山东大学

硕士学位论文

主题爬虫搜索策略研究

姓名：陈丛丛

申请学位级别：硕士

专业：计算机软件与理论

指导教师：石冰

20090405

摘 要

随着Web多元化信息的增长，传统的搜索引擎，即通用搜索引擎已经不能满足人们对个性化信息检索服务日益增长的需要。近年来，面向主题的搜索引擎应运而生，以提供数据更全面精确、时间复杂度更低的因特网搜索服务。

在主题搜索引擎中，网络蜘蛛以何种搜索策略访问Web，以提高效率，是近年来主题搜索引擎研究中的热点问题之一。Web的动态性、异构性和复杂性要求网络蜘蛛能够高效率地实现Web链接信息抓取。

首先，本文基于现阶段国内外网络爬虫的研究进展，在分析和比较现有主题网络爬虫搜索策略的优缺点的基础上，探讨了网络爬虫主题价值预测的准确性、重要性。

其次，作为主题网络蜘蛛搜索策略的核心部分，本文对主题信息的表示和主题相关性判断算法做了详细介绍。对于网页的主题相关性判别，使用目前较为常用的向量空间模型进行判别。

再次，本文提出了HITS改进算法Topic-HITS，把主题特征加入到HITS算法中，网页的链接结构从主题这个更细化的粒度进行链接分析，针对每一个页面，引入主题权威值向量，并进一步讨论了网站级别的权威值和中心值向量计算公式。

最后，为了提高网络爬虫的自适应性，本文针对传统网络爬虫存在的价值评价标准单一的问题，提出了一种基于综合价值的综合爬行策略，此策略根据不同的搜索阶段选择采用符合实际情况的最优搜索策略。本研究采用改进的HITS算法和自行设计的综合爬行策略相结合，实现了一个基于多种搜索策略的主题搜索引擎网络爬虫系统原型。实验结果表明，在此系统上不仅能够准确、自动地爬行到主题相关网页，而且还可节约网络带宽，具有良好的稳定性。

关键词：主题搜索引擎；爬行策略；爬行算法；内容分析；链接分析

ABSTRACT

With the growth of diversified Web information, the traditional search engines, namely, general search engines have been unable to satisfy people's personalized information retrieval service. In recent years, the topic-oriented search engine came into being in order to provide more comprehensive and accurate data, lower time complexity of Internet search services.

In the subject search engines, which search strategy Web spiders use to visit Web efficiently is one of hot issues in the study of search engines in recent years.The dynamic, heterogeneous and complex nature of networks demand Web spider to crawl Web link information efficiently.

First of all,based on domestic and international network research progress,based on the analysis and comparison of the existing search strategy 's advantages and disadvantages of Web spider,this paper discuss the accuracy and importance of topic value prediction to Web documents.

Secondly, as the core of a topic search strategy of Web spider, this artical detailed introduce the expression of topic information and relevance algorithm between topic and Web page. For the page relevance judgement, vector space model which is currently more commonly is used.

Thirdly,this paper presents enhanced HITS Algorithm,that is Topic-HITS, put the topic characteristics into HITS algorithm, analyze the link structure of Web pages from the topic which is a more detailed particle, for each page,introduce authority vector based on topics, and further discuss the calculation of authority and hub vectors from the site level.

Finally, in order to enhance the self-adaptive of Web spider, In this paper,to solve the single evaluation criteria,present an integrated comprehensive crawling strategy.This strategy changes according to different stages of search.In this study, the improved HITS algorithm and a comprehensive strategy are combined, implement a

search engine prototype based on a variety of search strategies.The experiment results show that this system not only be able to crawling pages related to the topic accurately and automatically, but also to save network bandwidth and have a good stability.

Keywords: topic search engine;crawling strategy;crawling algorithms;content analysis;link analysis

原创性声明和关于学位论文使用授权的声明

原 创 性 声 明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的科研成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本声明的法律责任由本人承担。

论文作者签名：陈丛丛　　日　期：09.4.5

关于学位论文使用授权的声明

本人完全了解山东大学有关保留、使用学位论文的规定，同意学校保留或向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅；本人授权山东大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或其他复制手段保存论文和汇编本学位论文。

(保密论文在解密后应遵守此规定)

论文作者签名：陈丛丛　导师签名：　日　期：09.4.5

第一章 绪论

1.1 研究背景

随着网络信息的迅速膨胀，人们对搜索引擎的首要关注是怎样从许多相关信息找到准确、有效的信息，查准率成为搜索引擎的首要目标。这也是人型搜索引擎面临的最人挑战。它们通常查准率极低，返回给用户是成千上万的查询结果，而有效结果可能只有几个甚至根本没有。由于客观存在的各种约束，查准率问题对于门户型搜索引擎来说是较难解决的，因为它们要在数秒内为数万用户同时服务，在上亿记录中查找出符合用户需求的信息。信息量大、时间短、语言的二义性等给门户型搜索引擎带来极大的挑战性。如何解决这个问题?借鉴专题型网站的出现、发展和成熟，我们认为搜索引擎朝着主题化的方向发展应该是解决问题的一条思路。

主题式搜索引擎[1, 2]，就是专门为查询某一学科或主题的信息而出现的查询工具，其设计思路是围绕着某一主题而展开的。借助于领域专家的经验和知识，可以提高内容编排的质量。对解决主题领域内的专业信息查询，主题式搜索引擎要比门户型搜索引擎有效得多。这种高度目标化、专业化的搜索引擎最大的优势在于，能够把具有相同兴趣点的人们集中在一个“主题社区”内，不仅集中提供各种专业资源，而且给人家提供了一个相互交流、共享经验和教训，展望行业发展前景的机会和场合，因此受到越来越多用户的欢迎。

网络蜘蛛即 Web Spider 是一个形象的名字，把互联网比喻成一个蜘蛛网，那么 Web Spider 就是在网上爬来爬去的蜘蛛。它是通过网页的链接地址来寻找网页，从网站的某一个页面(通常是首页)开始，读取网页的内容，找到该网页中的所有链接地址，然后通过这些链接地址寻找下一个网页，这样一直循环下去，直到把这个网站进而整个互联网上所有的网页都抓取完为止。主题蜘蛛是主题式搜索引擎的一个重要组成部分。由于主题式搜索引擎只提供主题领域内的信息查询，也就是说，非主题领域内的信息对其而言是无效信息。这就要求搜索引擎在进行网上信息采集时，必须采用主题式搜索策略。搜索引擎按照管理员顶先设定

的主题去采集网上相关信息，可以减少被采集的信息数量，提高索引数据库中的信息质量。

在主题蜘蛛的相关研究上涉及到许多方面，比如，相关主题的种子网站的选取，采用什么样的分布策略最大限度的搜集资源，主题相关度的确立等等。这些因素都会直接影响到主题蜘蛛的好坏。

主题式搜索引擎本身所具有独特优势，使得它能够解决门户搜索引擎无法解决或很难解决的问题。我们相信，主题式搜索引擎的出现将会成为今后搜索引擎的发展新动向，而主题蜘蛛作为主题式搜索引擎的重要组成部分，必然也会得到更多研究者的重视。

1.2 国内外相关研究现状

目前，主题搜索引擎的研究发展迅速，已经成为人们关注的热点。在国外，已经有相应的系统，下面介绍一些典型的系统：

1. Elsevier 的 Scirus 系统

Scirus 科学搜索引擎是一种专为搜索高度相关的科学信息而设计的搜索引擎，获得 2001 年《搜索引擎观察》授予的“最佳专业搜索引擎”奖。Scirus 是目前互联网上最全面、综合性最强的科技文献门户网站之一。它只面向包含有科学内容的网站，如大学和作者个人主页以及 Elsevier 自己的数据库。

2. 美国的国家科学数字图书馆的 Collection Building Program(CBP)

该项目旨在为科学、数学、工程和技术创建大规模的在线数字图书馆，试图研究在某一主题资源上自动建设数字图书馆的可能性。CBP 具有三个突出的特点：

第一，由于 CBP 项目面向教育、教学，主题精确度(Precision)比覆盖度(Recall)更为重要；

第二，CBP 并不存储资源原文，只是提供资源的 URL；

第三，CBP 只需要用户输入少量信息，如关键词，系统就可以自动的将有关该主题的最相关的 URL 返回给用户。

3. NEC 研究院的 CiteSeer

CiteSeer 是因特网上使用最广泛的针对计算机领域的科学论文检索系统。

CiteSeer 的核心是 ACI(Automaticaly Citation Index)，它可以自动地对网上的电子文件(Postscript 和 PDF 等格式)进行索引并分类。

4. Berkeley 的 Focused Project

该系统通过两个程序来指导爬行器：一个是分类器 Classifier 用来计算下载文档与预订主题的相关度。另一个程序是净化器 Distiller，用来确定那些指向很多相关资源的页面(在 HITS 算法中，称之为中心网页)。

在国内，研究主题搜索引擎的团队也越来越多。现在开始研究该领域的主要是一些大学的研究机构和一些搜索引擎公司。比如，北京化工大学就推出了关于化工方面的专业搜索引擎，其他方面如医药、林业等也有相应的产品。百度等多家搜索引擎提供商也相应的推出了图片搜索、mp3 搜索、行业搜索，这些都可以看作是主题式搜索引擎的应用。

1.3 主要研究内容

本文主要针对基于主题的网络爬虫，围绕提高主题网络爬虫的主题价值预测[3]的准确度、重要性及主题网络资源搜索的覆盖度来对主题网络爬虫的搜索策略问题进行了深入研究。具体研究的内容包括以下几个方面：

首先，本文基于现阶段国内外网络爬虫的研究进展，在分析和比较现有主题网络爬虫搜索策略的优缺点的基础上，探讨了网络爬虫主题价值预测的准确性、重要性及主题网络资源覆盖度的搜索策略，以提高主题爬虫的搜索效率。

其次，作为主题网络蜘蛛搜索策略的核心部分，本文对主题信息的表示和主题相关性判断算法做了详细介绍。对于网页的主题相关性判别，使用目前较为常用的向量空间模型进行判别。

再次，为了提高网络爬虫预测链接价值的准确性，提出了种子网站的选取方法，针对某一个特殊专业的领域，怎么样快速、高效的确定首先进行搜索的种子网站，在主题蜘蛛遍历搜索的过程中，怎样去不断的扩充种子网站，使主题蜘蛛持续的运行下去，搜集到更多的相关网页内容。在此基础上还提出了 HITS 改进算法 Topic-HITS，其主要思想是通过改进 HITS 算法来提高搜索相关网页的能力以及降低优先排序空间复杂度和时间复杂度，提高搜索效率、节约大量时间和资源。

最后，为了提高网络爬虫的自适应性，本文针对传统网络爬虫存在的价值评价标准单一的问题，提出了一种基于综合价值的综合爬行策略，此策略根据不同的搜索阶段选择采用符合实际情况的最优搜索策略。本研究采用改进的HITS算法和自行设计的综合爬行策略相结合，实现了一个基于多种搜索策略的主题搜索引擎网络爬虫系统原型。本系统综合了网页的相关性和重要性两方面的需要，不仅能够准确、自动地爬行到主题相关网页，从而提高信息搜索的查全率和查准率，而且还可节约网络带宽，具有良好的稳定性。

1.4 本文的组织

第一章 概述了主题搜索引擎的研究背景，研究现状，提出了本文所做的工作和组织结构。

第二章 指出了搜索引擎的分类，系统架构，指出通用搜索引擎技术已经不能适应主题的搜索，进而指出了进行面向主题的网络蜘蛛搜索策略研究的必要性。

第三章 提出了一种主题表示方法，并概述了网页分析和Web主题相关性判定的方法。

第四章 提出了基于内容评价的启发搜索策略，讨论了基于主题的Web信息提取的基本问题，重点是对主题页面在Web上的分布特征和主题相关性判定算法的研究。

第五章 讨论了基于链接结构评价的启发搜索策略，提出了HITS改进算法Topic-HITS算法；进而提出了综合内容和链接结构的搜索策略。

第六章 给出了我们设计的面向主题的网络蜘蛛的系统模型，并就搭建主题网络蜘蛛所面临的关键问题和相应对策做了简单的描述，试验验证了我们提出的主题表示方法以及改进的HITS算法无论从准确度还是从时间复杂度上具有优越性。

第七章为本论文的总结与展望。

第二章 WEB 搜索引擎概述

爬虫在设计之初，其目的是在给定爬行周期内，尽可能多地下载 Web 网页。但当面临着 Web 网页发展规模和数量成几何增长的现实时，通用的爬虫要想在爬行网页时既保证网页的质量和数量，又要保证网页的时效性显然已经是力不从心了。于是，爬虫的设计目的就变成了在给定的爬行周期内尽可能多地爬行高质量的网页，并且保持网页的时效性。而主题爬虫在上述设计目的的基础上，还考虑了网页与主题的相关度，尽可能多地下载相关网页，尽可能少地下载无关网页，提高主题资源的覆盖度。

一个好的爬虫需要达到以下两个要求：

a. 它必须要有一个好的爬行策略，即决定下一步要爬行哪些网页的策略。

b. 它必须要有一个高度优化的系统结构，且健壮性、可控性良好。

对于主题爬虫而言，它还必须对下载的网页进行与搜索主题的相关度分析，以决定其是否符合主题搜索的要求。因此要有一个好的分析方法。

由于相关主题资源的规模相对整个因特网来说要小得多，也相对容易控制和掌握，所以主题爬虫可以提供更精确的搜索结果。但相对普通爬虫，主题爬虫还需要解决以下两个主要问题:

1. 为了提高爬行效率，主题爬虫需要解决如何从待爬行 URL 队列中挑出最可能包含主题相关信息的网页进行爬行。
2. 对于每个已下载的网页，主题爬虫需要判断它与主题的相关性，用来指导以后的爬行过程。主题爬虫应尽量避免爬行主题不相关的和低质量的网页。

2.1 搜索引擎的分类

按照信息搜集方法和服务提供方式的不同，搜索引擎系统通常可分为以下三类：

1. 基于 Crawler（爬虫）的搜索引擎

这种搜索引擎的特点是利用爬虫程序（Crawler）自动访问 Web 站点，搜集站点上的网页，并根据网页中的链接进一步搜集其它网页，或转移到其它站点上。

爬虫搜集的网页经过分析处理后,建立索引,加入数据库中。用户查询时,检索数据库,返回结果。Internet 上最早出现的搜索引擎就是利用 Crawler 来建立数据库,"搜索引擎"这个词的原义也只是指这种狭义上的基于 Crawler 的搜索引擎。其优点是信息量大、更新及时、不需人工干预,缺点则是不能真正反映出网页的质量,返回信息过多,有很多无关信息。其例子:如 Google,百度,天网等。基于爬虫的搜索引擎是搜索引擎的主要类别。

