四川大学

硕 士 学 位 论 文

题 目 基于链接和页面内容的主题爬虫算法的研究与应用

作 者 2012223040119 完成日期 2015 年 4 月 7 日

培 养 单 位 四 川 大 学

指 导 教 师

专 业 计算机科学与技术

研 究 方 向 机器智能

授予学位日期 2015 年 月 日

基于链接和页面内容的主题爬虫算法的研究与应用

计算机科学与技术　专业

研究生：　　**指导老师：**

当今的社会是信息的社会，每天我们都会获取到各种各样的信息，及时地获取信息对于企业和个人的重要性越来越高。一方面，企业决策者和企业各个部门需要及时获取相关行业和技术的最新动态，以便及时了解行业发展趋势而快速准确地做出相应的对策；另一方面，个人的成功很大程度上也取决于其是否能及时地获取需要的信息。然而，随着互联网信息的快速增长，传统的搜索引擎已经不能满足用户对特定领域和主题的搜索需求，忙碌的工作和学习也常常让人们没有时间去获取自己想要的信息。这时，个性化信息推送系统便应运而生。个性化信息推送系统，即运用爬虫技术、信息推送技术等，向用户推送其感兴趣的信息的新型信息传播系统。其实质就是一个面向主题的搜索引擎，又称为垂直搜索引擎，而主题爬虫是主题搜索引擎的主要部分，亦是本文的主要研究内容。

本文对主题搜索引擎与主题爬虫的工作原理及流程进行了详细的介绍和分析。在此基础上，研究了主题爬虫如何计算页面的主题相关度和如何计算待爬取链接的优先级两个关键问题。在如何计算页面的主题相关度方面，本文研究并改进了鱼群搜索（Fish Search）算法的主题相关度计算方法，提出了一种基于关键词位置的页面主题相关度计算算法。该算法对页面上出现的不同位置的关键词赋以不同的权值，综合加权后计算得到一个连续值，用这个连续的值来代表页面的主题相关度。在如何计算待爬取链接的优先级方面，本文研究了基于网页链接重要度的网页排序（Page Rank）算法，并针对其容易出现的主题漂移问题，结合基于关键词位置的 Fish Search 算法，提出了一种基于页面主题的 Page Rank 算法，该算法计算待爬取 URL 的优先级时同时考虑了网页链

接的重要性和网页内容的主题相关度。

此外，本文实现了个性化信息推送系统-及时推信息推送系统的爬虫子系统，并将上述算法应用于该爬虫子系统中。该系统具有 PC、Android 和 IOS 三个客户端，并已成功上线。上线以来，该系统的爬虫子系统能够爬取到用户所需的信息。

本文的主要工作如下：

(1) 针对主题爬虫算法的页面主题相关度计算的关键问题，详细分析和研究了 Fish Search 算法的主题相关度计算方法，该算法在计算页面的主题相关度时没有考虑页面关键词位置的重要性，并且使用离散值表示页面的主题相关度，没有完全区分页面的主题相关程度的高低，因此，本文提出了一种基于关键词位置的页面相关度计算算法。实验表明，该算法能够更加准确地计算页面的主题相关度，提高爬虫的查准率；

(2) 针对主题爬虫算法的待爬取链接（URL）的优先级计算的关键问题，详细分析和研究了基于网页链接重要度来确定爬行顺序的 Page Rank 算法，针对 Page Rank 算法容易出现的主题漂移问题，考虑到只用页面链接的重要性计算待爬取 URL 的优先级是不够的，结合网页内容的主题相关度一起来判断待爬取 URL 的优先级，便能避免这个问题。因此本文结合基于关键词位置的 Fish Search 算法，提出一种基于页面主题的 Page Rank 算法。实验表明，改进后的算法能够更加准确地计算待爬取链接的优先级，保证爬虫一直爬取主题相关的页面，提高爬虫的查准率；

(3) 在对两个算法进行改进的基础上，本文实现了个性化信息推送系统–及时推信息推送系统的爬虫子系统，并将上述算法应用于该爬虫子系统中。

关键词：主题搜索引擎；主题爬虫；鱼群搜索算法；网页排序算法；信息推送系统；

Research and Application of the focused crawler based on links and page content of the web

Major: Computer Science and Technology

Graduate: **Supervisor:**

Recent civilization is a society of information. Every day we are getting into all sorts of information. The importance of timely access to information for enterprise and individuals is increasing. On the one hand, the enterprise decision-makers and departments need timely access to the latest developments related to industry and technology, in order to understand the industry development trends rapidly and accurately make the appropriate steps. On the other hand, individual success is largely depends on whether he can obtain required information in a timely manner. However, with the rapid growth of the Internet information, traditional search engines cannot meet the needs of users to search in specific areas and themes, and people are so busy in their daily routine that they don't have enough time to get the proper information that they want, when personalized information push system came into being. Personalized information push system, also called the new information dissemination system, uses the crawler technology and the information technology to push their users information that interested. Its essence is a vertical search engine. The focused crawler is the central theme of the vertical search engine. The focused crawler is the main content of this thesis.

In this thesis I introduce and analyze the work principle and process of the vertical search engine and focused crawler in detail. On the basis of these information I studied two key issues. One is how to calculate the topic relevance of pages, and the second is how to calculate the priority of links to be crawled. In terms of how to calculate the topic relevance of pages, in this thesis I improve the

relevance calculation method of Fish Search and proposes a page topic relevancy calculation algorithm based on keyword locations. The algorithm assigns different weights for different positions keywords appear on the page, and calculate a weighted composite continuous value to represent the topic relevance of the page. In terms of how to calculate the priority of URL to be crawled, this thesis studies Page Rank algorithm based on the importance of links to calculate the priority of URL to be crawled. In order to solve the problem of the Page Rank algorithm, combined with the improved Fish Search algorithm, an improved Page Rank algorithm based on the topic of the page is proposed. This algorithm considers both the importance of web links and web content when calculating the URL to be crawled.

In addition, in this thesis I implement the crawling subsystem of the personalized information push system “Timely Push Information” and apply the above algorithm to the subsystem. The system has three clients PC, Android, and IOS. These are successfully running on-line. The crawling subsystem have crawling the information required by the user.

The main work is as follows:

(1) On the calculation of the topic relevance of pages, topic relevancy calculation method of Fish Search algorithm is detailed analyzed and studied. This algorithm does not consider the importance of the page keyword position when calculating the topic relevance of the page, and use a discrete value to represent the topic relevance of page. So the topic relevance of pages are not fully distinguished related to the degree of discretion of the page. In order to solve these problems, this thesis proposes a page relevance computation algorithm based on keyword locations, experiments show that the proposed algorithm calculate the topic relevance of page more accurately, and improve the precision of the crawler.

(2) On the calculation of the priority of URL to be crawled, the Page Rank algorithm based on the importance of web links is detailed analyzed and studied to determine the priority of URL to be crawled. According to the theme drift problem of the Page Rank algorithm, only use the importance of the web links is not enough

to calculate the priority of the URL to be crawled, combined the topic relevance of the web content to determine the priority of URL to be crawled can avoid this problem. In order to solve the problem of the Page Rank algorithm used to predict the link, combined with the improved Fish Search algorithm, in this thesis I propose an improved Page Rank algorithm based on the topic of the page. Experiments show that the improved algorithm can calculate the priority of URL to be crawled more accurately, always ensure crawling into topic related pages, and improve the precision of the crawler.

(3) On the basis of both the improved algorithms, in this thesis I implement the crawling subsystem of the personalized information push system "Timely Push Information" and apply the above algorithm to the subsystem.

Key Words: vertical search engine; focused crawler; Fish Search algorithm; Page Rank algorithm; information push system;

目　录

1 绪论

1.1 研究背景以及意义

1998 年，联合国新闻委员会正式提出网络媒体的概念，网络媒体是继报纸、广播、电视三大传统媒介之后出现的新媒体[1]。现代网络媒体还包括网络报纸、网络电视和网络短信等，是网络媒体通过强大的计算机技术将不同的媒体融合在一起产生的新媒体形态。近几年来，网络发展很快，据中国互联网信息中心（CNNIC）第 35 次《中国互联网网络发展状况统计报告》显示，截至 2014 年底，我国网民数量达 6.49 亿，计算机使用率达 47.9%，与 2004 年比较，网民数量增加了 9 倍[2]。

表 1-1 全球互联网信息分布表

全球互联网文本信息	超过 100 亿条
新闻信息	超过 80 亿条
招聘信息	超过 10 亿条
招商信息	超过 5 亿条
通知信息	超过 3 亿条
其他信息	超过 2 亿条

随着计算机技术的发展和信息化的日益深入，传统的信息获取方式已不能满足人们的需要。一方面，当今网络上的信息正在以几何级数的速度增长，如表 1-1 所示，互联网上的文本数据达到了 100 亿条，其中招聘信息超过 10 亿条，新闻信息超过 8 亿条，招商信息超过 5 亿条，通知信息超过 3 亿条，使得用户可以在越来越丰富和广阔的网络信息中，查询自己所需要的信息。但是，网络上的关键信息没有任何规律和结构特征，人们很难从大量网络信息中准确找到对自己有用的信息。另一方面，及时地获取信息对于企业和个人的重要性越来越高。现代企业中，企业管理者如果及时地掌握了相关行业的最新、最准确的信息，就能在任何关头做出合适的决策，不错过任何商机；同样的，个人准确及时的获取信息也很重要，个人的成功很大程度上依赖于其信息获取的快

慢，有了信息上的准备才能不错过机会，才能成功。

那么如何准确及时地找到对自己有用的信息呢？搜索引擎是目前最高效最常用的查找信息的途径。搜索引擎系统通过网络爬虫、索引、排序等技术将网络上的信息归类，只需要在搜索引擎系统中输入关键词，就能快速找到相关的信息。但是，网络上的信息数量很大，人们感兴趣的只是其中的一小部分，每天花费大量的时间浏览网页，非常浪费时间；并且，现在人们的生活节奏越来越快，极易忘记搜索网络信息而错失重要通知或者招商机会。

根据用户的特点，定制其感兴趣的网站或者关键词，每天推送网站更新的网页信息给用户，这种新的信息获取途径被称为个性化信息推送。个性化信息推送的关键在于网络信息的爬取，其实质就是一个用来搜索网络上符合查询主题的信息的搜索引擎，而主题爬虫又是主题搜索引擎的核心[1]，因此是本文的主要研究内容。

1.2 国内外研究现状

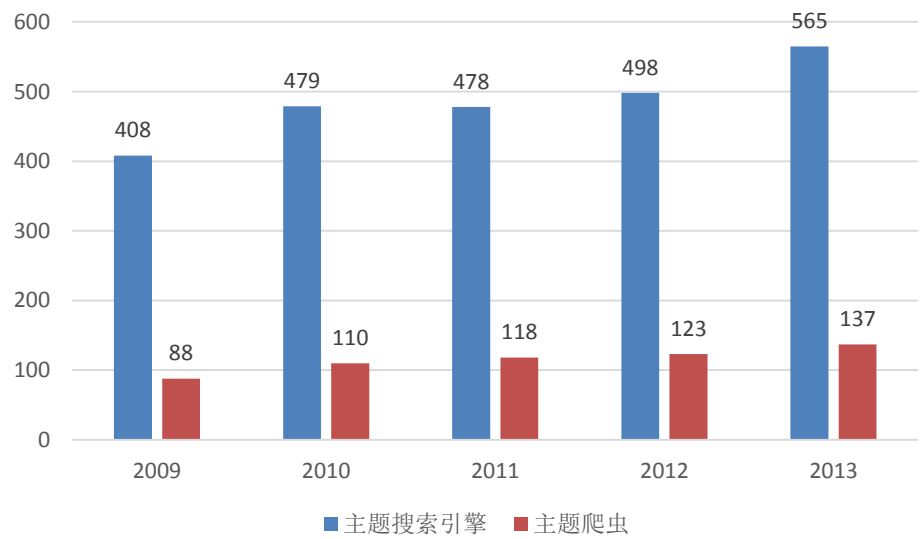

图 1-1 主题搜索引擎以及主题爬虫相关论文发表情况

主题搜索引擎是通用搜索引擎的延伸和细化，为特定行业的用户提供相关

的具体信息，成为了继通用搜索引擎之后的新的研究热点。主题搜索引擎的关键部分主题爬虫，也受到了广大搜索引擎研究者的关注。图 1-2 是在 Web of Science 网站分别以关键词“vertical search engine（主题搜索引擎）”和“focused crawler（主题爬虫）”检索 2009 年到 2013 年所发表文章情况，从图中可以看出，有关主题搜索引擎和主题爬虫的文章数量逐年递增，表明主题搜索引擎和主题爬虫仍然是搜索引擎研究的热点。

下面分别介绍传统搜索引擎、主题搜索引擎、传统爬虫和主题爬虫的国内外研究现状。

■ 传统搜索引擎

国外的搜索引擎主要有 Google、Yahoo、Bing、AltaVista 等，国内的搜索引擎主要有百度、SOSO 搜索、网易有道搜索等[3]，根据市场研究公司 Net Applications 最新数据展示，百度市场份额上升至第二位，Google 依然保持在第一位，如图 1-2 所示。其中，Yahoo 是最早的提供目录索引的搜索引擎，是第一代搜索引擎的代表，于 1994 年 4 月发布，在很长一段时间内是搜索引擎的领跑者。直到 1998 年 10 月 Google 的出现，Google 采用 PageRank 算法对所有网页进行排序，根据网页的重要程度返回搜索结果，已经证明是非常有效的搜索引擎。目前，Google 已经有 40 多种语言版本，可以根据 IP 地址自动切换语言，Google 作为全球规模最大的搜索引擎已经是一个不争的事实。Bing 是微软公司 2009 年推出的，发展很快，已成为北美第二大搜索引擎。AltaVista 虽然没有 Google 那么多用户数量，但是它的搜索结果更丰富，搜索范围更大，就连一些鲜为人知的偏僻站点也能访问到，AltaVista 也被公认为最好用的搜索引擎之一。百度公司于 2001 年 1 月成立，并于 2001 年 10 月发布百度搜索引擎。百度搜索引擎主要针对中文搜索，经过多年的发展，其中国市场份额超过 Google，成为国内搜索引擎的领跑者。SOSO 搜索是腾讯公司 2006 年推出的搜索引擎，依托于腾讯用户群和平台，有着广阔的前景。网易有道搜索是网页公司于 2007 年推出的搜索引擎，也在不断的发展中。

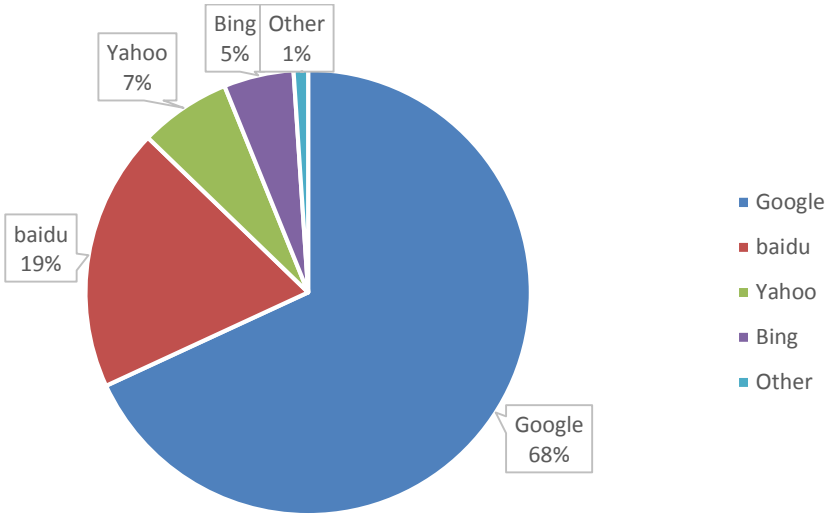

图 1-2 全球搜索引擎份额分布图

随着信息的多元化增长，就连 Google 和 Baidu 等功能强大的搜索引擎也难以满足特定领域和主题的信息检索，这是需要一个数据更专业、分类更细致、更新更及时的新型搜索引擎，于是面向主题的搜索引擎诞生了，并成为当今社会的研究热点[4-7]。

■ 主题搜索引擎

由于使用主题搜索引擎的用户都是需要了解某一特定领域或者主题的信息，并且都是相关行业的专业人士，因此，主题搜索引擎为用户提供的信息范围很小，但极具针对性。

国外代表性的主题搜索引擎有 Scirus 系统、Medical Matrix 系统等，国内也出现了一些非常成功的主题搜索引擎，代表性有搜房网、搜职网、百度视频搜索等[8]，如表 1-2 所示。Scirus 系统主要用于检索科学信息，是目前全球最全面的科技论文网站之一，在 2001 年被《搜索引擎观察》评为最佳专业搜索引擎。Medical Matrix 系统主要用于检索医学行业信息，为医药行业人员提供医药类关键词的检索，是目前全球最具权威的医学行业主题搜索引擎。搜房网拥有非常全面的房产项目信息，主要用于房产项目相关信息搜索，改变了传统房地产的营销模式。搜职网主要用于招聘信息和求职信息的搜索。百度视频搜索主要用于中文视频的搜索。

表 1-2 国内外主要主题搜索引擎表

主题搜索引擎	域名	功能
MEDICALMATRIX Ranked, Peer-Reviewed, Annotated, Updated Clinical Medicine Resources	http://www.medmatrix.org	搜索医学信息
SCIRUS for scientific information only	http://www.scirus.com	搜索科技论文
Baidu 视频	http://www.v.baidu.com	搜索中文视频
搜房 SouFun	http://cd.fang.com	搜索房产项目
搜职网 GlobeHR.com	http://www.soozhi.cn	搜索招聘和求职信息

