智能识别的差别在于后者更多地利用了领域知识和人工智能技术,智能识别的难点在识别知识的获取。文献 24 提出了一种机器学习字符识别系统 mlcrs,该系统结构如图 9-1 所示。

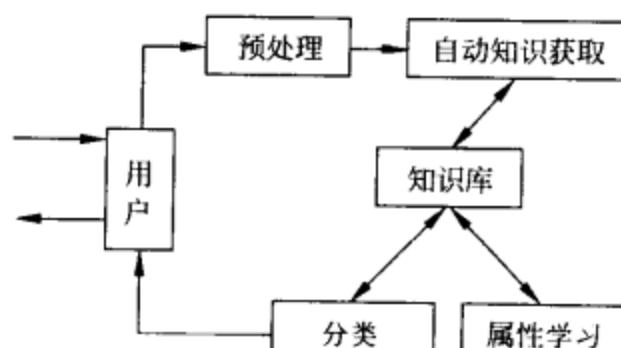


图 9-1 mlcrs 系统结构

mlcrs 知识库中的知识采用树形结构,其节点为字符实例的优选属性,识别规则的形式为

$$A_1 \wedge A_2 \wedge \cdots \wedge A_i \rightarrow "e_i"$$

表示如下的推理路线:

$$A_1 \rightarrow A_2 \rightarrow \cdots \rightarrow A_i \rightarrow "e_i"$$

最终推出的树叶就是识别结果。该系统不同于 ID3 算法,它采用概念函数来指导优选属性的工作。在各概念节点上结合神经网络的 Kohonen 自组织映射网来模仿人类的“形象思维”,构造节点的微神经网络连接结构,形成了系统独特的工作方式,使机器既具有部分“逻辑思维”,又具有部分“形象思维”的能力,取得了较好的实验效果。

9.4 典型联机手写体汉字识别系统

本节将简要概述中国科学院自动化所的“汉王中文手写”和哈尔滨工业大学的“豪文”中文手写两个典型的中文手写系统。

9.4.1 汉王中文手写体汉字识别系统

汉王中文手写汉字识别系统采用传统的结构模式识别方法,借助模糊信息处理方法,以不变嵌入原理为着眼点,提出了一种模糊属性文法及模糊属性自动机,把在线手写汉字分为笔划、笔段、字跟、整字等几个层次,最后进行词组校对。该系统流程图如图 9-2 所示。

该系统首先采用以模板为核心的模糊属性自动机进行手写汉字的笔划识别,去掉笔划中的畸变笔划,将汉字笔划拆分为笔段,在笔段一级采用字形结构排序法对笔段重新排序,从原理上克服了笔顺变化对识别的影响。在字根及整字一级采用与识别汉字笔划类似的文法进行描述与归纳,在词组阶段采用上下文相关信息,通过粗分类匹配一部分词,达到加快速度的目的。该系统的优点是对笔划变化的容忍度大,缺点是运算比较复杂,自动机的设计比较困难。

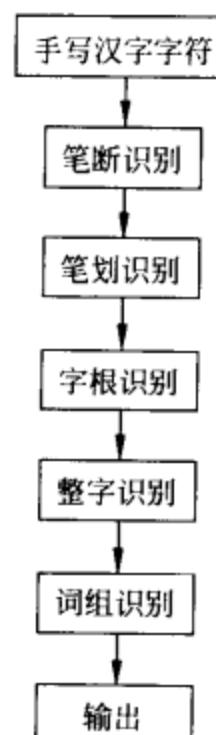


图 9-2 汉王中文手写汉字识别系统流程图

9.4.2 豪文中文手写体汉字识别系统

豪文中文手写体汉字识别系统采用了传统的统计模式识别与结构模式识别相结合的方法,建立了识别系统的信息传递模型。采用了多分支多级分类技术,解决了多级分类过程中误差累积的问题,整个识别过程分为笔段识别、字根识别、整字识别 3 个层次,其识别流程图如图 9-3 所示。

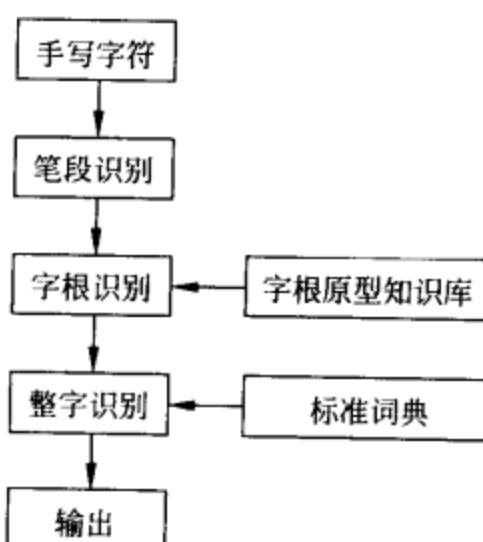


图 9-3 豪文中文手写汉字识别系统流程图

该系统在笔段排序和部件分析识别过程中采用的知识是一种专家知识,表现形式为一条条的规则。在整字识别中采用的模板也是基于专家知识确定的,因而具有知识表达精炼,内存开销小,识别效果好,识别速度快等优点。

