网络引文的爬虫系统初探

张磊

摘要

随着互联网技术的高速发展，网络信息资源越来越受到人们的关注和重视，逐渐成为人们获取信息和进行科学研究的主要信息来源。网络引文，即网络参考文献也逐渐受到众多学者的喜欢，并成为了一种不可忽视的参考文献。统计分析网络引文，不仅可以了解各学术领域对网络引文的引用现状，而且可以对其进行数据分析、数据挖掘。本文以万方数据库为例，通过爬虫抽取部分引用了网络引文的论文，为后续分析数据提供了强有力的数据保障。

Abstract

With the rapid development of Internet technology, more and more attention has been paid to network information resources. It has become the main information source for people to obtain information and carry out scientific research. Network citation, the network reference, has been gradually favored by many academic workers, and has become a kind of citation type which can’t be ignored. Statistical analysis of network citation can’t only understand the citation status of network citations in various academic fields, but also can analyze data and excavate data. This paper takes Wanfang database as an example, extracts some papers citing network citations through crawler, and provides a powerful data guarantee for subsequent analysis of data.

1. 引言
   1. 选题的缘起和意义

近年来，计算机和信息技术有了突飞猛进的发展，产生了许多新概念和技术，如更高性能的计算机和操作系统、因特网（internet）、数据仓库（data warehouse）、神经网络等等。在市场需求和技术基础都满足的情况下，数据挖掘的概念和技术应运而生。

数据挖掘（Data Mining）旨在从大量的、不完全的、模糊的随机数据中提取出人们事先不知道的潜在的有价值的信息或知识[1]。

网络的迅速发展使各行各业都开始采用计算机及相应的信息技术进行管理和运营。网络引文由于其传播方式的自由性、动态性，也逐渐成为论文参考文献的主要来源之一。为了能更方便的对论文中引用到的网络引文进行数据分析和数据挖掘，我们需要更方便的收集论文的网络引文，然而由于数据量较大、手工操作较为复杂，导致网络引文分析所需的数据供给不足，本文以此为出发点通过设计并实现一个网络引文爬虫系统来爬取文献信息，提供数据分析和数据挖掘所需的海量数据。

* 1. 研究内容和研究方法

本文以万方数据库为例，讲述网络引文的爬取过程。首先介绍了网络爬虫的技术基础，然后研究了网络引文爬虫的一些关键问题如根URL的选取、URL消重等，最后实现了一个针对万方数据库的爬虫系统。

本文的主要研究内容如下，（1）分析系统需求；（2）设计各功能模块；（3）描述爬虫系统的工作流程；（4）解析和保存数据；（5）测试数据。

本文首先对文中涉及到的相关概念进行解释，主要包括：网络爬虫、数据挖掘、运行环境等，然后以万方数据库为例，分析研究涉及爬虫的相关网页的文档结构及数据格式，通过系统分析和系统测试的研究方法，设计并实现了一套基于Node.js的网络引文爬虫系统。最后对系统进行测试（包括数据抽取准确性和抽取效率），得出本系统满足目前的业务需求的结论。

1. 相关概念解析
   1. 网络爬虫

网络爬虫（Network Spider），顾名思义是在网络中爬行的爬虫。其实质为一段计算机程序，该程序对数据的收集通过网页链接地址实现，往往从某一根URL出发，通过向远程服务器发起HTTP请求，读取网页内容，并从中获得与其相关的节点或子节点的链接地址，然后利用获取到的链接地址进行深入循环爬取[2]。

网络爬虫系统主要是用来在浩瀚网络中收集网页或特定类型的数据信息，然后从中解析出客户所需要的信息，并持久化到数据库系统[3]。

网络爬虫根据其系统架构和技术细节的不同，大致可以分为以下几种：通用网络爬虫（General Purpose Web Crawler）、聚焦网络爬虫（Focused Web Crawler）、增量式网络爬虫（Incremental Web Crawler）、深层网络爬虫（Deep Web Crawler）。然而实际生活中的爬虫系统通常是以上几种爬虫技术相结合实现的[4]。

通用网络爬虫[5]又称为全网爬虫（Scalable Web Crawler），它们往往从某个根URL爬取，并逐步将范围扩大到整个网络环境，主要用来给各大门户网站（如：网易、搜狐）和大型Web服务提供商（如：天气采集系统）采集数据。这类爬虫系统的特点是：爬行范围和数量巨大。因此它们通常采用多线程并行执行，以加快爬取速度。此类网络爬虫系统主要分为页面爬行、页面分析、链接过滤、页面数据库、URL队列、初始URL集合几个模块。

