一种轻量级中文搜索引擎模型的设计与实现

黄宇达[1,2],魏 霞[2],王迤冉[3]
(1. 西南科技大学 计算机科学与技术学院,四川 绵阳 621010;
2. 周口职业技术学院 信息工程系,河南 周口 466000;
3. 周口师范学院 计算机科学与技术学院,河南 周口 466001)

摘 要:首先详细介绍了一种建构在 PC Windows 平台上的轻量级中文搜索引擎系统模型的总体设计,然后采用基于多线程技术的广度优先遍历法及最大匹配法和最小匹配法相结合的中文分词法等技术进行了各个主要功能模块的具体设计和实现,对模型进行了基于多线程的网络爬虫、用户接口等测试。测试实验结果表明:构建并实现的轻量级中文搜索引擎系统模型能较好地实现一个简单中文搜索引擎所具有的基本功能,系统界面简单实用,具有较高的资源检索率并能够保证检索结果的准确性。

关键词:网络爬虫;URL 库;中文分词;倒排文件索引;多线程
中图分类号:TP31 **文献标识码**:A **文章编号**:1673-629X(2012)09-0201-04

Design and Implementation of System Model of a Lightweight Chinese Search Engine

HUANG Yu-da[1,2], WEI Xia[2], WANG Yi-ran[3]
(1. College of Computer Science and Technology, Southwest University of Science and Technology, Mianyang 621010, China;
2. Information and Engineering Department, Zhoukou Vocational and Technical College, Zhoukou 466000, China;
3. College of Computer Science and Technology, Zhoukou Normal University, Zhoukou 466001, China)

Abstract: First described in detail the overall design of the lightweight Chinese search engine system model based on PC Windows platform, and then the major functional blocks were designed and realized by using breadth-first traversal method based on multi-threading technology and the Chinese sub-lexical method of the combination of the maximum matching method and the minimum matching method and other technology, then carried out some tests based on multi-threaded Web crawler and user interface on the model. Experimental results show: the lightweight Chinese search engine system built and realized is able to achieve the basic functions of a simple Chinese search engine and good operating results, the system interface is simple and practical, with higher rates of resource retrieval and to ensure the accuracy of search results.

Key words: Web crawler; URL library; Chinese word segmentation; inverted file index; multi-threaded

0 引 言

随着计算机和互联网技术的飞速发展,信息获取手段经历了从手工获取到计算机获取,直至今天的通过网络进行获取,因此,在存有海量信息的互联网上进行高效地搜索信息,搜索引擎是必不可少的[1]。

搜索引擎是仅次于门户的互联网第二大核心技术[2],是一种使用某种策略在互连网中搜索、发现信息,对信息进行理解、提取、组织和处理,并为用户提供检索服务,从而起到信息导航目的的工具[3]。搜索引擎技术既适用于互联网络,也适用于企/事业单位和部门。伴随互联网的普及和网上信息的爆炸式增长,它越来越引起人们的重视,可以说在过去的 20 年中,用于信息检索的领域已经得到发展和壮大,并且也超越了它标引文本和在某一集合中检索出有用文档的最初目标[4,5]。

文中利用 JAVA+JSP+TOMCAT6.0+SQL SERVER2005 作为开发工具,设计并实现了一个基于网络爬虫的轻量级中文搜索引擎模型的构建。

收稿日期:2012-02-10;**修回日期**:2012-05-11
基金项目:河南省科技基础与前沿技术研究计划项目(112300410307)
作者简介:黄宇达(1975-),男,河南周口人,硕士,讲师,研究方向为知识工程、并行计算等。

1 模型总体设计

简单说来,搜索引擎是一类服务软件,该软件包括前台部分和后台数据库两部分[6];具体来讲,搜索引擎由四部分组成[7]:网络爬虫(Crawler,又称 Robot,Spider,Web Wanderer 等)、索引器(Indexer)、检索器(Searcher)和用户接口。网络爬虫负责从网络上搜索并采集网页;索引器对网络爬虫采集到的信息进行过滤、整合,然后建立相应的索引;检索器(又称查询器)根据用户的查询请求,从后台索引库中检索出与之相关的信息并反馈给用户;用户接口提供系统的检索界面,是用户与搜索引擎交流的窗口[8]。

