

西安市数字方志全文检索系统的设计与实现

吴代文¹, 詹海生²

(1. 渭南师范学院 传媒工程系, 陕西 渭南 714000;
2. 西安电子科技大学 计算机学院, 陕西 西安 710071)

摘 要:通过 Lucene API 实现对 PDF 文档的一次全文检索, 为了更精确地定位搜索关键词, 设计并实现了一种新的二次索引算法, 该二次索引带有关键词的页码、坐标及其上下文等信息。利用该二次索引可将检索结果定位到 PDF 文档的具体页, 然后在页面上标示出关键字的具体位置, 使对 PDF 文档的二次检索达到了类似 Google Book 的图书检索效果。系统测试结果表明系统具有良好检索性能, 有较高的查全率和查准率, 能够满足用户快速检索的需求。系统作为西安市数字方志全文检索平台投入使用已有 2 年, 取得了较好的应用成果。

关键词:全文检索; 二次索引; 二次检索; 查全率; 查准率

中图分类号: TP391.3

文献标识码: A

文章编号: 1673-629X(2011)10-0121-04

Design and Implementation of Full-Text Retrieval System for Xi'an Data Chorography

WU Dai-wen¹, ZHAN Hai-sheng²

(1. Department of Communication Engineering, Weinan Teachers University, Weinan 714000, China;
2. College of Computer Science, Xidian University, Xi'an 710071, China)

Abstract: In the paper, it implements the first index in PDF document by Lucene API. In order to locate the search keyword more accurately, this paper designs and implements a new algorithm for the second index. It contains the information about the keywords' page number, coordinates, context and so on. Which can be made used of locating the retrieval results in the specific page of the book and marking the specific positions of the keywords. Thus, the effect of the second retrieval in PDF document is as similar as Google Book. The test result proved that this system is provided with high retrieval performance, recall rate and precision rate. It can be satisfied with the requirement of quickly retrieving websites' documents. This system has been using for 2 years as the full-text retrieval system for Xi'an data chorography and it gets lots of application fruit.

Key words: full-text retrieval; second Index; second retrieval; recall; precision

0 引言

2008 年 4 月, 由西安电子科技大学网络教育学院申请的科研项目“西安市地情资料信息中心基础数据库”获得西安市信息办立项资助。同年 12 月项目成果通过西安市信息办验收, 下面将详细报告该项目实施过程和研究成果。

项目合同书规定了项目实施方(西安电子科技大学网络教育学院)的主要任务是: 将西安市地方志所有志书利用计算机技术扫描成具有高分辨率的 JPEG 图像, 并在进行文字转换后制作成 PDF 文档。然后利

用 Lucene API 建成全文检索系统, 实现对 PDF 文档的全文检索。该系统可以让浏览者通过互联网查阅和全文检索西安市地方志的全部电子志书, 基本上可以代替原始文献使用。这样既可以最大限度地保护原始纸质版志书, 又满足了社会各界人士阅读志书的需求^[1]。

1 关键技术分析

本项目的关键技术在以下几个方面。

1) 利用 Lucene API 实现对 PDF 文档全文检索, 这一功能需要经过一次索引和检索来实现。

2) 利用 PDFBox API 对 PDF 文档进行分析, 提取二次索引信息, 这是本项目最关键的核心理念。

3) 利用二次索引信息实现在 PDF 文档内容的二次检索。在进行文献二次检索时, 需要在文献具体页用特殊颜色标识出检索关键字, 最终达到的显示效果要和 Google 图书搜索接近。

收稿日期: 2011-03-18; 修回日期: 2011-06-11

基金项目: 教育部特色专业建设点(TS11772)

作者简介: 吴代文(1979-), 男, 硕士, 讲师, 主要研究方向为远程教育、教育信息检索; 詹海生, 博士, 副教授, 主要研究领域为计算机图形学, 数据与知识工程等。

2 项目总体解决思路

项目首先将图书逐页扫描,存储为图像文件,同时制作出 PDF 文档。以 Lucene API 作为全文检索内核,通过引入 PDFBox API 对 Lucene 索引模块加以修改。使系统在原来 Lucene API 只能索引 html,txt 文件的基础上加入对 PDF 文档的索引,实现了对 PDF 格式文档的全文索引要求,这一过程中成为一次索引。

项目除了对 PDF 文档进行全文索引外,还需要利用 PDFBox API 对 PDF 文档进行分析,分析的过程是逐页提取 PDF 文档文本内容,然后分词并获取每个最小分词的坐标,连同其页码及上下文信息一起存入数据库,这些信息即构成了分词的二次索引信息。