聚焦网络爬虫（Focused Crawler），是指根据需要爬取的主题，有选择性地爬取那些与主题相关的页面的网络爬虫[4]。由于这个特点，聚焦爬虫比通用网络爬虫更加节省硬件和网络资源，因为它们只需要爬取与主题相关的页面。分类器和净化器是此类爬虫系统的两个核心模块。分类器，主要用来计算所爬行的页面与主题的相关度，通过相关度判断该页面数据是否有效；净化器，通过某种算法来识别那些只需通过较少的链接就可以访问到大量相关页面的中心页面。

增量式网络爬虫（Incremental Web Crawler）[4]是指对已经下载过的网页进行再次爬取，目的是增量式的更新老数据，在某种程度上，它能够保证当前数据库中保存的数据是尽可能新的数据。增量式网络爬虫的体系结构包含爬行模块、排序模块、更新模块、本地页面集、待爬行URL集以及本地页面URL集。增量式爬虫的目标是保持本地页面集中存储的页面为最新页面以及尽可能保证本地页面集中页面的质量。

Deep Web爬虫中将Web页面按存在方式可以分为表层网页和深层网页。表层网页是指传统搜索引擎可以索引的页面，以超链接可以到达的静态网页为主构成的Web页面。Deep Web是那些大部分内容不能通过静态链接获取的、隐藏在搜索表单后的，只有用户提交一些关键词才能获得的Web页面。

通用网络爬虫是从一个或几个初始URL开始，获得初始页面上的URL列表，在主线程中爬取网页的过程中，不断从当前页面筛选新的URL放入待爬行的URL列表，作为待执行的任务队列，由子线程执行，直到满足系统的停止条件[6]。

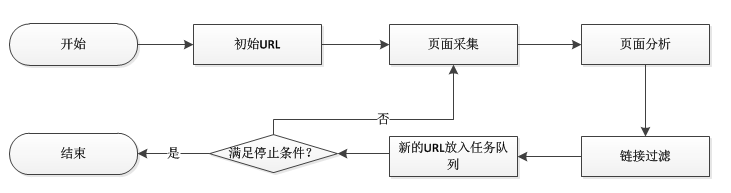


图1 网络爬虫工作流程

其各个部分的主要功能介绍如下。

（1）初始URL：该模块提供初始的URL，以启动爬虫程序。

（2）页面采集模块：该模块是整个网络爬虫系统的入口模块，主要是通过各种web协议（通常为HTTP）来对网页数据进行采集。

（3）页面分析模块：该模块的功能是把页面采集模块采集到的页面数据进行分析，筛选出满足系统需求的URL，加入到任务队列中，如果URL为站点的相对路径，则需要加上网络协议、端口号、域名，使之成为可直接访问的完整的URL。

（4）链接过滤模块：该模块主要是过滤那些满足系统需求的正确格式的链接，并去除已经爬取的重复链接。

1. 1. 运行环境

运行环境是指代码执行的环境，主要指软件环境，本系统运行于Node.js环境下。

Node.js是一个基于Chrome’s V8 JavaScript engine的JavaScript运行环境（Runtime）。通过这个运行环境，可以让JavaScript像后端语言（Java，Python之类）那样，进行文件、数据库等操作。Node.js产生的初衷，是Ryan Dahl想设计一个高性能的Web服务器。他发现高性能Web服务器有几个要点：事件驱动，非阻塞I/O。此外，Ryan Dahl设计了Node.js[7]。

Node.js具有以下特点：

Node.js以单进程、单线程的模式运行，它采用事件驱动的的机制实现并行操作。避免了系统分配线程及线程间通信的开销，可以更高效率的利用cpu的性能，降低内存的消耗。

不同于其他编程语言，Node.js采用非阻塞I/O，提高程序性能，更好的提高单个线程的利用率。Node.js的设计思想是以事件驱动为核心的，Node.js提供的绝大多数API都是基于事件的、异步的风格[8]。事件驱动、异步编程使得程序在执行时无需阻塞等待某个操作（如：HTTP请求、磁盘读写、数据库操作）的完成，使有限的系统资源能得到充分利用。

