厦门大学

硕士学位论文

基于最大公共子图的中文Web文本分类研究

姓名：赖兴瑞

申请学位级别：硕士

专业：计算机软件与理论

指导教师：张东站

201106

摘 要

随着网络信息技术的高速发展，Internet 上的 Web 页面数量呈指数增长，如何有效的组织和处理这些海量信息，如何更好地搜索、过滤和管理这些网络资源，成为一个亟待解决的问题。Web 文本挖掘技术就是解决上述问题的一种方法，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 文档的集合中发现潜在的、有价值的知识。Web 文本分类是 Web 文本挖掘的重要技术，是一种快速、有效的组织网上海量信息的关键技术，是 Web 信息处理的基础，有着很高的研究价值和广泛的应用前景。

本文研究的对象是中文 Web 文本，目的是提高 Web 文本分类的精度和速度，主要针对中文 Web 文本的表示以及分类算法进行了深入地探讨。

Web 文档包含大量的与主题内容无关的噪音数据，因此本文提出了一种基于网页分块的主题信息自动提取算法。首先对 Web 文档依据布局标签分块构建文本内容块层次树，然后自底向上遍历层次树，计算每个块节点的语义属性和主题相关度，同时删除主题无关节点，最终通过遍历文本块层次树的最大内容节点路径，提取当前网页的主题信息。实验表明该主题信息提取算法对大多数中文门户网站的主题型网页均有效，适用性比较强。

传统的向量空间文本表示方法不能有效表示文本的结构信息，缺乏对文本特征词条上下文环境的考虑，因此本文探讨了 Web 文档的图表示方法、文档图之间距离度量选择等问题，并在此基础上发展了 KNN 算法，得到了基于最大公共子图的 Web 文本分类算法：MCS-KNN 算法。MCS-KNN 算法为每个 Web 文档生成表示图，通过计算两个 Web 文档表示图之间的相似度来计算两者的相似度，进而计算出待分类文档在训练集中的 K 近邻，根据 K 近邻的所属类别确定待分类文档的类别。实验表明，MCS-KNN 算法分类速度快，精度高，具有比 KNN 算法更优越的分类性能。

关键字： Web 文本分类；主题信息提取；最大公共子图

Abstract

With the rapid development of network information technology, the number of web pages on the Internet grows exponentially. It has become an urgen problem to be solved that how to organize and process these huge amounts of information effectively and how to search, filter and manage these network resources better. Web text mining is one of the solutions of these problems. By borrow ideas from the basic ideas and theoretical methods of data mining, it discovers the potential, valuable knowledge from large semi-structured, heterogeneous collection of web documents. Web text categorization is an important technology of text mining. It's a critical technology for organizing the mass online information, and it's the basis of web information processing. Web text categorization has high research value and broad application prospects.

This paper focused on Chinese web text, the purpose of which is to improve the accuracy and the speed of web text classification. In this paper, the representation of Chinese web text and its classification algorithm were discussed in depth.

For web document usually containing a large amount of noise that is irrelevant to its topic, this paper puts forward an algorithm for automatic extraction of topic information which is based on web partition. First, partition web document according to layout labels to build the hierarchical tree of text blocks. Second, traversing the hierarchical tree bottom-up, we figure out semantic attributes and theme correlativity of each block node, and remove irrelevant nodes. Finally, by traversing through the path of maximum content nodes of the hierarchical tree, we extract the topic information of current web page. Experiments show that this extraction algorithm is effective for topic pages of most Chinese web portals, while it has relatively strong applicability.

For that the traditional method of vector space text representation can not effectively capture the structure information of the text, and lack of consideration on

the context of feature terms, this paper first discussed problems in depth such as the graph representation model of web documents, how to measure the distance between two representation graph of two web documents. And then, we put forwards a web text categorization algorithm based on maximum common graph on the basis of the KNN algorithm, called MCS-KNN algorithm. The MCS-KNN algorithm first generates the representation graph for each web document. We calculate the similarity of two web documents by means of the similarity of their representation graphs. And then, figure out the K nearest neighbor of the unclassified web document among all the train instances, and decide the target class of current unclassified web document in line with categories of the K nearest neighbor. Experiments show that MCS-KNN algorithm has more superior classification performance than KNN algorithm for its fast speed and high precision.

Key Words: web text categorization; topic information extraction; maximum common subgraph

Contents

第一章 绪论

1.1 研究背景与意义

随着网络技术的高速发展，Internet 上网络信息资源呈指数级增长，据估计网页数量每 4 到 6 个月翻一翻，使得 Web 上的信息量以惊人的速度增长，Internet 包含了大量的信息资源，在这些大量的、异质的信息资源中，隐藏着具有巨大潜在价值的知识，所以它已经成为了世界性的图书馆，变成了各行各业人们交流思想、获取信息的平台。但是面对 Web 如此丰富的内容，巨大的数据量，加上万维网动态开放的特点，人们要去快速准确的寻找自己需要的信息不是一件容易的事情，通常需要耗费大量的人力和物力，人们面临着“信息爆炸”而“知识贫乏”的窘境。

由于网络上的信息大多以文本的形式表达，文本提供给用户大量丰富的信息，在这些信息中包含了潜在的、有巨大价值的知识，而面对如此庞大的文本资源，传统的文本分析处理工具已经无法满足现实需要的要求，人们迫切需要研究出有效的方法和手段从大规模文本信息资源中提取符合需要的简洁、精炼、可理解的知识。因此，Web 文本挖掘成为了数据挖掘中一个日益流行而重要的研究课题，是 Web 挖掘研究的重心。

Web 文本挖掘中最关键的技术是 Web 文本分类。网页分类是指在给定的分类体系下，根据网页文本的内容自动确定文本类别的过程，是一种典型的有指导学习的方法。Web 文本分类作为 Web 文本挖掘的重要技术和重要内容，有着越来越重要的意义，已经成为智能信息检索和处理领域的一个新兴和重要的研究方向。

现有的文本分类系统基本都是基于向量空间模型 VSM（Vector Space Model）。向量空间模型的基本思想是把文本表示成向量空间中的向量，采用向量之间的夹角余弦作为文本间的相似性度量。向量空间模型是一种不考虑特征项出现顺序的词袋文本表示模型，虽然带来了计算和操作上的便利，却损失了大量

的文本结构信息，缺乏对特征词条上下文环境的考虑，而这些文本结构信息或者上下文环境在自然语言中是至关重要的。向量空间模型的这些局限，客观上也限制了基于向量空间模型的传统文本分类算法分类性能的进一步提高。

与一般的数据结构相比，图能够表达更加丰富的语义。图结构能够模拟几乎所有事物间的联系，它能应用到半结构化和非结构化的数据挖掘中。建立 Web 文档的图表示模型，使之可以反映文档中特征词条、特征词条间的联系以及特征词条的共现程度等文本信息，并在此模型表示的基础上发展相应的文本分类方法是当前进一步提高文本分类性能的有益探索方向。

1.2 文本分类研究现状

1.2.1 国外研究现状

文本分类的研究可以追溯到上世纪六十年代，早期的文本分类主要是基于知识工程（Knowledge Engineering），通过手工定义一些规则来对文本进行分类，这种方法费时费力，且必须对某一领域有足够的了解，才能写出合适的规则。

到上世纪九十年代，随着网上在线文本的大量涌现和机器学习的兴起，大规模的文本（包括网页）分类和检索重新引起研究者的兴趣。文本分类系统首先通过在预先分类好的文本集上训练，建立一个判别规则或分类器，从而对未知类别的新样本进行自动归类。大量的结果表明它的分类精度比得上专家手工分类的结果，并且它的学习不需要专家干预，能适用于任何领域的学习，使得它成为目前文本分类的主流方法。

1971 年，Rocchio[1]提出了在用户查询中不断通过用户的反馈来修正类权重向量，来构成简单的线性分类器。1979 年，Van Rijsbergen[2]对信息检索领域的研究做了系统的总结，里面关于信息检索的一些概念，如向量空间模型（Vector Space Model）和评估标准如准确率(Precision)、召回率(Recall)，后来被陆续地引入文本分类中，文中还重点地讨论了信息检索的概率模型，而后来的文本分类研究大多数是建立在概率模型的基础上。

1992 年，Lewis 在他的博士论文[3]中系统地介绍了文本分类系统实现方法的

各个细节，并且在自己建立的数据集 Reuters22173（后来去掉一些重复的文本修订为 Reuters21578[4]）上进行了测试。这篇博士论文是文本分类领域的经典之作。后来的研究者在特征降维和分类器设计方面作了大量的工作，Yiming Yang[5]对各种特征选择方法，包括信息增益（Information Gain）、互信息（Mutual Information）、统计量等，从实验上进行了分析和比较。她在 1997 年还对文献上报告的几乎所有的文本分类方法进行了一次大阅兵，在公开数据集 Reuters21578 和 OHSUMED[6]上比较了各个分类器的性能，对后来的研究起到了重要的参考作用。

1995 年，Vipnik[7]基于统计理论提出了支持向量机（Support Vector Machine）方法，基本思想是寻找最优的高维分类超平面。由于它以成熟的小样本统计理论作为基石，因而在机器学习领域受到广泛的重视。Thorsten Joachims[8]第一次将线性核函数的支持向量机用于文本分类，与传统的算法相比，支持向量机在分类性能上有了非常大的提高，并且在不同的数据集上显示了算法的鲁棒性。至今，支持向量机的理论和应用仍是研究的热点。

在支持向量机出现的同时，1995 年及其后，以 Yoav Freund 和 Robert E. Schapire 发表的关于 AdaBoost 的论文[9]为标志，机器学习算法的研究出现了另一个高峰。Robert E. Schapire 从理论和试验上给出 AdaBoost 算法框架的合理性。其后的研究者在这个框架下给出了许多的类似的 Boosting 算法，比较有代表性的有 Real AdaBoost[10]，Gentle Boost[11]，Logit Boost[12]等。这些 Boosting 算法均已被应用到文本分类的研究中，并且取得和支持向量机一样好的效果。

到目前为止，国外的文本自动分类研究已经从最初的可行性基础研究经历了试验性研究进入到了实用化阶段，并在邮件分类、电子会议、信息过滤等方面取得了较为广泛的应用。

至今技术成果主要表现在以下几个方面[13-14]：

1. 向量空间模型研究日益成熟
2. 对特征项的选择进行了较深入的研究
3. 较完整的分类算法的研究和比较
4. 存在比较标准的语料库

5. 较为规范的测试方法

1.2.2 国内研究现状

1981 年，侯汉清教授对于计算机在文本分类工作中的应用做了探讨，并介绍了国外计算机管理分类表、计算机分类检索，计算机自动分类、计算机编制分类表等方面的概况[15]。此后，我国陆续研究产生了一些文本分类系统[16]，其中比较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，中国科技大学的范众等人，复旦大学和富士通研究中心的黄营著、吴立德等人，上海交通大学的刁倩、王永成等人，都在文本分类算法方面取得了突出的成就[17-18]。

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，特别是对中文文本分类研究是从上世纪 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，使得中文文本分类和英文文本分类有很多不同，难度也就更大。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，而由于汉语是一种分析型语言，语义分析在汉语研究中起着举足轻重的作用，其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历史大致经历了三个阶段：国外研究成果引进阶段、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[19-20]。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研究策略上，结合中文文本的特点，继而形成中文文本分类研究体系。

汉语分词是中文文本分类的一个基础环节。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[21]。目前国内分词系统所采用的或者正在研究的方法基本上分为三类：机械分词、基于理解的分词和基于统计的分词[22-23]。

在很长一段时间内，中文文本分类的研究没有公开的数据集，使得分类算法难以比较。现在一般采用的中文测试集有：北京大学建立的人民日报语料库、清华大学建立的现代汉语语料库等。

其实一旦经过预处理将中文文本变成了样本矢量的数据矩阵，那么随后的文本分类过程和英文文本分类相同，也就是随后的文本分类过程独立于语种。因此，当前的中文文本分类主要集中在如何利用中文本身的一些特征来更好地表示文本样本。

总而言之，尽管机器学习理论对于文本分类的研究起了不可低估的作用，在这之前文本分类的研究曾一度处于低潮，但是文本分类的实际应用和它自身的固有的特性给机器学习提出新的挑战，这使得文本分类的研究仍是信息处理领域一个开放的、重要的研究方向。

1.2.3 中文 Web 文本分类研究现状

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。

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。在对中文 Web 文本进行分类的过程中，包括几个关键步骤：文本预处理、分词、权重计算、特征提取、特征降维，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。

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，其中有基于 VSM（Vector Space Model）的向量距离法、贝叶斯分类算法、KNN 分类算法、SVM（Support Vector Machine）分类算法、决策树分类算法和神经网络分类算法等等[24-26]，近些年还出现了基于粗糙集理论的文本分类算法[27]和一些结合多种方法的混合分类方法[28]。

国内对 Web 文本分类的研究还没有到达一个成熟的阶段，其中还存在一些有待进一步研究的问题：

(1) 分词是影响文本分类的重要因素之一，分词的速度和准确率与最终的分类结果密切相关。尤其是 Web 上不断出现新词汇，对分词理论的创新和词典的构造都提出了较高的要求。就中文文档分类而言，分词是一项非常复杂的工作，分类系统一般都比较复杂和庞大，分词速度慢，且准确度不高，因此，研究无须词典支持、领域独立的文本分类系统无疑具有重要价值，这使得文档分类系统成为真正意义上的通用系统。

(2) 目前还没有发现“最佳”的特征选择方法，针对中文 Web 文本分类的组织特点，需要结合特定的特征选择，因此在使用不同分类算法时如何选择最佳的特征选择方法也是我们需要深入研究的问题。

(3) 由于中文文本分类起步晚和中文不同于英文的特性，目前中文 Web 文本分类还没有标准的开放的文本测试集，各研究者大多使用自己建立的文本集进行训练和测试，其分类结果没有可比性，不利于交流和提高。一般地，训练文档集应该是公认的经人工分类的语料库。国外文档研究都使用共同的测试文档库，这样就可以比较不同分类方法和系统的性能，而就中文文档分类而言，各研究者使用自己建立的训练文档库进行测试，测试结果没有可比性，这一现状应当引起国内文本处理研究者的重视。

(4) 将自然语言理解和处理技术、语义 Web 概念、Agent 技术和机器翻译等

技术应用于 Web 自动文本分类中，进一步解决中文文本分类的难点，提高文本分类的智能化水平。

(5) 目前存在多种成熟的文本分类算法，大部分分类系统都是应用某一种分类算法，分类性能受到制约。

1.3 研究内容

本文研究的对象是中文 Web 文本，目的是提高 Web 文本分类的精度和速度，主要针对中文 Web 文本的表示以及分类算法进行了深入地探讨，主要内容包括：

- 针对 Web 文档包含大量的与主题内容无关的噪音数据，提出了一种基于网页分块的主题信息自动提取算法，准确提取 Web 文档中的主题信息，为后续 Web 文本文类性能的提高奠定基础。
- 针对传统的向量空间文本表示模型的局限，探讨了 Web 文档的图表示方法，并在此基础上发展了 KNN 方法得到基于最大公共子图的 Web 文本分类方法 MCS-KNN（Maximum Common Subgraph Based K-Nearest Neighbour）。