该系统的缺点是知识库维护困难,需要专

家才能改动知识库中的规则,适应性差,制作不方便。

9.5 联机手写体汉字识别后处理系统

典型的手写体汉字识别系统一般由两个部件组成:前端单字识别器(single character recognizer, SCR)和汉字识别后处理器(post-processor)^[25,26],后者有时也称做语言解码器。由于 SCR 一般可以根据输入的汉字图像序列生成一个由候选汉字组成的矩阵,因此目前绝大多数的汉字识别后处理系统很自然地用了基于字的汉语 N-gram 模型计算每个候选语句的出现概率,而这种后处理方法没有充分利用汉语词典中丰富的构词信息。实际上对于汉字识别后处理来说,汉语词典应该是一种非常重要的信息源。从直观上看,相邻列上的候选汉字能否组成一个词的信息有助于正确选择最优的语句候选。再者,基于词的汉语 N-gram 模型的信息熵远小于基于字的汉语 N-gram 模型的信息熵。因此本章将介绍基于词的汉语 N-gram 模型的汉字识别后处理系统。另外一些研究者试图通过词法、句法和语义信息的分析来进行后处理。遗憾的是,这些方法需要耗费大量的系统时间,不适用于联机手写体汉字识别系统。

9.5.1 手写体汉字识别模型

一个典型的手写体汉字识别系统的结构示意图如图 9-4 所示。

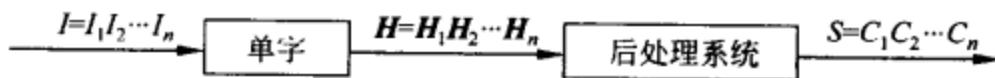


图 9-4 手写体汉字识别系统结构示意图

在图 9-4 中,SCR 的输入端接收一个手写体汉字图像序列,该图像序列记作 $I=I_1 I_2 \cdots I_n$,其中 I_i 表示单个手写体汉字图像。对于每个输入的手写体汉字图像 I_i , SCR 生成

一个候选矢量 H_i , $H_i = \begin{cases} (C_{1i}, c_{1i}) \\ (C_{2i}, c_{2i}) \\ \vdots \\ (C_{mi}, c_{mi}) \end{cases}$, 其中 H_i 由一组形如 (C_{ji}, c_{ji}) 的二元对组成, C_{ji} 表示

第 i 个手写体汉字图像 I_i 的第 j 个候选汉字, c_{ji} 是一个概率值,一般称之为该候选字的可信度(confidence),该可信度表示手写体汉字图像 I_i 的第 j 个候选汉字的模式类和 I_i 之间的相似度。显然,对于一串输入的手写体汉字图像 $I=I_1 I_2 \cdots I_n$, SCR 将生成一个候选矩阵 $H=H_1 H_2 \cdots H_n$ 。如果每个汉字候选用一个节点表示,那么 H_i 的所有候选汉字将组成一列节点, H 的所有的候选汉字可以用 n 列节点表示。如果将相邻列的节点全部用有向边连接起来,则它们将构成一个有向图。该有向图(如图 9-5 所示)构成汉字识别后处理中语言解码器的状态空间。

将该有向图中自左向右的每一条路径看作一个候选语句。显然,一个候选语句 S 由一个由 H 中的相邻列的候选汉字组成的串组成, $S=C_{j_1} C_{j_2} \cdots C_{j_n}$, 这里每个 $C_{j_i} \in H_i$, 并且 $0 < i \leq n$, C_{j_i} 是 H_i 中的一个候选汉字。汉字识别后处理中语言解码器将使用统计语