文中系统模型具有如下基本功能:对网络进行爬行获取互联网信息;对这些信息进行一定的分析和处理后,存储在数据库中并建立索引;提供用户接口供用户进行检索使用。

在设计本模型时,将重点设计网络爬行器(网络爬虫)、索引器以及检索器。另外,由于这个系统主要是对中文网页进行检索,所以还需要有中文分词功能。系统模型的总体结构简图如图 1 所示。

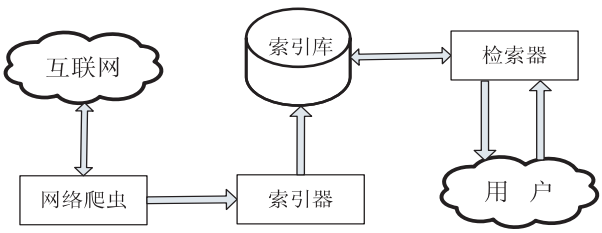


图 1 系统模型总体结构简图

2 模型主要功能模块设计及实现

2.1 数据库功能模块的设计

本系统模型采用 SQL SERVER2005 作为后台数据库。由于 SQL SERVER 对每一行的最大长度有限制,即每行最大长度为 8060 字节,根据统计,每个网页的平均长度为 14.8KB[8],所以一行是有时不能够保存整个网页内容,对此,本设计设定每行最多保存 8000 个字节,多的再另行处理。为保存数据,又另外设立了两个变量 Number 和 isLast。其中,Number 保存当前行是针对于所保存的网页而言的序号;isLast 则指示当前记录是否是本网页的最后一行记录,如果是则值为 1,否则值为 0。

其实搜索引擎并不真正搜索互联网,它搜索的实际上是预先整理好的网页索引数据库[9]。另外,为提高检索速度,建立了一个基于关系数据库改进的倒排文件表,即将该表次关键字作为索引来进行检索,当然索引的目的是为了提高检索速度[10]。在将网页进行分词并得到相应的关键词后,将他们插入到数据库中,同时保存对应的网页 ID 和偏移量。在使用这个倒排文件表的时候,只需根据所给的关键字对该表进行检索,如果检索成功,则得出所有含有这个关键字的记录行,从而得出所有含有这个关键字的文件信息。

2.2 网络爬虫功能模块的设计及实现

2.2.1 网络爬虫功能模块的设计

网络爬虫是搜索引擎的核心组件,用于遍历 Web 和下载页面。爬行网页的数量、速度和种类都直接影响检索结果,因此,搜索引擎的性能、规模、扩展能力在很大程度上都依赖于网络爬虫的处理能力。

针对爬虫爬行策略,由于本系统模型是对所有中文站点而言的,而且并非主要针对专业文献检索,所以为了提高程序的效率和对机器硬件资源的使用率,本模型在程序具体编码时采用多线程技术,在遍历策略上采用广度优先搜索遍历。

针对爬虫爬行范围,由于互联网资源庞大,即使目前最通用的搜索引擎的数据库中也只保存了互联网上的其中一部分,而且鉴于本引擎模型又是轻量级的,所以本模型将爬虫爬行范围定义在整个互联网上的中文静态 HTML 文件,同时排除了图片和 MP3 等多媒体文件。