* 1. 正则表达式

对于网络爬虫系统来说，其实现对网页内容的爬取后，下一步就是利用正则表达式对网页内容进行筛选，提取所需的有效信息。

正则表达式是对字符串的一种逻辑操作公式。它不仅可以判断某一给定的字符串是否符合我们预定的匹配逻辑，而且可以从字符串中提取到我们想要的特定子字符串。正则表达式使用一系列元字符来描述和匹配一个或多个符合规则的字符序列，在很多文本编辑器中正则表达式通常用来做文本的检索和替换。数据提取和模式匹配因为正则表达式的存在将变得不再是一件困难的事情[9]。

简而言之，正则表达式主要有两个作用：

（1）匹配：判断某字符串是否符合正则表达式的过滤逻辑

（2）提取：通过正则表达式我们可以从字符串中提取出我们想要的那一部分

因此，正则表达式的主要功能是实现字符串的匹配和提取，在爬虫程序中正则表达式对逻辑判断意义重大。本文实现爬虫的过程中，对URL及其他重要字段的解析与判断都是依靠正则表达式实现的。

常见的正则表达式元字符解释如表1所示。

表1 常用正则表达式元字符

| ^ | 匹配一个输入或一行的开头 |
| --- | --- |
| $ | 匹配一个输入或一行的结尾 |
| \* | 匹配前面元字符0次或多次 |
| + | 匹配前面元字符1次或多次 |
| ? | 匹配前面元字符0次或1次 |
| {n} | 精确匹配n次 |
| {n,} | 匹配n次以上 |
| [xyz] | 匹配这个集合中的任意一个字符(或元字符) |
| {n,m} | 匹配n-m次 |
| \d | 匹配一个数字字符 |
| \D | 匹配一个非数字字符 |

本章对网络引文爬虫相关的概念及技术背景进行了简单的介绍，为后续的系统设计与实现提供了理论基础，具体包括：网络爬虫的概念和分类、网络爬虫的工作流程以及网络爬虫系统的运行环境Node.js和正则表达式。

1. 需求分析
   1. 业务需求描述

本系统以万方数据库为例，爬取万方数据库中引用有网络引文的文献，然后将其持久化到数据库中。以《聚焦爬虫系统研究综述》为例，在论文详情页可以看到如下图2内容，点击图中（1）处的参考文献会出现右侧具体的参考文献的列表，如果参考文献较多，则会如（3）处所示分页显示。在（2）处出现有[EB/OL]标识的即为引用的网络引文，系统首先根据搜索关键字筛选文献列表，再针对每篇文献分页爬取参考文献列表，并解析筛选含有“[EB/OL]”的参考文献，然后将该文献持久化到数据库中。



图2 参考文献网页截图

在数据爬取流程中，最重要的任务有三个：其中最重要也是最核心的任务就是发送HTTP请求下载网页；另一个重要的任务就是依据爬取规则对待爬取的URL生成爬取任务，并将任务放入自己维护的任务队列中去，如果队列为空，爬虫程序将会停止工作；最后一个重要的任务就是对爬取到的页面进行解析和数据抽取，保证抽取到的数据是有效的。

1. 1. 功能性需求分析

（1）网页下载与信息过滤

系统首先对某搜索关键字搜索到的论文页面进行下载，并根据规则对网页内容进行初步筛选，将响应超时及响应的状态码不正确的网页过滤掉，然后根据DOM树分析得到的DOM选择器筛选出需要的论文列表，包括：论文的标题、论文在万方数据库中的访问URL、论文在万方数据库中存储的ID。之后再根据论文的URL爬取论文详情页的参考文献列表部分。

（2）抽取有效数据

根据论文URL爬取论文详情页面的参考文献时，需要分页爬取所有的参考文献页面，并从这些参考文献列表中抽取后缀含有“[EB/OL]”的参考文献，即表示该论文引用了网络引文。最终将这些引用了参考文献的文献持久化到数据库中，以方便后续数据分析。

（3）任务队列

系统需要提供一个或多个拥有一定并发量的任务队列来管理和维护任务，将任务的分发和任务的执行独立开，降低系统耦合度和各模块之间的依赖。

（4）数据的更新与存储

网络爬虫根据爬取规则下载HTML页面，并根据DOM选择器筛选出所需数据后，需要将获取到的有效数据存储到MySQL数据库中。同时由于数据源可能不定时更新数据，因此在系统进行了一次完整的爬取后，需要不定期对源数据进行增量爬取，也就是所谓的数据更新，在数据更新过程中，我们通过唯一索引来实现对文献的唯一标识。