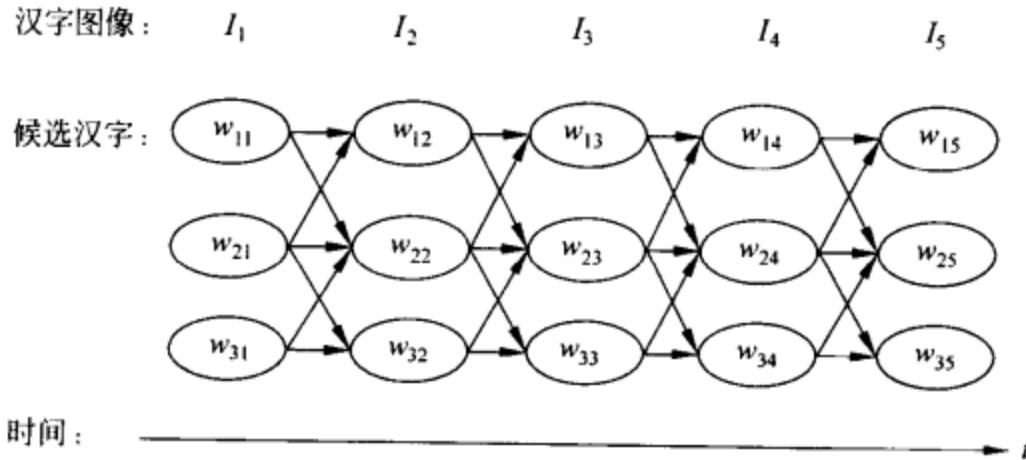


图 9-5 语言解码器的状态空间

言模型对候选矩阵 $H = H_1 H_2 \cdots H_n$ 进行解码，并选择最优的候选语句 S 作为输出。即选择具有最大概率 $P(S|I)$ 的候选语句 \hat{S} 。根据贝叶斯公式，可以推导出下面的公式：

$$\hat{S} = \arg \max_s P(S | I) = \arg \max_s \frac{P(I | S)P(S)}{P(I)} = \arg \max_s P(I | S)P(S) \quad (9-7)$$

式中，SCR 提供对 $P(I | S)$ 的估计，而 $P(S)$ 由汉字识别后处理中的汉语统计语言模型计算。

9.5.2 $P(I | S)$ 估计

SCR 一般通过比较单个手写体汉字图像 I_i 和其所有的候选汉字对应的模式类对 I_i 进行识别。显然，SCR 没有依赖图像 I_i 的上下文信息。因此公式(9-7)中的概率 $P(I | S)$ 可以由下面的公式计算：

$$P(I | S) = \prod_{i=1}^n P(I_i | C_{j_i}) \quad (9-8)$$

式中， $S = C_{j_1} C_{j_2} \cdots C_{j_n}$ ， C_{j_i} 是汉字图像 I_i 的候选汉字之一。公式(9-7)可以重写为

$$\hat{S} = \arg \max_s P(I | S) = \arg \max_s P(I | S)P(S) = \arg \max_s \prod_{i=1}^n P(I_i | C_{j_i}) \cdot P(S) \quad (9-9)$$

根据 Bayes 公式，可知

$$P(I_i | C_{j_i}) = \frac{P(C_{j_i} | I_i) \cdot P(I_i)}{P(C_{j_i})} \quad (9-10)$$

所以公式(9-9)可以重写为：

$$\hat{S} = \arg \max_s \prod_{i=1}^n P(I_i | C_{j_i}) \cdot P(S) = \arg \max_s \prod_{i=1}^n \frac{P(C_{j_i} | I_i) \cdot P(I_i)}{P(C_{j_i})} P(S) \quad (9-11)$$

在公式(9-11)中，对于 H_i 中所有的候选汉字来说， I_i 是相同的，所以 $P(I_i)$ 可以被省略。 $P(C_{j_i})$ 是候选汉字 C_{j_i} 对应的模式类的先验概率，并且 SCR 一般认为 $P(C_{j_i})$ 是均匀分布的，所以 $P(C_{j_i})$ 可以被省略。因此公式(9-11)可以进一步重写为

$$\hat{S} = \arg \max_S P(S | I) = \arg \max_S \prod_{i=1}^n P(C_{j_i} | I_i) \cdot P(S) \quad (9-12)$$

$P(C_{j_i} | I_i)$ 可以通过候选汉字 C_{j_i} 的模式类和 I_i 之间的可信度 c_{ji} 来替代:

$$P(C_{j_i} | I_i) = c_{ji} \quad (9-13)$$

9.5.3 $P(S)$ 估计

统计语言模型的任务一般用于对句子的出现概率 $P(S)$ 进行估计。首先,根据某种汉语词的切分算法,候选语句 $S=C_{j_1}C_{j_2}\dots C_{j_n}$ 可以被分割成一个词的序列。令

