基于 Lucene 的 XML 文件相似度检索系统①

吴新强, 周 娅, 王如意, 张敬伟, 林煜明

(桂林电子科技大学 计算机科学与工程学院, 桂林 541004)

摘 要: 经分析研究开源的 Lucene 系统架构以及特殊 xml 数据源, 针对 Lucene 搜索得分公式的不足, 提出了结合词项位置和二次检索的公式, 设计一种文本搜索系统; 并以提高检索性能、相似性搜索的准确率、索引的空间效率和支持查询的时间效率为目标进行实验, 最后通过部署 Tomcat 服务器实现. 经实验验证, 改进的系统较之于原 Lucene 系统提高了建立索引效率、查询效率、准确率.

关键词: Web Lucene; 相似度; 词项位置; 二次检索; XML

XML File Similarity Retrieval System Based on Lucene

WU Xin-Qiang, ZHOU Ya, WANG Ru-Ri, ZHANG Jin-Wei, LIN Yu-Ming

(School of Computer Science and Engineering, Guilin University of Electronic Technology, Guilin 541004, China)

Abstract: On the basis of analysis and study on the open source Lucene system architecture, a semantic search system is designed based on the special XML data sources in this paper. What's more, we use the word item location and word semantic to improve the Lucene's search results and conduct experiments to test and verify the retrieval performance, the accuracy of similarity search, the space efficiency of index and the time-efficiency of supporting inquiry: And finally by deploying the Tomcat server to implement our implement system. The experiment results prove that compared with the original Lucene indexing system, our system can improve the indexing efficiency, query efficiency and accuracy.

Key words: Lucene; similarity; lexical item location; secondary retrieval; XML

与传统的目录索引、关键字索引相比, 语义索引方式更加接近人类的生活方式, 使检索系统与用户的交流越发人性化. 但因为汉语中语义的多样性和复杂性, 当前还没有很好的支持汉语的自然语言搜索引擎.

Lucene[1]中提到其本身不是一个相对完整的全文搜索引擎,而是一个简单的架构应用与全文搜索引擎. Lucene 是一个全文检索引擎工具包, 应用与开放源代码. 它作为一个全文搜索引擎,有其本身比较突出的特性. 但也有不足之处: 偏向于短文本得分高; 对于查询词在一个文档中位置并不重要; 一个文档中, 除该查询词外, 其他的词越多, 得分越低; 没有考虑到汉语自然语言中词项语义的权重, 检索精确度不高.

本文针对特殊的 XML 数据源进行检索, 数据源结构由问题单 XML 文档构成, 除了编号以外, 还包括问题单号、简要描述、详细描述四个域, 分别从简到详进行描述. 其中问题单号可以看成查询中关键词的重量级词项, 而简要描述和详细描述则可看成对问题单号的具体描述信息.

由于 Lucene 自身相似度评分公式忽略了词语位置, 对本文中的特殊数据源无法有效的计算出其正确的得分并进行倒排, 同时造成了一定的精确度误差. 所以, 本文提出了一种新的计算相似度的评分公式, 即在原始相似度的基础上结合词语位置相似度和二次检索相似度. 并通过程序实现基于 Lucene 的 XML 相似度检索系统. 经实验结果验证, 本系统检索结果的准确率较 Lucene 原始系统有明显的提高.

① 基金项目:广西教育厅高校科技项目(2013YB095);广西信息实验科学中心重点项目(20130111);广西教育厅一般资助项目(20103YB051);桂林电子科技大学研究生创新项目(GDYCS201465)

收稿时间:2014-05-13;收到修改稿时间:2014-06-13

© 中国科学院软件研究所 http://www.c-s-a.org.cn

1 相关工作

1.1 XML 文件的解析

XML 文件解析器是用来解析基于 Lucene 的相似度检索系统的数据源-问题单, 其通常有四种经典的方法来解析 XML 文件. 基于事件流的解析的 SAX 方法, 也是最常用的解析方法; 基于 XML 文档树结构解析的 DOM 方法; 另一种是一个开放源代码的软件, Dom4j 具有性能优越、功能强大和非常易用的特点并是一个非常优秀的 Java XML API. 最后一种主要是为了减少 DOM、SAX 的编码量, 出现了 JDOM, 其优点是极大减少了代码量.

