南京理工大学

硕士学位论文

Web文本分类系统中文本预处理技术的研究与实现

姓名：王之鹏

申请学位级别：硕士

专业：计算机应用技术

指导教师：王玲

20090501

摘 要

网络信息量的迅速增长对信息检索提出了更高的要求。在使用搜索引擎时，为了方便用户快速、准确地从网上获取所需的信息，有必要对搜索引擎检索到的大量 Web 页面按内容进行分类。Web 文本挖掘技术是解决上述问题的一种有效的方法。它借鉴数据挖掘的基本思想和理论方法，从大量半结构化、异构的 Web 文档集中发现潜在的、有价值的知识。

Web 文本分类技术是 Web 文本挖掘中的一项重要技术。目前，针对中文网页的分类技术逐渐成为 Web 数据挖掘研究的热点。它的关键技术包括网页清洗、中文分词、特征选择、文本表示以及分类算法。其中网页清洗、中文分词、特征选择和文本表示称为对网页文本的预处理，预处理结果的好坏是决定分类效果的重要因素。本文详细研究了预处理的各个过程并实现了预处理系统。

预处理过程中，特征集的选取对文本分类的训练时间、分类的准确率都有显著的影响。传统的特征选择方法将每一个特征项都单独对待，分别进行特征评估，忽略了特征项之间的相关性、相似性等语义特征。本文在传统特征选择的基础上，引入了基于同义词统计的特征选择方法，在进行特征选择之前，先进行同义词的替换。这样可以进一步降低特征空间的维数，而且通过采用支持向量机算法进行 Web 文本分类实验，并利用分类正确率对分类结果进行评价，与不使用同义词统计的特征选择方法相比，基于同义词统计的特征选择方法具有更好的分类正确率。

关键词：信息检索，Web 文本分类，文本预处理，特征选择，同义词统计

Abstract

The quick growth of web page information has raised a new challenge for information retrieval. In order to make users obtain information on line more quickly and exactly when people use search engine, it is necessary to classify the plentiful web pages according to page content. Web text mining is an efficient method to resolve the problem. It uses the basic thinking and theory of data mining for reference, and discovers potential and valuable knowledge from the half-structural and heterogeneous web pages.

Text categorization is an important technology of web text mining. At the present time, Chinese text categorization has become more and more popular in the research of web data mining. Its pivotal techniques include web page cleaning, word segmentation, features extraction, text expression and text classification. Web page cleaning, word segmentation, features extraction, text expression are called web page preprocessing. The result of preprocessing is an important fact that will affect the result of text categorization. This article researched each part of preprocessing, and implemented a preprocessing system.

During the preprocessing, the effect of features extraction will affect the train time and accuracy of text classification evidently. Traditional feature extraction method treats each feature separately and disregards the semantic feature such as relativity and comparability. This article introduces a feature extraction method based on synonymy statistic. Before the feature extraction, we replace the synonymy with one word first. It can reduce the dimension of feature space. Through the experiment on web text categorization using support vector machine, we evaluated the accuracy of the result. It proved that the accuracy of categorization using feature extraction method based on synonymy statistic was higher than that using feature extraction without synonymy statistic.

Keywords: Information retrieval, web text categorization, text preprocessing, feature extraction, synonymy statistic

声　明

本学位论文是我在导师的指导下取得的研究成果，尽我所知，在本学位论文中，除了加以标注和致谢的部分外，不包含其他人已经发表或公布过的研究成果，也不包含我为获得任何教育机构的学位或学历而使用过的材料。与我一同工作的同事对本学位论文做出的贡献均已在论文中作了明确的说明。

研究生签名：王沛鹏　　2009年6月20日

学位论文使用授权声明

南京理工大学有权保存本学位论文的电子和纸质文档，可以借阅或上网公布本学位论文的部分或全部内容，可以向有关部门或机构送交并授权其保存、借阅或上网公布本学位论文的部分或全部内容。对于保密论文，按保密的有关规定和程序处理。

研究生签名：王沛鹏　　2009年6月20日

1 绪论

1.1 研究背景及意义

随着 Internet 的普及和网上信息的迅速增长，对于网络中杂乱无章的网页信息，用户很难从这些资源中快速、准确地找到自己需要的信息。搜索引擎越来越引起人们的重视。一般来说，搜索引擎是指一种基于 Web 的应用系统，它以一定的策略在 Web 上搜集和发现信息，并对信息进行重新处理和组织，为用户提供便捷的 Web 信息查询服务。从用户的角度看，搜索引擎提供了一个网页界面，让用户通过浏览器提交一个或者多个词语的集合，通过相关查询处理，返回给用户一个可能与用户的输入内容相关的信息列表。这个列表中的每一条目代表一个网页，每个条目至少包含三个元素：网页标题、网页的 URL 地址和网页内容的摘要。

尽管搜索引擎能够帮助用户过滤网络中的部分信息，使用户可以利用搜索引擎找到与特定查询词相关的文档，但却存在一个困扰用户的问题，就是给定一个确定的关键词查询，通常会返回一个庞大的相关网页集列表，而其中大部分网页并不是用户所需要的。因此需要一种方法来更准确地确定某个文档是否能满足用户的检索需求。

如何决定一个网页是否与检索需求相关是件很困难的事情，因为它是一个主观性的判断。确定一个文档相关性的问题包括：页面所呈现的格式是否与用户的需求一致；页面内容体现的学术等级是否满足需求；理解页面的难易程度是否与用户的能力相符；等等。衡量文档的相关性往往需要综合考虑文档的主题和形式上的特征。

目前搜索引擎提供两种信息查询方式：分类浏览和关键词检索。分类浏览一般是基于网站的分类目录。它浏览的对象是网站内的页面，目录分类的质量较高，检索效果好；但是成本高、信息更新慢、维护的工作量大。关键词检索的对象不是网站，而是符合条件的网页。关键词检索信息量大、更新及时、不需要人工干预；但是返回信息过多，质量太低。

目前，很少有搜索引擎提供对网页的分类浏览或检索，其原因之一是由人工进行网页的分类几乎是不可能的。如果能够实现网页的自动分分类，就可以实现网页索引和检索的分类主题一体化，搜索引擎就能够兼有分类浏览、检索和关键词检索的优点，帮助用户迅速的判断返回的结果是否符合自己的检索要求。例如在关键词检索中用熊猫作为检索词，返回的结果中作为动物的熊猫、作为一种病毒的熊猫和作为一种电子产品的熊猫等内容是夹杂在一起的，用户要对结果进行分析判断，才能确定哪些是自己需要的信息。

网页自动分类根据在搜索引擎中应用的时机分为在对网页数据进行索引的时候实施和在搜索引擎返回检索结果之后实施。如果采用了自动分类技术，就可将不同的内容

分到不同的类目中去，从而节省用户的判断时间，提高检索效率。目前关于文本分类的研究多是基于普通文本来进行的，网页分类方面的研究较少。整体来看，网页分类研究还处于全面探索阶段，技术还不够成熟，尤其是针对中文的研究更是刚刚起步。因此，在网页类别的确定、特征项的选择等方面都存在很大的困难。

1.2 搜索引擎的发展

在互联网发展初期，网站数目相对较少，查找网上的信息比较容易。然而随着互联网迅速的发展，网络资源迅速膨胀，用户很难快速准确地找到自己所需的资料，这时为满足用户信息检索需求的专业搜索网站便应运而生了。

最早的搜索引擎是 1990 年由蒙特利尔大学学生 Alan Emtage 发明的 Archie。Archie 工作原理与现在的搜索引擎很接近，它依靠脚本程序自动搜索网上的文件，然后对文件进行索引，供使用者查询。美国内华达 System Computing Services 大学于 1993 年开发了另一个与之非常相似的搜索工具，不过此时的搜索工具除了索引文件外，已能检索网页。

随着互联网规模的不断扩大，检索所有新出现的网页变得越来越困难，这也促进了搜索引擎的发展。根据搜索引擎在不同时期的研究重点和服务性能的不同，搜索引擎的发展分为三代[1]。

第一代搜索引擎出现于 1994 年，即 Archie。这类搜索引擎索引的网页较少，通常少于 100 万个，很少更新网页并刷新索引，而且其检索速度非常慢，一般都要等待 10s 甚至更长的时间。其实现技术主要使用较为成熟的 IR（Information Retrieval）、网络、数据库等技术，相当于利用一些已有技术实现的一个 WWW 上的应用。

大约在 1996 年出现了第二代搜索引擎系统，第二代搜索引擎大多采用分布式方案(多台计算机协同工作)来提高数据规模、响应速度和同时可以访问的用户数量，它们一般都保持一个大约 5000 万网页的索引数据库，每天能够响应 1000 万次用户检索请求。

自 1998 年到现在，是搜索引擎空前繁荣的时期，一般称这一时期的搜索引擎为第三代搜索引擎。第三代搜索引擎的发展有如下几个特点：

（1）索引数据库的规模不断增大，一般的商业搜索引擎的数据库都有几千万甚至上亿个网页的规模。

（2）除了一般意义上的信息检索以外，开始出现主题搜索和地域搜索。很多小型的垂直门户网站开始使用这种技术。

（3）由于搜索结果返回网页数量非常大，检索结果相关度评价成为研究的重点。这方面的研究又可以分为两类：一类是对超文本链接的分析，在这方面 Google 和 IBM 的 Clever 系统做出了很大的贡献；另一类是用户信息的反馈，DirectHit 系统采用的就是这种方法。

（4）开始使用自动分类技术。Northern Light 和 Inktomi 的 Directory Engine 都在一定程度上使用了自动分类技术。

第三代搜索引擎的研究重点主要是对搜索结果进行处理，提高用户查询的效率。目前，针对搜索结果的处理方法主要有相关搜索、搜索结果重组、相近搜索、搜索条件延伸等。相关搜索根据搜索条件，列出与用户查询字符串类似的关键词。由于用户在进行搜索时不能准确地表达自己的搜索需求，因此相关搜索功能就会提供一些相关词给用户选择，以进一步靠近用户的搜索需求，达到提高查询准确度的目的。搜索结果重组避免了来自同一个网站的内容重复显示的情况，通过只允许一个网站最多只能有一个页面出现在排名靠前的搜索结果中，保证用户可以有更多更好的选择机会。相近搜索方法使搜索引擎将与用户指定的网页相似的网页也列出来，方便用户更深入的查询相关信息。搜索条件延伸方法让用户可以在查询一个关键词的基础上，查询由此关键词延伸而来的其他词。上述的几种搜索结果处理方式在一定程度上方便了用户查询信息，但是没有从根本上解决搜索结果过多的问题，用户在查询信息时依然要在大量的搜索结果中逐条查找。

因此，把网页自动分类技术应用于搜索引擎检索结果的处理中，通过对检索结果进行自动分类，可以在两个方面帮助用户。其一，如果分类结果中某类正好是用户所需要的，那么用户不必浏览检索结果就可以直接找到需要的信息。其二，如果分类结果中没有一类是符合用户需求的，那么用户可以通过分类情况来了解检索结果内容、结构等方面的情况，帮助用户改进检索策略。

根据分类知识的获取方法不同，可以将网页自动分类系统分为两种类型：基于知识工程的分类系统和基于统计的分类系统。基于知识工程的方法主要依赖语言学知识，需要编制大量的推理规则作为分类知识，实现相当复杂，而且其开发费用相当昂贵。目前研究比较多的是基于统计的网页自动分类技术，它忽略文档的语言学结构，从网页内容中抽取最能代表网页内容的特征词作为特征向量，形成向量空间，然后根据向量之间的相似性，使用各种算法实现文档的自动分类。

1.3 Web 文本分类的研究现状

在 Web 出现之前，人们已经对文本自动分类问题进行了大量的研究，形成了文本自动分类技术。随着 Web 海量的文本信息的增加，文档自动分类技术的处理对象从普通的文档扩展到了 Web 文本。很显然，文本自动分类技术也成为 Web 文本分类技术的基础。

国外对于文本分类技术的研究开展较早。50 年代末，H. P. Luhn 对该领域进行了开创性的研究，提出了基于词频统计思想的文本自动分类方法。1960 年，Maron 发表了关于自动分类算法的第一篇论文，随后以 K. Spark，G. Salton 以及 K. S. Jones 等人为代表

的众多学者也在这一领域进行了卓有成效的研究工作[2]。目前国外的文本分类研究已经从实验性阶段进入实用化阶段，并在邮件分类，电子会议等方法取得了广泛的应用。近年来，国外开发的一些文本自动分类系统主要有美国卡内基梅隆大学的 Rainbow/Libbow 文本自动分类系统[3]，AT&T 实验室的基于非确定性分类技术实现的自动分类系统、美国斯坦福大学计算机系的基于很少语料词汇的层次自动分类、美国 Just Research 公司的基于信息熵和贝叶斯理论实现多类的自动分类、美国马萨诸塞州大学计算机系的针对文本库的自动分类系统、德国多特蒙德大学计算机系的基于向量空间模型的自动分类系统等等[4]。

自从国内提出文本分类的概念以来，文本分类技术在国内得到了长足的发展。然而和国外的发展状况相比，发展水平仍相对滞后。一方面由于国内起步较晚，对中文文本分类的研究是从上世纪 90 年代后期才开始的；另一方面则由于国内的工作主要针对的是中文文本。由于汉语本身的特点，中文文本分类和英文文本分类有很多不同。另外，在不同的语言的研究工作中，句法分析和语义分析所占的比例是不同的。在英语中，句法分析比语义分析的比例要大，而在汉语中，语义分析在汉语研究中起着举足轻重的作用，所占的比例比句法分析要大得多。这使得在中文文本分类中，通过句法分析等基于语法的手段把握文本的内容变得更加困难。国内的文本分类的发展过程大致经历了三个阶段：国外研究成果的引进阶段、分类技术完善阶段以及面向汉语分类技术的发展阶段；国内文本分类技术的发展方向则有基于外延的分类方法和基于概念的分类方法之分。国内对于文本自动分类的研究主要集中在复旦大学、中科院计算所、北京大学、清华大学等。由于中文与英文存在较大的差异，不能照搬国外的研究成果，中文文本分类的研究基本上是在英文文本分类的技术的基础上，结合中文文本的特点，继而形成中文文本分类的研究体系。

1981 年，侯汉清教授对计算机在文本分类工作中的应用作了探讨和阐述[5]。此后，我国陆续研究产生了一些文本分类系统，其中具有代表性的有上海交通大学研制的基于神经网络算法的中文自动分类系统，清华大学的自动分类系统等等。同时在不同的分类算法方面也展开了广泛的研究和实现，中科院计算所的李晓黎、史忠植等人应用概念推理网进行文本分类[6]。复旦大学和富士通研究中心的黄萱菁、吴立德等人研究了独立语种的文本分类，并以词汇和类别的互信息为评分函数，考虑了单分类和多分类[7]。上海交通大学的刁倩、王永成等人结合词权重和分类算法进行分类，在基于 VSM 的封闭测试实验中取得了较好的效果[8]。

虽然文本自动分类技术可以为 Web 文本分类提供较好的技术基础，并已经得到了广泛应用，但是 Web 文本和普通文本的分类又有所不同，如：

（1）网页信息比文本信息更开放，风格不固定；

（2）网页的设计比较随意，通常包含大量的广告、程序源代码、HTML 标记、设

计人员的注释以及版权声明等无关信息，这些“噪音”降低了分类的查准率；

（3）网页分类的类别比文本分类的类别更多，为了便于用户浏览和选择，一般要求类别有层次关系；

（4）网页的分类体系随着信息的变化会做一些变动，并且很难有一个统一的标准等等。

对 Web 文本的处理也不同于普通文本，面临更多的问题，所以，Web 文本分类比普通文本的分类更复杂、更困难，需要针对其特点进行研究。

目前，一些比较成熟的文本分类算法已经被应用到了 Web 文本分类中。北京科技大学的唐菁等人采用向量空间的距离测度分类算法，构建了一个适用于现代远程教育的文本挖掘系统[9]。它能充分利用 Web 站点（远程教育站点）上的大量文本信息，更好地服务于远程教育。该系统的查准率和查全率都比较高，表现出了较好的性能。

