第30卷 第5期
2009年10月

暨南大学学报(自然科学版)
Journal of Jinan University(Natural Science)

Vol. 30 No. 5
Oct. 2009

基于Lucene的全文检索系统模型的研究和开发

朱岸青[1]， 黄 杰[2]

(暨南大学 1. 管理学院; 2. 计算机系, 广东 广州 510632)

[摘 要] 设计实现了一个基于Lucene的全文检索系统模型. 在该系统模型中,针对中文分词实现了基于词库的采用正向最大匹配算法的中文分词模块;针对多种格式文档的处理采用接口实现的方式和动态实例化的方法,实现了可以有效地处理txt、xml、html、pdf、doc和rtf等常见格式文档.

[关键词] 全文检索; 中文分词; 格式文档

[中图分类号] TP393 [文献标识码] A [文章编号] 1000-9965(2009)05-0504-05

Research and development on a Lucene-based full-text retrieval model

ZHU An-qing[1], HUANG Jie[2]

(1. Management School; 2. Department Of Computer Science, Jinan University, Guangzhou 510632, China)

[**Abstract**] A Lucene-based full-text retrieval model was designed and implemented. For Chinese words segmentation, a module which is based on word library and uses the positive direction maximum matching algorithm was presented. Further more, 1for processing the documents of various formats, interfaces and dynamic instantiation are used in the system model, so it can effectively process common formatted documents such as txt, xml, html, pdf, doc and rtf, etc.

[**Key words**] full-text retrieval; Chinese words segmentation; formatted documents

目前,通用搜索引擎是集成信息检索技术的典型代表. 然而,通用搜索引擎虽然功能十分强大,但是通过这些通用搜索引擎的站内搜索来实现的全文检索功能仍不完善,经常会导致搜索结果不全和出现"坏链接"的情况;更重要的是,作为商业化的信息检索工具,开发人员不可能将庞大的通用搜索引擎无缝地嵌入自己开发的应用程序中以提供全文索引和检索功能.

作为开源组织Apache Jakarta的成员项目,Lucene是一个用Java语言实现的成熟的、自由的、开源的软件项目,是一个高性能的、可扩展的信息检索工具集,可以方便快捷地融入到应用程序中,以增加索引和搜索功能[1].

本文在剖析全文检索系统和Lucene相关技术的基础上,设计了一个基于Lucene的全文检索系统模型,并设计了其实现的算法. 在中文分词功能上,采用基于词库的正向最大匹配算法作为基本的分词方法,较之Lucene内核包的单汉字切分方法[2]和扩展包的二元切分方法更为优化;在多种格式文档的统一处理上,采用了接口的实现方式,并且通

[收稿日期] 2009-03-20

[基金项目] 国家自然科学基金-广东省科学基金联合重点项目(U0775001)

[作者简介] 朱岸青(1976-),女,讲师,博士研究生,研究方向:信息系统,社会保障管理等

过动态实例化的方法为用户最大限度地屏蔽了各种文档格式间的差异性,使系统模型在不作中间格式转换的情况下具有了统一处理多种格式文档的能力.

1 基于Lucene的全文检索系统的设计与实现

1.1 系统的设计目标

系统利用了开放源代码的全文检索引擎工具包Lucene来构建一个通用的易于扩展的全文检索系统模型.该系统模型预期达到下面的设计目标:

(1)设计一个基于词库的采用正向最大匹配算法的中文分词模块,该模块应比目前Lucene内核包和扩展包所使用的中文分词方法更为优化.

(2)将各种类型的原始文档进行适当的组织和预处理,以便进一步索引和存储.这些文档类型包括目前常用的各种格式文档,从而弥补Lucene内核只能处理纯文本文档的功能上的不足.

(3)该系统模型在实用化的过程中应具有很好的可移植性和扩展性,包括查询接口的扩展、定制检索结果排序规则等.

1.2 系统总体结构和功能模块

全文检索系统要处理的数据源通常是以目前常见的各种格式文档保存,为了有效地处理这些格式文档,本文在文档抽取模块中实现了6种文档解析器.系统通过这些解析器可以提取文档的原始数据并包装成Lucene能够处理的抽象文档类型(Document).然后Lucene的索引器将接收这些抽象文档,对其内容进行分析(包括中文分词),最后提取索引项并生成索引库.以索引库为基础,可以定制满足各种查询需求的检索器和符合用户使用特征的可视化操作界面.图1是该系统模型的总体结构图.