一次检索通过 Lucene 的一次索引信息检索到具体的 PDF 文档,用户可浏览一次检索的志书,此时用户实际看到的是志书扫描图像,为了在书籍的具体页划出关键词的具体位置,二次检索时从数据库中提取关键词的二次索引信息,利用该二次索引信息就可以将检索定位到书籍的具体页码,并在页中划出关键字的坐标位置。

3 项目系统的结构图

系统利用 Lucene API 生成一次索引,搜索时可以将关键字定位到具体文档。由于 PDF 文档的结构性较好,利用开源工具 PDFBox API 可以从 PDF 文档中分页提取文本,并且还可以提取文本在页面中的坐标位置,所以可以给 PDF 文档做更为精确的二次索引,将关键字定位到 PDF 具体页面。

系统的总体结构图如图 1 所示^[2]。

图 1 中,文本数据库保存的是所有 PDF 格式的数字志书,这些志书需要经过文本分析引擎分别提取纯文本后并进行索引,将索引信息存入索引库内供一次检索使用。当用户输入查询语句后,文本分析引擎会对查询语句进行分析,查询引擎利用查询语句的分析结果检索索引库,并对命中文档按一定规则进行排序后反馈给用户。此时用户获得的是“某书中有某个关键词”的信息,为了在书籍的具体页划出关键词的位置,就需要在本书内进行二次检索,此时需要提取数据库中的二次索引信息。利用该二次索引信息就可以将检索定位到书籍的具体页码,并在页中划出关键字的坐标位置。

4 项目系统实现

4.1 一次索引和检索的实现

一次索引是通过 Lucene API 实现的,具体需要用到与索引文档相关的几个核心类:Field, IndexWriter,

Document, Analyzer 和 Directory 等^[3]。一次索引建立好后,会在本地磁盘创建一份倒排索引文件,对 PDF 文档的全文检索主要依靠这份倒排索引文件。Lucene API 利用与检索相关的核心类来读取和检索倒排索引文件,从而实现对 PDF 文档全文检索。这些核心类有 Term, TermQuery, Query, IndexSearcher 和 Hits 等^[4]。一次索引的核心代码如下所示^[5,6]。

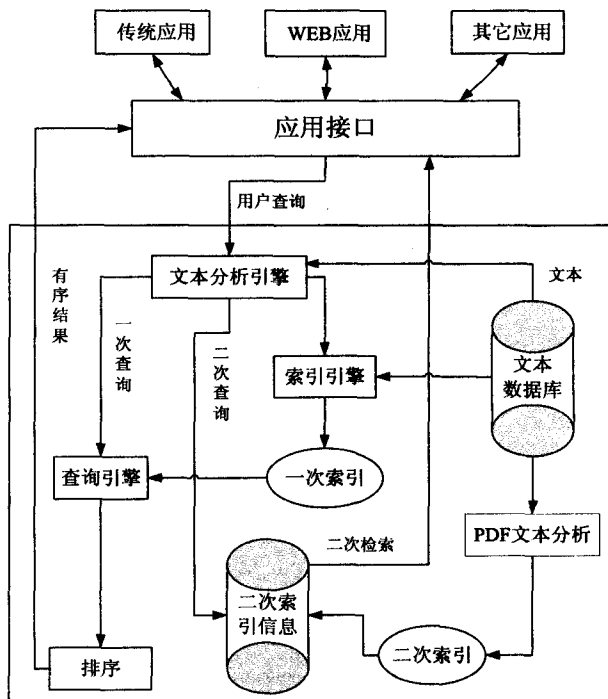


图 1 系统总体结构图

```
public class FirstIndex {
    private FirstIndex () {}
    private static IndexWriter writer;
    /** 索引 PDF 文档. */
    public static void main(String[] argv) {
        try {
            File root=null;
            String index=argv[0];
            root=new File(argv[1]);
            writer = new IndexWriter(index, new IK_CAnalyzer());
            indexDocs(root); //索引文档
            System.out.println("Optimizing index...");
            writer.optimize();
            writer.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

4.2 二次索引核心算法的实现

4.2.1 二次索引概述

系统对 PDF 文档提供了更深层次的检索,可将检索结果定位到书籍的具体页,并在页面标示出关键字

的具体位置。该层次的检索用 Lucene 的倒排索引文件是无法实现的。文中自定义的二次索引组织格式如下。

Book_id#keyword#page#以逗号隔开的 X,Y 坐标#关键词出现的上下文。

当关键词在页面出现多次时,坐标间用“|”隔开,坐标单位为像素,代表关键词以文档左上角为原点的水平向右和垂直向下方向上的距离。同样其多个上下文之间也用“|”隔开。如下为一条存于文本文件中的二次索引示例。

B001#北京#22#274,101#略重点,先后在北京、香港、曼谷、东京。

这条二次索引表示编号为“B001”的书籍中“北京”关键词存在于第 22 页,该关键词的坐标为:274,101,关键词出现的上下文为:略重点,先后在北京、香港、曼谷、东京。

4.2.2 二次索引的实现

二次索引的流程比较简单,二次索引是通过 PDF 文档进行深入解析后生成的,调用 PDFBox API 对 PDF 文档进行解析,在解析过程中对 PDF 文档分页提取文本,再对提取的文本分词后做索引,然后将二次索引信息插入到数据库即可。本算法的核心在 PDF 文档分析和分词模块,二次索引核心代码如下^[7]:

```
public class SecondIndex {
    public static COSDocument cosdoc = null;
    private static PDDocument document = null;
    private static PDFTextStripper stripper = null;
    private static void loadPDDocument(String filepath) {
        .....//加载 PDF 文档到全局 PDDocument 对象 document 中
    }

    public static String getpdfText(int start,int end){
        String outstr=new String();
        .....// 从 PDF 文件提取 start 到 end 页面的文本并返回
        return outstr;
    }

    public static void main(String[] argv) throws IOException {
        String bookname=argv[0];
        String bookpath=argv[1];
        loadPDDocument(bookpath);
        List pages=document.getDocumentCatalog().getAllPages();
        Iterator pageIter = pages.iterator();
        int i=1;
        Analyzer analyzer=new IK_CAnalyzer();
        //循环遍历 pdf 文档所有页,并对该页进行相应的处理
        while( pageIter.hasNext() )
        {
            .....//为前页分词建二次索引,将生成的二次索引写入文本文件中
        }
    }
}
```

```
i++;
}

cosdoc.close(); // 关闭 COSDocument
document.close(); // 关闭 PDF Document
}
```

运行完上述代码后,一个 PDF 文档即被处理成一个文本格式的二次索引文件,再将文本格式的二次索引文件导入到数据库即可。

4.3 二次检索的实现

二次检索的流程比较简单,用户查询在提交数据库查询之前,首先需要进行文本分析,对用户的查询语句进行分词。将分词后的多个结果提交数据库查询。然后对查询返回的多个二次索引结果集按照布尔逻辑关系合并,并对合并后的结果集进行排序。反馈给用户的是按照页码进行排序的二次索引集。最后利用这些二次索引(包含关键词所在页面和坐标等信息)在文献具体页用特殊颜色标识出检索关键字,最终检索效果类似 Google 图书搜索。

5 实验分析

通常利用查全率、查准率、检索时间和检索结果排序等指标来评估文本信息检索和搜索系统的性能。文中主要就查全率和查准率两个指标来分析和讨论本项目研究成果的性能^[8]。

一般就一个检索系统来讲,同时具有最优的查全率和查准率是不可能的。因为查全率高时,则其查准率就会偏低;而查准率高时,则其查全率也同样会偏低^[9]。

本系统使用数字志书做测试的试验数据。在这个数据集的基础上对一次索引的检索结果进行了 500 次抽样测试,一次检索的查全率和查准率如表 1^[10,11]。

表 1 一次检索的查全率和查准率

关键词	北京	中兴	华为	平均值
结果类别				
资料库中正确答案(个)	30	29	12	
查询的结果(个)	35	32	14	
正确结果(个)	22	21	9	
查全率	73.3%	72.4%	75%	79.1%
查准率	62.9%	65.6%	64.3%	68.0%

从上表可以看出,“一次检索”(全文检索)的查全率高于其查准率。由于二次索引主要是在某一 PDF 书籍中进行查找,因此在统计其查全率和查准率时要借助 PDF 自带的搜索功能,令通过 PDF 的字符串匹配搜索得到的结果为“PDF 书籍中正确答案(个)”;而从数据库中提取的结果为“查询的结果(个)”;“查询返

回的正确结果(个)”代表从数据库中提取结果中的有效结果数。二次检索的查全率和查准率如表 2 所示^[12]。

表 2 二次检索的查全率和查准率

关键词 结果类别	北京	中兴	华为	平均值
PDF 书籍中正确答案(个)	20	140	35	
查询的结果(个)	17	110	29	
正确结果(个)	12	85	21	
查全率	60.0%	60.7%	60.0%	63.6%
查准率	70.6%	77.3%	72.4%	78.5%

从上表可以看出,二次检索的查准率高于其查全率,且二次检索的查准率明显高于一次检索的查准率。分析其原因,主要是因为二次检索是在 PDF 书籍内部进行搜索,故其查准率高于其查全率;又由于二次检索搜索的范围较一次检索更加精确,故其查准率明显高于一次检索的查准率。

6 结束语

全文检索系统在 J2EE 环境下实现,并作为西安市数字方志全文检索系统投入试用,取得了一定的应用成果。系统运行期间通过收集和整理用户使用后的反馈信息,发现该全文检索系统还存在一定的不足,今后系统进一步改进的工作包括:完善二次索引算法,研究可以在 PDF 文档解析前的预处理和提取更准确的位置坐标方面深入下去;增强检索反馈的智能性,系统的反馈还缺乏智能性,若系统能记录用户的搜索习惯,并根据其搜索习惯反馈搜索结果将会更好地改善用户体验。虽然本系统离真正的智能化系统还有相当的距