每种解析方式都各有利弊, 本文只针对问题单这种特殊数据源进行测试, 为了比较几种方法的解析性能, 实验过程中使用了 500MB 大小、600 万行以上的 XML 问题单文件进行 20 轮测试, 如图 1.

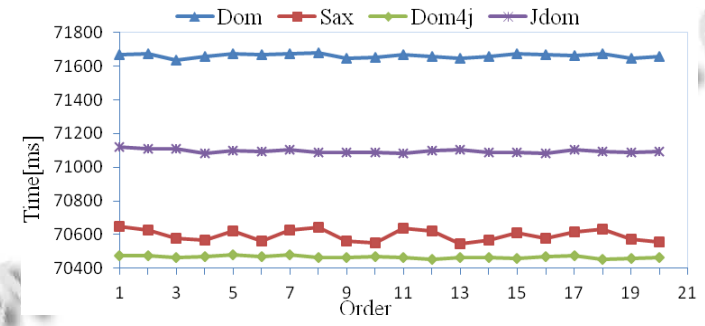

图 1 XML 解析器性能比较

由图 1, 可以看到使用 Dom4j 解析器解析问题单数据的性能更好, 故选择 Dom4j 解析器.

1.2 中文分析器

中文分词器主要是进行切分问题单数据源中文本, 使其基于 Lucene 的相似度检索查询词有更完美的匹配结果. 对问题单这种特殊的数据源来说, 中文分词器也要根据其自身结构的特性来选择, 一下介绍 Lucene 内置的分词器以及第三方分词器.

StandardAnalyzer 对于中文分词来说其本质是一元分词, 得到的结果是独立的中文单字, 而不是真正的词汇分割.

CJKAnalyzer 中文分析器实现原理是二元分词, 即根据汉语中 2 元较多的特点将两个字组成一个 2 元词. 然而二元切分的正确率比较低, 是因为分割后的词项的结果会造成大量的索引冗余, 以至于导致检索结果不是预期的结果, 得到错误的检索结果.

开源中文分析器 IK[3]的分词原理是实现以词典为基础的正反向全切分, 即最细粒度切分算法, 它针对一个汉语句子, 会罗列出一个字的左右分别切分组成词组, 直到切分出所有可能的词汇. IK 分析器分词结果数量随句子长度呈指数增长, 时间和空间开销较大.

开源分析器庖丁解牛(Paoding)分词器的分词原理是 采用完全面向对象设计, 默认将内容按最大的词和最小的进行切分, 即最大的词中若是包含小的词则小词将也会切分出来.

1.3 测试

本文通过一个测试用例来以及创建索引用例来分析各种中文分词器的性能, 然后根据针对特殊的数据源结构选择出性能优越的分词器, 说明测试用例: “【问题分析】A 类板没有将帧头异常作为上报 MASTER_ERR 告警的条件, 但是是 SLAVE_BAD 告警的条件.”分词结果如表 1.

表 1 分词结果比较

分词器	分词效果	时 间 (ms)
Standard Analyzer	问|题|分|析|类|板|没|有|将|帧|头|异|常|作|为|上|报|master_err|告|警|的|条|件|但|是|是|slave_bad|告|警|的|条|件|	17
CJK Analyzer	问题|题分|分析|类板|板没|没有|有将|将帧|帧头|头异|异常|常作|作为|为上|上报|master_err|告警|警的|的条|条件|但是|是是|slave_bad|告警|警的|的条|条件|	14

© 中国科学院软件研究所 http://www.c-s-a.org.cn

Paoding Analyzer	问/题/分/析/类/板/没/有/将/帧/帧/头/异/常/作/为/上/报/master/err/master_err/告/警/条/件/slave/bad/slave_bad/告警/条件/	54
IK Analyzer	问题|分析|a|类|板|没有|将|帧|头|异常|作为|上报|master_err|告警|的|条件|但是|是|slave_bad|告警|的|条件|	386

从表 1 可以看出, Standard Analyzer 采用一元切分, CJKAnalyzer 采用二元切分, 而基于词典的 IK 与 Paoding 分析器都能正确得到{帧头}这一词. IK 分词器对空字符串分词也要 386ms.