本章从系统的业务需求、功能性需求两个方面描述了系统的各方面需求。并具体阐述了网页下载与信息过滤、抽取有效数据、数据的更新与存储这些功能性需求，为下一步网络引文爬虫系统的设计提供了思路。

1. 系统设计方案
   1. 系统结构

根据上一章对系统需求的分析，本章设计网络引文爬虫系统，主要采集引用了网络引文的文献及具体的网络引文标题。该系统应达到以下条件：

（1）易配置性。仅需通过简单的配置即可完成系统参数配置。

（2）爬取速度快。采用任务队列的方式，异步执行任务，充分利用Node.js任务驱动的特性，提高系统资源的利用率和系统吞吐量。

（3）较高的数据准确度。通过正则表达式准确匹配有效信息[10]。

网络引文爬虫系统的系统结构如图3所示。

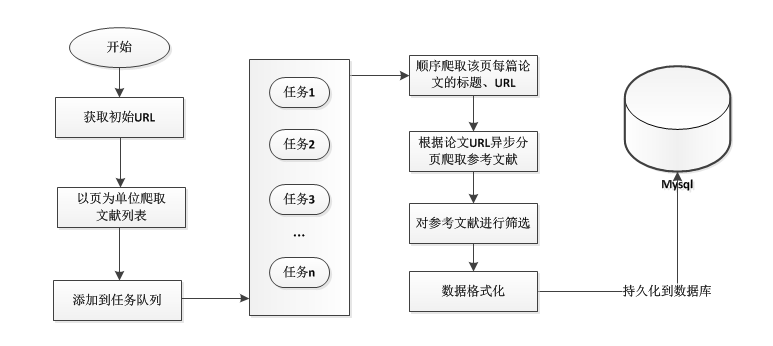


图3 网络爬虫系统结构

1. 1. 功能设计

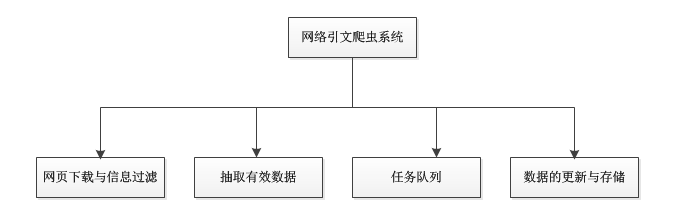
网络引文爬虫系统主要包含了网页下载与信息过滤、抽取有效数据、任务队列、数据的更新与存储四大功能模块，如图4所示。

图4 网络引文爬虫系统功能模块

网页下载与信息过滤模块主要负责发送HTTP请求，下载HTML网页，并直观地过滤掉一些不可能存在有效信息的部分，以避免无意义的操作，浪费系统资源。本系统从初始URL开始下载网页，过滤掉网页中除了文献列表以外的其他网页部分，筛选出文献列表并生成任务添加到任务队列的末尾。

抽取有效信息模块负责对下载的HTML进行有效数据的进一步筛选。系统通过正则表达式调用相应的匹配规则对文献列表中各标签的内容进行匹配，匹配出最大页数用来分页爬取参考文献以及判断引文标题是否含有“[EB/OL]”字样等。

任务队列模块负责提供任务执行的环境，该任务队列可根据系统性能合理配置并发量，等待执行的任务进入队列末尾等待执行。这样在提高资源利用率的同时，可方便的监控队列中任务的积压情况。

数据的更新与存储模块负责将爬取到的有效数据存储到MySQL数据库中，MySQL通过唯一索引来达到数据消重的效果。

本章主要从系统总体设计、系统各功能模块设计两个方面详细阐述了系统的结构及主要功能，包括：网页下载与信息过滤、抽取有效数据、任务队列、数据的更新与存储，为下一步系统的具体实现提供了方案。

1. 系统实现
   1. 系统环境

本系统在win7环境下测试运行通过，具体测试环境参数如下表2所示。

表2 系统环境

|  |  |
| --- | --- |
| 硬件环境 | CPU：Intel i7-5500U；内存：8GB DDR3；硬盘：1TB SATA3 |
| 操作系统 | Windows 7企业版 |
| 开发工具 | Web Strom 2017.1  Node.js v4.4.4  MySQL v5.0.95 |

