基于Lucene的全文检索系统的研究与实现

周锦程[1],王 丹[1],余 泉[1,2],张 维[1]
(1.黔南民族师范学院 数学系,贵州 都匀 558000;
2.中山大学 信息科学与技术学院,广东 广州 510275)

摘 要:Lucene是一个优秀的开源全文搜索技术框架,Lucene全文检索技术是信息检索领域广泛使用的基本技术。它能非常方便地为各种应用程序加入全文索引和搜索功能,快速有效地索引企业累积的大量信息资源。文中阐述了建立全文检索系统的必要性,介绍了全文检索系统的概念并分析了Lucene的系统结构和实现机制,最后结合实际应用背景从系统设计、文挡抽取、索引的建立及执行检索等方面介绍了全文检索系统的实现过程。实践证明,该系统的查准率、查询速度等均达到了设计要求。

关键词:全文检索;索引;信息检索
中图分类号:TP311.5 **文献标识码**:A **文章编号**:1673-629X(2011)03-0067-05

Research and Implementation of Full-Text Retrieval Engine Based on Lucene

ZHOU Jin-cheng[1], WANG Dan[1], YU Quan[1,2], ZHANG Wei[1]
(1. Department of Mathematics, Qiannan Normal College for Nationalities, Duyun 558000, China;
2. School of Information Science and Technology, Sun Yat-Sen University, Guangzhou 510275, China)

Abstract: Lucene is an excellent technology frame of full-text retrieval engine of open source code. Lucene full-text retrieval technology is a basic technology used widely in information retrieval field. It is very convenient for various applications by adding full-text index and search functions, quickly and efficiently index the accumulation of large enterprise information resources. Expound the necessity of establishing full-text retrieval system, then introduce the concept of full-text retrieval system and analyze the structure of Lucene system and the implementation of the mechanism. Finally, give the implementation process of full-text retrieval system from the system design, the text extraction, the index establishment and the executive index search with a practical application. Practice shows that the system's precision ratio and speed can satisfy the design demands well.

Key words: full-text retrieval; index; information retrieval

1 概 述

近年来,信息技术的快速发展加快了企业信息化的进程,同时也促进了企业的发展;随着企业信息的大量增加,电子文档数目也急剧膨胀,如何在海量信息中迅速、准确、全面地查找企业所需的资料信息已成为信息检索研究领域内的一个热门课题[1]。全文检索技术是信息检索的核心技术,它是指计算机通过索引程序来扫描电子文档中的每一个词,并对其建立相应索引,指明该词在文档中出现的位置和次数。当用户进行查询时,检索程序根据已经建立的索引进行查找,并将查找的结果反馈给用户[2],全文检索能帮助人们进行大量文档资料的管理和整理等工作。

通常的数据库全文检索系统是将企业内部各部门的文件资料汇总到数据库,并利用数据库提供的全文检索功能统一向外提供流文件的全文检索服务。但这种为了满足全局检索而要求将所有资料汇总到数据库的方案会加大系统额外的开销。另外,这种解决方案对服务节点的处理能力和网络带宽要求很高,当大量用户并发访问时服务器的服务质量会明显降低,甚至会导致系统崩溃。同时,商业数据库对于流文件全文检索的功能支持也非常有限。

虽然大型桌面搜索引擎功能已经越来越强大,并且很多站点都使用了Google站内检索代替了自己的站内数据库"全文"检索,但依靠Google这样的大型搜索引擎做全文检索会有以下弊端[3]:

(1)数量有限:搜索引擎并不会深度遍历个网站而将站点所有内容都索引进去,Google甚至会定期将

收稿日期:2010-07-23;**修回日期**:2010-10-25
基金项目:贵州省自然科学基金资助项目(黔教科2008090)
作者简介:周锦程(1981-),男,贵州开阳人,硕士,讲师,CCF会员,研究方向为软件开发方法。