中科院软件研究所、北京邮电大学模式识别与智能实验室、微软亚洲研究院等多家研究机构也都进行着相关的理论研究。

目前主要的分类算法有贝叶斯分类算法、KNN 分类算法、SVM 分类算法、决策树分类算法和神经网络分类算法等等，近些年还出现了基于粗糙集理论的文本分类算法和一些结合多种方法的混合分类方法。Web 数据挖掘在国内逐渐引起人们的关注 。

1.4 本文的主要研究内容

Web 文本预处理是在进行 Web 文本分类之前首先做的工作。本文主要研究了 Web 文本预处理对自动分类系统分类效果的影响，对预处理过程的各个阶段进行了深入的研究。

本文所做的工作主要有以下几点：

（1）研究比较了当前常用的几种分类算法：KNN 算法、贝叶斯算法、决策树算法和支持向量机算法，并采用支持向量机算法作为本文的分类算法。

（2）分析了 Web 文本预处理的一般过程，详细阐述了网页清洗、分词、去停用词、特征选择、文本向量表示的实现技术和实现方法。

（3）考虑到特征选择结果的好坏对分类效果的重要影响，本文重点对特征选择的方法进行了阐述，在传统的特征选择方法的基础上引入了基于同义词统计的方法，即在对特征项进行函数评估之前，首先对文本中的同义词进行合并处理，提高特征项的函数评估值。

（4）用 C++语言实现了 Web 文本预处理系统，并用支持向量机方法进行了分类实验，利用查全率、查准率、F1 值以及分类正确率等评价措施对分类结果进行了评价。比较了传统特征选择方法与基于同义词统计的特征选择的方法对分类效果的影响。

1.5 本文的组织结构

本文的组织结构如下：

第一章 绪论。本章介绍了 Web 文本分类在搜索引擎性能优化中的作用和国内外的研究现状，说明了论文研究的主要内容和组织结构。

第二章 Web 文本挖掘。本章介绍了 Web 文本挖掘的基本概念、Web 文本分类系统的主要任务和分类过程。介绍了当前常用的几种文本分类算法。

第三章 Web 文本预处理技术。本章对 Web 文本预处理的各个阶段的实现技术进行了详细的阐述，说明了利用同义词统计进行特征选择的具体步骤。

第四章 支持向量机在 Web 文本分类中的应用。介绍了 LIBSVM 2.88 分类器，以及 SVM 在文本分类中优势。

第五章 预处理系统的实现及实验结果分析。本章具体实现了预处理系统各个模块。并对实验结果进行了分析比较。

第六章 结束语。对本文所做的工作和取得的成果进行了总结，并阐述了下一步的主要工作。

2 Web 文本挖掘

2.1 Web 文本挖掘的特点

Web 页面用 HTML 写成，由多种媒体对象和指向其他文档的指针组成。Web 上的数据主要有以下特点[10]：

（1）从数据库的角度出发，Web 网站上的信息可以看作一个更大、更复杂的数据库。每一个站点是一个数据源，每个数据源都是异构的，这就构成了一个巨大的异构数据库环境。

（2）从数据模型的角度出发，半结构化是 Web 上数据的最大特点。Web 上的数据与传统的数据库中的数据不同，Web 上的数据非常复杂，没有特定的模型描述，每一个站点的数据都是独立设计的，并且数据本身具有自述性和动态可变性，是一种非完全结构化的数据。

（3）从数据更新的角度出发，Web 上的数据是一个动态性极强的信息源，不仅增长的速度非常快，而且信息也在不断快速地更新，各站点的链接信息和访问记录的更新非常频繁。

（4）从用户的角度出发，Web 面对的是一个庞大的用户群体，每个用户都有不同的背景、兴趣和使用目的，大部分用户不清楚信息网络结构，极易在信息的海洋中迷失方向。

Web 数据挖掘就是指在 WWW 上挖掘潜在的、蕴藏的信息及有用的模式[11]。Web 挖掘一般分为三大类：

（1）Web 内容挖掘。它是指从网页内容出发，从中获取潜在的、有价值的知识，以实现 Web 资源的高效率自动检索，提高资源的利用率。通常可以分为 Web 文本挖掘和 Web 多媒体挖掘。Web 文本挖掘是指对文档的内容进行总结、分类、聚类、关联分析等；Web 多媒体挖掘是指从 Web 多媒体数据如 Web 音频、视频、图像中抽取潜在信息的过程。

（2）Web 结构挖掘。它是指从 WWW 组织结构及其链接关系中获取数据的过程，主要针对的是外部文档的超链接结构。Web 结构挖掘的目的是发现 Web 的结构和页面的结构及其蕴含在这些结构中的有用模式；对页面及其链接进行分类和聚类，找出权威页面；发现 Web 文档自身的结构，这样更有助于用户的浏览，也利于对网页进行比较和系统化。

（3）Web 使用挖掘。它的研究对象是 Web 使用数据或 Web 日志。Web 日志是一系列网页访问数据。它通过从 Web 的访问信息的挖掘获取信息，分析用户和 Web 网页之间交互结果，包括点击次数、以及一组相关网页间的数据交换等。对服务器进行分析，

挖掘的结果可以帮助改善网站的设计。分析客户的点击序列，可以发现用户的信息，这些信息可以帮助实现网页的预存取和缓存。

Web 文本挖掘是指对 Web 页面内容、页面之间的结构、用户访问信息等各种 Web 数据应用数据挖掘的方法，发现有用的知识来帮助人们从大量的 Web 文档集中发现隐含的模式[12]，主要包括文本摘要、文本分类、文本聚类、关联分析等。

（1）文本摘要

文本摘要是从文章中抽取关键信息，以简洁的形式，对文本内容进行摘要和描述。用户不需要阅读全文就可以从文本的摘要中了解文章的大致内容。例如一些搜索引擎返回给用户的查询结果都是文本的摘要。

（2）文本分类

文本分类是指根据预先确定的主题类别，为文本集合中的每一个文本都确定一个类别的过程。分类是一个有指导的机器学习问题，分类的目的是让机器学会一种分类规则，该规则能将 Web 文本映射到一个或多个已经存在的主题类别中，这样用户就能够快速、准确地查找自己需要的文本。文本分类一般分为训练和分类两个阶段。

（3）文本聚类

文本聚类与分类虽然都是将文本归类，但实现方法不同，分类是将文本归类到预先已经存在的类别中，而聚类是在没有预先定义类别的情况下，根据文本的内容，将文本分成若干类，要求同一类文本间的相似度尽可能大，不同类的文本间的相似度尽可能小，是一种无指导的机器学习过程。

（4）关联分析

关联分析是指从文本集合中找出不同词语之间的关系。它揭示了数据项间的未知的依赖关系，根据所挖掘的关联规则，可以从一个数据对象的信息来推断另一个数据对象的信息。

2.2 Web 文本分类系统

简单地说，Web 文本自动分类系统的任务就是：在给定的分类体系下，根据网页的主题内容自动地确定文本关联的类别。从数学角度来看，Web 文本分类是一个映射的过程，它将未标明类别的网页文本映射到已有的类别中，该映射可以是一一映射，也可以是一对多的映射，因为通常一篇文本可以同多个类别相关联。用数学公式表示如下：

$f: A \rightarrow B$

其中，A 为待分类的文本集合，B 为分类体系中的类别集合。文本分类的映射规则是分类系统根据已有的若干类别的样本数据信息，总结出分类的规律性，并建立的判别公式和判别规则。然后在遇到新文本时，根据总结出的判别规则，确定文本的类别，是一种典型的有监督的文本分类方法。

Web 文本自动分类的过程如图 2.2.1 所示：

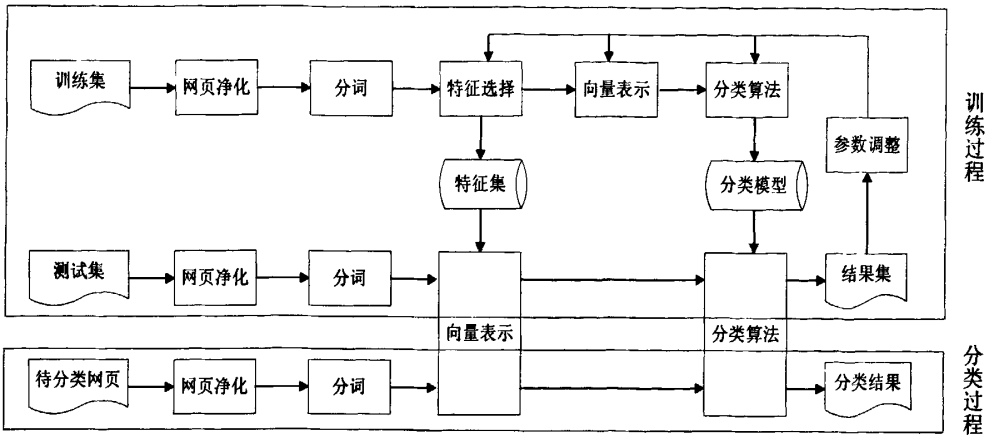

图 2.2.1 Web 文本自动分类的一般过程

具体过程如下：

（1）训练过程

使用已经标好类别的网页文档集作为训练集，首先对训练集进行网页净化，去除页面中的广告、导航栏等噪音信息；然后对得到的正文文本进行分词，形成用一系列特征词表示的文本；通过特征选择方法选取一部分特征词，形成训练集的特征集合；计算特征集中每一个特征项在训练集中权重，将文本表示成用权重表示的向量形式；最后用分类算法对训练集进行分类，形成分类模型。

同样使用已经标好类别的网页文档集作为测试集，对测试集进行网页净化和分词处理后，利用前面得到的特征集，将测试集中的文本表示成向量形式；然后利用前面得到的分类模型进行分类测试，根据分类结果对特征选择、分类算法等步骤进行调整，直到得到较好的分类效果。

（2）分类过程

对未标明类别的网页文档集进行网页净化和分词后，利用经过训练过程生成的分类器进行分类，得到分类结果。

Web 文本自动分类技术的一个重要应用就是服务于搜索引擎。通常搜索引擎主要分为三部分：网络爬虫（Crawler）、文本自动分类系统（Classifier）和面向用户的文档检索系统[13]。文本自动分类系统的作用就是对网络爬虫收集到的网页进行处理、分类并组成文档索引库，供文档检索系统使用。

2.3 常用的文本分类算法

2.3.1 KNN 分类算法

KNN 算法很早就已经被广泛研究并用于文本分类问题，是一种基于实例的文本分类方法[14][15]。给定一个要分类的测试样本，KNN 算法计算它与训练样本集中每个文本的相似度，依文本相似度从中找出 k 个最相似的训练文本，然后在此基础上给每一个文本类打分，分值是 k 个训练文本中属于该类的文本与测试文本之间的文档相似度之和，按分值进行排序，依分值指定测试文本的类别。

KNN 中的决策规则可写作：

$$y(x,C_j)=\sum sim(x,d_i)y(d_i,C_j)-b_j \quad (d_i \in KNN) \tag{2.3.1}$$

其中 $y(d_i,C_j)\in\{0,1\}$，当 d_i 属于 C_j 时取 1；当 d_i 不属于 C_j 时取 0；$sim(x,d_i)$ 测试文档 x 和训练文档 d_i 的相似度；b_j 是决策的阈值。两个文本的相似度用两个向量的夹角余弦表示：

$$sim(x,d_i)=\frac{\sum_{k=1}^{m} w_k \times w_{ik}}{\sqrt{(\sum_{k=1}^{m} {w_k}^2)(\sum_{k=1}^{m} {w_{ik}}^2)}} \tag{2.3.2}$$

其中，x 为新文本的特征向量，d_i 为训练集中文本的向量，m 为特征向量的维数，w_k 为向量第 k 维的权重，w_{ik} 为文本 d_i 中第 k 维的权重。

KNN 算法只需存储训练文本集的特征向量空间即可，不需要训练，分类性能稳定。实验表明，KNN 在只有较少的训练样本的情况下仍可取得较好的分类效果。但 KNN 算法的分类速度很慢，需要占用大量的计算机资源。

2.3.2 朴素贝叶斯算法

朴素贝叶斯（Naive Bayes，NB）算法以贝叶斯定理为理论基础，是一种在已知先验概率和条件概率的情况下的分类方法[15]。NB 算法应用贝叶斯公式，将类型的先验概率转化为后验概率，从而判定文档的类别。其优势在于训练和分类的速度快，在实际应用中表现出不错的分类性能。

假设 d_i 为一任意文本，它属于类别 $C=\{C_1,C_2,...,C_m\}$ 中的某一类 C_j。NB 算法的公式如下：

$$p(C_j \mid d_i)=\frac{p(C_j)p(d_i \mid C_j)}{p(d_i)} \tag{2.3.3}$$

$$p(d_i)=\sum_{j=1}^{m} p(C_j)p(d_i \mid C_j) \tag{2.3.4}$$

根据公式（2.3.3）计算文档d_i属于C_j的概率，概率最大的那个类别就是d_i所在的类，即：

$$d_i \in C_j，当且仅当 p(C_j \mid d_i) = \max_{j=1}^{m}\{p(C_j \mid d_i)\} \tag{2.3.5}$$

由公式(2.3.3)、(2.3.4)知，用 NB 算法进行文本分类就是计算$p(C_j)$和$p(d_i \mid C_j)$，计算$p(C_j)$和$p(d_i \mid C_j)$的过程就是建立分类模型的过程。

在实际的应用过程中，为避免$p(C_j)$=0，采用拉普拉斯概率估计：

$$p(C_j) = \frac{1+|D_{Cj}|}{|C|+|D_C|} \tag{2.3.6}$$

其中，$|C|$为训练集中类的数目，$|D_{Cj}|$为训练集中属于类C_j的文档数，$|D_C|$为训练集中包含的总文档数。

2.3.3 决策树算法

决策树分类方法将搜索空间划分为一些矩形区域，根据元组落入的区域对元组进行分类[16][17][18]。给定一个数据库$D=\{t_1,t_2,...,t_n\}$，其中$t_i=<t_{i1},...,t_{ih}>$，数据库模式包含下列属性$\{A_1,A_2,...,A_h\}$。同时给定类别集合$C=\{C_1,C_2,...,C_m\}$。对于数据库 D，决策树是指具有下列性质的树：

（1）每个内部节点都被标记一个属性A_i。

（2）每个弧都被标记一个谓词，这个谓词可应用于相应父结点的属性。

（3）每个叶结点都被标记一个类C_j

决策树分类方法包括两个步骤：

（1）决策树归纳。利用训练数据构建一棵决策树。

（2）对每个元组$t_i \in D$，应用决策树确定元组的类别。

基于信息论构建决策树的 ID3 技术是一种流行的决策树算法，它的基本策略是首先选择具有最高信息增益的属性作为分裂属性。一个属性值的信息量是与发生概率相关的。

设$C_{j,D}$是D中C_j类的元组的集合，$|D|$和$|C_{j,D}|$分别是D和$C_{j,D}$中元组的个数。对D中的元组分类所需的期望信息为：

$$Info(D) = -\sum_{j=1}^{m} p(C_j)\log_2(p(C_j)) \tag{2.3.7}$$

假设按属性 A 划分D的元组，根据训练数据得到v个不同的值$\{a_1,a_2,...,a_v\}$，根据这v个值将D划分为v个子集$\{D_1,D_2,...,D_v\}$，D_j包含D中元组，在 A 上具有值a_j。则基于按 A 划分对D的元组分类所需要的期望信息为：

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j) \tag{2.3.8}$$

$\frac{|D_j|}{|D|}$为第 j 个划分的权重。

信息增益定义为：

$$Gain(A) = Info(D) - Info_A(D) \tag{2.3.9}$$

选择具有最高信息增益 $Gain(A)$ 的属性 A 作为节点的分裂属性。这等价于按能做“最佳分类”的属性 A 划分，使得完成分类还需要的信息最小（即最小化 $Info_A(D)$）。