■ 通用爬虫

通用网络爬虫[9]是最早出现的爬虫，它的目标是遍历整个互联网，搜集尽可能全面的页面，随着信息的不断膨胀，对整个网络的一次爬行的代价越来越大，出现了性能瓶颈。

根据需求的不同，通用爬虫发展成多种不同的形式，包括增量式网络爬虫、主题网络爬虫、元搜索的网络爬虫等[9]。增量式网络爬虫与通用网络爬虫的区别是只爬取新增加或者发生了变化的页面，大大减少了爬取的页面数，节约了时间开销。主题网络爬虫主要针对用户在特定领域和主题的搜索，它优先采集主题相关性高的页面，丢弃主题相关性低的页面，很好的满足了特定用户对特定领域的需求，成为最近几年的研究热点[10-14]，也是本文的主要研究内容。元搜索的网络爬虫是元搜索引擎的核心，它将用户的搜索请求提交给多个不同的大型搜索引擎，并再次整合返回的搜索结果后提供给用户。

■ 主题爬虫

主题爬虫是网络爬虫的一种，由 chackrabarti 于 1999 年提出[15]。其与通用网络爬虫最主要的区别就是爬行策略的不同。通用爬虫一般采用深度优先、广

度优先或者深度优先和广度优先结合的搜索策略，由于上述策略忽略了网页爬行顺序和重要性的计算，会爬取大量主题无关的页面，并且效率很低。

基于通用爬虫的爬行策略，主题爬虫提出了一种结合已知的网页信息而自我调整的搜索策略，主要有基于网页内容和基于网页链接结构的搜索策略。Best First Search[16]和 Fish Search[17]算法等是基于网页内容的主题爬虫算法。Best First Search 算法是通过比较网页内容与主题关键词的相关度来评估网页的重要程度，以确定待爬取链接队列中链接的爬行顺序；Fish 算法首先通过匹配页面中的关键词与主题关键词作为判断主题相关度的重要参数，使用离散值 0、1 和 0.5 代表页面的主题相关度，以确定待爬取链接队列中链接的爬行顺序。PageRank[18]和 Hits[19]算法是基于网页链接结构的主题爬虫算法，都是通过计算网页链接的重要度，来确定待爬取链接队列中链接的爬行顺序。

1.3 论文研究内容

本文的主要研究内容如下：

一、针对爬虫的爬行策略关键问题，介绍了传统爬虫和主题爬虫的几种爬行策略及其算法，并分析比较了这几种爬行策略的优缺点。

二、针对主题爬虫爬行策略的页面主题相关度计算的关键问题，对 Fish Search 算法进行了分析与研究，提出了一种基于关键词位置的页面主题相关度计算算法。实验的结果显示，改进的算法能够更加准确地计算页面的主题相关度。

三、针对主题爬虫爬行策略的待爬取链接的优先级计算的关键问题，结合 Page Rank 算法和基于关键词位置的 Fish Search 算法，提出了一种基于页面主题的 Page Rank 算法，取合适权重综合计算得到待爬取链接的优先级。实验的结果显示，改进的算法能够更加准确地计算待爬取链接的优先级，爬取更多主题相关的页面。

四、将相关算法应用到《及时推信息推送》项目的爬虫子系统中，并将爬取到的信息在多个客户端里展示。

1.4 论文组织结构

本文共分为七个章节，其内容安排如下：

第一章在详细论述了本文的研究背景的基础上，对主题搜索引擎及主题爬虫在全球范围内的学术研究现状进行了描述，进而指出了研究与分析主题爬虫的重要性，引出本文的研究内容。

第二章首先根据搜索引擎的背景，介绍了搜索引擎的工作原理，接着分别介绍了传统搜索引擎和主题搜索引擎的特点，进一步展示出主题爬虫在搜索引擎的作用。然后重点介绍了主题爬虫，通过对比主题爬虫与传统爬虫的基本原理和结构，详细的讨论了主题爬虫的几个关键问题。最后，介绍了相关的技术包括信息推送技术和消息队列技术。

第三章主要针对主题爬虫爬行策略的页面主题相关度计算的关键问题，对 Fish Search 算法及其页面主题相关度计算算法进行了分析与研究，针对其计算页面主题相关度的不足，提出了一种基于关键词位置的页面主题相关度计算算法，能够更加准确地计算页面的主题相关度。

第四章主要针对主题爬虫爬行策略的待爬取链接的优先级计算的关键问题，对 Page Rank 算法进行了分析与研究，针对其计算待爬取链接的优先级的不足，结合基于关键词位置的 Fish Search 算法，提出了一种基于页面主题的 Page Rank 算法，取合适权重综合得到待爬取链接的优先级，能够爬取到更多与主题相关的页面。

第五章属于实验设计和实验结果分析部分，实现了一个通用的爬虫框架，分别对原算法和改进后的算法进行实验，实验结果表明本文提出的算法是切实有效的。

第六章将本文提出的算法运用到《及时推新闻推送》项目的爬虫子系统中，并概述了系统的架构，最后展示系统相关模块的实现效果。

第七章是对全文工作的总结以及对未来工作的展望。该章总结了全文的内容与本文的贡献，并指出了将来工作所要解决的问题，并且对未来的工作进行了相应的展望。

2 主题爬虫相关理论与技术

2.1 主题搜索引擎技术

搜索引擎（Search Engine）是指根据一定的爬虫爬行策略，运用爬虫程序在网络上搜集大量网页信息，在对网页信息分析处理后存入数据库，当用户搜索信息时，从数据库中查询搜索结果返回给用户的系统[20; 21]。

2.1.1 搜索引擎的背景与原理

在网络诞生的初期，由于网络上的信息很少，查找比较容易。但是，随着互联网的发展，网络上的信息量越来越大，信息查找变得越来越难，这样，为了满足用户检索信息的搜索引擎便出现了。

现代搜索引擎的起源是 McGill University 三名学生（Alan Emtage、Peter Deutsch、Bill Wheelan）发明的 Archie[22]。Archie 虽然是一个基于文件名查找的系统，搜索的不是网页，但是它已具有搜索引擎的基本特征，即通过计算机程序自动搜索信息，然后对其进行处理和组织后，提供检索服务。一直到九十年代中期，随着万维网（WWW）的出现，搜索引擎得到进一步的发展。出现了提供分类服务的目录式搜索引擎，以 Yahoo[22]为代表，可以支持简单的搜索。到现在为止，一批使用先进计算机技术的大型搜索引擎，如 Google、Bing 和 Baidu[22]等，已成为互联网上搜索信息的必不可少的一部分。随着技术的不断提高，搜索引擎必然向专业化、个性化和智能化的方向不断发展。

现代搜索引擎主要由搜索器、检索器、索引器和用户接口四个模块组成，如图 2-1 所示，下面分别介绍各个模块的功能：

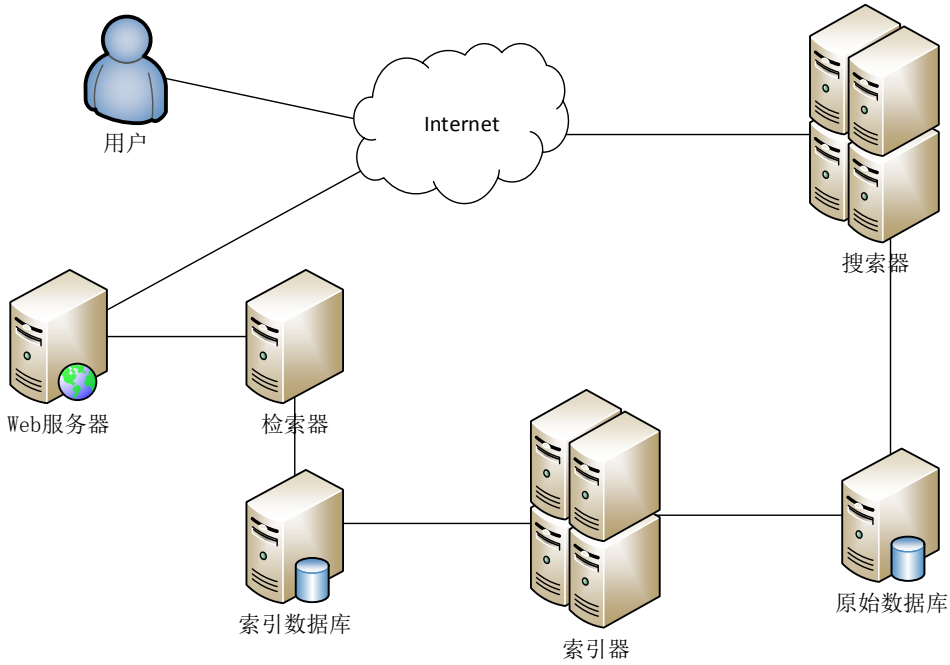

图 2-1　搜索引擎架构图

(1) 搜索器：即爬虫程序，他的作用主要是采集、通过 HTTP 协议访问网络上的超链接并爬取其对应的页面信息。

(2) 索引器：索引器通过分析爬取到的网页，提取网页内容的一些重要信息，根据一些相关度算法，计算得到网页中的所有关键词的相关度，最后将这些相关度信息建立网页的索引数据库。

(3) 检索器：检索器根据用户输入的搜索关键词，在索引数据库中查找包含该关键词的网页索引数据，并将搜索到的网页返回给用户。其中，检索器根据一定的评估算法，将用户最感兴趣的内容放在前面。

(4) 用户界面：搜索引擎的用户界面主要提供搜索关键词的输入接口，和显示搜索结果的区域。

搜索引擎的工作流程十分复杂，但可以简单归纳为三步：如图 2-2 所示，首先，搜索引擎的搜索器在互联网上尽可能多的发现网页超链接并爬取其对应的网页内容，从而建立起一个网页数据库；然后，索引器对爬取到的网页中的关键信息建立索引，保存这些索引到数据库中；最后，由检索器根据用户搜索的关键词，在索引数据库中查找用户最关心的页面，按用户感兴趣程度依次排

序返回到用户界面。

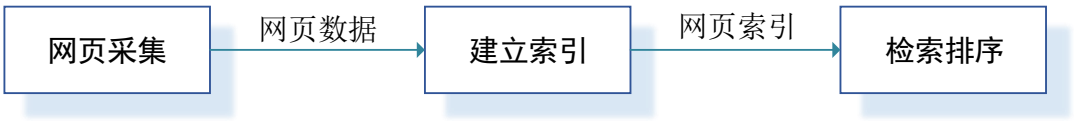

图 2-2　搜索引擎简单流程图

其中网页采集是搜索引擎的基础，每个搜索引擎都有自己网页爬取程序（spider）来采集网页，也称为机器人（robot），就是所谓的爬虫程序。该爬虫程序从初始链接节点开始，访问该链接并爬取其链接对应的网页内容，这些网页内容被称为网页快照。然后，继续根据一定的爬行策略，访问其他未访问过的链接，保存对应页面，一直爬行，达到结束条件就停止。

2.1.2 传统搜索引擎

搜索引擎按照其出现的年代可以分为三代：第一代是目录式的检索，用户只能按预先设定的分类进行搜索；第二代是基于关键词的检索，用户可以在分类的基础上输入一个或者多个关键词进行搜索；第三代是在第二代上的进一步改进，加入了智能分析，用户能更快更多的搜索到对自己有用的信息。传统搜索引擎根据工作方式的不同，可以分为三类：

(1) 目录式搜索引擎：一般以人工方式搜集网页信息，形成摘要，并存放于事先确定的网页分类框架中，用户能在具体网页分类目录搜索相关信息。代表的搜索引擎有 YAHOO、搜狐和新浪等。

(2) 机器人搜索引擎：一般使用特定的爬虫程序以一定的爬虫爬行策略自动在互联网上爬取页面信息，由索引器对爬取到的页面信息建立索引，由检索器根据用户输入的查询关键词搜索索引数据库，最后返回检索结果给用户。机器人搜索引擎是目前应用最多的搜索引擎，用户能搜索所有网页信息，代表的搜索引擎有 Baidu、Google 等。

(3) 元搜索引擎：这类搜索引擎自己不搜集网页，而是将用户的查询请求提交其他给搜索引擎进行检索，并将检索结果经过重新排序和去重后返回给用户。代表的搜索引擎有搜星等。

传统的搜索引擎都是面向所有信息的，随着信息的多元化增长，这种面向所有信息的综合性搜索引擎显然不能满足特定用户的特殊搜索要求，这类用户需求的信息主要是针对某一特定领域或者主题的，综合性搜索引擎搜索不到大量领域相关的页面。为了解决这类用户的问题，需要一种数据更加专业、分类更加细致的搜索引擎，主题搜索引擎便出现了。

2.1.3 主题搜索引擎

搜索引擎经过前三代的发展，人们检索信息的效率已经得到很大的提高，但是，搜索引擎的查全率和查准率还有很大的提高空间[21]，因此只针对特定领域或者特定主题的、用户能定制检索方向和内容的新一代搜索引擎应运而生。

所谓的面向主题的搜索引擎即主题搜索引擎，是指以某一特定领域或主题的网页资源库为目标，通过一定的策略爬取符合这一领域或者主题的网页信息，能够为特定行业用户，提供其行业相关信息的搜索服务[22]。

主题搜索引擎只爬取特定领域或者主题的网页信息，建立的索引较小，容易管理。针对这一特定领域，能够检索出大量相关信息。主题搜索引擎和传统搜索引擎的主要有以下几个方面的不同：

(1) 服务目的不同：传统搜索引擎面向所有用户，提供对所有领域信息的搜索，而主题搜索引擎，则是面向专业用户提供对某一特定领域和主题的信息搜索服务。

(2) 搜索策略不同：传统搜索引擎是在整个互联网上搜索信息，采用遍历整个互联网的搜索策略，而主题搜索引擎则采用一定的爬行策略预测下个需要爬取的链接，动态改变爬虫的爬行方向，使爬虫尽可能地选择主题相关的网页进行爬取，从而不必遍历整个互联网。

(3) 硬件和网络要求不同：传统搜索引擎需要遍历整个互联网，其硬件和网络需求一般较高，而主题搜索引擎没有对整个互联网进行遍历，不仅使用了较少的网络资源，而且只需要维持一个小规模的索引数据库，所以硬件要求相对也较低。

2.2 主题爬虫技术

网络爬虫[23]，也称为 WebSpider、WebCrawler、WebRobot 或者 WebWorm，是一个根据网页链接来遍历网络上的页面，并按照标准 HTTP 协议访问其页面内容的程序。

2.2.1 通用爬虫

通用爬虫是最基本的网络爬虫，主要用于搜索引擎的数据采集。作为搜索引擎的核心部分，为了每次检索的信息能够准确全面，需要把网页爬取下来，建好索引存储在数据库中，检索结果直接从数据库中查找并返回，通用爬虫的爬取结果直接影响搜索引擎的整体搜索质量。随着技术的发展和人们需求的提高，网络爬虫越来越多地应用于用户兴趣分析、微博话题挖掘以及个性化信息推送等多种应用中。

通用爬虫的基本流程如图 2-3 所示，在爬取过程中，通用爬虫从一个或者多个初始 URL 开始，不断的增加和移除 URL。首先，利用 HTTP 协议访问初始 URL 并下载其对应的页面，然后，分析并抽取出网页中出现的新 URL，并过滤掉非法 URL，最后将未被访问的新 URL 加入到待爬取 URL 队列中。整个过程一直循环，直到队列为空或者满足提前定义的结束条件。通用爬虫实际上是沿着网页超链接，根据深度优先、宽度优先、或者其他爬行策略遍历网络信息的过程。现代网络爬虫除了必须要具有好的健壮性，还需要具有分布式、可扩展、高性能、高效率、高时效等特性[24]。