2. 基于目录(Directory,也叫做 Catalog)的搜索引擎

以人工方式或半自动方式搜集信息,由编辑员查看信息之后,人工形成信息摘要,并将 URL 或站点置于事先确定的分类框架中。当用户查询某个关键词时,搜索软件只在这些描述中进行搜索。目录一般都是依靠一群专职编辑来建立和维护的。类似的例子有 Yahoo,该公司雇用了大约一两百名编辑来维护自己的目录。其优点是信息比较准确、导航质量高,缺点则是需要人工介入、维护量大、信息量少、信息更新不够及时。目前 google 也提供了目录式服务。

3. Meta 搜索引擎(也叫 "元搜索引擎")

Meta 搜索引擎[4] 也叫做 Multiple Search Engine,它的特点是本身并没有存放网页信息的数据库,当用户查询一个关键词时,它把用户的查询请求转换成其它搜索引擎能够接受的命令格式,并行地访问数个搜索引擎来查询这个关键词,并把这些搜索引擎返回的结果经过处理后再返回给用户。其优点是实现起来比较简单,返回结果的信息量更大、更全,其缺点则是不能够充分利用所使用搜索引擎的高级搜索功能。类似的例子有:WebCrawler、InfoMarket 等。

2.2 通用爬虫模型

2.2.1 通用爬虫的结构

爬虫是搜索引擎中最关键的部分,它的性能好坏直接影响着搜索引擎整体性能和处理速度。爬虫一般都要维护一个 URL 队列,用来存储已经发现但还没访问的 URL。URL 的访问次序一般有:广度优先、深度优先、随机访问等。通用爬虫基本流程图见图(2-1),结构见图(2-2)。

其各个模块的主要功能介绍如下:

1. 页面采集模块：该模块是爬虫和因特网的接口，主要作用是通过各种Web协议(一般以HTTP，FTP为主)来完成对网页数据的采集，然后将采集到的页面交由后续模块作进一步处理。
2. 页面分析模块：该模块的主要功能是将页面采集模块采集下来的页面进行分析，提取其中满足用户要求的超链接[5]，加入到超链接队列中。页面链接中给出的URL一般是多种格式的，可能是完整的包括协议、站点和路径的，也可能是省略了部分内容的，或者是一个相对路径。所以为处理方便，一般先将其转化成统一的格式。
3. 链接过滤模块：该模块主要是用于对重复链接和循环链接的过滤。
4. 页面库：用来存放已经爬行下来的页面，以备后期处理。
5. URL队列：用来存放经链接过滤模块过滤得到的URL，当URL为空时爬虫程序终止.
6. 初始URL：提供URL种子，以启动爬虫。

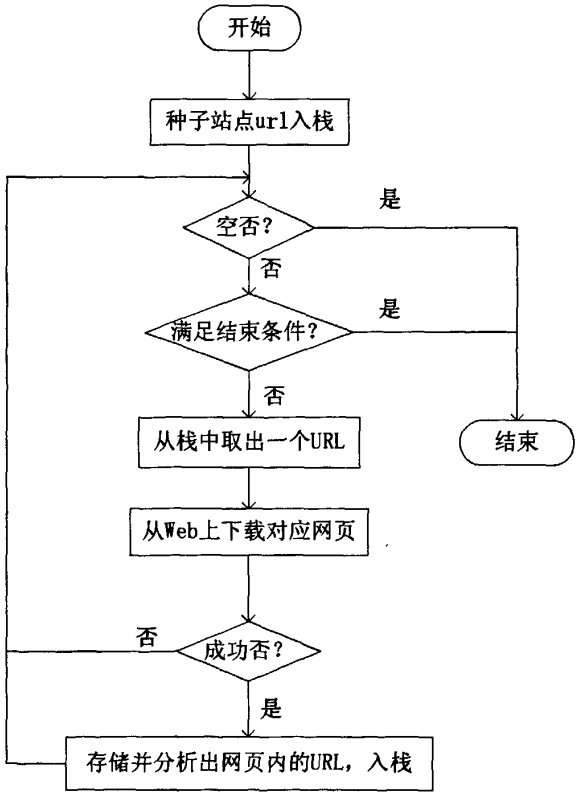


图2-1 通用爬虫流程图

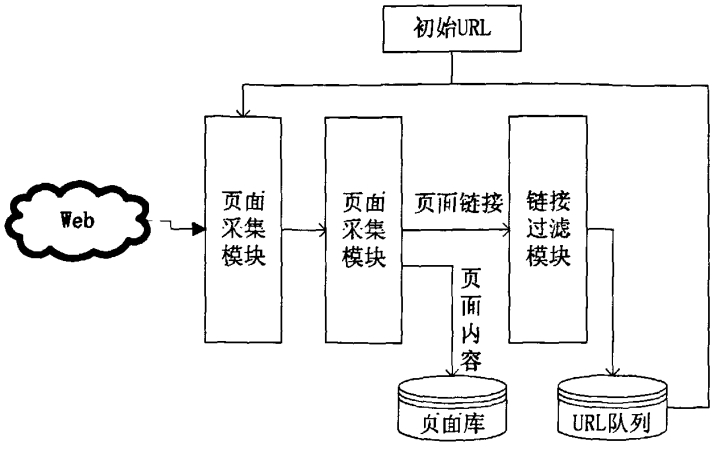


图 2-2 通用爬虫结构

2.3 主题爬虫模型

2.3.1 主题爬虫的原理

假设一个用户的 Web 信息检索表示为一组目标主题集合$T=\{t_1,t_2,t_3,...t_n\}$，每个主题t_i分别由主题爬虫来处理。我们可以用一组关键词$K=\{k_1,k_2...,k_n\}$来表示一个主题的关键特征，而用一系列样本集 W 来详细描述一个主题，其中$W=\{<ui,li>|ui$是一个样本$URL;li=0$或1,是该URL的正反例标号$\}$。l_i值为 1 表示u_i与主题相关，为 0 表示与主题不相关。主题爬虫一般从一组种子 URL 开始沿着已爬行页面中的超链接遍历 Web 以搜集更多的主题相关页面。对于主题爬虫来说，K 与 W 是它的初始学习资源。

整个 Web 从逻辑上可以看作一个有向图 G=(V,E)，其中图的节点集 V 表示页面的集合，有向边集 E 表示页面之间的超链接。给定一个目标主题，根据页面内容与目标主题的相关度，节点集 V 可以分为两部分：相关集 V+和不相关集 V-。主题爬虫的爬行过程可以看作对一个有向图的遍历过程，即从一组节点(种子节点)出发，尽可能多地搜索到那些属于 V+集合的节点，同时尽可能避免搜集到那些属于 V-的节点。

简单来说，主题爬虫就是指具有识别主题功能的爬虫，它尽可能多地爬行与

某个主题相关的 Web 资源，扩大该主题资源的覆盖度。

主题爬虫的基本思路是按照事先给出的主题，分析超链接和已经下载的网页内容，来预测下一个要爬行的 URL，保证尽可能地多下载与主题相关的网页、尽可能少下载无关网页。因此，主题爬虫需主要解决以下三个关键问题：

①怎样判断一个已经下载的网页是否与主题相关？对于已经下载的网页，因为我们可以知道它的文字内容，可以采用传统的文本挖掘技术来实现。

②怎样决定 URL 的访问次序？许多主题爬虫是根据已下载的网页的相关度，按照一定的原则，将相关度进行衰减，分配给该网页中的超链接，而后插入到优先级队列中。此时的爬行次序就不是简单的以深度优先或者广度优先为序，而是按照相关度大小排序，优先访问相关度大的 URL。不同主题爬虫之间的主要区别也就在于它是如何决定 URL 的爬行次序。

③怎样提高主题爬虫的覆盖度呢？这个问题要解决的就是如何穿过质量不够好(与主题不相关)的网页得到我们所感兴趣的网页，从而提高主题资源的覆盖度。

2.3.2 主题爬虫的结构

主题爬虫是在普通爬虫的基础上发展起来的，最早的主题爬虫是在通用爬虫的基础上改造而成的。其结构见图 2-3：

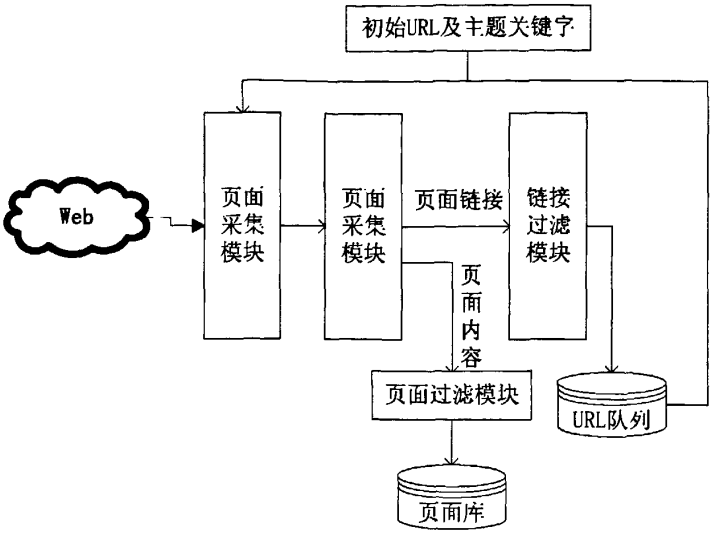


图 2-3 主题爬虫结构

设计者只是为爬虫提供了主题关键字。并在存储之前增加了一个主题识别步骤 (相关度判定)，若页面与主题相关就存储，否则就丢弃。尽管这样的爬虫也能实现对主题资源的爬行，但它在爬行中依然要遍历整个网络，并没有提高爬行的效率。网页爬行的数量和范围也依然严重依赖于给定种子站点的数量和质量。同时，这种爬虫还会下载很多与主题无关的资源，然后丢弃，造成对带宽和网络资源的严重浪费。

为解决以上的不足，研究者们采用了很多轻巧的算法和策略，来保证爬虫尽可能多地爬行相关网页，尽可能少地爬行无关网页，并且确保网页有较高的质量。研究的主要工作集中在如何将待爬行的 URL 按一定策略进行排序，使得与主题相关且质量高的 URL 优先爬行。在接下来的第四、五章中，我们将详细介绍并讨论 URL 排序的启发策略。

主题爬虫发展到现在，其结构要比原始的复杂得多，也有效得多。一般，一个主题爬虫[6] 包括以下三个关键组成部分：

页面相关度评价器。该模块主要特点是引入了文本分类的思想。在系统爬行之初，页面相关度评价器根据用户输入的关键字和初始文本信息进行学习，训练一个页面相关度评价模型。当一个被认为是主题相关的页面爬行下来之后，该页面就被送入页面相关度评价器计算其主题相关度值，若该值大于或等于给定的某

阈值，则该页面就被存入页面库，否则丢弃。

超链接评价器。该模块是主题爬虫的核心模块，主要用于评估从主题相关页面解析出来的 URL 与主题的相关度，并提供相关的爬行策略用以指导爬虫的爬行过程。URL 的超链接评价得分越高，爬行的优先级就越高。反之，若通过一定的评价策略，发现某链接与主题无关，则将该 URL 及其所有隐含的子链接一并去除，这个过程我们称之为剪枝。通过剪枝，爬虫就无需遍历与主题不相关的页面，从而保证了爬行效率。但是，剪枝的行为也可能将潜在的与主题相关的页面也剪掉。因此，超链接评价器所用的评价策略的好坏直接影响着爬虫的爬行效率以及爬行质量。

爬行器(页面采集模块)。该模块是任何爬虫都不可或缺的通用模块。该模块承担着连接超链接评价模块和页面相关度评价模块的重任。首先，爬行器从待爬行 URL 队列中取出超链接得分最高的 URL，将该 URL 相应的网页爬行到本地，然后，将该页面交由页面相关度评价器处理。在整个爬行过程中，爬行的次序和爬行策略都有超链接评价器提供。

2.3.3 性能瓶颈分析

WebSpider 是搜索引擎和信息检索系统的信息搜集代理，是系统中相当重要的组成部分。WebSpider 的实现原理很简单，一个 WebSpider 的基本工作流程如上所述，可参照图2-2和图2-3。但设计和实现一个好的、有实用价值的WebSpider，面临着许多方面挑战的。这包括：设计一个高度优化的系统结构(实现高性能)、提高 I/O 和网络利用的效率，保证系统的稳定性和健壮性。几乎所有大型通用搜索引擎的 WebSpider 都有一个高度优化的系统结构，因为现在的 Web 静态网页规模非常巨大，WebSpider 的爬行速度直接决定了搜索引擎的服务质量，即索引网页数据库的覆盖率和新鲜度。目前，大型通用搜索引擎均采用分布式的 WebSpider，利用多机并行工作，极大提高 WebSpider 的爬行速度。对于研究主题式搜索引擎的研究人员来讲，也需要一个良好设计的 WebSpider，因为研究人员设想的许多好的爬行策略都需要一个大的数据集来验证。下面是在设计时需要考虑的几个方面。

1. 网络通信延迟

在下载网页时，WebSpider 通过网络向 Web 服务器发送 HTTP 服务请求，Web 服务器处理该请求后，将数据包传输给 WebSpider 所在的主机。这一过程在程序中的实现是通过调用操作系统的 I/O 操作来完成的。对于阻塞式 I/O 操作，WebSpider 进程将被阻塞，以等待 I/O 操作的完成。在这种情况下，对于每一个 HTTP 服务请求，CPU 和网络都要有一段时间的空闲。为了避免阻塞在一个 I/O 操作上，我们采用多线程机制，让 WebSpider 同时启动多个线程，这样就可以并发地向多台 Web 服务器发送请求并且并发地处理 Web 服务器的返回响应，从而大大提高了 WebSpider 的爬行速度并且充分利用网络带宽。

除了上述采用的多线程机制外，还可以像 Google 的 WebSpider 采用单线程与异步 I/O 操作来并发执行多个请求，但这种设计的程序结构比较复杂。不过两种技术殊途同归，都能得到同样的效果。

在 JAVA 语言中设计多线程机制比较方便，因此我们将采用多线程技术来提高 Web Spider 爬行效率。

2. 礼貌爬行问题

所谓礼貌爬行问题就是当一个 Web Spider 向同一个 Web 服务器同时发出多个 HTTP 请求(即同时建立多个 TCP 连接)或在短时间内下载大量数据时，将会造成该 Web 服务器的负载过大，这样有可能导致 Web 服务器的崩溃或拒绝服务(因为有些 Web 服务器内运行了入侵监测系统，对同时过多的来自同一主机的 TCP 连接会被认为是入侵行为)。为避免这个问题的发生，在设计 WebSpider 时，必须限制 WebSpider 并发地向同一 Web 服务器发出 HTTP 请求的数量。