缺少入口站点内容逐渐抛弃；

(2)更新慢：搜索引擎针对站点更新频率也是定周期的，很多内容需要一定时间后才能进入 Google 索引，目前 GoogleDance 的周期是 21 天左右；

(3)内容不精确：搜索引擎需要通过页面内容提取技术将导航条页头页尾等内容过滤掉反而不如直接从后台数据库提取数据来得直接；

(4) 无法控制输出：也许有更多输出需要按时间排序、按价格、按点击量、按类目过滤等。

2 全文检索系统以及 Lucene 全文检索引擎介绍

2.1 全文检索系统

全文检索系统是指按照全文检索理论建立起来的用于提供全文检索服务的软件系统。全文检索系统的核心应具有建立索引、优化索引结构、处理查询返回结果集、增加索引等功能。此外还应具有方便的用户接口和二次应用开发接口、文本分析引擎、查询引擎、索引引擎等等。

2.2 开源全文检索引擎 Lucene 介绍

Lucene[4] 是 Apache 软件基金会 Jakarta 项目组的子项目，它是一个采用 Java 语言实现的可伸缩、高性能的信息搜索库，提供了完整的索引引擎和查询引擎，同时还为数据访问和管理提供了简单的函数调用接口，便于在目标系统中建立起完整的全文搜索引擎或实现全文检索功能。但 Lucene 并不是完整的搜索应用程序，因此针对不同的应用背景还需进行相应的二次开发[5-10]。目前有很多著名项目都使用 Lucene 作为其后台全文检索引擎，如 Web 论坛系统 Jive、邮件列表系统 Eyebrows、机构知识库 DSpace、Java 开发平台 Eclipse、基于 XML 的 Web 发布框架 Cocoon、苹果的 iTunes、微软的 Outlook 搜索插件等[11]，采用 Lucene 作为全文检索引擎，具有如下五方面的优点：[12]

(1)在倒排索引的基础上，实现了分块索引，能针对新文件建立增量或者小批量索引，使索引速度得到了提升，并可通过合并原有索引达到优化索引的目的。

(2) 索引文件格式与应用平台独立。Lucene 定义的索引文件格式以 8 位字节为基础，使其能在不同平台的应用或不同系统中共享建立的索引文件。

(3) 支持多语言，也可编写解析器(Analyzer)扩展支持其他语言。

(4) Lucene 采用独立于文件格式和语言的文本分析接口，索引器通过接收 Token 流来完成索引文件的创建，用户只需实现文本分析接口就可实现新的语言和文件格式的支持。

(5) 采用基于组件式的面向对象的系统架构，降低了对 Lucene 的扩展难度，易于在其基础上扩充新的功能。Lucene 提供的接口函数功能强大，接口简单。

2.2.1 Lucene 的体系结构

作为一个优秀的全文检索引擎，Lucene 的系统结构采用了面向对象的设计思想。首先定义一个与平台无关的索引文件格式，然后通过抽象将系统的核心组成部分设计为抽象类，平台实现部分则设计为抽象类的实现，而与具体平台相关的部分如文件存储也封装为类，经过分层的面向对象式的处理，最终达成了一个高效率、低耦合、易于二次开发的检索引擎系统。图 1 为 Lucene 的系统结构[13]。

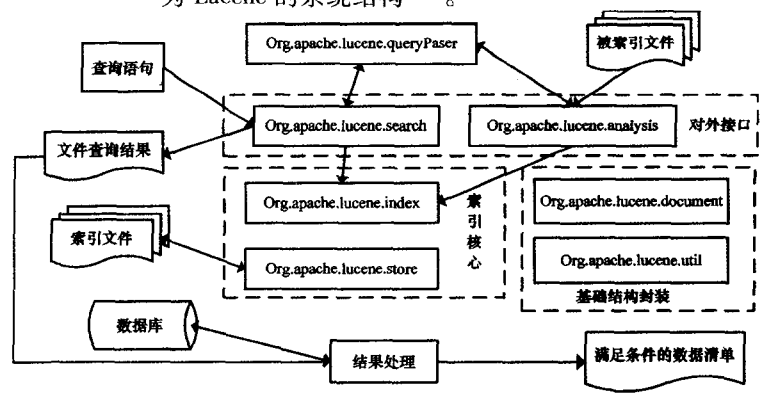

图 1 Lucene 系统结构

从图 1 可看出，Lucene 系统主要由索引核心、基础结构封装与对外接口三部分组成，其中重点是直接操作索引文件的索引核心。Lucene 系统源码主要由 7 个包组成，各个包完成特定的功能，表 1 中列出了其中核心的 7 个包及其特定的功能说明。