2.2.2 网络爬虫功能模块的实现

本系统模型的网络爬虫的工作流程如图 2 所示。

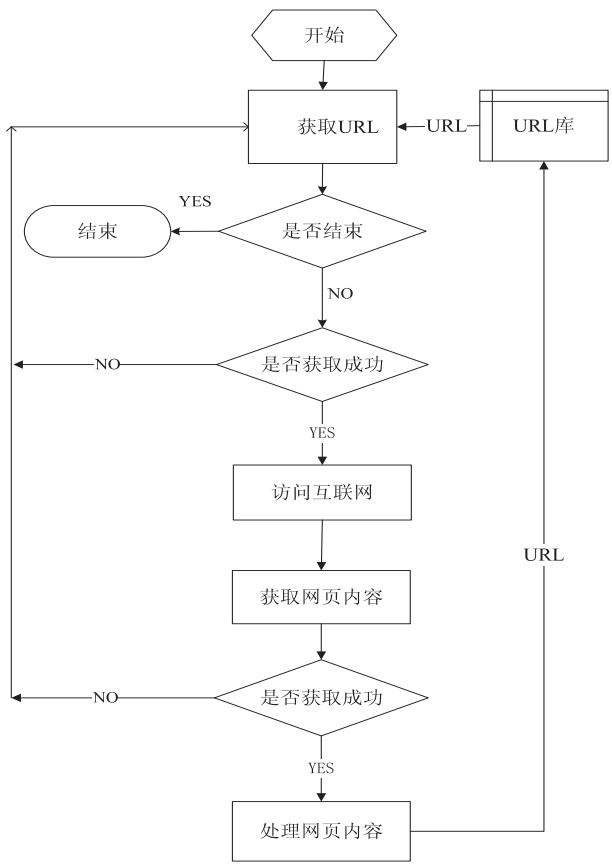


图 2 网络爬虫基本流程图

首先获得 URL 地址,然后判断当前 URL 库中 URL 是否访问完了,若已经访问完了则就不再生成新

线程,否则进行下一步;根据所获得的 URL 派生出新线程来访问互联网的资源,从而获取对应网页的内容;接着判断是否获得网页内容成功,若失败则转向开头重新处理,否则继续进行下一步;最后分析所获得的网页内容并提取相关信息后,将其含有的 URL 加入到 URL 库中,同时转向开头继续处理。

对于 URL 库的实现,本模型采用队列来保存所获取的 URL。分别设计了三个队列即未访问队列、正在访问队列、已访问队列。它们之间的约束关系是:未访问队列的 URL 可以进入正在访问队列,但是不能直接进入已访问队列;正在访问队列中的 URL 只能进入已访问队列;已访问队列则在程序运行期间不能移动。

对于获取新的 URL 链接,由于本模型爬虫只是对静态网页进行爬取,所以在提取 URL 链接的时候只需提取相应后缀名为 html 或者 htm 的文件,同时还考虑到所爬行到的 URL 可能不是静态网页,所以也默认为它们是静态网页进行保存 URL。在获得新的链接之后,将把这个链接保存在未访问队列中。至于其是否存在,则由队列的相关定义约束。

本模型网络爬虫实现的关键代码如下:

```
try{//从未访问队列获取一个链接
curr_URL = notAccQue.pull().toString();URL proc = new URL(curr_URL);
URL tproc = proc.toExternalPage();//调用函数 toExternalPage()将获取的当前链接 proc 转为完整 URL 格式
BufferedReaderinpu = new BufferedReader(
new InputStreamReader(proc.openStream(),"GBK"));//新建一输入流对象且将输入编码设置为"GBK"
while((k = inpu.readLine())! =null){cont+= k;} cont = cont.toLowerCase();//读取互联网中网页内容并将当前已获取的网页内容都转为小写格式
cont = cont.replaceAll(" "," ");cont = cont.replaceAll("'","\"");//将当前获取的网页中的空格转为标准 HTML 格式
getTitle = new GetDescription(cont); pageTitle = getTitle.GetTitle();//调用 GetTitle()函数获得当前网页的标题
pageAddress= proc.toString();//获取当前网页所在网站的网址
pageChange = new GetPrefix(pageAddress); temp = pageChange.ChangeString();
inpu.close();//关闭输入流}catch(MalformedURLException e){
System.out.println(e);}catch(IOException e){System.out.println(e);//若访问当前网页时出错,则输出错误信息}
```

上面代码段功能是根据给定的 URL 链接来访问网络,以获取相应链接的网页内容。当获取网页内容之后,将其全部转换为小写,同时提取标题,并去除无用的代码,最后,关闭网页输入流。

2.3　中文分词功能模块的设计

本系统模型采用机械匹配法的中文分词法。首先建立一个词库,由于构建词库是一个长期而繁琐的工作并需要不断地修改和更新,因此本模型采用在网上搜索到的以前百度使用的分词词库。另外,由于词库数据量的巨大,所以首先使用 SQL SERVER 的导入和导出功能再加上相应的处理后,得到了每列只具有词语的词典。同时约定:当行是由"#"开始的,则认为这是一个注释行。当词典建立好之后,具体实现方法就只和分词策略和方向有关[11]。

