一种混合搜索算法在智能 Web 中的应用

樊同科,谢 勇
(西安外事学院 现代教育技术中心,陕西 西安 710077)

摘 要:一种好的智能搜索算法对智能 Web 应用是非常重要的。为了在智能 Web 应用中实现快速智能搜索且能有效地去除垃圾信息,首先介绍了 Lucene 开源系统,详细分析了 Lucene 的系统结构以及 PageRank 算法。按照 Lucene 的框架规范,将 Lucene 很好地嵌入到自己的搜索引擎中,利用爬虫从互联网上收集数据,使用目前流行的 Lucene 和 PageRank 搜索技术在收集的数据上进行了实例研究。研究表明若在 Lucene 搜索中添加 PageRank 分数,进行混合搜索排序时,相关性高的网页就会排到前面,从而有效提高在智能 Web 中搜索的准确率及效率。

关键词:智能 Web;Lucene;PageRank
中图分类号:TP39 **文献标识码**:A **文章编号**:1673-629X(2013)08-0220-03
doi:10.3969/j.issn.1673-629X.2013.08.056

Application of a Hybrid Search Algorithm in Intelligent Web

FAN Tong-ke, XIE Yong
(Modern Education Technology Center of Xi'an International University, Xi'an 710077, China)

Abstract: An efficient intelligent search algorithm is very important to intelligent Web. It is an emergent subject for intelligent Web to achieve fast intelligent search and effectively screen out waste messages. Lucene is introduced, system structure and PageRank are analysed in detail. Collect information on Internet by using Web crawler and conduct case analysis on such information with currently popular Lucene and PageRank search techniques. If PageRank score is added in Lucene search, highly relevant results will be shown on the top during combined search sequencing, which will significantly improve accuracy and efficiency of search in intelligent Web.

Key words: intelligent Web; Lucene; PageRank

0 引 言

随着 Internet 技术的发展,网络已经成为人们生活必不可少的一部分。面对网络上的海量信息,人们一般是利用搜索引擎在 Internet 中进行搜集、发现信息。然而大部分传统的 Web 应用都是非智能的。主要表现在系统做出反馈时不会考虑用户在此之前的输入和行为。当然,非智能不是说的 UI 设计,而是系统对给定的输入所做出的一成不变的反馈。智能 Web 应用指的是系统在做出反馈时会考虑到整个系统中所有的用户在不同时间的输入和行为,并对其他各种可能有用的信息加以利用[1]。智能的 Web 应用就需要有智能的搜索算法,目前流行使用的有 Lucene 和 PageRank 算法,但都各自有优缺点,研究分析此类智能算法为智能 Web 应用服务,已经成为一项迫切而重要的课题。

1 Lucene 技术研究

1.1 Lucene 简介

Apache Lucene 最初是由 Doug Cutting 创立的[2],是一个基于 Java 的全文搜索引擎工具包。Lucene 提供了一组 API,利用它可以轻易地为 Java 软件加入全文索引和搜寻功能,以方便开发人员在现有系统中嵌入全文检索服务,并且索引和搜索的效率比传统的逐字比较大大提高。除此之外,它可以使用户随时根据自己的需要实现自订功能,从而实现一个完整意义上的搜索引擎系统[3]。

1.2 Lucene 框架组成

Lucene 全文检索引擎的系统结构中运用了面向对象的设计思想,它定义的索引文件格式与平台无关,并将系统的核心部分和具体的平台部分设计为抽象

收稿日期:2012-11-05 **修回日期**:2013-02-21 **网络出版时间**:2013-04-22
基金项目:陕西省教育科学"十二五"规划 2012 年课题(SGH12534);陕西省 2012 年度自然科学基础研究计划项目(2012JM8045);2012 年西安市社会科学规划基金项目(12IN32)
作者简介:樊同科(1979-),男,陕西扶风人,讲师,硕士,研究方向为数据仓库、数据挖掘。
网络出版地址:http://www.cnki.net/kcms/detail/61.1450.TP.20130422.1727.059.html