对 500MB 的文件进行建索引, 并进行 20 轮的测试. 创建索引所用时间以及生成索引所占空间如表 2

表 2　对 500MB XML 文件建立索引及其所占空间

分词器	Standard Analyzer	CJK Analyzer	Paoding Analyzer	IK Analyzer
索引时间/MS	77175.15	77187.45	92769.15	92375.85
索引空间/MB	27.5	27	36	39.6

由表 2 可以看出, Lucene 自身的分词器虽然创建索引时间及生成的索引文件较小, 不足之处是分词结果准确率较低; 第三方分词器, IK 分词器在时间上优于 Paoding,但由于 IK 采用全切分算法得到词汇较多, 故所占空间较大. 本系统选用 IK 分词器来分词.

2　Lucene中相似度得分公式的改进

Lucene 在查询时, 对于问题单 xml 文件来说, 查询 q 在某记录中的位置没有要求, 也就是在不同的文档(d)中的重要程度是一样的. 但一般情况下, 在“简要描述”里的查询词重要程度要比“详细描述”重要, 这样就应当考虑查询词在文章里出现的位置. 同时, Lucene 搜索返回结果精确度不高, 为了提高精确度, 引入了二次检索. 因此, 对其评分算法进行针对性的修改公式可以得到令人更满意的结果.

Lucene 内部自身默认的相似度得分计算[4]公式如公式(1):

$$OldScore(q,d) = coord(q,d) \bullet queryNorm(q) \bullet \sum_{t\_in\_d} tf(t\_in\_d) \bullet idf(t)^2 \bullet t.getBoost() \bullet norm(t,d) \quad (1)$$

文献[4]在 Lucene 中应用了自己实现的新中文分析器, 从而使检索的准确率和召回率有所提高. 文献[7]改进 Lucene 原始的相似度评分机制的方法是通过平均词频来改进公式以及引进了文档长度标准因子使得公式对短文本文档的得分计算的更加合理. 众多学者通过不同的方式在 Lucene 原有的相似度算法中增加语义信息的相似度算法, 如文献[5]、[6]和 [8], 以此实现对文档语义信息的检索.

如上所述各种研究和改进虽然在不同程度上改进了 Lucene 的搜索结果的准确性, 但均未考虑关键字词项在查询文档中的位置、位置关系在查询文档中的问题以及 Lucene 检索结果中文档的二次检索的研究. 因此, 本文在 Lucene 自身默认的词频相似度计算得分函数的基础上结合词语位置特征、二次检索的相似性, 提出一种新的计算相似度得分算法. 通过这些改进, 对比传统的基于词频的方法(公式(1)), 本文提出的方法能够取得较好的检索精确度和召回率.

针对系统的模型结构如下图 2 所示.

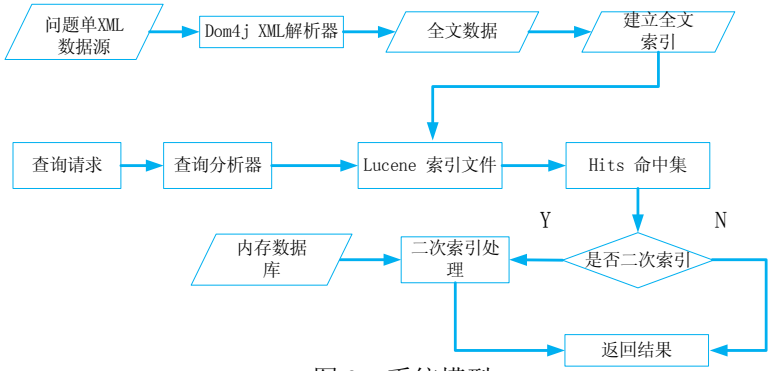

图 2　系统模型

© 中国科学院软件研究所 http://www.c-s-a.org.cn

2.1 词语位置相似度得分

Lucene 内置默认的相似度得分函数算法是基于传统的词频分析方法. 传统的词频分析方法忽略了词语位置关系这一重要特征. 基于此, 本文在 Lucene 内置默认的相似度得分函数中引入词语位置特征因子: 词语位置相似度得分, 用来反映查询串(q)和检索文档(d)中的词语在位置关系上的相似程度.