表 1 Lucene 核心包功能

核心包名	功能说明
Org. apache. lucene. analysis	主要用于分词的语言分析器，可以通过扩展此类来支持中文
Org. apache. lucene. index	包括建立、删除、修改索引等
Org. apache. lucene. document	存储索引时的文档结构管理
Org. apache. lucene. store	主要用于数据存储管理，包括相关的底层的 I/O 操作
Org. apache. lucene. util	相关的公用使用类
Org. apache. lucene. search	用于检索的相关核心方法
Org. apache. lucene. query Paser	查询分析器：实现查询关键词之间的运算

2.2.2 全文检索的实现机制

Lucene 实现复杂，功能强大，主要包括两个主要功能：一是建立索引库，即将待索引的纯文本内容切分词后索引入库；二是检索索引库，即根据查询条件从索引库中找出符合查询条件的相应文档。图 2 是 Lucene

全文检索的实现机制[14]。

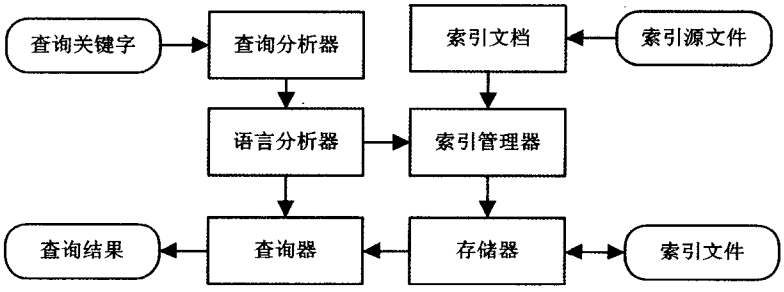

图 2　全文检索的实现机制

3　企业信息系统中全文检索系统的设计与实现

3.1　全文检索系统框架设计

笔者参与开发的某军用数据库信息系统是基于 B/S 的网络数据库系统,为某工程建设项目的数十家成员单位提供公共/行业数据库应用服务,能够将该领域的公共与行业数据库内容纳入管理,为该行业提供较为全面的公共与行业数据服务。根据需求,其中的公共专业期刊数据库和行业文献数据库要具备全文检索和按检索项检索的功能。文中仅介绍公共专业期刊数据库的全文检索的实现,行业文献数据库的全文检索实现方式与此类同。

公共专业期刊数据库存储的期刊主要包括文献标题、主题、摘要、作者、作者单位、文献出处等属性,存储包括. pdf,. doc,. html,. txt,. rtf 等多种格式的期刊文件。

下面从文档抽取、对抽取的文档进行索引、对索引文件进行检索及检索结果显示以及与用户交互等功能的实现做详细介绍。

3.2　系统实现

3.2.1　文档抽取

Lucene 只定义了一个抽象文档的结构 Document,没有定义具体的数据源,因此在各种应用中,只要采用合适的转换器把数据源转换成相应的 Document 结构即可,对于. pdf,. doc,. html,. txt,. rtf 等格式文件均可方便地借助于第三方解析器来进行转换,比如用 HTMLParser 组件可抽取. html 格式文档,POI 组件可抽取. doc 格式文档,Xpdf 组件及其中文补丁包可抽取. pdf 格式的文本,而. txt 格式文件可直接使用 Java 的字符流来读取等等。为使系统能统一处理多种格式文档,采用接口实现和动态实例化的方法为用户屏蔽了各种文档格式间的差异性,使其具有统一处理多种格式文档的能力。下面是解析. doc 格式文件的部分代码。

```
public class DocDocumentHandler implements DocumentHandler {
    public Document getDocument(InputStream is) throws Exception {
        String bodyText = null;
        try {
        WordDocument wd = new WordDocument(is);
        StringWriter docTextWriter = new StringWriter();
        wd.writeAllText(new PrintWriter(docTextWriter));
        docTextWriter.close();
        bodyText = docTextWriter.toString();
        } catch (Exception e) {//异常处理略);
        }
        if ((bodyText != null) && (bodyText.trim().length() > 0)) {
        Document doc = new Document();
        doc.add(Field.UnStored("body", bodyText));
        //内容只索引不存储
        return doc;
        }
        return null;
    }
}
```