1.4 本文结构

本文共分五章，文章结构及各章主要内容组织如下：

第一章为绪论，介绍了中文 Web 文本分类的研究背景、研究意义以及国内外研究现状，概述论文的研究内容和组织结构。

第二章系统介绍了 Web 文本分类的相关内容。首先介绍了数据挖掘、Web 挖掘以及 Web 文本挖掘的定义、主要内容及其挖掘流程，然后系统地介绍了 Web 文本分类的相关概念和技术，包括 Web 文本预处理、文本表示、特征选取等，并详细介绍了几种文本分类算法，包括 Rocchio 方法、KNN 和 SVM 等。最后，介绍了文本分类算法的评价指标。

第三章提出了一种基于网页分块思想的主题信息自动提取方法，以准确提取分类任务中训练集和测试集 Web 文档的主题信息，降低网页噪声对文本分类的

干扰。

第四章提出了一种基于最大公共子图的中文 Web 文本分类方法 MCS-KNN，并通过实验证明方法的有效性。

第五章是总结与展望，总结了本文对中文 Web 文本分类的研究工作，包括所取得的成绩与存在的不足，提出以后的主要研究方向。

第二章 Web 文本分类概述

2.1 数据挖掘

数据挖掘（Data Mining，DM）是指从大量数据中提取或“挖掘”知识[29]。数据挖掘又被称为数据库知识发现（Knowledge Discovery in Databases，KDD）。它通常是指从数据源（如数据库、文本、图片、万维网等）中探寻有用的模式（Patterns）或知识的过程。这些模式必须是有用的、有潜在价值的，并且是可以理解的。数据挖掘是一门多学科交叉的学问，包括及其学习、统计、数据库、人工智能、信息检索和可视化。

数据挖掘的过程大致可以分为[30]：问题定义、数据收集与预处理、数据挖掘实施，以及挖掘结果的解释与评估。如图 2.1 所示。

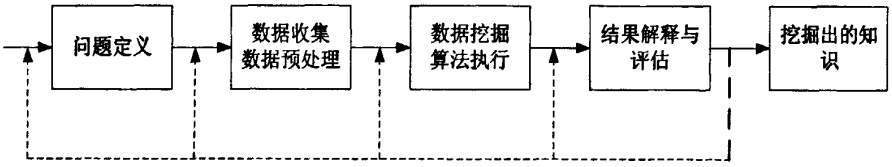

图 2.1 数据挖掘的过程

- 问题定义

数据挖掘是为了从大量数据中发现有用的令人感兴趣的信息，因此发现何种知识就成为整个过程中第一个也是最重要的一个阶段。在这个过程中，必须明确数据挖掘任务的具体要求，同时确定数据挖掘所需要采用的具体方法。

- 数据收集与预处理

这个过程主要包括：数据选择、数据预处理和数据转换。

数据选择的目的就是确定数据挖掘任务所涉及的操作数据对象（目标数据），也就是根据数据挖掘任务的具体需求，从相关数据源中抽取出与数据挖掘任务相关的数据集。

数据预处理通常包括消除噪声、遗漏数据处理、消除重复数据、数据类型转换等处理。

数据转换的主要目的就是消减数据集合的特征维数（简称降维），即从初始特征中筛选出真正与挖掘任务相关的特征，以便有效提高数据挖掘效率。

- 数据挖掘实施

根据数据挖掘任务及已有的方法（分类、聚类、关联等）选择数据挖掘实施算法。

- 结果解释与评估

实施数据挖掘所获得的挖掘结果，需要进行评估分析，以便有效发现有意义的知识模式。此外还需要对所发现的模式进行可视化，将挖掘结果转换为用户易懂的另一种表示方法。

整个数据挖掘过程是可迭代的，一般都要通过多轮迭代才能获得最终结果。随后这些结果就可用来促进现实世界的各项工作。

2.2 Web 挖掘

随着万维网和文本文件规模的不断增大，Web 挖掘和文本挖掘正变得越来越重要，越来越流行。

O. Etzioni[31]指出 Web 挖掘是运用数据挖掘技术从 Web 文档和服务中自动地发现和抽取信息。也就是对文档的内容、可利用资源的使用以及资源之间的关系进行分析，从 Web 数据中发现潜在的有用信息和先前不知道的知识的整个过程。它是以从 Web 上挖掘知识为目标，以数据挖掘、文本挖掘、多媒体挖掘为基础，并综合运用计算机网络、数据库与数据仓库、人工智能、信息检索等技术，将传统的数据挖掘技术与 Web 结合起来的一门新兴学科。

Web 挖掘从数据挖掘发展而来，在研究方法上有很多相似之处。但是，Web 挖掘与数据挖掘相比有许多独特之处。首先，Web 挖掘的对象是大量、异质、分布的 Web 文档。其次，Web 在逻辑上是一个由文档节点和超链接构成的图，因此 Web 挖掘所得到的模式可能是关于 Web 内容的，也可能是关于 Web 结构的。

如图 2.2，依据在挖掘过程中使用的数据类别，Web 挖掘任务可以被划分为三种主要类型：Web 结构挖掘、Web 内容挖掘和 Web 使用挖掘[32]。

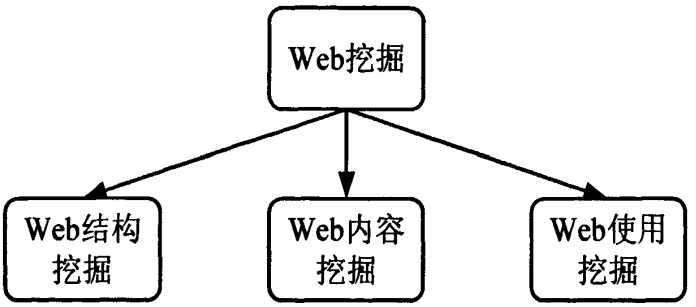

图 2.2 Web 挖掘的分类图

- Web 结构挖掘是指挖掘 Web 潜在的链接结构模式，找到隐藏在一个页面之后的链接结构模型，该模型可用于网页重新分类，寻找相似的网站，获得有关不同网页间相似度及关联度的信息，有助于用户找到指向相关主题的权威站点。在 Web 结构挖掘领域最著名的两个算法是：PageRank 算法和 HITS 算法。
- Web 内容挖掘是指对 Web 页面进行挖掘，从文本、图像、声音、视频、动画等各种形式的网络资源中发现所需的特定类型消息，以实现 Web 资源的自动检索。
- Web 使用挖掘是指自动发现和分析模式，这些模式来自于收集的点击流和相关数据或用户与一个或多个网站互动的结果。其目标是捕捉、建模并分析用户与网站交互的行为模式和模型。所发现的模式经常被表示成有着共同需求或兴趣的一群用户频繁访问的页面、对象或者资源的集合。分析不同 Web 站点的访问日志可以帮助人们理解用户的行为和 Web 结构，从而改进站点的结构，或为用户提供个性化的服务。

Web 挖掘的处理流程[33]包括如下四个步骤：资源发现、信息选择和预处理、模式发现、模式分析。

- 资源发现

网络爬虫在线收集 Web 文档、网站的日志等数据，并从中得到有用的数据。

- 信息选择和预处理

剔除 Web 资源中无用信息并将信息进行必要的整理，如 Web 文档中自动去除广告连接、去除多余格式标记、英文单词的词干提取、停止词的过滤、中文分词等。

- 模式发现

自动进行模式发现。可以在同一个站点内部或多个站点之间进行，以自动发现 Web 站点的共有模式。

- 模式分析

验证、解释上一步骤产生的模式，并进行可视化。

2.3 Web 文本挖掘

Web 中的信息多样化，其中最主要的信息资源是文本，因此 Web 文本挖掘成为 Web 挖掘的一个重要研究领域。Web 文本挖掘可以对 Web 上大量文档集合的内容进行总结、分类、聚类、关联分析，以及利用 Web 文档进行趋势预测等。

文本总结是指从文档中抽取关键信息，用简洁的形式对文档内容进行摘要或解释。这样，用户不需要浏览全文就可以了解文档或文档集合的总体内容。文本总结在有些场合十分有用，例如，搜索引擎在向用户返回查询结果时，通常需要给出文档的摘要。

文本分类是指按照预先定义的主题类别，为文档集合中的每个文档确定一个类别。这样，用户不但能够方便地浏览文档，而且可以通过限制搜索范围来使文档的查找更为容易。

文本聚类与分类的不同之处在于，聚类没有预先定义好的主题类别，它的目标是将文档集合分成若干个簇，要求同一簇内文档内容的相似度尽可能地大，而不同簇间的相似度尽可能地小。

关联分析是指从文档集合中找出不同词语之间的关系。Brin 提出了一种从大量文档中发现一对词语出现模式的算法，并用来在 Web 上寻找作者和书名的出现模式，从而发现了数千本在 Amazon 网站上找不到的新书籍[34]。

分布分析与趋势预测是指通过对 Web 文档的分析，得到特定数据在某个历史时期的情况或将来的取值趋势。Feldman 等人使用多种分布模型对路透社的两万多篇新闻进行了挖掘，得到主题、国家、组织、人、股票交易之间的相对分布，揭示了一些有趣的趋势[35]。

需要说明的是，Web 上的文本挖掘和通常的平面文本挖掘的功能和方法比较类似，但是，Web 文档中的标记，例如<Title>，<Heading>等蕴含了额外的信息，

我们可以利用这些信息来提高 Web 文本挖掘的性能。

Web 文本挖掘的流程一般包括：Web 文本收集与预处理、特征的表示和提取、Web 文本挖掘、挖掘结果评价、信息表示与信息导航。Web 文本挖掘的一般流程如图 2.3 所示。

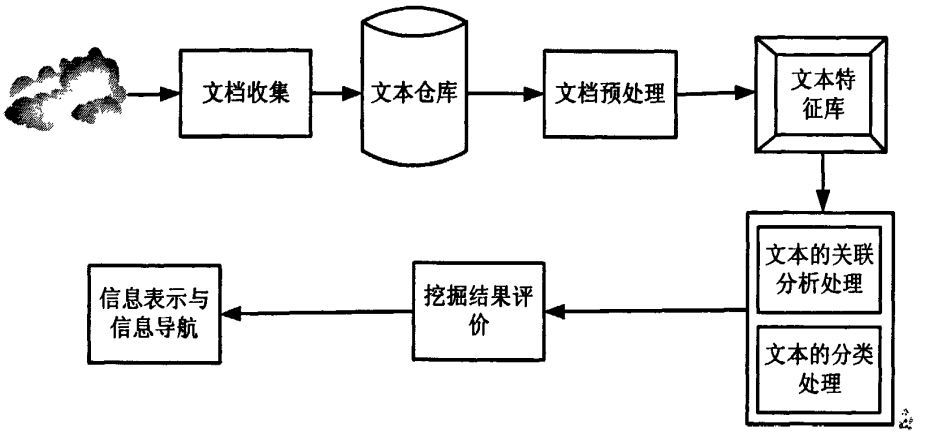

图 2.3 Web 文本挖掘的流程

- Web 文本收集与预处理：程序(Robot)能自动利用网页中超链接来收集相关主题的网页。为了提高数据的质量，可以对文本作一些预处理，如清除图像文件、脚本程序等。
- 特征的表示和提取：从 Web 文本中抽取代表其特征的元数据，这些特征可以用结构化的形式保存，作为文档的中间表示形式。特征的提取是为了减少特征向量的维度。
- Web 文本挖掘：Web 文本挖掘是对大量 Web 文档进行分类、聚类、关联分析以及对 Web 文档进行自动文摘的过程。
- 挖掘结果评价：对挖掘得到的知识或者模式进行评价，将符合一定标准的知识或者模式呈现给用户。
- 信息表示与信息导航：将反馈的结果用可视化的方式进行显示，同时对用户提供信息导航功能，从而在极大的程度上方便用户浏览和获取信息。

在 Web 文本挖掘中，文本的特征表示与提取是挖掘工作的基础；文本分类、文本聚类是两种最重要也是最基本的挖掘功能。

2.4 Web 文本分类

2.4.1 文本分类概念

文本分类（Text Categorization，TC）是为每个元组对$\langle d_j,c_i\rangle \in D\times C$赋予一个布尔值的过程，其中$D$是文本域，$C=\{c_1,...,c_{|C|}\}$是预定义的类别集。若文本$d_j$被分到$c_i$类，那么$\langle d_j,c_i\rangle$就为 1，否则为 0。进而分类任务可以形式化地表示为寻求未知的目标函数h'：$D\times C\rightarrow\{1,0\}$，$h'$说明待分类文档该如何被分类，该函数被称为分类器（classifier），有时也称为分类规则（ruler）、分类假设（hypothesis）或分类模型（model）。函数h'与实际的分类模型h：$D\times C\rightarrow\{1,0\}$应该尽可能一致，这种一致性可以通过具体的分类模型评估指标来确定。

若文本集中的每个文本必须属于也只能属于一个类别，即只能为文本指定一个类标号，那么这种分类称为单标号文本分类（single-label text categorization），也称为非重叠分类。

若文本集中的每个文本可以属于一个或多个类，那么这种分类称为多标号文本分类（multi-label text categorization），也称重叠分类。

利用分类器分类文档有两种不同的模式：类别中心分类（Category Pivoted Categorization，CPC）和文档中心分类（Document-Pivoted Categorization，DPC）。类别中心分类是指给定类别c_i，发现c_i类的所有文档。文档中心分类则是指给定文档d_i，发现文档d_i所属的所有类别。

一般来说，文本分类过程如图 2.4 所示：

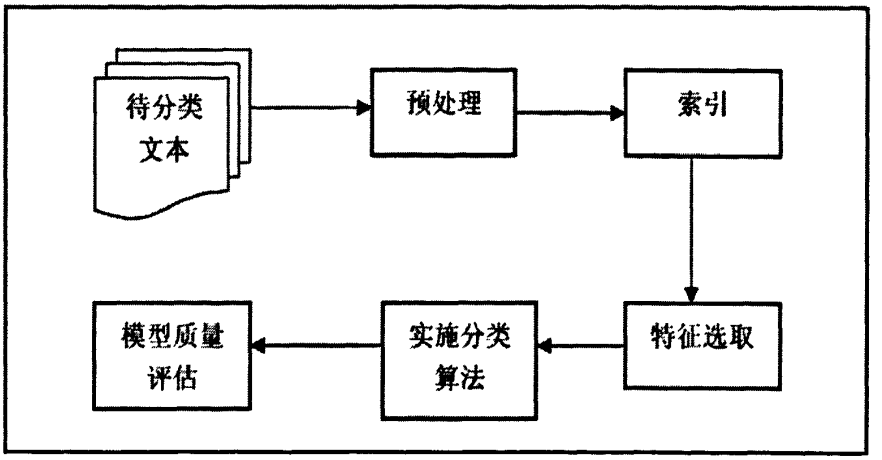

图 2.4 文本分类过程

Web 文本分类的概念和普通文本分类是一致的，普通文本分类的技术是 Web 文本分类的基础，但是 Web 文档有其特殊性。通常我们所获得的 Web 文本都是带有 HTML 标记的网页，必须去除 HTML 标记，才能使用文本分类技术。同时，Web 文档中丰富的 HTML 标记蕴含了额外的信息，为 Web 文本分类性能的提高提供可能。

2.4.2 Web 文本预处理

文本预处理是文本分类的前提，如何将一个普通的文本转化为可以处理的数据是研究的重点。同时为了适应各种不同的情况，文本预处理也会有所不同，其主旨是将文本处理为适合分类器的数据集。一般来说中文 Web 文本预处理主要包括：中文分词、去除停用词、处理 HTML 结构化信息等问题。