在本文的 WebSpider 设计时，将采取两方面的措施：

一是为每个工作线程(专门负责发送 HTTP 请求并下载网页的线程)设置一个待访问 URL 队列，所有主机 IP 地址都相同的 URL 放入同一个队列中，即每个队列的 URL 指向同一 Web 服务器，这样可以保证在任意时刻只有一个工作线程和某一特定的 Web 服务器连接；

二是在蜘蛛管理器(负责监控所有的工作线程)所维护的工作线程状态表中增加一项“通信时间间隔”项，用来记录某个工作线程与它所连接的 Web 服务器的通信时间间隔，并监控每个工作线程的通信时间间隔，如果通信时间间隔小于事先规定的值时，将对应的工作线程暂时放入工作线程的等待队列中，以等待时

间间隔大于事先规定的值后，唤醒等待的工作线程继续与所连接的 Web 服务器进行通信，这样可以防止短时间内从某一 Web 服务器上去下载大量数据。

3. 域名解析

一个 Web Spider 在与一个 Web 服务器建立连接前，必须使用域名解析服务(DNS)将 Web 服务器的主机名映射为一个 IP 地址。另外，每次新发现的 URL 在进行 URL 是否已访问的判断前，也要通过 DNS 来获取 URL 的主机名。在多线程的机制下，许多工作线程都向本地 DNS 服务器发出 DNS 请求，会给本地 DNS 服务器造成很大负荷。这样，域名解析也就可能成为系统的性能瓶颈。解决这个问题的方法，一般是在 WebSpider 主机上建立缓存区存放 DNS 结果来缓解这个问题。另外，许多 WebSpider 都编写了自己的 DNS 解析器，如康柏系统研究中心(SRC)的 Mercator 就是使用自己编写的多线程 DNS 解析器，极大缩减了每个工作线程在 DNS 查找上所花的时间，从而使域名解析不再成为一个性能瓶颈。

本文的 WebSpider 在实现时，也是采取了 DNS 结果缓存机制，即在解析 URL 时，先在本机的缓存区中查找，如果未命中，才向本地 DNS 服务器发出 DNS 请求。

2.3.4 与普通爬虫的区别

举例来说，假设在整个 Web 上有 T 个网页文件，其中关于某主题的文件个数为 S。通过普通爬虫在一个爬行周期(7 天)能够爬行所有 Web 资源的 20%，而其中关于某个主题的资源为 5%，则在一个爬行周期内，只能采集到 1%(20%*5%)的主题相关资源，而且还有大量的资源与主题无关。但如果采用主题爬虫，在一个爬行周期内，能够爬行所有 Web 资源的 10%，其中 50%与主题相关，则在一个爬行周期内采集到了 5%(10%*50%)的主题相关资源，而且只有 5%(10%-5%)是资源浪费部分。

这样，主题爬虫的主题爬行覆盖度 R=T*10%*50%/S=5%*(T/S)，而普通爬虫的主题覆盖度 R=T*20%*5%/S=1%*(T/S)

通过上述分析发现，主题爬虫爬行资源的数量只有普通爬虫的二分之一，而它的主题资源覆盖度却是普通爬虫的五倍，能发现更多的 Web 主题资源。这也正是主题爬虫存在的意义。它除了可以提高主题资源的覆盖度外，在减少网络负

担、优化网络结构等方面也均有显著意义。

相比于普通爬虫，一个主题爬虫需要能有效地发现主题相关的文档，并且能通过网络内容和链接结构来指导自身的资源发现过程。

第三章 爬行策略概述

3.1 主题表示

主题的表示形式一般有示例型文档集合和分类法两种方式。Sogou 分类数据属于前者；ODP[7] 属于后者。可采用传统的 TFIDF 方法将前者的每个主题转化为词语向量的形式。而文中提出一种处理方法，可将后者的主题转换为词语向量的方式，从而将网页与主题、主题与主题的相似度计算转化为向量余弦值的求解。基本步骤如下：

(1) 取分类法的前 m 层节点，对每一个节点分别建立 CT(category-term)矩阵，该矩阵包含两行 n 列，元素 $MatrixCT(i,j)$ 代表在第 i 行出现的第 j 个关键词的权重，即词频。第 1 行代表本节点描述，第 2 行代表所有子节点的描述。

(2) 将每个节点的 CT 矩阵的两行合并，合并时，第一行与第二行元素权重比例为 $s:1$，得到每个节点的 CT 向量，即 $V_{CT}=s\times V_{Row1ofMatrixCT}+V_{Row2ofMatrixCT}$。

(3) 自底向上，将子节点的 CT 向量以 $1:t$ 的权重比例加入父节点的 CT 向量，即 $V_{CT}(father)=t\times V_{CT}(father)+\sum_{foreachchild_i} V_{CT}(child_i)$。层层传递，最后得到顶级节点 node 的 TT 向量 $V_{TT}(node)$。

$V_{TT}(node)$ 是根据分类法形式得到的关于节点 node 的词语向量，通常可以表示某一类主题，如示例中的健康主题。有时分类法中的主题信息不够充分，含有的关键词比较少，此时，可以将该向量与示例型文档集合得到的向量以一定的比例组合，即 $V_{Ci}=\beta\times V_{TT}(node)+(1-\beta)\times V_{DocsCi}$

3.2 网页分析算法

网页分析算法可以归纳为基于网络拓扑、基于网页内容和基于用户访问行为三种类型。

1. 基于网络拓扑的分析算法

基于网页之间的链接，通过已知的网页或数据，来对与其有直接或间接链接关系的对象（可以是网页或网站等）作出评价的算法。又分为网页粒度、网站粒度和网页块粒度这三种。

PageRank 和 HITS 算法是最常见的链接分析算法，两者都是通过对网页间链接的递归和规范化计算，得到每个网页的重要度评价。PageRank 算法虽然考虑了用户访问行为的随机性，但忽略了绝大多数用户访问时带有的目的性，即网页和链接与查询主题的相关性。针对这个问题，HITS 算法提出了两个关键的概念：权威型网页(Authority)和中心型网页(Hub)。在本文的第五章对这两种算法还将展开进一步的讨论。

2. 基于网页内容的网页分析算法

基于网页内容的分析算法指的是利用网页内容（文本、数据等资源）特征进行的网页评价。网页的内容从原来的以超文本为主，发展到后来的动态页面（也称为 Hidden Web）数据为主，后者的数据量现在已大大超过了直接可见页面数据（PIW，Publicly Indexable Web）的数据量。另一方面，多媒体数据，Web Service 等各种网络资源形式也日益丰富。因此，基于网页内容的分析算法也从原来的较为单纯的文本检索方法，发展为涵盖网页数据提取、机器学习、数据挖掘、语义理解等多种方法的综合运用。

3. 用户协作型网页的分析算法

链接提供的网页关联度往往带有噪音，网络的异构性和动态性使得对链接结构的建模很难达到令人满意的效果。而用户的访问模式往往可靠反映了资源的主题相关性，且具有时效性，可即时反应网络链接的变更等情况。因此可以通过用户协作、学习浏览模式来抓取网页，协作抓取需要获取用户浏览行为，一般有两种方法：日志挖掘和用户标注。

用户浏览模式挖掘法对与某一特定查询谓词相关的网页作相似性建模，以大

量公共域名代理用户访问日志为参考，经过对大群组用户信息过滤，统计并总结出了三种需要考虑的用户访问信息：对不同网页访问频率；对不同网页特征访问频率；访问同一主题网页的时间局域性。试验表明，协作抓取比基于链接的智能抓取（Intelligent Crawling）策略有更好的准确性。

3.3 WEB 文本相关性判定方法

目前，主题相关性判别算法的研究主要可以分为四个大类：基于元数据的判别；基于链接标签数据的判别；基于链接结构分析的判别；基于页面内容的语义判定。

基于元数据的判别

元数据（metadata)包含有数据的数据、信息的信息含义。人们在研究 Web 信息检索的早期就发现，利用元数据来增加 HTML 的结构特征对 Web 信息检索有帮助。

因此，HTML 规范从 2.0 版本开始引入了<Meta>这一标签（tag），用以在 Web 页面中标注元数据，通常表达形式为<:Meta name="…", content="…">Metadata 信息对于主题相关性判别是有用的，比如包含一些 description 属性、keywords 属性。但是没有形成一个普遍接受的标准。因此，Metadata 信息在主题 Web 信息提取领域应用不多见。

基于链接标签数据的判别

页面作者添加到另外一个页面的超级链接的行为，实际上表明了对所链接页面的认可，通过对链接标签属性、链接文本、链接上下文文本等链接标签数据进行主题相关的评价和描述，链接标签数据信息能够有效地指导主题 Crawler 进行面向主题的 Web 信息提取。

在 HTML 页面中，主要有 4 种标签用于超链接：1.Anchor 标签；2.Image 标签；3.Map 和 area 标签；4.Frame 和 iFrame 标签。在这 4 种标签中，Anchor 标签最常用，和超链接相关的有 title，href 和内嵌文本(链接文本)等几种属性。Image 标签和超链接相关的属性有 src 和 alt。对于 Map 和 area 标签，它们相关的属性和 Anchor 标签基本相同。Frame 和 iFrame 一般与 Frameset 一起使用，用以网页进行分割，相关的属性主要包括 src 和 name 等属性。针对 HTML 链接标签的统

计分析表明，链接标签的 href 属性和 title 属性、链接文本和链接上下文文本在 Web 中分布最为常见。但是引入链接上下文信息的判断后，虽然可以使得检索得到的相关文档数增加，检索精确度却反而下降。

因而在实际的 URL 的主题相关性判别中，我们主要使用了链接 URL、链接文本、链接 title 属性等作为启发信息进行主题相关性判别。

为了估算链接所指页面与主题的相关性，常见的方法是判断在 URL、链接文本(AnchorText)、链接标题(Anchor Title)中是否包含主题关键字来计算相关性权重值。考虑到自然语言中常见的词性近似现象，因而，还需要考虑近义词的情况。

基于分析链接标签数据的相关性权重算子如下：

$rel(link,t)=A*R(url,t)+B*R(txt,t)+C*R(ttl,t)$ 其中，A,B,C 为 3 个大于等于零小于等于 1 的常数，用于控制各个类别的链接标签数据在整体判别中的权重。该算子需要一个相似词库，用以判定链接标签数据中词与主题词之间的相似度，即公式中的 R 计算，同义词之间的相似度为 100%，近义词之间的相似度 50%-100%，远义词之间的相似度 0%-50%。

基于链接结构分析的判别

Web 是 Internet 上的超文本系。多项研究表明，利用 Web 中丰富的超链接[8]，可以从中挖掘出 Web 中许多重要的信息。针对超链接信息的分析对进一步理解超文本语义，以及给用户提供更优质服务有着相当大的帮助。这些研究超链接的工作叫做结构分析。

链接分析的研究思路基于这样一个认知假设：即把超链接看作是对它所指的页面的认可。在这样的假设之下，当页面 A 通过超链接指向页面 B 时说明两点：1.页面 B 与页面 A 是相关联的：2.页面 B 是值得关注的质量较好的页面。

当然链接并不都是完全可靠，超链接中也有纯粹起导航作用的或者是广告链接，或者有时为了达到某种目的而添加的欺骗性链接。不过，总体上来看，Web 上整个链接集合所反映的情况还是比较可靠和准确，因为不良链接的整体效应远没有重要链接的整体效应强。为了更准确和有效地评估链接，在进行具体的算法分析之前需要识别和除噪的工作。

Web 超链分析算法可以用来提高搜索引擎的查询效果，可以发现因特网上的重要社区，可以分析某个网站的组织结构和权威性，可以用来实现文档的自动分

类。目前 Web 超链分析算法主要有两种：基于随机漫游模型的 PageRank 算法和基于 Hub 和 Authority 相互加强模型的 HITS 算法。这两种算法在第五章中做了详细介绍。另外，在第五章还提出了改进的 HITS 算法（Topic-HITS 算法）。

基于页面语义信息的判别

最好的页面主题相关性判别方法还是从基于语义理解的方面着手解决，尽管这样做往往要花费更高的计算代价。从目前应用的实际情况来看，文本的主题相关性判别方法仍然是基于关键词的。主要有全文本扫描，布尔模型，向量空间模型，概率模型等，这些方法均是信息检索领域中的经典方法。

第四章 基于内容的爬行策略

基于内容的爬行策略根据所有已经下载到的页面的文本内容以及链接文本的内容来给所有看到的链接评分。也就是说，链接的重要与否是跟所有已经看到的文本内容有关的。在本节中，首先讲述对链接文本进行判断所需要的自然语言处理的技术：中文的分词和文本向量模型以及文本相似度的计算，然后介绍我们的系统中用到的两种基于文本内容的爬行策略。

4.1 WEB 文档处理

为了便于 Web 文档进行各种处理，首先需要以数学的方式表示出来。需要做以下两项工作：

1) 预处理

包括：去噪、去停用词(Stop Words)、词根还原 Stemming)、分词、词性标注、短语识别等。

2) 文档表示模型选择

文本表示方法有多种，比如：布尔逻辑模型(Boolean Model)、隐性语义索引(LSI)模型、概率模型(Probablisitc Model)、向量空间模型(VSM)等。

其中，向量空间模型是应用较多且效果较好的方法之一。

构成文本的词汇，数量是相当大的，表示文本的向量空间的维数也相当大，可以达到几万维。而实际上，其中只有部分词条具有区分度。因此，我们可以通过获取文档的特征词条，构成特征向量。以特征向量来表示文档，不仅达到降维效果，也提高以后工作(比如查询匹配)的准确率。

向量空间模型的基本思想是以向量来表示文本：$(w_1, w_2, ...w_n)$，其中 w_i 为第 i 个特征项的权重，权重可以有多种计算方法，一般采用的是 TF-IDF 的方法，关于向量空间模型我们将在后面做详细介绍。

传统的文本特征向量获取一般是通过一些分词算法和词频统计方法。不过这些方法构成的文档向量其实维数还是很大，于是有很多这方面的研究，比如基于

遗传算法的、基于因子分析思想的、基于文本质心的。

4.1.1 中文的分词处理

词是最小的能够独立活动的有意义的语言成分。但汉语是以字为基本的书写单位，词语之间没有明显的切分标记，因此，中文词语分析是中文信息处理的基础与关键。对于限定主题的网络爬行器来说，中文的分词[9-11] 处理也是最基本的一步。采用隐马尔科夫模型进行的中文分词是目前中文分词的最常见也是最好的方法。但是对于我们的网络爬行器中的文本处理来说，并不需要很高的切分准确率，但是分词的速度确实很重要的指标，因为我们要迅速的处理大量的网页。所以，在我们的系统中，采用了简单的前向最大匹配的分词方法。并且去除了单个字作为词的情况，所有的英文及标点都作为词的边界来处理了。