另外,采用最大匹配法和最小匹配法进行分词,不过在实际使用过程中,所使用的都是同一个词库。

2.4　用户接口功能模块的设计及实现

2.4.1　用户界面模块的设计及实现

文中采用嵌入 HTML 的 JSP 方式来开发用户查询窗口。在这里将本搜索引擎命名为"Search It"。在搜索引擎的主页中,拥有三个区域,分别是 Logo、查询操作区和最下面的区域。

在查询操作区中,用户可以输入关键字进行检索,其主要功能是由查询分析和处理模块来实现的。当用户输入想要查找的关键字后,用鼠标单击"Search"图标按钮时,便会转向 SearchEngine.jsp 文件,并将关键字以 Get 的方式传递给该文件进行分析处理。当 SearchEngine.jsp 文件得到传来的关键字后,就调用相关的 JavaBean 对关键字加以分词,然后将分词结果对数据库加以检索,最后得到相应的结果[12]。

2.4.2　查询分析和处理模块的设计及实现

文中采用 JavaBean 技术,即通过 JSP 来调用 JAVA 文件以对网页数据进行处理。JavaBean 是一种 JAVA 语言写成的可重用组件。其中,JavaBean 文件的声明方式为:

```
<jsp:useBean id="spl" class="searchEngine.SplToWord" scope="application"/>
```

在这里调用了 searchEngine 文件夹下的名为"SplToWord"的类文件,并将其在本 JSP 文件中的调用名定为"spl",同时设定此 JavaBean 的生存期与整个应用程序相同。

JavaBean 文件的调用方式如下所示:

```
splKeys = spl.segmentLine(keyWords," ");
```

在这里调用了在上面所声明的 spl,并调用了 spl 所对应的 JAVA 文件的函数 segmentLine 来完成分词功能。

3　系统模型测试

3.1　基于多线程的网络爬虫测试

测试用例基本情况描述如下:

测试功能:使用网络爬虫对网络进行爬行;

测试目的:能够实现对网络的正常爬行功能;

测试方法:黑盒测试;

测试前提条件:已经建立数据库并且数据库服务器正在正常运行;

输入情况:无;

期望测试结果:能成功对网络进行爬行并提取相关信息。

测试结果如图 3、图 4 所示。

图 3 基于多线程的网络爬虫爬行结果一

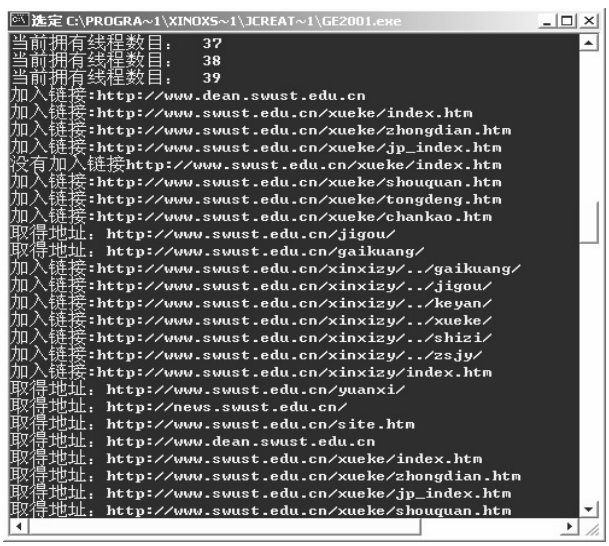


图 4 基于多线程的网络爬虫爬行结果二

在图 3 中,首先完成装载词库,然后再加入初始设定的两个测试连接(界面第 4、5 行),接着派生出线程并成功获取 URL 链接,然后开始对网络进行爬行,爬行结果如图 4 所示。

由图 4 可以很清晰地看到在爬行到新的 URL 时的处理策略。图中有打印的显示中有“加入链接+URL”、“没有加入链接+URL”和“取得地址+URL”。这是由队列约束条件实现的。当新生成线程的时候,将从队列中取出链接,因此会打印“取得地址+URL”,从而对这个 URL 所对应的网页进行爬行;当对爬行的网页进行分析的时候获得的新链接,在加入 URL 队列中时,将对其进行判断,如果存在则不会加入到队列中,于是便打印“没有加入链接+URL”;但如果没有则加入到 URL 队列中,于是便打印“加入链接+URL”。