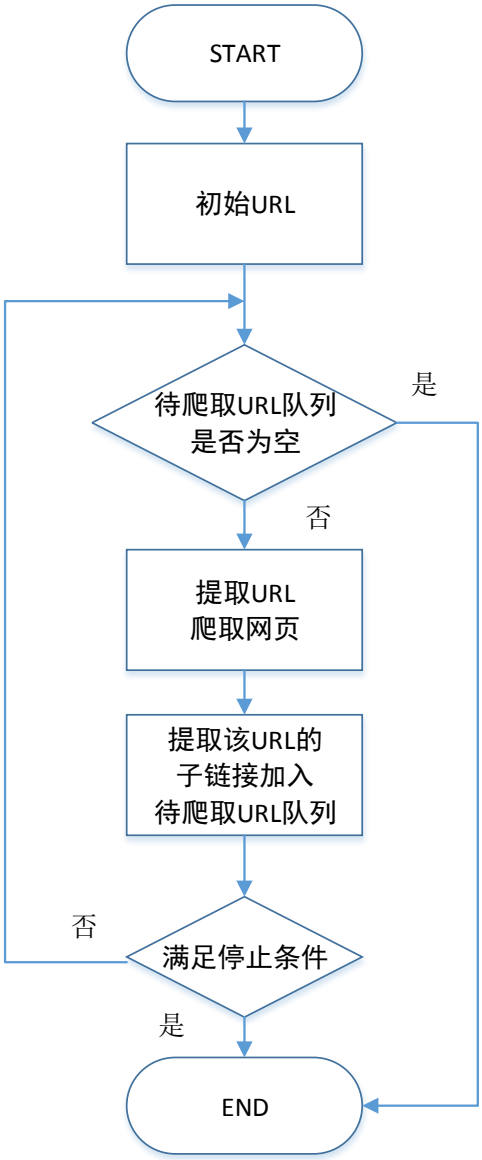

图 2-3 通用爬虫基本流程图

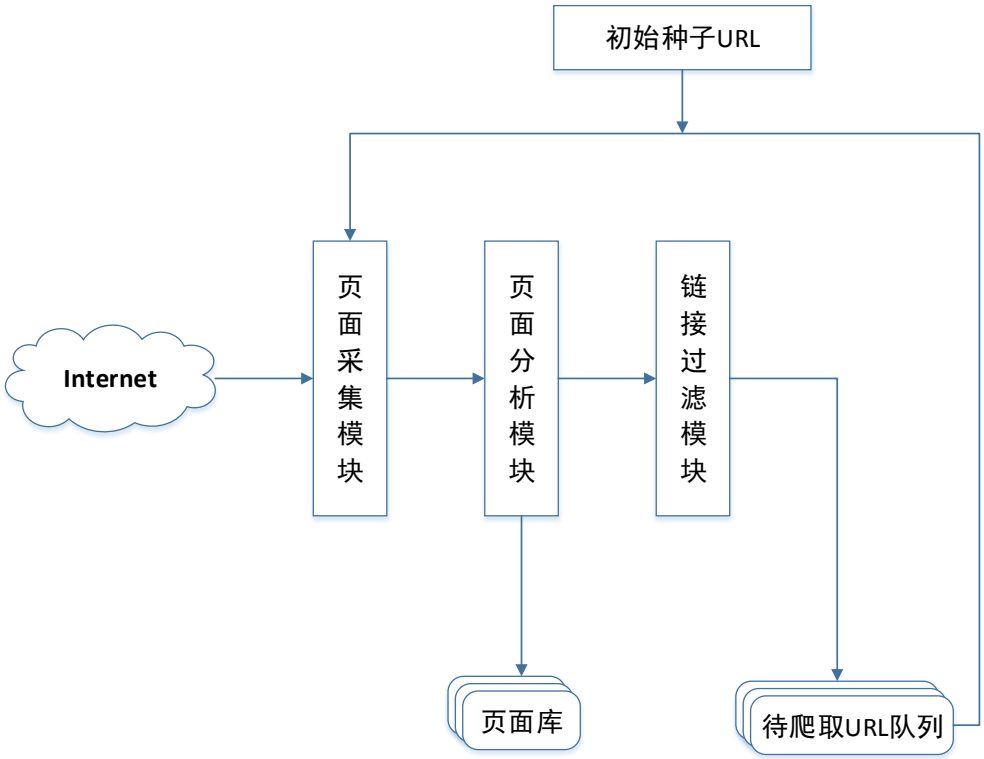

图 2-4 通用爬虫体系结构图

通用网络爬虫的结构如图 2-4 所示，下面分别介绍通用网络爬虫各个模块的功能：

(1) 页面采集模块，该模块根据 HTTP 协议访问初始种子 URL 或者待访问 URL 队列中的链接，爬取其对应的网页内容，保存下来以供后续模块使用。

(2) 页面分析模块，该模块根据页面采集模块爬取到的页面信息进行分析，抽取特征信息和 URL 超链接，并将提取的链接规范化。

(3) 链接过滤模块，该模块对页面分析模块分析过的链接进行过滤，过滤掉已访问过的，或者已存在待爬取 URL 队列中的链接。

(4) 页面库：存放页面采集模块采集的页面。

(5) 初始种子 URL：预先设定的一条或者多条初始 URL。

(6) 待爬取 URL 队列：存放采集下来的页面经过页面分析模块分析后提取出的新 URL，以供页面采集模块选择队列里面的 URL 进行爬取，一直到队列为空或者达到预先设定的结束条件。

2.2.2 通用爬虫爬行策略

通用爬虫的目标是遍历整个互联网，尽可能多的采集网页，其爬行策略就是从初始 URL 节点开始遍历所有子链接，不分析网页链接和网页内容的重要性和主题相关度，直接爬行所有页面。其爬行策略如图 2-5 所示。

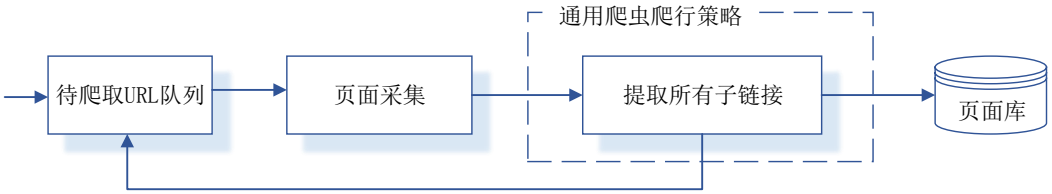

图 2-5 通用爬虫爬行策略

目前，通用爬虫的爬行策略主要为深度优先[25]、广度优先[26]和深度优先和广度优先结合[27]三种。深度优先、广度优先以及深度优先和广度优先结合的爬行策略都是将网络看成一个复杂而庞大的有向图，将网络上的所有网页当成节点，网页之间的超链接当成节点之间的连接，根据图论的搜索方法，对网页节点采取深度优先、广度优先或者深度优先和广度优先结合的遍历策略。下面以一个结构图如图 2-6 来说明这三种算法的区别。

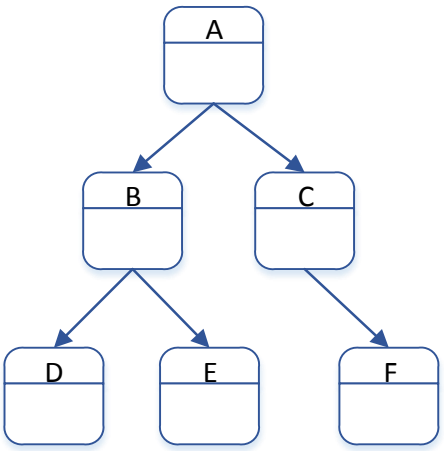

图 2-6 网页节点结构图

(1) 深度优先

深度优先的爬行策略是从初始网页节点开始，首先访问该节点，然后沿着一条超链接路径出发，访问其邻结点，一直爬行到没有包含任何超链接信息的网页节点为止，再返回访问其他未被访问的节点，循环下去，直到所有的节点都被访问到。那么根据深度优先的爬行策略，图 2-9 上的网页节点访问路径应该是 ABDECF。深度优先的爬行策略优点是爬取的页面很全面；缺点则是效率很低，当页面目录很深的时候，容易导致爬虫陷入。

(2) 广度优先

广度优先的爬行策略是从初始网页节点开始，访问此节点后，就访问其他没访问过的相邻节点，然后再访问剩下节点的邻结点，直到爬行结束。比如上图 2-9，从 A 节点访问开始，继续访问时可以选择 A 的邻结点 B 或者 C 访问，所以广度优先的爬行策略有多种爬行顺序。按照上图总共会出现多种爬行顺序，ABCDEF 是其中的一种。广度优先的爬行策略优点是当初始网页有足够多的链接时，会得到很好的爬取结果；但是，广度优先的爬行策略缺点也很明显，当爬行深度越来越高时，则会产生许多无效的网页，导致算法的效率降低。

(3) 深度优先和广度优先结合

针对深度优先和广度优先各自的优缺点，又出现了深度优先和广度优先结合的爬行策略。这种爬行策略会预设爬行深度或广度，当爬虫爬行到预设深度或者预设广度时就停止工作。所有它相比深度优先和广度优先，都提高了爬行效率，但还是会爬取到不少的无效的网页。

深度优先、广度优先以及深度优先和广度优先结合都几乎遍历整个网络，能爬取大量的网页，非常适用于通用爬虫的爬行。但是，对于主题爬虫来说，上述三种策略都没有对网页和网页内容作出评估分析，会爬取大量与主题无关的网页，不能满足主题爬虫的需求。

2.2.3 主题爬虫

2.2.3.1 主题爬虫工作原理

主题爬虫是通用爬虫的扩展，不仅仅是对页面的简单抓取，是有选择性的根据主题爬取更多的与主题相关的页面。

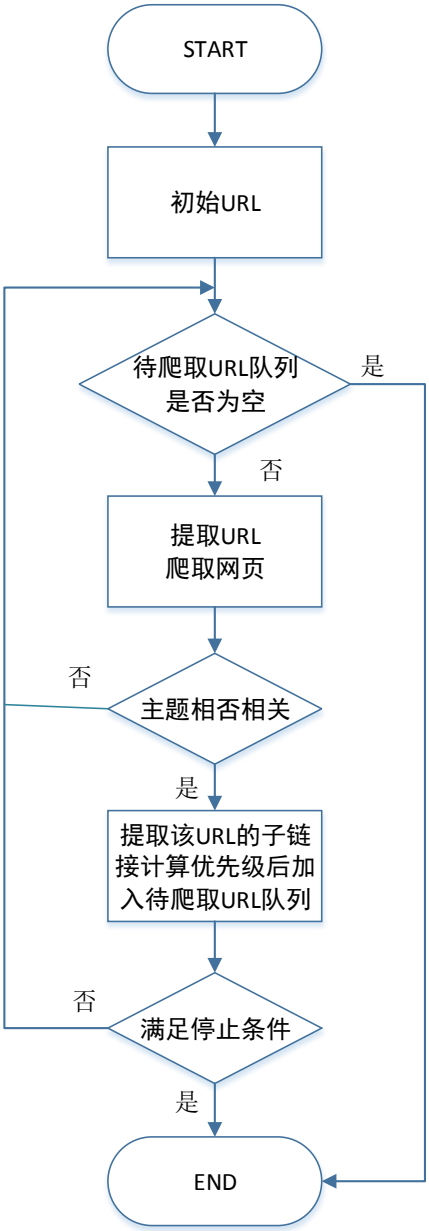

图 2-7 主题爬虫流程图

主题爬虫的基本流程如图 2-7 所示，主题爬虫在第一次爬取页面时，需要一个初始种子 URL，初始种子 URL 必须是与主题相关度很高的大型门户网站，这样才能获得较多与主题相关的 URL，提高爬取的准确率。接下来，从爬取下来的页面中提取页面内容，计算页面的主题相关度，相关度高于某个阈值则保

留，否则就丢弃；然后提取上述页面的超链接，过滤掉非法链接，结合网页内容的相关度或者链接重要度综合得出 URL 的优先级，放入待爬取的 URL 队列中，最后，取出优先级最高的 URL 继续爬取它的内容，循环此过程，直到 URL 队列为空或者满足结束条件就退出。

2.2.3.2 主题爬虫体系结构

主题爬虫的一般结构如图 2-8 所示。主题爬虫基于通用爬虫，增加了主题相关性分析的模块和链接优先级评价模块，其他模块的功能都大同小异，下面主要介绍这两个新增加的模块的功能：

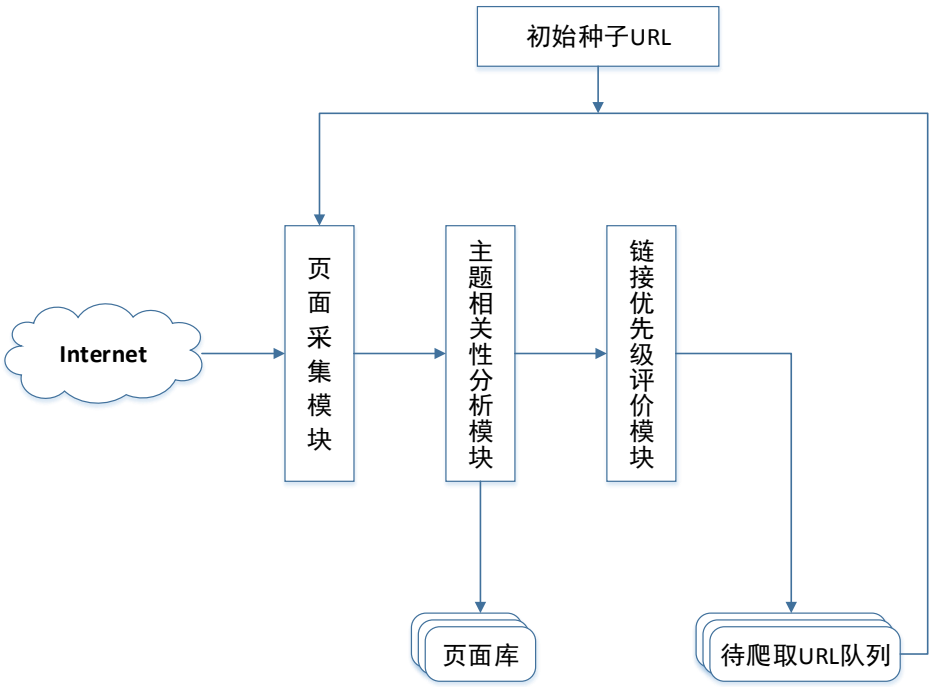

图 2-8 主题爬虫结构图

(1) 主题相关性分析模块：该模块采用多种相关度计算方法，计算采集到的页面是否属于某一个主题。通过页面的主题相关度值来判断页面是否与主题相关，如果该页面的主题相关度值高于某个阈值，就存入页面库中，否则，就丢掉该网页。

(2) 链接优先级评价模块：该模块的用于对待爬取的 URL 优先级的评价，此优先级用于指导爬虫选择下一次爬行的 URL，即确定下一个需要爬取的链接。一个 URL 优先级得分越高，就会越早被爬取，反之，得分低到一个阈值时则丢弃该页面及其所有子链接。好的 URL 优先级评价算法不仅能爬取更多主题相关的页面，还能减轻爬虫抓取到无关页面的负担，节约带宽。主题爬虫是通用爬虫的扩展，不仅仅是对页面的简单抓取，是有选择性的根据主题爬取更多的与主题相关的页面。

2.2.3.3 主题爬虫设计目标

主题爬虫的目标是在较少的时间内以较少的资源爬取到更多与主题相关的网页[23]。页面的主题相关性分析和 URL 优先级评价方法策略是实现这一目标的关键所在，其与主题爬虫的目标是紧密相关的。因此主题爬虫主要需要解决以下两个问题：

(1) 如何判定爬取下来的网页与主题的相关性？ 对于已爬取下来的页面，一般通过文本检索和挖掘技术计算其主题相关度。

(2) 如何确定待爬取的URL的优先级顺序？通常是根据已爬取下来的页面的重要程度，这个重要程度可以是页面相关度等多种角度描述，将此重要度按一定规则分配给子链接，并按照重要程度高低插入到待爬取的 URL 队列中。

以上两个问题就是爬虫的爬行策略的两个关键问题，就是指如何在海量的信息网站中爬取到人们感兴趣的信息，根据已有的信息选择下一个爬行的链接的策略，因此，爬虫爬行策略的好坏直接影响整个爬虫系统的性能。如果能设计出一个好的爬行策略，那么爬虫就能最大可能达到我们需要的效果。

2.2.4 主题爬虫爬行策略

主题爬虫的爬行策略[28]首先需要对页面进行分析，即分析网页链接和页面内容，确定网页的重要度以及主题相关度，然后提取子链接并计算子链接的优先级，确定待爬取 URL 的爬行顺序，其流程如图 2-9 所示。

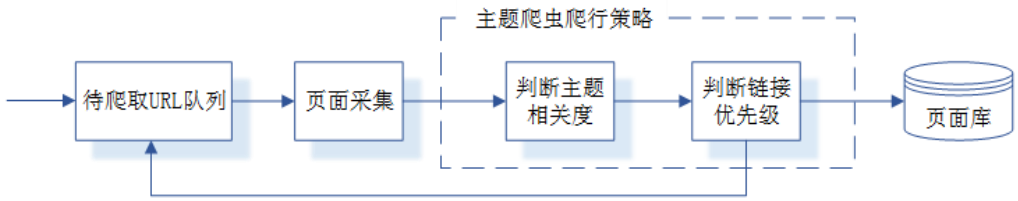

图 2-9 主题爬虫爬行策略

通用爬虫的爬行策略效率低，会产生大量无效网页的问题，不能应用于主题爬虫算法中。目前主题爬虫的爬行策略主要有基于网页链接、基于网页内容的爬行策略。基于网页内容的爬行策略需要对纯文本和超文本进行分类聚类，首先使用对 Html（网页超文本）进行分类聚类，将其中的纯文本抽取出来，再对纯文本进行分类聚类，并统计其中的关键词与主题关键词同时出现的次数，判断页面与主题的相关度；基于网页链接的爬行策略需要计算页面节点的入度和出度值，只有在爬虫对网络拓扑结构有相对完整的统计情况下，才能计算出准确的网页的重要性程度值。基于网页链接的爬行策略没有考虑网页内容的重要程度，也许页面在当前网页范围里很重要，但并不是所需要的。所以在主题爬虫系统中，基于网页链接的爬行策略一般不单独使用，需结合能判断网页主题的基于网页内容的爬行策略。

2.2.4.1 基于网页链接的主题爬虫爬行策略

基于网页链接的主题爬行策略主要是对超文本文档的链接结构进行分析，万维网是超文本文档的集合，超文本文档相互之间有一定的链接关系，形成一定的链接结构。基于网页链接的主题爬行策略就是通过对链接的有效评估来决定链接爬行的顺序。万维网信息分散，但经研究表明，其链接结构具有自组织性：通过超链接，相同或者相关主题的网页结点聚集在一起，形成结点区域，区域内部的结点存在紧密联系，而区域之间的结点很少或者根本没有联系。并且，区域内部的结点也存在一定的规律，其结点主要分为两类，一类指向其他结点，一类被前一类结点所指向，前一类结点称为中心结点，后一类结点称为权威结点。如何评估链接主要有两种假设，从页面 A 到页面 B 的一条链接是页面 A 对页面 B 的一种推荐。若页面 A 与页面 B 被同一页面链接，则他们可