4.2 关键词抽取

关键词抽取中不存在预定义的词汇表，关键词是从文档正文中抽取的，文档经过规范化处理被分成一系列的候选词，最后用一种有监督的学习算法确定候选词是否为关键词。目前关键词抽取的方法主要分为以下三种：基于词典的方法、基于词义方法、以及基于统计学习的方法。

基于词典的方法

基于词典的方法的基本思想是：将文档的种类可以按不同的领域进行划分，一般而言，有相当一部分词它们表达文档主题特征的能力受到文档类型的限制，针对不同类型的文档，各抽取一定量文本样本，采用一定的训练算法提取特征词，建立针对该领域的特征词词典，以此作为文档的背景词典。在进行关键词抽取时，根据特征词词典，并综合考虑词频、词长等信息，对文档中出现的词进行加权，最终按照权重抽取关键词。如图 4-1 所示：

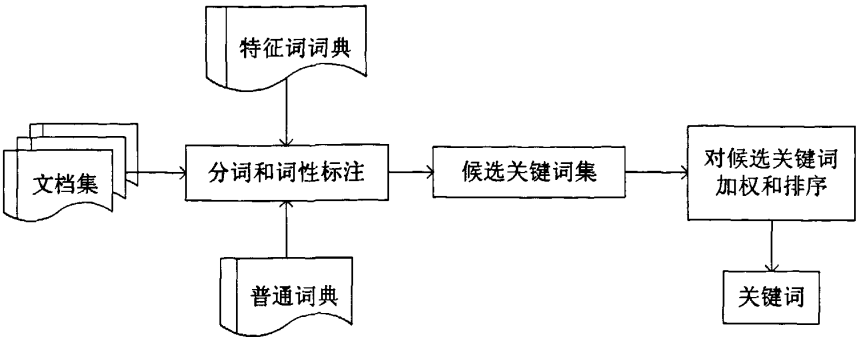


图 4-1 基于词典的关键词抽取

这种方法在文本自动分类、主题词提取、主题标引时经常被用到，而且效果显著。其主要特点为：提取准确率较高，但由于受到背景词典的限制，该方法查全率较低，而且抽取范围仅限于出现在特征词词典中的关键词，其抽取结果直接受到背景词典的限制。

基于词义的方法

基于词义的方法的基本思想是：给定的文档是按照一定意义对词汇进行排列组合的符号串，是围绕文献主题有关方面所做的判断、推理、结论等等，一个词如果是关键词的话，就不可能是孤立的，围绕着它必然会展开论述，主题关键词之间构成一个语义结构图，词语按照所讨论的关键词形成语义聚类。将这些语义聚类划分出来，深入挖掘文献中所包含的语义信息，就可以提高关键词的提取准确率。如图 4-2 所示：

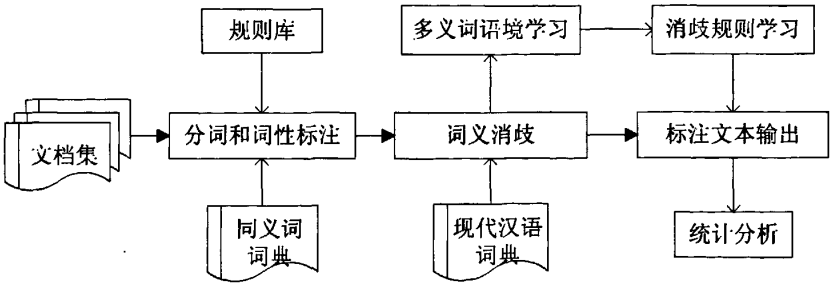


图 4-2 基于词义的抽取

基于词义的方法虽然在一定程度上代表了自然语言理解的发展方向，但是这种方法实现结果直接受到用户所建立“规则库”性能的影响，还需要进行大量的词义排歧、同义词识别工作，目前计算机在处理这些技术方面还存在着一定的局限性。同时对于未登录词，也缺乏相应的处理机制，因此这种方法还只能处于试

验阶段。

基于统计学习的方法

基于统计学习方法的基本思想是:在表达文章主题时，起主要作用的是名词、动词等实词，这部分词中出现频率越高，出现位置越特殊(如标题，首段等等)标记、特殊提示的词，对表达文章的主题越有意义。在进行关键词提取时，综合考虑这些特征进行加权。

4.3 相关词抽取

基于领域知识抽取相关词

随着金融经济学、生物医学、分子生物学等新型领域的兴起，大量的领域知识随之涌现，为了使领域专家能够有效的获取、分析和利用这些信息，这些领域的资源库(如美国 Mdelien 生物医学文献数据库)必须不断的被调整更新使之包含最新的领域知识。而这些资源库的维护和更新需要的大量的领域专家来为新的信息进行收集和分类，不利于长期发展，越来越多的专家开始通过文本挖掘的方法从大量的领域文献中定位、收集和抽取相关知识。

基于查询扩展抽取相关词

随着 Web 技术的巨大发展和日益普及，Internet 越来越成为人们搜寻各方面信息的主要来源，搜索引擎也在人们的日常生活和学习中发挥着无法替代的重要作用.然而，由于大量同义词和多义词的存在，用户在提交查询时使用的词往往不尽规范，与文档索引使用的词或词组有很大差别，这就给现今基于关键词的查询系统带来巨大的困难，也成为长期困扰信息检索领域的基本问题。

基于查询扩展[12, 13] (Query Expansion)抽取相关词，即在原来查询的基础上加入与用户用词相关联的词，组成新的更长、更准确的查询，这样就在一定程度上弥补了用户查询信息不足的缺陷，也即所谓的“词典问题“。

Furnas 第一个发现了所谓的“词典问题”(Dictionary Problem)。他们的实验表明，通常情况下，两个人使用同样的关键词描述同一物体的几率小于 20%。在当前的搜索引擎使用过程中，这个问题变得更加尖锐。通过对微软旗下的 MSN 中的 Encarta(http://encarta.msn.com)在线百科全书网站连续两个月的用户查询记录进行分析发现，49%的用户查询仅有一个单词，33%的查询由两个单词组成，

用户平均使用 1.4 个单词描述他们的查询。这样，传统的基于关键词的向量空间模型无法发挥正常的作用。同时，在许多情况下，用户使用的词即使在文中出现，也未必在相关文章中具有足够的权重，即仅靠用户提交的短查询无法提供检索出相关文档的足够信息，需要通过相关词进行扩展。

4.4 特征选择及权值计算

在前面关于向量空间模型[14]的介绍中，曾提到索引项的权重分配问题。在基于文本的分类中，一般认为某个词在某个文档类别中出现的频次较高，而在其他类别中不经常出现，就说明该词对该文档类别的识别有较大的贡献。由于在分类问题中，文本集合的特征(索引项)的维数都很高，通常达到几万维。对于 t 维的空间向量，仅仅按照特征出现与否，其各维特征的组合就有 2^t 种，其计算量是非常巨大的。此外，由于一些噪声特征的存在，会导致分类性能的下降。因此需要进行特征选择，剔除无用的和分类参考价值较小的特征，以达到降低空间维数和减小计算复杂度的目的。

在特征选择时，通常的做法是根据在训练数据集合上统计得到的特征分布情况，采用一些权值计算方法对各维特征进行加权处理，从中选择权值较高的一些特征，从而实现特征的选择和降维。常用的特征选择方法包括文档频率，信息增益、交叉熵、互斥信息、χ^2 统计量、文本证据权、几率比和特征强度。

文档频率(DF)

文档频率(Document Frequency)是指在训练数据集中包含一个词的文本的个数。在训练集中，如果一个词出现的文档频率过大，如一些功能词几乎在所有的文档中都出现，那么这个词对分类的作用不大，可以将其滤除。反之，如果词的文档频率过小(极端地，只在一个文档中出现)，则它对文档类没有代表性，甚至可能是噪声，应予以滤除。在传统的信息检索领域，通常认为那些文档频率过高或过低的词对于信息检索没有帮助。这种方法降维最为简单，容易扩展到大的数据集，复杂度随数据集线性增长。DF 也存在缺点，比如某些特征在全部文档集合中出现次数较少，却相对集中于某些类中，如果被滤除会使分类信息受到损失。另外对于类别较少的情况(两类问题)，很多 DF 高的特征反而会对分类起到重要

作用。

信息增益(IG)

信息增益(Information Gain,IG)是机器学习领域中被经常用来衡量一个特征是否良好的指标。它所衡量的是在获知一个特征文本中出现或不出现时，所获得的信息的比特数。设$\{Ci\}, i=1,2,...k$为k个类别的集合，则特征w的信息增益定义为：

$$IG(w)=P(w)\sum_{i=1}^{k}P(Ci|w)\log\frac{P(Ci|w)}{P(Ci)}+P(\overline{w})\sum_{i=1}^{k}P(Ci|\overline{w})\log\frac{P(Ci|\overline{w})}{P(Ci)} \qquad (4\text{-}1)$$

其中$P(Ci)$为类别Ci的概率，$P(w)$为特征w在训练集中的概率，$P(Ci|w)$为w出现时，类别Ci出现的概率，$P(\overline{w})$为在训练集中不出现特征w的概率，$P(Ci|\overline{w})$为w不出现时，类别Ci出现的概率。通过设定一定的阈值，将IG值低于阈值的对应特征滤除，认为它不是合格的特征。

交叉熵(CE)

与信息增益相类似，交叉熵(Cross Entropy)也是利用特征的出现情况来衡量分类性能的：

$$CE(w)=P(w)\sum_{i=1}^{k}P(Ci|w)\log\frac{P(Ci|w)}{P(Ci)} \qquad (4\text{-}2)$$

交叉熵与信息增益的不同在于，它只考虑了特征发生时的情况，不象信息增益那样同时兼顾特征出现和不出现的情况。通过对公式4-1和4-2比较可以发现，交叉熵没有第二项，即特征不出现时的权值分配。一般的，如果一个特征倾向于某个类，那么，该特征不出现也会提供类别的分布信息，只不过这个信息是间接的、辅助的，不如直接信息有力度。

互斥信息(MI)

互斥信息(Mutual Information)用来衡量特征与类别之间的统计独立关系的，考虑特征与单个类别的互斥信息为：

$$MI(w,Ci)=\log\frac{P(w\wedge Ci)}{P(w)\times P(Ci)} \qquad (4\text{-}3)$$

其中，$P(w\wedge Ci)$定义为特征w与类别Ci的同现概率，$P(w)$定义为特征w出

现的概率，$P(Ci)$ 定义为类别 Ci 出现的概率，在包含 k 个类别的集合上，特征 w 的互斥信息有两种定义方法；

(1) 平均互斥信息定义为：

$$MI_{avg}(w)=\sum_{i=1}^{k}P(Ci)MI(w,Ci) \qquad (4\text{-}4)$$

(2) 最大互斥信息定义为:

$$MI_{max}(w)=\max_{i=1}^{k}MI(w,Ci) \qquad (4\text{-}5)$$

互斥信息对于边缘概率密度很敏感，在相同条件概率的情况下，训练集中出现概率低的特征将获得较大的互斥信息，因此在词频分布较宽时，分类性能得不到保证。

4.5 基于内容评价的搜索模型

4.5.1 布尔模型

布尔模型[15] 是建立在集合理论和布尔运算上的一种非常简单的检索模型。在布尔模型中用一组索引项表示文档，只考虑每个索引项是否出现在文档中，并为其赋予一个布尔值，因此索引项与文档之间的联系可表示为 $W_{i,j}\in\{0,1\}$。检索字 q 由“与(and)”、“或(or)”、“非(not)”连接起来一系列索引项构成。设 $\vec{q}_{dnf}$ 是检索字 q 的主析取范式，其中的任意一个析取项，则文档 d，与检索字 q 之间的相似度按照下式计算：

$$sim(d_j,q)=\begin{Bmatrix}1 & if\ \exists\vec{q}_{cc}\,|\,(\vec{q}_{cc}\in\vec{q}_{dnf})\wedge(\forall k_i,g_i(\vec{d}_j)=g_i(\vec{q}_{cc}))\\ 0 & otherwise\end{Bmatrix} \qquad (4\text{-}6)$$

其中 g_i 是取向量中第 i 个分量的操作。

$sim(d_j,q)$ 也是用布尔值表示。相似度为 1，则预测文档 d_j 与检索字 q 相关，否则不相关。这种模型的最大优点是简单、直观，文档和索引之间就是有或没有关系，同样检索结果也只是相关或不相关。检索过程值进行简单的布尔运算，系统的运算开销非常小。因此当前多数系统都支持布尔模型，作为系统检索的一个

基本支持功能。但是该模型检索性能较差。因为所有检索出的文档要么太多，要么太少，且不能对输出结果进行排序，不能利用相关反馈提高检索性能。

4.5.2 向量空间模型

计算机并不具有人类的智能，人在阅读文章后，根据自身的理解能力可以产生对文章内容的模糊认识，而计算机并不能轻易地“读懂”文章，从根本上说，它只认识 *0* 和 *1*，所以必须将文本转换为计算机可以识别的格式。根据“贝叶斯假设”，假定组成文本的字或词在确定文本类别的作用上相互独立，这样，可以就使用文本中出现的字或词的集合来代替文本，不言而喻，这将丢失大量关于文章内容的信息，但是这种假设可以使文本的表示和处理形式化，并且可以在文本分类中取得较好的效果。

目前，在信息处理方向上，文本的表示主要采用向量空间模型(VSM)。向量空间模型的基本思想是以向量来表示文本：$(w_1,w_2,...w_n)$，其中 w_i 为第 i 个特征项的权重，那么选取什么作为特征项呢，一般可以选择字、词或词组，普遍认为选取词作为特征项要优于字和词组，因此，要将文本表示为向量空间中的一个向量，就首先要将文本分词，由这些词作为向量的维数来表示文本，最初的向量表示完全是 *0-1* 形式，即，如果文本中出现了该词，那么文本向量的该维为 *1*，否则为 *0*。这种方法无法体现这个词在文本中的作用程度，所以逐渐 *0-1* 被更精确的词频代替，词频分为绝对词频和相对词频，绝对词频，即使用词在文本中出现的频率表示文本，相对词频为归一化的词频，其计算方法主要运用 TF-IDF[16] 公式，目前存在多种 TF-IDF 公式，我们在系统中采用了一种比较普遍的 TF-IDF 公式：

$$W(t,\bar{d})=\frac{tf(t,\bar{d})\times\log(N/n_t+0.01)}{\sqrt{\sum_{t\in\bar{d}}\left[tf(t,\bar{d})\times\log(N/n_t+0.01)\right]^2}} \qquad (4\text{-}7)$$

