基于 Lucene 的中文分词器的设计与实现*

彭焕峰
(南京工程学院 计算机工程学院,江苏 南京 211167)

摘 要:针对 Lucene 自带中文分词器分词效果差的缺点,在分析现有分词词典机制的基础上,设计了基于全哈希整词二分算法的分词器,并集成到 Lucene 中,算法通过对整词进行哈希,减少词条匹配次数,提高分词效率。该分词器词典文件维护方便,可以根据不同应用的要求进行定制,从而提高了检索效率。

关键词:Lucene;哈希;整词二分;最大匹配
中图分类号:TP391.1 **文献标识码**:A **文章编号**:1674-7720(2011)18-0062-03

Design and implementation of Chinese words segmentation machine based on Lucene

Peng Huanfeng
(School of Computer Engineering, Nanjing Institute of Technology, Nanjing 211167, China)

Abstract: According to the low efficiency of the Chinese words segmentation machines of Lucene, this paper designs a new word segmentation machine based on all-Hash segmentation mechanism according to binary-seek-by-word by analyzing many old dictionary mechanisms. The new mechanism uses the word's Hash value to reduce the number of string findings. The maintenance of dictionary file is convenient, and the developers can customize the dictionary based on different application to improve search efficiency.

Key words: Lucene; Hash; binary-seek-by-word; maximum matching

信息技术的发展,形成了海量的电子信息数据,人们对信息检索的要求越来越高,搜索引擎技术也得到了快速发展,并逐渐地被应用到越来越多的领域。由于搜索引擎技术涉及信息检索、人工智能、自然语言处理等多种学科,很多搜索算法都不公开[1]。Lucene 是一个优秀的开源全文搜索引擎框架,通过 Lucene 可以方便地将全文搜索技术嵌入到各种应用当中,有针对性地实现强大的搜索功能,因此近年来 Lucene 的应用越来越广泛。

Lucene 在对信息进行索引前,需要进行分词,西方语言使用空格和标点来分隔单词,而中文使用表意文字,不能通过空格和标点来进行分词。Lucene 自带中文分词器有 StandardAnalyzer、ChineseAnalyzer、CJKAnalyzer,这些分词器要么是单字切分,要么采用二分法切分,它们并不能有效地解决中文分词问题。本文设计并实现了基于全哈希整词二分算法的分词器,并集成到 Lucene 中,从而提高了 Lucene 处理中文信息的能力。

1 Lucene 简介

Lucene 是 Apache 软件基金会 jakarta 项目组的一个子项目,是一个优秀的开源全文搜索引擎工具包,并不是一个完整的全文检索应用。它提供了丰富的 API 函数,可以方便地创建索引,嵌入到各种应用中实现全文检索[2]。Lucene 作为开源的全文搜索引擎,架构清晰,易于扩展,而且索引文件格式独立于应用平台,从而使索引文件能够跨平台共享,能够对任意可转换为文本格式的数据进行索引和搜索,例如网页、本地文件系统中的 WORD 文档、PDF 文档等一切可以从中提取文本信息的文件[3]。

2 基于全 Hash 的整词二分分词器

目前中文分词算法大致可分为三大类:机械分词方法、基于理解分词方法和基于统计分词方法[4]。其中机

* 基金项目:南京工程学院科研青年基金项目(QKJB2009026)

械分词把待分解的汉字串与词典中的词条进行匹配来判断是否是一个词，是当前应用广泛的一种分词方法。在深入研究整词二分、TRIE 索引树和逐字二分这三种传统的机械分词算法及其改进算法的基础上，本文设计并实现了基于全哈希的整词二分分词器。

2.1 词典设计

词典设计是机械分词的关键，词典结构应与分词算法相结合，这样设计的分词器分词效率才能得到最大限度的提高[5]。全哈希整词二分词典机制由首字哈希表、哈希节点、词哈希表和词碰撞表构成，如图 1 所示。

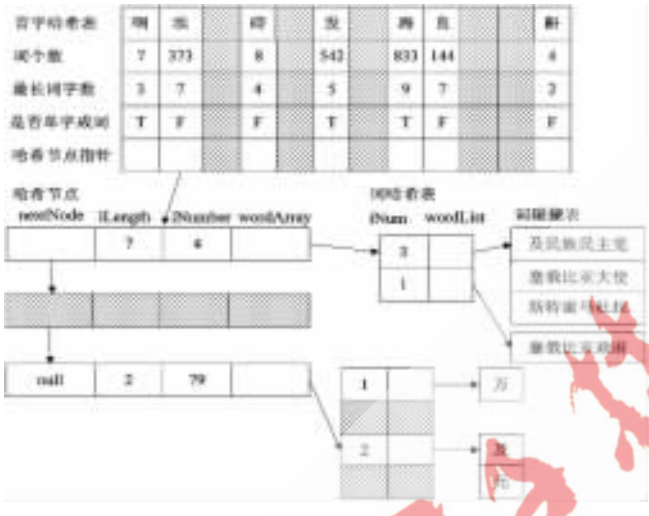

图 1 全 Hash 整词二分词典组织结构

(1)首字哈希表

汉字在计算机中是以内码的形式存储。根据内码获取汉字对应的区位码，从而给定一个汉字，可以通过哈希函数直接得到其在首字哈希表中的位置[6]。用 lowByte 表示汉字内码的低字节，highByte 表示汉字内码的高字节，则哈希函数设计如下：

$$Index=((highByte-160)\times 100+lowByte-160)-1601$$

词个数：记录以该字为首字的词的个数。

最长词字数：记录以该字为首字的最长词所包含的字数。

是否单字成词：标识该字是否可以单独成词。

哈希节点指针：指向以该字为首字的第一个哈希节点。

(2)哈希节点

记录了词长(iLength)、相同词长的词的个数(iNumber)、词哈希表的地址(wordArray)以及下一个哈希节点的指针(nextNode)等信息。对相同首字的词条按包含的字数进行分组，再对除首字外的剩余字符串进行全词哈希，哈希节点按照词长倒序排列。