从上面的测试用例和运行结果,可以看出本系统模型网络爬虫程序的运行结果已经达到所需要求,能够成功的访问网络并下载网页数据。

3.2 用户接口测试

在搜索文本框中输入“新闻”,由于数据库中有和“新闻”相关的信息,所以从数据库中检索结果为 26 条。

测试结果表明:本中文搜索引擎模型的用户接口具有较好的用户界面,同时能够完成所需要的功能且具有一定安全性。

4 结束语

实验结果表明:该搜索引擎系统模型已经基本实现了一个简单的中文搜索引擎所需要的功能,比如网络爬行、中文分词、基于关系数据库的倒排文件索引表的建立及基本数据的查询和浏览等一系列功能。

但该模型还有一些需要改进的地方。比如:没有考虑网页更新策略和方式,只是在管理员需要的时候进行执行;在中文分词时使用的是基于机械匹配法的最大/小匹配法,在改进中应考虑使用基于理解的方式来进行分词,以进一步提高分词准确性等等,这些正是作者下一步着重研究和改进的方向。

参考文献:

[1] 周 军,迟呈英. 基于校园网的中文搜索引擎系统[J]. 沈阳师范大学学报(自然科学版),2006,24(1):55-56.

[2] 徐宝文,张卫丰. 搜索引擎与信息获取技术[M]. 北京:清华大学出版社,2003.

[3] 赵立刚. 搜索引擎的研究与设计[D]. 长春:吉林大学,2005.

[4] Joachims T. A statistical learning model of text classification for support vector machines[C]//Proceedings of the ACM SIGIR Conference. [s. l.]:[s. n.],2001.

[5] Qiu F,Cho J. Automatic identification of user interest for personalized search[C]//Proceedings of the 15th International Conference on World Wide Web. [s. l.]:[s. n.],2006.

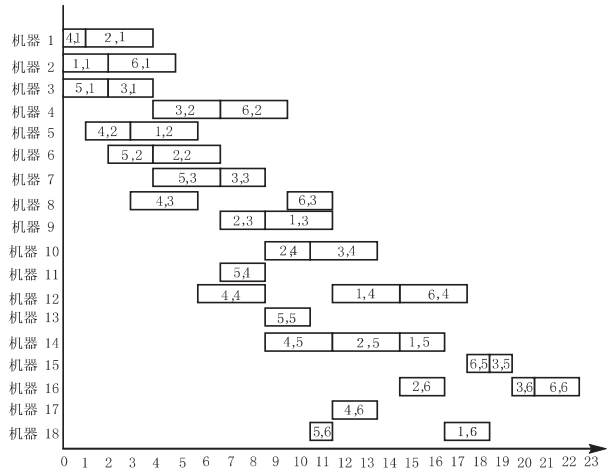
[6] 王小林,刘宏甲. 搜索引擎的设计研究[J]. 计算机技术与发展,2007,17(2):6-7.

[7] 刘 挺,秦 兵,张 宇,等. 信息检索系统导论[M]. 北京:机械工业出版社,2008.

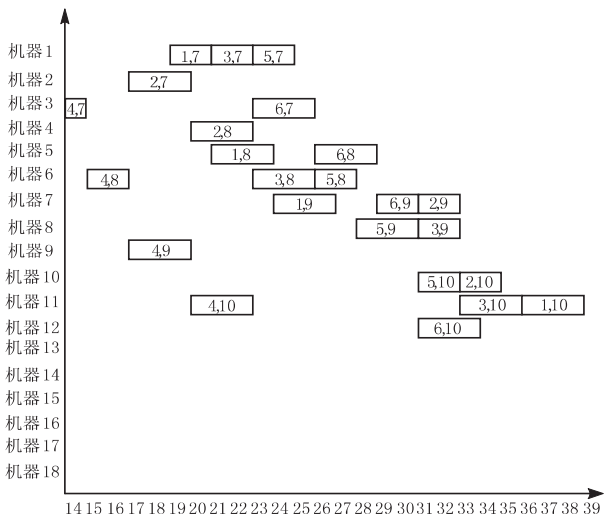
[8] 闰宏飞,孟 涛. 北大天网报告:2002年底中国网页知多

(下转第 209 页)