1. 1. 网页下载与信息过滤功能

在此模块中，首先通过分析搜索结果页面的网页结构得到初始URL，在图5中（1）处，可观察到（2）处出现URL，由此可粗略判断该URL为包含文献列表结构的URL页面。

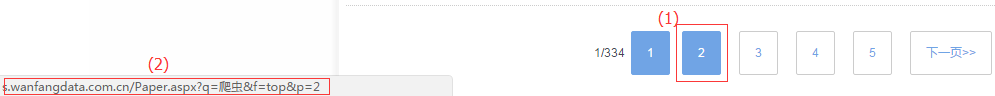


图5 搜索结果页码链接网页截图

在图6中，通过Chrome的开发者工具，我们清楚的看到并验证了图5中（2）处的URL，即：“http://s.wanfangdata.com.cn/Paper.aspx?q=keyword&f=top&p=page”为包含文献列表的初始URL，其中keyword为搜索的关键字，如果是中文则需要重新编码，page为当前要爬取的页码。在图6中，从右半部分图片中可清晰的看到HTML文档的DOM层次结构，通过“.left-record .record-title .title”选择器即可对每篇文献的标题部分HTML进行定位及进一步筛选数据。

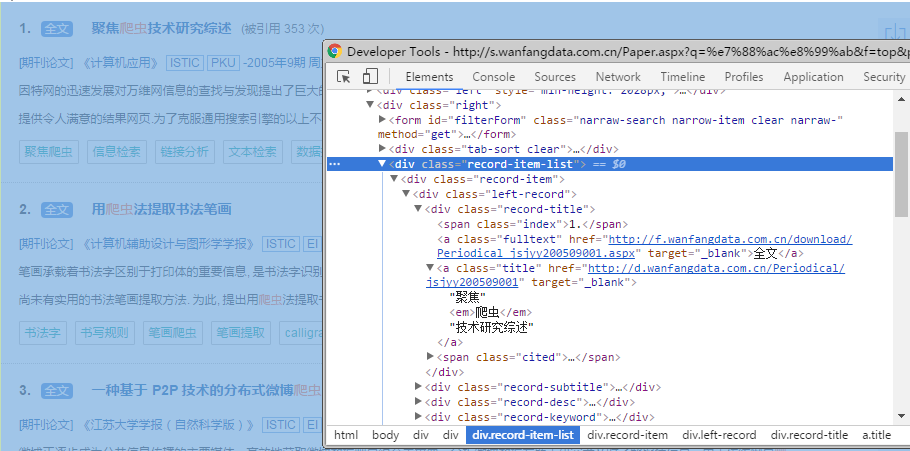


图6 网页DOM结构截图

该模块通过Node.js第三方包axios方便的实现HTTP请求的发送，下载网页内容，接下来通过cheerio对下载的HTML进行过滤，过滤出文献的列表及每篇文献对应的标题、URL、文献ID，提供给数据抽取模块进行下一步的数据抽取。如果在此过程中发生错误或出现非预想的结果，则处理异常并打印异常日志。该部分的两个步骤是爬虫系统的公共步骤，除了具体实现细节不同外，大体逻辑和结构都是一样的。

* 1. 抽取有效数据

在论文的详情页面中，我们需要爬取参考文献列表部分的HTML文档，但是在详情页面中我们并没有找到对应的HTML。因此，初步判断参考文献列表通过AJAX异步加载。为了验证这一假设，再次打开Chrome工具，保证图7中（2）处Network面板下内容为空，点击图7中（1）处的参考文献，可以看到在图8的Network面板下（1）处出现了一次HTTP请求，在图8中（2）处可以看到该HTTP请求的具体参数，如URL、请求的方法、远程IP等信息，在图8中（3）处出现了参考文献列表，由此我们得到了参考文献的HTTP链接，即：“http://d.wanfangdata.com.cn/CiteRelation/Ref?id=articleId&page=page”，其中articleId为万方数据库中存储的每篇文献的唯一标识，page为当前爬取的页码。