$$S = w_1^m = w_1 w_2 \dots w_m \stackrel{\text{def}}{=} w_1^m$$

则

$$P(S) = P(w_1) \prod_{i=2}^m P(w_i | w_1^{i-1})$$

这样,句子概率的估计 $P(S)$ 就可以通过词的上下文条件概率 $P(w_i | w_1^{i-1})$ 计算出来。

$P(w_i | w_1^{i-1})$ 经常记作 $P(w_i | h_i)$, 这里 $h_i = w_1^{i-1}$ 称做词 w_i 的上下文历史。前文曾经提到,假设词典 V 的规模为 $|V|$, 那么由 h_i 组成的上下文参数空间为 $|V|^{i-1}$ 。显然, $P(w_i | h_i)$ 是不可计算的。因为词 w_i 的上下文 h_i 在训练语料库中重复出现的次数十分有限,一般难以满足其统计充分性的要求,也就无法得到其可靠的概率估计。实际上,大多数统计语言模型均通过对上下文 h_i 的空间进行等价类划分,以压缩其参数空间,近似估计 $P(w_i | h_i)$,从而提供词 w_i 所在句子的出现概率 $P(S)$ 的计算机制。

根据统计语言模型对词的上下文空间的等价类的划分方法,可以构造不同结构的基于词的上下文出现概率的统计语言模型。这种统计语言模型主要包括: 基于词的 N -gram 模型、基于词类自动划分的 class N -gram 模型、基于词性的 N -POS 模型、HMM 语言模型和基于决策树的统计语言模型等。

在上述的统计语言模型中,基于词的 N -gram 模型在语音识别和文字识别应用中,占据着绝对主流的地位,其性能优越于其他任何一种统计语言模型。因此本章将主要介绍基于词的汉语 N -gram 模型计算手写汉字识别后处理中的候选语句的出现概率。

9.5.4 基于词网格的手写体汉字识别的语言学解码方法

当输入一个汉字图像序列 $I=I_1 I_2 \dots I_n$ 时,对于每个汉字图像 I_i , SCR 将生成一个候选汉字集合,记作一个候选矢量 \mathbf{H}_i , $I=I_1 I_2 \dots I_n$ 所有的候选汉字将构成后处理的状态空间,如图 9-5 所示。由于图 9-5 中的每个节点代表一个候选汉字,所以大多数的汉字识别后处理方法很自然地用基于字的汉语 N -gram 模型来计算一个候选语句 $S=C_{j_1}C_{j_2}\dots C_{j_n}$ 的概率。遗憾的是,这些方法没能使用汉语词典。而对于汉字识别后处理中的语言解码器来说,汉语词典是一种非常重要的信息资源。直观上,相邻列的候选汉字能否组成一个词是正确选择一个最优的候选语句的关键。由于图 9-5 中的状态空间不能直接处理词,因此通过增加词节点来扩展汉字识别后处理的状态空间。对于所有的相邻汉字节点,

如果它们可以组成一个词，就把它们捆成一个词节点，并把该词节点挂在该词的尾字所在的列中。相邻列的所有字词节点均用有向边连接起来，它们就构成了一个有向图，称之为词网格（如图 9-6 所示）。

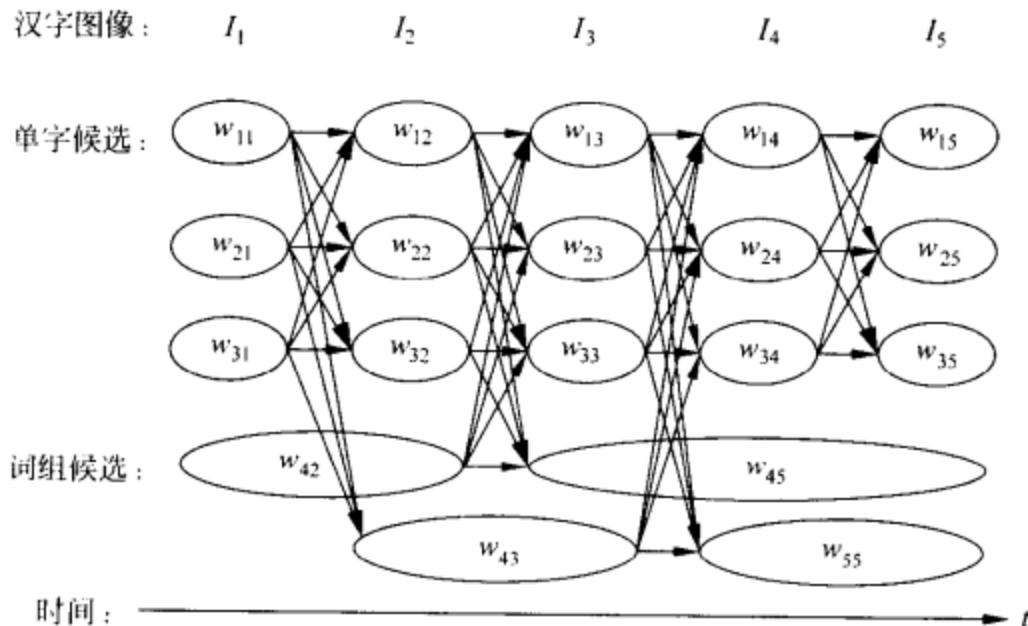


图 9-6 词网格示意图

在词网格中，自左向右且首尾相连的字词节点构成若干条路径，本文把每条这样的路径称做一个候选语句。对于由词序列构成的每条候选语句 $S = w_1 w_2 \dots w_m$ ，本文定义一个候选语句评价函数 $f(S)$ ，来判别该候选语句的优劣，其形式如下：