（1）中文分词

中文分词技术是中文分类中特有的概念，由于英文单词之间有空格将各个词自然分开，而中文则是根据词和词之间的概念来区分，所以中文分词是中文文本分类预处理的关键一步。

汉语句子中词与词之间的边界标志是隐含的，为其添加明显的词语边界标志，即分词。目前针对中文分词已经提出许多方法，主要有两类：基于词典分词和基于统计分词。

- 基于词典分词

基于词典的分词方法，又称机械匹配分词法，其基本思想是：建立一个词库，其中包含可能出现的词。对给定的待切分的汉字串 S，按照某种确定的原则切分 S 和子串，若该子串与词库中的某个词条匹配，则该子串是词，继续分割剩余的部分，直到剩余部分为空；否则该子串不是词，回溯重新对 S 的子串进行匹配。根据切分子串的方向不同，机械匹配分词法分为正向匹配法和逆向匹配法。使用该方法，词典的涵盖程度决定了词汇切分的准确率，要做到这一点很不容易。此外，该方法无法正确切分出词表中未收录的新词，不具备自适应性。

- 基于统计分词

基于统计的分词方法也称无词典分词方法，它是基于这样一个语言信息：在文章中，相邻间字同时出现的次数越多，就越有可能构成一个词，所以字与字相邻共现的频率或概率能够较好地反映它们成为词的可信度。通过设定一个阈值，使得当可信度高于某一个阈值时，便认为两个字可能构成了一个词，采用无词典分词，得到的非真实词条会非常多，识别精度较差，时空开销大。

在实际应用的分词系统中通常都采用词典和统计相结合的方法：使用一部基本的分词词典（常用词词典）进行串匹配分词，同时利用统计方法来消除歧义和未登录词识别，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算法结合上下文识别生词、自动消除歧义等优点。

（2）去除停用词

停用词是指一些常见常用但对文本分类没有贡献的词或字符，这类词主要包含三类，其一：常见名词没有特殊的含义，如“我们”、“他们”、“什么”等；其二：语气词、助词等，例如：“吗”、“呀”、“的”等；上述这些词的特点是没有明确的含义，同时在文本中多次出现，对文本分类没有贡献或者贡献很小。而第三类停用词是标点符号和非法字符，因为这些字符不能表达文本的内容信息，对文本分类没有实质性的意义。因此，去除停用词就是把这三类词或字符过滤掉，进而改善最终文本表示和分类效果。

（3）处理 HTML 结构化信息

互联网是信息的载体，构成互联网的 Web 网页则是载体的基本单元。Web 网页不同于普通的文本，多数是由半结构化的 HTML 语言构成，如何提取出 Web 网页中相对有用的信息是 Web 文本分类中重要的一环。大体上有以下几种处理

方法，首先将 Web 网页中所有文字都认为是有用信息，这就造成可能包含大量的噪音信息；第二种情况是将诸如网页标题、字体加粗文字和链接信息等看成是有用信息，而其他的信息则过滤掉，这种方法相对于第一种而言做出了相对的平衡，可以达到较好的效果；最后一种是根据一定的规则判断出要保留哪些信息，对于特定的网页分类而言，有特定的规则，这种方法无疑是预处理中最好的，但该方法无法普遍应用。

2.4.3 文本表示

计算机不具有人类的智能，不能像人一样根据自身理解能力对文章产生模糊认识，因此在进行文本分类之前，首先应将文本转化为易被计算机理解的形式，然后通过具体的文本分类方法对文本类别进行划分。

文本的表示既要使其方便计算机的处理，又要能够有效表达文本内容。大量研究表明，向量空间模型[36]（Vector Space Model，VSM）是一种适合于大规模语料的文本表示模型。

向量空间模型由哈佛大学的 G Salton 提出。向量空间模型实际上是用于文本表示的统计模型。该模型中，文本空间被视为一组正交特征向量组成的向量空间。

向量空间模型将任意一个文本表示成空间向量的形式，并以特征项作为文本表示的基本单位。向量的各维对应文本中的一个特征项，而每一维本身则表示了其对应的特征项在该文本中的权值。权值代表了特征项对于所有文本的重要程度，也反映了该特征项对文本内容的反映能力。

对于中文文本而言，可以选择字、词、词组、短语、句子或者句群作为文本特征项。特征项的选择要以处理速度、精度和存储空间等方面的要求为原则。

由于词汇是文本的最基本表示项，在文本中出现的频率较高而且呈现一定的统计规律，不同的特征词就可以区分不同内容的文本，因此在向量空间模型中一般选择词或词组作为特征项。

文本的向量空间表示可以描述为如下的公式：

$$v(d_i)=(\omega_1(d_i),\omega_2(d_i),...,\omega_n(d_i)) \qquad (2.1)$$

其中，n 表示文本特征抽取时所选用的特征项数目，$\omega_j(d_i)$ 表示第 j 个文本特征项在文档 d_i 中的权值。

在文本向量空间表示中，每一个特征项都有一个权值，权值的大小反映了特征项对于文本的重要程度，即一个特征项在多大程度上能够将所在文本与其他文本区分开来。

假定特征 t 在文档 k 中的词频为 f_{tk}，权值为 d_{tk}，N 表示文档集中的文档数，M 表示所有文档的词汇量，n_t 表示特征 t 在整个文档集中的出现频率，则常见的权值计算方法包括如下算法：

- 词频权值法

词频权值法是根据特征词在文档中的出现频率来确定其重要程度的一种加权方法，即 $d_{tk}=f_{tk}$。

- TF/IDF 权值法

TF（term frequency）表示特征词在某文档中的出现频率；IDF（inverse document frequency）表示特征词在整个文档集中的出现频率。TF/IDF 定义文档 k 中词 t 的权值与其在文档中的出现频率成正比，而与其在整个文档集中的出现频率成反比，它的具体定义形式如式（2.2）所示：

$$d_{tk}=f_{tk}\times\log\frac{N}{n_t} \tag{2.2}$$

TF/IDF 方法是目前研究和应用最为广泛的一种权值法。

- TFC 权值法

TF/IDF 权值法虽然最常用，但它没有考虑文档长度对权值的影响。TFC 权值法在 TF/IDF 方法的基础上利用文档长度对其进行规范化，定义形式如式（2.3）所示：

$$d_{tk}=\frac{f_{tk}\times\log\frac{N}{n_t}}{\sqrt{\sum_{t=1}^{M}(f_{tk}\times\log\frac{N}{n_t})^2}} \tag{2.3}$$

向量空间模型构造简单，系统容易实现，但其最主要的缺点在于文档表示时，维数过高，而且在整个模型中均假定特征之间互相独立，这使得词所在的上下文以及相邻词之间的潜在关系均被丢失。

文本相似度是用来衡量文本之间相似程度大小的一个统计量。文本相似度一

般定义为界于[0，1]之间的一个值。如果两文本之间相似度为 1，则说明两文本对象完全相同，如果相似度为 0，则说明两文本没有相似之处。

在向量空间模型中，文本相似性的度量有内积法、余弦法和距离函数法等。

- 内积法

通常在文档向量空间中，最常使用的相似度的计算公式就是两个文档向量之间的“内积”运算，内积定义为：

$$SIM(d_i,d_j)=\sum_{k=1}^{n}\omega_{ki}\times w_{kj} \tag{2.4}$$

- 余弦法

$$SIM(d_i,d_j)=\cos(d_i,d_j)=\frac{\sum_{k=1}^{n}\omega_{ki}\times\omega_{kj}}{\sqrt{\left(\sum_{k=1}^{n}\omega_{ki}^{2}\right)\left(\sum_{k=1}^{n}\omega_{kj}^{2}\right)}} \tag{2.5}$$

上述各算法中，$SIM(d_i,d_j)$表示文本d_i和d_j之间的相似程度，ω_{ki}和ω_{kj}分别表示文档d_i和d_j的第k个特征项的权值，n为文档特征项数。SIM值越大表示两个文本越相似，SIM值越小则表示两个文本区别越大。

- 距离函数法

可以使用两文本之间的距离来度量文本之间的相似程度。常使用的距离公式簇如下：

$$DIS(di,dj)=\left[\sum_{k=1}^{n}\left|\omega_{ki}-\omega_{kj}\right|^{p}\right]^{\frac{1}{p}} \tag{2.6}$$

这些公式计算的是向量d_i和d_j在向量空间中的距离。其中ω_{ki}表示第k个特征项在第i个文档中的权值，参数p决定了选择了何种距离计算。

当$p=1$时

$$DIS(d_i,d_j)=\sum_{k=1}^{n}\left|\omega_{ki}-\omega_{kj}\right| \tag{2.7}$$

当$p=2$时

$$DIS(di,dj)=\sqrt{\sum_{k=1}^{n}\left(\omega_{ki}-\omega_{kj}\right)^2} \tag{2.8}$$

这就是欧氏距离，也就是向量空间中的直线距离。

在距离函数法中，$DIS(d_i,d_j)$ 表示文档 d_i 和 d_j 之间的欧氏距离，n 为文档特征项数，ω_{ki} 和 ω_{kj} 分别表示文档 d_i 和 d_j 的第 k 个特征项的权值。DIS 的值越大，文本之间的相似度越小，DIS 的值越小，文本之间的相似度越大。

2.4.4 特征选择

特征选择是指从一组特征中挑选出一些最有用的特征以达到降低特征空间维数的目的。换句话说，特征选择就是从一组数量为 W 的特征中选择出数量为 w（$w<W$）的一组特征来。常用的特征选择方法有：文档频率（DF）、互信息（MI）、信息增益（IG）、χ^2 统计量（CHI）等，实验[37][38]表明，IG 和 CHI 是两种相对不错的方法。

1. 文档频率

文档频率（Document Frequency，DF）是最简单的一种特征选择方法，但也是很有效的一种方法。该方法通过文档频次进行特征选择，就是将文档频次小于某一阈值的词删除，从而降低特征空间的维数。DF 评估函数的理论假设是稀有单词要么不含有用信息，要么有用信息太少而不足以对分类产生影响，所以可以删去。显然它在计算量上比其它评估函数要小得多，在实际运用中它的效果也不错，但 DF 也有缺点，因为稀有单词可能在某一类文本中并不稀有，而且包含着重要的标志信息。

2. 互信息

互信息（Mutual Information，MI）在统计语言模型中被广泛采用。互信息是信息熵的引申概念，是对两个随机事件的相关性度量。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。将低于特定阈值的特征从原始特征空间中移除，保留高于阈值的特征项。词条和类别之间的互信息计算公式如下：

$$MI(t,c_i)=\log\frac{P(t,c_i)}{P(t,c)} \tag{2.9}$$

其中 $P(t,c_i)$ 表示特征项 t 在类别 c_i 中出现的概率，$P(t,c)$ 表示特征项在所有文档中出现的概率；如果在现有的测试集上定义类 c，若 A 用来表示特征项 t 和类 c 同时发生的次数；B 表示特征项 t 发生而类 c 不发生的次数；C 表示特征项 t 不发生而类 c 发生的次数；N 是指总体的文档数目，则特征项 t 和类 c 之间的 MI 值计算公式如下：

$$MI(t,c)=\log\frac{A\times N}{(A+C)\times(A+B)} \tag{2.10}$$

互信息的缺点是受临界特征的概率影响较大，它经常倾向于选择稀有单词。然而对于文本分类而言，出现次数较多的单词比出现次数较少的单词具有更大的作用，所以互信息在文本分类中表现较差。

3. 信息增益

信息增益（Information Gain，IG）基于信息论（Information Theory）的熵（Entropy）概念。

$$entropy(D)=-\sum_{i=1}^{m}P(c_i)\log P(c_i) \tag{2.11}$$

$$\sum_{i=1}^{m}P(c_i)=1 \tag{2.12}$$

公式（2.11）为熵的计算公式，其中 $P(c_i)$ 表示 c_i 类在数据集 D 中的概率，也即 c_i 类的在 D 中的实例数目除以 D 中实例的总数。熵可以作为数据混杂度或者混乱度的衡量指标，数据越混杂，熵值越大。

信息增益是特征项作为分类属性前后，数据集的混杂度变化情况。计算公式如下：

$$\begin{aligned}IG(t)=&\left(-\sum_{i=1}^{m}P(c_i)\log P(c_i)\right)-\\&\left(P(t)\left(-\sum_{i=1}^{m}P(c_i|t)\log P(c_i|t)\right)+P(\bar{t})\left(-\sum_{i=1}^{m}P(c_i|\bar{t})\log P(c_i|\bar{t})\right)\right)\\=&-\sum_{i=1}^{m}P(c_i)\log P(c_i)+\\&P(t)\sum_{i=1}^{m}P(c_i|t)\log P(c_i|t)+P(\bar{t})\sum_{i=1}^{m}P(c_i|\bar{t})\log P(c_i|\bar{t})\end{aligned} \tag{2.13}$$

其中 $P(c_i)$ 表示任意一篇文本属于第 i 类的概率，$P(t)$ 表示特征项 t 在文本集中出现的概率。$P(\overline{t})$ 表示除 t 外的特征项在文本集中出现的概率。$P(c_i|t)$ 表示任意一篇包含 t 的文本属于第 i 类的概率。$P(c_i|\overline{t})$ 表示不包含 t 的文本属于第 i 类的概率。

4. χ^2 统计量

该方法度量词条 t 和文档类别 c 之间的相关程度，并假设 t 和 c 之间符合具有一阶自由度的 χ^2 分布；词条对于某类的 χ^2 统计值越高，它与该类之间的相关性越大，携带的类别信息也越多。令 N 表示训练语料中的文档总数，c 为某一特定类别，t 表示特定的词条，A 表示属于 c 类且包含 t 的文档频数，B 表示不属于 c 类但是包含 t 的文档频数，C 表示属于 c 类但是不包含 t 的文档频数，D 是既不属于 c 也不包含 t 的文档频数。则 t 对于 c 的 CHI 值计算公式如下：

$$\chi^2(t,c)=\frac{N\times(AD-BC)^2}{(A+C)\times(B+D)\times(A+B)\times(C+D)} \qquad (2.14)$$

2.4.5 文本分类方法

文本分类方法的选择是决定文本分类效果好坏的关键。用于文本分类的方法有多种，我们应该根据实际情况，对各种方法进行综合考虑，然后选择合适的方法进行文本处理。

训练和分类方法是文本分类系统的核心，目前存在多种基于向量空间模型的训练和分类方法，如 Rocchio 方法[39]、K-近邻（K-Nearest Neighbours，KNN）方法[40][41]、贝叶斯（Bayes）方法[42][43]，支持向量机（Support Vector Machine，SVM）算法[44-49]、决策树（Decision Tree，DT）方法[50][51]和神经网络（Neural Networks，NN）方法[52][53]等。

这里只介绍几种主要的文本分类方法。

- Rocchio 方法（相似度计算法）

该方法的主要想法是根据待分类文档向量与每个类别中心向量的距离来判断文档的类别属性。

首先，根据算术平均为每类文本集生成一个代表该类的中心向量；然后，在

新文本到来是，确定新文本的向量，计算该向量与每类中心向量间的距离，也就是所谓“相似度”；然后，判定文本属于与文本距离最近的类。具体步骤如下：

（1）计算每类文本集的中心向量。通过对每个类别中所有训练文本向量进行简单的算术平均即可得到该类别的中心向量。

（2）将新文本表示为特征向量。直接将新文本映射到文档特征空间上。

（3）计算新文本的特征向量与每个类别文本中心向量间的相似度。其计算方式如下：

$$SIM(d_i,d_j)=\frac{\sum_{k=1}^{n}\omega_{ki}\times\omega_{kj}}{\sqrt{(\sum_{k=1}^{n}\omega_{ki}^2)(\sum_{k=1}^{n}\omega_{kj}^2)}} \tag{2.15}$$