2.3.4 SVM 分类算法

支持向量机（Support Vector Machines，SVM）是 Vapnik 在 1995 年提出的一种基于统计学习理论的新型通用学习方法[19][20][21][22]，它建立在统计学习理论的 VC 维理论和结构风险最小化原理的基础上[23]，根据有限样本信息在模型的复杂性和学习能力之间寻求最佳折中，以获得更好的学习能力。其基本思想是首先通过非线性变换将输入空间映射到一个高维特征空间，然后在这个新空间中求取最优分类超平面，这种非线性变换是通过定义适当的内积函数（核函数）来实现的。

（1）数据线性可分的情况

首先考虑只有两类的情况，为了更有助于理解，可以将样本看成是空间中的点，通常希望利用最优超平面，将这些样本区分开来，使得两样本点分别在超平面的两侧，同时使分开的两类样本点距离分类超平面最远。从图 2.3.1 中可以看出，当数据为线性可分的时候，一定存在平行的两个超平面 H_1 和 H_2，将不同类别的数据最大限度的分开，即 H_1 和 H_2 具有最大分类间隔，边缘数据点自然地会落在平面上，而最优超平面就是要求超平面不但能将两类正确分开，而且要使分类间隔最大。

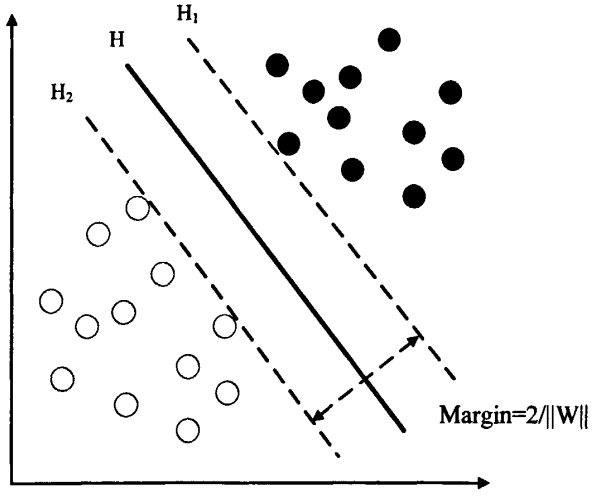

图 2.3.1 数据线性可分的情况

设训练样本为 $\{x_i, y_i\}, i=1,...,N$，N 为样本数，$x_i \in R^n$ 为样本点，$y_i \in \{-1,1\}$ 为类别。分类超平面可以记作：

$$wx+b=0 \tag{2.3.10}$$

其中，w 是权重向量，即 $w=\{w_1, w_2, ..., w_n\}$，b 是标量，通常称作偏倚。

在线性可分的情况下，存在两个平行的超平面将这两类样本完全分开，这两个超平面可表示为

$$\begin{cases} wx_i + b \geq 1 & y_i = +1 \\ wx_i + b < -1 & y_i = -1 \end{cases} \tag{2.3.11}$$

可将其统一描述为下面的形式：

$$y_i(wx_i + b) - 1 \geq 0 \quad i = 1,..., N \tag{2.3.12}$$

此时分类间隔等于 $2/\|w\|$，使分类间隔最大等价于使 $\|w\|$ 最小。满足公式（2.3.10）且使 $2/\|w\|$ 最大的超平面就叫做最优分类超平面（Maximum Marginal Hyperplane，MMH）。落在超平面 H_1 或 H_2 上的训练元组使式（2.3.12）等于零成立，称为支持向量。

利用 Lagrange 优化方法可以把上述最优分类超平面问题转化为其对偶问题，即在约束条件 $\alpha_i \geq 0$，$i=1,...,N$ 和 $\sum_{i=1}^{N} \alpha_i y_i = 0$ 下，对 α_i 求解下列函数的最大值：

$$\max(-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j (x_i \bullet x_j) + \sum_{i=1}^{N}\alpha_i) \tag{2.3.13}$$

解上述问题后得到的最优分类函数为：

$$f(x)=sign(\sum_{i=1}^{N}\alpha_i^* y_i(x\bullet x_i)+b^*) \tag{2.3.14}$$

其中，x_i是支持向量，α_i^*是对应的 Lagrange 乘子，b^*是分类阈值，可以用任一个支持向量求得，或通过两类中任意一对支持向量取中值求得。

（2）数据非线性可分的情况

前面描述的是数据线性可分的情况，在现实世界中，很多分类问题都是非线性可分的，如图 2.3.2 所示。对于非线性可分的情况，SVM 的基本思想是：通过某种非线性变换将原样本空间映射到一个高维的特征空间中，在这个高维特征空间中构造最优分类超平面。

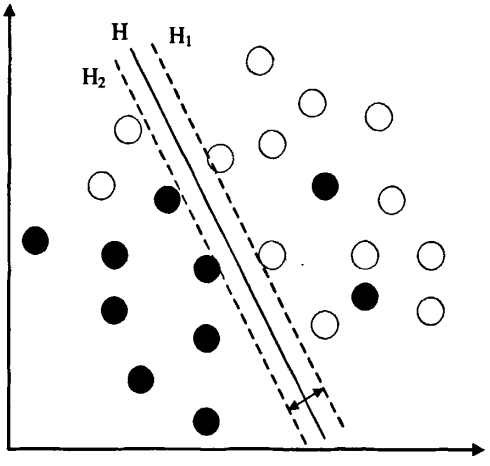

图 2.3.2 数据非线性可分的情况

这种非线性变换是通过定义适当的核函数实现的。令：

$$K(x_i,x_j)=\phi(x_i)\cdot\phi(x_j) \tag{2.3.15}$$

用核函数$K(x_i,x_j)$代替最优分类面中的点积$x_i^T x_j$，就相当于把原特征空间变换到了某一新的特征空间，此时优化函数变为：

$$\max(-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j K(x_i,x_j)+\sum_{i=1}^{N}\alpha_i) \tag{2.3.16}$$

最优分类函数也变为：

$$f(x)=sign(\sum_{i=1}^{N}\alpha_i^* y_i K(x_i,x)+b^*) \tag{2.3.17}$$

一般常用的核函数如下：

① 多项式核函数：

$$K(x,y)=[a(x\bullet y)+b]^q \tag{2.3.18}$$

② 径向基函数(RBF)：

$$K(x,y)=\exp(-\gamma\|x-y\|^2) \tag{2.3.19}$$

③ 双曲正切核函数：

$$K(x,y)=\tanh[a(x\bullet y)+b] \tag{2.3.20}$$

其中，a、b、q、γ都是参数，根据需要进行设置。

由于核函数的重要性，如何去构造、选择核函数及参数成为人们关注的问题。通常的做法是找出样本集分布特点与最优分类器之间可能的对应关系的一些先验知识选择分类器类型和参数；或直接构造新的类型，可以预先确定或在训练过程中逐步优化。

（3）多类问题

SVM 本质上是一个两类分类器，但实际上我们还会遇到多类分类的问题，用数学语言可以把多类分类问题描述如下：

根据给定训练集

$$T=\{(x_1,y_1),...,(x_l,y_l)\}\in(\mathrm{X}\times\mathrm{Y})^l$$

其中，$x_i\in\mathrm{X}=R^n, y_i\in\mathrm{Y}=\{1,2,...,M\}, i=1,2,...,l$，寻找一个决策函数 $f(x):\mathrm{X}=R^n\rightarrow\mathrm{Y}$。

由此可见，求解多类分类问题，实质上就是找到一个把R^n上的点分成 M 部分的规则。常用的 SVM 多值分类器构造方法有[24][25]：

① 一对多的方法：在第 k 类和其他 k-1 类之间构建超平面。在这种方式下，系统仅构建 k 个 SVM，每一个 SVM 分别将某一分类的数据从其他分类的数据中鉴别出来。对第 i 个 SVM 用第 i 个类中的训练样本作为正训练样本，而将其他的样本作为负训练样本。

② 一对一方法：它也是基于两类问题的分类方法，不过这里的两类问题是从原来的多类问题中抽取的。对 M 类问题，就有$(M-1)M/2$个两类分类器，对 M 个分类的训练集进行两两区分。分类器的数目常常要比一对多的方法得到的分类器的数目大很多，但是它的每一个分类问题的规模却小了很多。尽管如此，如果 M 太大，$(M-1)M/2$就会非常大，这时这个方法就会比一对多的方法慢许多，其原因可以认为是实际上可以能不需要求解$(M-1)M/2$个两类分类机。测试时，常用投票法，得票最多的类为测试样本所属的类。

③ SVM 决策树方法：将 SVM 和二叉决策树结合起来，构成多分类器。该方法的缺点是如果在某个节点上发生了分类错误，则会把分类错误延续到该节点的后续下一级节点上。

④ 多类 SVM：通过改写 SVM 二值分类中优化的目标函数，使得其满足多值分类的需要。多类 SVM 法的一次规划形式复杂，求解计算量大，一般较少采用。

总体来说，上述四种多类分类方法各有利弊，可以针对实际问题的不同条件限制，选择不同的方法。但是，应用比较广泛的还是第一种方法。

3 Web 文本预处理技术

Web 文本预处理是整个分类系统中非常关键的一步，任何原始数据在计算机中都必须采用特定的数学模型进行表示。在对中文 Web 文本进行分类的过程中，包括几个关键步骤：网页预处理、分词、特征提取、权重计算、向量表示，这些关键技术的研究和实现对最终的分类算法都有一定程度上的影响。

3.1 DOM 树结构

DOM 树结构是研究网页布局结构的主要依据，把半结构化的 HTML 页面转化为结构化的 DOM 树结构，可以更好的对网页进行分析研究。

按照 W3C 的定义，DOM（Document Object Model，文档对象模型）是一个允许程序或者脚本动态地存取和更新 HTML/XML 文件内容、结构以及风格的接口和平台[26]。DOM 目前主要由两部分组成：DOM 核和 DOM 扩展。DOM 核主要定义了处理 XML 文件所需的功能；DOM 扩展定义了处理 HTML 文件所需的功能。

DOM 是一种用于 HTML 和 XML 文档的应用程序编程接口（API）。使用 DOM 模型，程序员可以构造文档，增加、修改或删除元素和内容，HTML 中的任何内容都可以使用 DOM 模型进行存取、修改、删除或增加。DOM 是由一组对象和存取、处理文档对象的接口组成。下面介绍常用的几种对象：

（1）文档（Document）：DOM 的文档是由分层的节点对象构成，这些节点对象构成一个 HTML 页面。文档是一个节点，该节点只有一个元素，这个元素就是它自己。文档接口表示整个 HTML 文档，从概念上讲，它是文档树的根，提供对文档数据的存取。

（2）节点（Node）：节点是一般类型，它涉及一个文档中存在的所有对象。

（3）元素（Element）：在细读一个文档时，最常碰到的对象就是元素，元素是除文本之外的大多数对象。元素是从节点类型推导出来的。元素包含属性，而且可以是另一个元素的父类型。

（4）文本节点（Text Node）：文本节点处理文档中的文本。

（5）属性（Attribute）：属性是元素的基本属性，因此它们不是元素的子节点。即使它们是从一般节点类型推导出来，它们的行为也与其它节点的行为不同。

根据 DOM，HTML 文档中的每个成分都是一个节点。DOM 是这样规定的：整个文档是一个文档节点；每个 HTML 标签是一个元素节点；包含在 HTML 元素中的文本是文本节点；每一个 HTML 属性是一个属性节点；注释属于注释节点 。HTML 文档中的所有节点组成了一个文档树（或节点树）。HTML 文档中的每个元素、属性、文

本等都代表着树中的一个节点。树起始于文档节点，并由此继续伸出枝条，直到处于这棵树最低级别的所有文本节点为止。

图 3.1.1 表示一个文档树（节点树）：

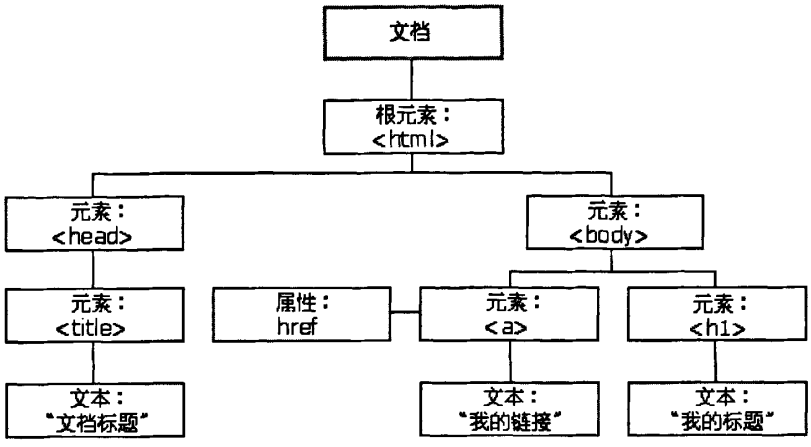

图 3.1.1 文档树结构

对于图 3.1.2 所示的 HTML 文档：

```
<HTML>
    <HEAD>
        <TITLE>文档标题</TITLE>
    </HEAD>
    <BODY>
        <H1>我的标题</H1>
        <P>文档内容</P>
    </BODY>
</HTML>
```

图 3.1.2 一个简单的 HTML 文档

上面所有的节点间都存在关系。

除文档节点之外的每个节点都有父节点。例如，<HEAD>和<BODY>的父节点是<HTML>节点，文本节点“文档内容”的父节点是<P>节点。

大部分元素节点都有子节点。比方说，<HEAD>节点有一个子节点：<TITLE>节点。<TITLE>节点也有一个子节点：文本节点“文档标题”。

当多个节点有相同的父节点时，它们就是兄弟节点。比方说，<H1>和<P>是兄弟节点，它们的父节点均是<BODY>节点。

节点也可以拥有后代，后代指某个节点的所有子节点，或者这些子节点的子节点，以此类推。比方说，所有的文本节点都是 <HTML>节点的后代，而第一个文本节点是

<HEAD> 节点的后代。

节点也可以拥有祖先节点。祖先节点是某个节点的父节点，或者父节点的父节点，以此类推。比方说，所有的文本节点都可以把< HTML >节点作为先辈节点。

一个 HTML 文件可以转化成如图 3.1.1 所示的树型结构，使用树结构对页面结构进行分析具有以下优点：

（1）对节点操作：添加节点、删除节点、在特定的节点中增加新的属性或节点，以及修改节点的内容。在网页视图重构和转化中可以通过这样一些操作改变内容的表现形式和视图的大小，而不改变网页内容。

（2）导出新结构：在标记树的结构上根据不同的需要导出或生成一种新的代表 HTML 文档某方面特征的新的结构。

DOM 的接口都是符合工业标准的 IDL（Interface Definition Language，界面定义语言）描述的，不限制用何种语言具体实现这些接口。DOM 的核心是将面向对象（Object-Oriented）的概念引入 HTML/XML 文件的处理中。在 DOM 产生以前，无论是 HTML 还是 XML，均被看作是包含各种组件的数据集合，以面向数据的方式管理文件。引入对象后，在 DOM 看来，HTML/XML 的组件不只包含数据本身，每一个 HTML/XML 中的元素（Element）还包含有方法（Method）和属性（Attribute）。DOM 使用包含这些方法和属性的 API，通过方法和属性来存取和管理组件。

3.2 网页清洗

WWW 上的网页通常包含两部分内容：一部分内容体现的是网页的主题信息，比如一个新闻网页中的正文部分，称之为“主题内容”；另一部分则是与主题内容无关的导航条、广告信息、版权信息等内容，称之“噪音”内容。噪音内容通常分布在主题内容周围，有时也夹杂在主题内容中间，它们通常是以链接导航文字的形式出现的。噪音内容一般与网页的主题内容不相关，因此，网页中的噪音内容给 Web 上基于网页内容的应用系统带来了很大的困难，降低了系统的处理效率和准确性。

快速准确地识别并清除网页内的噪音内容（称之为网页清洗）是提高 Web 应用程序处理结果准确性的一项关键技术[27][28]。首先，网页清洗后，没有了噪音内容的干扰，Web 应用程序可以以网页的主题内容为处理对象，从而提高处理结果的准确性。其次，网页清洗可以显著简化网页内标签结构的复杂性并减少网页的大小，从而节省后续处理过程的时间和空间开销。因此，网页清洗已成为 Web 信息系统与处理环节中一个必不可少的工作。