类,与平台相关的文件存储类操作也封装为类,从而形成一个容易二次开发的检索引擎系统。系统结构图如图 1。

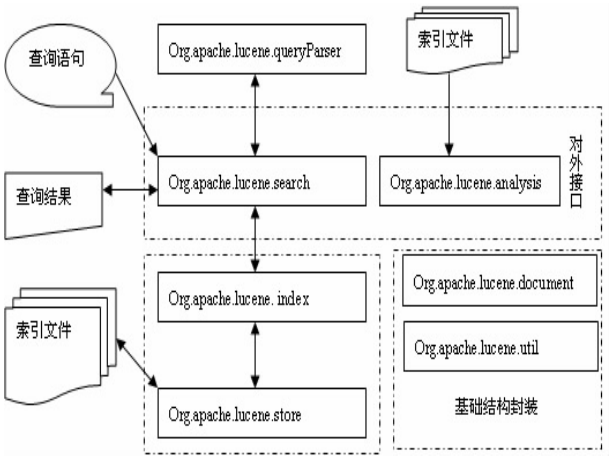

图 1　Lucene 的系统结构图

Lucene 系统由基础结构封装、索引核心、对外接口三大部分组成。其中索引核心部分是系统的重点。在 Lucene 的核心类包中含有以下三个子包:org. apache. lucene. analysis;org. apache. lucene. Index;org. apache. lucene. Search。其中 queryPaser 包是做为 Search 包的语法解析器存在,没有当作对外接口看待。

Lucene 在系统结构上的一个特点就是充分地从面象对象的观点来引入额外的抽象层以降低耦合性,使得 Lucene 的实现容易理解,易于扩展。

Lucene 在系统结构上的另一个特点就是 Lucene 开放源代码的特征。其表现为 Lucene 引入了新的应用结构,使得 Lucene 被作为应用本身的运行库,而不是一个单独的索引服务器存在[4]。

1.3　Lucene 检索机制

在搜索引擎的使用中,用户体验最深的是搜索部分,直接决定用户对搜索引擎的满意程度。Lucene 功能强大的索引机制就是为检索机制服务的。检索过程如图 2 所示。

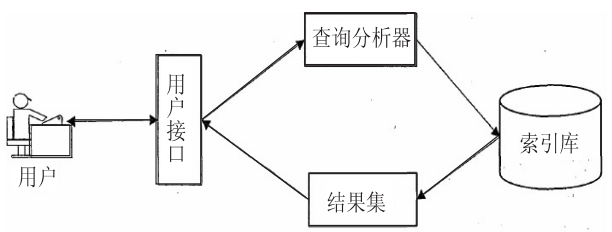

图 2　Lucene 检索机制架构图

所要搜索的数据可以在数据库中、互联网上,或者应用可以访问的任何其他网络中。也可以用爬虫从互联网收集数据,这些数据经过整理,相当于是一个简化的互联网。Lucene 可以帮助人们分析、索引并搜索这些文档,用来快速读取和存储网页的类叫 FethAndProcessCrawler,这个类还可以用于从互联网采集数据。

Lucene 在搜索时以文档的 id 排序,这样做的好处是多关键词查询时,处理简单。文件定位快速,倒排压缩高效。Lucene 还具有如下突出的优点[5]:

1)索引文件格式独立于应用平台。

2)在倒排索引的基础上,实现了分块索引和优化索引,提升了索引速度。

3)优秀的面向对象的系统架构,降低了 Lucene 学习和扩展的难度。

4)设计了独立于语言和文件格式的文本分析接口。

5)具有强大的查询引擎,用户无需自己编写代码即可使系统实现强大的查询能力。

但是,它的一个致命的缺陷在于:当某个期间的倒排很长时,在处理一次搜索时,系统需要对倒排所有元素都进行处理。这样的代价是不可接受的。这就注定了 Lucene 不适合海量数据的检索。同时,频繁更新的的数据将使 Lucene 对磁盘的输入与输出影响巨大。Lucene 的增量索引是通过它的 merge 算法来实现的。而该 merge 算法导致频繁的磁盘操作。一个新的数据的更新,可能导致一部分根本没有变化的索引被重写很多次,造成了搜索性能的下降。