能有相同或者相关的主题。万维网的这种链接自组织性结构为链接的评估提供了依据[29]。

Hits 算法和 Page Rank 算法是常见的链接分析算法，两者都是通过分析网页的拓扑结构，得出网页的重要程度评价。Hits 算法由 Kleinberg 于 1999 年提出，是一著名的基于网页链接的计算页面重要程度的算法。在 Hits 算法里面，Kleinberg 提出了两个新概念：权威型网页（authority Page）[30]和中心型网页（hub Page）[30]。权威型网页指的是被很多链接指向的页面，即与主题相关度很高的页面，是 Hits 算法最后得到的页面；中心型网页则是本身没有什么特点，却指向很多权威型网页的页面。每一个页面都有一个权威值（authority）和中心值（hub），如果一个页面被很多页面链接，则其权威值就会很高，如果一个页面链接到很多权威型网页，则其中心值也应该会很高。权威型页面和中心型页面之间的关系如图 2-10 所示所示，页面 D 的权威值等于指向它的页面 A、页面 B 和页面 C 的中心值的和；页面 D 的中心值则等于页面 D 指向的页面 E、页面 F 和页面 G 的权威值的和。通过权威型页面和中心型页面的这种相互关系，来发现重要程度高的网页，这就是 Hits 算法的基本原理。

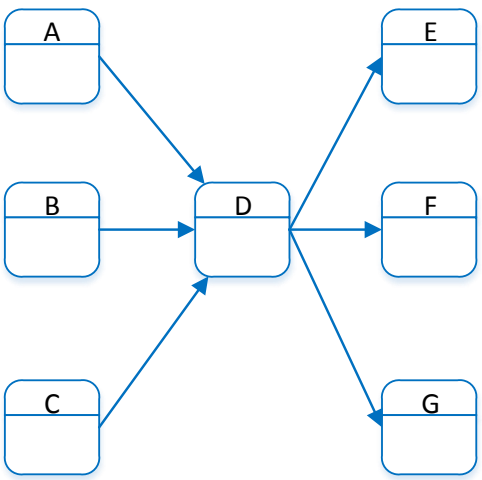

图 2-10 页面权威值和中心值计算图

Hits 算法构造 Web 子图主要有以下两个步骤：

(1) 构造根集合R_0（Root Set）：将预先设定的初始 URL 加入到跟集合R_0；

(2) 扩展根集合R_0到基集合B_0（Base Set）：将根集合R_0中的网页所指向的

URL 加入根集合R_0，将指向根集合R_0中的网页的 URL 也加入到根集合R_0，最后形成基集合B_0。

Hits 算法通过以上的步骤得到 Web 子图后，进行迭代计算直到收敛，计算出每个页面的权威值和中心值，最后，找出权威型页面。Hits 算法虽然收敛速度快，但其具有基于网页链接的爬行策略的同样的问题，忽视了网页内容与主题的相关性判断，造成“主题漂移”现象，从而爬取到许多与主题不相关的页面。Page Rank 算法将在第四章进行详细的介绍和分析改进。

2.2.4.2 基于网页内容的主题爬虫爬行策略

基于网页内容的爬行策略，主要以向量空间模型、布尔模型等信息检索模型为基础，利用网页页面内容（包括网页中文本信息、URL、锚文本和锚文本的上下文等）来评估网页的重要性，以此决定爬行顺序。基于网页内容的爬行策略的核心问题就是对上述超文本和纯文本进行分类和聚类。

纯文本的分类聚类[31]就是对页面纯文本进行检索，准确的说，指通过计算机使用自然语言中的词语对纯文本进行匹配查找。由于纯文本的检索忽略了网页页面内部的结构信息，必须限定检索的关键词集合，再进行检索，很少单独使用。超文本的分类聚类[31]是对页面超文本进行检索，超文本里面不仅包含了纯文本文字，还包含了一些对纯文本信息检索有用的结构信息。网页文本使用的超文本标记语言是一个半结构化的语言，具有结构特征和语义信息，根据特定的 HTML 标签检索文本，排除了无关的干扰信息，使得文本检索更加精确、快速。从而提高了爬虫爬行的效率。

Best First Search、Fish Search 是常见的基于页面内容的搜索算法。都是通过分析网页内容与主题的相关度，来得出网页的重要程度评价。Best First Search 算法是最早出现的基于网页内容的搜索算法，该算法的基本思想是通过比较网页内容与主题关键词的相关度来评估网页的重要程度，以确定待爬行 URL 队列中的爬行顺序[32]。Best First Search 一方面不断从待爬取 URL 队列中选择主题相关度高的链接爬行，另一方面，如果待爬取 URL 队列满了，则从队列中丢弃主题相关度低的页面。Best First Search 算法的具体步骤如下：

(1) 首先选择一个初始 URL 插入待爬取 URL 队列中；

(2) 从待爬取 URL 队列中取出初始 URL，下载初始 URL 对应的页面，提

取初始网页的子链接，并计算链接对应的页面与主题的相关度值；

(3) 如果待爬取 URL 队列未满，将新 URL 插入到该队列中。

(4) 如果待爬取 URL 队列已满，比较新 URL 与队列中的所有 URL 的相关度值，丢弃队列中相关度值最低的 URL。

(5) 队列为空时停止爬取。

Best First Search 算法根据网页内容与主题的相关度来评价网页的重要程度，以确定待爬取 URL 的爬行顺序。它的优点是算法流程简单清晰，容易实现，在主题相关性强的网页爬行时效果较好。但评估网页重要性的时候只考虑网页内容，忽视了网页链接的结构。Fish Search 算法将在第三章进行详细的介绍和分析改进。

2.3 其他相关技术

2.3.1 信息推送技术

随着互联网信息数量的快速增长，用户很多时候花费大量时间也难以在互联网上查到自己需要的信息。信息推送技术正是在这样的情况下诞生的，该技术通过用户特定的需求，按时将用户感兴趣的最新内容主动发送给用户，这样，用户就不必浪费精力在互联网上搜索信息。

信息推送技术[33]是一种新型的信息传播技术，指通过特定的渠道，以一定的标准或者协议向用户推送其感兴趣信息的技术。目前的信息推送的方式主要有以下几种：

(1) 邮件型信息推送：需要用户设置自己的邮箱，然后将用户感兴趣的信息推送到用户的邮箱中。

(2) 网页型信息推送：在一个特定的网页上推送用户感兴趣的信息，用户登录自己的账号查看。

(3) 专用型信息推送：在一个特定的 APP 软件上推送用户感兴趣的信息，用户登录自己的账号查看。

信息推送技术的核心主要是通过用户访问纪录等信息对用户兴趣进行建模，然后跟据用户兴趣模型对用户推送其感兴趣的内容。主要方法有基于内容的推送、基于协同过滤的推送和混合推送[33]。个性化信息推送系统[34]则是指运

用信息推送技术、爬虫技术等，向用户推送其感兴趣的信息的新型信息传播系统[33]。个性化信息推送系统让用户自己输入关注的网站和领域，及时推送网站上该领域上更新的信息给用户，弥补了传统搜索引擎难以解决的信息过载的不足，极大提高了搜索的准确率。

2.3.2 消息队列技术

消息队列技术[35]是应用程序间的通信方法。应用程序间通信不需要建立专用连接，通过读取队列里的消息来通信，使得应用程序接收方和发送方不必同时执行。其成熟的产品主要有有 IBM WEBSPHERE MQ、RabbitMQ 等。

RabbitMQ[35]是基于 AMQP 的 erlang 实现的。其核心是 Message Exchange、Message Queue 两种实体，一个应用程序产生的消息不是直接发送给另一个应用程序，而是发送给 Message Exchange，同时指定一个 Rounting Key，如果与 Message Queue 的 Binding Key 匹配，Message Exchange 就将消息发送给 Message Queue，等待另一个应用程序的读取。

主要有以下几个特点：

(1) 可以在多个平台上运行，支持多种操作系统。

(2) 提供消息的持久性支持，即服务器意外崩溃后消息能够恢复。

(3) 提供了消息接受者和消息发布者的确认机制，保证消息准确接收。

消息队列技术应用到爬虫系统中，使得爬虫能够在多个爬虫客户端爬取数据，然后再返回爬取结果到爬虫服务端存入数据库，极大得提高爬虫系统的效率，保证了信息的准确性和安全性。

2.4 本章小结

本章首先根据搜索引擎的背景，简单介绍了搜索引擎的基本概念和工作原理，接着分别介绍了传统搜索引擎和主题搜索引擎的特点，进一步展示出主题爬虫在搜索引擎的作用；然后，重点介绍了主题爬虫，通过对比主题爬虫与传统爬虫的基本原理和结构，详细的讨论了主题爬虫的几个关键问题；并针对爬虫的爬行策略关键问题，介绍了常用的爬虫爬行策略；最后，介绍了个性化信息推送系统其他相关的技术，包括信息推送技术和消息队列技术。

3 页面主题相关度计算

主题爬虫与通用爬虫的最大区别在于其主题相关度计算及相关的爬行策略，所以要实现一个完整的爬虫系统，采用合适的主题相关度计算算法极其重要。本章主要对如何分析页面得到相关度高的页面关键问题进行了研究，首先，深入研究了基于页面内容的爬行策略的代表算法 Fish Search 算法，针对该算法计算页面主题相关度的不足，改进了其计算页面主题相关度的方法，提出了一种基于关键词位置的页面主题相关度计算算法。

3.1 Fish Search 算法

Fish Search 算法是一种基于网页内容的爬行策略的经典算法，本节会对该算法进行研究和分析。

3.1.1 算法基本思想

Fish Search 算法，也称为鱼群算法。该算法将互联网假想成海洋，互联网中的链接假想成鱼，主题相关性高的网页假想成鱼的食物，因此爬虫的爬行过程就是鱼群的捕食过程，沿某个方向的遍历就是鱼群的捕食路径。当爬行过一个网页后，即捕食成功后，鱼群会在这里繁衍后代，提取的新的链接就是鱼群的后代，接着鱼群继续前行寻找食物。如果鱼群游到下一个地方时又发现了食物，就将父结点的遍历深度赋值给儿子结点；如果鱼群游到下个地方时没有发现食物，即网页与主题不相关，就将儿子结点的遍历深度设置为父结点的遍历深度减一，继续前行。若一直找不到食物，链接的遍历深度会逐渐变为 0，鱼就饿死了，即爬虫在一定的深度找不到主题相关的链接的时候就结束爬行了。这就是 Fish Search 算法的基本原理[17]。

其简单流程如图 3-1 所示。爬虫采集页面后，计算页面内容的主题相关度，若相关，其子链接的相关度值为 1，选择其子链接的前 m*width（m，width 为预先确定的参数）个链接插入到前端队列 B 中；若不相关，如果预设的 width 参数不为 0，其子链接的相关度值为 0.5，选择其子链接的前 width（width 为预先确定的参数）个链接插入到中间队列 M 中;否则，其子链接的相关度值为

0，并将其子链接都插入到末尾队列 E 中。接着爬虫选择相关度最高的 URL 进行爬行，一直循环下去，直到待爬取 URL 队列为空或者时间限制达到就结束。

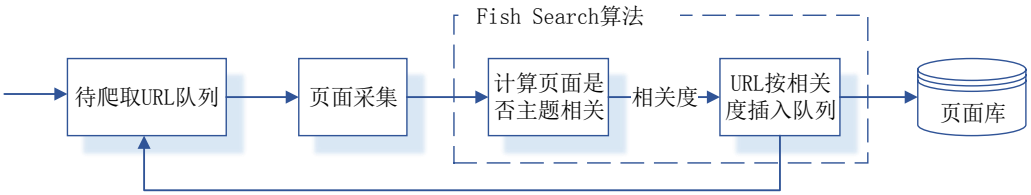

图 3-1 Fish 算法简单流程图

3.1.2 算法过程

Fish Search 算法的关键之处在于将待爬行 URL 队列分为了三部分，并动态维护了这三个 URL 队列，分别是前端队列 B、中间队列 M 和末尾队列 E。

其算法描述如下：

(1) 先选择一个初始 URL 放入待爬取 URL 队列中，并对最大爬取深度 depth 赋值。

(2) 提取初始网页的子 URL，如果初始网页是主题相关的，则其子 URL 的相关度值为 1，depth 值不变，如果初始网页不是主题相关的，则其相关度值为 0 或者 0.5，depth 值减一。并按照（3）（4）（5）将新的 URLs 插入队列。

(3) 若该 URL 对应的网页与主题相关，就选取 m*width（m，width 为预先确定的参数）个 URL 插入到前端队列 B 中，即队列的头后。

(4) 若该 URL 对应的网页与主题无关且 depth 不为 0，就选取 width（width 为预先确定的参数）个 URL 插入到中间队列 M 中，即队列中最后一个主题相关的节点后。

(5) 剩余的 URL 插入到末尾队列 E 中，即队列的尾巴。有充足的时间时才会爬取这个队列的 URLs。

(6) 队列为空或者时间限制达到时停止爬取。

其算法伪代码如表 3-1 所示：

表 3-1 Fish Search 算法伪代码

```
1.  Get 初始 URL 节点，width（width），depth（D），size（S），查询关键词，时间限制；
2.  Set 初始 URL 节点的深度，depth = D，并将初始 URL 插入空的队列中；
3.  While 队列不为空 and 已处理的节点个数小于 S and 时间限制未到
4.      Pop 队列中的第一个节点为当前节点；
5.      Compute 相关度 of 当前节点；
6.      If depth < = 0 Then continue
7.        If 当前节点是不相关的
8.       Then For 当前节点的前 width 个子节点
9.                Set 子节点的 potential-score = 0.5；
10.           For 当前节点的其余子节点
11.               Set 子节点的 potential-score = 0；
12.     Else For 当前节点的前 a*width 个子节点（a 通常为 1.5）
13.               Set 子节点的 potential-score = 1；
14.           For 当前节点的其余子节点
15.               Set 子节点的 potential-score = 0；
16.     For 当前节点的每个子节点
17.       If 子节点已经存在于优先级队列中
18.         Then Compute 队列中该节点与子节点 potential-score 的最大值；
19.              Replace 队列中该节点的 potential-score 值为上一步计算的最大值；
20.              Move 当前节点节点到排好序的队列中正确的位置；
21.         Else 将子节点按照它的 potential-score 值插入到排好序的队列中；
22.     For 当前节点的每个子节点
23.       If 当前节点是相关的
24.           Then Set 子节点的深度 depth = depth（当前节点）；
25.       Else 子节点的深度 depth = depth（当前节点） – 1；
```

```
26.          If 子节点已经存在于优先级队列中
27.          Then Compute 队列中该节点与当前子节点的 depth 的最大值；
28.                  Replace 队列中该节点的 depth 值为上一步计算的最大值；
29.    END While
```

3.1.3 算法优缺点

Fish Search 算法对待爬行的 URL 进行了优先级判断，且其原理简单，容易理解。但是该算法存在两个问题，第一个问题是只是考虑简单匹配关键词来计算相关度，导致结果准确度不够；第二个问题是使用离散点（0、0.5 和 1）来表示主题是否相关，无法全面表示页面与主题的相关性。

3.2 基于关键词位置的页面主题相关度计算算法

3.2.1 算法分析

由上一节我们了解到，Fish Search 算法存在一些问题。首先，Fish Search 算法计算页面的主题相关度时，计算的是主题关键词在页面出现的次数与主题关键词的个数的比值。该算法对页面上不同位置的关键词没有区分，忽视了关键词位置的重要性。一个网页上有多种文本，不同标签里的文本意义不同，对网页主题的重要性影响也不同，不考虑关键在网页中的位置，导致页面的主题相关度计算结果不准确。

其次，Fish Search 算法以上面计算的页面主题相关度为基础，通过一定的阈值判断网页是否主题相关，如果网页的主题相关度高于该阈值，即主题相关，其子节点 Potential-score 值就为 1，如果网页的主题相关度低于该阈值，即不相关，根据 depth 的值是否为 0，其子节点 Potential-score 值分别为 0 或者 0.5。该 Potential-score 是三个离散值，用来决定待爬取 URL 队列的优先级的话，许多 URL 都具有相同的优先级，没有完全区分网页的重要程度。因此本节对 Fish Search 算法页面的主题相关度计算方法进行了改进，提出了一种基于关键词位置的页面主题相关度的计算方法，改进的地方如图 3-2 所示。

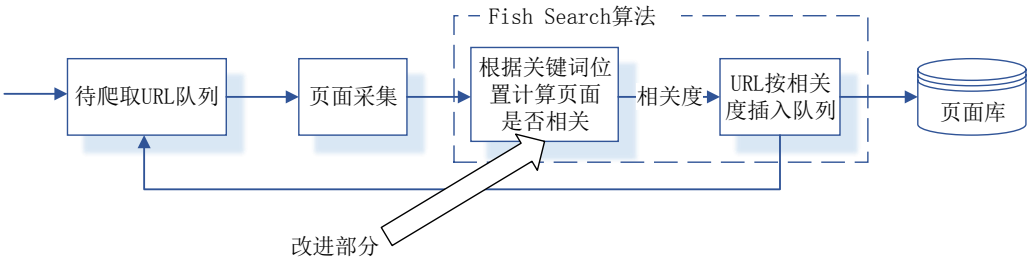

图 3-2 改进的 Fish 算法示意图

3.2.2 算法改进

准确获得一个网页的主题是计算网页与主题相关度的前提，网页的标签是语义化的，能表示网页主题的文本主要存在于页面标题 title 和关键词 keywords、页面里链接 a 的锚文本以及页面里的其他标签。