Lucene 的一大优点是可以由程序员控制 Document 中 Field 的分词与存储。在公共专业期刊数据库中,对于文件名 Field 需要存储和索引,而文件的内容 Field 通过文件路径来读取,所以不需要存储,而需要分词和索引。

3.2.2　建立索引

Lucene 建立索引主要有以下两步,即建立索引器和添加索引文件,建立索引器:IndexWriter writer = new IndexWriter(文件路径,分析器, 增量/重建);添加索引文件 writer.addDocument(doc);为了使删除、更新文件时系统能同步检索到文件,文中建立索引时采取清除以前建立的索引重新建立索引,但这样的同步索引方式在时间开销上较大,因此本系统采取的方法是:在数据库中用一个字段来存储索引标志(称其为索引开关,其值可设置为 1:表示索引开关打开,0:表示索引开关关闭),索引开关与建立索引的操作界面如图 3 所示。当系统管理员登录时,系统自动检查索引开关状态,并将当前的值放在 HttpSession 中;新增、修改、删除文件时检查索引开关是否打开(通过 HttpSession 取得其值),若为打开,则同步建立索引,否则,不建立索引(系统管理员也可以在某一时刻点击"建立索引"按钮来为系统中的所有文件重新建立一次索引)。

3.2.3　检索文献

系统将提供包括按指定字段进行检索和全文检索

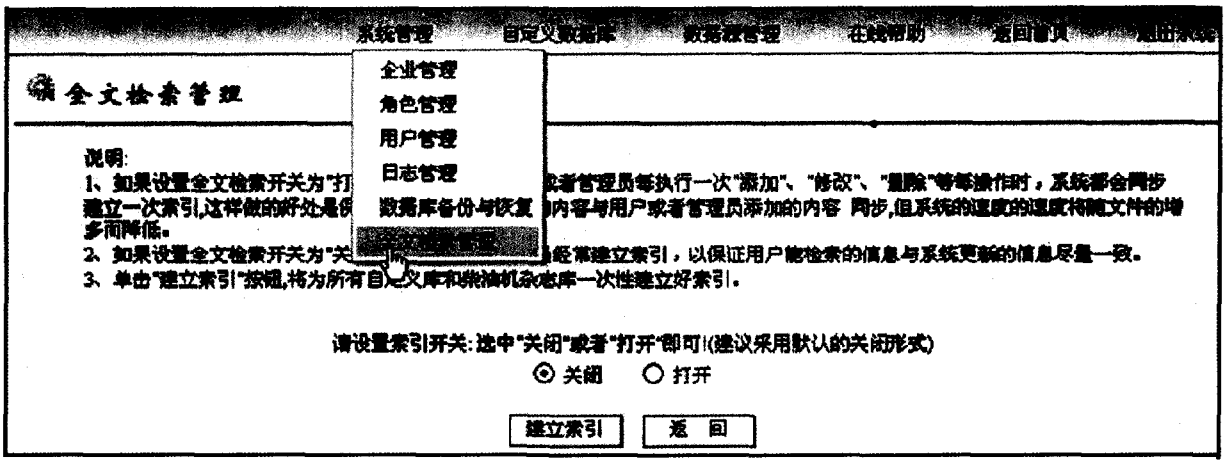

图3 索引开关与建立索引的操作界面

两种检索方式,其中按指定字段的检索方式即在数据库对应字段中检索匹配的内容,这里不再赘述。Lucene提供给用户的检索接口是IndexSearcher类,由Query类封装查询字段查询词、和语言分析器,交由IndexSearcher类查询。检索结果返回的是Hits对象。为保证检索结果的准确性和检索效率,本系统对应的数据库中仅保存该文献的文件名,文件保存在服务器端文件夹中,在添加或修改某一具体文献信息时,检验该文献对应的文件名是否在服务器端已经存在,若已存在,则在文件保存到服务器端前将该文件的文件名前加上一个五位数的随机数,也确保该文件名唯一。