$W(t,\bar{d})$ 为词 t 在文本 $\bar{d}$ 中的权重，而 $tf(t,\bar{d})$ 为词 t 在文本 $\bar{d}$ 中的词频，N 为训练文本的总数数，n_t 为训练文本集中出现 t 的文本数，分母为归一化因子。

另外还存在其他的 TF-IDF 公式，例如：

$$W(t,\bar{d})=\frac{(1+\log_2 tf(t,\bar{d}))\times\log(N/n_t)}{\sqrt{\sum_{t\in\bar{d}}\left[(1+\log_2 tf(t,\bar{d}))\times\log(N/n_t)\right]^2}} \qquad (4\text{-}8)$$

该公式中参数的含义与上式相同。

文本经过分词程序分词后，首先去除停用词，合并数字和人名等词汇，然后统计词频，最终表示为上面描述的向量。

向量空间模型的最大优点在于它在知识表示方法上的巨大优势。在该模型中，把文档内容简化为特征项及其权重的向量表示，即文本内容被形式化为多维空间中的一个点，通过向量的形式给出，把对文本内容的处理简化为向量空间中向量运算，使问题的复杂性大为降低。而权重的计算既可以用规则的方法手工完成，又可以通过统计的办法自动完成，便于融合统计和规则两种方法的优点。也正是因为把文本以向量的形式定义到实数域中，才使得模式识别和其他领域中的各种成熟的计算方法得以应用，极大提高了自然语言文本的可计算性和可操作性。所以说，文本的形式化表示方法—向量空间模型是基于文本处理的各种应用得以实现的基础和前提。

向量空间模型是一种不考虑特征项出现顺序的词袋(Bag of Words)文本表示模型，这种模型虽然带来了计算和操作上的方便，但是却损失了大量的文本结构信息。而这些信息在自然语言中是至关重要的(如句子中词序信息等)。另外，在权重和相似度的计算中也做了许多简化工作：一是对不同的语言单位构成的特征项大都只考虑其统计信息并采用统一的权重计算方法，而这种计算只是经验公式并没有很好的理论基础，所以计算出的权重未必能真实反映各项的重要性。二是向量空间模型是建立在所有项两两正交这一假设基础之上的($W_{ik}*W_{jt}=0$ 当 $k\neq t$ 时)，没有考虑特征项之间的相关性，这显然是不太合理的。对于自然语言这种有着非常丰富语言现象的研究对象来说，这种假设显然是过于严格的，不能很好地反映自然语言的特征。目前已经有许多改进项权重计算的方法，但是效果并不明显，原因在于语义关系实际上是一个很复杂的运算，采用简单的初等运算代替它，误差势必存在。

文本相似度计算

利用 VSM 可以计算两个文本的相似度[17, 18] (Similarity)：两个文本 D_1 和 D_2 之

间的(内容)相关程度(Degree of Relevance)常常用它们之间的相似度 $Sim(D_1,D_2)$ 来度量。当文本被表示为向量空间模型时，我们可以借助于向量之间的某种距离来表示文本间的相似度，常用向量之间的内积进行计算，即下式所示：

$$Sim(D_1,D_2)=\sum_{k=1}^{N}W_{1k}*W_{2k} \qquad (4\text{-}9)$$

或者用夹角余弦值来表示为式：

$$Sim(D_1,D_2)=\cos\theta=\frac{\sum_{k=1}^{N}W_{1k}*W_{2k}}{\sqrt{\sum_{k=1}^{N}W_{1k}^{2}\sum_{k=1}^{N}W_{2k}^{2}}} \qquad (4\text{-}10)$$

与布尔模型相比，向量空间模型有很多优点，由于权值的连续取值，使得在实际运用中，向量空间模型的检索效果有很大的提高。同时相似度函数的取值也是连续的，因此对检索结果可以进行排序，从而反映出结果的相关程度。

4.6 基于内容评价启发搜索算法

基于内容评价的网络爬虫，主要是根据主题信息(如:关键词、主题相关文档)与页面或链接文本“语义”的相似度来评价链接价值的高低，以此决定其搜索策略。不同的分析方法构成了不同的启发策略和相应的算法。主要策略有 Best First Search、Fish Search、Shark Search 等算法。

4.6.1 Best First Search 算法

Best First Search 启发策略的基本思想是：给定一个待爬行 URL 队列，从中挑选最好的 URL 优先爬行。在一般的爬行器中，爬行主题是采用关键词集合表示的。URL 的优先级是根据主题词与已下载网页 p 的文字内容来计算，用它们的相关度来估计 P 所指向网页的相关度。相关度大的网页，它所指向的网页的优先级就高，从而决定了待爬行队列中 URL 的优先级。如果待爬行队列的缓冲区满了，则将优先级最低的 URL 从该队列中移去。它采用如下公式来计算网页与主题之间的相关度。

$$sim(q,p)=\frac{\sum_{k\in q\wedge p} f_{kq}f_{kp}}{\sum_{k\in p} f_{kp}^2 \sum_{k\in q} f_{kq}^2} \text{ 或 } sim(q,p)=\sum_{k\in q\wedge p} f_{kq}f_{kp} \qquad (4\text{-}11)$$

其中q表示主题，p表示抓取的网页，f_{kq}表示词$1-\gamma$在q中出现的频次，f_{kp}表示词k在p中出现的频次。Best First Search 算法的伪代码如图 4-4 所示。

该算法维护两个队列：url_queue 和 crawled_queue，分别用来存放待爬行 URL 和已爬行 URL。程序运行时，先将初始 URL 放入 url_queue 中，通过公式 4-11 计算出优先级最高的 URL 进行爬行，并将该 URL 放入 crawled_queue。将爬行下来的网页通过锚文本和(或)链接信息计算相似度，并根据该相似度值将从该页面中解析出来的相应链接插入 url_queue。循环该操作，直到系统访问的网页数超过最大页面数或者 url_queue 中无 URL。

这种算法的优点是计算量比较小，对宽泛主题比较有用。但在关键字很多的情况下，由于锚文本和链接信息不能很好地反映页面主题，导致效果不佳。也容易使算法过早陷入 Web 搜索空间中局部最优子空间的陷阱，导致整体回报率不高。Best First Search 算法伪代码如下：

```
Best_First_Search(Starting_URLs,topic)
{
    enqueue(url_queue,Starting_URLs); //将起始站点 URL 全部压栈;
    int numVisited=0;
    WHILE(numVisited<MAX_PAGES AND number(ulr_queue)!=0){
        url=dequeue_top_link(url_queue);
        page=crawl_page(link);
        numVisited++;
        enqueue(crawled_queue,url);
        url_list=Extract_link(page);
        sim_score=sim(topic,page); //计算相似度
        enqueue(buffered_page,page,sim_scroe); //将结果保留
        FOR EACH u IN url_list{
            If(u not in url_queue and u not in crawled_queue)
```

```
                Enqueue(url_queue,sim_score);
            If(number(url_queue)>MAX_BUFFER)
                Dequeue_bottom_links(url_queue);
        }
        Reorder_queue(url_queue);
    }// END OF WHILE
}
```

4.6.2 Fish Search 算法

此算法是 De Bra 于 1994 年提出来的，他把一个 URL 比喻成一条鱼。鱼能活多久，它们能繁殖多少后代，取决于它们找到食物的多少。当一个文件被爬行下来之后，很多的新的 URL 就会被解析出来(鱼繁殖很多的后代)。其中，有用的 URL 的数量取决于该文件是否与主题相关(包含多少食物)，以及它本身包含链接(后代)的数量。每次，当鱼找到大量食物(主题相关页面)之后，它就能变得强壮，并繁殖更多的后代。反之，鱼就变得虚弱，后代也少。当鱼找不到食物(无相关页面)或者水被污染(带宽不够)的时候，鱼就死掉。该算法的关键是根据种子站点和查询关键字，动态地维护一个 ULR 爬行优先队列。当一个网页抓取过来后，抽取它所有的 URL，这些 URL 所对应的网页，称为“孩子”网页。如果抓取的网页相关，孩子网页的深度(depth)设成一个预先定义的值，否则孩子网页的深度设置成一个小于父亲网页深度的值。当这个深度为零的时候，这个方向的搜索就停止。

深度大于 0 的孩子网页的 URL 按照下述启发策略来插入到 url_queue 中：

相关网页的前面 a*width 个孩子(a 是预定义的大于 1 的常量)加入到 url_queue 的顶部；

无关网页的前 width 个孩子 URL 加入到 url_queue 队列中紧靠着相关网页的孩子节点后面；

剩下的孩子 URL 加入到 url_queue 的尾部(也就是说只有在时间允许的情况下，才有可能爬行它)。

上述三种情况，我们可以用一个变量 potential_score 来等价描述它们。第一

种 potential_score 设置为 1，第二种设置为 0.5，第三种设置为 0。待爬行 ULR 队列就按照 potential_score 来排序。Fish Search 算法的入口参数包含种子站点、查询式(主题)、查询宽度 Width、深度 D。

该算法是一种基于客户端的搜索算法，因为其模式简单、动态搜索，有一定的吸引力，但存在如下缺点：只使用简单的字符串匹配分配 potential_score 的值，并且该值是离散的(只有 1、0.5 和 0 三种)，分配的值不能完全代表与主题的相关度。在 url_queue 中，优先级值之间的差别太小。当很多 URL（节点)具有相同的优先级并且在爬行时间受到限制的时候，可能后面更重要的网页被忽略掉了。另外，使用 Width 参数来调节删除网页后面 URL 的个数也有点过于武断，可能导致丢掉很多重要资源。

4.6.3 Shark Search 算法

顾名思义，Shark Search[19, 20] 是在 Fish Search 的基础上改进而成的算法。鲨鱼 (Shark)是一种比一般的鱼(Fish)更凶猛的鱼，寓意该算法比 Fish Search 更有效。它在 Fish 基础上，作了如下改进：

1.在 Fish 算法中，只有是否相关的二值判断，而在 Shark 算法中引入了相似度度量方法，取值在 0 到 1 之间;

```
Fish_Search(Starting_URLs,topic,width,D)
{
    Enqueue(url_queue,Starting_URLs,D); //将种子站点入栈，深度为 D
    Int numVisited=0;
    WHILE(numVisited<MAX_PAGES AND number(ulr_queue)!=0){
        (url,depth)=dequeue_top_link(url_queue);
        page=crawl_page(url);
        numVisited++;
        enqueue(crawled_queue,url);
        url_list=Extract_link(page);
        sim_score=sim(topic,page); //计算相似度
        enqueue(buffered_page,page,sim_scroe); //将结果保留
```

```
IF(depth>0){
STEP1:IF(当前页面不相关){
        对 url_list 的前 width 个孩子节点（child_node）
        potential_score=0.5;
        对所有剩余的孩子节点，potential_score=0;
    }ELSE {
        对 url_list 的前 a*width 个孩子节点(a 等于预先设置的
        常量，一般为 1.5)
        Potential_score=1;
        对所有剩余的孩子节点，potential_score=0;
    }
STEP2: FOR EACH u IN url_list{
    If(u in url_queue){
        比较 url_queue 中的 score 和 u 的 score，用最大值取代
        url_queue 中的 score;
        如果有需要，按照 score 对 url_list 排序;
    }ELSE
        如果有需要，按照 potential_score 的大小在 url_list 寻找
        合适的位置插入;
    }
STEP3:FOR(EACH u IN url_list){
    计算深度 depth，depth（u），如下计算：
    If（当前页面相关）
        Depth(u)=D;
    ELSE
        Depth(u)=depth(page)-1;
    If(u in url_queue)
        比较 url_queue 中的深度和 depth(u),用最大值取代
        url_queue 中的 depth;
```

```
            }
        }//END OF IF(depth>0)
    }//END OF WHILE
}
```

2)在计算 URL 的 potential_score 上，不但继承了双亲的值，而且充分利用了锚文字和锚文字的上下文[21]（围绕在锚文字周围一定距离的文字）。下面给出了与 Fish 算法不同的地方，用来取代 Fish 算法中的 STEP1。

上面讨论的三种算法是主题爬虫广泛采用的基于内容评价的启发搜索策略算法。它们只注重文本在主题搜索的重要性，而忽略了 Web 结构的作用。而且它们也没有采用分类器，它们没有明显的学习和训练阶段，这样很可能会导致网页的误选。

用下面的部分，取代 Fish 算法的 STEP1。Shark 算法在计算 score 的时候，采用下述方法：

1、计算孩子节点(child_node)继承相关度值 Inherited_score(child-node)

IF(current_ node 相关)

Inherited_score(child-node)=δ *sim_score; //δ 为预先定义的衰减因子

ELSE

Inherited_score(child-node)=δ *Inheirtedcsorec(current_node);

2、anchor_text 为 child_node 的锚文字，anchor_text_context 为锚文字的上下文(前后出现的文字)

3、计算锚文字的相似度:anchor_score=sim(topic, anchor_text)；

4、计算 anchor_text_context 相关度的值：

IF(anchor_score >0)

anchor_text_context=1；

ELSE

anchor_context_score=sim(topic,anchor_text_context)；

5、计算 anchor 的值，用 neighborhood_score，方法如下：

neighborhood_score=β * anchor_score +(1−β)* anchor_context_score；

//β 是预先定义的常量

6、计算 potential_score(child_node) 如下：

potential_score(child_node)= γ * Inherited_score(child-node)+($1-\gamma$) neighborhood_score(child_node)

第五章 基于链接结构的搜索策略研究

基于领域主题的搜索引擎通过领域或主题的设定使得返回的搜索结果对用户查询要求的满足在针对性和准确性方面要好于传统的通用搜索引擎，但是传统的网页排序技术在基于领域主题的搜索引擎中仍然极为重要，因为用户在对于网上海量信息检索时看重的总是准确性，希望搜索引擎将最关心的网页排在最前面，通常绝大多数用户只会浏览搜索结果的前 3 页左右。

目前的网页排序算法主要还是基于链接分析算法[22-24]，主要有以下两种：

5.1 PageRank 算法

这是 Google 采用的网页排序算法。PageRank[25-30] 可以认为是一个用户行为的模型。假设有一个“随机的网上冲浪者”，他被随机给定一个 Web 页面并且持续点击上面的链接，从不点击“返回”，但最后变得无聊了开始点击另一个随机页面。随机网上冲浪者访问一个页面的概率就是它的 PageRank。这里定义衰减因子 d 为随机网上冲浪者变得无聊并且请求另一个随机页面的概率。PageRank 被定义如下：