在网页分类领域，由于噪音内容与主题无关，因此训练集中的噪音内容会使各个类别的特征不够明显，而待分类网页中的噪音内容则会导致该网页类别不明确，因而影响

网页自动分类的效果。

在主题搜索领域，传统的搜索算法以网页为粒度构造的网络图不够准确，大量的广告、导航条等噪音内容会使网页的主题发生漂移，必须深入到网页内部将处理单元的粒度缩小，才能提高内容分析的准确性。

在网页信息提取领域，自动识别模式的方法必须要从整个网页中提取模式，而不是只针对主题内容进行提取。因此，在清洗后的网页上作信息提取不仅可以排除噪音信息对信息提取的干扰，提高信息提取的准确性，而且可以使得网页的结构简单化，提高信息提取的效率[29]。

从上述分析可以看到，噪音内容对基于网页研究工作的影响是普遍而严重的，虽然各个领域采用的方法各不相同，但处理的目的都是为了排除网页中噪音内容的干扰，得到真正的主题内容。

在视觉上，一张网页的页面可以划分为若干个区域，把一个区域称为一个内容块。有的内容块包含主题内容，有的则包含噪音内容。通常，同一个内容块中的内容是紧密相关的，这就意味着可以以内容块为单位对网页中的内容进行取舍。基于这样的分析，网页清洗过程就是保留网页中包含主题内容的内容块而去掉包含噪音内容的内容块。因此，网页清洗过程可以分为两个步骤：网页内容结构的表示和网页内容块的取舍。

HTML 是一个超文本标记语言，它定义了一系列标签来描述网页显示时的页面布局以及显示方式。对于 HTML 网页最常用的结构表示方法是构造网页的标签树。DOM 是现在常用的标签树构造工具[30]，它可以将网页中的标签按照嵌套关系整理成一棵树状结构。针对网页净化的特殊要求，我们首先对 HTML 规范中的标签按照功能进行分类，进而提出更加适合网页净化的标签树的构造方法。

根据标签的作用可以将 HTML 标签分为两类：

（1）规划网页布局的标签。一个网页的内容块是由特定的标签（称之为容器标签）规划出来的。常用的容器标签有<TABLE>，<TR>，<TD>，<P>，<DIV>等。

（2）描述显示方式的标签。除了描述布局结构的标签外，HTML 标准中还定义了一套标签用来描述其包含的内容本身。比如：<b>标签说明它所所包含的内容要用粗体显示，<img>标签说明它包含的是一个图片。

网页去噪是以内容块为单位进行保留和删除的，因此，依照容器标签构造的标签树中的结点是比较合理的。而其它类型的标签树信息可以作为它所在的内容块的属性而存在。大多数网页的构造特征比较明显，也就是以<TABLE>标签来划分网页。这样网页去噪工作可以在以< TABLE >标签划分内容块的基础上进行保留和删除工作。

本文网页清洗的目的是获取网页中的主题信息，也就是正文文本，去掉页面中导航信息、广告信息、版权信息等“噪音”内容，清洗过程如图 3.2.1 所示：

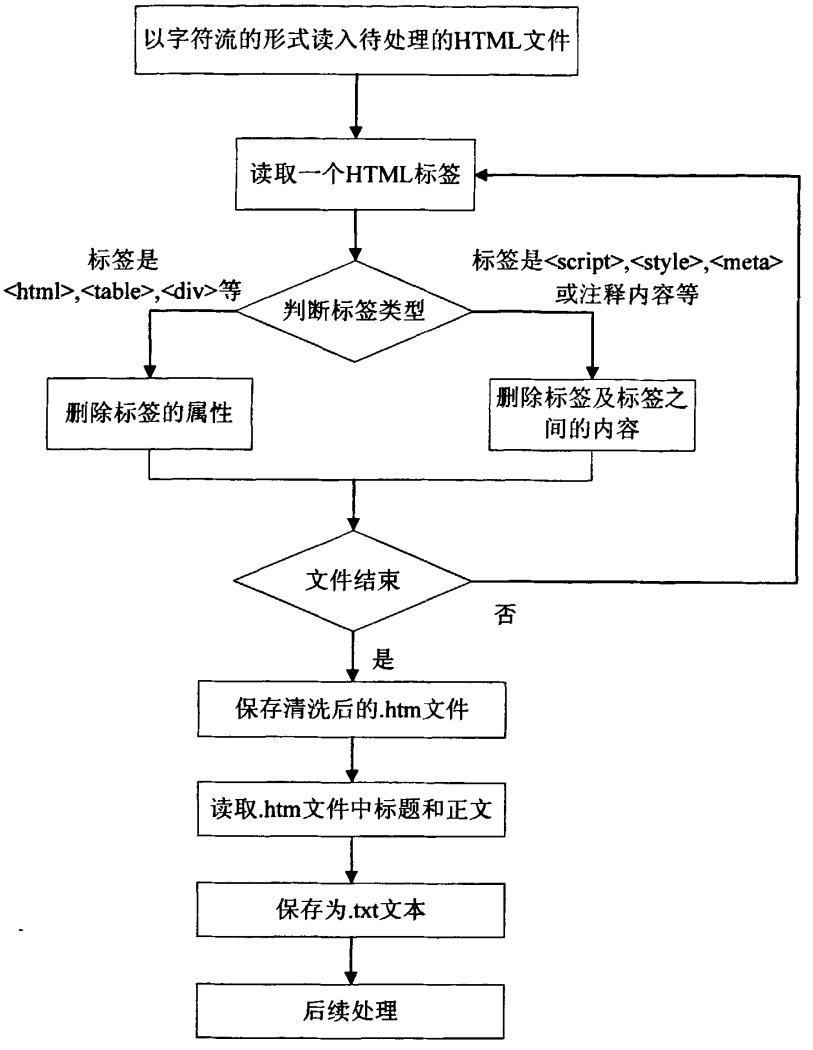

图 3.2.1 网页清洗过程

具体解释如下：

（1）对于节点是<SCRIPT>，<STYLE>，<META>，<FORM>，<INPUT>，<SELECT>，<BUTTON>等标签元素时，这类标签元素的特点是它们没有子节点，而且代表特定含义；对于节点是<NOSCRIPT>，<IMG>等这种标签元素时，这类标签元素的特点是当页面不支持它们对应的对象显示时，会使用锚文本代替显示出来；对于节点是<TABLE width = 500>等标签元素时，这类标签元素的特点是含有标签属性。如果要过滤这种标签元素的属性时，就可以把标签的属性和属性值都删除即可。比如<TABLE width = 500>，可以过滤成<TABLE>。

遍历 HTML 页面的 DOM 树结构，当发现遍历到的当前节点是诸如<SCRIPT>，<NOSCRIPT>，<IMG>，<STYLE>，<META>，<FORM>，<INPUT>，<SELECT>，<BUTTON>等标签时，可以删除此节点元素。若遍历到的节点是<TABLE width = 500>

这种标签时，只要把标签的属性值去除即可。

（2）对于像导航栏、分类表以及广告链接块这样的内容块，它最大的特征就是内容块中几乎全部都是链接，所以这个部分处理的实际上是一个链接表。根据这个特点，可以计算一下内容块中文字的总数和锚文本字数的总数，如果他们的比值接近于 1，那么，就可以认定这是一个和主题无关的链接列表了。可以删除这样的链接列表。

（3）移除那些已经被清空了内容的表格。通常情况下一个表格的内容是应该达到一定的字数的，如果一个表格的字数低于了一个阈值，这个表格就可以认为是空表了。经过前面几步的处理，一些表中内容会被清空，就会产生空表。这样的表格完全没有任何信息，应该将它们从 DOM 树之中去除。

（4）去掉页面中的注释。页面中的<!>为注释内容，可以直接删除。

3.3 中文分词

对中文文本做预处理时，首要任务是对中文文本进行分词处理，中文文本分词效果的好坏将直接影响到文本自动分类的最终效果。

自动分词是针对中文的一种自然语言处理技术，中文与英文不同，句子中各个词条间没有固有的间隔，为了对文本信息进行分类、索引等处理，首先需要对中文文本进行词条切分（简称分词）。中文文本的分词处理是指在中文文本中切分出连续的能够代表语义单元的词或者 n 元词条，将中文文本的连续的字节流形式转化为离散单词流形式的过程。自动分词技术是各种中文信息处理技术的基础，也是中英文之间研究文本自动分类的主要差异所在，中文文本自动分类要在自动分词的基础上进行，对中文文本进行分词的过程也是文本特征集的确定过程。

目前中文分词方法主要有机械分词法、基于理解的分词方法、基于统计的分词方法等三种[31][32]。

3.3.1 机械分词法

机械分词法的基本思想是按照一定的策略切取待分析汉字串的子串，然后与预先建立好的词典中的词条进行匹配，若与词典中某项吻合，则匹配成功，识别出一个单词；若在词典中找不到项与该子串匹配，则匹配失败，当前子串并非中文单词。

机械分词法根据扫描原始汉字串方向的不同可分为正向匹配法和逆向匹配法；按照不同长度的优先匹配，可分为最大匹配法和最小（短）匹配法；按照是否和词性标注过程相结合，又可以分为单纯分词方法和分词与标注相结合的一体化方法。简单来说，机械分词方法主要是以下几种：最大匹配法（Maximum Matching method，MM），逆向最大匹配法（Reverse Maximum Matching method，RMM），双向扫描法（Bi-direction Matching method，BM）以及最佳匹配法（Optimum Matching method，OM）等等。

3.3.2 基于理解的分词方法

通常的分词系统，都力图在分词阶段消除所有歧义切分现象，有些系统则在后续过程中来处理歧义切分问题，其分词过程只是整个语言理解过程的一个小部分。其基本思想就是在分词的同时进行句法、语义分析，利用句法信息和语义信息来处理歧义现象。它通常包括3个部分：分词子系统、句法语义子系统、总控部分。在总控部分的协调下，分词子系统可以获得有关词、句子等的句法和语义信息来对分词歧义进行判断，即它模拟了人对句子的理解过程。这种分词方法需要使用大量的语言知识和信息。由于汉语语言知识的笼统、复杂性，难以将各种语言信息组织成机器可直接读取的形式，因此，目前基于理解的分词系统还处于试验阶段，联想—回溯法就是其中的一种。

联想—回溯法（Association-backtracking method，AB）：需要建立知识库，包括特征词词库、实词词库和规则库。首先将待切分的汉字字符串序列分割为若干子串，子串可以是词，也可以是由几个词组合成的词群，然后就利用实词词库和规则库将词群细分为词。切词时，要利用一定的语法知识，建立联想机制和回溯机制。联想机制由联想网络和联想推理构成，联想网络描述每个虚词的构词能力，联想推理利用相应的联想网络来判定所描述的虚词究竟是单独的词还是作为其他词中的构成成分。回溯机制主要用于处理歧义句子的切分。联想—回溯算法虽然增加了算法的时间复杂度和空间复杂度，但是这种方法的切词正确率得到了提高，是一种行之有效的方法。

3.3.3 基于统计的分词方法

从形式上看，词是固定的字的组合，因此在上下文中，相邻的词同时出现的次数越多，就越有可能构成一个词。因此字与字相邻共现的频率或概率能够较好地反映成词的可信度。互信息是一种度量不同字符串之间相关性的统计量。对于字符串X和Y，其互信息的计算公式如下：

$$MI(X,Y)=\log_2\frac{P(X,Y)}{P(X)P(Y)} \tag{3.3.1}$$

其中，$P(X,Y)$是字符串X和Y共现的概率，P(X)，P(Y)分别是字符串X和Y在语料中出现的概率。互信息体现了字符串之间结合关系的紧密程度。当紧密程度高于某一个阈值时，就可以认为此字组可能构成一个词。这种方法只需要对语料中字的频度进行统计，不需要切分词典，因而又称为无词典分词法或统计取词方法。但这种方法也有一定的局限性，会经常抽出一些共现频度高，但并不是词的常用字组，例如“这一”、“之一”、“有的”、“我的”等，并且对常用词的识别精度差，时空开销大。实际应用的统计分词系统都要使用一部基本分词词典进行串匹配分词，同时使用统计方法识别一些新词，即将字频统计和串匹配结合起来，既发挥匹配分词切分速度快、效率高的特点，又利用了无词典分词结合上下文识别生词、自动消除歧义的优点。常用的有基于词频统计

的切词法和基于期望的切词法。

（1）基于词频统计的切词法

这种方法利用词频统计的结果帮助在切词过程中处理歧义切分字段。这种方法的缺点是：由于只考虑词频，出现频率较低的词总是被错误地切分。

（2）基于期望的切词法

这种方法认为一个词的出现，它后面紧随的词就有一种期望，据这种期望，在词典中找到所有的词从而完成切分。这种方法增加了切词的空间复杂度，但在一定程度上提高了切词的正确率。

中文文本自动分词技术一般以词典作为分词依据，使用专门的分词算法将文本中出现于词典中的词识别出来。通过这种方法获得的文本特征只能是词典中出现的词汇，但是自然语言领域相关性和随时间变化的特性，词典中不可能包含文本中所有词汇，因此，对不同类型文本进行分类时，就需要不断修整和扩充词典并改进分词技术，才能获得良好的分类性能。

自从 80 年代初自动分词被提出以来，有众多的专家和学者为之付出了不懈的努力，涌现了许多成功的汉语分词系统，主要有北京航空航天大学研制的 CDWS 和 CWSS 分词系统，清华大学黄昌宁、马晏等开发的 SEG 系统，东北大学姚天顺建立的基于规则的汉语分词系统，南京大学王启祥等人实现的 WSBN 分词系统，中科院计算所研制出的汉语词法分析系统 ICTCLAS 等等[33]。

本文采用中科院计算所研制的汉语词法分析系统 ICTCLAS（Institute of Computing Technology，Chinese Lexical Analysis System）进行分词。ICTCLAS 的主要功能包括中文分词，词性标注，命名实体识别，新词识别，同时支持用户词典。该系统分词正确率达 98.45%(937 专家组评测)，未登录词识别召回率高达 90%，其中中国人名的识别召回率接近 98%。

ICTCLAS 的主要思想是先通过层叠形马尔可夫模型进行分层[34]，共分五层，通过分层，既增加了分词的准确性，又保证了分词的效率。其基本思路是：先对字符串进行原子切分，然后在此基础上进行 N-最短路径粗切分[35]，找出前 N 个最符合的切分结果，生成二元分词表，然后生成分词结果，接着进行词性标注并完成主要分词步骤。如图 3.3.1 所示：

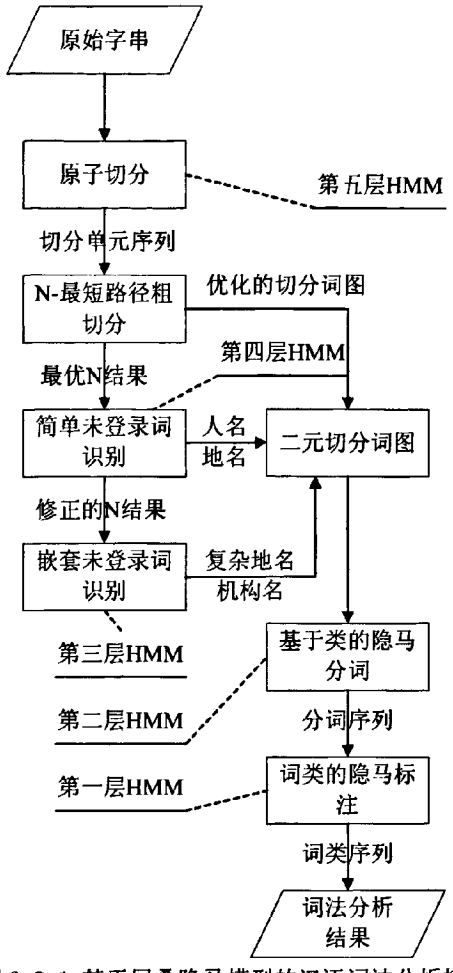

图 3.3.1 基于层叠隐马模型的汉语词法分析模型

3.4 去除停用词

在对文本进行分词之后，文本就变成了一系列词集的表示，本文先利用了 ICTCLAS 的词性标注功能，去掉连词、代词、介词、虚词等没有实际意义的词以及标点符号，初步缩减特征维数。进行了初步降维之后，还有一些词在整个文本集中出现的频率很高，但对分类来说作用不大，我们称之为停用词，这些词也应该去掉。