2　PageRank 技术

2.1　PageRank 简介

PageRank 算法最早出现在 Sergey Brin 和 Larry Page 于 1998 年在世界万维网大会上发表的题为“The anatomy of a large-scale hypertextual Web search engine”的论文中[6]。正是 PageRank 技术使得 Google 从无名小辈变成为全球网络搜索巨头。

2.2　PageRank 算法

PageRank 算法是基于链接结构的算法[7]。在链接结构中有两个关键概念:权威页面和中心页面。如果一个页面地址被多个页面引用,就表明该页面有较高的认可度和较好的参考价值,就称该页面为权威页面。如果一个页面上具有较多的其他页面的链接地址,就像一个枢纽一样,就称该页面为中心页面或枢纽页面[8]。PageRank 算法在利用爬虫收集数据时,该算法会提取网页之间的链接结构,根据该结构会发现一些重要性较高的链接,来指导爬虫下一步的爬取。PageRank 算法就是通过网页链接结构来计算其中各网页的重要性的[9]。

PageRank 值的计算是基于下面两个假设的[10]:

1)用户浏览完页面 A 后,打开页面 A 上的其他链接进入其他页面的概率假设为 $1-d$,即用户不从页面 A 出发,重新打开一个新页面的概率为 d;此时的 d 称作阻尼因子。

2)假设用户只会从页面 A 出发向前浏览页面或重新打开一个新页面,不会再看以前浏览过的页面。

如今假设有三个页面 L、M、N 均包含页面 A 的链接,而页面 A 中又包含了页面 B、C、D 的链接地址。此时页面 A 的 PageRank 值计算如下[11]:

$$LN(A) = d + (1-d)\left(\frac{LN(L)}{C(L)} + \frac{LN(M)}{C(M)} + \frac{LN(N)}{C(N)}\right) \quad (1)$$

其中的 L、M、N 为含有页面 A 的页面。d 的取值一般为 0.15。其中的 $LN(L)$、$LN(M)$、$LN(N)$ 是页面 L、M、N 的 PageRank 值。$C(L)$ 是页面 L 含有的链接数。

PageRank 算法的核心是幂方法,幂方法中的 alpha 值在 0.7 ~ 0.9 之间一般可以做到兼顾应用效果和效率,Google 使用的 alpha 值为 0.85。有些技术可以加速幂方法的收敛,还有一些完全不依赖于幂方法的手段,称之为直接方法,它更适合于一些较小的网络[12]。

3 基于 Lucene 和 PageRank 相结合的搜索算法

利用爬虫从互联网上收集一些数据,并在其中加入三个垃圾网页(名为 spam-biz-0x.html,其中 x 为一个数字),这些垃圾网页可以瞒过基于索引的搜索,但却欺骗不了 PageRank。利用 Lucene 在这些网页中搜索“nvidia”的代码如下:

```
FetchAndProcessCrawler crawler =
new FetchAndProcessCrawler("c:/iWeb2/data/",5,200);
crawler.setUrls("biz");
crawler.addUrl("file:///c:/iWeb2/data/spam-biz-01.html");
crawler.addUrl("file:///c:/iWeb2/data/spam-biz-02.html");
crawler.addUrl("file:///c:/iWeb2/data/spam-biz-03.html");
crawler.run();
LuceneIndexer luceneIndexer = new LuceneIndexer(crawler.getRootDir());
luceneIndexer.run();
```

执行后,结果如表 1 所示。

在基于 Lucene 搜索“nvidia”的代码基础上,增加 PageRank 搜索,代码如下:

```
PageRank pageRank = new PageRank(crawler.getCrawlData());
pageRank.setAlpha(0.99);
pageRank.setEpsilon(0.00000001);
pageRank.build();
Mysearcher oracle = new MySearcher(luceneIndexer.getLuceneDir());
oracle.search("nvidia",5,pageRank);
```