本文结合文献[9]中的语序相似度算法和文献[10]、文献[11]词语紧邻相似度算法, 提出词语位置相似度得分公式为公式 2:

$$PrScore(q,\ d) = \frac{PosScore(q,d) + \alpha \bullet ConScore(q,d)}{1 + PosScore(q,d) + \alpha \bullet ConScore(q,d)} \quad (2)$$

说明: PosScore (q,d) 表示查询串 q 与文档 d 的词序相似度得分; $ConScore(q,d)$ 表示查询串 q 与文档 d 之间的词语紧邻相似度得分; α 表示词语紧邻相似度相对于词序相似度的重要权重的因子, α 为非负数, 当认为词语紧邻相似度相对于词序相似度重要时, $\alpha > 1$; 当认为词语紧邻相似度和词序相似度权重一样时, $\alpha = 1$; 否则, $\alpha < 1$. $0 \le PosScore(q,d) \le 1$, $0 \le ConScore(q,d) \le 1$, $0 \le \Pr Score(q,d) < 1$. 由词语紧邻相似度得分算法、词序相似度得分算法可知: 当词序相似度得分为 0 时, 词语位置相似度得分也为 0.

公式(2)中的词语位置相似度对于问题单数据源的关键字检索结果进行得分优化, 使大多位于“简要描述”区域中的关键字匹配到结果得分更高, 返回排序的结果位于最前面, 从而大大提高检索的正确率以及极大的提高检索的效率.

算法 1: 词语位置相似度得分算法

1) 预处理文档 d、查询关键

2) 计算(q,d)间的词序相似度得分 $\mathrm{PosScore}(q,d)$

3) If $\mathrm{PosScore}(q,d)$=0 then

4) 返回结果 0

5) Else 计算(q,d)间的词语紧邻相似度得分 $ConScore(q,d)$

6) 调用公式(2)计算出(q,d)间的词语位置相似度得分, $\Pr Score(q,d)$ 并输出结果.

2.2 二次检索相似度得分

人类对新信息的要求会更加高, 仅利用词语的表面不准确的信息(如词频) 的文本相似度计算方法, 现在已经满足不了人们的需求. 大数据时代, 如何有效、快捷的搜索到客户需求的正确信息, 是当前极其需要的考虑的问题. 正因为如此, 人们着手研究通过在检索结果中再次进行检索来提高精确度. 为了实现更精确的检索关键词定位,文献[12]中提出了一种新的二次检索算法, 利用该算法可将检索关键词定位到具体的某个位置,并在搜索界面指定出关键字的具体位置. 据此提出了二次检索相似度得分计算的公式:

$$\mathrm{ReScore}(q,d) = \mathrm{K} + (\mathrm{hitNum} - 1) \bullet \gamma \quad (3)$$

其中, K 表示二次检索的常数项; hitNum 表示第一次检索命中的集合; γ (0< γ)是一个调节因子, 需要大量实验确定.

公式 3 中的二次检索相似度是针对第一次检索结果中返回结果太多, 不能及时得到用户需要的正确信息, 故需要在已得到的结果中进行再次检索, 以达到返回的结果是用户需要的数据. 二次检索使得检索结果的精确率大大太高, 但同时也牺牲了二次创建索引的时间.

算法 2: 二次检索相似度得分算法

1) 预处理第一次检索得到 hitNum 个文档 d,进行解析、分词、二次创建索引.

2) 把分词后的多个结果提交查询 q, 进行二次索引

3) 调用公式(3)计算(q,d)间的二次检索相似度得分 ReScore (q,d), 并输出结果

2.3 改进后的 Lucene 相似度评分算法

Lucene 相似度评分算法改进后, 此算法不仅仅是根据查询词在文档中出现的频率次数来计算其相似度得分, 而且考虑了关键词在文中出现的位置. 由于关键字位置的不同, 其在文中所表达的意义也是不同的, 即权重不同. 如, 出现在标题中的关键字会比出现在文中的权重高, 这是理所当然的事情. 再通过结合二次检索来提高检索的精确率, 使得改进后的 Lucene 相似度评分算法相对 Lucene 内置的评分算法不但可以正确的检索出与查询关键词匹配或相似匹配的词序的文档具有较高的得分, 还可以保证一个与查询串有越长、越多完全匹配子串的文档具有越高的得分.