在试验过程中，人工地将我们认为对分类作用不大的词选出，如“是”，“不”，“如上所述”，“没有”，“the”等放入停用词表，并在试验过程中不断对停用词表进行扩充。

停用词表包括：（1）中文词组约 550 个；（2）英文字母及单词约 160 个；（3）各类符号及数字约 130 个。

在本文的试验样本中，去除停用词前特征集中有 27567 个特征项，去除停用词后，特征集中有 20313 个特征项，维数减小了 26.31%。

3.5 常用的特征选择方法

特征选择是 Web 文本分类中的一个重要环节。对文本进行分词处理并去除停用词之后，特征词的数量仍然非常庞大，一般的学习算法无法对其进行类别学习，因此有必要进行特征子集的抽取，进一步缩小特征空间维数。

特征选择是选择描述文本的最佳特征子集的过程，使得到的特征子集至少与原始特征集相同，即 $F' \subseteq F$，其中 F 是原始特征集，F' 是经过特征选择后得到的特征子集。

特征选择可以从两个方面提高系统的性能：一是分类速度，通过特征选择，可以大大减少特征集合中的特征数，降低文本向量的维数，提高系统运行速度。二是准确率，通过适当的特征选择，可以去掉停用词、低频词和类别区分度不大的特征词，不但不会降低系统的准确率，反而会使系统提高精度。

下面介绍了常用的特征选择方法。

3.5.1 文档频度

文档频度（Document Frequency，DF）是一种最简单的特征选择算法，它是指在训练文本集中包含特征项 t 的文档数[36]。DF 的主要思想是：计算训练文档集中每个特征项的文档频次，若该项的 DF 值低于最低阈值或高于最高阈值则将其剔除，因为它们分别代表了“没有代表性”和“没有区分度”两种极端的情况。

文档频度不需要依赖类信息，所以是一种无监督的特征选择，常被集成在文本预处理中用来删除出现次数过少或者出现次数过多的单词以提高后续处理的效率。文档频度在计算量上比其它的评估函数小得多，但是在实际运用中的效果却较好。它的时间复杂度跟文本规模成线性关系，非常适合于超大规模的文本集的特征选择。

3.5.2 信息增益

信息增益（Information Gain，IG）是一种基于熵的特征评估方法[37]，在机器学习领域应用较为广泛。当它用于文本数据的特征选择时，定义为特征项为分类提供的信息量。定义如下：

$$IG(t) = -\sum_{i=1}^{|c|} P(c_i)\log P(c_i) + P(t)\sum_{i=1}^{|c|} P(c_i \mid t)\log P(c_i \mid t) + P(\bar{t})\sum_{i=1}^{|c|} P(c_i \mid \bar{t})\log P(c_i \mid \bar{t}) \qquad (3.5.1)$$

其中，$P(\bar{t})$ 表示词 t 不出现的概率，$P(c_i \mid t)$ 表示词 t 出现的情况下文本属于 c_i 类的概率，$P(c_i \mid \bar{t})$ 表示词 t 不出现的情况下文本属于 c_i 类的概率。

3.5.3 互信息

互信息(Mutual Information，MI)体现了特征项和类别的共现程度[38]，特征项对于类别的互信息越大，它们之间的共现概率也越大，其定义如下：

$$MI(t)=\sum_{i=1}^{|c|}p(c_i)\log\frac{p(t\mid c_i)}{p(t)} \tag{3.5.2}$$

$p(c_i)$表示第 i 类文本在训练文本集合中出现的概率，$p(t)$表示词 t 在训练文本集合中出现的概率，$p(t\mid c_i)$表示在第 i 类的文本中 t 的出现概率。

设 A 为包含特征项 t 且属于c_i类的文档数目，B 为包含特征项 t 但不属于c_i类的文档数目，C 为属于c_i类文档但不包含特征项 t 的文档数目，N 为文档总数，则有如下近似计算公式：

$$MI(t)\approx\sum_{i=1}^{|C|}p(c_i)\log\left(\frac{A\times N}{(A+C)\times(A+B)}\right) \tag{3.5.3}$$

3.5.4 χ^2 统计

χ^2 统计类似于互信息，也用于衡量单词与类别之间的共现程度[39]，但它比互信息更强，因为它同时考虑了特征存在与不存在时的情况。χ^2 越大，单词与类别相关性越大。χ^2 统计的计算公式如下：

$$\chi^2(t,c_i)=\frac{N\times(AD-BC)^2}{(A+C)\times(B+D)\times(A+B)\times(C+D)} \tag{3.5.4}$$

其中，A 是类c_i中含有特征 t 的文档数量；B 是类c_i以外含有特征 t 的文档数量；C 是类c_i中不含特征 t 的文档数量；D 是类c_i以外不含特征 t 的文档数量；N 表示文档总数。

公式(3.5.4)仅仅只是单词相对于某一个类的χ^2值，它相对于整个文本集的χ^2值是其相对于所有类χ^2值的综合。综合的方式通常有两种，一种是加权平均，另一种是取最大值：

$$\chi^2_{avg}(t)=\sum_{i=1}^{|c|}p(c_i)\chi^2(t,c_i) \tag{3.5.5}$$

$$\chi^2_{\max}(t)=\max_i(\chi^2(t,c_i)) \tag{3.5.6}$$

其他常见的方法还有期望交叉熵（Expect Cross Entropy），文本证据权（the weight of evidence for text），词条强度（Term Strength）等，后来也提出了一些新的特征选择方法，如低损降维方法、频率差方法、Bayes 准则法、F1 值准则法和 Fisher 简便量法等。

3.6 基于同义词统计的特征选择

在汉语表达中，存在着大量的同义词，同义词的使用丰富了语言的表达，但分散了同一概念在文本集合中的频率，造成了特征数据的稀疏性。上节所述的特征选择方法对每一个特征词都单独统计其函数评估值，没有考虑同一主题下同义词的合并处理问题。如果在特征选择之前，先统计文档集中的同义词，并将这些词合并替换，作为一个词来处理，将有效地降低特征空间维数，提高自动分类系统的效率。

3.6.1 《同义词词林》扩展版介绍

传统的《同义词词林》按照树状的层次结构把所收录的词组织在一起，提供三级编码[40]，把词汇分成大、中、小三类，大类 12 个，用一位大写字母表示，中类 97 个，用一位小写字母表示，小类 1400 个，用二位十进制整数表示。每个小类里有很多词，这些词又根据词义的远近和相关性分成了若干个词群（段落）。每个段落中的词语又进一步分成了若干行，同一行的词语要么语义相同，要么词义有很强的相关性。为了将词义相关的行和同义的行区分开，《同义词词林》在行的左端加上“**”作为标记。例如：“Ae07 农民 牧民 渔民”，“Ae07”是编码，“农民 牧民 渔民”是该类的标题，标题由一个或多个段首（每段的第一个词）组成。如表 3.6.1 所示。

表3.6.1 词典结构示例

Ae07 农民 牧民 渔民
农民 农夫 农人 农 庄稼人 庄稼汉 田父 泥腿子 农家 耕夫 老乡
小农 个体农民
佃农 佃户
上中农 富裕中农
** 菜农 棉农 茶农 烟农 蔗农 花农 药农 林农
雇农 贫农 下中农 中农 上中农 富农
自耕农 半自耕农 集体农民 人民公社社员

哈工大信息检索实验室在传统的《同义词词林》的基础上做了进一步的扩充，形成了《同义词词林》扩展版[41][42]。

《同义词词林》扩展版在原有的三级编码的基础上，将小类中的段落看作第四级的分类，段落中的行看作第五级的分类。这样，《同义词词林》扩展版就具备了五级编码。新增的第四级和第五级编码与原有的三级编码构成一个完整的编码，唯一地代表词典中的词语。第四级用一位大写字母表示，第五级用二位十进制整数表示。如：

Ba01A02= 物质 质 素

Cb02A01= 东南西北 四方

由于第五级中有的行是同义词，有的行是相关词，有的行只有一个词，可以分为具体的三种情况，需要区别对待，所以有必要增加标记来分别代表三种情况。具体标记参见表 3.6.2。

表3.6.2 同义词林扩展版编码表

编码位	1	2	3	4	5	6	7	8
符号举例	D	a	1	5	B	0	2	=\#\@
符号性质	大类	中类	小类		词群	原子词群		
级别	第一级	第二级	第三级		第四级	第五级		

表中的编码位是按照从左到右的顺序排列。第八位的标记有 3 种，分别是“=”、“#”、“@”， “=” 代表 “相等”、“同义”；“#” 代表 “不等”、“同类”，属于相关词语；“@” 代表 “自我封闭”、“独立”，它在词典中既没有同义词，也没有相关词。

在实际的实验中，《同义词词林》扩展版在进行同义词的比较和替换过程中有许多的不便，需要重新组织《同义词词林》的词典结构。根据《同义词词林》扩展版，本文重新生成了数据文件和索引文件，去掉了原有的相关词、多义词以及独立的词。

索引文件中任一条记录的格式如下：

lexicalName synsetOffset

例如：大伙 Aa01C03=

数据文件中任一条记录的格式如下：

synsetOffset wordNumber <word>

例如：Aa01C03= 6 大伙 大伙儿 大家伙儿 各户 一班人 众家

其中<>表示可以为有限多项，各个字段的含义如表 3.6.3 所示：

表 3.6.3 数据文件和索引文件格式说明

数据文件		索引文件	
字段名	含义	字段名	含义
synsetOffset	同义词集合编号，长度为 8 的字符串	lexicalName	词语名称
wordNumber	集合中词语的个数，十进制整数表示	synsetOffset	包含该词语的同义词集合的编号
word	各个词语名称		

根据本文实验的需要，我们只选取了五级编码中第八位为 “=” 的同义词集，即本

文只关心训练文本集中的同义词，不考虑词性相关的词和独立的词。同时，由于同义词集中还包括一词多义的词，还需要做进一步的处理，去除多义词，这样就大大缩小了词典的大小，提高了效率。

3.6.2 基于《同义词词林》的特征选择方法

基于同义词统计的特征选择就是根据上一节生成的同义词词典，对经过分词处理后的文本中的特征词进行处理，如果多个特征词出现在词典中一个同义词词组中，则合并为一个特征词。

其特征选择步骤如下：

步骤 1：对训练文档集进行分词：分词方法采用中国科学院计算所的 ICTCLAS 分词系统，利用 ICTCLAS 的词性标注功能，去除虚词、助词、语气词等特征词，然后根据停用词表去除停用词。

步骤 2：同义词合并：首先剔除《同义词词林》扩展版中具有一词多义的词，词性相关的词以及独立的词，生成本文实验所需要的同义词词典，词典主要包括数据文件和索引文件，然后根据同义词词典对分词后的文本中的特征词进行同义词合并。

步骤 3：进行了同义词合并后，重新统计文本中特征词的频度，得到特征词的概率估算公式如下：

$$P(t) = (Nt + N't) / N \tag{3.6.1}$$

$$P(C_i \mid t) = (Nt\_C_i + N't\_C_i)/(Nt + N't) \tag{3.6.2}$$

$$P(t \mid C_i) = (Nt\_C_i + N't\_C_i) / N\_C_i \tag{3.6.3}$$

其中，$N't$ 表示训练文本中不出现特征词 t 但出现其同义词的文档数，$N't\_C_i$ 表示 C_i 类文档集中不出现特征词 t 但出现其同义词的文档数。

步骤 4：根据步骤 3 的概率估算公式(3.6.1)-(3.6.3)，利用前面介绍的特征评估函数计算文档集中每一个特征项的函数评估值，选取满足某一阈值的特征项生成特征空间。对于阈值大小，我们可以在试验过程中，选取具有最佳分类效果的阈值。

具体流程图如图 3.6.1 所示：

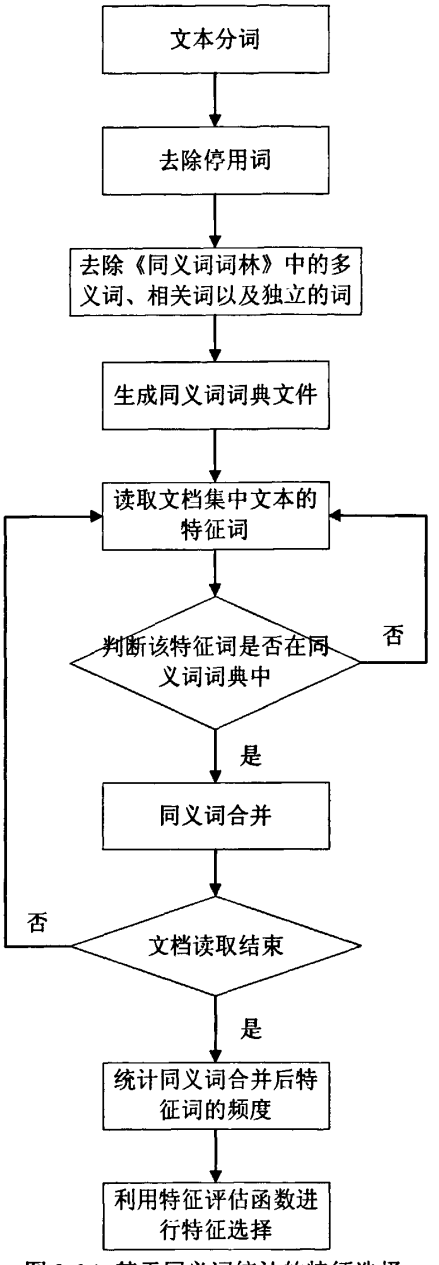

图 3.6.1 基于同义词统计的特征选择

在进行同义词统计时，由于考虑了词及其同义词在文档集中的频次，这就使特征提取从词的层而上升到了主题概念层面，不仅极大缩小了原始的特征空间，而且有利于提高分类精度。

3.7 文档的表示

3.7.1 文档的向量表示

文档的内容是用自然语言描述的，计算机很难理解其语义，所以必须将文档的内容转化成一种格式化的数据表示。转化的过程叫做文本表示，转化的方法叫做文本表示模型。文本表示模型有许多种，如布尔模型、向量空间模型、概率模型、聚类模型、基于知识的表示模型以及混合模型等。其中向量空间模型(Vector Space Model，VSM)是近年来应用较多、效果较好的一种[43][44][45]。

经过前面的分词、特征选择等处理过程后，一个文本由自然语言描述转化为由一系列特征词描述。在 VSM 中，将从文本中提取的特征词组成特征向量，并计算每一个特征词的权重。例如文档表示为 $T=(t_1,t_2,...,t_n)$，其中 $t_i(1\le i\le n)$ 是特征词。根据特征词的重要程度，可以赋予不同的权重 w_i 进行量化，这样文档也可表示为 $T=(w_1,w_2,...,w_n)$，其中每一项 w_i 与相应的特征词 $t_i(1\le i\le n)$ 相对应。在 VSM 中，不考虑特征词在文中出现的先后顺序，只保证特征词的唯一性，然后把 n 个特征词看成一个 n 维坐标系，则相应的权重 w_i 为文档在坐标系中的坐标值，一个文档就可以被表示成一个 n 维空间中的向量。

一个文档的集合可以表示为 $A=(T_1,...,T_k,...,T_m)$，其中，$T_k(1\le k\le m)$ 表示文档集中的一个文档，即 $T=(w_1,w_2,...,w_n)^T$，m 表示文档集中的文档数，这样一个文档集就可以表示为一个 $n\times m$ 的矩阵。例如对一个包含 6 个文本的文档集合，每一个文档包含 8 个特征项，则该文档集可以表示为：

$$\begin{bmatrix} w_{11} & w_{12} & \cdot & \cdot & \cdot & w_{16} \\ w_{21} & \cdot & & & & \\ \cdot & & & & & \\ \cdot & & \cdot & & & \\ \cdot & & & & & \\ \cdot & & & \cdot & & \\ \cdot & & & & & \\ w_{81} & & & & & w_{86} \end{bmatrix} \tag{3.7.1}$$