表 1 基于 Lucene 的搜索结果

```
bsh % oracle.search("nvidia",5,pr);
Search results using Lucene index scores:
Query:nvidia
Document Title:NVIDIA shares plummet into cheap medicine for you!
Document URL:file:/c:/iWeb2/data/spam-biz-02.html
Relevance Score:0.519243955612183
Document Title:NVIDIA shares up on PortalPlayer buy
Document URL:file:/c:/iWeb2/data/biz-05.html
Relevance Score:0.254376530647278
Document Title:NVIDIA Now a Supplier for mp3 Players
Document URL:file:/c:/iWeb2/data/biz-04.html
Relevance Score:0.190782397985458
Document Title:Chips Snap:Nvidia,Altera Shares Jump
Document URL:file:/c:/iWeb2/data/biz-06.html
Relevance Score:0.181735381484032
Document Title:Economic stimulus plan helps stock prices
Document URL:file:/c:/iWeb2/data/biz-07.html
Relevance Score:0.084792181849480
```

执行后,结果如表 2 所示。

表 2 基于 Lucene 和 PageRank 混合的搜索结果

```
Search results using combined Lucene scores and pagerank scores:
Query:nvidia
Document URL:file:/c:/iWeb2/data/biz-04.html
Relevance Score:0.087211910261991
Document URL:file:/c:/iWeb2/data/biz-06.html
Document URL:file:/c:/iWeb2/data/biz-05.html
Relevance Score:0.062737066556678
Document URL:file:/c:/iWeb2/data/spam-biz-02.html
Document URL:file:/c:/iWeb2/data/biz-07.html
Relevance Score:0.000359708275446
```

4 结束语

从以上两个结果可以看出,如果只使用 Lucene 搜索,会出现垃圾网页在结果中排名第一。但当添加进 PageRank 分数,进行混合搜索排序时,相关性高的网页则排到了前面,垃圾网页则回到了它应该的位置。由此两者结合进行混合搜索排序时提高了在智能 Web 中搜索的准确率及效率。

参考文献:

[1] Marmanis H,Babenko D. 智能 Web 算法[M]. 阿 稳,陈钢,译. 北京:电子工业出版社,2011.

[2] 索红光,孙 鑫. 基于 Lucene 的中文全文检索系统的研究与设计[J]. 计算机工程与设计,2008,29(19):5083-5086.

[3] 朱学昊,王儒敬,余锋林,等. 基于 Lucene 的站内搜索设计与实现[J]. 计算机应用与软件,2008,25(10):6-8.

[4] 张正龙. 基于 Lucene 的主题搜索引擎研究与实现[D]. 重

(下转第 226 页)

能使用算术平均的方法,可以考虑采用与时间有关的加权平均值,也可以进行分时统计,这里采用后者。通过 FlowG [24]数组存放每一个小时的流量统计值,每个时间段的阈值采用略大于该时间段内平均流量的方法。

其实现过程主要包含三个部分:第一步,周期性发送查询相关 OID 命令;第二部,设置异步接收函数并处理接收到的数据;第三部,进行阈值判断,如果流量超过上限启动流动控制服务。以下列出了实现过程中三大部分的主要语句,采用 Visual C++环境下的 WinSNMP API 实现,其核心代码如下:

```
pSnmp.CreateVbl(OID,NULL);
for(int i=2;i<=6;i++)
{
pSnmp.SetVbl(m_initOid[i]);
}
pSnmp.CreatePdu(SNMP_PDU_GET,NULL,NULL,NULL);
pSnmp.Send(IPString,"public");
……
smiINT sNumber;
sNumber=m_value[i]->value.sNumber;
nIpin=sNumber;
wsprintf(str[i],"%d",sNumber);
NodeFlow=(ifIOOctetsL- ifIOOctetsH)/Dt;
……
if(NodeFlow>Fg)
CallFlowServer(N[i]);
……
```