经观察分析，页面 title 标签和 keywords 标签中的文本最能代表网页的主题，其权值最高；页面里 a 标签中的文本一般是其子页面的标题，能一定程度上反映父页面的主题，其权值较小；其他正文中的文本一般只是提到而已，对页面的主题判断作用不大，不予考虑。所有针对主题关键词在网页中出现的位置的不同，赋以不同的因子，公式如下：

$$w=\begin{cases} \alpha，\ 0<\alpha\le 0.5\ (\text{关键词在网页的 a 标签中}) \\ \beta，\ 0.5<\beta<1\ \ (\text{关键词在网页的 keywords 标签中}) \\ 1，(\text{关键词在网页的 title 标签中}) \end{cases} \tag{3-1}$$

假设主题关键词出现在网页 title 标签中的权值为w_t，并根据公式（3-1）带入参数，得到w_t的计算公式如式（3-2）所示。

$$w_t=\frac{N_t}{S_t}\times w=\frac{N_t}{S_t} \tag{3-2}$$

其中，S_t是网页 title 标签文本分词后的词语个数，N_t是主题关键词在网页 title 标签中出现的次数，w 为公式（3-1）中主题关键词出现在网页 title 标签中的权重。

假设主题关键词出现在网页 keywords 标签中的权值为w_k，并根据公式（3-1）带入参数，得到w_k的计算公式如式（3-3）所示。

$$w_k=\frac{N_k}{S_k}\times w=\beta\frac{N_k}{S_k} \tag{3-3}$$

其中，S_k是网页 keywords 标签文本分词后的词语个数，N_k是主题关键词在网页 keywords 标签中出现的次数，w 为公式（3-1）中主题关键词出现在网页 keywords 标签中的权重。

假设主题关键词出现在网页 a 标签中的权值为w_l ，并根据公式（3-1）带入参数，得到w_l的计算公式如式（3-4）所示。

$$w_l = \frac{N_l}{S_l} \times w = \alpha \frac{N_l}{S_l} \tag{3-4}$$

其中，S_l是网页 a 标签文本分词后的词语个数，N_l是主题关键词在网页 a 标签中出现的次数，w 为公式(3-1)中主题关键词出现在网页 a 标签中的权重。

页面的主题相关度就是主题关键词出现在页面标题、页面关键词和页面锚文本的各项权值之和，用 *sim* 表示页面的相关度值，因此可得出页面的权值 *sim* 的公式如下。

$$sim = W_t + W_k + W_l \tag{3-5}$$

将公式（3-2）、（3-3）、（3-4）的带入公式（3-5）得：

$$sim = \frac{N_t}{S_t} + \beta \frac{N_k}{S_k} + \alpha \frac{N_l}{S_l} \tag{3-6}$$

由公式（3-6）观察得 *sim* 的值在 0 到$1 + \alpha + \beta$之间，是连续的。该方法中计算页面的主题相关度时，使用了页面 title 标签、keywords 标签和 a 标签中的文本判断页面的主题，并赋以不同的比例因子，不仅提高了计算速度，而且可以更加准确地代表页面的主题相关度；并且该方法使用 *sim* 代表页面的主题相关度值，不使用原始 Fish Search 算法的 0、0.5 和 1 这三个离散值，改进了 Fish Search 算法待爬取 URL 的 URL 优先级区分度太小的问题，因此可以通过 sim 大小评价网页与主题的相关程度。

3.2.3 算法实现

基于关键词位置的页面主题相关度计算算法的伪代码如表 3-2 所示。

表 3-2 基于关键词位置的页面主题相关度计算算法的伪代码

1.	Set 对标题进行分词，对关键词进行分词，对锚文本进行分词，初始相关度 = 0，初始链接个数= 0，权值因子α、β；

```
2.  For 标题词组
3.      If 当前的词组与主题相关
4.          Then Set 相关度加 1，链接个数加 1；
5.      Else
6.          Then Set 链接个数加 1；
7.  Set i 等于相关度除以链接个数；
8.  Set 链接个数 = 0，相关度 = 0；
9.  For 关键词词组
10.     If 当前的词组与主题相关
11.         Then Set 相关度加 α，链接个数加 1；
12.     Else
13.         Then Set 链接个数加 1；
14. Set j 等于相关度除以链接个数；
15. Set 链接个数= 0，相关度等于 0；
16. For 锚文本词组
17.     If 当前的词组与主题相关
18.         Then Set 相关度加 β，size 加 1；
19.     Else
20.         Then Set 链接个数加 1；
21. Set k 等于相关度除以链接个数；
22. Set 当前网页的相关度等于 i+j+k；
```

基于关键词位置的 Fish Search 算法的完整伪代码如表 3-3 所示。

表 3-3 基于关键词位置的 Fish Search 算法伪代码

```
1.  Get 初始 URL 节点，width（width），size（S），depth（D），查询关键词，时间限
    制；
2.  Set 初始 URL 节点的深度，depth = D，并将初始 URL 插入空的待爬取队列中；
3.  While 待爬取队列不为空 and 已处理的节点个数小于 S and 时间限制未到
```

```
4.      Pop 队列中的第一个节点为当前节点；
5.      Compute 相关度 of 当前节点；
6.      If depth < = 0 Then continue；
7.      If 当前节点的相关度 < 0.5
8.              For 当前节点的前 width 个子节点
9.                  Set 子节点的 potential-score = 当前节点的相关度；
10.             For 当前节点的其余子节点
11.                 Set 子节点的 potential-score = 0；
12.     Else
13.             For 当前节点的前 a*width 个子节点（a 通常为 1.5）
14.                 Set 子节点的 potential-score = 当前节点的相关度；
15.             For 当前节点的其余子节点
16.                 Set 子节点的 potential-score = 0；
17.     For 当前节点的每个子节点
18.         If 子节点已经存在于优先级队列中
19.          Then Compute 队列中该节点与子节点 potential-score 的最大值；
20.                 Replace 队列中该节点的 potential-score 值为上一步计算的最大值；
21.                 Insert 子节点到排好序的队列中正确的；
22.          Else 将子节点按照它的 potential-score 值插入到排好序的队列中；
23.     For 当前节点的每个子节点
24.          If 当前节点的相关度 > 0.5
25.            Then Set 子节点的深度 depth = depth（当前节点）；
26.          Else 子节点的深度 depth = depth（当前节点） – 1；
27.          If 子节点已经存在于优先级队列中
28.          Then Compute 队列中该节点与当前子节点的 depth 的最大值；
29.                 Replace 队列中该节点的 depth 值为上一步计算的最大值；
30. END While
```

3.3 本章小结

本章针对爬行策略中第一个关键问题，如何计算页面的主题相关度进行了研究，首先研究了基于网页内容的爬行策略常用算法 Fish Search，介绍了 Fish Search 算法的基本原理、算法步骤和优缺点；然后针对其计算页面主题相关的的不足，改进了其页面主题相关度的计算方法，提出了一种基于关键词位置的页面主题相关度计算算法；最后实现了该算法以及基于关键词位置的 Fish Search 算法。

4 链接优先级计算

主题爬虫的目的是爬取到更多与主题相关的页面，减少爬取到与主题不相关的页面的可能性。爬行过程中，希望优先选择与主题最相关或者重要度最高的网页进行爬行，并丢弃与主题不相关或者不重要的网页。爬行策略的第二个关键问题就是如何计算待爬取 URL 的优先级，保证一直爬取到主题相关度高的页面。本章针对如何计算待爬取 URL 的优先级的关键问题进行了研究，首先，对基于网页链接的爬行策略的代表算法 Page Rank 算法，进行了详细的分析和研究，并针对 Page Rank 算法通过链接重要性计算待爬取 URL 的优先级造成的主题漂移问题，结合上一节提出的基于关键词位置的 Fish Search 算法，提出了一种基于页面主题的 Page Rank 算法。

4.1 Page Rank 算法

Page Rank 算法是一种基于网页链接的爬行策略的经典算法，本节会对该算法进行研究和分析。

4.1.1 算法基本思想

PageRank 算法由 S.Brin 和 L.Page 发明，S.Brin 和 L.Page 利用网络的超链接结构给所有的页面定义了一个 Page Rank 值，用 Page Rank 值的高低代表页面的重要性，以此确定爬行的页面顺序。Page Rank 算法的原理是将学术论文的引用分析方法应用到网页链接中[36]。一篇论文的重要性是可以根据其被引用的次数来判断，同样一个网页的重要性也可以根据其被链接的次数来判断。具体来讲，如果网页 A 链接到网页 B，相当于网页 B 的被链接次数（Page Rank 值）增加，同时网页 A 的 Page Rank 值减小，最后由每个网页的 Page Rank 值来评估链接的重要程度。

Page Rank 算法最开始是运用于 Google 搜索引擎搜索结果的排序，最近较多应用于网页链接重要性的评估中[37-39]。如图 4-1 所示，爬虫采集页面后，首先通过计算该链接的 Page Rank 值判断页面的重要性，并根据该链接子链接的个数计算其子链接的 Page Rank 值并将其子链接插入排好序的待爬取 URL 队

列中正确的位置；然后爬虫优先选择 Page Rank 值最高的 URL 进行爬取，再次计算其 Page Rank 值，一直循环下去，直到队列为空或者达到结束条件。该算法中，由于初始网络拓扑结构还不完整，计算的 Page Rank 值并不准确，但随着爬取的页面数越来越多，网页的拓扑结构越来越完整，计算的 Page Rank 值会越来越准确。

图 4-1 Page Rank 算法简单流程图

4.1.2 算法过程

Page Rank 计算公式如下：

$$R(u) = (1-c) + c\sum_{v \in B_u} \frac{R(v)}{N_v} \tag{4-1}$$

其中，$R(u)$代表网页 u 的重要度，$R(v)$代表指向网页 u 的页面的重要度，N_v代表指向页面 u 的超链接数，B_u为指向 u 的网页的集合，c 为常数，也成为阻尼系数，指的是用户到达某一页面后向后浏览的概率，一般 c 取 0.85。经观察公式，可以看出影响页面 u 的重要度的因素有以下几点：

(1) 页面 u 的入度，如果链接到 u 的网页数量越多，$R(u)$越大。

(2) 链接到页面 u 的网页重要度，链接到 u 的网页重要度越高，$R(u)$越大。

(3) 链接到 u 的网页的出链数量，链接到 u 的页面的出度越小，平均传递给 u 的重要度值就越大，$R(u)$越大。

本文中，Page Rank 算法计算链接重要性时维护了两个队列：待爬取 URL 队列和已爬取 URL 队列。每次从待爬取 URL 队列中取 Page Rank 值最高的 URL 进行爬取，并在已爬取队列中查找其父链接，重新计算其链接的 Page Rank 值；然后根据其子链接的个数，计算其子链接的 Page Rank 值，并赋值给当前链接；

最后将当前链接放入已爬取 URL 队列，将其子链接插入待爬取 URL 队列中，爬虫继续从待爬取 URL 队列中选择 Page Rank 值最高的 URL 进行爬取，一直循环下去，直到到达结束条件。伪代码如表 4-1 所示：

表 4-1 Page Rank 算法伪代码

```
1.  Get 初始 URL 节点，并将初始 URL 插入空的待爬取队列中，结束条件 S，初始已爬
    取队列为空，初始页面的重要度为 1；
2.  Then 将初始页面加入已爬取队列
3.  While 待爬取队列不为空 and 已处理的节点个数小于 S
4.      Pop 队列中的第一个节点为当前节点；
5.      Set 当前节点的子链接数等于 size，当前节点的重要度等于 0；
6.      For 已爬取队列的每一个节点
7.          If 该节点是当前节点的父链接
8.          Then 当前节点的重要度增加该节点的重要度
9.      Set 当前节点的重要度等于上面计算的值除以 size，将当前链接加入已爬取队列
10.     For 当前节点的每个子节点
11.         If 子节点已经存在于待爬取 URL 优先级队列中
12.         Then Compute 队列中该节点与子节点的重要度的最大值；
13.                 Replace 队列中该节点的重要度值为上一步计算的值；
14.                 Insert 子节点到排好序的待爬取 URL 队列中正确的位置；
15.         Else 将子节点按照当前节点的重要度值插入到排好序的队列中；
16. END While
```

4.1.3 算法优缺点

Page Rank 算法通过网页链接的 Page Rank 值的高低代表网页的重要性程度，用于网页排序，成功运用于 Google 搜索引擎，证明了其有效性。但如果通过网页链接的 Page Rank 值的高低决定待爬取 URL 队列中 URL 的爬行顺序，这样只考虑了网页链接的重要度，没有考虑到网页内容与主题的相关性，如果运用到主题爬虫中，易造成“主题漂移”现象，导致爬取到许多与主题不相关

的页面。但是如果将基于网页链接和网页内容的爬行策略结合起来计算待爬取 URL 的优先级，便能弥补各自的不足，下一节将详细介绍此改进。

4.2 基于页面主题的 Page Rank 算法

4.2.1 算法分析

根据上一节的分析，基于网页链接的 Page Rank 算法的关键问题是“主题漂移”，会爬取大量主题无关的页面，是基于网页链接的爬行策略的通病；并且，在爬虫初期由于爬取的网页数量少，基于网页链接的 Page Rank 算法，无法确定比较完整的网络拓扑结构，会导致算不出准确的 Page Rank 值，不能准确地计算待爬取 URL 的优先级。如果将基于网页内容的 Fish Search 算法与基于网页链接的 Page Rank 算法相结合，能够一定程度上避免这个问题。

根据上一章的分析，基于关键词位置的 Fish Search 算法度量页面的主题相关度时，使用了基于关键词位置的主题相关度计算算法，考虑到了主题关键词在网页中的位置不同对主题相关度的影响不同，改进了其计算页面主题相关度的方法，并最后得到一个连续的主题相关度度量公式，对比原始的 Fish Search 显然要好些。但是，基于关键词位置的 Fish Search 算法在计算待爬取 URL 的优先级时，没有考虑网络拓扑结构的影响，不能发现高重要度的网页。如果结合基于网页链接的爬行策略的相关算法，能更加准确地计算待爬取 URL 的优先级，解决这个问题。

因此，针对改进后 Fish Search 算法和 Page Rank 算法计算待爬取 URL 优先级的不足，本节将 Page Rank 算法与基于关键词位置的 Fish Search 算法结合，即将基于网页内容的爬虫策略与基于网页链接的爬虫策略结合起来，设计出基于页面主题的 Page Rank 算法。该算法在计算待爬取 URL 优先级时，不仅根据 Page Rank 算法计算其链接的重要度，还根据基于关键词位置的 Fish Search 算法计算其页面内容的主题相关度，最后将这两个值综合起来计算待爬取 URL 的优先级。其简单流程图如 4-2 所示。

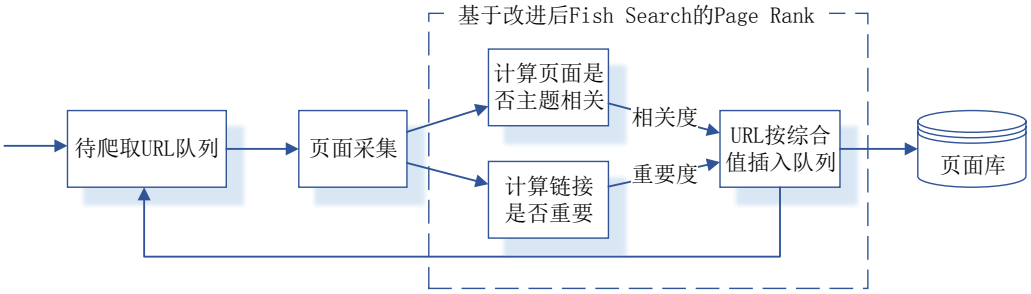

图 4-2 基于页面主题的 Page Rank 算法简单流程图

4.2.2 算法改进

该算法由基于关键词位置的 Fish Search 算法与 Page Rank 算法组成，决定待爬取 URL 的优先级的综合权值便是这两种算法计算得到的相关度值综合加权得到。因此在对待爬行 URL 优先级排序时，除了考虑 Page Rank 算法计算的网页链接的重要度值外，还加上了基于改进后 Fish Search 算法计算的网页内容的主题相关度。算法中考虑到无法根据整个互联网拓扑图计算链接的 Page Rank 值，只计算了从初始链接开始后的局部链接的 Page Rank 值。

设待爬取 URL 的综合权值为W_{PF}，其计算公式如下：

$$W_{PF} = \gamma \times R + (1-\gamma) \times sim(0 < \gamma < 1) \qquad (4\text{-}2)$$

其中， R 是公式（4-1）计算出来的页面链接的 Page Rank 值，sim 是公式（3-6）计算出来的页面内容的主题相关度值，权值γ为预设的比例因子。

因此，基于页面主题的 Page Rank 算法的基本思路是：首先爬取根节点的页面，并提取其子链接。采用基于关键词位置的 Fish Search 算法计算根节点对应页面的主题相关度，同时，采用 Page Rank 算法计算根节点的网页重要度，即其根节点的 Page Rank 值。根据两个相关度值以及预设的比例因子计算综合得出根节点的重要性等级。如果该重要性等级高于某个阈值，就将此重要性等级赋值给其子链接，这样其子链接就有一个来自父链接节点的初始重要性等级，接着结合得到的重要性等级，也就是主题相关性和重要性等级各自乘以预设比例因子综合排序得到的得分大小插入到待爬取 URL 队列中。接着，爬虫选择得分最高的子链接进行爬取，一直循环下去。

该算法在对页面进行爬取时，子链接的排序依据于父节点的初始重要性等

级，包括其网页链接的权威值和其网页页面的主题相关度，即通过 Page Rank 算法和 Fish Search 算法进行计算得出其子链接的初始重要性等级。用通俗的语言来讲就是在广度优先爬取的时候，对每一层的链接都进行重要性评价，取某一个阈值以上的重要性链接进行爬取，这样可以放弃一部分无用链接，避免爬取无关的页面，同时每一层都进行筛选，以保证整个爬取的过程都是直往主题，是一个圆柱形的通道，不会扩散。