$$f(S) = P(S) \cdot \prod_{i=1}^m P(w_i | I_{w_i}) = P(w_1) \cdot \prod_{i=2}^m P(w_i | w_{i-N+1}^{i-1}) \cdot \prod_{i=1}^m P(w_i | I_{w_i}) \quad (9-14)$$

式中， I_{w_i} 表示一个对应词 w_i 的手写体汉字图像序列；词 w_i 由一个汉字序列组成，记作 $w_i = C_{j_1 i} C_{j_2 i+1} \dots C_{j_{l-1} i+l-1}$ 。令 $l(w_i)$ 为词 w_i 的长度，它表示词 w_i 包含的汉字个数。 $P(w_i | I_{w_i})$ 可以通过下面的经验公式计算出来：

$$P(w_i | I_{w_i}) = l(w_i) \cdot \alpha \cdot \prod_{k=t}^{t+l-1} P(C_{j_k k} | I_k) \quad (9-15)$$

式中， α 是词节点的加权系数，可以通过大规模训练确定它的最优值；词 w_i 的词长 $l(w_i)$ 与概率 $P(w_i | I_{w_i})$ 成正比。手写体汉字识别后处理的任务是发现一个具有最大评价值 $f(S)$ 的语句候选 S 。

9.5.5 联机手写体汉字识别后处理系统

联机手写体汉字识别系统具有实时性，因此主要采用基于词的汉语 Bigram 模型和简单的 Cache 自学习机制作为语言解码器的统计知识库，以保证系统的实时性要求。

联机手写体汉字识别后处理系统的结构如图 9-7 所示。该系统的核心算法由 3 个步骤组成：第一，词网格生成模块根据候选汉字矩阵生成词网格，构成语言解码器的搜索空间。第二，语言解码器使用 Viterbi 搜索算法在词网格中搜索具有最大路径评价值的候选语句，并将搜索结果作为语言解码器的识别结果，再经过用户的联机校正成为正确的句

子,输出到 Cache 自学习模块。第三,Cache 自学习模块根据用户校正的句子修改相应的 Cache 学习库,调整系统,以逐渐适应用户的习惯。

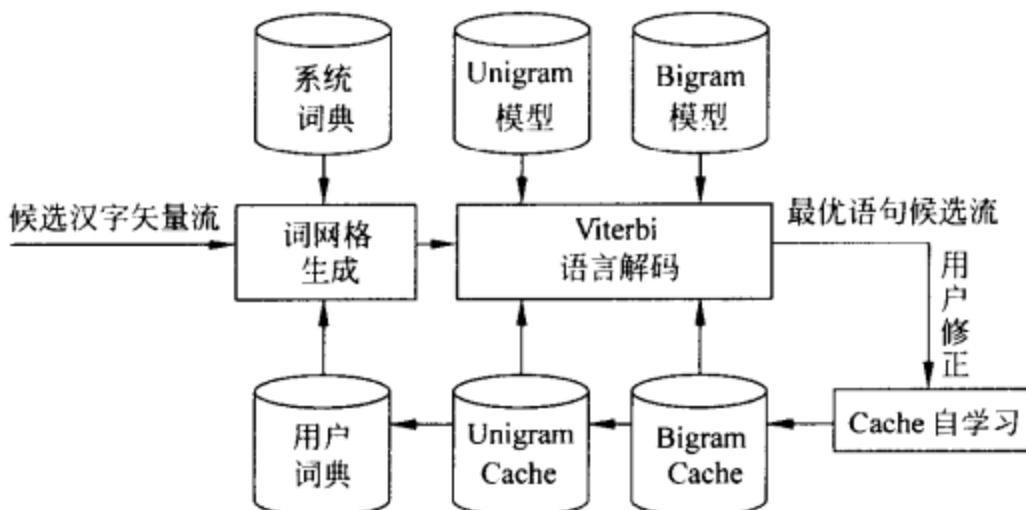


图 9-7 联机手写体汉字识别后处理系统结构示意图

1. 词网格生成模块

词网格生成模块根据联机手写体汉字识别的前端识别器生成的候选汉字矢量序列生成词网格。词网格生成模块由两步组成:首先遍历所有的候选汉字,生成全部的字节点;然后遍历所有的字节点,把相邻列的可组词的字节点全部捆绑成词节点。所有的节点按其结束时刻进行排序,最后生成的词网格构成语言解码器的搜索空间。