其中，ω_{ki}，ω_{kj} 分别表示为文档 d_i 和 d_j 的第 k 个特征项的权值，n 为文档特征项数。

（4）比较上面所得出的相似度的值，找出其中最大值，将文本分到最大相似度所对应的文本类别中。

Rocchio 分类法优点在于方法实现简单，分类的复杂度不高，但是缺点也很明显：该方法直接使用特征空间的特征分布，受训练文档中的噪声影响很大，同时对某些特征空间非线性可分情况没有处理能力。

- KNN 分类方法

KNN 文本分类方法是模式识别中最重要的非参数法之一。该算法的分类方式是通过查询已知类似文档的分类情况，来判断新文档与已知文档是否属于同一类别。

其基本思想是：给定一个新文本，由算法搜索模式空间即训练文本集，找出与新文本距离最近（最相似）的 K 篇文本，最后根据这 K 文本所属的类别判别新文本所属类别。

KNN 文本分类方法的具体步骤如下：

（1）根据特征项集合描述训练文本向量。

（2）新文本到达后，将新文本表示为文本向量模式。

（3）在训练集中选出与新文本最相似的 K 个文本，相似度采用公式（2.15）计算。

（4）根据与新文本最近的 K 个邻居的类属关系，计算新文本属于每类的权重。权重计算公式如下：

$$p(x,c_j)=\sum_{i=1}^{K} SIM(x,d_i)y(d_i,c_j) \qquad (2.16)$$

其中，x 为新文本特征向量，d_i 为 x 的第 i 个邻居，$SIM(x,d_i)$ 为新文本与第 i 个邻居的相似度，而 $y(d_i,c_j)$ 为类别属性函数，如果 d_i 属于 c_j，那么函数值为 1，否则为 0。

（5）比较各类权重，将文本分配到权重最大的类别。

整体来说，KNN 方法的优点在于：

（1）该方法简单、有效，另外，方法重新训练的代价较低（包括类别体系的变化和训练集的变化，在 Web 环境和电子商务应用中是很常见的）。

（2）方法的计算复杂度不高，计算的时间和空间复杂度都在训练集规模的线性变化空间内，在文本分类场合，这种算法复杂度相对较低。

KNN 方法的缺点在于：

（1）存放所有的训练样本，需要较多的空间开销；

（2）每次分类都要计算待分类样本和所有训练样本之间的距离，当训练样本数量很大时，需要较多的时间开销；

（3）难以找到一个最优的 K 值，一般要采用不同的 K 值进行一系列实验才能确定取哪个值比较好。

● 支持向量机（SVM）算法

支持向量机是由 Vapnik 和他的合作者共同提出的一套学习算法，是统计理论的一种实现方法，它较好地实现了结构风险最小化（Structural Risk Minimization，SRM）原则。SVM 通过引入核函数，将样本向量映射到高维特征空间，然后在高维空间中构造最优分类面，获得线性最优决策函数。SVM 可以通过控制超平面的间隔度量来抑制函数的过拟合；通过采用核函数巧妙解决了维数问题，避免了学习算法计算复杂度与样本维数的直接相关；也由于 SRM 原则的使用，SVM 具有良好的推广能力。

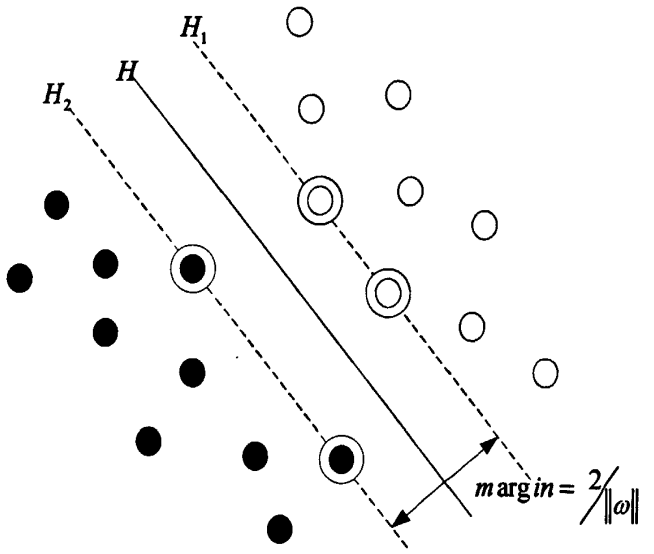

图 2.5 二维线性可分情况下的最优分类面

SVM 算法的主要思想是针对两类分类问题，在高维空间中寻找一个超平面作为两类样本的分割，以保证最小的分类错误率。它通过非线性变换，将输入向量映射到高维空间 H，并在 H 中构造最优分类超平面，从而达到最好的泛化能力。其基本思想可用图 2.5 的两维情况来说明。

假设对于样本集 (x_1,y_1)，(x_2,y_2)，…，(x_n,y_n)，$x_i \in R^n$，$y_i \in \{-1,1\}$，支持向量机寻找一个最优超平面，使它的分类间隔最大。图 2.5 中实心点和空心点分别表示两类样本，H 为把两类样本没有错误地分开的分类线，H_1，H_2 分别为过两类样本中离分类线最近且平行于分类线的直线，它们之间的距离叫做分类间隔。设分类线方程为 $x \bullet \omega + b = 0$，$y \in \{-1,1\}$，$y_i[(\omega \bullet x_i)+b] \geq 0$，$i=1$，…，$n$，则分类间隔为 $\frac{2}{\|\omega\|}$，使分类间隔最大等价于使 $\|\omega\|$ 最小，因此满足此条件且使 $\frac{\|\omega\|^2}{2}$ 最小的分类线就是最优分类线，H_1，H_2 上的训练样本点为支持向量。支持向量机通过对分类间隔最大化来控制泛化能力，这正是支持向量机的特点。将二维空间推广到高维空间，最优分类线就成为最优分类面，即最优超平面。

支持向量机的主要优点有：

（1）对高维、稀疏数据不敏感；

（2）更好的捕捉了数据的内在特征；

（3）准确率较高。

支持向量机的主要缺点有：

（1）对于非线性问题，核函数选择较为困难；

（2）分类结果召回率较低。

2.1.6 分类性能评估

如何判断文本效果的好坏是文本分类的重要课题，在文本分类研究领域，较常用的评价准则有三种，分别是准确率，召回率，和 F-Score。

◆ 准确率（Precision）

准确率指一个文本被分类器分到某一类别而且的确属于这个类别的概率，它一般指所有判断的文本中与人工分类结果吻合的文本所占的比率，其数学公式表示如下：

$$\text{准确率}=\frac{\text{分类正确的文本数}}{\text{实际分类的文本数}} \tag{2.17}$$

◆ 召回率(Recall)

召回率又称查全率是指一个文档应该属于某一类别而分类器也确实将其分到该类别的概率；它一般指人工分类应有的文本中与分类系统吻合的文本所占的比率，其数学公式表示如下：

$$\text{召回率}=\frac{\text{分类正确的文本数}}{\text{应有文本数}} \tag{2.18}$$

◆ F-Score

准确率和查全率反映是从两个不同的角度说明文本分类效果，因此人们提出了新的评判准则，即 F-Score。F-Score 由 Van Rijsbergen 在 1979 年首先提出，该指标将查全率和准确率两者结合为一个指标，两者相对重要性用参数 β 来刻画，F_β 的计算公式如下：

$$F=\frac{(1+\beta^2)\times \mathrm{Precision}\times \mathrm{Recall}}{\beta^2\times \mathrm{Precision}+\mathrm{Recall}}\times 100\% \tag{2.19}$$

其中，β 的取值不同代表着不同的涵义，β 取值范围为[0,∞]；当 β=0 时，F_β 就是代表准确率，当 β=1 准确率与查全率在评判的过程中都起到了很大的作用，

当 $\beta=\infty$ 时，F_β 就是查全率。因此根据 β 的不同取值，对于评判的准则有不同的意义，在实际中广泛应用的是 $\beta=1$，综合两种方法来考察分类器，计算公式如下：

$$F=\frac{2\times \Pr ecision\times \operatorname{Re} call}{\Pr ecision+\operatorname{Re} call}\times 100\% \qquad (2.20)$$

以上定义的三个评价指标都是针对某一个类别的，所以这些指标只能代表局部意义；但有时为了在全局意义上来评价分类器，就必须考虑所有的类别。目前有两种方法来综合所有类别的评价指标，即宏平均(Macro Average)和微平均(Micro Average)。

2.5 本章小结

本章对 Web 文本分类的相关概念、具体流程和常用方法进行了较为详细的介绍。文本分类的主要流程包括文本预处理、文本表示、特征选择、实施分类算法、质量评估等步骤。

文本预处理是将文本转化为能够被计算机所处理的表示方法，中文 Web 文本预处理包括文本去噪、分词和停用词处理等步骤，本章对这些做了简要的介绍。本章对文本的特征选择方法做了较为详细的阐述，介绍了文档频率、互信息、信息增益、χ^2 统计量等特征选择方法。

本章重点介绍了 Rocchio 分类法、KNN 算法和支持向量机三种文本分类算法，对这三种文本分类算法的基本思想、步骤和优缺点进行了较为详细的阐述。Rocchio 分类法实现简单，分类复杂度不高；SVM 算法基于统计学习理论和结构风险最小化原理，有扎实的理论基础，准确率较高；而 KNN 算法虽然是一种懒散的学习方法，但是其算法的特殊性和思想简单易于实现的优点使其也得到了广泛应用并取得了不错的效果。

最后，本章简要介绍了文本分类算法评价指标，主要包括准确率、召回率和 F_1 测试值等。

第三章 基于网页分块的主题信息自动提取

网页主题信息通常湮没在大量的无关文字和 HTML 标记中，给应用程序迅速获取主题信息增加了难度。为了更好地服务于 Web 文本分类，本文引入分块思想，构造相应文档的文本块层次树，遍历层次树，删除冗余及没有内容的分块节点，最终将网页中主题无关的噪音数据清除。实验表明，本方法能够正确提取主题信息，通用性较强，且易于实现。

3.1 相关知识

3.1.1 HTML 简介

```
<html>
    <head>
      <title>Here comes the DOM</title>
    </head>
    <body>
      <h2>Document Object Model</h2>
      <p>
          This is a simple
          <code>HTML</code>
          page to illustrate the
          <a href="http://www.w3.org/DOM/">DOM</a>
      </p>
    </body>
</html>
```

图 3.1 HTML 文本示例

HTML(Hyper Text Markup Language)即超文本标记语言，是目前网络上应用最为广泛的语言，也是构成网页的语言。它不是编程语言，而是一种标记语言，包含一整套的标记标签，借以描述网页。

HTML 标记标签通常被称为 HTML 标签(HTML tag)，指的是由尖括号包围的关键词，比如<html>。HTML 标签通常成对出现，比如和。标签对中

的第一个标签是开始标签，第二个标签是结束标签。开始和结束标签也称为开放标签和闭合标签。

HTML 文档即网页，包含有 HTML 标签和纯文本。一个网页的所有标签和纯文本包含在<html>与</html>之间。<html>标签之下有<title>、<body>两个标签。<title>标签设置网页标题，且嵌套包含其他设置网页编码、关键词、网页描述等信息的标签。一个网页的所有可显示内容均包含在<body>标签内。图 3.1 是一个简单的 HTML 文档，包含有网页的常见标签<html>、<title>、<body>等。在该文档中，设置网页的标题为 Here comes the DOM，<body>标签下的内容则是该文档的重要部分，为该网页的显示内容。

Web 浏览器读取 HTML 文档，并以网页的形式显示出它们。图 3.2 为浏览器读取图 3.1 的 HTML 文档后的显示结果。浏览器不会显示 HTML 标签，而是使用标签来解释页面的内容。

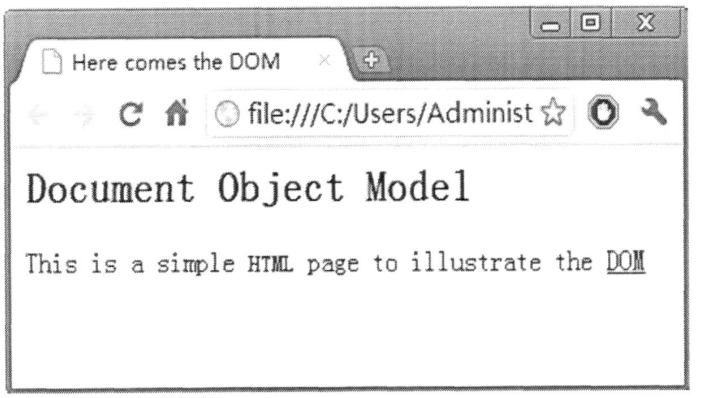

图 3.2 浏览器显示结果

3.1.2 文档对象模型

按照 W3C 的定义，文档对象模型（Document Object Model，DOM）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。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。

一般来说，HTML 文件由标题(Title)、头(Head)、段落(paragraph)、超链接(Hyperlink)以及其它各种组件组成，并且组件在文件中的存储顺序与显示顺序相同。DOM 通过对 HTML 文件的解析，生成一个文件的树型结构，称为文件的树型逻辑结构或逻辑结构。

树型结构可以准确地描述元素的相对位置关系，很适合描述 Web 的半结构化数据。从 HTML 文档到标记树的转化可以通过 HTML 的语法分析器来完成。文件的树型逻辑结构与 Web 文档一一对应，可以相互转化，这种结构便于计算机处理，是用来表示 HTML/XML 文档的一种数据结构。

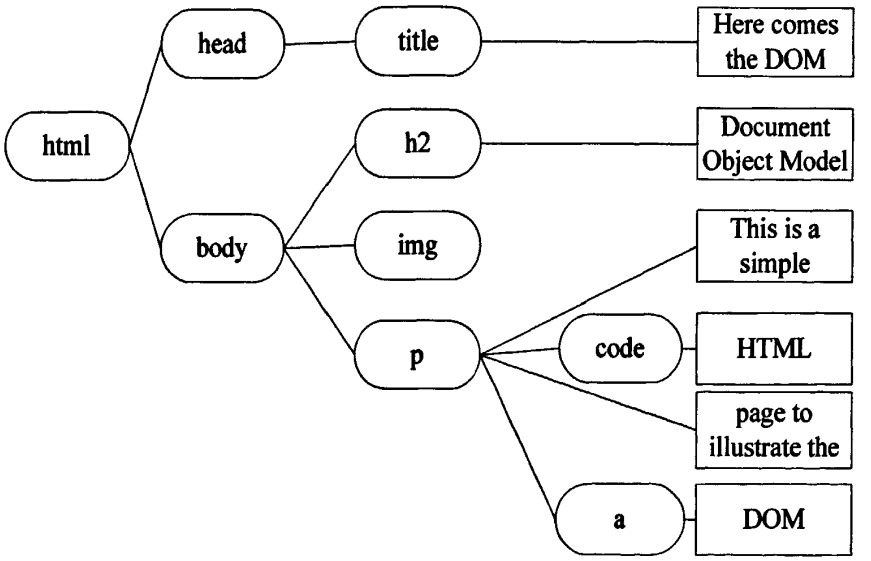

图 3.3 DOM 树

DOM 在进行文件解析时，将 HTML 文件看成一棵树。<HTML>作为树的根，而 HTML 文件的其它组件被看作树中的节点(Node)；节点可以作为父节点来包含子节点，也可以作为其它节点的子节点；同一层的节点称为兄弟节点。

图 3.3 是从图 3.1 的 HTML 页面构建的 DOM 树（或标签树）。内部节点（用椭圆表示）是 HTML 标签，<html>标签是根节点。叶节点（用方块表示）对应文本段。