知,为了保证加工时间尽可能地短,需满足加工任务分配尽可能均匀,工件的加工工序尽可能保持连续。从图 4 可知,在调度前期基本上满足了这些条件,但是随着加工的进行,后面加工工序任务分配稍有分散,这是混合制造车间的必然现象。



(a)前六道工序调度结果甘特图



(b)后四道工序调度结果甘特图(可重入工序)

图 4　调度结果甘特图

最后,对该算法连续运行 10 次,每次由运算结果得到的目标函数值:40、39、41、39、40、41、40、39、39、40。上述数据表明本算法在 10 次运行中有 4 次达到最小值,运行结果基本上趋于稳定,由于遗传算法本身的特点,可知本算法是稳定有效的。

4　结束语

文中针对航空发动机装配车间这一类可重入混合 Flowshop 问题提出了基于遗传算法的解决途径。文中采用随机矩阵法对染色体进行编码,这使得遗传操作变得简单,同时不会产生非法染色体,通过实验仿真结果表明遗传算法也可以较好地解决可重入混合 Flowshop 问题。虽然遗传算法可以解决文中提出的调度问题,但是问题的规模还比较小,复杂度还比较低,并不能保证随着问题规模的增加,算法依然保持较好的效果,同时遗传算法本身还存在一些缺陷。因此,在接下来的时间中,笔者将继续研究如何对算法作必要的改进,克服算法本身所存在的一些缺陷,用于解决实际情况下的可重入混合 Flowshop 问题。

参考文献:

[1] Kumar P R. Re-entrant lines[J]. Queueing Systems,1993,13(1-3):87-110.

[2] Brah S A,Hunsucker J L. Branch and bound algorithm for the flowshop with multiple processors[J]. European Journal of Operational Research,1991,51(1):88-99.

[3] Wittrock R J. An adaptable scheduling algorithm for flexible flow lines[J]. Operations Research,1988,36(3):445-453.

[4] Chen J S,Pan J C H,Lin C M. A hybrid genetic algorithm for the re-entrant flow-shop scheduling problem[J]. Expert Systems with Applications,2008,34(1):570-577.

[5] Holland J. Adaptation in Natural and Artificial Systems[M]. [s.l.]:MIT Press,1992.

[6] 孙承夏,郭　禾. 一种有效的遗传算法在重入式生产调度问题中的应用[J]. 软件,2010,31(11):62-67.

[7] 王　永,吴智铭,隋　义. 基于遗传算法的可重入半导体生产线的调度[J]. 计算机仿真,2007,24(12):247-250.

[8] 吴云高,王万良. 基于遗传算法的混合 Flowshop 调度[J]. 计算机工程与应用,2002,38(12):82-84.

[9] 冯碧琤,乔　非,王　坚. 基于遗传算法的半导体生产线方法研究[J]. 计算机工程,2005,31(13):145-147.

[10] 刘　民,吴　澄. 制造过程智能优化调度算法及其应用[M]. 北京:国防工业出版社,2008:43-48.

[11] 玄光男,程润伟. 遗传算法与工程优化[M]. 北京:清华大学出版社,2004:6-7.

[12] 刘小华,林　杰,邓　可. 基于遗传粒子群混合的可重入生产调度优化[J]. 同济大学学报,2011,39(5):726-730.

(上接第 204 页)

少?[N]. 计算机世界,2003-01-17.

[9] 沈贺丹,潘亚楠,邵良杉. 关于搜索引擎的研究综述[J]. 计算机技术与发展,2006,16(4):147-148.

[10] 丁兆贵,金　敏. 基于 Lucene 的个性化搜索引擎研究与实现[J]. 计算机技术与发展,2011,21(2):106-107.

[11] 王　坚,赵恒永. 专业搜索引擎中文分词算法的实现与研究[J]. 福建电脑,2005(7):55-57.

[12] Wen Kunmei, Lu Zhengding, Li Yuhua, et al. A Cooperative Schema between Web Server and Search Engine for Improving Freshness of Web Repository[J]. Wuhan University Journal of Natural Sciences,2006,11(1):11-14.