根据系统的总体结构,本文进一步设计出系统的功能模块,如图2所示:其中,标准数据交换层将Lucene的内核、输入输出模块、系统配置管理模块和本文设计实现的文档分析模块(包括中文分词)和文档抽取模块(包括多种格式文档统一处理框架)连成一个完整的系统.

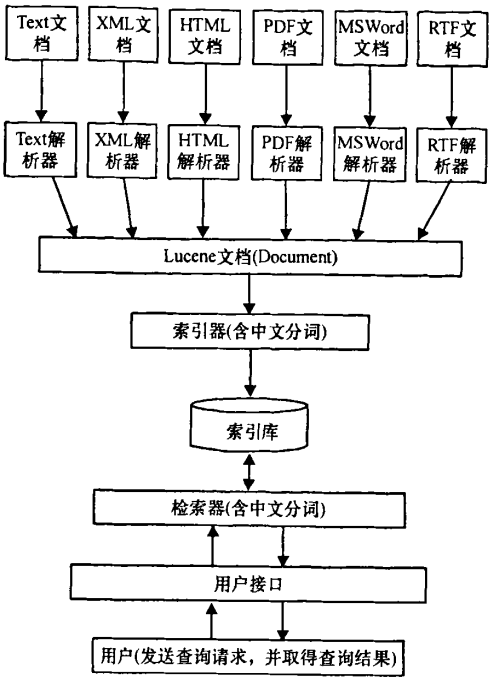

图1 基于Lucene的全文检索系统的总体结构

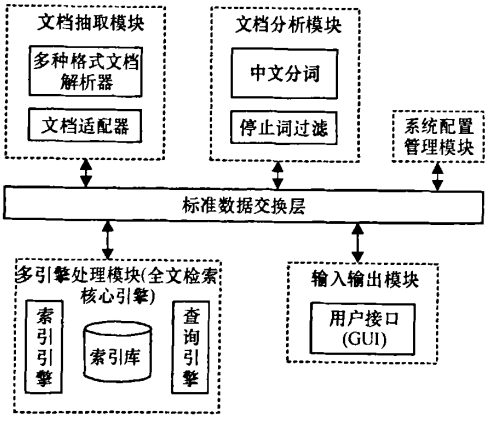

图2 基于Lucene的全文检索系统的功能模块

1.3 中文分词模块设计

Lucene的核心包和扩展包对中文分词采取类似英文的机械式切分方法,这些方法由于完全不考虑中文独特的语法和语义,所以很难保证用户在中文环境下对全文检索的需求.针对中文处理,目前使用最广泛的分词方法是词库分词,这种方法的核心是词库的匹配算法.对此,本文采用改进的正向最大匹配算法,有效地减少了字符串匹配操作,提高了系统性能.

1.3.1 设计思想

最大匹配算法的核心是"长词优先"原则,这一原则在绝大多数情况下是符合人们的日常阅读与写作习惯的. 正向最大匹配算法的形式定义为:对于文本中的字符串 ABC,$A \in W$,$AB \in W$,ABC 不属于 W,W 为字典,那么该字符串就切分为 AB/C.

1.3.2 基于词库的正向最大匹配算法

```
//输入:词库 D;停止词表 ST;中文字串 CN = "w1w2……wn";中间集合 TS;
//输出:切分出的单词集合 S
  将集合 TS 和 S 置空;
while (i < = n) {
   //如果当前字在停止词表中,转循环开始
   if (wi ∈ ST) { i++ ; countinue; }
   //查通用词典
   else { 从 D 中,按降序,选取所有词首字等于 wi 的词,放入 TS;}
   //词库中不存在以 wi 为首字的单词,为免丢失原文信息,本文将其切分为单字
   if (TS 为空) { wi 切分为单字; i = i + 1;}
   //按最长词匹配
   else { 取字串 wi ……wi+j-1 ∈TS,其中 j 取最大值;
   if (wi ……wi+j-1 ∈ ST) { i = i + j; countinue; } //属停止词
    else
        { S = S ∪ { wi ……wi+j-1 }
          i = i + j;
        }
   } //end while //循环结束;集合 S 即为所求
```

1.4 多种格式文档统一处理框架设计

Lucene 的内核被设计得非常小巧,它的处理对象局限于纯文本数据. 目前,针对各种格式文档的处理,一种常见的实现方式是先统一转换为 XML 格式作为中间格式,然后对中间格式进行索引和查询,通过 XML 强大的数据表达能力和平台无关的特性,这种方法可以为 Lucene 提供一个统一的、通用的数据源格式. 但是,这种方法存在以下问题[3]:

(1)极大地增加系统实施的时间复杂度和空间复杂度——简单的抽象使系统的实现过程更加复杂化.