3.2 相关研究

Web 信息抽取是一个关于从网页中抽取目标信息的问题。其中包含两大问题：即从自然语言文本中和网页的结构化数据中抽取信息。本文考虑面向 Web 内容的信息提取方法，此类方法的目标不是提取细粒度的结构数据，而是提取主题内容或兴趣区域。

Finn 等人[54]将 HTML 文档看做字符和标签组成的序列，在字符集中的区域提取文字。这种方法仅适合主题文字集中的网页，如果段落间有表格或链接等标签丰富的结构，就不能有效处理。

Buyukkokten 等人[55][56]提出了 STU（Semantic Textual Unit）模型，STU 对应网页中的块(Block)，将网页分割为平行的 STU。算法采用了分块思想，减少了定位时间，但是它们都改变了源网页的结构和内容，而且没有提取出主题信息，保留了无关的文字和链接。

FU YAN[57]在发现网页正文内容块时是以内容块中链接文本的特征和多少来判断的，这种算法很难将一些非链接的冗余信息如版权信息滤除。

RIPB（Recognizing Informative Page Blocks）算法[58]将网页分割成若干视觉上的内容块，并将结构相似的内容块聚类，最后通过内容块中图片和文字所占的比重来确定哪个内容块为网页正文内容块。该算法在冗余信息中文字占得比较多时效果不好。

KIM Y[59]通过 HTM（HTML Tree Matching Algorithm）算法来计算文档树的相似度并删除噪声结点。该算法需要经过复杂的计算，而且对于同一模板生成的网页信息抽取效果不好。

黄玲[60]提出一种基于网页分块的正文信息提取算法。该算法首先识别和提取网页正文内容块，然后利用正则表达式和简单的判别规则滤除内容块中的 HTML 标记和无关文字。该算法可能无法准确定位网页正文内容块，且经过提取处理的结果网页中仍然有不少冗余信息。

为此，本文提出了一种实现简单、通用性较强的网页正文信息提取方法。该方法首先根据 HTML 布局标签将整个网页分块，建立当前网页的文本块层次树，根据计算得到的文本块节点的语义属性进行剪枝操作。最终，通过遍历文本块层次树的最大内容节点路径，提取当前网页的主题信息。

3.3 主题信息提取算法

文本网页可分为两种类型：主题型网页、目录型网页。主题型网页通常通过成段的文字描述一个或多个主题。虽然主题型网页也会出现图片和超链接，但这些图片和超链接并不是网页的主体[61]。目录型网页通常提供一组相关或者不相关的链接。本文所研究的主题信息提取是指主题型网页中成段文字的提取。

在主题型网页中，正文信息是成堆出现的，从视觉上看是处在一个内容块中，称为网页正文内容块。而我们算法的目标便是正确识别出每一个主题型网页的网页正文内容块。

3.3.1 基本定义

定义 3.1 DOM（Document Object Model）即文档对象模型，是 W3C 制定的标准接口规范。HTML 文档被解析后，转化为 DOM 树，树的每个结点是一个对象。DOM 模型不仅描述了文档的结构，还定义了结点对象的行为，利用对象的方法和属性，可以方便地访问、修改、添加和删除 DOM 树的结点和内容。

定义 3.2 将 HTML 文档转化 DOM 树的过程称为解析(parse)。解析后，HTML 文档的内容包含在树结点中，对 HTML 文档的处理可以通过对 DOM 树的操作实现。

定义 HTML 文档集合：$P=\{p|p$ 是 HTML 文档$\}$，DOM 树集合 $T=\{t|p\in P$，t 是 p 的 DOM 树$\}$。解析器以 HTML 文档作为输入，清洗(tidy)HTML 文档，生成 DOM 树。故解析过程可以定义为 $\text{Parser}(P)=T$。

定义 3.3 HTML 网页可以划分为不同区域，每个区域称为块（Block）。Block 是页面中在内容和显示上独立的、闭合的矩形区域。每个块可以进一步划分为子块。

定义 3.4 STU（Semantic Textual Unit）即语义文本单元，每个 STU 对应一个块，STU 嵌套构成 STU 树。STU 树模型扩展了 STU 模型，具有强大的语义描述能力。

定义 DOM 树结点集合：$D=\{d|t\in T$，d 是 t 的结点$\}$。块由 DOM 树结点组成，任一块 B_i，有 $STU_i=\{d|t\in T$，d 是 t 的结点$\wedge d\in B_i\}$，STU 树结点集合 $S=\{STU_i\}$。故 $S\subseteq D$。

定义 3.5 STU-DOM 树是具有语义属性的 DOM 树。在 STU-DOM 树中，具有语义属性的结点称为 STU 结点。

由于 STU 树模型具有与源 HTML 网页相对应的树状结构，利用 HTML 与 DOM 树的映射关系，可以将 STU 树与 DOM 树结合：向 DOM 树的某些结点添加描述语义的属性，生成的 DOM 树称之为 STU-DOM 树，树中具有语义属性的结点称为 STU 结点。这样，STU-DOM 树兼有 DOM 树和 STU 树的结构和语义，避免了使用额外的存储空间，简化了处理流程，而且使提取后的网页具有与源网页一致的结构和内容，可靠性和可扩展性较高。

定义 3.6 将 DOM 树转换为 STU-DOM 树的过程称为分块(partition)。

3.3.2 算法描述

信息提取系统分为 5 个部分：HTML 解析器、过滤器、分块器、语义分析器和剪枝器。解析器（HTML parser）将 HTML 文档转化为 DOM 树，本系统采用 Jsoup 解析器。过滤器（filter）从 DOM 树中删除无关结点。分块器(partitioner)向 STU 结点添加语义属性，将 DOM 树转化为 STU-DOM 树，语义属性值由语义分析器(semantic analyser)计算。剪枝器（pruner）从 STU-DOM 树中删除无关链接列表和没有内容的块，最后输出只含有主题信息的 HTML 文档。

- 过滤和分块

过滤和分块是将 DOM 树转化为 STU-DOM 树的过程。过滤器从 DOM 树的根结点开始，递归地遍历 DOM 树，删除所有无关结点，遇到分块结点时调用分块器，向该结点添加语义属性，使该结点成为 STU 结点，当 STU 结点的语义属性值满足剪枝条件时，调用剪枝器处理该结点。无关结点通常是图片（img）、脚本（script）等，无关结点的标签列表是系统配置的一部分。分块结点决定了分块的粒度，分块粒度过粗或过细都将导致抽取结果不完整或保留多余信息。本文采用 DIV、TABLE、TD 标签结点作为分块结点，其语义属性是 *contentlength* 和 *linkcount*，属性值由语义分析器计算。

- 语义分析

算法采用的语义信息是块中非链接文字总数（字符数）和链接总数，在 STU-DOM 中对应子树中的非链接文字总数和链接总数，分别用 *contentlength* 和 *linkcount* 属性表示。非链接文字指不在链接上的文字，一个块的非链接文字总数

可以代表它的内容。

该策略基于对 HTML 网页的抽样分析结果，经分析发现与主题无关的块总是含有大量无关链接和极少非链接文字。我们利用这一特征计算主题相关度。

定义 3.7 一个 STU-DOM 结点的主题相关度表示该结点与 HTML 文档主题的关联程度。

相关度（correlativity）由块内链接和内容决定，其计算公式可以表达为

$$Correlativity(STU_i)=\frac{ContentLength(STU_i)}{LinkCount(STU_i)} \tag{3.1}$$

$$LinkCount(STU_i)=\sum_{j=1}^{N}LinkCount(STU_{ij}) \tag{3.2}$$

$$ContentLength(STU_i)=\sum_{j=1}^{N}ContentLength(STU_{ij}) \tag{3.3}$$

其中，STU_{ij}表示STU_i的第 j 个子树，$LinkCount(STU_i)$是STU_i的 *linkcount* 属性值，用其所有子树中的链接数之和计算；$ContentLength(STU_i)$是STU_i的 *contentlength* 属性值，用其所有子树中的非链接文字的字符数之和计算。

语义分析器用上述算法对 STU 结点进行语义分析，计算 *contentlength* 和 *linkcount* 属性值。

- 剪枝

定义 3.8 主题相关度阈值为TC_m，如果$Correlativity(STU_i)\geq TC_m$，称为$STU_i$主题相关。如果$Correlativity(STU_i)<TC_m$，则称$STU_i$主题无关。

定义 3.9 规定 *STU* 结点的非链接文字至少为 C_m 个字符，即 *contentlength* $\geq C_m$；如果 *contentlength*$<C_m$，称 *STU* 为空或没有内容。

剪枝器判断 *STU* 结点是否主题无关，是则删除当前 *STU* 节点。如果一个 *STU* 为空，即 *contentlength*$<C_m$，则删除该结点。根据这一算法，可以删除主题无关链接列表和没有内容的块。

- 正文内容块提取算法

经过上述环节的处理，页面中已经不存在主题无关链接列表和没有内容的块，但可能还存有诸如网站版权信息之类的噪音内容。因此，我们仍然需要进一

步的处理。处理思路为：逐层遍历 STU 树，查看同一层 STU 节点中的非链接文本数，沿着拥有最大非链接文本数的 STU 节点（最大内容节点）继续往下遍历，直至到达 STU 树叶子节点。在自上而下的遍历过程中，删除 STU 树中非最大内容节点，最终得到只包含网页正文文本的 Web 文档。

注意，上述的 STU 树，不同于前面提到的 STU-DOM 树。STU 树中只有 STU 节点，而 STU-DOM 树只是在 DOM 树的基础上对为分块节点的标记添加语义属性，构成 STU 节点，本质上仍然为 DOM 树。

3.4 实验与分析

文献[61]在进行页面分块时需要首先根据标签的数量来判断网页使用何种页面布局标签，然后针对单一的页面布局标签对网页进行分块。然而，通过大量分析可知，仅仅根据标签数量多少来判断页面使用何种布局标签是不准确的。与文献[61]分块算法不同的是，本文在分块时同时考虑了所有可能的页面布局标签。而不需要实现判断网页使用何种标签。

文献[57]在判断冗余内容块时，由于只考虑冗余文本是链接文本的情况而可能会误将一些非链接的冗余文本如版权信息当成网页正文保存下来。与文献[57]算法相比，本文不仅考虑了冗余文本是链接文本的情况，还考虑了冗余文本是非链接文本的情况，解决了文献[57]中存在的问题。同时，本文考虑到在主题型网页中，图片并不是网页的主题，因此将图片作为冗余内容删除了，而没有像 RIPB 算法那样根据图片在内容块中的比重来判断是否冗余内容块。一些内容块在删除图片后仍然存在零散的文字。本文通过分析文字的结构与文字占整个网页内容的比重来判断并删除冗余内容块。可见，本文能够如 RIPB 算法那样判断并删除图片较多的冗余内容块，又解决了 RIPB 算法无法删除包含文本而图片较少的冗余内容块的问题。

目前多数文本提取算法在实验时都是选取同一网站的或者少数几个网站的

若干网页进行测试。然而这种测试方法对于类似本文的以网页结构为基础的提取算法并不适合。因为通常情况下，同一网站的主题型网页具有相同或类似的页面结构，正文信息的提取效果也是相同或者类似。为了更好的检测算法的效果，我们随机抽取了网易、凤凰、腾讯、新浪、及搜狐等知名门户网站的部分主题型网页进行实验。实验环境处理器为 Intel(R) Core(TM)2 Duo CPU T5870 2.00GHz，3.00GB 内存，Windows 7 旗舰版系统，开发平台为 Eclipse 3.5，开发语言 Java。本文使用完整率、压缩比及平均提取时间来评估本文提取算法的性能。其中完整率为提取之后保留了正文信息的网页数与被测试网页总数的比值；压缩比为提取后文档集大小与原始文档集大小的比值；平均提取时间为提取一个网页文件的平均时间。

记本文提出的算法为主算法，文献[60]的算法为对比算法。主算法与对比算法的提取结果对比如表 3.1 所示。

表 3.1 提取结果对比

来源网站	网页数目	主算法			对比算法		
		完整率/%	压缩比	时间/s	完整率/%	压缩比	时间/s
网易	100	100	0.036	0.027	100	0.037	0.023
凤凰	100	100	0.025	0.024	99	0.027	0.027
腾讯	100	98	0.023	0.030	98	0.023	0.027
新浪	100	100	0.030	0.019	96	0.030	0.020
搜狐	100	100	0.026	0.035	94	0.029	0.047

图 3.4、图 3.5 分别为主算法与对比算法的完整率对比和压缩比对比的柱状图表示。图 3.6 为主算法与对比算法的平均提取时间对比的柱状表示。

由表 3.1、图 3.4 及图 3.5，可以发现：对于实验数据集，本文提出的算法尽可能的剔除冗余数据且不丢失网页主题信息。这一点可以从主算法具有较高的完整率和较低的压缩比得到印证。对比算法的提取结果中常包含一些主题无关的冗余信息，使得经过提取处理的结果文档集较大，而主算法产生的提取结果则较少包含此类冗余信息。

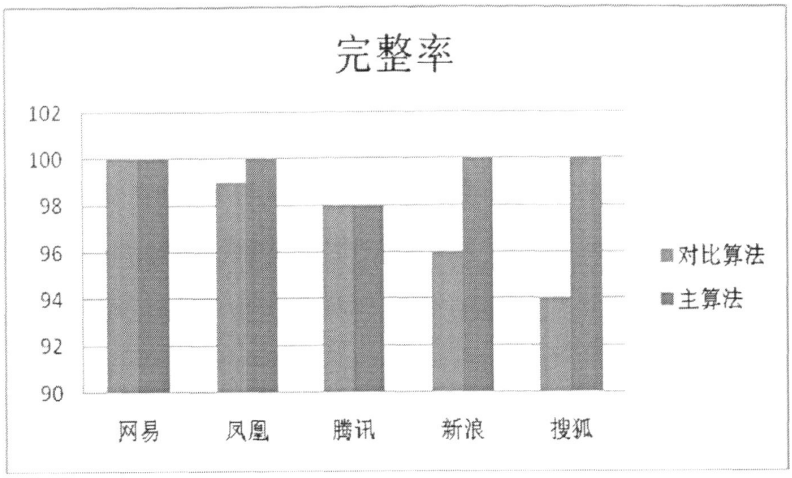

图 3.4 完整率对比

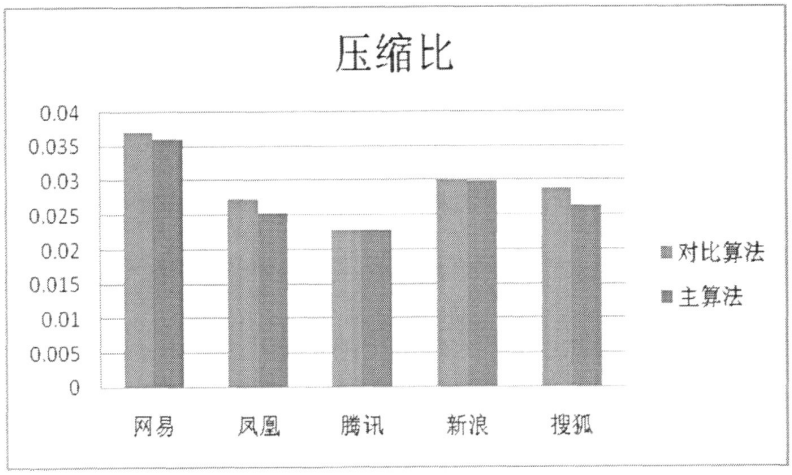

图 3.5 压缩比对比

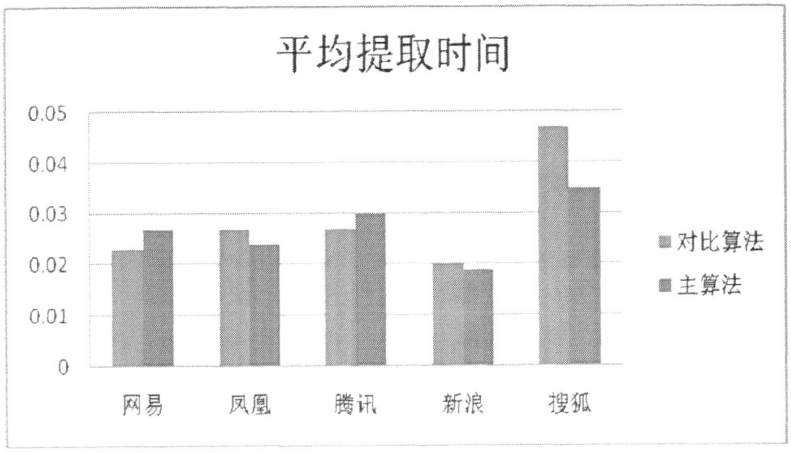

图 3.6 平均提取时间对比

本文算法针对主题型网页的正文信息提取，此类网页中，通常通过成段的文字描述一个或多个主题。网页中以布局标签做整体布局，网页正文分布于由布局标签分隔开的某一个文本内容块中。实验表明，本文提出的算法对于这样的网页正文内容块具有较高的识别率，且较大程度上剔除非正文内容块，具有较强的抗噪音干扰性。