其中 w_{nm} 为文档集中第 m 个文档的第 n 个特征项的权重。

3.7.2 特征项权重的计算

给每个特征项赋予权重时，应该使文本中越重要的特征项权重越大，但是应该避免一个强的特征把其它特征淹没，这会造成一种信息的不正确放大。一种方法是由专家或者用户根据自己的经验与所掌握的领域知识，人为赋予权重。这种方法随意性强，而且效率低，但是对于特定领域的分类有着重要的作用，可以作为一个辅助的权重计算方法。

还有一种方法是由文章自身的特点确定特征项的权重，这种方法的优点是简单易懂，计算简单，但它的缺点是其计算的权重往往不能充分代表该词在分类中的重要程度。可能对于文章本身而言其权重正确，但是对于分类而言其特征权重却不能很好地反映类别的区分度。第三种方法就是把特征项和所有类别中其他的文本进行比较从而得出其自身的特征权重，这种方法的理论依据在于信息度量是来源于对比获得，从对比中，知道自身的信息含量。

目前广泛使用的是基于统计的 TF-IDF 公式[46]：

$$W_{ik} = TF_{ik} * IDF_{ik} \tag{3.7.2}$$

其中 W_{ik} 代表第 k 个特征项在文档 D_i 中的权重，TF_{ik} 为 T_k 在文档 D_i 中的词频。IDF_{ik} 为逆文本词频，其计算方法有很多种。目前较为常用的公式为：

$$IDF_{ik} = \log\left(\frac{N}{n_k} + 0.01\right) \tag{3.7.3}$$

其中 N 代表所有的训练文本的总数，n_k 代表的是训练文本中出现该特征项的文本数。

该公式的本质就是香农原理：认为某项如果在其他文本中出现的次数越少，那么认为该项就越具有代表性，其所含有的信息就越多；相反如果在其他文本中出现的次数越多，认为其越不具有代表性，其所含有的信息就越少。

为消除文本长度对计算特征词权值的影响，还应该对公式（3.7.2）做规范化处理，公式如下：

$$W_{ik} = \frac{TF_{ik} * \log\left(\frac{N}{n_k} + 0.01\right)}{\sqrt{\sum_{k=1}^{n} (TF_{ik})^2 \log^2\left(\frac{N}{n_k} + 0.01\right)}} \tag{3.7.4}$$

另外，对于特征较为明显的文本类别，往往有少数项的出现频率数远远大于其他项。根据上述计算公式计算出的权值会很高，如果个别项的权值很高，在分类过程中往往会抑止其他项的作用。因此在计算各项权重时，应对统计出的词频做适当的均衡处理，较为简单的均衡处理方法是对统计的权值进行开平方。经过均衡处理的权值计算公式为：

$$W_{ik} = \frac{\sqrt{TF_{ik} * \log\left(\frac{N}{n_k} + 0.01\right)}}{\sqrt{\sum_{k=1}^{n} (TF_{ik}) \log\left(\frac{N}{n_k} + 0.01\right)}} \tag{3.7.5}$$

权重的计算只能视具体情况而定，至今没有普遍使用的“最优公式”。另外，前面的讨论中特征项的权重一般为正值，其实权值也可以取负值，用来描述用户厌弃某特征。

TF-IDF 公式是一种经验公式，并没有坚实的理论基础。但是，多年的实验表明，上述公式是文本处理中的一个有效的工具。事实上，这一公式不仅在信息检索中得到了成功的应用，它对于其他文本的处理领域，如信息分发、信息过滤都有很好的借鉴意义。

4 支持向量机在 Web 文本分类中的应用

4.1 LIBSVM 分类器介绍

本文采用LIBSVM-2.88作为文本分类器。LIBSVM是台湾大学林智仁(Chih-Jen Lin)博士等开发设计的一个开源的 SVM 软件包[47]，可以解决分类问题（包括 C-SVC、n - SVC)、回归问题（包括 e - SVR、n - SVR）以及分布估计（one-class-SVM）等问题，提供了线性、多项式、径向基和 S 形函数四种常用的核函数供选择，可以有效地解决多类问题、交叉验证选择参数、对不平衡样本加权、多类问题的概率估计等。他不仅提供了 LIBSVM 的 C++语言的算法源代码，还提供了 Python、Java、R、MATLAB、Perl、Ruby、LabVIEW 以及 C#.net 等各种语言的接口，可以方便的在 Windows 或 UNIX 平台下使用。

LIBSVM 使用的训练数据和测试数据文件格式如下：

<label> <index1>：<value1> <index2>：< value2> ...

其中<label>是训练数据集的目标值，对于分类，它是标识某类的整数(支持多个类)；对于回归，是任意实数。<index>是以 1 开始的整数，表示特征的序号；<value>为实数，也就是我们常说的特征值或自变量。当特征值为 0 时，特征序号与特征值 value 都可以同时省略，即 index 可以是不连续的自然数。<label>与第一个特征序号、前一个特征值与后一个特征序号之间用空格隔开。测试数据文件中的 label 只用于计算准确度或误差，如果它是未知的，只需用任意一个数填写这一栏，也可以空着不填。

4.2 支持向量机在文本分类中的应用

1998 年，Dortmund 大学的 Joachims 报道了将 SVM 用于文本分类的实验结果，实验在 Reuters 和 Ohsumed 两个标准语料库上进行，通过与贝叶斯、Rocchio、k 最近邻和决策树这四种分类方法的进行比较，SVM 方法不仅取得了更好的分类效果，还表现出了更强的鲁棒性和处理高维数据的优良特性。Joachims 对 SVM 方法在文本分类中的应用进行了大量深入细致的研究，取得了一系列的成果，并实现了一个简单有效的工具箱 SVMLight。该工具箱非常适合于解决文本分类问题，被各国学者广泛使用。

目前，越来越多的人开始研究 SVM 在文本分类中的应用，并提出了很多的改进方法。都云琪等采用线性支持向量机（LSVM）实现了一个中文文本分类系统[48]，并针对系统准确率较高、召回率较低的问题提出一种采用训练集中拒识样本信息对分类器输出进行改进的方法，通过给最优分类面加入一个松弛项，松弛项的值通过统计拒识样本与分类面距离的期望平均得到，从而可以使分类器识别出更多的样本，提高系统的召回率。

李晓黎等提出了一种将支持向量机与无监督聚类相结合的方法[49]，该方法首先利用无监督聚类分别对训练集中的正例和反例聚类，然后挑选一些例子参加训练并获得 SVM 分类器。任何网页可以通过比较其与聚类中心的距离来决定采用无监督聚类或 SVM 进行分类。该算法充分利用了 SVM 准确率高和无监督聚类速度快的优点。李蓉等提出一种将 SVM 与最近邻分类相结合的方法[50]，该方法在对 SVM 分类时出错样本点的分布进行研究的基础上，在分类阶段先计算待分类样本和最优分类超平面的距离，如果距离差大于给定的阈值则直接用 SVM 分类，否则代入以每类的所有支持向量为代表点的 k 近邻分类。萧嵘等将 SVM 与增量学习方法结合起来，提出了一种基于遗忘因子 α 的 SVM 增量学习方法(α-ISVM)[51]，该方法通过在增量学习中逐步积累样本的空间分布知识，使得对样本进行有选择地遗忘成为可能。

上面介绍的学习方法都属于归纳式学习（Inductive Inference），即希望设计一个分类器能够对将来所有的样本都有好的分类性能。而在很多实际问题中，不可能也没有必要用这样一个分类器对所有的样本进行识别，可以考虑设计一种直接从已知样本出发对待分类样本进行识别和分类的方法，这种方法叫做直推式学习（Transductive Inference）。较之归纳式学习方法，直推式学习更具有普遍性和实际意义。基于 SVM 的直推式学习是一个较新的研究领域，Joachims 等对此进行了研究，提出了一种训练直推式支持向量机（TSVM）的有效方法，成功地把无标签样本中隐含的分布信息引入到 SVM 的学习过程中。在 Reuters，WebKB，Ohsumed 三个语料库上的实验结果表明，TSVM 比单纯使用有标签样本训练得到的分类器在性能上有了显著提高，并且可以大大减少对有标签样本的需求，这对于大规模的文本分类问题来说无疑具有很重要的实际意义。但 TSVM 算法执行之前必须人为地指定待训练的无标签样本中的正标签样本数，这个值是很难做出比较准确地估计的。针对这一问题，陈毅松等提出了一种新的渐进直推式 SVM 学习算法(PTSVM)，这种算法没有对无标签样本中的正标签样本数做出盲目的规定，而是在训练过程中渐进地对无标签样本赋予标签并动态地予以调整，因而产生的分类器可以更好地描述样本的分布特征，具有更好的推广能力。因为基于直推式学习的 SVM 分类器是一个较新的研究领域，它在文本分类中的应用还有很多值得进一步研究的地方。

4.3 SVM 用于 Web 文本挖掘中的优点

与传统的学习方法相比较，SVM 有许多突出的优点[52]，使它更适合于网上文本信息自动分类的任务，主要有：

（1）网上信息的覆盖面广泛，无所不包，因此当网上信息用特征项来表示时，通常会比面向某一个领域的文献集更多，一般多于 10000 个。由于 SVM 可以有效避免“过学习”的现象发生，不必依赖特征的数量，因此，SVM 对处理这些高维特征空间的情况特别有效。

（2）网络用户的个性化问题和动态性问题突出，不同的用户对信息内容的需求不同，因此网络分类体系需要面向不同用户。网上信息的更新速度非常快，需要对网络信息分类的类别不断进行增删和调整，使得网络信息分类体系具有动态性的特点。由于SVM 只需要较小的训练样本空间，因此，进行类别体系设计时，只需要用较少的样本就可以迅速训练出性能较好的、适合用户要求的分类器。

（3）特征项相关问题。在网上信息处理的传统方法中避免高维输入空间的一个方法是假定大部分特征项是不相关的，但在文本分类中，仅有很少的不相关的特征项。因此，一个好的分类器应该能够利用相关特征项的综合信息，SVM 利用的少数特征向量（支持向量）就具备了这一特点，这也相当于自动地完成了传统方法中比较困难、且易于丢失有用信息的特征项选择工作。

5 预处理系统实现及实验结果分析

5.1 预处理系统设计

Web 文本分类过程主要包括网页预处理、训练过程和分类过程。预处理部分主要功能是为后面的训练和分类过程提供训练文本集和测试文本集。本文的预处理系统包括四个模块：网页清洗模块、分词模块、特征选择模块、文本表示模块。下面分别介绍各个模块的功能和实现过程。

5.1.1 网页清洗模块

网页清洗是网页预处理过程中首先进行的步骤，主要功能是分析网页，去除网页中包含噪音的内容块，保留网页中包含主题信息的内容块。在清洗后的网页上再进行信息提取，不仅可以排除噪音对信息提取的干扰，提高信息提取的准确性，而且可以使网页的结构简单化，提高信息提取的效率。

Web 页面是用 HTML 语言编写的，这种文本含有大量的标记和超链接。目前，基于 DOM 的 HTML 网页解析器很多，主要有 NekoHTML、HTMLParser、HTML Cleaner 等，本文使用 NekoHTML 解析器。NekoHTML 是一个简单的 HTML 扫描器和标签补偿器，使用 NekoHTML 能够解析 HTML 文档，并用标准的 XML 接口来访问其中的信息，而且在扫描 HTML 文件时能修正许多在编写 HTML 文档过程中常犯的错误，增补缺失的父元素、自动用结束标签关闭相应的元素及不匹配的内嵌元素标签。NekoHTML 基于 Xerces 开发，使用时需要与 Xerces 一起使用。

（1）利用 NekoHTML 迭代遍历页面的 DOM 树结构，详细解析步骤如下所示：

输入：一篇 HTML 文档
输出：抽取的网页正文文本
具体实现如下：

```
//新建一个 DOMParser 对象用来解析页面
DOMParser parser = new DOMParser();
//设置网页的默认编码
parser.setProperty("http://cyberneko.org/html/properties/default-encoding","GB1803
0");
//将要清洗的页面 news.htm 转换为输入流
BufferedReader in = new BufferedReader(new FileReader("news.htm"));
//解析输入源
parser.parse(new InputSource(in));
```

```
//得到 DOM 树结构
Document doc = parser.getDocument();
//获得 body 节点，以此为根，遍历页面内容
Node body = doc.getElementsByTagName("BODY").item(0);
//以 body 节点为根，迭代遍历 DOM 树，输出正文文本
System.out.println(TextExtractor(body));
```

（2）迭代遍历函数 String TextExtractor(Node root)返回页面中文本节点的内容，具体的实现如下：

```
String TextExtractor(Node root){
    //如果是文本节点，直接返回节点内容
    if (root.getNodeType() == Node.TEXT_NODE )
        return root.getNodeValue().trim();
    //如果是元素节点，提取元素节点内的文本内容
    if(root.getNodeType() == Node.ELEMENT_NODE) {
        Element elmt = (Element) root;
    //对于以下标签的元素节点，忽略其文本内容
     if (elmt.getTagName().equals("H3") || elmt.getTagName().equals("H4")
        || elmt.getTagName().equals("H5") || elmt.getTagName().equals("A")
        || elmt.getTagName().equals("FORM"))
         return "";
         //得到当前节点下的子节点，迭代遍历节点内的内容
        NodeList children = elmt.getChildNodes();
        StringBuilder text = new StringBuilder();
        for (int i = 0; i < children.getLength(); i++) {
            text.append(TextExtractor(children.item(i)));
     }
     return text.toString();
    }
    return "";    //对其它类型的节点，返回空值
}
```

对于如图 5.1.1 所示的新闻网页，它除了包括主题信息外，还包括导航栏，广告栏，版权信息等噪音内容。经过网页清洗后，得到如图 5.1.2 所示的主题内容。可以看出，

清洗后的网页较好的去掉了页面中的导航栏和广告栏等内容。对清洗后页面提取正文信息后的结果如图 5.1.3 所示。

图 5.1.1 一个未清洗的新闻网页

图 5.1.2 清洗后的网页

图 5.1.3 提取的正文文本

5.1.2 分词模块

分词模块的主要功能是对经过网页正文提取后得到的文本集进行分词，并进行词性标注。本文采用中科院计算所研制的汉语词法分析系统 ICTCLAS（Institute of Computing Technology，Chinese Lexical Analysis System）进行分词。ICTCLAS 的主要功能包括中文分词，词性标注，命名实体识别，新词识别，同时支持用户词典。该系统分词正确率达 98.45%（937 专家组评测），未登录词识别召回率高达 90%，其中中国人名的识别召回率接近 98%。利用分词模块的词性标注功能，可以直接去掉连词、代词、介词、虚词等对文本内容表达意义不大的词和标点符号，提高后面的特征选择过程的效率。词性标注采用中科院计算所一级标注集。

算法如下：

```
//初始化 ICTCLAS，为 ICTCLAS 准备必要的数据
ICTCLAS_Init();
//采用中科院计算所一级标注集
ICTCLAS_SetPOSmap(ICT_POS_MAP_FIRST);
//对字符串 para 进行分词，并将结果返回给 result
char *result=ICTCLAS_ParagraphProcess(para);
//对文件"news.txt"进行分词，并保存为"news.txt"
```

```
ICTCLAS_FileProcess("news.txt","news.txt");
```