(2)由于索引和查询都异化为对 XML 中间格式的处理,因此无可避免地会造成信息在一定程度上的失真.

(3)正是由于 XML 格式的规范性,为 Lucene 添加对 XML 文档的处理能力恰恰是实现过程最复杂和规范要求最多的[4].

1.4.1 设计思想

考虑到以上提到的问题,本系统建立通用的 DocumentHandler 接口,进而开发一个能够用来索引多种格式文档的统一处理框架. 接口的实现方式和动态实例化的方法将为系统的灵活性和扩展性预留下广阔的空间.

1.4.2 多种格式文档统一处理框架的 UML 设计图

多种格式文档统一处理框架的 UML 设计图如图 3 所示,DocumentHandler 接口中仅声明了返回 Document 类型的 getDocument( )方法,该方法将由 TextHandler、XMLHandler、HTMLHandler、PDFHandler、WordHandler、RTFHandler 实现,完成对相应文档原始数据的提取操作. 对于某些私有格式的文档(比如 pdf 和 doc)将通过目前流行的开源的第三方工具包进行组织和预处理. 此外,以键值对(后缀名/解析器名)形式建立起的 handler. properties 文件,使得系统可以以动态实例化的方式在运行时根据文档的后缀名自动选择匹配的解析器.

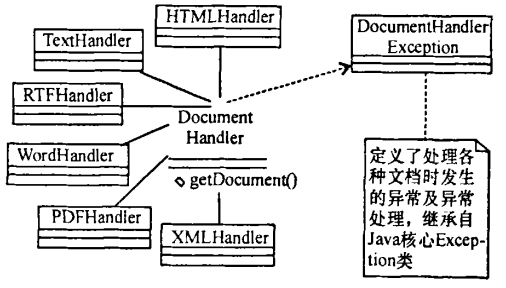

图3 多种格式文档统一处理框架的 UML 设计图

2 性能评测

2.1 中文分词模块测试

2.1.1 评测依据、实验方法和选取测试数据

为 StandardAnalyzer(Lucene 内核包中的标准分析器,对中文文本采用单汉字切分方式)、CJKAnalyzer(Lucene 扩展包中的针对中日韩等亚洲国家语种的分析器,对中文文本采用二元切分方式)和本文的 CustomTokenizer 分别编写一段测试程序,对同一中文文本进行分词实验[5]. 通过测试,该模块可以有效地减少字符串匹配操作,提高系统性能.

考虑到作为性能评测依据的各项指标,实验不宜选择过长和包含过多生僻字的测试材料. 因此,以下举例的这次实验选取以下文本作为测试材料:"走在社会主义大道上的中华人民共和国是一个无产阶级领导的伟大国家. 在新时期新形势下,全国各族人民紧密团结在胡锦涛总书记周围,高举邓小平理论伟大旗帜,全面建设小康社会,为在本世纪中叶实现社会主义现代化而努力奋斗."

2.1.2　实验过程概述

运行测试程序,收集实验数据列表1所示:

表1　三种切分方法对同一中文文本的切分效果对比

测试方法	使用的分析器	切分出的单词(字)总数/个	有检索意义的单词数(人工检验)/个	切分性能[1)]	运行10次取平均时间/ms
StandardDemo	StandardAnalyzer	98	—	—	62
CJKDemo	CJKAnalyzer	92	41	44.57%	46
CustomDemo	CustomTokenizer	48	32	66.67%	17

1)有检索意义的单词数与切分出的单词总数的比率

2.1.3　实验结果评述

从表1的统计结果可以看出:

(1)在切分出的单词总数方面,CustomTokenizer 切分出的单词总数是 Lucene 内核包 StandardAnalyzer 的 48.98%,是 Lucene 扩展包 CJKAnalyzer 的 52.17%,这说明 CustomTokenizer 比起单汉字切分法和二元切分法能够根据中文词汇的语义和语境,按单词在自然语言中所代表的真实含义进行切分,因此切分出的单词总数比完全不考虑中文语义和语境的单汉字切分法和二元切分法要少得多.