3.5 本章小结

本章提出了一种新的 Web 信息提取方法，将输入的 HTML 根据布局标签划分为各个内容块，构建相应的内容块层次树。自底向上遍历该嵌套树，计算每个内容块节点的语义属性。根据计算得到的语义属性，删除冗余内容块和没有内容的块节点。通过这样的基于结构的过滤和基于语义的剪枝，最后生成只含有网页正文的 HTML 文档。该方法不需要借助于其他一些系统如分词系统，也不需要进行复杂的计算，且不依赖于特定的 HTML 标记，因此实现简单且具有一定通用性。

第四章 基于最大公共子图的中文 Web 文本分类

传统的文本表示方法是基于特征词条集合的向量空间文本表示方法（VSM），是一种基于统计学的文本表示方法，它不考虑文章中词与词之间的相互关联关系，不能有效表达文档的结构信息，缺乏对特征词条上下文环境的考虑。而这些文本结构信息或者上下文环境在自然语言处理中是尤为重要的。从自然语言的角度来看，向量空间模型尽管是比较成功的，但还是很不完善的[62]。

针对向量空间表示模型的缺陷，许多学者提出了基于图模型的文档表示方法。如 Svetlana 提出的基于辅助词典 Verb Net 和 Word Net 的文档概念图表示模型[63]；Bhoopesh 和 Pushpak 提出了根据 UNL 图来构造代表文档的特征向量，并采用 SOM 技术对文本进行聚类[64]；还有 Inderjeet Mani 和 Eric Bloedorn 提出了用于多文档摘要提取的文档图模型表示方法[65]。这些图模型很好地体现了文档的语义信息。

本章研究图模型下的 Web 文本分类方法，通过考察文本的结构信息而提高文本表示模型的表达能力从而优化文本分类性能

4.1 图的基本定义

定义 4.1 一个图是一个四元组 $G=(V,E,\alpha,\beta)$，其中，

(1) $V\neq\varnothing$ 称为 G 的顶点集，其元素称为顶点或结点；

(2) $E\subseteq V\times V$ 称为 G 的边集，其元素称为边；

(3) $\alpha:V\to\Sigma_V$ 为图 G 的顶点标记函数，说明顶点与其标记的对应关系；

(4) $\beta:E\to\Sigma_E$ 为图 G 的边标记函数，说明边与其标记的对应关系。

Σ_V、Σ_E 分别为出现在图 G 中的顶点标记、边标记集合。为简化表示，我们可以省略标记映射函数，将图 G 表示为 $G=(V,E)$。

定义 4.2 一个图 $G_1=(V_1,E_1,\alpha_1,\beta_1)$ 为图 $G_2=(V_2,E_2,\alpha_2,\beta_2)$ 的一个子图，记为

$G_1 \subseteq G_2$，如果有，

(1) $V_1 \subseteq V_2$；

(2) $E1 \subseteq E2 \cap (V1 \times V1)$；

(3) $\alpha_1(x) = \alpha_2(x)$，$\forall x \in V_1$；

(4) $\beta_1((x,y)) = \beta_2((x,y))$，$\forall (x,y) \in E_1$。

此时，我们也称图 G_2 为图 G_1 的一个超图。

两个图是同构的，如果两个图包含相同的顶点个数，且两个图的顶点间存在有一个一一映射，使得两图的对应顶点标记以及对应边标记保持一致。形式的定义如下：

定义 4.3 图 $G_1 = (V_1, E_1, \alpha_1, \beta_1)$ 与图 $G_2 = (V_2, E_2, \alpha_2, \beta_2)$ 是同构的，记为 $G_1 \cong G_2$，如果存在一个双射函数 $f: V_1 \to V_2$，使得：

(1) $\alpha_1(x) = \alpha_2(f(x))$，$\forall x \in V_1$；

(2) $\beta_1((x,y)) = \beta_2((f(x), f(y)))$，$\forall (x,y) \in E_1$。

这样的函数 f 也称为图 G_1 与图 G_2 的图同构。

定义 4.4 给定图 G_1 与 G_2 的一个图同构 f 以及另一个图 G_3，如果 $G_2 \subseteq G_3$，则 f 为图 G_1 与 G_3 的一个子图同构。

现在还不知道图同构问题是否是 NP 完全的，但子图同构是一个 NP 完全问题。图同构仅仅说明两个图之间存在确切的匹配，即两个图是拓扑相同的。除了说明两个图同构与否，图同构并未能提供任何有关图之间相似度的有用提示。子图同构则说明一个图作为另外一个图的一部分出现。

定义 4.5 图 G_1 与图 G_2 的相似度，记为 $s(G_1, G_2)$，为满足如下性质的函数：

(1) $0 \le s(G_1, G_2) \le 1$；

(2) $s(G_1, G_2) = 1 \to G_1 \cong G_2$；

(3) $s(G_1, G_2) = s(G_2, G_1)$；

(4) 如果图 G1 相比图 G3 与图 G2 更相似，则 $s(G_1,G_2)\geq s(G_2,G_3)$。

上述相似度定义会产生这样一个难题：什么条件下两个图的相似度为 0？问题源于我们不清楚什么是一个图的完全相反。故而，一般使用距离度量方式来衡量图之间的数值相似度。

定义 4.6 图 G_1、G_2之间的距离，记为 $d(G_1,G_2)$，为满足如下性质的函数：

(1) $d(G_1,G_2)\geq 0$；

(2) $d(G_1,G_2)=0\rightarrow G_1\cong G_2$；

(3) $d(G_1,G_2)=d(G_2,G_1)$；

(4) $d(G_1,G_3)\leq d(G_1,G_2)+d(G_2,G_3)$。

可以通过图之间的相似性度量得到图之间的距离度量，如

$$d(G_1,G_2)=1-s(G_1,G_2) \tag{4.1}$$

定义 4.7 图 g 为图 G_1 和图 G_2 的最大公共子图（Maximum Common Subgraph，MCS），记为 $mcs(G_1,G_2)$，如果

(1) $g\subseteq G_1$；

(2) $g\subseteq G_2$；

(3) 不存在其他子图 g'，使得 $g'\subseteq G_1$，$g'\subseteq G_2$，$|g'|>|g|$。

定义 4.7(3)中 $|g|$ 一般意为图 g 的顶点大小，用于说明图 g 的大小。

4.2 最大公共子图求解

图 G_1 和图 G_2 的最大公共子图 g 的求解过程包括两个步骤：

(1) 遍历图 G_1 和图 G_2 的结点，对结点进行比较，取图 G_1 和图 G_2 的公共结点，作为 g 的结点；

(2) 取步骤(1)中得到的图 g 的任意两个结点，如果这两个结点在图 G_1 和图 G_2 中都是邻接的，那么产生一条边，作为 g 的边。求解过程的伪代码如算法 4.1 所示。

算法 4.1 最大公共子图求解

输入：

- 图 $G_1=(V_1,E_1,\alpha_1,\beta_1)$
- 图 $G_2=(V_2,E_2,\alpha_2,\beta_2)$

输出：

- 图 G_1 与图 G_2 的最大公共子图 $g=(V,E,\alpha,\beta)$

算法步骤：

1: **for each** $v_i \in V_1$

2: **for each** $v_j \in V_2$

3: **if** $\alpha_1(v_i)=\alpha_2(v_j)$ //v_i，v_j 的顶点标记一致

4: $V=V\cup\{v_i\}$

5: 对 V 中的顶点 v_i 按顶点标记的字典序排序

6: **for** $i=1$ to $|V|$-1

7: **for** $j=i+1$ to $|V|$

8: **if** $<v_i,v_j>\in E_1$ **and** $<v_i,v_j>\in E_2$

9: $E=E\cup\{<v_i,v_j>\}$

10: **if** $<v_j,v_i>\in E_1$ **and** $<v_j,v_i>\in E_2$

11: $E=E\cup\{<v_j,v_i>\}$

由于最大公共子图的定义过于严格，有时会使两个图之间的重叠部分过小，因此在本研究中，我们引入广义最大公共子图的概念，分别求解两个图中相同的结点和相同的边，即允许最大公共子图中有孤立结点的存在。

4.3 基于最大公共子图的图相似度

Bunke[66]指出两个图之间的编辑距离与它们的最大公共子图有着直接关联。

特别地，当对编辑距离的操作代价函数作一定的限制，两者是等价的。

图 G_1 和 G_2 的最大公共子图 $mcs(G_1,G_2)$ 是两图中的共有部分，不会因为插入结点和删除结点的编辑操作而改变。为将图 G_1 编辑成图 G_2，只需执行如下操作：

(1) 从图 G_1 删除不在最大公共子图 $mcs(G_1,G_2)$ 出现的结点和边；

(2) 执行任意的顶点替换或边替换；

(3) 插入不在最大公共子图 $mcs(G_1,G_2)$ 出现的图 G_2 的结点和边。

根据以上观察，图 G_1 与图 G_2 的相似度与两者的最大公共子图 $mcs(G_1,G_2)$ 的大小有关系。Bunke 和 Shearer[67]基于最大公共子图 $mcs(G_1,G_2)$ 给出一种图 G_1 与图 G_2 的距离度量方式，公式如下：

$$d_{MCS}(G_1,G_2)=1-\frac{|mcs(G_1,G_2)|}{\max(|G_1|,|G_2|)} \tag{4.2}$$

该距离定义满足上节图距离定义的 4 个条件。

(1) $0\le d_{MCS}(G_1,G_2)\le 1$；

(2) 当图 G_1 与图 G_2 相同，$d_{MCS}(G_1,G_2)=0$；

(3) $d_{MCS}(G_1,G_2)$ 满足对称性，即 $d_{MCS}(G_1,G_2)=d_{MCS}(G_2,G_1)$；

(4) $d_{MCS}(G_1,G_2)$ 满足三角不等式。

该距离定义相比图编辑距离，避免了编辑操作代价系数和其他参数的选择。

另外一种图距离度量，由 Wallis 等人[68]提出，公式如下：

$$d_{WGU}(G_1,G_2)=1-\frac{|mcs(G_1,G_2)|}{|G_1|+|G_2|-|mcs(G_1,G_2)|} \tag{4.3}$$

公式（4.3）中的除数表示从集合理论角度看来的两个图的并集的大小。

4.4 基于最大公共子图的中文 Web 文本分类

大多数 Web 分类方法来自于信息检索，采用向量空间模型 VSM 来表示 Web 文本。基于该模型，从训练文档集中的词来构造特征项集合。每个文档 D_i 表示

为向量$(d_{i1},d_{i2}\cdots d_{i|d|})$，其中向量维数$|d|$为特征项集合的基数。多数分类算法都可以应用此种文档表示模型。人们已经提出了许多度量两个向量距离和相似度的方法[29,51,69]，因此 K 最近邻算法可以简单地应用于 Web 文本分类。

然而，这种文档表示模型并没能捕捉一些重要的结构信息，例如文档中词的出现顺序以及邻近关系，而这些结构信息可能有助于我们提高文本分类的精度。

向量空间模型中，已经有很多相当成熟的分类算法，如果能够将已有的分类算法与本文的图模型有效结合起来，这是一种非常好的思路。因此，本文致力于研究这方面的内容。K 最近邻方法（KNN）是基于向量空间模型的文本分类效果最好的方法之一。它的基本思想是在训练样本中找到测试样本的 K 个最近邻，然后根据这 K 个最近邻的样本来决定测试样本的类别。

与一般的数据结构比较，图能够表达更加丰富的语义。图结构能够模拟几乎所有事物之间的联系，它能应用到半结构化和非结构化的数据挖掘中。采用图结构，我们能更大程度地保留原始文档中的信息，为分类服务，以提高分类算法的性能。

本文提出一种基于最大公共子图的 Web 文本分类方法 MCS-KNN（Maximum Common Subgraph based K-nearest Neighbour，MCS-KNN）。将训练集和测试集的 Web 文档表示成一个以特征词为顶点的有向图，图的每一条有向边则表示两端顶点对应的特征词在同一个语义单元（句子）中相邻出现以及先后关系。这样，传统分类算法中文档之间的相似性度量问题便转化为图之间的相似度度量问题。我们利用公式（4.3）计算两个图之间的相似性。计算待分类文档的表示图与训练集中每个 Web 文档的表示图之间的相似度，找出训练集中具有最大相似度的 K 个 Web 文档，然后依据这 K 个最近邻的 Web 文档来决定测试文档的类别。

4.4.1 算法描述

给定一个带有类别标记的 Web 文档训练集$D=\left(D_1\cdots D_{|D|}\right)$和一个类别集合$C=\left(c_1\cdots c_{|C|}\right)$。其中任意的文档$D_i\in D$，$1\le i\le|D|$，属于且仅属于类别集合中的一个类别$c_v\in C$，$1\le v\le|C|$。对于一个未知类别的待分类 Web 文档 d，MCS-KNN

执行如下：

算法 4.2 MCS-KNN 算法

输入：

- Web 文档训练集 $D=\left(D_1 \cdots D_{|D|}\right)$
- 待分类的 Web 文档 d
- 近邻数 K
- 文档图表示的最大结点数 N