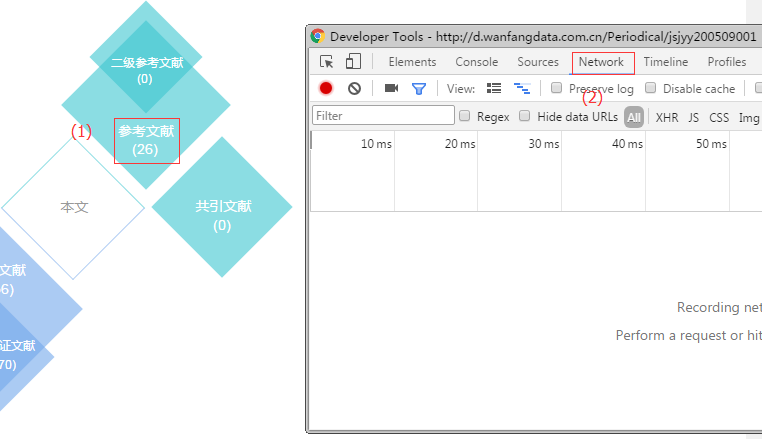


图7 Chrome开发者工具截图1

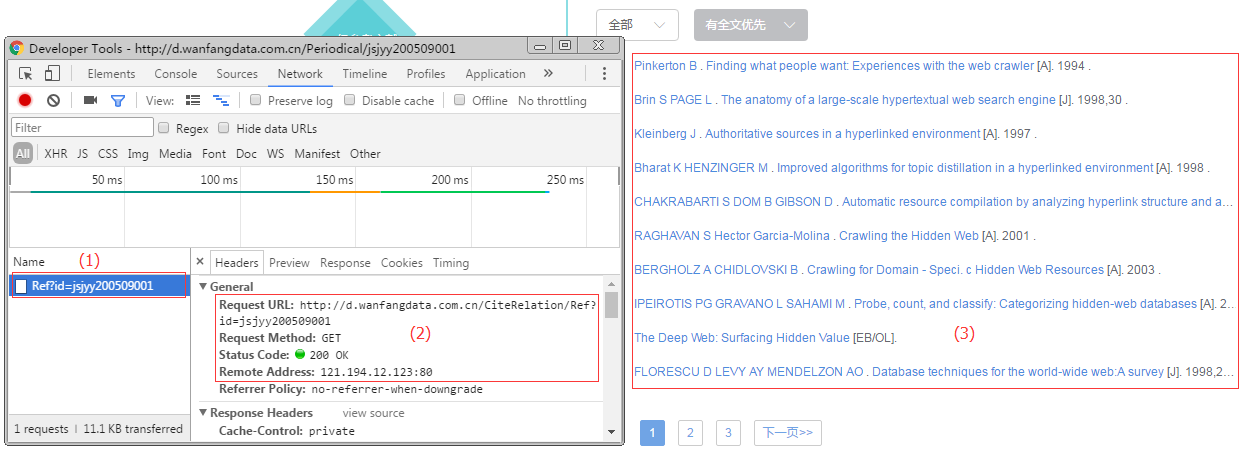


图8 Chrome开发者工具截图2

在该模块实现中，我们依然通过网页下载与信息过滤模块的公共逻辑实现发送HTTP请求，解析HTML的功能。在实现细节上，通过一系列与业务相关的逻辑判断参考文献列表中是否包含“EB/OL”，即网络引文。

* 1. 任务队列

在该模块中，通过async创建了一个可配置并发数量的任务队列，即当并发数为n时，同一时刻最多同时有n个任务在执行，后续任务进来时被加入任务队列末尾和其他待执行任务一起排队等待执行。

* 1. 数据的更新与存储

该模块根据文章标题生成对应的MD5加密字符串，然后通过MySQL数据库连接包mysql实现数据的更新与存储，加密后的md5作为唯一索引用于防重。

1. 系统测试

该部分以“爬虫”关键字搜索到的第一页论文（每页10篇）为例，逐篇下载并查看对应的参考文献，通过人工对比的方式验证抓取数据的准确性。

系统配置如下所示。

|  |
| --- |
| {  // 关键字  keyword: '爬虫',  // 分页并发数  pageConcurrency: 2,  // 单篇论文并发数  itemConcurrency: 5,  // 开始页  startPage: 1,  // 页数  pageCount: 1,  // 响应超时时间(ms)  timeout: 3000  } |