2. 最优候选语句搜索模块

在联机手写体汉字识别后处理中,为了保证系统的实时性,采用 Viterbi 搜索算法求解词网格中的最优路径代表的候选语句。这一阶段,根据候选汉字的可信度、汉语中基于词的 Bigram 模型以及 Cache 模型提供的 Bigram、Unigram 概率计算出每个候选语句的评价函数值,以便选择最优的候选语句。

假设当前候选字词节点中的词为 w_j ,那么该节点对应的路径评价值可以记为 $f(w_j)$,其递归计算公式如下:

$$f(w_j) = \max_{w_i \in S_j} (f(w_i) + \log P(w_j | w_i) + \log P(w_j | I_{w_j})) \quad (9-16)$$

其中, S_j 为当前候选节点的前 1 列所有的候选字词节点中的词集; w_i 为 S_j 中的任意一个词; $P(w_j | I_{w_j})$ 为词 w_j 与其对应的文字图像序列之间的相似度,通过公式(9-15)计算出来; $P(w_j | w_i)$ 为二元转移概率。

3. 基于 Cache 的自学习模块

Cache 是一种短期记忆机制。它由一组用于学习目的的统计数据组成,数据来源于用户联机校正的句子,数据存储格式与相应的 Unigram 和 Bigram 相同。语言解码器工作时,Cache 中的概率通过加权可以与系统语言模型的概率结合起来,Cache 的加权系数同系统的学习率成正比。

9.6 小结

本章主要讨论了应用汉语统计语言模型进行手写体汉字识别的语言学后处理的方法。首先描述了统计语言模型在手写体汉字识别的重要性;然后构造了手写体汉字识别后处理的数学模型,提出了基于词网格的手写体汉字识别的语言学解码方法,描述了统计语言模型和系统词典的资源优化问题;最后分别描述了联机和脱机两种方式的手写体汉字识别后处理系统及其各自的最优语句候选的搜索算法。

参考文献

- 1 张忻中. 汉字识别技术. 北京: 清华大学出版社, 1992
- 2 吴佑寿等. 汉字识别原理、方法与实现. 北京: 高等教育出版社, 1992
- 3 舒文豪. 汉字识别研究的现状与展望. 计算机世界, 1987, 11(8)
- 4 J. Mantas. An Overview of Character Recognition Methodologies. *Pattern Recognition*, 1986, 19(6): 425~430
- 5 V. K. Govindan. Character Recognition-A Review. *Pattern Recognition*, 1990, 23(7): 671~683
- 6 Charles. C. Tappent, Ching. Y. Suen and Toro. Wakahara, The State of the Art in On-line Handwriting Recognition. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 1990, 12(8): 768~808
- 7 J. Lu. Real Time Chinese Handwriting Recognition Machine. MIT: [E. E Thesis]. Cambridge, 1996
- 8 E. F. Yhap. An On-line Chinese Recognition System. IBM, *Journal of Research and Development*, 1981, 25(3): 120~124
- 9 R. Oka. Handwritten Chinese, Japanese Character Recognition by Using Cellular Feature. *Proc. 6th Int. Conf. On Pattern Recognition*, 1982, 783~785
- 10 H. Arakawa. On-line Recognition of Handwritten Characters. *Pattern Recognition*, 1984, 16(1): 9~16
- 11 K. Yoshida. On-line Handwritten Character Recognition for a Personal Computer System. *IEEE Trans.*, 1982, 28(3): 202~208
- 12 William Stallings. Approaches to Chinese Character Recognition. *Pattern Recognition*, 1976, (8): 87~98
- 13 W. T. Chen, T. R. Chou. A Hierarchical Deformation Model for On-line Cursive Script Recognition. *Pattern Recognition*, 1994, 27(10): 1365~1373
- 14 C. K. Lin, K. Chin, T. P. Lee. On-line Recognition by Deviation-Expansion Model and Dynamic Programming Matching. *Pattern Recognition*, 1995, 26(2): 259~268
- 15 刘迎建, 戴汝为. 识别联机手写体汉字笔划的模糊属性自动机. 自动化学报, 1988, 014(2): 87~104
- 16 Xia Ying, Sun Chengjiang. Recognizing Restricted Handwritten Chinese Characters by Structure Similarity Method. *Pattern Recognition Letters*, 1990, 11: 67~73
- 17 T. Zong, K. C. Fan. Coarse Classification of On-line Chinese Characters via Structure Feature-Based Method. *Pattern Recognition*, 1994, 27(10): 1365~1373
- 18 陈辉堂. 动态规划在手写在线识别的应用. 见: 第四届全国模式识别与机器智能学术会议论文集, 1986, 204~209