输出：

- 待分类 Web 文档的类别 c

算法步骤：

1： **for** $i = 1$ **to** $|D|$ **do**

2： 提取出训练集文档 D_i 的标题、关键字、描述、正文等信息；

3： 对 D_i 的提取信息分词，进行频数统计。取频数最高的 N 个特征词构建其表示图 G_i ；

4： **end for**

5： 提取待分类文档 d 的标题、关键字、描述、正文等信息，对信息文本分词，取最频繁出现的 N 个特征词构建其表示图 g；

6： **for** $i=1$ **to** $|D|$ **do**

7： 计算图 G_i 与图 g 的最大公共子图 $mcs(G_i,g)$；

8： 计算图 G_i 与图 g 的相似度 $s(G_i,g)=\dfrac{mcs(G_i,g)}{|G_i|+|g|-mcs(G_i,g)}$；

9： **end for**

10：根据相似度 $s(G_i,g)$，选择与测试文档 d 最相似的 K 个文档，即为测试文档 d 的 K 个近邻；

11：计算 K 个近邻中每个类别的相似度和；

12：返回具有最大相似度和的类别为测试文档 d 的类别。

下面就 MCS-KNN 算法的细节作详细描述，包括 Web 文档预处理，图表示

建立，图相似度计算，分类过程等。

4.4.2 Web 文档预处理

在典型的网页中，特别是一些商业网页，包含很多不是主要内容的信息。例如：可能包含一些横幅广告、导航条以及版权声明等等，这些内容都会降低 Web 文本分类的性能。

MCS-KNN 通过采用前述的基于网页分块的主题信息自动提取算法去除网页中冗余噪音，获得网页的主题内容。

4.4.3 Web 文档图建立

A. Schenker[70]介绍了五种不同的基于特征词邻接的从 Web 文档中建立文档图表示的方法。在本文中，我们选择简单图表示方式（the simple graph representation）来生成训练集和测试集中所有 Web 文档的图表示。在简单图表示方式中，每一个在 Web 文档中出现的特征词都成为表示该文档的图中的一个顶点。每一个顶点以它所代表的特征词为标记。在一个文档图中，顶点的标记是唯一的。如果一个特征词在 Web 文档中多次出现，我们仍然只创建一个标记为该特征词的顶点。如果 Web 文档的一个句子中，特征词 A 与特征词 B 相邻而且特征词 A 在特征词 B 前边，则在 Web 文档的表示图中存在一条有向边连接特征词 A 和特征词 B 的相应顶点。但如果特征词 A 和特征词 B 相邻，但却不在同一个句子单元中，则在该文档的表示图中我们并不创建连接特征词 A 和特征词 B 相应顶点的有向边。

通过上述方式建立的 Web 文档表示图，相比向量空间表示方式，能够传达更多的信息。Web 文档的图表示方法，能很好地体现文本的结构信息，并保留了重要的语义信息。表示图中的节点体现了文本的特征项信息，有向边体现了特征项的共现信息及语序信息。

4.4.4 图相似度计算

Web 文档的简单图表示模型具有如下性质[71]：

性质 1 两个文档 D_1 和 D_2 语义越接近，则它们的对应的文档图也越相似。相反，两个文档图越相似，则它们的在语义上是越接近的。

性质 2 两个文档 D_1 和 D_2 语义越接近，体现在图的特征上，两个图就有更多的相同的顶点和边。

根据定义，最大公共子图体现了两个图之间的最大共同出现组件（顶点和边），在其之上建立图之间相似性度量是可行的。

本文我们选择的衡量两个文档对应的表示图之间的相似性度量为：

$$s(D_1,D_2)=s(G_1,G_2)=\frac{mcs(G_1,G_2)}{|G_1|+|G_2|-|mcs(G_1,G_2)} \qquad (4.4)$$

4.4.5 Web 文本分类

本节我们考虑如何对待分类文档进行分类。用公式（4.4）来计算待分类文档与训练集中每一个文档的相似度。KNN 是一种传统的惰性学习算法，分类性能良好，得到了应用，我们选择 KNN 算法作为我们的分类算法。

对于一个待分类的 Web 文档，计算其与训练集中所有文档的相似度。按相似度大小，选择训练集中与待分类文档最相似的 K 个文档作为待分类文档的 K 个近邻。计算 K 个近邻中每一个类的分数，方法是将 K 个近邻中属于该类的相似值相加。将待分类文档分类为分数最高的那个文档类，如出现两个具有最高分数的文档类，则将文档分类为这两类文档中含有与待分类文档最大相似值的类。

4.5 实验与分析

实验目的是验证 MSC-KNN 算法的有效性并与传统的向量空间模型（VSM）进行对比，以准确率、召回率及 F_1 值作为评价标准。

为了验证算法的有效性，本文随机抽取了凤凰资讯网合计 1511 个新闻主题型网页，分为三个类别 C000001、C000002、C000003，其类别划分与网站的分类一致。对于每个类别的 Web 文档，随机选取 2/3 的文档作为该类别的训练集，剩下 1/3 的文档作为该类别的测试集。表格 4.1 给出了数据集中每个类别的分布情况。

本实验环境处理器为 Intel(R) Core(TM)2 Duo CPU T5870 2.00GHz，3.00GB 内存，Windows 7 旗舰版系统，开发平台为 Eclipse 3.5，开发语言 Java。

表 4.1 数据集分布情况

类别	训练(篇)	测试(篇)
C000001	326	164
C000002	350	175
C000003	330	166
合计	1006	505

当 K=10，N=50，MCS-KNN 的分类结果：

表 4.2 MCS-KNN 分类结果（K=10，N=50）

	C000001	C000002	C000003	宏平均
准确率	0.885	0.921	0.934	0.914
召回率	0.939	0.942	0.855	0.912
F1 值	0.911	0.932	0.893	0.913

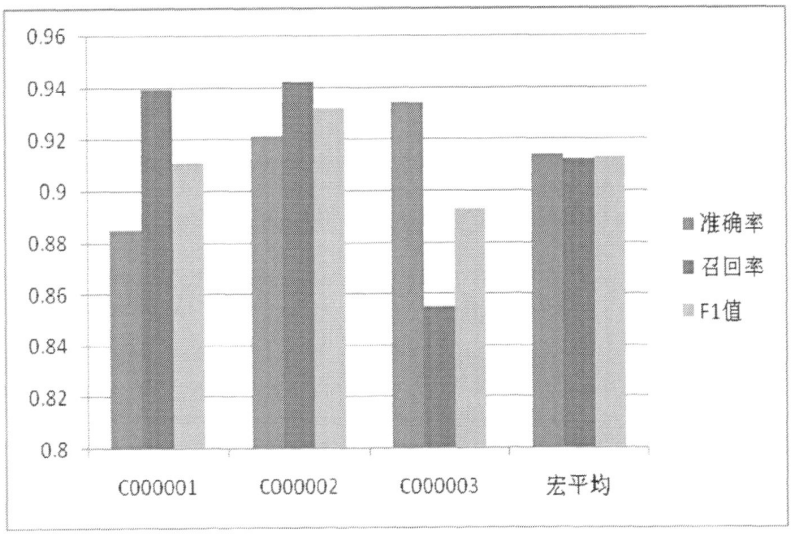

图 4.1 MCS-KNN 分类柱状图

由表 4.2 和图 4.1，MCS-KNN 对实验数据集的分类效果较好，整体准确率、召回率、F_1 值均达到了比较高的水平，说明本文提出的 MCS-KNN Web 文本分类算法是可行和有效的。

我们还对 MCS-KNN 算法与基于向量空间模型的传统 KNN 算法作了比较。

表 4.3 给出 MCS-KNN 与 KNN 的分类一个文档的平均时间对比，可见 MCS-KNN 算法分类要比 KNN 来的快。

表 4.3 MCS-KNN 与 KNN 的平均分类时间对比

方法	分类一个文档的平均时间(s)
KNN	0.572
MCS-KNN(N=50)	0.102

MCS-KNN 算法中，选择每个 Web 文档中最频繁出现的 N 个词构建 Web 文档的图表示，且每个结点的标记唯一，使得图表示相比向量表示规模较小。又在每个结点各不相同的情况下，两个图的最大公共子图可以在多项式时间求解。因而，MCS-KNN 能够在比较短的运行时间内达到相当的分类性能。

图 4.2、图 4.3、图 4.4 分别给出了 MCS-KNN 和 KNN 在不同 K 值下的准确率、召回率及 F_1 值对比。

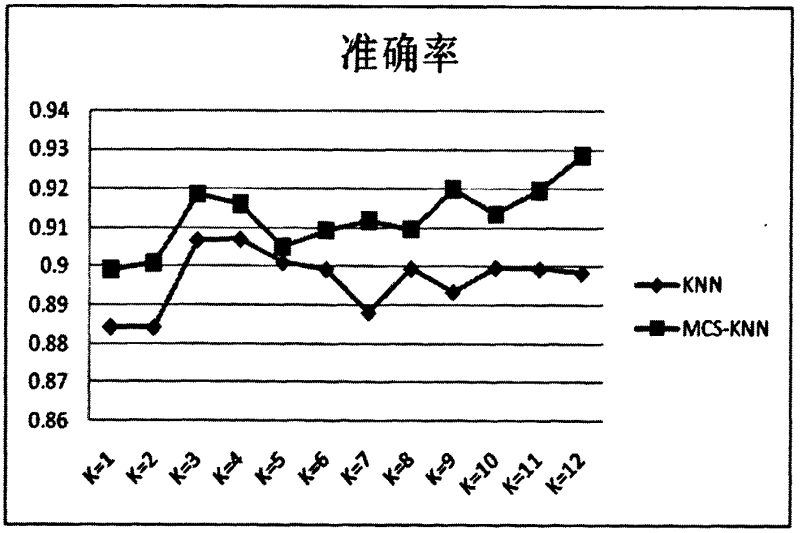

图 4.2 MCS-KNN 与 KNN 的准确率对比图

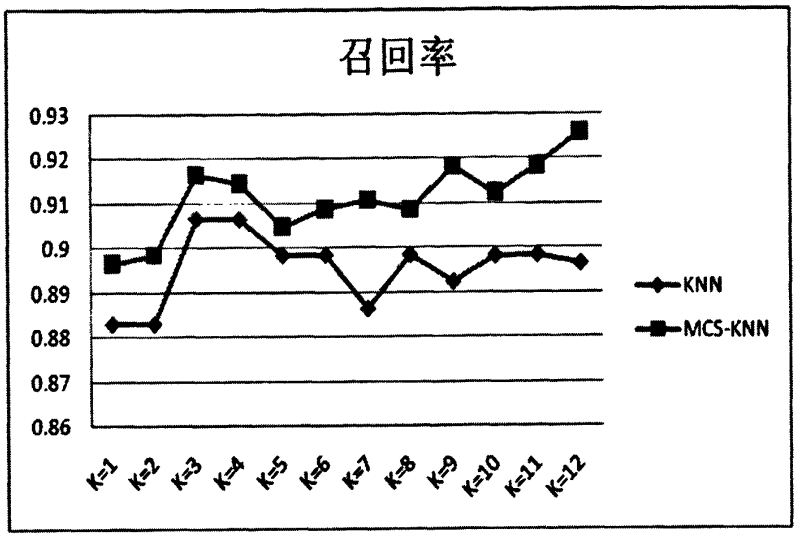

图 4.3 MCS-KNN 与 KNN 的召回率对比图

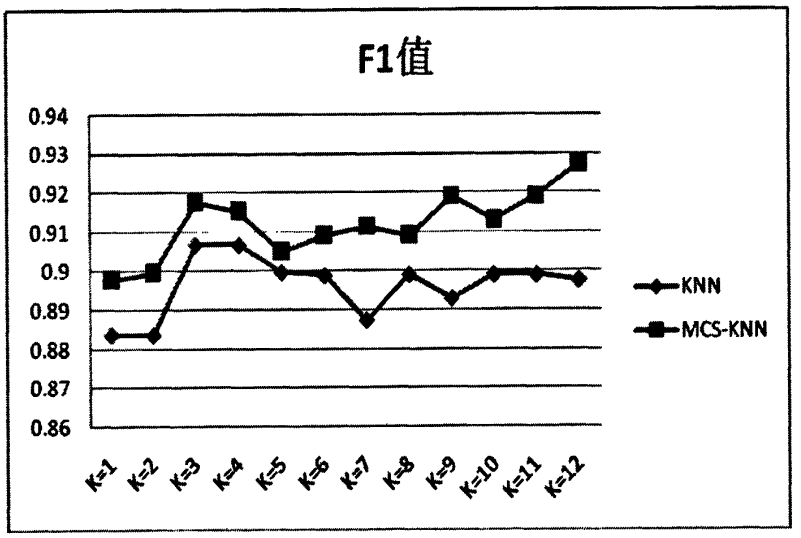

图 4.4 MCS-KNN 与 KNN 的 F_1 值对比图

从图 4.2、图 4.3、图 4.4 可以看出，MCS-KNN 相比传统的基于向量空间模型的 KNN 具有一定的优势，从另一方面再次说明了基于最大公共子图的中文 Web 文本分类方法 MCS-KNN 具有可行性。

4.6 本章小结

本章首先介绍了有关图的一些基本定义，并在此之上引申出图的之间的相似性度量，并介绍了几种基于最大公共子图的两图之间的距离度量。

其次，说明传统的基于向量空间模型的文本表示方式的局限，指出利用图的强大的数据表示能力来进行文本表示是一个可行的方向。在此基础上探讨了 Web 文档的图表示建立方式、文档图之间距离度量选择等问题，将文本的图表示方式通过基于最大公共子图的图相似性度量与传统的 KNN 方法有机结合，形成 MCS-KNN 分类算法。

最后，通过实验验证了 MCS-KNN 算法的有效性和可行性。

第五章 结论

5.1 总结

Web 文本分类技术是 Web 文本挖掘的一个重要组成部分，可以对大量 Web 进行快速、有效地自动分类与归档，使得用户可以更加方便地浏览文档，而且可以通过限制搜索范围来查找文档，大大提高了信息检索效率。而由于国内相关技术起步较晚，加上中文文本的特殊性，中文 Web 文本的分类技术相对落后。虽然通过近几年的研究与实践，已有的文本分类算法在中文 Web 文本分类中得到了广泛的应用并取得了不错的效果，但是随着信息技术和互联网的进一步发展，中文 Web 文本信息更加丰富与复杂，对中文 Web 文本分类技术提出了更高的要求，已有的文本分类算法都存在着相应的不足之处，渐渐难以满足实际要求，需要加以改进。

本文的主要研究工作如下：

1. 分析了 Web 文本分类的意义和国内外研究现状，分析了中文 Web 文本类的研究现状和存在问题。

2. 简要介绍了 Web 文本挖掘的概念、特点、主要内容、处理过程和应用领域。

3. 介绍了中文 Web 文本分类的主要过程及关键技术，包括文本预处理、文本表示、索引生成、特征选取以及文本分类算法和评价指标，详细分析了几种文本分类算法，包括 KNN、SVM 等其它分类算法。

4. 提出一种基于网页分块的主题信息自动提取算法，自动提取网页中的主题信息，经试验，其精度和时间均达到了一定的应用水准。通过主题信息的正确提取，去除无关噪音，为 Web 文本分类助力。

5. 提出基于最大公共子图的中文 Web 文本分类算法 MCS-KNN。Web 文档图表示模型能够有效地避免传统向量空间文本表示模型的诸多局限，保留更多的文档语义结构信息，为文本分类性能的提高提供可能。实验证明，MCS-KNN 算法具有可行性和优越的分类性能。

5.2 后续工作

本文研究了中文 Web 文本分类的关键技术和常用方法，并在已有技术的基础上进行了改进，但是还存在着一些不足之处，还需要进一步完善，是以后工作的努力方向。主要有以下方面：

1. 中文分词的质量有待进一步提高。中文信息处理离不开中文分词的支持，中文 Web 文本分类更需要高可用性的中文分词手段。在网络语料中，新词层出不穷，这就要求我们的分词方法能够有学习能力，能够识别未登录词。

2. 进一步利用 HTML 文档的结构信息。本文的 Web 文档表示方法对 HTML 结构信息的利用较少，而 HTML 文档标记含有丰富的信息，合理利用，必将有助于分类。

3. 研究新的 Web 文档图表示方式。本文采用的简单图表示方式仅考虑了特征词条、特征词条的位置关系信息，而对于自然语言处理，特征词条的共现程度同样重要。故而建立新的 Web 文档图表示模型，充分表达文档中特征词条、特征词条的位置关系以及特征词条的共现程度等信息，也是将来工作的一个方面。

4. 研究新的图相似性度量，将文档相似性计算问题转化为图相似性计算问题，运用图分类的方法实现 Web 文本分类。

参考文献

[1] J. Rocchio. Relevance feedback in information retrieval[C]//The SMART retrieval system: experiments in automatic document processing. Englewood Cliffs: Prentice-Hall, 1971.