4.2.3 算法实现

算法的步骤如下，具体流程如图 4-3 所示：

(1) 初始化 URL 列表和主题关键词；

(2) 提取初始 URL 的子链接；

(3) 根据公式 2 的综合权值计算方法计算其子链接的综合权值，并跟预先设置的阈值进行比较。高于阈值的链接按照综合权值大小插入到待爬取 URL 队列，低于阈值的链接则直接丢弃；

(4) 一直循环下去直到队列没有 URL 或者达到结束条件。

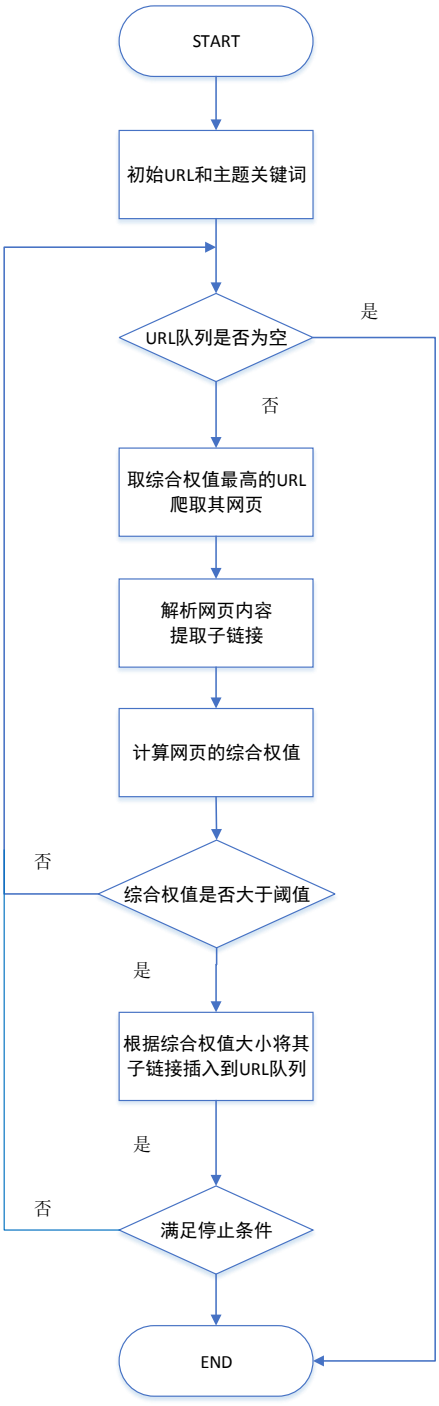

图 4-3 基于页面主题的 Page Rank 算法具体流程图

基于页面主题的 Page Rank 算法伪代码如表 4-2 所示。

表 4-2 基于页面主题的 Page Rank 算法伪代码

```
1.  Get 初始 URL 节点，depth（D），结束条件 S，初始待爬取队列、已爬取队列为空，
    初始 URL 页面的重要度为 1；
2.  Set 初始 URL 节点的深度 depth = D，并将初始 URL 插入空的待爬取队列和已爬取队
    列中；
3.  While 待爬取队列不为空 and 已处理的节点个数小于 S
4.   Pop 队列中的第一个节点为当前节点；
5.   If depth <= 0 Then Continue；
6.   Set 当前节点的子链接数等于 size，当前节点的重要度等于 0；
7.   For 已爬取队列的每一个节点
8.     If 该节点是当前节点的父链接
9.     Then 当前节点的重要度增加该节点的重要度；
10.  Set 当前节点的重要度等于上面计算的值除以 size，将当前链接加入已爬取队列；
11.  Compute 相关度 of 当前节点；
12.  Compute 根据当前节点的重要度和相关度计算出综合相关度值；
13.  If 当前节点的综合相关度值 < 0.5
14.     For 当前节点的前 width 个子节点
15.       Set 子节点的综合相关度值 = 当前节点的综合相关度值；
16.     For 当前节点的其余子节点
17.       Set 子节点的综合相关度值 = 0；
18.  Else
19.     For 当前节点的前 a*width 个子节点（a 通常为 1.5）
20.       Set 子节点的综合相关度值 = 当前节点的综合相关度值；
21.     For 当前节点的其余子节点
22.       Set 子节点的综合相关度值 = 0；
23.  For 当前节点的每个子节点
24.     If 子节点已经存在于优先级队列中
25.     Then Compute 队列中该节点与子节点的综合相关性的最大值；
```

```
26.                 Replace 队列中该节点的综合相关性值为上一步计算的值；
27.                 Insert  子节点到排好序的待爬取 URL 队列中的正确的位置中；
28.         Else 将子节点按照当前页面的综合相关性值插入到排好序的队列中；
29.     For 当前节点的每个子节点
30.         If 当前节点的相关度 > 0.5
31.             Then Set 子节点的深度 depth = depth（当前节点）；
32.         Else 子节点的深度 depth = depth（当前节点） – 1；
33.         If 子节点已经存在于优先级队列中
34.             Then Compute 队列中该节点与当前子节点的 depth 的最大值；
35.                 Replace 队列中该节点的 depth 值为上一步计算的最大值；
36.   END While
```

4.3 本章小结

本章对爬虫爬行策略的第二个关键问题，如何计算待爬取 URL 的优先级进行了研究。首先研究了基于网页链接的爬行策略的常用算法 Page Rank，介绍了 Page Rank 算法的基本原理、算法步骤和优缺点；然后针对 Page Rank 算法的不足，考虑了结合基于页面内容的基于关键词位置的 Fish Search 算法，提出了一种基于页面主题的 Page Rank 算法，该算法既考虑了网页内容的主题相关度，又考虑到了网页链接的重要性，弥补了两种算法的不足；最后，实现基于页面主题的 Page Rank 算法。

5 实验结果及分析

本章设计了一个通用爬虫框架，根据第三章和第四章提出的改进的算法，分别实现基于 Fish Search、基于关键词位置的 Fish Search、 Page Rank 以及基于页面主题的 Page Rank 算法的主题爬虫。并分别对 Fish Search、基于关键词位置的 Fish Search 、Page Rank 以及基于页面主题的 Page Rank 算法的实验结果进行分析并得出结论。目标是：改进后的算法能在较短的时间内，爬取大量与主题相关的页面，并且系统能稳定运行。

5.1 实验环境

实验使用 Java 编程语言实现了一个简易的通用爬虫框架，其软硬件环境如表 5-1 所示：

表 5-1 软硬件环境表

名称	配置
CPU	Intel® Pentium® Dual CPU E2200 @2.20GHz
内存	4GB
操作系统	Windows 7
开发工具	MyEclipse、MySQL

下面是实验的主要步骤：

（1）输入初始 URL 和主题关键词。本实验采用新浪网体育类目的主页作为初始 URL，主题关键词为“足球”。

（2）选择不同算法开始爬取。系统下载初始 URL 对应的页面，抓取页面上的链接，并选择优先级高的链接爬行。

（3）爬取下来的页面保存在 MySQL 数据库。

5.2 评价指标

为了很好的评价改进后算法的效果，引入了查准率[40]（Precision）、查全率

[40]（Recall）两个性能指标。查准率指的是爬取到的主题相关的页面数与爬取到的页面总数之比。查全率指的是爬取到的主题相关的页面数与实际主题相关的页面数之比。查准率和查全率是评价主题爬虫的主要指标，

其计算公式如下：

$$Precision = {}^{k}/_{n} \tag{5-1}$$

$$Recall = {}^{k}/_{r} \tag{5-2}$$

其中，k 为爬取到的主题相关的页面数，n 为爬取到的页面总数，r 为实际主题相关的页面。由于互联网上的数据十分巨大，想要统计整个互联网上的主题相关网页的数量非常困难，我们没法知道具体的主题相关的页面数量。所以本文实验不用查全率而是查准率来评价爬虫的性能。

本文还考虑到了主题爬虫的爬取效率，如果一个主题爬虫爬取的主题相关页面数很多，但是消耗了大量的爬取时间，那么这个算法也不被认为是一个有价值的算法。因此，本文综合考虑了主题爬虫爬取到的主题相关页面数和爬取这些页面所用的时间，引入了算法价值（value）的性能指标，即单位时间爬取到的主题相关的页面数。

其计算公式如下：

$$value = \frac{k}{T} \tag{5-3}$$

其中，k 为爬取到的主题相关的页面数，T 为爬取页面所用的时间。

本节从查准率和算法的价值两个不同的角度，以原始 Fish Search 算法、基于关键词位置的 Fish Search 算法、 Page Rank 算法和基于页面主题的 Page Rank 算法的爬行结果对比，来综合评测本文提出的主题爬虫算法。

5.3 实验设计及数据

实验采用新浪网体育类目主页和关键词足球，分别使用基于原始的 Fish Search 算法、基于关键词位置的 Fish Search 算法、原始 Page Rank 算法以及基于页面主题的 Page Rank 算法的主题爬虫进行实验。每爬取 100 个页面，统计与主题相关的页面个数、爬取时间，累计爬取 10 轮，共 1000 个页面，计算每次抓取到的主题相关的页面个数与爬取页面总数的比值，即查准率。以及每次抓取到的主题相关的页面个数与爬取时间的加权值，即算法价值。

其中，实验中使用基于关键词位置的的 Fish Search 算法以及基于页面主题的 Page Rank 算法时，将爬取下来的页面的标题、关键词和锚文本分词，分别设置标题权重w_t 的因子为 1，关键词权重w_k 的因子为 0.8，锚文本权重w_l 的因子 0.3，综合权值的阈值在实验中进行调整。

5.4 实验结果及分析

5.4.1 查准率结果分析

爬虫从初始 URL 开始抓取“足球”有关的页面，计算得到基于原始的 Fish Search 算法、基于关键词位置的 Fish Search 算法、原始 Page Rank 算法以及基于页面主题的 Page Rank 算法的主题爬虫的查准率结果如曲线图 5-1、5-2 所示，平均查准率如表 5-2 所示。

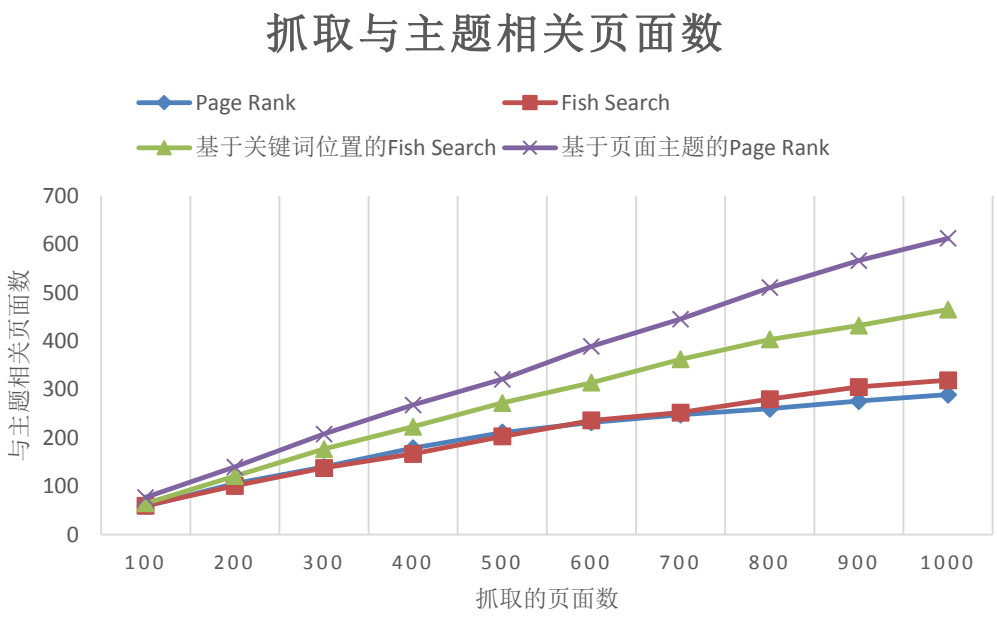

图 5-1 抓取的与主题相关页面数对比图

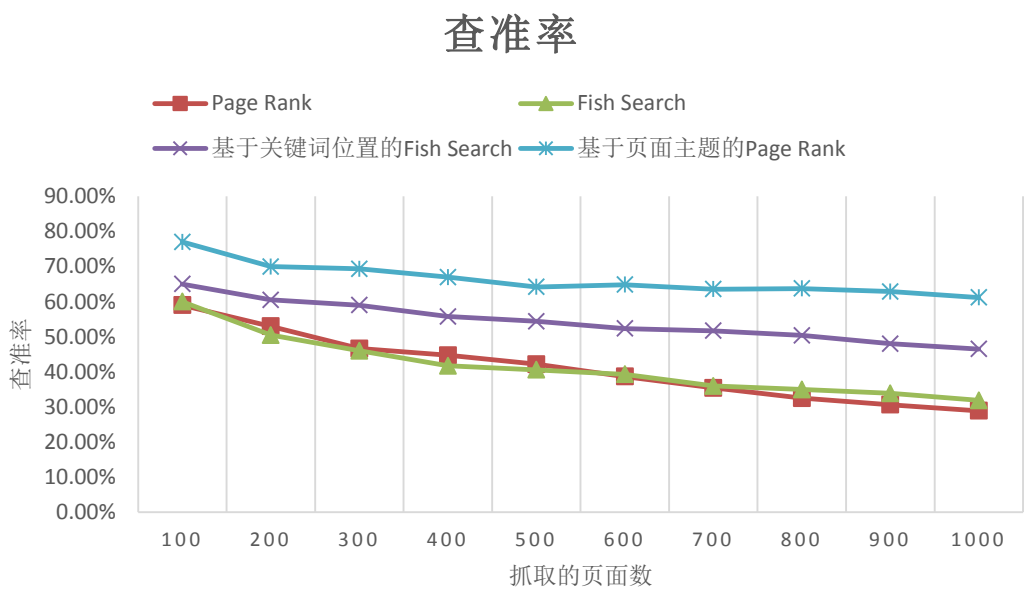

图 5-2 查准率结果对比图

表 5-2 平均查准率结果表

算法	平均查准率
Fish Search	41. 50%
基于关键词位置的 Fish Search	54. 36%
Page Rank	41. 18%
基于页面主题的 Page Rank	66. 39%

通过图 5-1、5-2 和表 5-2 中基于原始的 Fish Search 算法、基于关键词位置的 Fish Search 算法、原始 Page Rank 算法以及基于页面主题的 Page Rank 算法实验结果数据对比可以看出：

(1) 基于关键词位置的 Fish Search 算法与原始的 Fish Search 算法相比，能抓取更多主题相关的网页，具有更高的查准率。随着网页爬取数量的增加，基于关键词位置的 Fish Search 算法和原始 Fish Search 算法的查准率都有一定比例的下降，但基于关键词位置的 Fish Search 的算法的查准率总是高于原始的 Fish Search 算法的查准率，这是因为基于关键词位置的 Fish Search 算法使用了

基于关键词位置的页面主题相关度计算算法，该算法使用一个连续值表示页面的主题相关度，避免了原始 Fish Search 算法中过度剪枝的问题。因此，基于关键词位置的页面主题相关度计算算法能够更加准确地计算页面的主题相关度，提高主题爬虫的查准率。

(2) 基于页面主题的 Page Rank 算法与基于关键词位置的 Fish Search 算法、原始 Page Rank 算法相比，基于 Page Rank 算法的主题爬虫开始爬虫效果较好，但是随着网页爬取数量的增多，其查准率直线下降，这是因为 Page Rank 只考虑了网页链接的重要度而忽视了网页内容的重要性，导致发生了主题漂移现象；基于基于关键词位置的 Fish Search 算法不仅考虑了页面内容，还考虑了页面关键词由于位置不同对网页重要性的影响也会不同，整个曲线较稳定；而基于页面主题的 Page Rank 算法的主题爬虫查准率总是高于其他两种算法，这是因为基于页面主题的 Page Rank 算法的主题爬虫计算待爬取 URL 优先级时既考虑了网页链接的重要性，又考虑了网页页面内容与主题的相关度，正好弥补了以上两种算法的不足，因此，基于页面主题的 Page Rank 算法能够更加准确地计算待爬取 URL 的优先级，提高主题爬虫的查准率。

5.4.2 算法价值结果分析

爬虫从初始 URL 开始抓取“足球”有关的页面，计算得到基于原始的 Fish Search 算法、基于关键词位置的 Fish Search 算法、原始 Page Rank 算法以及基于页面主题的 Page Rank 算法的主题爬虫的算法价值结果如表 5-3 所示。

表 5-3 算法价值结果表

算法	主题相关页面数	爬取时间（秒）	算法价值
Fish Search	319	760	0.42
基于关键词位置的 Fish Search	465	710	0.65
Page Rank	289	790	0.37
基于页面主题的 Page Rank	612	900	0.68

通过表 5-3 中基于原始的 Fish Search 算法、基于关键词位置的 Fish Search

算法、原始 Page Rank 算法以及基于页面主题的 Page Rank 算法的算法价值计算结果对比可以看出：

(1) 在算法价值方面，排名由低到高分别为原始 Page Rank 算法、原始 Fish Search 算法、 基于关键词位置的 Fish Search 算法、基于页面主题的 Page Rank 算法。原始的 Page Rank 算法之所以算法价值最低，是因为该算法需要大量的时间计算页面的 Page Rank 值，并且由于 Page Rank 的主题漂移现象，爬取的主题相关的页面数最少，因此在实际应用时一般考虑结合其他算法运用于主题爬虫。