爬取到的数据如下所示。

|  |
| --- |
| [2017-05-20 16:30:14.843] [INFO] out - { articleTitle: '基于度排序的P2P IPTV分布式爬虫系统设计与实现',  articleUrl: 'http://d.wanfangdata.com.cn/Periodical/scdxxb-gckx201403016',  articleId: 'scdxxb-gckx201403016',  page: 1,  md5: '06e4dabd1d4339e120822e3a8f1f27df',  text:  [ '上海聚力传媒技术有限公司PPTV homepage',  '上海众源网络有限公司PPStream homepage',  '北京时越网络技术有限公司UUSee homepage' ] }  [2017-05-20 16:30:14.874] [INFO] out - { articleTitle: '广域网分布式Web爬虫',  articleUrl: 'http://d.wanfangdata.com.cn/Periodical/rjxb201005016',  articleId: 'rjxb201005016',  page: 1,  md5: '465e710f3f45fb33c6ad2eb660217226',  text:  [ 'CNNICThe 21st statistical survey report on the Internet development in China',  'Boswell DDistributed high-performance Web crawlers:A survey of the state of the art' ] }  [2017-05-20 16:30:14.890] [INFO] out - { articleTitle: '聚焦爬虫技术研究综述',  articleUrl: 'http://d.wanfangdata.com.cn/Periodical/jsjyy200509001',  articleId: 'jsjyy200509001',  page: 1,  md5: '770ae4654db34c1f9c86f4e6c4f217ca',  text: [ 'The Deep Web: Surfacing Hidden Value' ] }  [2017-05-20 16:30:15.124] [INFO] out - { articleTitle: '基于度排序的P2P IPTV分布式爬虫系统设计与实现',  articleUrl: 'http://d.wanfangdata.com.cn/Periodical/scdxxb-gckx201403016',  articleId: 'scdxxb-gckx201403016',  page: 1,  md5: '06e4dabd1d4339e120822e3a8f1f27df',  text: [ 'CNNIC中国互联网络信息中心第32次中国互联网络发展状况调查统计报告' ] }  [2017-05-20 16:30:15.140] [INFO] out - { articleTitle: '聚焦爬虫技术研究综述',  articleUrl: 'http://d.wanfangdata.com.cn/Periodical/jsjyy200509001',  articleId: 'jsjyy200509001',  page: 1,  md5: '770ae4654db34c1f9c86f4e6c4f217ca',  text: [ 'Wu JABERER KUsing SiteRank for Decentralized Computation of Web Document Ranking' ] }  [2017-05-20 16:30:21.567] [INFO] out - { articleTitle: '面向蒙古文主题的网络爬虫采集策略模型',  articleUrl: 'http://d.wanfangdata.com.cn/Periodical/bjgydxxb201507009',  articleId: 'bjgydxxb201507009',  page: 1,  md5: 'b5abfceda1a5902a6e70f4c938a5fbd9',  text: [ '文化中国中国少数民族语言现状', '中国互联网络信息中心第34次中国互联网络发展状况统计报告' ] } |

在手工下载该页论文后，发现确实存在且仅存在以下网络引文，如下所示。

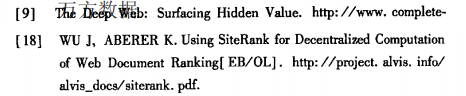


图9 《聚焦爬虫技术研究综述》论文截图

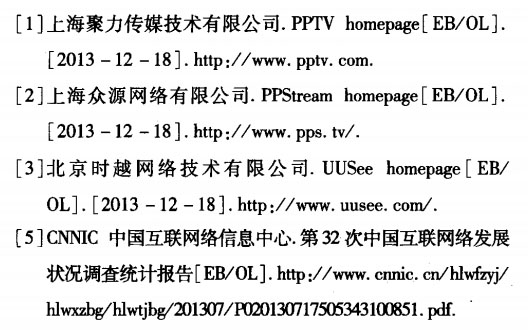


图10 《基于度排序的P2P IPTV分布式爬虫系统设计与实现》论文截图

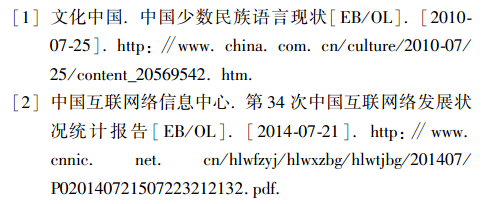


图11 《面向蒙古文主题的网络爬虫采集策略模型》论文截图

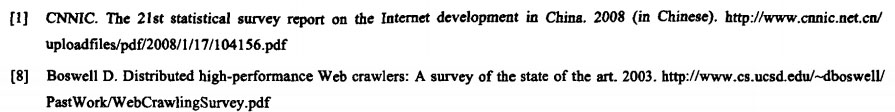


图12 《广域网分布式Web爬虫》论文截图

* 1. 抽取准确性

通过手工验证爬取的第一页论文数据，发现爬取数据与手工处理的数据完全一致，满足数据抽取的需求。测试时，爬取10页论文，并对爬取到的论文进行随机抽查验证，均通过验证，数据抽取的准确性接近100%。这样的数据准确性为后续的数据分析和数据挖掘提供了高质量、高标准的数据保证，满足了业务需求。