分词模块的界面及分词结果如图 5.1.3 所示：

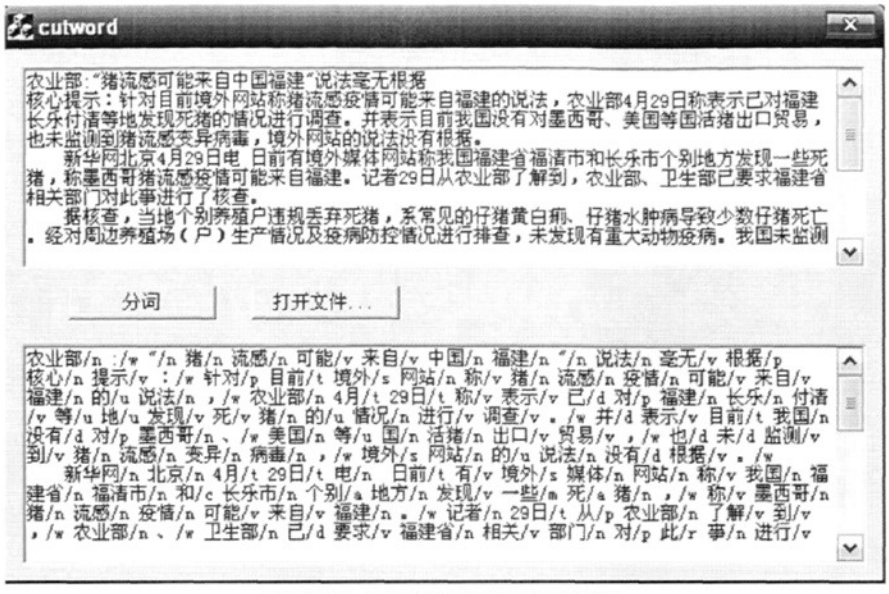

图 5.1.3 分词模块界面及分词结果

5.1.3 特征选择模块

特征选择模块的主要功能就是对训练文本集进行特征选择，降低特征空间的维数。高维特征向量不仅会降低分类系统的处理效率，而且区分度低的特征项的存在会使分类系统的分类正确率降低。本模块提供多种特征评估函数，并且考虑到同义词的处理在特征选择中的作用，在评估函数的基础上引入了同义词统计。根据评估函数得到特征项的评估值后，选择满足某一阈值的特征项，得到特征空间。下面简要介绍本模块的部分数据结构及算法。

（1）set<string> stopwordslist

停用词表。

（2）map<string,set<string> > SimilarDic

同义词词典的数据文件，包括同义词的索引号和同义词组。

（3）map<string,string> WordToID

同义词词典的索引文件，包括单词的索引号和单词。

（4）set<string> AllWords;

保存同义词词典中的所有同义词

（5）map<string,int> FeatureToFiles

保存特征项的文档频度（DF）。

（6）map<string,double> IG_Features

保存特征项的信息增益（IG）。其他保存特征项评估函数值的数据结构这里不再一一列举。

（7）vector<string> SelectedFeatures

保存经过特征选择后得到的特征空间。

算法如下：

（1）生成停用词表：

```
Void CreateStopWords(set<string> stopwordslist){
        FILE *stopword_fp=fopen(保存停用词的文件);
        while(文件未结束){
            读取一个停用词;
            stopwordslist.insert(停用词);
        }}
```

（2）生成同义词词典：

```
Void CreateSynonymyWord(map<string,set<string> > SimilarDic,
                        map<string,string> WordToID，set<string> AllWords){
        打开词典的数据文件;
        while(文件未结束){
            读取同义词集的索引号;
            读取一组同义词组;
            SimilarDic[索引号].insert(同义词);
        }
        打开词典的索引文件;
        while(文件未结束){
            读取同义词集的索引号;
            读取同义词;
            WordToID[同义词]=索引号;
            AllWords.insert(同义词);
        }}
```

（3）对已经分词的文档中的同义词进行合并处理：

```
void ReplaceSimilarWord(char *filepath,map<string,set<string> > &SimilarDic,
                        map<string,string> &WordToID,set<string> &AllWords){
```

```
        打开路径为 filepath 的文件;
        while(文件未结束){
            读取文本中的特征词;
            //判断该特征词是否在同义词词典中
            if(AllWords.count(特征词))
                替换为该特征词在词典中的索引号;
        }}
```

（4）计算文档集中特征项的文档频度：

```
void ComputeDF(map<string,int> &FeatureToFiles){
        //依次读取文档集中的文本
        while(文档集中有文本未读){
            //打开一个文本
            while(文本未结束){
                读取特征词;
                //一个文本中的特征词只统计一次
                ++Feature2Files[特征词];
            }}}
```

（5）计算文档集中特征项的信息增益：

```
void ComputeIG(map<string,int> &FeatureToFiles,map<string,double>
                                                    &IG_Features){
        //依次读取文档集中的文本
        while(文档集中的文件未读完){
                每次读取一个类别的文档集;
                统计一个类别的文档中各特征词的 DF 值;
                //Pt 为特征词 t 在文档集中频率，N 为文档总数
                Pt=FeatureToFiles[特征词]/N;
                //Pct 为类 c 中特征词 t 的概率
                Pct=特征词在一类文档中的 DF 值/N;
                //Nc 为属于类 c 的文档数
                IG_Features[特征词] +=
                    Pt*Pct*log(Pct)+(1-Pt)*(1-Pct)*log(1-Pct)-(Nc/N)*log(Nc/N);
            }}
```

5.1.4 文本表示模块

根据特征选择模块的计算，可以得到的各特征项的文档频度（DF）值和信息增益（IG）值，选取满足一定阈值的特征项，得到文档集的特征集。文本表示模块的主要功能是根据特征选择模块生成的特征集，利用 TF-IDF 公式，分别计算训练文本集和测试文本集中各特征项的权重，将文本表示成向量形式，形成文本集的特征向量空间。文本向量表示过程如图 5.1.4 所示。

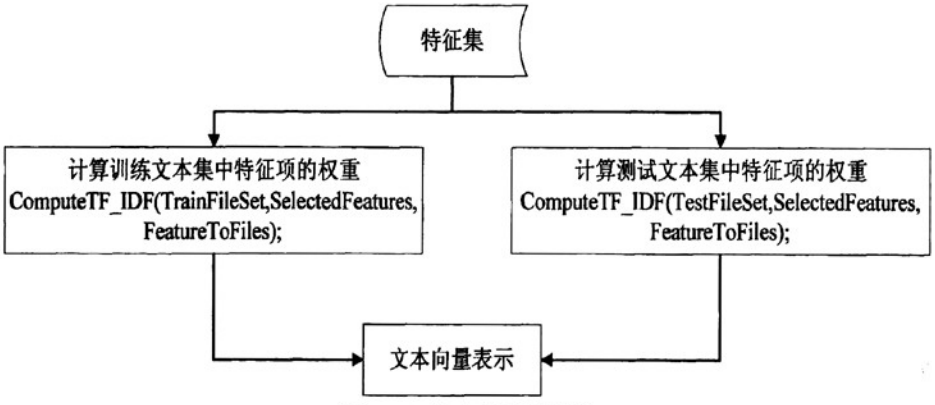

图 5.1.4 文本表示流程图

对于一个训练文档集，其向量表示形式如图 5.1.5 所示：

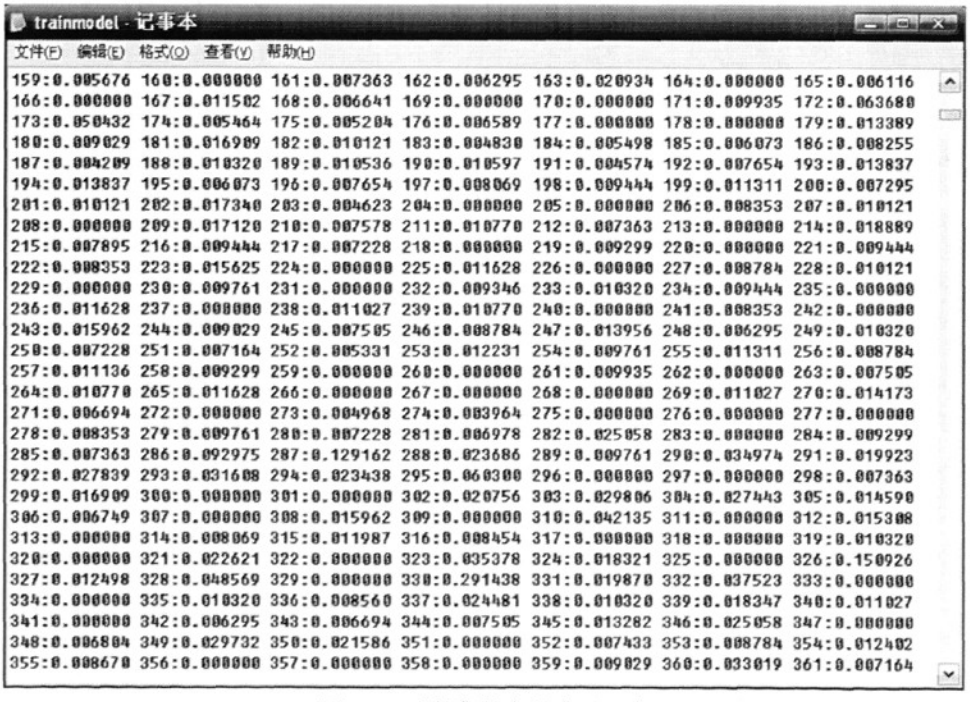

图 5.1.5 文档集的向量表示形式

5.2 试验环境及采用的语料

由于目前在中文 Web 文本分类领域还没有出现标准的中文网页语料库，本文从网上搜集了包括汽车、财经、IT、健康、体育、教育、招聘、军事共 8 个类别的 800 篇网页，每一个类别 100 篇，分为两个部分，其中，600 篇作为训练集，200 篇作为测试集。采用开放测试方式，即采用训练文本集以外的文本集作为测试集。

本文使用台湾大学林智仁博士开发的 LIBSVM-2.88 作为 SVM 分类器。

本文的实验过程如图 5.2.1 所示：

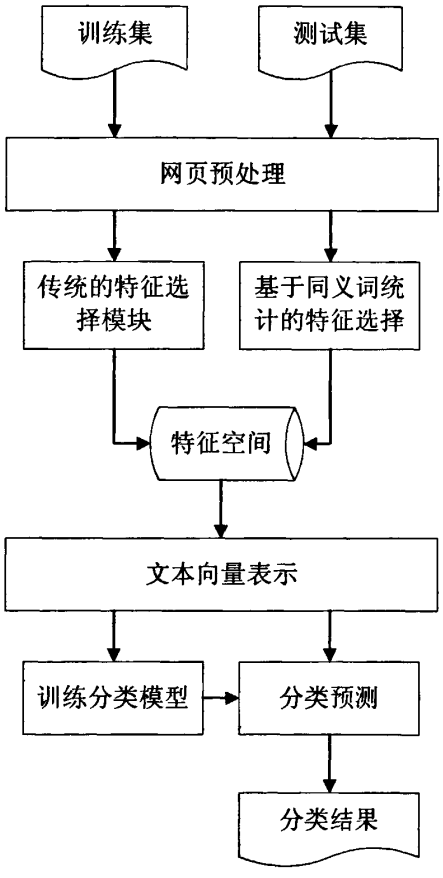

图 5.2.1 Web 文本分类实验过程

5.3 试验结果分析

5.3.1 采用的分类评价指标

本文使用了三个评价指标来评测分类的性能，分别是查准率（Precision），查全率（Recall），分类正确率（Accuracy），综合分类率（又称 F1 测试值），它综合了准确率和召回率两个指标的评估值。各评价指标的定义公式如下：

（1）查准率(Precision)： $Precision = \frac{a}{a+b}$ （5.3.1）

（2）查全率(Recall)： $Recall = \frac{a}{a+c}$ （5.3.2）

（3）分类正确率（Accuracy）： $Accuracy = \frac{a+d}{a+b+c+d}$ （5.3.3）

其中，a—正确地分入该类的文档数；

b—错误地分入该类的文档数；

c—错误地划出该类的文档数；

d—正确地划出该类的文档数。

（4）F_β是另一个把查全率和查准率统一进行考虑的参数，F_β的计算公式：

$$F_\beta = \frac{(1+\beta^2) \times Recall \times Precision}{\beta^2 \times Precision + Recall} \times 100\% \quad (5.3.4)$$

其中，β取值为[0，∞)。β=0 时，F_β就是查准率；β=∞时，F_β就是查全率；β=1 时的F_β在应用中广泛，公式如下：

$$F_1 = \frac{2 \times Recall \times Precision}{Precision + Recall} \times 100\% \quad (5.3.5)$$

5.3.2 试验结果

本文分别针对文档频率（DF）和信息增益（IG）两种特征选择方法进行了分类实验，表 5.3.1 是开放测试条件下使用不同特征集时各个类别的分类结果，其中集合 1 是使用部分特征集合（6256 维向量空间，由 IG 方法得到）得到的分类结果，集合 2 是使用全部特征集合（20347 维向量空间）得到的分类结果。

表 5.3.1 不同特征集合下的查准率、查全率及 F1 值

编号	类别	查准率		查全率		F1	
		集合 1	集合 2	集合 1	集合 2	集合 1	集合 2
1	汽车	1.0	1.0	0.96	0.36	0.9796	0.5294
2	财经	0.9474	0.5294	0.72	0.72	0.8182	0.6102
3	IT	0.88	0.9259	0.88	1.0	0.88	0.9615
4	健康	0.6757	0.7188	1.0	0.92	0.8065	0.8070
5	体育	0.8182	1.0	0.72	0.6	0.7660	0.75
6	教育	0.6957	0.6957	0.64	0.64	0.6667	0.6667
7	招聘	1.0	0.8333	0.84	1.0	0.9130	0.9091
8	军事	0.7586	0.8333	0.88	1.0	0.8148	0.9091

从表 5.3.1 中可以看出，集合 1 的平均 F1 值要比集合 2 的平均 F1 值稍大，同时，对于分类正确率，集合 1 的分类正确率是 83%，而集合 2 的分类正确率是 78%。特征集

合大，分类正确率反而降低了。通过对测试集的分析，测试集的全部特征项只有 10825 个，当使用对训练集进行特征选择得到的全部特征空间时，生成的测试集的向量矩阵十分稀疏，影响了分类结果的正确率。而使用部分特征项却可以得到较好的分类效果，同时也具有较高的效率。

表 5.3.2 使用 DF 和 IG 得到的不同特征集的分类正确率

	DF		IG	
	特征项	Accuracy	特征项	Accuracy
1	544	0.625	542	0.68
2	963	0.785	1002	0.725
3	1569	0.74	1499	0.72
4	1935	0.785	2082	0.805
5	2382	0.775	2596	0.81
6	3017	0.775	3000	0.815
7	3482	0.78	3500	0.795
8	4094	0.815	4095	0.835
9	5104	0.795	4549	0.815
10	6647	0.825	6256	0.83

表 5.3.2 是使用文档频度（DF）和信息增益（IG）两种特征选择方法，根据计算出的 DF 和 IG 的函数评估值，选取满足不同阈值的特征项，得到的不同特征维数下的分类正确率（Accuracy），其分类正确率曲线如图 5.3.1 所示，从图中可以看出，使用 IG 方法进行特征选择得到的分类效果要优于 DF 方法。但是 DF 方法的计算量较小，在进行大规模样本集的特征选择时，时间复杂度低，效率高。同时，在实验过程中，随着特征维数的增加，分类正确率不断提高，但效率也逐渐降低，因此，在进行文本分类时，需要兼顾正确率与效率的要求，反复实验，选取合理的特征空间。