(2) 基于关键词位置的 Fish Search 算法与原始 Fish Search 算法相比，其算法价值要高一些，这是因为基于关键词位置的 Fish Search 使用了基于关键词位置的页面主题相关度计算算法，该算法计算页面相关度时，只考虑了页面标题、关键词和锚文本，而不是像原始 Fish Search 算法那样匹配全文的关键词。不仅爬取到了更多主题相关的页面，还提高了爬行效率，因此基于关键词位置的 Fish Search 算法与原始 Fish Search 算法相比爬行结果提高不少。

(3) 基于页面主题的 Page Rank 算法的价值最高，这是因为基于页面主题的 Page Rank 算法计算待爬取 URL 的优先级时不仅考虑了网页的拓扑结构，还考虑了网页页面内容的主题相关度，爬取到了更多的主题相关的页面，虽然其也耗费了不少的爬取时间，但综合爬取的主题相关的页面数来看，其算法的价值最高。

5.5 本章小结

本章首先介绍了爬虫实验的开发环境、设计目标、评价指标和实验步骤，接着分别以对第四章提出的算法进行实验，最后通过图表的方式对主题爬虫的爬行结果进行对比，并对实验结果进行了分析。

6 及时推新闻推送系统的实现

6.1 项目背景

随着时代的进步和计算机技术的迅速发展，以互联网为特征的第四媒体给传统的新闻媒体带来冲击和挑战，伴随着互联网信息时代的带来，信息数量高速增长，囊括了政治、经济、文化等多方面的大量的资讯。并且互联网信息更新速度飞快，在浩如烟海的网络资源里，想要及时的获取想要的信息变得越来越难。

针对上述问题，有以下几种解决办法：

(1) 使用搜索引擎，目前的搜索引擎都是面向网络上所有的信息的。但是，随着信息的增长和全面化，针对受限领域和特定主题的搜索的准确率和查全率都是很低的。

(2) 使用新闻网站及其客户端，用户可以设置自己感兴趣的主题，阅读自己感兴趣的内容。但是，新闻网站或者新闻客户端上的新闻大部分是用户不关心的，由于不能定制关注的网页，不能推送用户真正需要的信息。

综上，我们知道，用户真正需要的是一个完全定制的信息推送系统。在这个系统里，通过用户定制的关注的网站和关键词，将每天推送该网站更新的包含该关键词的页面给用户。

该系统的实质就是一个主题爬虫搜索引擎，该系统的核心和技术就是爬虫子系统，即如何准确全面地爬取用户定制的感兴趣的针对特定主题和领域的网页，因此将本章将本文设计并实现的主题爬虫算法应用到该系统中，为系统提供用户定制的感兴趣的网页。

6.2 系统概述

及时推新闻推送系统是四川省成果转换项目，分别实现了 PC 端、Android 客户端和 IOS 客户端，并已全部成功上线。该系统主要由 3 个子系统组成，其网络拓扑图如图 6-1 所示：

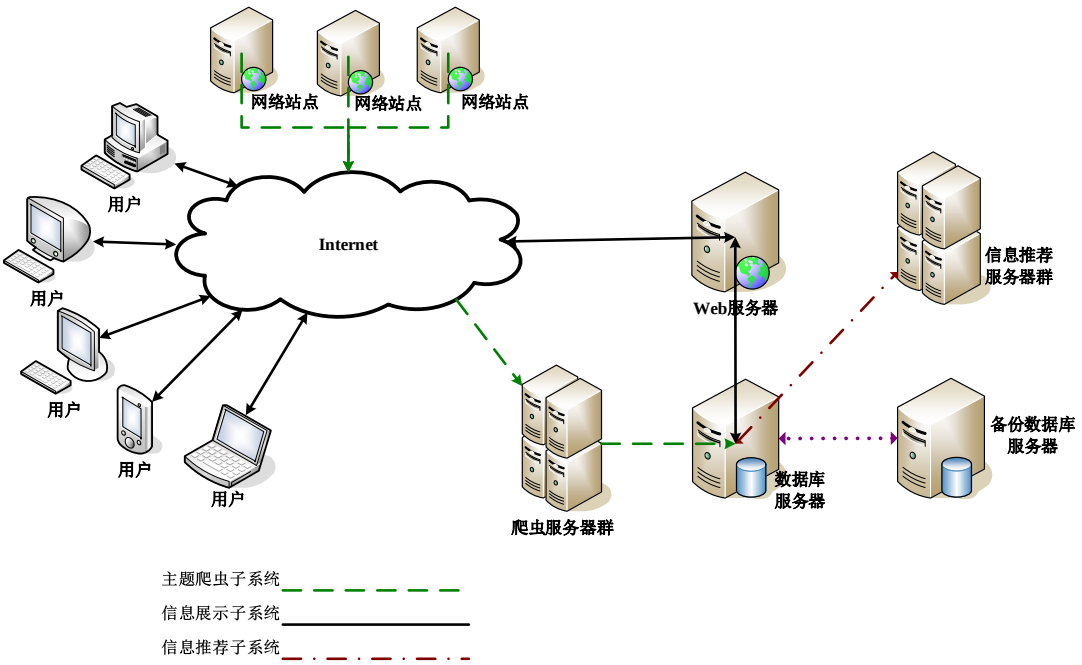

图 6-1 系统软件架构图

(1) 爬虫子系统：根据用户定制的感兴趣的网页和关键词，爬取该网页上每天更新的包含关键词的所有 URL 存入数据库。该系统是个分布式的爬虫系统，根据用户所关注的网站内容，将所需爬取链接和关键词分发到各个爬虫客户端，并且客户端的的类型可以是任意平台（Windows，Linux 等），爬虫客户端收到任务后爬取页面并分析出需要的数据返回给服务器并存入数据库。

(2) 信息推荐子系统：通过分析服务器日志，获取用户数据，然后将这些数据用于训练受限玻尔兹曼机，得到训练好的权值矩阵以及其他相关参数之后，给用户提供相关的新闻推荐服务，每天为用户推送信息到 PC 网页客户端、Android 客户端、IOS 客户端以及手机或邮箱。

(3) 信息展示子系统：通过 PC 网页客户端、Android 客户端和 IOS 客户端显示系统给用户推送的信息，方便用户随时随地查看。

整个过程中，首先，爬虫子系统为信息推送子系统和信息展示子系统提供爬取到的用户定制的网页信息，接着，信息推送子系统根据特定规则给用户提供相关的新闻推荐服务，信息展示子系统查询数据库显示用户定制的网页信息。

因此，爬虫子系统是整个系统的基础，也是关键的一个步骤，如果爬取到的网页信息的准确率和覆盖率不够，那么后面的推荐和展示就没有办法实现。

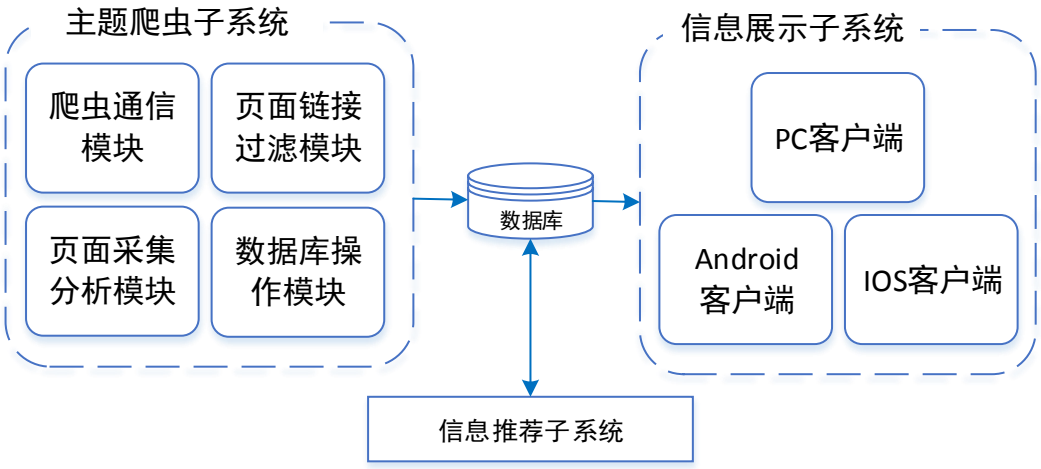

图 6-2 系统软件架构图

从图 6-1 可以看出，系统使用了多台服务器，包括主题爬虫服务器群、推荐系统服务器群、web 服务器、数据库服务器和备份数据库服务器。为了充分使用服务器的资源进行主题爬虫，爬虫子系统将用于主题爬虫的服务器一台作为爬虫服务器端，其他用于爬虫客户端，分别装 linux 操作系统，各个爬虫客户端接收爬虫任务爬取页面后将数据返回给爬虫服务器。Web 服务器采用开源的 Tomcat 服务器，用于给用户推送网页信息，并展示在各个客户端上。数据库服务器则是用于存储爬虫子系统爬取的网页信息数据。

从图 6-2 中可以看出，系统的几个关键点主要是主题相关网页信息爬取、网页信息推荐和网页信息展示。主题相关网页信息爬取是最基础关键的部分，主要包括爬虫通信模块、页面采集分析模块、页面链接过滤模块和数据库操作模块。主题相关网页信息爬取模块将爬取的主题网页信息存入数据库中，网页信息推荐模块从数据库中获取所有的网页信息进行分析后得到信息推荐给用户，网页信息展示模块就直观地显示用户定制的感兴趣的网站的信息。

以上三个子系统中我主要负责的是爬虫子系统的实现以及各个客户端展示子系统的前台界面实现。

6.3 爬虫子系统实现

爬虫子系统包括一个爬虫服务器端和多个爬虫客户端，采用了如图 6-3 所示的体系结构。爬虫服务器端由通信模块和数据库操作模块组成，爬虫客户端都由爬虫模块和通信模块组成。其中通信模块负责爬虫客户端与服务端的通信；数据库操作模块负责从数据库待爬取链接和主题关键词，并将爬行结果保存到数据库；爬虫模块是整个爬虫子系统的核心部分，主要负责完成爬虫服务端发送过来的爬取任务主要负责主题相关网页的爬取，分为页面采集分析模块和页面链接过滤模块。

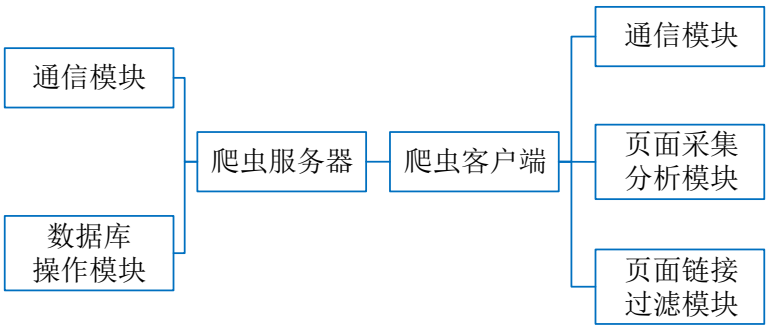

图 6-3 爬虫子系统体系结构图

6.3.1 爬虫服务器端实现

爬虫服务器端的程序以 main 函数作为入口地址，通过 RabbitMQ 消息队列实现与爬虫客户端的通信。具体流程如下：

(1) 调用 init()方法，初始化爬虫服务器端各项配置，连接数据库；

(2) 执行 while()语句，循环地从数据库读取待爬取的链接和主题关键词；

(3) 调用 createChannel ()方法，创建并启动两个 RabbitMQ 通道。一个通道用于接收爬虫任务，一个通道用于返回爬行结果；

(4) 通过 RabbitMQ 通道将待爬取的链接和关键词发送给各个爬虫客户端；

(5) 通过 RabbitMQ 通道提取爬虫客户端返回的结果并插入到数据库中。

6.3.1.1 服务器端通信模块

该模块负责爬虫服务器端与爬虫客户端之间的通信，维护了两个RabbitMQ 通道，一个是接收任务的 TASK_QUEUE 通道，另一个是返回爬行结果的 TODO_QUEUE 通道。该模块一边将待爬取的链接和主题关键词放入TASK_QUEUE 通道中，一边监听 TODO_QUEUE 通道，一旦 TODO_QUEUE 通道有数据，就将通道中的数据取出来，以供后续模块使用。该模块主要通过以下两个方法实现爬虫服务器端与爬虫客户端的通信：

- sendTask( )将待爬取的链接和关键词放入 TASK_QUEUE 通道。
- receiveInfo( )监听 TODO_QUEUE 通道，一旦接收到爬行结果，即通道有数据时，取出数据供数据库操作模块使用。

6.3.1.2 数据库操作模块

该模块负责从数据库中读取待爬取链接和主题关键词数据，以及将爬行结果存入数据库。该模块主要有以下两个方法：

- getWebsitesAndKeywords( )从数据库中读取待爬取的链接和主题关键词，供服务器端通信模块使用。
- insertNews( )将返回的爬行结果加入到数据库中。

6.3.2 爬虫客户端实现

爬虫客户端的程序同样以 main()函数作为入口地址，当接收到爬行任务时，实例化了一个 ICrawler 对象，开始爬取网页，爬行完毕后将爬行结果返回给爬虫服务器端。具体流程如下：

(1) 调用 createChannel 方法，创建并启动两个 RabbitMQ 通道。一个通道用于接收爬虫任务，一个通道用于返回爬行结果；

(2) 调用 while 语句，等待爬虫服务器发送的爬虫任务；

(3) 当接收到爬行任务后，实例化一个 ICrawler 对象，从接收任务的RabbitMQ 通道中取出待爬取的链接和主题关键词，开始爬取网页，并将爬行结果返回给爬虫服务器。

6.3.2.1 客户端通信模块

该模块主要负责爬虫客户端与服务端之间的通信，维护了两个 RabbitMQ 通道，一个是接收任务的 TASK_QUEUE 通道,另一个是返回爬行结果的 TODO_QUEUE 通道。首先从 TASK_QUEUE 通道中取出爬行任务，爬行完成后将爬行结果放入 TODO_QUEUE 通道，等待爬虫服务器读取。

该模块主要通过以下两个方法实现客户端与服务端的通信：

- receiveTask( )监听 TASK_QUEUE 通道，一旦接收到爬行任务，即通道有数据时，按顺序取出待爬行的链接和主题关键词，供页面采集分析模块使用。
- sendInfo( )将爬行结果放入 TODO_QUEUE 通道，等待爬虫服务器读取数据。

6.3.2.2 页面采集分析模块

该模块是本文主题爬虫的核心模块之一，首先根据待爬行 URL 的优先级的高低，通过 Http 协议访问优先级得分最高的 URL 并下载其对应的页面。然后对已下载的页面进行解析，提取页面的子链接和页面内容（包括页面标题、关键词和锚文本），并对页面内容进行分词，根据本文第三章提出的基于关键词位置的 Fish Search 算法计算其主题相关度。

该模块主要通过以下三个方法实现页面的采集与分析：

- getURL( )获取待爬取的 URL。
- getPage(URL url)根据 URL 爬取其对应的页面后，提取页面的子链接和页面内容。
- calRelevance(String keywords)计算其子链接的主题相关度。

6.3.2.3 页面链接过滤模块

该模块也是本文主题爬虫的核心模块之一，主要是根据本文第四章提出的基于页面主题的 Page Rank 算法，计算其子链接的综合权值，最后将该综合权值与某个阈值比较，高于阈值的子链接就按照综合权值的大小插入到待爬取 URL 队列中，低于阈值的子链接直接丢弃。

该模块主要通过以下两个方法实现页面链接的过滤：

- calPageRank( )计算子链接的 Page Rank 值。
- calAllWeights( )计算子链接的综合权值。
- insertURL(float threshold)将大于阈值 threshold 的子链接按照综合权值大小插入到待爬取的 URL 队列中。