假设页面 A 有页面 $(T_1,T_2...T_n)$ 指向它，参数 d 是一个衰减因子它可设在 0 到 1 之间，通常设置 d 为 0.85，同样 $C(A)$ 定义为链接到页面 A 外页面内容的数量。页面 A 的 PageRank 给出如下计算公式：

$$PR(A)=(1-d)+d(PR(T_1)/C(T_1)+...+PR(T_n)/C(T_n))$$

注意 PageRank 形成了 Web 页面上的一个概率分布，因此所有 Web 页面 PageRank 值的总和将会是 1。PageRank 或者 $PR(A)$ 可以通过使用一个简单的迭代算法计算出来。

PageRank 算法有很多扩展和改进，实际中 Google 除了主要用 PageRank 算法来衡量网页的重要程度外，还有其它上百种因素来参与排序，其它搜索引擎也是如此。

5.2 HITS 算法

Kleinberg 提出了一种更为完善的衡量网页重要程度的 HITS 算法[31, 32]，他认为网页的重要程度是与所查询的主题相关的，由此提出了权威网页(Authority)和中心网页(Hub)的概念。权威网页是给定主题底下的一系列重要的权威的网页，其内容本身对于这个给定主题来说是重要的，并被其他网页承认为权威的，与这个主题相关的很多网页都有链接指向这个网页。中心网页，是包含很多指向权威网页的超链接的网页。很多情况下,同一主题下的权威网页之间并不存在相互的链接(相互间并不"认可")。例如,"Microsoft"和"Netscape"虽然都是浏览器主题中的权威站点,但它们却并不存在相互的链接。然而,它们通常同时被一些不知名的中心网页所共同指向。通过权威网页和中心网页，链接结构可以描述为它们之间的一种依赖关系：一个好的中心网页应该指向很多好的权威网页,而一个好的权威网页则应该被很多好的中心网页所指向。如图 5-1 所示：

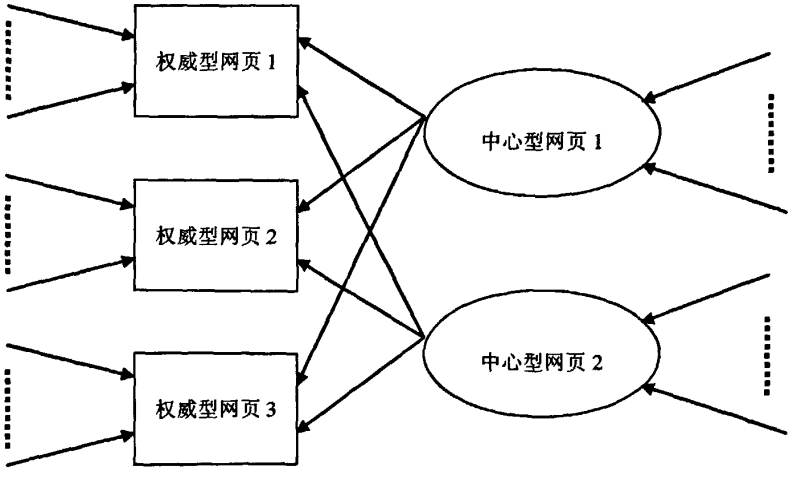


图 5-1 权威网页与中心网页

基于以上这种链接结构描述的概念，可以定义一种区分网页价值程度的度量。具体来说，首先利用一个传统的文本搜索引擎(例如 AltaVista)获取一个与主题相关的网页根集合(Root Set)。然后向根集合中扩充那些指向根集合中网页的网页和根集合中网页所指向的网页，这样就获得了一个更大的基础集合(Base

Set)。假设最终基础集合中包含N个网页,那么对于HITS算法来说，输入数据就是一个$N \times N$的相邻矩阵A,其中如果网页i存在一个链接到网页j,则$A_{ij}=1$,否则$A_{ij}=0$。

HITS 算法为每个网页 i 分配两个度量值：中心度h_i和权威度a_i。设向量$a=(a_1,a_2,...,a_N)$代表所有基础集合中网页的权威度，而向量$h=(h_1,h_2,...,h_N)$则代表所有的中心度。最初，将这两个向量均置为$u=(1,1,...,1)$。操作$In(a)$使向量$a=A^T h$，而操作$Out(h)$使向量$h=Aa$。反复迭代上述两个操作，每次迭代后对向量a和h范化，以保证其数值不会使计算溢出。Kleinberg证明经过足够的迭代次数，向量a和h将分别收敛于矩阵$A^T A$和AA^T的主特征向量。通过以上过程可以看出，基础集合中网页的中心度和权威度从根本上是由基础集合中的链接关系所决定的，更具体地说，是由矩阵$A^T A$和AA^T所决定。

PageRank 算法和 HITS 算法比较

在文献中指出，PageRank 算法实质上是一种通过离线对整个互联网结构图进行幂迭代的方法。PageRank 所计算出的网页等级值实际上就是互联网结构图经过修改后的相邻矩阵的特征值，对这些值的计算有非常有效的方法(事实上，仅需要若干次的迭代计算即可以得到)，因此能够很好地应用到整个互联网规模的实践中。这种方法的另一个主要优点是所有的处理过程都是离线进行的,因此不会为在线的查询过程付出额外的代价。但是，PageRank 算法也同样存在一个显著的问题，即网页等级值的计算是不是针对查询的，对于某个特定主题的查询，在返回结果中一些与主题无关的“强壮”网页将会排在较前的位置。

HITS 算法在概念的定义上比 PageRank 算法多提出了一个中心网页的概念。通过中心网页和权威网页的相互作用，HITS 算法更好地描述了互联网的一种重要组织特点:权威网页之间通常是通过中心网页而彼此发生关联的。HITS 算法和 PageRank 相似，也是通过迭代的方法计算相邻矩阵的特征向量。但 HITS 算法所针对的不是整个互联网结构图，而是特定查询主题的互联网子图。规模上的极大减小可以使 HITS 算法的迭代收敛速度比 Page Rank 要快得多，但因为与查询相

关，所以查询过程需要考虑排序的代价。

5.3 基于主题的 HITS 算法

Topic_HITS 算法核心思想

传统的 HITS 排序算法使用了单一的 Authority 值来衡量一个网页的权威，并没有考虑到这个权威是来自哪个主题的，导致的结果是，一个网页在一个主题上是权威的，但是对于另外的主题并不一定是权威的。例如关于足球的新闻网页在体育这个主题上是权威的，但是在其他主题上就未必权威。如果用传统的 HITS 算法就很难区分和决定是哪个主题使得这个网页的权威值很大。为了解决这个问题，针对每一个网页事先计算出一个权威向量，用这个向量中的每一项来区分不同的主题的贡献，本文把这种思想融合进了 HITS 算法，提出了 Topic-HITS。在多个主题上进行了试验，都表明了此算法优于其他的排序算法。

打个比方，一个人在音乐方面很出色，我们不能说他在美术方面同样出色，这个道理对一个网页来说同样成立。本文提出了在计算 A 网页的权值时，把所有链入 A 网页的网页按主题进行分类，这样不同权威值构成一个权威值向量。

这种方法还有许多潜在的应用。第一，这种对链入页面主题的分类避免了高权威值对于不相关查询的误导，可使得搜索更加准确。另外可以应用在分类上，对于一个网页，在权威向量中找出最大的一项，此最大项所对应的主题类别即为该网页所属的类别，这样提供了一种自动网页分类方法，比 ODP（Open Directory Project）和 Yahoo Directory 更加灵活，经济。

本文，提出了一种新的方法，把主题特征加入到 HITS 算法中，网页的链接结构从主题这个更细化的粒度进行链接分析。为了便于下文的介绍，我们引入了下面的概念：

$I(v)$：网页 v 的入度；

$O(v)$：网页 v 的出度；

$A(v)$：v 的权威值

$H(v)$：v 的中心值；

W：网页集合；

N：W 中的网页数；

d：随即跳转概率；

$p \to q$：网页 p 指向网页 q 的超链接。

给每一个网页分配三个向量：内容向量；权威值向量；中心值向量。内容向量 $C_u[C(u_1),C(u_2),...C(u_T)]$ 代表了页面 u 的内容，每一维代表了一个主题，这个向量是静态的，可以由网页的内容来决定，用文本分类器提供输入，内容向量被标准化为各项和为 1。

另外，我们给每一个网页 u 分配了权威值向量 $A_u[A(u_1),A(u_2),...A(u_T)]$ 来衡量他的权威性，$A(u_k)$ 标记了网页在主题 k 上的权威值；还分配了一个中心值向量 $H_u[H(u_1),H(u_2),...H(u_T)]$，$A_u$ 和 H_u 的计算方法在后面会有详细介绍，对于一个特定的查询 q，排序的结果可以这样计算 $S_q(u)=\sum_k A(u_k)\times C(q_k)$。

网页之间的跳转有以下两种情况：

1. 与权威值计算相关：

Fs：网页 v 有到 u 的超链接，且主题保持不变的概率是 α；

F_J:网页 v 有到 u 的超链接，且主题以概率 $1-\alpha$ 转换；

于是有下列公式：

$P(Fs \mid v_k)=\alpha$；

$P(F_J \mid v_k)=1-\alpha$；

$$P(u_i \mid v_k, Fs)=\frac{1}{O(v)}$$

$$P(u_i \mid v_k, F_J)=\frac{C(v_i)}{O(v)}$$

2. 与中心值计算相关：

Bs：网页 u 是从指向他的链接直接链接而来，且主题保持不变的概率是 α；

B_J：网页 u 是从指向他的链接直接链接而来，且主题以概率 $1-\alpha$ 转换

于是有下列公式：

$$P(Bs|u_k)=\alpha$$

$$P(B_J|u_k)=1-\alpha$$

$$P(u_i|v_k,Bs)=\frac{1}{I(v)}$$

$$P(u_i|v_k,B_J)=\frac{C(v_i)}{I(v)}$$

一个网页的主题中心值由两部分组成，一部分是该网页以概率α保持主题不变向后链接，一部分是该网页以概率$1-\alpha$转换主题向后链接。这两部分的和包括了该网页包含的所有超链接。

于是有下面的主题中心值计算公式：

$$H(v_i)=\sum_{u:v\to u}P(v_i|u_i,Bs)P(Bs|u_i)A(u_i)+\sum_{u:v\to u}\sum_{k\in T}P(v_i|u_i,B_J)P(B_J|u_i)A(u_k)$$
$$=\sum_{u:v\to u}\frac{\alpha A(u_i)+(1-\alpha)C(u_i)A(u)}{I(u)} \quad (5\text{-}1)$$

其中，$A(u)=\sum_{k\in T}A(u_k)$

一个网页的主题权威值也是由两部分组成，一部分是该网页的前向网页（即链接向该网页的网页）是以概率α保持主题不变指向他，另一部分是该网页的前向网页以概率$1-\alpha$转换主题指向他。

于是有下面的主题权威值计算公式：

$$A(u_i)=\sum_{v:v\to u}P(u_i|v_i,Fs)P(Fs|v_i)H(v_i)+\sum_{v:v\to u}\sum_{k\in T}P(v_i|u_i,F_J)P(F_J|u_i)H(v_k)$$
$$=\sum_{v:v\to u}\frac{\alpha H(v_i)+(1-\alpha)C(v_i)H(v)}{O(v)} \quad (5\text{-}2)$$

其中，$H(v)=\sum_{k\in T}H(v_k)$

进一步讨论

在这个模型中，α是主题不变的概率，当当前页面的内容符合用户的希望的主题时，主题转化的概率较小，反之，较大，因此，可以用当前页面的内容向量值$C(v_k)$来代替α。

另一方面，这个模型我们考虑的是网页到网页之间的链接，事实上，同一个网站M的多个网页指向另外一个网页n，说明这个网站M对网页n是信任的，于是让这个网站M上的所有网页m平分这个信任度，这种方法貌似更合理一些，

我们后面的试验也验证了这种方法的合理性。这样一来，公式(5-2)就变成了下面的这个公式：

$$A(u_i)=\sum_{N:M_j\in N}\frac{\sum_{v:v\in M_j\wedge v\to u}P(u_i\mid v_i,Fs)P(Fs\mid v_i)H(v_i)+\sum_{v:v\in M_j\wedge v\to u}\sum_{k\in T}P(v_i\mid u_i,F_J)P(F_J\mid u_i)H(v_k)}{m_j}$$

$$=\sum_{N:M_j\in N}\sum_{v:v\in M_j\wedge v\to u}\frac{\alpha H(v_i)+(1-\alpha)C(v_i)H(v)}{O(v)m_j}$$

其中，$H(v)=\sum_{k\in T}H(v_k)$，N 表示指向网页 u 的网站的集合，M_j 表示指向 u 的网页 v 所在的网站，m_j 表示指向 u 并且与 v 所在的网站相同的网页个数。

同理，如果一个网页 n 指向另外一个网站 M 的所有网页，说明这个网页 n 对这个网站 M 有很好的引用，基于平分的理念，公式(5-1)就变成了下面的这个公式：

$$H(v_i)=\sum_{N:M_j\in N}\frac{\sum_{u:u\in M_j\wedge v\to u}P(v_i\mid u_i,Bs)P(Bs\mid u_i)A(u_i)+\sum_{u:v\to u}\sum_{k\in T}P(v_i\mid u_i,B_J)P(B_J\mid u_i)A(u_k)}{m_j}$$

$$=\sum_{N:M_j\in N}\sum_{u:u\in M_j\wedge v\to u}\frac{\alpha A(u_i)+(1-\alpha)C(u_i)A(u)}{I(u)m_j}$$

其中，$A(u)=\sum_{k\in T}A(u_k)$，　N 表示 v 指向的网站的集合，M_j 表示 u 所在的网站集合，m_j 表示被 v 指向（即前向链接为 v）并且与 u 所在的网站相同的网页个数。

第六章 基于综合价值的主题爬行策略研究

第四、五章重点研究了目前常用的主题爬虫爬行策略:基于内容评价的搜索策略和基于 Web 链接结构的搜索策略。基于内容评价的搜索策略的优点是具有较好的理论基础且计算简单。但由于这类方法忽略了链接结构信息，因而在预测链接价值的重要性方面存在一些不足。以 PgaeRank 和 HITS 为代表的基于 Web 链接结构的搜索策略，通过分析 Web 页面之间的相互引用关系来确定网页的重要性，进而决定链接访问顺序。该方法虽然考虑了链接结构和页面之间的引用关系，但忽略了页面与主题的相关性，在某些情况下，HITS 会出现搜索偏离主题的“主题漂移” 问题；而 PageRank 算法只适合于发现权威网页，不适合于发现主题资源。