* 1. 抽取用时

网络爬虫程序通常被用来为搜索引擎、数据挖掘提供数据支持。本文通过在100Mbps网络环境下对万方数据库进行定向爬虫，在爬取用时方面进行了测试，测试数据如表3所示（由于受到网络状况及万方服务响应时间的影响，数据仅供参考）。

表3 测试结果

|  |  |
| --- | --- |
| 页数 | 平均用时（S） |
| 1 | 3.395 |
| 5 | 9.068 |
| 10 | 13.056 |
| 30 | 28.976 |

本章主要给出了网络引文爬虫系统的模块关系图，并简要介绍了网络引文爬虫系统各功能模块的具体实现，然后在测试环境下测试数据爬取的速度和准确率，并给出了测试数据。从测试的结果看，系统运行状态良好，能基本满足业务需求。

1. 总结

本文首先通过大量的文献调查确定了论文的写作方向及研究内容，解释了一系列与本文内容有关的概念，包括：网络爬虫、运行环境、正则表达式。然后针对当前学术环境下关于对网络引文的数据分析、数据挖掘不方便的现状提出了业务需求，并对具体需求进行分析和思考，设计了一套专门用于爬取万方数据库网络引文的的爬虫系统，并实现了预期的各个功能：

（1）网页下载与信息过滤：系统通过定向数据爬取的方法从万方数据库下载HTML，并初步过滤掉部分网页。

（2）抽取有效数据：系统采用正则表达式对网页中的信息进行精确匹配以抽取有效数据。

（3）数据的更新与存储：系统使用MySQL来存储抽取到的数据。

（4）任务队列：使用任务队列管理爬虫任务，将任务细分，提高系统效率和资源的利用率。

本文在最后在系统完成之后对系统进行了简单的测试（包括：爬取速度、数据准确性），并以表格的形式给出了系统测试的结果，表明了本系统运行正常，可以满足业务需求。

然而由于本人时间和能力有限，系统还有很多问题需要进一步解决，具体问题如下：

（1）系统监控：系统应该有一个监控模块，将系统细分并监控每个细节的性能，如：HTTP请求响应时间、任务队列积压情况等。

（2）日志管理：目前系统日志虽然能满足当前业务需求，但在系统逐渐扩展之后，就需求一个日志模块对系统的日志进行统一管理，如错误日志的处理、失败的HTTP请求的重试策略等。

（3）可视化：目前系统只是在控制台实现操作，没有一个可视化的页面供使用者操作。

参考文献

[1] 数据挖掘基础知识 [EB/OL]. [2017-05-09]. http://www.xuexila.com/baikezhishi/1244762.html

[2] 赵茉莉. 网络爬虫系统的研究与实现[D]. 成都: 电子科技大学, 2013

[3] 李海燕. 网络舆情爬虫系统的设计与实现[D]. 厦门: 厦门大学, 2014

[4] 孙立伟, 何国辉, 吴礼发. 网络爬虫技术的研究[J]. 电脑知识与技术, 2010, 6(15): 4112～4114

[5] 李盛韬, 余智华, 程学旗, 白硕. Web信息采集研究进展[J]. 计算机科学, 2003(2): 151～156

[6] 曾伟辉. 支持AJAX的网络爬虫系统设计与实现[D]. 合肥: 中国科学技术大学, 2009

[7] node.js[EB/OL]. [2017-05-11]. http://baike.baidu.com/item/node.js?fromtitle=nodejs

[8] 王越. 基于nodejs的微博系统的设计与实现[D]. 成都: 电子科技大学, 2014

[9] 正则表达式[EB/OL]. [2017-05-15]. http://www.jianshu.com/p/12e17ff09176

[10] 陈瑜芳. 主题爬虫系统的研究[D]. 武汉: 武汉理工大学, 2010