6.3.3 爬虫实现效果

由于爬虫子系统是在后台运行的，展示不了前台界面，本节只能展示爬虫客户端和服务端通信的日志，如图 6-4 所示。

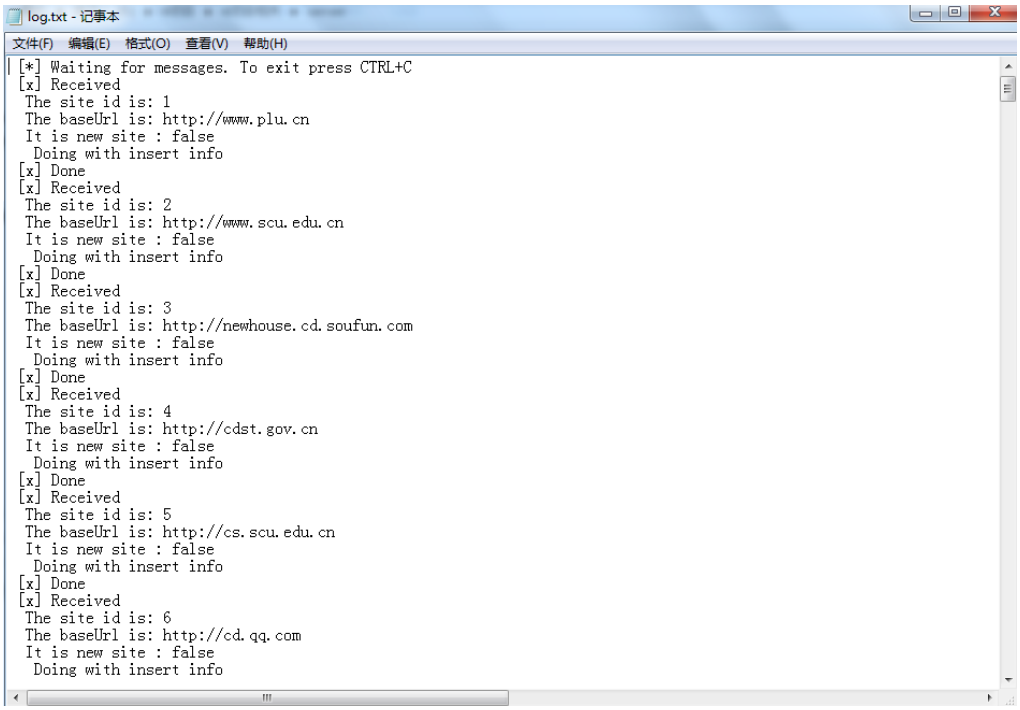

图 6-4 爬虫客户端和服务端通信的日志记录图

6.4 信息展示子系统实现

信息展示子系统主要实现了 PC 客户端、Android 客户端和 IOS 客户端的信息展示，本节分别以不同客户端展现信息展示子系统的实现效果。

6.4.1 PC 客户端

(1) 用户登录主界面如图 6-5 所示：

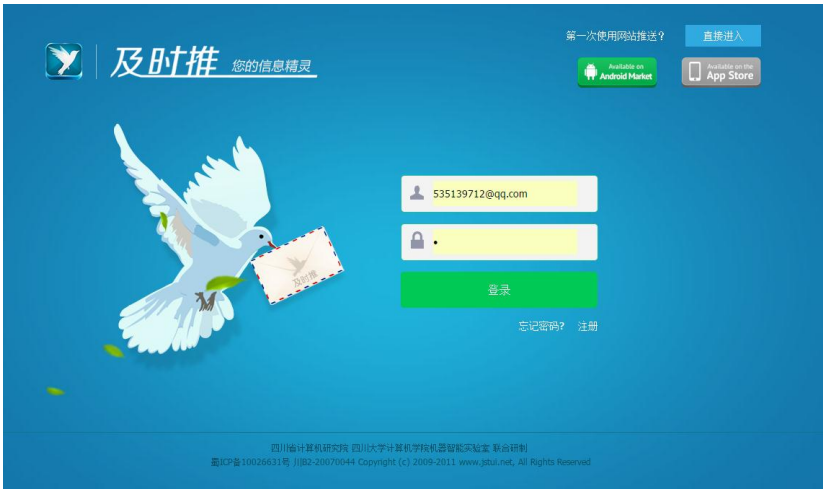

图 6-5 登录主界面图

(2) 用户登录后添加关注的网页和关键词界面如图 6-6 所示：

图 6-6 关注的网站界面图

(3) 用户登录后查看系统推送的信息界面如图 6-7 所示：

图 6-7 系统推送的信息界面图

(4) 用户登录后可以选择信息是否推荐到手机或者邮箱，如图 6-8 所示：

图 6-8 推送到手机或者邮箱设置界面图

6.4.2 移动客户端

图 6-9 移动端整体界面结构图

6.5 本章小结

本章首先对及时推新闻推送系统进行了需求分析，接着介绍了系统的软硬件结构，并重点介绍了爬虫子系统的结构和各个模块的功能；最后分别以 PC 客户端和 IOS 移动端客户端的运行情况展示了系统的实现效果。

7 总结与展望

7.1 工作总结

随着互联网信息的快速增长，要在互联网上查找对自己有用的信息越来越难，传统的搜索引擎满足不了用户针对特定领域和主题的搜索需求，人们迫切需要一个系统能每天推送他们感兴趣的信息，个性化信息推送系统应运而生。个性化信息推送系统的主要是互联网上特定主题信息的爬取，其实质就是一个面向主题的搜索引擎，而主题爬虫是面向主题的搜索引擎的核心。因此，本文对主题爬虫进行了深入的学习与研究，并将本文改进的主题爬虫算法运用到实验室项目及时推信息推送系统中。

本文首先对研究背景和国内外现状做了详细的阐述和研究，然后对当前主流的主题爬虫策略进行了介绍，继而，对主题爬虫的如何计算页面的主题相关度和如何计算待爬取 URL 的优先级两个关键问题进行了深入的研究和分析。

在如何计算页面的主题相关度方面，为了选择合适的主题相关度计算方法，本文对基于页面内容的 Fish Search 算法进行了深入的研究，由于 Fish Search 算法计算页面的主题相关度时没有考虑页面关键词位置的重要性，并且使用离散值表示页面的主题相关度，没有完全区分页面的主题相关程度的高低，因此，本文提出了一种基于关键词位置的页面主题相关度计算算法。该方法中，对页面中不同位置出现的关键词赋予不同的权值，出现在页面标题中的关键词权值最高，出现在页面关键词中的关键词权值次之，出现在锚文本中的关键词权值最低，然后综合加权后计算得到一个连续值，用这个连续的值来代表页面的主题相关度，这样改进了 Fish Search 使用离散值代表页面的主题相关度的问题，更加准确的表示了页面的主题相关度，能够爬取到更多主题相关的页面，提高爬虫的查准率。

在如何计算待爬取 URL 的优先级方面，为了选择合适的待爬取 URL 优先级计算方法，本文详细的分析和研究了基于网页链接爬行策略的代表算法 Page Rank，针对 Page Rank 算法容易出现的主题漂移问题，考虑到只用页面链接的重要性确定待爬取 URL 的优先级是不够的，如果结合网页内容的主题相关度一起来判断待爬取 URL 的优先级，便能避免这个问题。因此，结合基于关键

词位置的 Fish Search 算法，本文提出了一种基于页面主题的 Page Rank 算法。该方法中，首先通过 Page Rank 算法计算其链接的重要度，然后通过基于关键词位置的 Fish Search 算法计算其链接对应页面的主题相关度，将计算出来的主题相关度和重要度综合加权赋值给其子链接，最后将子链接按照综合权大小插入到待爬行 URL 队列中。该算法既考虑了网页链接的重要性，又考虑了网页页面的主题相关度，从而更加准确地计算待爬取 URL 的优先级，能够爬取更多的主题相关的页面，提高爬虫的查准率。

对全文进行总结，本文的主要贡献可以归纳为下几个方面：

(1) 页面的主题相关性计算方面。本文基于 Fish Search 算法，通过分析其计算页面主题相关度的不足，提出了一种基于关键词位置的页面主题相关度计算算法。该算法根据页面中出现的关键词位置的不同赋以不同的权值，然后综合加权后计算得到一个连续值，用这个连续的值来代表页面的主题相关度，更加准确的表示了页面的主题相关度，提高了爬虫的查准率。

(2) 待爬取 URL 的优先级计算方面。本文基于 Page Rank 算法，通过分析其预测链接的不足，结合基于关键词位置的 Fish Search 算法，提出了一种基于页面主题的 Page Rank 算法。该算法首先根据 Page Rank 算法计算链接的重要度，然后根据基于关键词位置的 Fish Search 算法计算其对应页面的主题相关度，综合加权后赋值给其子链接，然后，将子链接按照综合权值大小插入待爬取 URL 队列中。改进后的方法把网页链接的重要度和网页内容的主题相关度都考虑到了，从而能更加准确地计算待爬取 URL 的优先级，爬取到更多主题相关的页面，提高爬虫的查准率。

(3) 及时推信息推送系统方面。本文实现了及时推信息推送系统的爬虫子系统和信息展示子系统的前台界面，该系统包括 PC、Android 和 IOS 三个客户端，并都于成功上线。

7.2 下一步工作展望

本文将改进的主题爬虫算法应用到及时推信息推送系统的爬虫子系统中，该子系统能够根据用户关注的网页和关键词，为用户爬取包含关键词的网页链接，但是其爬取的主题相关页面的准确度仍然有不小的提升空间，本文也有一

些遗留问题与研究的不足，主要表现在：

(1) 对于基于页面主题的 Page Rank 算法，虽然考虑了网页链接的重要性和网页内容的主题相关度来进行链接预测，取得了较好的结果，但是没有加入人工智能的方法，如果能引入机器学习方法对链接进行预测可能效果更好。

(2) 对于本文提出的基于关键词位置的 Fish Search 算法以及基于页面主题的 Page Rank 算法，其中的各项权值和阈值的设置可直接影响爬虫的性能，需要数据进行大量的实验得到最佳权值和阈值，需要进一步改进。

(3) 本文实现的及时推信息推送系统的爬虫子系统都是每天早上给用户爬取主题相关的页面，还不能实现实时的消息推送，后续版本会对这个问题进行改进。

以上是本文主要有待提升的地方，这些问题将在以后继续深入进行研究。

参考文献

[1] 韩磊. 新闻预定服务系统[D]. 山东大学, 2007.

[2] 中国互联网网络中心. 第 35 次中国互联网络发展状况统计报告[R]. http://www.cnnic.net.cn/gywm/xwzx/rdxw/2015/201502/t20150203_51631.htm, 2015.

[3] 林彤, 江志军. Internet 的搜索引擎[J]. 计算机工程与应用 ISTIC PKU, 2000, 36: 160-163.

[4] 王晓伟. 垂直搜索引擎若干关键技术的研究[D]. 浙江大学, 2007.

[5] Bai Kun G G. STUDY AND APPLICATION OF VERTICAL SEARCH ENGINE BASED ON LUCENE AND HERITRIX [J]. Computer Applications and Software, 2009, 1: 212-215.

[6] Wu J-M, Ji D-D, Han Y-H. Research and design of vertical search engine for DCI based on web[J]. Computer Engineering and Design, 2013, 34(4): 1481-1487.

[7] Lesavich Z C. Method and system for creating vertical search engines with cloud computing networks[M]. Google Patents, 2013.

[8] 李东升. 主题搜索引擎研究[D]. 哈尔滨工程大学, 2005.

[9] Pavalam S, Raja S K, Akorli F K, et al. A survey of Web crawler algorithms[J]. International Journal of Computer Science Issues (IJCSI), 2011, 8(1): 309-313.

[10] Boanjak M, Oliveira E, Martins J, et al. Twitterecho: a distributed focused crawler to support open research with twitter data[C]. Proceedings of the 21st international conference companion on World Wide Web, 2012: 1233-1240.

[11] Jiang P, Song J. A method of text classifier for focused crawler[J]. Journal of Chinese Information Processing, 2010, 24(6): 92-96.

[12] Dong H, Hussain F K, Chang E. A framework for discovering and classifying ubiquitous services in digital health ecosystems[J]. Journal of Computer and System Sciences, 2011, 77(4): 687-704.

[13] Pavalam S, Jawahar M, Felix K A. Web Crawler in Mobile Systems[J]. International Journal of Machine Learning and Computing, 2012, 2(4): 531-534.

[14] Chen C, Lu S, Du P, et al. Silent geographical spread of the H7N9 virus by online

knowledge analysis of the live bird trade with a distributed focused crawler[J]. Emerging Microbes & Infections, 2013, 2(12): 1-7.

[15] Chakrabarti S, Berg M V D, Dom B. Focused crawling: a new approach to topic-specific web resource discovery[J]. Computer Networks, 1999: 1623--1640.

[16] Cho J, Garcia-Molina H, Page L. Efficient Crawling Through URL Ordering[J]. Computer Networks and ISDN Systems, 1998, 30: 161–172.

[17] Bra P M E D, Post R D J. Information Retrieval in the World-Wide Web : Making Client-based Searching Feasible[J]. Computer Networks and ISDN Systems, 1994, 27(2): 183-192.

[18] Brin S, Page L. The Anatomy of a Large-Scale Hypertextual Web Search Engine[J]. COMPUTER NETWORKS AND ISDN SYSTEMS, 1998, 30: 107--117.

[19] Kleinberg J. Authoritative Sources in a Hyperlinked Environment[J]. Journal of the ACM, 1997, 46: 668--677.

[20] 李晓明，闫宏飞，王继民. 搜索引擎：原理、技术与系统[M]. 北京: 科学出版社, 2005.

[21] 侯震宇. 主题型搜索引擎的研究与实现[D]. 中国科学院研究生院, 2003.

[22] 傅士光. 基于主题的搜索引擎的研究与实现[D]. 北京交通大学, 2007.

[23] 邱哲 符. 开发自己的搜索引擎[M]. 人民邮电出版社, 2007.6.

[24] 刘金红，陆余良. 主题网络爬虫研究综述[J]. 计算机应用研究, 2007, 24(10): 26-29.

[25] Tarjan R. Depth-first search and linear graph algorithms[C]. Switching and Automata Theory, Annual Symposium on, 1971: 114 - 121.

[26] Najork M. Breadth-First Search Crawling Yields High-Quality Pages[J]. In Proc. 10th International World Wide Web Conference, 2001: 114--118.

[27] O. O S, O. O E, A. O C, et al. Comparative Study Of Complexities Of Breadth-First Search And Depth-First Search Algorithms Using Software Complexity Measures[J]. WCE 2010 - World Congress on Engineering 2010, 2010.

[28] Ni X-G, Cai M. Focused crawler system based on combination of link structure and content similarity [J][J]. Computer Engineering and Design, 2008, 29(7): 1709-1711.

[29] 王晓宇，周傲英. 万维网的链接结构分析及其应用综述[J]. 软件学报，2003，14(10): 1768-1780.

[30] Kleinberg J M. Hubs, authorities, and communities[J]. ACM Computing Surveys, 1999, 31(2): 685-695.

[31] 杨震. 文本分类和聚类中若干问题的研究[D]. 北京邮电大学, 2007.

[32] Korf R E. Linear-space best-first search[J]. Artificial Intelligence, 1993, 62(1): 41–78.

[33] 余静, 段隆振, 熊必成. 基于 Agent 的信息推送系统的研究[J]. 计算机与现代化, 2007, 10: 18-21.

[34] 杨俊, 兰宏勇. 基于 RSS 的信息推送系统的设计和实现[J]. 计算机系统应用, 2008, 10: 64-68.

[35] 黄涛. 基于 Azure 平台的信息推送系统设计与实现[D]. 大连理工大学, 2013.

[36] Haveliwala T H. Topic-sensitive PageRank[C]. In Proceedings of the Eleventh International World Wide Web Conference, 2002.

[37] 曹姗姗, 王冲. 基于网页链接与用户反馈的 PageRank 算法改进研究[J]. 计算机科学, 2014, 41(12): 179-182.

[38] 赵鑫. 基于链接关系分析的 PgaeRank 改进算法研究[J]. 电脑编程技巧与维护, 2014, 12: 26-27.

[39] 钱功伟, 倪林, Miao, et al. 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: 160-164.

[40] Tang T T, Craswell N, Hawking D, et al. Focused crawling for both topical relevance and quality of medical information[C]. Conference on Information and Knowledge Management, 2005.

论文发表及科研成果

1. 发表论文：

**. 方形印章编号识别系统的研究与实现[J][1]. 四川师范大学学报(自然科学版)，2014，37:37-40.

2. 参与项目：

“Miner on web”网络数据挖据技术成果转化项目-“及时推”信息推送系统。

成都高新区地方创新基金项目-基于物联网印章的银行支票交易安全保障系统

3. 获奖情况：

2012-2014 四川大学硕士研究生二等奖

2014-2015 四川大学硕士研究生二等奖

独创性声明

本人声明所呈交的论文是本人在研究生期间的研究工作中，在导师指导下完成的学术研究及取得的研究成果。除了文中特别加以标注和致谢中所写的内容外，论文中没有包含其他人已经发表过或撰写过的研究成果。

本学位论文成果是本人于四川大学研究生期间在导师的指导下完成的，论文成果归四川大学所有，特此声明。

作者签名：

导师签名：

日期：　　年　　月　　日

致谢

论文写到这，也快接近尾声了，三年的研究生生涯也即将结束了，此刻，我要对我研究生学习、生活当中指导过、帮助过、支持过、关心过我的老师、父母、同学和朋友们表示诚挚的感谢。

我第一个要感谢的是我的研究生导师**教授。**教授是一名年轻、工作严谨、专心学术的好教授，是我一直学习的楷模。作为**教授本届唯一一名硕士学生，无论在学术上还是生活上，我都受到了**教授对我的悉心指导和关怀照顾。学术上，她严谨的科研思路深深地影响着我，使得我很快地适应了自己研究生的角色，完成了从一名本科生到研究生的转变。每当我在学术上有什么问题时，**教授总是能在百忙之中抽出时间，为我讲解其中的细节，并且指导我下一次应该如何解决这类问题，鼓励我多思考多学习，使我更加有信心将自己的精力投入到科研学习当中去。生活中，**教授也对我关心备至，我生活中遇到什么问题都可以跟她谈心，她也总是耐心地给我建议，让我非常感激。在此，我真心感谢**教授三年来对我的指导与关心，让我很好的适应了我研究生的角色，在以后的学习工作中，我也将更加努力，不辜负恩师的期望。

其次，我要诚挚感谢我所在机器智能实验室主任**教授。**教授是一名非常让人敬仰的好教授，用“德高望重”来形容他再适合不过了。作为实验室主任，他不仅在实验室营造了很好的学术氛围，带领大家一起读论文；还把实验室的同学当成家人一样，无论政务多么忙，都会抽时间和大家谈谈心。在此，我要深深的感谢**教授三年来对我的关心和照顾。

再次，我要特别感谢我所在机器智能实验室的**教授。**教授作为我们数据挖掘小组的组长，总是对我以及组内的其他成员细心指导，不仅带领我们完成了科研项目，还指导了我们完成小论文，在此，我要深深的感谢**教授三年来对我的关心和照顾。

我还要感谢我的父母亲，是你们含辛茹苦的地把我养大，无论我遇到什么事情都理解我、支持我，我会永远将这份恩情铭记在心。

最后，要感谢给予我帮助的朋友和同学们。非常感谢我的室友兼好友***、

、在三年共同的生活中对我的包容与支持，非常感谢**、**、**、***、***在三年同学相处过程中对我的帮助和关心，非常感谢**、***、***等数据挖掘小组成员在实验室科研学习中给我带来的欢声笑语。

我爱你们。