离,但相信随着应用需求的推动和计算机技术的进步,必将有更广阔的发展空间。

参考文献:

- [1] 王雅戈,朱原谅. “常熟数字方志全文检索数据库系统”项目研究报告[J]. 山东图书馆学刊,2009(1):78-79.
- [2] 郑轶媛. 基于 J2EE 的站内搜索引擎的研究[D]. 上海:上海交通大学,2005.
- [3] 邱 哲,符滔滔. 开发自己的搜索引擎——Lucene 2.0+Heriterx[M]. 北京:人民邮电出版社,2007.
- [4] 王学松. Lucene+nutch 开发搜索引擎[M]. 北京:人民邮电出版社,2008.
- [5] 孟 涛,闫宏飞,王继民. 一个增量搜集中国 Web 的系统模型及其实现[J]. 清华大学学报:自然科学版,2006(9):55-57.
- [6] 郑榕增,林世平. 基于 Lucene 的中文倒排索引技术的研究[J]. 计算机技术与发展,2010,20(3):81-83.
- [7] 李永春,丁华福. Lucene 的全文检索的研究与应用[J]. 计算机技术与发展,2010,20(2):13-15.
- [8] 于 丹. 关于查全率和查准率的新认识[J]. 西南民族大学学报,2009,2(210):283-285.
- [9] 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):6-7.
- [10] Ricardo Baeza-Yates, Berthier Ribeiro-Neto. Modern Information Retrieval[M]. 北京:机械工业出版社,1999:34-36.
- [11] Sun Jian-tao, Zeng Hua-jun, Liu Huan. CubeSVD: A Novel Approach to Personalized Web Search[C]//In WWW Conference, 2005.
- [12] Boldi P, Codenotti B, Santini M, et al. UbiCrawler: A scalable fully distributed web crawler[J]. Software: Practice & Experience, 2004, 34(8):711-726.

(上接第 120 页)

fornia:[s. n.], 1997.

- [2] 刘 波,潘久辉. 基于频繁模式图的多维关联规则挖掘算法研究[J]. 电子学报,2007,35(8):1612-1617.
- [3] 陶多秀,吕跃进,邓春燕. 基于粗糙集的多维关联规则挖掘方法[J]. 计算机应用,2009,29(5):1405-1409.
- [4] 朱 玉,张 虹,孔令东. 基于人工免疫的多维关联规则挖掘及其应用研究[J]. 计算机科学,2009,36(8):239-242.
- [5] 宋余庆,朱玉全,孙志辉,等. 一种基于频繁模式树的约束最大频繁项目集挖掘及其更新算法[J]. 计算机研究与发展,2005,42(5):777-783.
- [6] Wang C, Hong M S, Wang W, et al. Chopper: efficient algorithm for tree mining[J]. Comput Sci Technol, 2004, 19(3):309-319.
- [7] Zhu F, Yan X, Han J, et al. gPrune: a constraint pushing framework for graph pattern mining[C]//Asia conf on knowledge discovery and data mining. [s. l.]:[s. n.], 2007:388-

400.

- [8] Desrosiers C, Galinier P, Hertz A, et al. Improving constrained pattern mining with first-fail-based heuristics[J]. Data Min Knowl Disc, 2010, 23(1):63-90.
- [9] 杨德璋,李 雷,申方成. 快速多规则约束关联算法的入侵检测研究[J]. 计算机技术与发展,2010,20(12):173-176.
- [10] Han J, Pei J, Yin Y. Mining frequent patterns without candidate generation[C]//Proceeding of the 2000 ACM-SIGMOD international conference on management of data (SIGMOD'00). Dallas, TX:[s. n.], 2000:1-12.
- [11] 任永功,张 亮,付 玉. 一种基于频繁模式树的最大频繁项目集挖掘算法[J]. 小型微型计算机系统,2010,31(2):317-321.
- [12] 谭 军,卜英勇,杨 勃. 一种高效的闭频繁模式挖掘算法[J]. 计算机工程与应用, 2010,46(6):130-132.