考虑到采用单一的评价方法不能有效预测链接 URL 的真实价值，为了提高链接价值预侧的准确性，将两类评价标准结合，根据它们各自的优缺点，在不同搜索阶段给予其不同的信任值，以提高整体搜索效率。

6.1 基于综合价值搜索方案

为了充分利用各种文字内容和超链信息，本文提出一种基于综合价值的主题爬虫爬行搜索策略:将基于内容评价的搜索策略和基于 Web 链接结构的搜索策略相结合，来决定主题爬虫待爬行队列中的 URL 优先级别。

怎样将基于内容评价的搜索策略和基于 Web 链接结构的搜索策略相结合？

本文认为下面有几种方案：

1.首先基于 Web 链接结构搜索策略过滤 URL，然后基于内容评价搜索策略来决定过滤后的 URL 优先级别。

此策略先从分析网页超链结构的角度入手，先通过链接评价搜索策略来过滤掉不重要的 URL，然后根据过滤后的 URL 跟主题内容相似度大小决定爬虫爬行顺序。

2.首先基于内容评价的搜索策略过滤，再基于 Web 链接结构搜索策略排序。

此策略先从分析主题相似度的角度入手，先通过内容评价搜索策略来发现与

主题相关的URL(即过滤掉不相关的URL)，然后根据ULR预测质量高低决定爬行顺序。

3.基于Web链接结构搜索策略和基于内容评价搜索策略的数学模型组合。

此策略同时考虑网页质量和主题内容，采用Web链接结构和内容评价的综合值过滤URL，并决定爬行顺序。

4.首先其于Web链接结构搜索策略过滤，再采用基于Web链接结构的搜索策略和基于内容评价的搜索策略的数学模型组合值来决定URL优先级别。

此策略先通过Web链接结构的搜索策略发现质量高的URL，然后采用策略3来决定待爬行队列的URL的顺序。

5．首先基于内容评价的搜索策略过滤，再采用基于Web链接结构的搜索策略和基于内容评价的搜索策略的数学模型组合值来决定URL优先级别。

此策略先通过内容评价过滤产生与主题相关的URL队列，然后在此队列中再采用策略3来决定待爬行队列的URL的顺序。

6.首先基于Web链接结构搜索策略和内容评价搜索策略的数学模型组合过滤URL，再采用基于内容评价搜索策略决定URL优先级别。

此策略先通过采用策略3来选择URL，然后在此队列中再以内容相似度大小来决定待爬行队列的URL的顺序。

7.首先基于Web链接结构搜索策略和内容评价搜索策略的数学模型组合过滤URL，再采用基于Web链接结构的搜索策略排序。

此策略先通过采用策略3来选择URL，然后在此队列中再以网页质量高低来决定待爬行队列的URL的顺序。

以上7种基于内容评价搜索策略和基于Web链接结构搜索策略相结合方案，每一种都有自己的优点，像第一、第四种方法都能较快的搜索质量较高的网页，但不能保证网页的主题;第三种方法是取网页质量和网页相似度的平均值或有偏重，但过于简单:第五种在自动主题搜索引擎中，它可先对内容进行过滤，保证了网页的主题相关的前提下，再采用Web链接结构的搜索策略和内容评价的搜索策略的数学模型组合来决定待搜索的URL的爬行顺序，此方案具有很好的搜索效率。本系统主要采用第五种策略将基于内容评价和基于Web链接结构评价搜索策略相结合。

本研究基于综合价值主题爬行搜索策略基本思想为：在爬行初期，由于主题搜索引擎的爬虫是以站内搜索为主，PageRank 和 HITS 还不能发挥其强的排序作用，故本综合策略在爬行初期，只采用基于内容评价策略过滤并决定 URL 的优级别。在后来 Crawler 改为网络链接抓取时，采用策略 5——首先基于内容评价的搜索策略过滤，再采用基于 Web 链接结构的搜索策略和基于内容评价的搜索策略的数学模型组合值来决定 URL 优先级别。即在初期(探测阶段)采用基于内容评价策略，以后(发掘阶段)采用策略 5。这样既可利用基于内容相似度的评价来提高搜索内容与主题的相关度，同时又可利用基于链接结构的评价来提高主题资源搜索的覆盖率。

6.2 基于综合价值的主题爬虫搜索策略实现

基于综合价值搜索策略主题爬虫模型

本文提出了一种新的网络爬虫搜索模型。相对于传统的网络爬虫模型，新模型先基于内容评价过滤 URL，再采用 Web 链接结构的搜索策略和内容评价的搜索策略的综合价值来控制优先权队列(爬行初期除外)，决定链接的访问顺序。

主题爬虫对 Web 的搜索是一个循环迭代的过程：Crawler 首先从一个“种子集”(如用户查询、种子链接或种子页面)出发，通过 HTTP 协议请求并下载 Web 页;预处理器负责解析 Web 页面，提取链接文本、结构信息和链接的 URL；链接价值计算器按照内容评价的值(如链接文本与预先定义的主题集的相似度)算出链接的价值；暂时未访问的链接被暂时存放在一优先权队列中，链接优先权控制器按照某种策略(在初期采用基于内容评价策略，在发掘阶段按策略 5)决定下一步将要访问的链接；当 Crawler 获得新选定的链接时，以上过程重复进行。

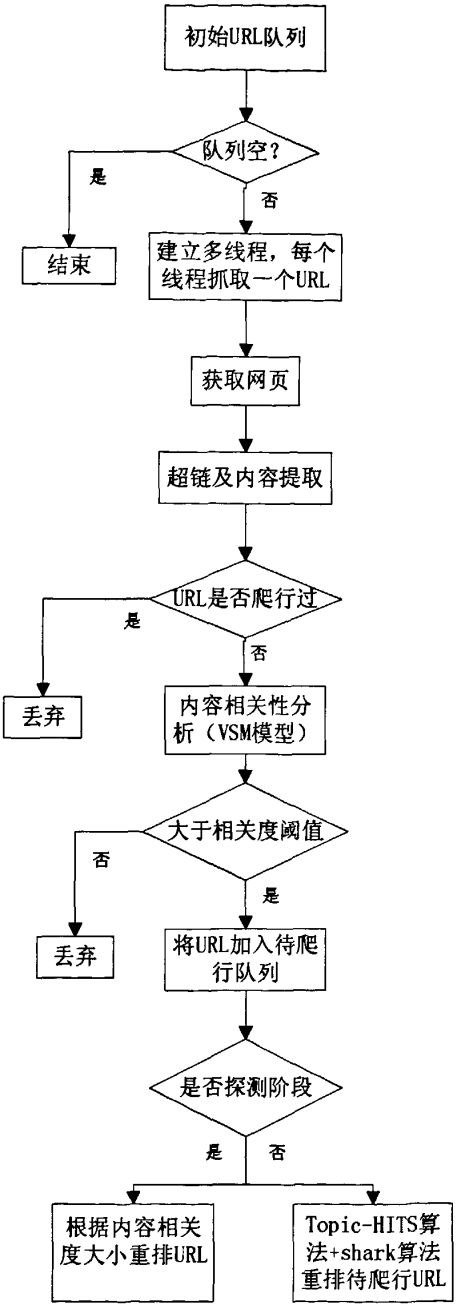


图 6-1 基于综合价值的搜索策略流程图

链接价值的计算

主题向量库可采用已有的在线专业字典(如英国帝国大学研制的 FOLDOC 在线计算机字典或自建，用第四章中的公式 4-10 计算链接文本与专业词库的相似

度，作为内容过滤。综合价值的计算采用公式：

$$D=\theta\times potential\_score(child\_node)+(1-\theta)\times Topic\_HITS(u) \qquad (6\text{-}1)$$

式中：$Topic_HITS(u)$表示网页的$Topic_HITS$值，θ调节参数，取值范围为0-1，$potential_score(child_node)$为子节点与主题预测相关度值。

6.3 试验分析

实验环境建立：

运行环境参数：

CPU：Intel 双核 1．8GMHz

内存：512M

硬盘：80G

操作系统：WindowsXP—Professional sp2

编程语言：Java

软件安装：

jdk5.0 : http://developers.sun.com/downloads/

nutch0.7.2 : http://www.apache.org/dyn/closer.cgi/lucene/nutch/

Cygwin2.29 : http://www.cygwin.com/mirrors.html

tomcat5.5 : http://tomcat.apache.org/

Ant1.7.0 :http://ant.apache.org/bindownload.cgi

本章主要验证本文提出的主题表示方法的优越性，以及 Topic-HITS 算法在主题爬行过程中能够搜索到更准确的网页，并且验证了爬行的时间复杂度大大的降低。系统主要可分为 HTML 预处理模块，主题相关性评分器和待爬行 URL 的优先级计算等模块，其中 HTML 预处理模块由 LUCENE 来索引实现。

整个系统在 nutch 抓取网页的基础上，由 java 编写 URL 优先级算法程序和主题相关性评分程序，在 Windows 平台上实现。

种子页面的选取

由于目前的知名网站的网页设计已基本上趋于模板化，而且主题比较明显突出，结构比较固定。因此仅仅针对知名网站的网页进行实验不足以具有说服力，

不能显示出算法的通用性。因此，为了保证实验具有通用的效果，我们的实验不仅选取了各门户网站的不同类别的网页进行实验，而且针对一些不知名的网站内容的网页进行实验，实验测试对象涉及新闻、健康，教育和医药等多领域，网页结构差别大，有助于验证算法性能。

此外，为了满足方法假设的前提，我们尽量挑选含有正文信息的网页。对于有些网页根本就没有正文信息，如一些大网站的索引，我们通过判定不把这样的网页作为处理范围之内。

这里我们先用关键字：工作、面试、笔试、技巧等等在通用搜索引擎 google 和百度进行搜索，选择排名前 20 位的网页，再结合身边同学推荐的一些网站进行筛选得到了需要的初始化种子站点。

我们把得到的这些初始化种子站点保存在一个 urls.txt 文件中。

分词

lucene 作为一个开放源码的搜索软件包应用越来越广泛，但是对于中文用户来说其提供的两个中文分词器(CJKAnalzyer、ChineseAnalyer)的功能很弱，为了提高搜索结果的准确性，于是采用中科院计算所 ICTCLAS 分词软件，由于网页中有部分超常的英文字符以及一些乱码，分词系统会报错并且停止 。对这些异常文件采用北大天网提供的基于词典的分词程序。其次，要进行词干化处理（Stemming）；再次，要去除中英文停用词，其中，中文停用词采用哈工大的中文停词表，共 494 个词，英文停用词采用 Rainbow 的禁止词，共 524 个。

网页抓取步骤

在 Nutch 中，Crawler 操作的实现是通过一系列子操作的实现来完成的。这些子操作 Nutch 都提供了子命令行可以单独进行调用。下面就是这些子操作的功能描述以及命令行，命令行在括号中。

1. 创建一个新的 WebDb (admin db -create)
2. 将抓取起始 URLs 写入 WebDB 中 (inject)
3. 根据 WebDB 生成 fetchlist 并写入相应的 segment(generate)
4. 根据 fetchlist 中的 URL 抓取网页 (fetch)
5. 根据抓取网页更新 WebDb (updatedb)
6. 循环进行 3－5 步直至预先设定的抓取深度

7. 根据 WebDB 得到的网页评分和 links 更新 segments (updatesegs)

8. 对所抓取的网页进行索引(index)

9. 在索引中丢弃有重复内容的网页和重复的 URLs (dedup)

10. 将 segments 中的索引进行合并生成用于检索的最终 index(merge).

面向主题的搜索在进行到第四步的时候，本系统对 fetchlist 中的 URL 进行排序，使与主题相关度大的网页排在前面，利用存储在 WebDB 中的页面之间的链接信息按照前面所述的 Topic-HITS 算法迭代计算出每一个页面对应的权威向量，利用综合价值评价的公式 6-1 计算出待爬行的 URL 的优先级顺序，Nutch 按照这个优先级顺序继续抓取下去，每抓到一个网页，本系统利用向量空间模型分析了该网页的与主题的相关度，这种再次过滤不但减少了爬虫所抓取网页的数量，提高了速度，而且提高了搜索的准确度。这样一直进行下去，直到所抓取的网页数达到 2000 或者已经达到了所要抓取的深度 3，为了作出比较，本系统同时还利用了 PageRank 和 HITS 算法进行 URL 优先级排序。

试验评测指标

1. MAP(Mean Average Precision)

单个主题的平均准确率是每篇相关文档检索出后的准确率的平均值。主题集合的平均准确率(MAP)是每个主题的平均准确率的平均值。MAP 是反映系统在全部相关文档上性能的单值指标。系统检索出来的相关文档越靠前(Rank 越高)，MAP 就可能越高。如果系统没有返回相关文档，则准确率默认为 0。

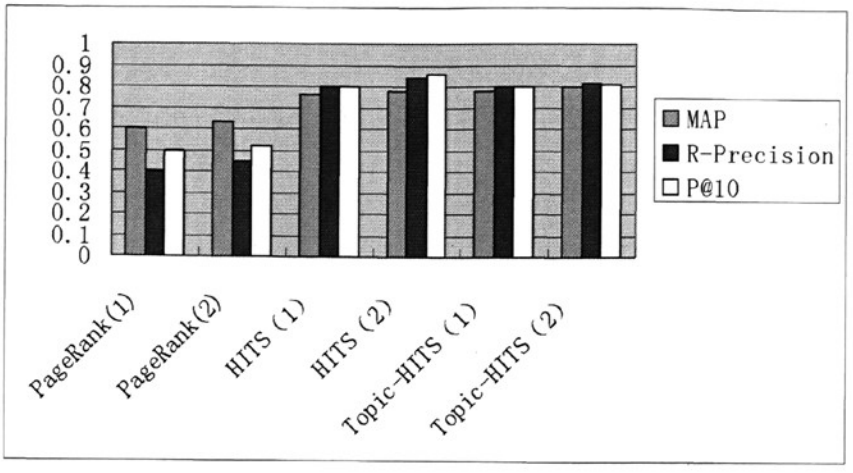
2. R-Precision

单个主题的 R-Precision 是检索出 R 篇文档时的准确率。其中，R 是测试集中与主题相关的文档的数目。主题集合的 R-Precision 是每个主题的 R-Precision 的平均值。