改进后的 Lucene 相似度评分公式如下:

© 中国科学院软件研究所 http://www.c-s-a.org.cn

$$Score(q,d)=K_1\bullet OldScore(q,d)+K_2\bullet \Pr Score(q,d)+K_3\bullet \mathrm{Re}Score(q,d) \quad (4)$$

其中，$OldScore(q,d)$ 为应用 Lucene 原始相似度评分算法得到的相似度得分；K_1、K_2、K_3为权重因子($0\le K_i\le 1$)，$K_1+K_2+K_3=1$.

当索引库较小时，查询无关的文档(即相似度很低的文档)都要参与算法中全部的计算，当索引库很大时，检索效率必然大大降低. 为此，本文设定两个个阈值 δ ($0<\delta<\theta$)，θ ($\delta\le\theta\le 1$)，当文档 d 的 $OldScore(q,d)$ 大于这个 θ 时，$\Pr Score(q,d)$=0，ReScore(q,d) =0；当 $\delta < OldScore(q,d) < \theta$ 时，ReScore(q,d) =0；$OldScore(q,d) < \delta$ 时，都参加计算. 因此，改进的 Lucene 相似度评分算法流程图如 3 所示.

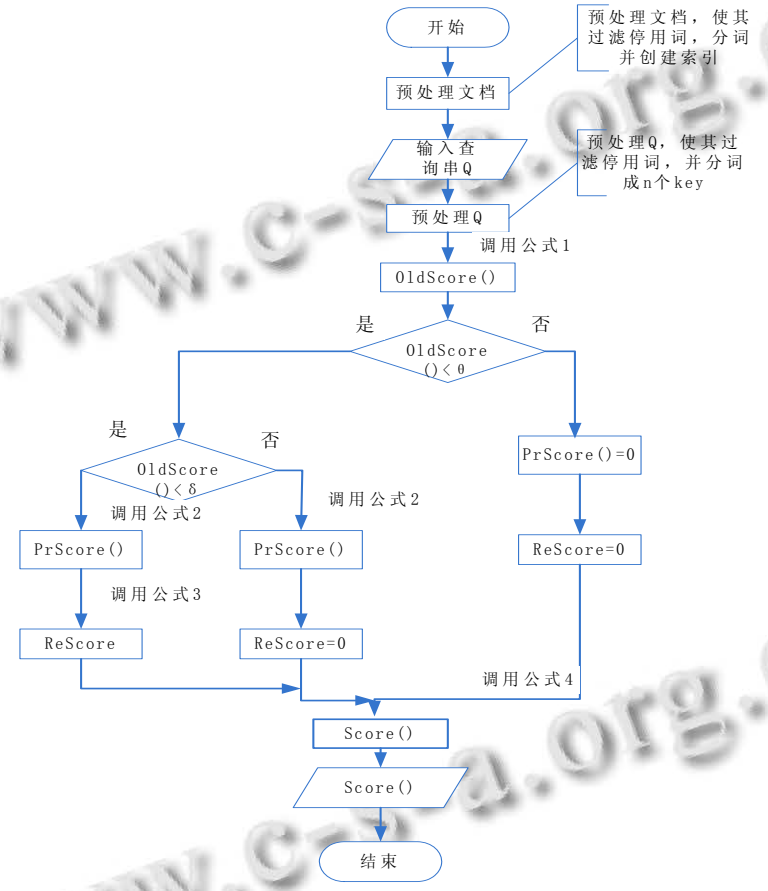

图 3　改进后 Lucene 相似度评分算法流程图

3　实验

3.1 实验数据

本文中问题单数据源是国内设备龙头公司生产设置工作中出现的问题的以及对问题进行描述并存储而形成的数据，针对这个结构的数据进行分析，使得进行检索时可以快速得到需求信息，从而该问题可得到及时的解决. 问题单数据源与一般数据源比较主要特性是文本短、结构固定，特殊问题单数据源由问题单 XML 文档构成，其中每一个单独的问题单是一条记录，除了编号以外，还包括问题单号、简要描述、详细描述四个域，分别从简到详进行描述. 其中问题单号可以看成查询中的权重词，而简要描述和详细描述则可看成对问题单号的具体描述信息. 为增强改进算法的可信度，实验使用包含 87680 条记录文档，大小约为 500MB.