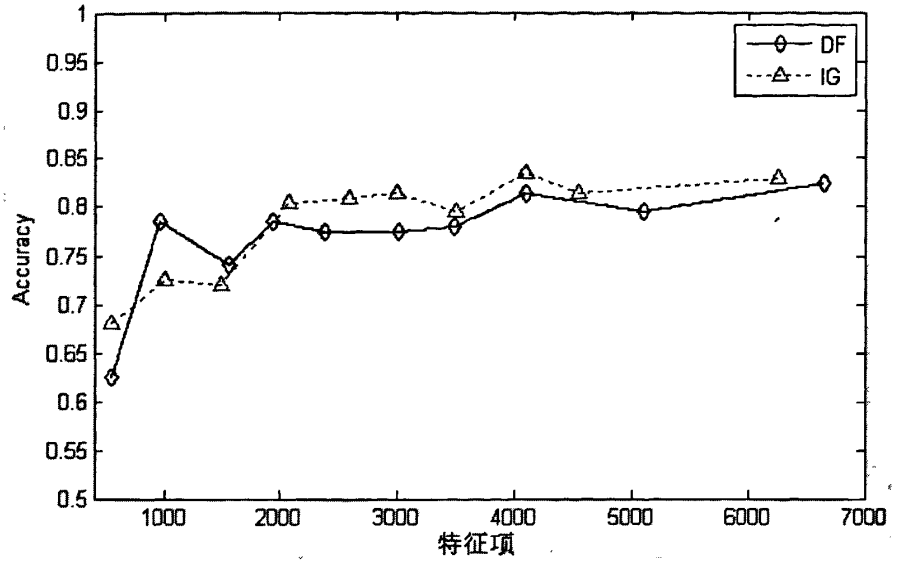

图 5.3.1 不同特征维数下的分类正确率

在中文文本中存在着大量的同义词，这些同义词的存在会分散特征词在文本中的频度。DF 和 IG 两种特征选择评估方法并没有考虑同义词的合并处理问题。因此在前面实验的基础上，我们引入了基于同义词统计的特征选择方法，在特征选择之前，首先利用《同义词林》扩展版的同义词词典，合并同义词，然后再利用公式（3.5.1）的特征选择评估函数重新计算信息增益。这样不仅大大减小了特征维数，减少了约 16.3%，而且通过与没有进行同义词处理时的 IG 特征选择方法相比较，提高了分类正确率，如图 5.3.2 所示。

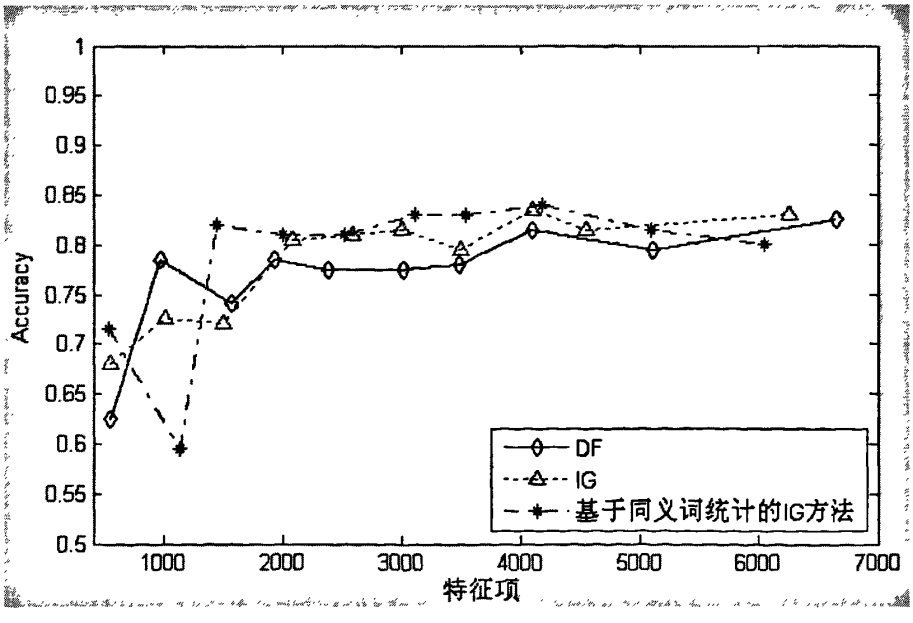

图 5.3.2 基于同义词统计的 IG 特征选择方法与 DF 和 IG 的分类正确率比较

由上图可以看出，进行了同义词统计后，在使用信息增益（IG）进行特征评估时，不同特征维数下，分类正确率基本维持在80%以上，平均分类正确率较之未使用同义词统计进行特征选择时有了很大的提高。另外，同义词词典中同义词的归类是否合适对分类效果也有很大的影响，以上试验结果是在对同义词词典进行了重新整理之后得到的结果，较之未整理前，分类效果有明显提高。

6 结束语

6.1 本文小结

随着 Internet 的日益普及，网上信息的迅速膨胀，使文本自动分类技术成为 Web 数据挖掘领域的一个重要的研究方向。文本分类是自然语言处理的一个重要应用领域，它作为一种信息处理技术，在信息的有效检索、文本数据库、数字图书馆、搜索引擎、信息的过滤等方面有着广泛的应用。

本文对网页预处理、特征选择方法和分类算法进行了研究和实验，取得了一定成果。

（1）利用 DOM 技术对网页进行解析，并实现了网页清洗和正文提取过程；利用 ICTCLAS 对文本进行分词处理，并利用 ICTCLAS 的词性标注功能和停用词表对特征向量进行了初步降维。

（2）对现有的特征选择方法进行了对比研究，并以文档频度、信息增益两种特征评估函数为例，通过实验比较了两种特征评估方法对分类正确率的影响。实验表明，使用信息增益进行特征选择，分类正确率要好一些。但文档频度的特征选择方法的时间复杂度小，有较好的效率。

（3）传统的特征选择方法在进行特征选择时，将每一个特征都单独计算评估函数值，没有考虑文本中的同义词、相关词，在一些文本中同义词、相关词往往表达的是同一个概念，这些词的存在会极大的分散特征词的函数评估值，对后面的分类准确性产生很大的影响。本文在常用的特征选择方法的基础上，引入了基于同义词统计的特征选择，即在进行特征选择之前先进行同义词的处理。实验表明该方法有效地提高了分类正确率。

（4）研究了四种常用的文本分类算法：K 最近邻算法，朴素贝叶斯算法，决策树算法和支持向量机算法，比较了各自的优缺点，并使用 SVM 分类算法进行了 Web 文本分类实验。

（5）对 Web 文本预处理的各个模块进行了编程实现。

6.2 进一步的工作

Web 文本分类涉及自然语言处理的许多问题，尽管本文的研究达到了预期的目的，但仍有许多不足之处，许多地方还需要进一步的研究和优化。今后的工作主要围绕以下几个方面展开：

（1）本文在进行同义词统计时，只对同义词进行了统计，忽略了词性相关的词语的影响，还需要对同义词词典作进一步的改进，在词典中加入相关词，另外，对同义词

的分组也需要作进一步的处理。

（2）在 Web 文本分类领域，英文语料库有多个标准的、开放的分类文档集，可以比较客观地对不同的研究结果进行比较。但对于中文文本的分类，目前还没有一个标准的、开放的语料库，因此构建一个中文语料库也是今后工作的一个重要内容。

（3）训练和测试语料库的大小对分类也有一定的影响，本文的结论是在一个小规模语料的基础上得出的，今后将扩大语料的规模进行测试和研究。

致 谢

在论文完成之际，我要感谢我的导师王玲副教授对我的指导和帮助。导师渊博的知识、严谨的治学风范、认真的工作精神、积极的人生态度是我人生的榜样，在她身上我不仅学到了专业知识，在为人处事方面也受益颇深。在此谨向我的导师致以最崇高的敬意和最诚挚的感谢，感谢老师两年来给予我的认真、耐心的指导和关怀，使我在各方面都取得了长足的进步。

在论文写作过程中，我还得到了王树梅教授的热情指导，她对科研工作的严谨和对学生的认真负责给我留下了深刻的印象。

感谢621教研室的袁瑞红、宦蕾、钱伟、徐波、吴新林、陈睿扬等人在我论文写作期间的帮助，与他们的在学术上讨论和交流，使我受益匪浅。

感谢一直给予我支持的父母和家人。感谢他们在我成长过程中付出的心血，正因为有他们的支持，我才能取得今天的成绩，我的每一步成长都离不开他们辛苦的付出。感谢我的女朋友在写论文期间的给予的支持和鼓励。

谨借此机会向所有关心、支持和帮助我的人致以最诚挚的谢意。

参考文献

[1] 傅欣. 第三代搜索引擎的智能化趋势研究. 信息检索技术, 2002, 6: 28-30

[2] 孙建军，成颖等. 信息检索技术. 北京：科学出版社, 2004

[3] Andrew McCallum. Bow: A Toolkit for Statistical Language Modeling, Text Retrieval, Classification and Clustering. http://www.cs.cmu.edu/~mccallum/bow/

[4] 程军. 基于统计的文本分类技术研究. 中国科学院博士学位论文, 2003

[5] 侯汉清. 分类法的发展趋势简论. 北京：中国人民大学出版社, 1981

[6] 李晓黎, 刘继敏, 史忠植. 概念推理网及其在文本分类中的应用. 计算机研究与发展, 2000, 37(9)

[7] 黄萱菁，吴立德，石崎洋之，徐国伟. 独立于语种的文本分类方法. 2000 International Conference on Multilingual Information Processing, 2000, 37-43

[8] 刁倩, 王永成, 张惠惠, 何骥. 文本自动分类中的词权重与分类算法. 中文信息学报, 2000, 14(3)

[9] 沈记全, 唐菁, 杨炳儒. Web 文本挖掘系统及其分类算法的研究与实现. 计算机工程, 2003, 29(17): 37-39

[10] 陈安, 陈宁, 周龙骧等. 数据挖掘技术及应用. 科学出版社, 2006, 289

[11] Wang Jicheng, Huang Yuan, Wu Gangshan, Zhang Fuyan. Web mining: knowledge discovery on the Web. IEEE SMC '99 Conference Proceedings, 1999, 2: 137-141

[12] Shiqun Yin, Gang Wang, Yuhui Qiu, Weiqun Zhang. Research and Implement of Classification Algorithm on Web Text Mining. Third International Conference on Semantics, Knowledge and Grid, 2007, 446-449

[13] 印鉴, 陈忆群, 张钢. 搜索引擎技术研究与发展. 计算机工程, 2005, 31(14): 54-56, 104

[14] Pascal Soucy, Guy W. Mineau. A Simple KNN Algorithm for Text Categorization. Proceedings IEEE International Conference on Data Mining, 2001, 647-648

[15] 王香港. 中文文本自动分类算法研究. 上海交通大学硕士学位论文, 2007, 17-24

[16] Jiawei Han, Micheline Kamber. 数据挖掘：概念与技术. 北京: 机械工业出版社, 2007, 188-192

[17] Margaret H. Dunham. 数据挖掘教程. 北京：清华大学出版社, 2005, 79-85

[18] 田苗苗. 基于决策树的文本分类研究. 吉林师范大学学报（自然科学版）, 2008, 1: 54-56

[19] A. Basu, C. Watters, M. Shepherd. Support Vector Machines for Text Categorization.

Proceedings of the 36th Hawaii International Conference on System Sciences, 2003.

[20] Christopher J. C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. Kluwer Academic Publishers, Boston, 1998

[21] Fubo Shao, Guoping He, Xin Zhang. An Improved Algorithm for Multiclass Text Categorization with Support Vector. 2008 International Symposium on Computational Intelligence and Design, 2008, 336-339

[22] Pang-Ning Tan, Michael Steinbach, Vipin Kumar. 数据挖掘导论. 北京：机械工业出版社, 2006, 156-168

[23] 焦李成, 刘芳, 侯水平, 刘静, 陈莉. 智能数据挖掘与知识发现. 西安电子科技大学出版社, 2006, 214-230

[24] 邓乃扬, 田英杰. 数据挖掘中的新方法—支持向量机. 科学出版社, 2004, 166-186

[25] 张苗, 张德贤. 多类支持向量机文本分类方法. 计算机技术与发展, 2008, 18(3)

[26] http://www.w3school.com.cn/htmldom/index.asp

[27] Jing Li, C. I. Ezeife. Cleaning Web Pages for Effective Web Content Mining. International Conference on Database and Expert Systems Applications, 2006, 560-571

[28] 刘斌. 基于 Web 的 HTML 网页清洗技术的研究与实现. 华北电力大学硕士学位论文, 2007, 6-20

[29] 张志刚, 陈静, 李晓明. 一种HTML网页净化方法. 情报学报, 2004, 23(4): 387-393

[30] 王琦, 唐世渭, 杨冬青, 王腾蛟. 基于 DOM 的网页主题信息自动提取. 计算机研究与发展, 2004, 41(10): 1786-1792

[31] 张海燕. 基于分词的中文文本自动分类研究与实现. 湖南大学硕士学位论文, 2002, 22-26

[32] 马玉春, 宋瀚涛. Web 中文分词技术研究. 计算机应用, 2004, 24(4): 134-135, 155

[33] 张华平. 汉语词法分析系统. http://www.nlp.org.cn/project/project.php

[34] 刘群, 张华平, 俞鸿魁, 程学旗. 基于层叠隐马模型的汉语词法分析. 计算机研究与发展, 2004, 41(8): 1421-1429

[35] 张华平, 刘群. 基于 N-最短路径方法的中文词语粗分模型. 中文信息学报, 2001, 16(5)

[36] Son Doan. An Effective Feature Selection Using Multi-Criteria in Text Categorization. Proceedings of the Fourth International Conference on Hybrid Intelligent Systems 2004.

[37] 代六玲, 黄河燕, 陈肇雄. 中文文本分类中特征抽取方法的比较研究. 中文信息学报, 2003, 18(1): 26-32

[38] 寇苏玲, 蔡庆生. 中文文本分类中的特征选择研究. 计算机仿真, 2007, 24(3): 289-291

[39] 申红, 吕宝粮, 内山将夫, 井佐原均. 文本分类的特征提取方法比较与改进. 计算机仿真, 2006, 23(3): 222-224

[40] 梅家驹, 竺一鸣, 高蕴琦等. 《同义词词林》. 上海：上海辞书出版社, 1983

[41] 哈工大信息检索实验室. 《同义词词林》扩展版. http://www.ir-lab.org/

[42] 程涛, 施水才, 王霞, 吕学强. 基于同义词词林的中文文本主题词提取. 广西师范大学学报（自然科学版）, 2007, 25(2): 145-148

[43] 曾致远, 张莉. 基于向量空间模型的网页文本表示改进算法. 计算机工程, 2006. 32(3): 134-135, 139

[44] 张东礼, 汪东升, 郑纬民. 基于 VSM 的中文文本分类系统的设计与实现. 清华大学学报（自然科学版）, 2003, 43(9): 1288-1291

[45] Tong Xiaojun, Cui Minggen, Song Guolong. Research on Chinese Text Automatic Categorization Based on VSM. International Conference on Wireless Communications, Networking and Mobile Computing, 2007, 3863-3866

[46] 张冬慧, 孙波, 徐照财, 程显毅. 文本自动分类关键技术研究. 微计算机信息, 2008, 24(2-3): 197-199

[47] http://www.csie.ntu.edu.tw/~cjlin/

[48] 都云琪, 肖诗斌. 基于支持向量机的中文文本自动分类研究. 计算机工程, 2002. 28(11): 137-139

[49] 李晓黎, 刘继敏, 史忠植. 基于支持向量机与无监督聚类相结合的中文网页分类器. 计算机学报, 2001, 24(1): 62-68

[50] 李蓉, 叶世伟, 史忠植. SVM-KNN 分类器—一种提高 SVM 分类精度的新方法. 电子学报, 2002. 5: 745-748

[51] 萧嵘, 王继成, 孙正兴, 张福炎. 一种 SVM 增量学习算法 α-ISVM. 软件学报, 2001, 12(12): 1818-1824

[52] 刘静. 基于web文本挖掘的SVM网页文本分类研究. 东北财经大学硕士学位论文, 2006, 54-55

Web文本分类系统中文本预处理技术的研究与实现

作者： 王之鹏
学位授予单位： 南京理工大学
被引用次数： 1次

本文读者也读过(3条)

1. 吴虎子 中文网页获取及自动分类技术研究[学位论文]2007
2. 刘晓志 文本预处理及其在多类分类中的应用[学位论文]2006
3. 何金凤 基于中文信息检索的文本预处理研究[学位论文]2008

引证文献(1条)

1. 邹丽娜.凌捷 一种基于特征提取的二级文本分类方法[期刊论文]-广东工业大学学报 2012(4)