当通过Lucene提供给用户的检索接口检索到含有检索词的文件时,接口程序将数据库中对应文件名的所有记录找到,存放在List对象变量中,因此可根据List中的各文件名得到文件的映射地址并提供给用户下载。以下为检索结果处理的部分关键代码:

```
//如果执行了全文检索
String searchvalue = queryform.getSearchvalue();
List<PublicPodDL> indexresult = new ArrayList<PublicPodDL>();
try {
ServletContextservletContext = this.getServlet().getServletContext();
String indexdir =servletContext.getRealPath("/") + "upload\\"+
PublicPodDLIndex";// indexdir 为索引目录
//执行检索
List hits=new ArrayList();
try{
hits = Searcher.dosearch(searchvalue, indexdir);
}catch (Exception e) {//异常处理;}
}
// indexresult 用于存放含检索关键字 searchvalue 的所有文件名
for (int i = 0; i < hits.size(); i++) {
StringTindex = null;
// Tindex t 用于存放当前含检索关键字 searchvalue 的文件名
Tindex = ((IndexFileName) hits.get(i)).getFilename();
// 通过文件名查找所有实体,并添加在 List 中
ListlistResult = PUBPodDLService.findByFileName(Tindex);
if (listResult.size() > 0) {
indexresult.add((PublicPodDL) (listResult.get(0)));
}
}
public class Searcher {
public static List dosearch(String keyword, String indexdir) throws Exception {
File indexDir =new File(indexdir);
String q = keyword;
if (! indexDir.exists() || ! indexDir.isDirectory()) {
throw new Exception(indexDir
+ " 目录不存在或者不是目录.");}
Directory fsDir=FSDirectory.getDirectory(indexDir, false);
IndexSearcher is=new IndexSearcher(fsDir);
Query query=QueryParser.parse(q, "body",new ChineseAnalyzer());
//在"body"中查找,已在创建索引中定义
//其中:query 为检索词, field 为检索的字段名, analyzer 为分析器
Hits hits = is.search(query); // search
List<IndexFileName> listResult=new ArrayList<IndexFileName>();
for (int i = 0; i < hits.length(); i++) {
IndexFileName Tindex = new IndexFileName();
Document doc = hits.doc(i);
Tindex.setFilename(doc.get("filename"));
// 通过文件名查找所有实体,并添加在 List 中
listResult.add(Tindex);
}
fsDir.close();
is.close();
```

```
return listResult;
}
}
```

4 结束语

文中介绍了优秀的开源全文检索引擎 Lucene,并在其上根据某军用信息系统的特点和要求进行了二次开发,按照 Lucene 的框架规范,将 Lucene 很好地嵌入到了该系统中,实践证明,该系统达到了预期的目标,全文检索效率高,准确率和查全率都达到了设计的要求。文中提出的解决全文检索系统的方法,也可方便地应用到其他企业的信息系统中,能有利地提高企业对目标文档的管理水平与检索能力。

参考文献:

[1] 林碧英,赵　锐,陈良臣. 基于 Lucene 的全文检索引擎研究与应用[J]. 计算机技术与发展,2007,17(5):184-185.

[2] 吴　青,夏红霞,赵广辉,等. 基于 Lucene 全文检索引擎的应用与改进[J]. 武汉理工大学学报,2008,30(7):145-146.

[3] 周　宁,谷宏群. 基于 Lucene /XML 的全文检索机制研究[J]. 图书·情报·知识,2005(6):75-76.

[4] Hatcher E, Gospodnetic O. Lucene in action[M]. [s. l.]: Manning Publications Co. ,2005.

[5] 阳　奇,林镇灿,黄　帆,等. 基于 Hibernate 搜索的数据库全文检索系统[J]. 计算机工程,2010,36(4):73-76.

[6] 徐爱春,魏艳华,何震晏. 基于 Lucene 的电子政务全文检索系统的设计与实现[J]. 现代情报,2008,7(7):223-225.