[2] Van Rijsbergen C.J. Information retrieval[M]. London: Butterworths, 1979.

[3] David Dolan Lewis. Rrepresentation and Learning in Information Retrieval[D]. America: University of Massachusetts, 1992.

[4] Lewis DD. Reuters-21578 text categorization test collection Distribution 1.0 [EB/OL]. http://www.daviddlewis.com/resources/testcollections/reuters21578/readme.txt, 1997-09-26.

[5] Yang Yiming. An evaluation of statistical approaches to text categorization[J]. Journal of Information Retrieval, 1999, 1(1/2): 67-88.

[6] Hersh WR, Buckley C, Leone TJ, et al. OHSUMED: An interactive retrieval evaluation and new large test collection for research[C]//Proceedings of the 17th Annual ACM SIGIR Conference. 1994: 192-201.

[7] Cortes C, Vapnik V. Sup of event models for naive port vector networks[J]. Machine Learning, 1995, 20: 1-25.

[8] Joachims T. Text categorization with support vector machines: learning with many relevant features[C]//Proceedings of 10th European Conference on Machine Learning(ECML-98). Chemnitz, DE, 1998: 137-142.

[9] Freund Y, Schapire R.E. A Decision Theoretic Generalization of On-Line Learning and anApplication to Boosting[J]. Journal of Computer and System Science, 1995, 55(1): 119-139.

[10] Robert E., Schapire, Yoram Singer. Improved Boosting Algorithms Using Confidence-rated Predictions[J]. Machine Learning, 1999: 80-91.

[11] Rainer Lienhart, Alexander Kuranov, Vadim Pisarevsky. Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection[C]//DAGM 25th Pattern Recognition Symposium. 2003: 297-304.

[12] Friedman J, Hastie T, Tibshirani R. Additive Logistic Regression: A Statistical View of

Boosting[J]. Annals of Statistics, 2000, 28(2): 337-407.

[13] Yang Y, Liu X. A Re-examination of Text Categorization Methods[C]//Proc of SIGIR'99. 1999: 42-49.

[14] Fabrizio Sebastiani. Machine learning in automated text categorization[J]. ACM Computing Surveys (CSUR), 2002, 34(1).

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

[16] 张滨. 中文文档分类技术研究[D]. 湖北: 武汉大学, 2004.

[17] 张治平. Web 信息精确获取技术研究[D]. 湖南长沙: 国防科学技术大学, 2004.

[18] 黄萱菁, 吴立德, 石崎洋之, 等. 独立于语种的文本分类方法[J]. 中文信息学报, 2000, 16(6): 1-7.

[19] 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[J]. 计算机研究与发展, 2000, 37(9): 1033-1038.

[20] 刘永丹. 文档数据库若干关键技术研究[D]. 上海: 复旦大学, 2004.

[21] 柯慧燕. Web 文本分类研究及应用[D]. 湖北: 武汉理工大学, 2006.

[22] 宋翰涛. Web 中文文本分词技术研究[J]. 计算机应用, 2004, 24(4): 134-135.

[23] 步丰林. 一个中文新词识别特征的研究[J]. 计算机工程, 2004, 30(B12): 369-370.

[24] 王继成, 潘金贵, 张福炎. Web 文本挖掘技术研究[J]. 计算机研究与发展, 2000, 37(5): 513-520.

[25] 王本年, 高阳, 陈世福等. Web 智能研究现状与发展趋势[J]. 计算机研究与发展, 2005, 42(5): 721-727.

[26] 彭雅. 文本分类算法及其应用研究[D]. 湖南: 湖南大学, 2004.

[27] 罗强. 基于粗糙集理论的知识发现在 Web 文本挖掘上的应用研究[D]. 广西: 广西大学, 2003.

[28] 李波, 李新军. 一种基于粗糙集和支持向量机的混合分类算法[J]. 计算机应用, 2004, 24(3): 65-70.

[29] JiaWei Han, Micheline Kamber. 数据挖掘概念与技术[M], 范明, 孟小峰译. 北京: 机械工业出版社, 2007.

[30] 朱明. 数据挖掘[M]. 合肥: 中国科学技术大学出版社, 2008.

[31] Etzioni O. The World-Wide Web: quagmire or gold mine[J]. Communications of ACM, 1996, 39(11): 65-68.

[32] Bing Liu. Web 数据挖掘[M], 俞勇, 薛贵荣, 韩定一译. 北京: 清华大学出版社, 2009.

[33] Yuefeng Lia, Ning Zhong. Web mining model and its applications for information gathering[J]. Knowledge-Based Systems, 2004(17): 207-217.

[34] Sergey Brin. Extracting patterns and relations from the World Wide Web[C]//Proc of WebDB Workshop at EDBT'98. Valencia, 1998.

[35] Ronen Feldman, Ido Dagan. Knowledge discovery in textual databases (KDT)[C]//Proc of the 1st Int'l Conf on Knowledge Discovery. Mont real, 1995: 112-117.

[36] Salton G, Wang A, Yang C S. A vector space model for automatic indexing[J]. Communication of the ACM, 1975, 18(11): 613-620.

[37] Y. Yang, J. Pedersen. A comparative Study on Feature Selection in Text Categorization[C]// International Conference on Machine Learning. 1997: 412-420.

[38] 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[J]. 中文信息学报, 2004, 18(1): 26-32.

[39] Joachims T. A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization[C]//Proc of ICML'97. 1997.

[40] Yang Y. Expert network: Effective and efficient learning from human decisions in text categorization and retrieval[C]//Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval. 1994: 13-22.

[41] Masand B, Linoff G, Waltz D L. Classifying News Stories Using Memory Base Reasoning[C]//AGM SIGIR. 1992: 59-65.

[42] Mccallum A, Nigam K. A comparison of event models for naive Bayes text classification[C]//Proc of the AAAI-98 Workshop on Learning for Text Categorization. Menlo Park, CA: AAAI Press, 1998: 41-48.

[43] Baker L D, Mccallum A K. Distributional clustering of words for text categorization[C]//Proc of SIGIR'98. 1998: 96-103.

[44] Cortes C, Vapnik V. Sup of event models for naive port vector networks[J]. Machine Learning, 1995, 20: 1-25.

[45] Joachims T. Text Categorization with Support Vector Machine: Learning with Many Relevant Features[C]//Proc of ECML'98. 1998: 137-142.

[46] 王国胜, 钟义信. 支持向量机的理论基础——统计学习理论[J]. 计算机工程与应用,

2001, 19: 19-20.

[47] Wu H, Gunopulos D. Evaluating the Utility of Statistical Phrases and Latent Semantic Indexing for Text Classification[C]//Proc of ICDM'02. 2002: 713-716.

[48] 张学工. 关于统计学习理论与支持向量机[J]. 自动化学报, 2000, 26(1): 32-42.

[49] 萧嵘, 王继成, 张福炎. 支持向量机理论综述[J]. 计算机科学, 2000, 27(3): 1-3.

[50] Lewis D D, Ringuette M. Comparison of two learning algorithms for text categorization[C]//Proc of SDAIR. 1994: 81-93.

[51] T. Mitchell. 机器学习[M], 曾华军, 张银奎等译. 北京: 机械工业出版社, 2003.

[52] Wiener E, Pedersen J O, Weigend A S. A Neural Network Approach to Topic Spotting[C]//Proc of SDAIR'95. 1995: 317-332.

[53] Miguel E, Ruiz, Padmini Strinivasan. Hierarchical neural networks for text categorization[C]//Proc of SIGIR'99. 1999: 281-282.

[54] A Finn, A Kushmerick, B Smyth. Fact or fiction: Content classification for digital libraries[C]//The 2nd DELOS Network of Excellence Workshop on Personalisation and Recommender Systems in Digital Libraries. Dublin, Ireland, 2001.

[55] O Buyukkokten, H Garcia-Molina, A Paepcke. Accordion summarization for end-game browsing on PDAs and cellular phones[C]//Proc of ACM Conf on Human Factors in Computing Systems (CHI 2001). New York: ACM Press, 2001: 213-220.

[56] O Buyukkokten, H Garcia-Molina, A Paepcke. Seeing the whole in parts: Text summarization for Web browsing on handheld devices[C]//Proc of the 10th Int'l Conf on World Wide Web. New York: ACM Press, 2001: 652-662.

[57] FU YAN, YANG DONG-QING, TANG SHI-WEI. Using XPath to discover informative content blocks of Web pages[C]//3rd International Conference on Semantics: Knowledge and Grid. Xian: IEEE Press, 2007: 450-453.

[58] KANG J, CHO I J. Detecting informative Web page blocks for efficient information extraction using visual block segmentation[C]//2007 International Symposium on Information Technology Conver-gence. Jeon ju, Korea: IEEE Press, 2007: 306-310.

[59] KIM Y, PARK J, KIM T, et al. Web information extraction by HTML tree edit distance matching[C]//2007 International Conference on Convergence Information Technology. Gyeongju, Korea: IEEE Press, 2007: 2455-2460.

[60] 黄玲, 陈龙. 基于网页分块的正文信息提取方法[J]. 计算机应用, 2008, 28: 326-328.

[61] 黄文蓓, 杨静, 顾君忠. 基于分块的网页正文提取算法研究[J]. 计算机应用, 2007(6): 24-26.

[62] 汪志圣, 李龙澍. Web 文档分类方法的比较与分析[J]. 滁州学院学报, 2007,9(6): 33-35.

[63] Svetlana Hensman. construction of conceptual graph representation of texts[C]//Proceedings of the Student Research Workshop at HLT-NAACL. Boston, 2004: 49-54.

[64] Zhang Weifeng, Xu Baowen, Cui Zifeng, et al. Document classification approach by rough-set-based corner classification neural network[J]. Journal of Southeast University, 2006, 22(3): 439-434.

[65] Inderjeet Mani, Eric Bloedom. Multi-document Summarization by Graph Search and Matching[Z]. 1997.

[66] H. Bunke. On a relation between graph edit distance and maximum common subgraph[J]. Pattern Recognition Letters, 1997, 18: 689–694.

[67] H. Bunke, K. Shearer. A graph distance metric based on the maximal common subgraph[J]. Pattern Recognition Letters, 1998, 19: 255–259.

[68] W. D. Wallis, P. Shoubridge, M. Kraetz, et al. Graph distances using graph union[J]. Pattern Recognition Letters, 2001, 22: 701–704.

[69] A. K. Jain, M. N. Murty, P. J. Flynn. Data Clustering: A Review[J]. ACM Computing Surveys, 1999, 31(3): 264-323.

[70] A. Schenker, M. Last, H. Bunke, et al. Graph Representations for Web Document Clustering[C]//Proceedings of the 1st Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA). 2003.

[71] 邹加棋, 陈国龙, 郭文忠. 基于图模型的中文文档分类研究[J]. 小型微型计算机系统, 2006, 27(4): 754-757.

攻读硕士学位期间发表的论文

[1] LAI Xing-rui, ZHANG Dong-zhan, DUAN Jiang-jiao. Stock Price Behavior Data Mining Based on Network Consensus. Mind and Computation, 2010, 4(1): 71-77.

致 谢

本文写作的成功,得到了许多老师、同学、朋友的帮助，在这里我要向他们表达我衷心的感谢。

首先要感谢我的导师张东站副教授，本文是在张老师的悉心指导下完成的。在论文的研究过程中，张老师始终给予我宝贵的意见和严格的要求，在论文选题、研究方法、论文定稿等各个环节都给了我大力的支持和有益的启发，同时张老师严谨的治学态度和认真的工作精神令我由衷敬佩，在此向张老师表示衷心的感谢和崇高的敬意。

其次，我还要特别感谢段江娇老师，是她在研究过程中毫无保留地将一些宝贵的经验和知识传授于我，段老师严谨的工作态度和坚定的信念将会一直鼓舞着我。

另外，感谢我的师兄黄智武、师姐陈冬菊和实验室的成员施秀升、张娜、王建东等，在实验室近三年的学习和研究中，他们给予了我很多的帮助和启发,共同学习的经历让我获益非浅，终身难忘。感谢我的父母和家人，谢谢他们给予我的一切，他们的爱是我克服困难、勇敢前进的不竭动力。

最后，感谢各位评审老师在百忙之中抽出宝贵的时间来审阅本文。

三年研究生生涯马上就要结束了，三年里有许多帮助过我的人，在此对他们表示感谢!

基于最大公共子图的中文Web文本分类研究

作者： 赖兴瑞
学位授予单位： 厦门大学

参考文献(22条)

1. 张滨 中文文档分类技术研究[学位论文]硕士 2004

2. 张治平 Web信息精确获取技术研究[学位论文]硕士 2004

3. 黄萱菁,吴立德,石崎洋之 独立于语种的文本分类方法[期刊论文]-中文信息学报 2000(06)

4. 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[期刊论文]-计算机研究与发展 2000(09)

5. 刘永丹 文档数据库若干关键技术研究[学位论文]博士 2004

6. 柯慧燕 Web文本分类研究及应用[学位论文]硕士 2006

7. 马玉春,宋瀚涛 Web中文文本分词技术研究[期刊论文]-计算机应用 2004(04)

8. 秦浩伟,步丰林 一个中文新词识别特征的研究[期刊论文]-计算机工程 2004(z1)

9. 王继成,潘金贵,张福炎 Web文本挖掘技术研究[期刊论文]-计算机研究与发展 2000(05)

10. 王本年,高阳,陈世福,谢俊元 Web智能研究现状与发展趋势[期刊论文]-计算机研究与发展 2005(05)

11. 彭雅 文本分类算法及其应用研究[学位论文]硕士 2004

12. 罗强 基于粗糙集理论的知识发现在web文本挖掘上的应用研究[学位论文]硕士 2003

13. 李波,李新军 一种基于粗糙集和支持向量机的混合分类算法[期刊论文]-计算机应用 2004(03)

14. 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[期刊论文]-中文信息学报 2004(01)

15. 王国胜,钟义信 支持向量机的理论基础--统计学习理论[期刊论文]-计算机工程与应用 2001(19)

16. 张学工 关于统计学习理论与支持向量机[期刊论文]-自动化学报 2000(01)

17. 萧嵘,王继成,张福炎 支持向量机理论综述[期刊论文]-计算机科学 2000(03)

18. 黄玲,陈龙 基于网页分块的正文信息提取方法[期刊论文]-计算机应用 2008(z2)

19. 黄文蓓,杨静,顾君忠 基于分块的网页正文信息提取算法研究[期刊论文]-计算机应用 2007(z1)

20. 汪志圣,李龙澍 Web文档分类方法的比较与分析[期刊论文]-滁州学院学报 2007(06)

21. 张卫丰,徐宝文,崔自峰,徐峻岭 一种基于粗糙集角分类神经网络的文档分类方法[期刊论文]-东南大学学报（英文版） 2006(03)

22. 邹加棋,陈国龙,郭文忠 基于图模型的中文文档分类研究[期刊论文]-小型微型计算机系统 2006(04)