- 19 王小滨等. 一种基于曲线拟合的联机手写汉字笔划的识别方法. 见:全国多媒体与高速信息网络会议 CMIN95,1995
- 20 K. Fukushima , S. Miyake. Necognition: A New Algorithm for Pattern Recognition Tolerant of Deformation and Shift in Position. Pattern Recognition, 1982,15(6):455~469
- 21 K. Fukushima. A Neural Network for Visual Pattern Recognition. IEEE Trans. On Computer, March,1988,65~75
- 22 吴佑寿,徐宁,丁晓青.一种用于神经网络汉字识别系统的自组织聚类方法. 电子学报,1994(5): 1~8
- 23 刘迎建等. 基于认知模型和神经元网络的汉字识别方法. 见:模式识别与人工智能 91 年学术会议论文集. 哈尔滨,1991,222~233
- 24 周力嘉等. 机器学习字符识别系统 MLCRS. 模式识别与人工智能,1992,5(2):148~153
- 25 Xu Zhiming, Wang Xiaolong. A New Linguistic Decoding Algorithm for Handwritten Chinese Character Recognition. Journal of Computer Science and Technology,2000,15(6): 597~603
- 26 Pak-Kwong Wong and Chorkin Chan. Online Handwritten Chinese Character Recognition as a Compound Bayes Decision Problem. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998,20(9): 1016~1022



[General Information]

书名 = 计算机自然语言处理

作者 = 王晓龙 , 关毅等编著

页数 = 170

出版社 = 北京市 : 清华大学出版社

出版日期 = 2004

S S 号 = 11376159

D X 号 = 000005245651

U R L = h t t p : / / b o o k . s z d n e t . o r g . c n / b o o k D e t a i l . j s

p ? d x N u m b e r = 000005245651 & d = 98878DDF93EE510A6

8FDBECD15839785

第 1 章	引言
第 2 章	数学基础
2 . 1	初等概率理论
2 . 1 . 1	基本概念
2 . 1 . 2	条件概率与独立
2 . 1 . 3	全概率公式与贝叶斯公式
2 . 1 . 4	随机变量
2 . 1 . 5	多维随机变量
2 . 1 . 6	数学期望与方差
2 . 1 . 7	常用分布
2 . 2	信息论基础
2 . 2 . 1	信息熵
2 . 2 . 2	联合熵和条件熵
2 . 2 . 3	互信息
2 . 2 . 4	相关熵
2 . 2 . 5	语言与熵
2 . 2 . 6	噪声信道模型
2 . 3	粗糙集
2 . 3 . 1	信息系统
2 . 3 . 2	不可分辨关系
2 . 3 . 3	集合近似
2 . 3 . 4	约简
2 . 3 . 5	属性依从
2 . 3 . 6	决策规则合成
2 . 4	小结
第 3 章	汉语自动分词技术
3 . 1	引言
3 . 2	分词规范
3 . 3	常用的分词方法
3 . 3 . 1	正向最大匹配分词
3 . 3 . 2	反向最大匹配分词
3 . 3 . 3	基于统计的词网格分词
3 . 4	歧义的分类和识别
3 . 4 . 1	歧义的分类
3 . 4 . 2	歧义的抽取和消歧
3 . 5	新词的识别
3 . 5 . 1	统计构词能力
3 . 5 . 2	汉字构词模式
3 . 5 . 3	未登录词识别算法
3 . 6	关于分词的若干统计结果
3 . 7	语言单位的统计分布规律 (Zipf 定律)
3 . 8	小结
第 4 章	基于数学统计的语言模型
4 . 1	统计语言模型概述
4 . 2	现有的主要统计语言模型
4 . 2 . 1	上下文无关模型
4 . 2 . 2	N 元文法模型
4 . 2 . 3	N - POS 模型
4 . 2 . 4	基于决策树的语言模型

4 . 2 . 5	动态、自适应、基于缓存的语言模型
4 . 3	数据平滑技术
4 . 3 . 1	数据平滑算法的评价标准
4 . 3 . 2	常见平滑方法
4 . 4	隐马尔科夫模型
4 . 4 . 1	随机过程
4 . 4 . 2	马尔科夫链和马尔科夫性
4 . 4 . 3	马尔科夫模型
4 . 4 . 4	隐马尔科夫模型
4 . 5	最大熵模型
4 . 5 . 1	模型介绍
4 . 5 . 2	模型评价
4 . 5 . 3	最大熵语言建模
4 . 6	小结