3. P@10

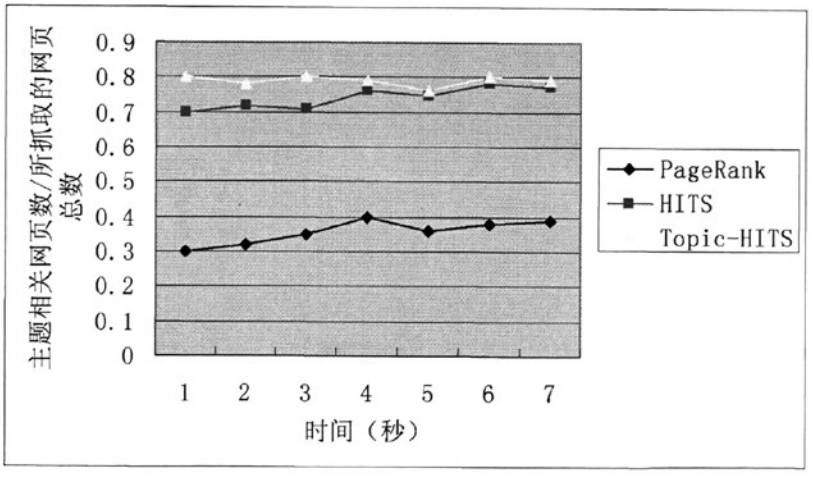
单个主题的 P@10 是系统对于该主题返回的前 10 个结果的准确率。主题集合的 P@10 是每个主题的 P@10 的平均值。

本系统验证了在两种主题表示方法下，在 MAP,R-Presion 和 P@10 上的表现，实验结果如下：



（1）表示只使用文档集表示主题，（2）表示使用本文提出的 ODP 和示例型文档集相结合的方法表示主题，显然在使用了本文所提出的主题表示方法后，在 MAP，R-Precision，P@10 方面都有改善。实验结果说明了 ODP 算法和示例型文档集相结合的算法更好的表达了主题。

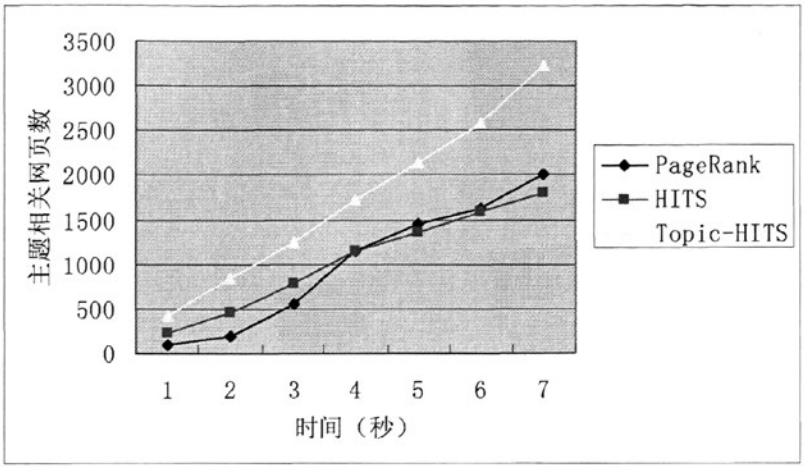
在本文提出的主题表示模型下，记录了随着时间的变化，主题相关网页数与所抓取的网页总数的比值，试验结果如下：



PageRank 算法主题相关网页数在所抓取的网页总数中所占的比重较小，也就是说，爬虫抓取了很多无用的网页，占用的带宽较大，效率较低。其原因是 PageRank 在抓取过程中没有进行链接过滤，使很多无用链接掺杂在其中。Topic-HITS 与 HITS 算法表现基本相同，他们都考虑了利用主题对链接过滤，考

虑了待爬行队列的 URL 优先级排序，只是两种所使用的排序方法不同，实验表明，Topic-HITS 算法表现较好，也就是他所使用的待爬行队列 URL 优先级排序算法较好，这正是本文所改进的地方。

此外，试验还验证了随着时间的变化，三种排序算法所抓取的主题相关网页数的变化。如下图所示：



实验表明，在相同的时间内，Topic-HITS 算法所抓取的主题相关网页数是最多的，而且随着时间的变化，Topic-HITS 算法的倾斜度最大，增加的最快。原因是，在 Topic-HITS 算法中针对每一个网页保存了一个权威值向量，这样就不像 HITS 算法那样针对主题去在线的计算网页的权威值。所以在相同的时间内 Topic-HITS 明显好于 HITS。而在开始的 4 秒钟，HITS 是好于 PageRank 的，其原因是开始时，链接结构并不复杂，HITS 在线计算的速度也较快，加上他对链接的过滤，所以要好于 PageRank,到了 4s 钟之后，由于链接结构变得比较复杂，因此，在线计算所花费的时间较多，在这段时间内，PageRank 优于 HITS。

第七章 总结与展望

7.1 总结

主题搜索引擎是近年来研究的热点，它对网络爬虫的性能和效率提出了较通用搜索引擎更高的要求。采用何种策略访问 Web 是决定网络爬虫搜索性能的关键，也是目前较为重要的研究课题。本文围绕着提高网络爬虫的搜索精确度和搜索效率的问题，对网络爬虫的搜索策略进行了深入研究，主要成果可归纳为以下几个方面：

1. 对现有的网络爬虫经典搜索策略进行了比较深入地研究。
2. 对著名的 HITS 算法进行了深入研究，指出 HITS 算法的优缺点，并对 HITS 算法进行了改进，改进后的 HITS 算法搜索网页的能力将大大加强，降低了算法的空间复杂度和时间复杂度，提高了爬行算法的效率。
3. 为了提高网络蜘蛛在搜索策略调整上的自适应性并解决传统的网络爬虫中存在链接价值评价标准单一的问题，本文提出一种基于综合价值的搜索策略，此策略根据不同的搜索阶段采用不同的综合策略。
4. 最后，本文将改进后的算法和技术相结合，实现了一个关于主题的网络爬虫系统原型，该原型可用于算法性能的比较以及实际的 Web 信息采集。

7.2 展望

随着主题爬虫在信息采集和数据挖掘方面的重要性日益突出，人们也越来越重视对它的研究与应用。本文虽然在主题爬虫算法方面做了诸多有益尝试，但后面还有很多值得进一步研究的地方：

1. 由于时间和条件的限制，我们对主题页面的分布规律只停留于前人的科研结论，并未进行更深入的研究。因此，在主题页面的分布规律方面应该还有很多值得讨论和研究的空间。
2. 页面过滤时所要设置的阈值的好坏可能直接影响到系统爬行到的网页的质量，这个最优值的确定需要大量的实验数据支持，目前做得还不够。

3. 在本研究中所用算法，一些必要的参数的设定往往需要根据实际运行来调整，目前还没有找到最理想的值，这些工作需要在今后的实践中不断完善。
4. 无论是基于主题的爬虫还是普通的爬虫，对动态网页的获取仍然是一个难点和重点，有待于进一步深入研究。
5. 人工智能与爬虫的结合，人工智能相关技术包括机器学习、神经网络、模式识别、专家系统等，一直是现在研究的热点，它的目的就是教会计算机自己去识别有用信息，发现信息，这也是以后需要研究的内容。

参考文献

[1] 聂哲. 基于 WEB 的面向主题搜索引擎的设计与实现[J]. 计算机工程与设计. 2003(02).

[2] 陈财森, 王韬, 郑伟, et al. 基于搜索引擎调用的主题搜索设计与实现[J]. 计算机工程与设计. 2008(21).

[3] 李勇, 韩亮. 主题搜索引擎中网络爬虫的搜索策略研究[J]. 计算机工程与科学. 2008(03).

[4] 王津涛, 兰皓. 面向主题元搜索引擎的设计与实现[J]. 计算机工程. 2005(07).

[5] Brin S, Page L. The anatomy of a large-scale hypertextual Web search engine[J]. COMPUTER NETWORKS AND ISDN SYSTEMS. 1998, 30(1-7): 107-117.

[6] Zhongming M A, Pant G, Sheng O. Interest-based personalized search[J]. ACM TRANSACTIONS ON INFORMATION SYSTEMS. 2007, 25(1).

[7] 于海龙, 刘丽萍, 邬伦, et al. 基于 RM-ODP 的模型复用框架 OMRF[J]. 计算机应用研究. 2008(03).

[8] 吴明礼, 施水才. 一种结合超链接分析的搜索引擎排序方法[J]. 计算机工程. 2004(15).

[9] 高飞, 刘云. Nutch 中文分词方法的实现[J]. 网络安全技术与应用. 2008(09).

[10] 吴涛, 张毛迪, 陈传波. 一种改进的统计与后串最大匹配的中文分词算法研究[J]. 计算机工程与科学. 2008(08).

[11] 王硕, 尤枫, 山岚, et al. 一种适用于专业搜索引擎的中文分词系统研究[J]. 计算机工程与应用. 2008(19).

[12] White R W, Ruthven I, Jose J M, et al. Evaluating implicit feedback models using searcher simulations[J]. ACM TRANSACTIONS ON INFORMATION SYSTEMS. 2005, 23(3): 325-361.

[13] Beitzel S M, Jensen E C, Lewis D D, et al. Automatic classification of Web queries using very large unlabeled query logs[J]. ACM TRANSACTIONS ON INFORMATION SYSTEMS. 2007, 25(2).

[14] 朱小娟, 陈特放. 基于 SVM 的词频统计中文分词研究[J]. 微计算机信息. 2007(30).

[15] 连浩, 刘悦, 许洪波, et al. 改进的基于布尔模型的网页查重算法[J]. 计算机应用研究. 2007(02).

[16] Wu H C, Luk R, Wong K F, et al. Interpreting TF-IDF term weights as making relevance decisions[J]. ACM TRANSACTIONS ON INFORMATION SYSTEMS. 2008, 26(3).

[17] 郭庆琳, 李艳梅, 唐琦. 基于 VSM 的文本相似度计算的研究[J]. 计算机应用研究. 2008(11).

[18] 潘国清. VSM 中用语片为特征项计算文本相似度[J]. 计算机与数字工程. 2007(10).

[19] 苏祺, 项锟, 孙斌. 基于链接聚类的 Shark-Search 算法[J]. 山东大学学报(理学版). 2006(03).

[20] 陈军, 陈竹敏. 基于网页分块的 Shark-Search 算法[J]. 山东大学学报(理学版). 2007(09).

[21] Melucci M. A basis for information retrieval in context[J]. ACM TRANSACTIONS ON INFORMATION SYSTEMS. 2008, 26(3).

[22] 汪涛, 樊孝忠. 链接分析对主题爬虫的改进[J]. 计算机应用. 2004(S2).

[23] 原福永, 张园园. 基于链接分析的相关排序方法的研究和改进[J]. 计算机工程与设计. 2007(07).

[24] 郑煜, 钱榕. 一个基于链接分析的相关度排序算法及其在专题搜索引擎中应用[J]. 计算机应用与软件. 2007(07).

[25] 钱功伟, 倪林, 曹荣. 基于网页链接和内容分析的改进 PageRank 算法[J]. 计算机工程与应用. 2007(21).

[26] 常庆, 周明全, 耿国华. 基于 PageRank 和 HITS 的 Web 搜索[J]. 计算机技术与发展. 2008(07).

[27] 王冬, 雷景生, 李壮. 基于 PageRank 的页面排序改进算法[J]. 计算机工程与设计. 2008(22).

[28] 钱杰, 张健, 高乐. Web 结构挖掘中的 PageRank 算法改进[J]. 计算机系统应用. 2008(07).

[29] 蔡建超, 蔡明. 搜索引擎 PageRank 算法研究[J]. 计算机应用与软件. 2008(09).

[30] Lamberti F, Sanna A, Demartini C. A Relation-Based Page Rank Algorithm for Semantic Web Search Engines[J]. IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING. 2009, 21(1): 123-136.

[31] 李昕, 朱永胜, 武港山. Web 结构分析算法 HITS 的改进及应用[J]. 计算机工程. 2005(06).

[32] 黄英铭. Web 结构挖掘及 HITS 算法分析[J]. 计算机与现代化. 2007(07).

致谢

首先要衷心的感谢我最尊敬的导师石冰教授！感谢导师在我攻读硕士学位期间无微不至的关怀和孜孜不倦的教诲。导师不仅在学业上严格要求，在为人处事、待人接物方面也言传身教，使我受益匪浅。三年来，导师渊博的知识、严谨的治学态度、活跃的学术思想、高尚的为师品德无时无刻不在影响着我，为我将来的工作和学习树立了榜样，在此，对石冰老师表示我最诚挚的谢意！

同时，还要感谢教研室的各位老师和同学。感谢他们三年中在学习、生活和实践中对我的大力帮助。感谢乔磊，杨兰仓等师兄们，感谢课题组成员刘兴涛、谢英文、闫宗奎等同学们，他们对我的工作给予了很大的支持和帮助，在我遇到疑问时给出耐心的解答，在研究中给予我具体细致的建议，并在课题的进展中热心地与我讨论，是他们的支持和帮助才使我顺利地完成学习和科研任务。

最后，我要深深地感谢支持、关心和鼓励我完成硕士学业的家人和朋友，他们的关爱和关怀是我克服困难、不断前进、并最终完成学业的精神支柱。

攻读学位期间发表的论文目录

1. 面向主题的查询相关网页排序算法，第三届研究生学术交流会—通信与信息技术会议论文集　第一作者
2. A Spiral-Decoding Method for Web Data Extraction，International Symposium on Education and Computer Science (ECS2009)　第三作者

主题爬虫搜索策略研究

作者：陈丛丛
学位授予单位：山东大学

本文读者也读过(9条)

1. 朱良峰 主题网络爬虫的研究与设计[学位论文]2008
2. 刘玮玮 搜索引擎中主题爬虫的研究与实现[学位论文]2006
3. 吴安清 主题搜索引擎爬行策略的研究[学位论文]2006
4. 张航 主题爬虫的实现及其关键技术研究[学位论文]2010
5. 刘玮 基于启发式搜索策略的主题网络爬虫算法的设计与实现[学位论文]2008
6. 刘喜亮 面向主题的网络爬虫设计与实现[学位论文]2009
7. 袁浩 主题爬虫搜索Web页面策略的研究[学位论文]2009
8. 杜一平 主题搜索网络爬虫的设计与研究[学位论文]2009
9. 李勇.韩亮.LI Yong.HAN Liang 主题搜索引擎中网络爬虫的搜索策略研究[期刊论文]-计算机工程与科学 2008,30(3)

引证文献(3条)

1. 姚瑞虹.张鹏洲.陈志国 互联网音视频主动搜索算法效率提高的研究[期刊论文]-广播与电视技术 2013(10)
2. 于成龙.于洪波 网络爬虫技术研究[期刊论文]-东莞理工学院学报 2011(3)
3. 张春菊.张雪英.朱少楠.徐希涛 基于网络爬虫的地名数据库维护方法[期刊论文]-地球信息科学学报 2011(4)