(3)词哈希表

记录了哈希值相同的词条的个数(iNum)、指向保存词条的动态数组(wordList)，即词碰撞表。词哈希表的大小为词长相同的词的数量，哈希函数的选取不能太复杂，否则会增加分词时间，同时也要考虑哈希结果的均匀分布。

(4)词碰撞表

词碰撞表实际上是一个动态数组，对于词典中首字相同且词长相同的词条，如果哈希值相同，则以动态数组的形式保存，且只保存除首字之外的剩余字串。对于哈希值相同的词条，采用二分查找。

2.2 分词算法

该分词词典机制适用于正向最大匹配算法和逆向最大匹配算法，本文采用正向最大匹配算法为例使用该词典机制进行分词。

第一步：取待匹配字符串

假设 String 为待分词语句，如果 String 不包含任何字符(即长度为零)，则表示语句分词完毕。从 String=$A_1A_2A_3A_4\cdots An$ 中读取第一个字 A_1，从首字哈希表中获取以该字为首字的最长词的字数 m，如果 String 剩余待分词的字数不足 m，则取 Str 为 String 的剩余待分词的字符串，否则取字符串 Str=$A_1A_2\cdots Am$ 为待匹配字符串。

第二步：判断 Str 的长度

(1)如果 Str 的长度为 1，在首字哈希表中找到对应的位置，判断单独成词标志是否为 F，如果是说明 Str 不是一个词；否则说明 Str 是一个词，分出该词。设置待切分语句 String=$A_2A_3A_4\cdots An$ 后转第一步。

(2)如果 Str 的长度大于 1，转第三步。

第三步：对待匹配字符串分词

在 A_1 对应的哈希节点链表中查找词长为字串 Str 长度的节点，有如下两种情况：

(1)如果没有找到，则 Str 字串不是一个词，则去掉 Str 的最后面的一个字，转第二步。

(2)如果找到对应的哈希节点，则计算 Str(除去首字，因为首字不保存)的哈希值，得到在词哈希表中的位置，有如下三种情况：

①若对应的碰撞数(iNum)等于 0，说明 Str 不是一个词。

②若对应的碰撞数(iNum)等于 1，则比较去掉首字的 Str 与词碰撞表中的词，如果相等，则 Str 成词，否则不成词。

③若对应的碰撞数(iNum)大于 1，则在词碰撞表中进行二分查找，如果找到则成词，否则不成词。

上述三种情况若都不成词，则去掉 Str 的最后面的一个字，转第二步；若成词，则在 String 中分出一个词 Str，将语句 String 设置为除去 Str 的剩余字串，转第一步。

2.3 实验结果及分析

可以很方便地将分词器集成到 Lucene 中，该分词器不妨命名为 MyAnalyzer。对 3 个分词器 ChineseAnalyzer、CJKAnalyzer、MyAnalyzer 进行实验，采用复合索引结构，只对文档内容创建域，且只对文档内容进行索引但不存储，在分词时间、索引文件大小两方面做对比，分别对两

表 1 不同分词器实验数据对比

词典机制	索引时间(文档集 1)/ms	索引大小(文档集 1)/MB	索引时间(文档集 2)/ms	索引大小(文档集 2)/MB
ChineseAnalyzer	1 042	0.065	37 214	1.912
CJKAnalyzer	1 222	0.183	42 897	5.953
MyAnalyzer	3 096	0.132	79 024	2.431

个文档集进行实验,文档集 1 含有 3 个文件,共 2.38 MB,文档集 2 含有 126 个文件,共 164 MB。表 1 为实验结果数据。

通过实验数据可知,采用 CJKAnalyzer 二分法切分形成的索引文件要远大于采用 ChineseAnalyzer 单字切分所形成的索引文件,但两者在索引时间上相差并不大,由于索引文件中记录关键字及其词频和所在文档等信息,所以当测试文档集增大时,采用本文设计的 MyAnalyzer 分词器所产生的索引文件大小与采用 ChineseAnalyzer 所产生的索引文件相差逐步减少,但远小于采用 CJKAnalyzer 分词器产生的索引文件大小。最为关键的是,采用 MyAnalyzer 生成的索引能大大提高全文检索的查准率和查全率。

3 应用

Lucene 具有方便使用、易于扩展等优点,越来越多的开发者将其嵌入到不同的应用中实现全文检索功能。各种应用有着不同的检索需求,本文通过扩展 Lucene 的中文分词器,使开发者可以针对系统的特点定制自己的分词词典,根据具体的需求进行分词,并创建索引,从而提高全文检索的效率。

参考文献

[1] 胡长春,刘功申.面向搜索引擎 Lucene 的中文分析器[J].计算机工程与应用,2009,45(12)157-159.

[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):5083-5085.

[3] 吴青,夏红霞.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):145-148.

[4] 孙茂松,左正平,黄昌宁.汉语自动分词词典机制的实验研究[J].中文信息学报,1999,14(1):1-6.

[5] 李庆虎,陈玉健,孙家广.一种中文分词词典新机制-双字哈希机制[J].中文信息学报,2002,17(4):13-18.

[6] 张科.多次 Hash 快速分词算法[J].计算机工程与设计,2007,28(7):1716-1718.

(收稿日期:2011-06-22)

作者简介:

彭焕峰,男,1978年生,硕士,讲师,主要研究方向:网络通信和软件开发。