第5章 基于语言理解的处理方法

5 . 1	引言
5 . 2	常用的基于语言理解的分类标注体系
5 . 2 . 1	词性分类体系
5 . 2 . 2	词义分类体系
5 . 3	常用的基于语言理解的语法理论
5 . 3 . 1	常用的语法理论
5 . 3 . 2	浅层语法分析技术
5 . 4	语料库多级加工
5 . 4 . 1	语料库的多级加工
5 . 4 . 2	分词
5 . 4 . 3	词性标注
5 . 4 . 4	词性标注的HMM模型
5 . 4 . 5	Viterbi词性标注算法
5 . 4 . 6	语法分析
5 . 4 . 7	概率上下文无关文法
5 . 4 . 8	语料库的应用

5 . 5	小结
-------	----

第6章 音字转换技术

6 . 1	引言
6 . 2	声音语句输入
6 . 2 . 1	声音语句输入的提出
6 . 2 . 2	声音语句的推理
6 . 2 . 3	声音语句输入的系统实现
6 . 3	汉字智能拼音键盘输入
6 . 4	拼音输入的多种表达形式
6 . 4 . 1	拼音助学和提示输入
6 . 4 . 2	简拼快速输入
6 . 4 . 3	用户自定义简拼
6 . 4 . 4	模糊拼音输入
6 . 4 . 5	面向数字键盘的数字拼音输入
6 . 5	拼音预处理
6 . 5 . 1	拼音流的切分
6 . 5 . 2	拼音纠错
6 . 6	音字转换的实现方法
6 . 6 . 1	基于理解的方法
6 . 6 . 2	基于语用统计的方法
6 . 6 . 3	基于模板匹配的方法
6 . 6 . 4	基于上下文关联的音字转换
6 . 7	小结

第7章 自动文摘技术

- 7 . 1 引言
- 7 . 2 文本的内部表示方法
- 7 . 3 基于浅层分析的文摘技术
 - 7 . 3 . 1 建立特征库
 - 7 . 3 . 2 文摘句抽取
- 7 . 4 基于实体分析的文摘技术
 - 7 . 4 . 1 特征提取
 - 7 . 4 . 2 文摘抽取
- 7 . 5 基于话语结构的文摘技术
 - 7 . 5 . 1 基于词汇衔接的文摘方法
 - 7 . 5 . 2 基于话语树的文摘方法
- 7 . 6 文摘系统评测方法
- 7 . 7 关键词自动抽取
- 7 . 8 小结

第8章 信息检索技术

- 8 . 1 信息检索综述
 - 8 . 1 . 1 信息检索的定义与术语
 - 8 . 1 . 2 信息检索系统
 - 8 . 1 . 3 信息检索系统的评价
 - 8 . 1 . 4 信息检索简史
- 8 . 2 信息检索的统计模型
 - 8 . 2 . 1 基于统计的信息检索模型
 - 8 . 2 . 2 布尔模型
 - 8 . 2 . 3 向量空间模型
 - 8 . 2 . 4 概率模型
- 8 . 3 信息检索中的自然语言处理方法
- 8 . 4 文本自动分类技术
 - 8 . 4 . 1 问题的提出
 - 8 . 4 . 2 分类预处理
 - 8 . 4 . 3 向量空间简化方法
 - 8 . 4 . 4 分类方法
- 8 . 5 小结

第9章 文字识别技术

- 9 . 1 引言
- 9 . 2 联机手写体汉字识别的国内外研究概况
 - 9 . 2 . 1 国外研究概况
 - 9 . 2 . 2 国内研究概况
- 9 . 3 联机手写体汉字识别方法综述
 - 9 . 3 . 1 基于统计的识别方法
 - 9 . 3 . 2 基于结构的识别方法
 - 9 . 3 . 3 基于神经元网络的识别方法
 - 9 . 3 . 4 基于机器学习的识别方法
- 9 . 4 典型联机手写体汉字识别系统
 - 9 . 4 . 1 汉王中文手写体汉字识别系统
 - 9 . 4 . 2 豪文中文手写体汉字识别系统
- 9 . 5 联机手写体汉字识别后处理系统
 - 9 . 5 . 1 手写体汉字识别模型
 - 9 . 5 . 2 $P(I|S)$ 估计
 - 9 . 5 . 3 $P(S)$ 估计
 - 9 . 5 . 4 基于词网格的手写体汉字识别的语言学解码方法
 - 9 . 5 . 5 联机手写体汉字识别后处理系统
- 9 . 6 小结