(2)在切分性能方面,CustomTokenizer 切分出的有检索意义的单词数与切分出的单词总数的比率是66.67%,明显高于 CJKAnalyzer 的 44.57%. 需要说明的两点:①单汉字切分法把中文文本切分成单字,而每个单字独立成为检索关键字的可能性非常小,同时中文单字在自然语言中是否具有实在意义也是一个难以判定的问题,因此统计单汉字切分法在这方面的切分性能并无意义;②CJKAnalyzer 分词不考虑中文的语义,并且一个二元词是否具有检索意义往往因人而异,所以本文在统计 CJKAnalyzer 切分性能时降低了这方面的标准,把切分出来的具有实在意义的二元实词当作具有检索意义的单词,因此,可以进一步得出结论:CustomTokenizer 在实际应用的时候,相对于 CJKAnalyzer 切分性能的优化程度会比统计情况更加明显.

(3)从程序的运行时间上看:切分同一个中文文本,CustomDemo 的运行时间是 StandardDemo 的 27.42%,是 CJKDemo 的 36.96%——由于这3个演示程序中均只执行了分词和输出操作,因此可以认为:CustomTokenizer 的时间性能优于 StandardAnalyzer 和 CJKAnalyzer.

2.2　多种格式文档统一处理框架测试

2.2.1　评测依据、实验方法和选取测试数据

通过编写测试程序,对一个包含各种文件类型的文档资料库建立索引,测试系统模型在不作中间格式转换的情况下是否运行良好;同时收集系统运行的时间和空间数据,进而对系统的时间和空间性能进行有效的评估.

实验选取了各种主题和格式的文件,包括了系统模型已经实现解析器的文档类型,也包括目前尚不能处理的文档类型;同时,实验材料的数量大小适中,适合于在实验环境中便捷地观测实验结果;此外,所选材料囊括中英文的各种主题的文档,可以更好地观测系统在中英文混杂的环境下的运行状况.

2.2.2　实验过程概述

运行程序对测试材料建立索引,收集实验数据如表2所示:

2.2.3　实验结果评述

观测的实验结果表明:系统正确地处理了 txt、xml、html、pdf、word 和 rtf 等各种格式文档,大大扩展了 Lucene 可以处理的文档类型;而且,这种统一处理的能力是通过接口实现的方式和动态实例化的方法直接获得,避免了中间格式的转换,相比较于以 XML 格式作为中间格式的实现方式,这不仅节省了由于转换所带来的直接的时间和空间开销,而且有效地避免了由于转换而有可能导致的原文信息的失真.

表2 实验结果一览表

观测项目	观测结果	观测项目	观测结果
测试材料包含的文件数量/个	55	索引文件占原始文件的大小	66%
测试材料的总大小/MB	8.21	系统运行时状态	正常运行,直至结束
被有效索引的文件数量/个	53	异常及错误处理	及时、正确地向用户提示[1)]
程序建立的倒排索引文件大小/MB	5.42	运行时间/ms	262 703

1)例如:对未实现解析器的文档格式,系统自动跳过,并向用户发出提示

另外,对8.21 MB的测试材料建立索引花去了超过262 s的时间,索引文件占原始文件的大小为66%,除去此次实验的硬件环境较落后的因素,我们可以得出一个结论:为了换取查找效率的大幅度提高,对文档资料库建立倒排索引消耗了大量的时间与空间资源. 因此,索引结构的优化和索引库的更新策略,是影响采用倒排索引的系统性能的重要因素.

Lucene的体系结构是非常灵活开放的,开发者可以在Lucene的基础上对其进行各个方面的扩展,有针对性地实现一个功能强大、性能良好的全文检索系统[6]. 本文以此为总的指导思想,在中文分词和多种格式文档统一处理框架方面进行了扩展,实验结果证明这两个功能均取得一定的优化效果,索引库的结构改进和索引更新策略是下一步研究的目标.

[参考文献]

[1] 郎小伟,王申康. 基于Lucene的全文检索系统研究与开发[J]. 计算机工程,2006,32(4):94-97.

[2] 陈士杰,张玥杰. 基于Lucene的英汉跨语言信息检索[J]. 计算机工程,2005,31(13):62-64.

[3] GOSPODNETIC O, HATCHER E. Lucene in Action [M]. Manning Publications Co, 2007: 153-155.

[4] 彭曙蓉,蔡 蕾,王耀南. 基于近似网页聚类的智能搜索系统[J]. 微计算机信息,2006,(12):283-285.

[5] 孔伯煊,李 祥. 基于Lucene/XML技术的Web搜索引擎设计与实现[J]. 航空计算技术,2006,36(4):5-8.

[6] 李 刚,宋 伟,邱 哲. 征服Ajax+Lucene构建搜索引擎[M]. 北京:人民邮电出版社,2006:100-101.

[责任编辑:王景周]