4 结束语

针对当前共享带宽的园区网络中存在异常流量导致网络节点部分端口形成速度瓶颈,从而导致访问速度变慢的问题,利用 SNMP 良好的通用性和丰富的监控功能,结合异常流量的特点,构建了一种能够实时对网络重要节点端口速度进行监控的模型,其实现主要包括两部分,其一,通过 SNMP 协议访问监控节点的 MIB,实现对节点流量的监控;其二,对监控结果设置阈值,监控流量超过门限时进行流量的管控。该模型设计可以有效评价网络的运行状态,当出现流量异常时可以及时地采取措施,避免了传统的网络防火墙进行流量控制过程中数据分析量过大造成数据延迟的问题。该模型设计也存在不足之处:监控的网络规模不宜过大。在大规模网络下可以考虑采用分布式的网络管理模式结合控制模型的方法,这也是需要进一步研究的问题。

参考文献:

[1] 张 彤,吴世荣. 基于 SNMP 计算机网络流量监控系统研究[J]. 计算机技术与发展,2011,21(1):88-91.

[2] 马华林,李翠凤,张立燕. 基于灰色模型和自适应过滤的网络流量预测[J]. 计算机工程,2009,35(1):130-131.

[3] 吴烨虹,张少娴. 网络分析仪在网络流量监测中的应用[J]. 计算机技术与发展,2012,22(8):237-240.

[4] 严斌宇,刘方圆,吴少华. 基于 SNMP 的网络管理软件的设计与实现[J]. 计算机与数字工程,2012,40(4):126-129.

[5] 陆飞跃. 网络流量控制系统的分析及实现[D]. 北京:北京邮电大学,2010.

[6] Chang Yanan, Xiao Debao, Chen Limiao. Design and Implementation of NETCONF-Based Network Management System [C]//Proc. of 2008 Second International Conference on Future Generation Communication and Networking. [s. l.]:[s. n.],2008:256-259.

[7] 赵永胜. MRTG 在网络管理中的应用[J]. 铁路通信信号工程技术,2005(6):43-45.

[8] Choi M,Choi H,Hong J W. XML-based Configuration Management for IP Networks Devices[J]. IEEE Communications Magazine,2004,42(7):84-91.

[9] Schonwalder J,Pras A,Martin-Flatin J P. On the Future of Internet Management Technologies[J]. IEEE Communication Magazine,2003,41(10):90-97.

[10] 蔡 琳. 在 VC++6.0 平台下基于 SNMP 网络管理软件的开发[J]. 信息与电子工程,2005,3(3):224-227.

(上接第 222 页)

庆:重庆大学,2008.

[5] 李建林. 基于 Lucene 的 Web 搜索引擎的研究[D]. 兰州:兰州理工大学,2010.

[6] 邓 攀,刘功申. 一种高效的倒排索引存储结构[J]. 计算机工程与应用,2008,44(31):149-152.

[7] 李晓明, 闫宏飞, 王继民. 搜索引擎-原理、技术和系统[M]. 北京:科学出版社,2006.

[8] 李远方,邓世昆,闻玉彪,等. Hadoop-MapReduce 下的 PageRank 矩阵分块算法[J]. 计算机技术与发展,2011,21(8):6-9.

[9] Wu Hengliang,Zhang Weiwei. An Improved Page Ranking Algorithm for Web Search Engine[J]. International Journal of Digital Content Technology and Its Applications, 2012, 13(6):38-44.

[10] 刘青伟. 搜索引擎中的 Pagerank 排序算法研究分析[D]. 成都:电子科技大学,2010.

[11] Han Min,Zhang Xianchao. Community Identification Based on a New Approximate Personalized PageRank Algorithm[J]. Advances in Information Sciences and Service Sciences,2012,20(4):649-657.

[12] 李广丽,刘觉夫. 垂直搜索引擎系统的研究与实现[J]. 情报杂志,2009,28(10):144-147.

一种混合搜索算法在智能Web中的应用

作者： 樊同科， 谢勇， FAN Tong-ke， XIE Yong
作者单位： 西安外事学院 现代教育技术中心,陕西 西安,710077

刊名： 计算机技术与发展

ISTIC

英文刊名： Computer Technology and Development

年，卷(期)： 2013(8)