[7] 叶　靓,王智斌,邵谦明. 基于相关反馈的语音检索引擎[J]. 计算机工程,2007,33(17):228-230.

[8] 蒋一峰,王　华,张玉红,等. 基于 Lucene 的语义检索系统的设计和实现[J]. 计算机工程与设计,2008,29(20):5336-5337.

[9] 高　琰,谷士文,谭立球,等. 基于 Lucene 的搜索引擎设计与实现[J]. 微机发展(现更名:计算机技术与发展),2004,14(10):27-30.

[10] 张雪燕,杨晟刚,黄　文,等. 企业级搜索引擎技术在博客网站中的应用[J]. 计算机工程与设计,2008,29(18):4856-4858.

[11] 赵　汀,孟祥武. 基于 Lucene API 的中文全文数据库设计与实现[J]. 计算机工程与设计,2003(20):179-181.

[12] 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.

[13] 张校乾,金玉玲,侯丽波. 一种基于 Lucene 检索引擎的全文数据库的研究与实现[J]. 现代图书情报技术,2005(2):41-42.

[14] 管建和,甘剑峰. 基于 Lucene 全文检索引擎的应用研究与实现[J]. 计算机工程与设计,2007,28(2):490-491.

(上接第66页)

3 结束语

生长型自组织神经网络的聚类过程是无监督、自学习和自组织的,不需要事先确定聚类中心。相对于 SOM 网络而言,其优越性在于:(1)不需要预定义网络的大小,而是通过神经元的增长来逐渐扩张网络直至达到某种性能或规模大小;(2)该种网络的各项参数都是固定值,不会影响到网络的收敛性,当然参数的取值是否合理会影响到最后的聚类结果,因此需要重复调整参数值的大小来不断观察聚类结果;(3)生长型自组织神经网络结构可以灵活调整,除了可以新增神经元,也可以删除"无用"神经元,对未来数据的学习能力强,这样形成的聚类结果更加清晰。网络利用阈值控制的触发机制有效地实现了神经元的增删,为了进一步提高网络的快速学习能力,如何确定一个相对较优的权值调整系数将是下一步研究的重点。

参考文献:

[1] 陈学进. 数据挖掘中的聚类分析的研究[J]. 计算机技术与发展, 2006, 16(9): 44-49.

[2] 倪步喜, 章丽芙, 姚　敏. 基于 SOFM 网络的聚类分析[J]. 计算机工程与设计,2006, 27(5): 855-878.

[3] 徐　勇, 戴逸松, 陈贺新. 神经网络矢量量化的设计与实现[J]. 长春邮电学院学报,1996, 14(3): 1-4.

[4] 赵　鹏, 耿焕同, 蔡庆生, 等. 一种基于加权复杂网络特征的 K-means 聚类算法[J]. 计算机技术与发展,2007, 17(9): 35-37.

[5] Kohonen T. The Self-organizing Map [J]. IEEE, 1990, 78(9):1464-1480.

[6] 董长虹. Matlab 神经网络与应用[M]. 北京:国际工业出版社, 2007: 185-196.

[7] 傅　彦, 周俊临. 基于自增长型多级自组织映射网络的模式识别[J] 计算机科学, 2004, 31(5): 159-162.

[8] Fritzke B. Growing cell structures - a self-organizing network for unsupervised and supervised learning [J]. Neural Networks, 1994, 7(9): 1441-1460.

[9] 吴　郢, 阎平凡. 结构自适应自组织神经网络的研究[J]. 电子学报, 1999, 27(7): 55-58.

[10] Fritzke B. Growing cell structures- a self-organizing network in k dimensions [M]// Artificial Neural Networks. [s. l.]: [s. n.],1992:1051-1056.

[11] 李　戈,邵峰晶,朱本浩. 基于神经网络聚类的研究[J]. 青岛大学学报, 2001, 16(4): 21-24.

[12] Fritzke B. Unspervised clustering with growing cell structures [C]//Proc. Int. Joint Conf. on Neural Networks. [s. l.]: [s. n.], 1991:531-536.