3.2 性能评价指标

实验采用的评测指标为信息检索系统中常用的平均查准率 MAP(Mean of Average Precision)和 P@n(返

© 中国科学院软件研究所 http://www.c-s-a.org.cn

回前 n 个结果的准确率)[13]. 并对改进算法中涉及到的几个参数进行设置值分别取 α = 1、K =0.05、γ =0.01、K_1 =0.6、K_2 =0.3、K_3=0.1、δ =0.2、θ=0.6. 分别采用 Lucene 原始的相似度评分算法和本文改进的算法在系统上针对数据集进行实验, 对检索结果进行评测. 实验评测结果如表 3 所示.

表 3 500MB XML 数据集上的实验评测结果

	MAP	P@20	P@50	P@100
Lucene 内置算法	0.2518	0.4175	0.3636	0.3100
改进后算法	0.3284	0.5262	0.4753	0.4228
提高比例(%)	7.66	10.87	11.17	11.28

实验评测对比结果表明, 相比 Lucene 原始内置的相似度得分算法, 本文改进后的新算法在 MAP 上有了 7%左右的提高, 在 P@n 有了 10%~11%的提高. 新算法是在 Lucene 自身默认评分算法的基础上结合词语位置关系特征以及二次检索来达到提高检索的准确率, 通过改进的新算法较 Lucene 自身默认的评分算法在检查准确率上有较大提高, 验证了本文提出的算法的可行性.

4 结语

1)本文针对特殊问题单的 XML 文件, 选用 Lucene 作为搜索机制. 通过对解析 XML 解析器的比较并选择 Dom4j 作为本系统的解析器. 再则根据 Lucene 自带的中分词器以及第三方分词器的比较, 选择 IK 分词器作为本系统的分词器. 然后, 再针对 Lucene 本身的得分和搜索结果排序不足之处, 进行改进, 使得 Lucene 可以充分利用查询词的位置关系信息以及词项语义来提高检索的准确率. 最后通过以相似性搜索的准确率为目标进行实验, 验证了改进后的系统使检索的结果更优化.

2)针对高效的索引机制, 相似性问题单是动态增长的, 且结构不尽相同, 如何针对这种情形设计高效动态自适应的倒排索引, 来提高索引的空间效率和支持查询的时间效率以及下一步将引入“语义域”的概念, 进一步提高系统的搜索效率以及准确率.

参考文献

1 ApacheLucene.Lucenjava4.5.0.[2013-10-05]. http://lucene.apache.or/.

2 谢谏.基于 Lucene 的 XML 索引与检索[学位论文].广州:华南理工大学,2012.

3 义天鹏,陈启安.基于Lucene的中文分析器分词性能比较研究.计算机工程,2012,38(22):79–282.

4 胡长春,刘功申.面向搜索引擎 Lucene 的中文分析器.计算机工程与应用,2009,45(12).

5 王欢,孙瑞志.基于领域本体和Lucene的语义检索系统研究.计算机应用,2010,30(6):1655–1657.

6 黄承慧,印鉴,陆寄远.一种改进的 Lucene 语义相似度检索算法.中山大学学报(自然科学版),2011,50(2).

7 Doron C, Einat A, Carmel D. Lucene andjuru at trec 2007: 1-million queries track. Proc. of the 16th Text Retrieval Conference (TREC 2007). Gaithersburg, Washington, USA. 2007. 321–327.

8 白培发,王成良,徐玲.一种融合词语位置特征的 Lucene 相似度评分算法.计算机工程与应用,2014(2).

9 朱红权.基于 HowNet 多特征结合的句子相似度计算[学位论文].长沙:湖南大学,2009.

10 Kadhim MH, Omar N. Automatic arabic text categorization using Bayesian learning. 2012 7th International Conference on Computing and Convergence Technology (ICCCT). IEEE. 2012. 415–419.

11 蒋琪夏.相似性搜索中的近似算法研究[学位论文].北京:清华大学,2012.

12 吴代文.基于 Lucene 的二次全文检索系统设计与实现[学位论文].西安:西安电子科技大学,2009.

13 张俊林.这就是搜索引擎核心技术详解.北京:电子工业出版社,2012.

© 中国科学院软件研究所 http://www.c-s-a.org.cn