文献搜索引擎中特征项及权重的应用①

李光敏

(湖北师范学院 计算机科学与技术学院, 黄石 435000)

摘 要: 针对目前用户在使用搜索引擎过程中, 检索结果冗余、效率低下等问题, 本文在对文献垂直搜索系统中 Lucene.Net 的索引算法研究基础上, 结合用户实际专业检索需求, 改变其算法中的激励因子,实验结果证明该方法确能提高搜索结果的相关度.

关键词: 垂直搜索; Lucene.Net 引擎; 索引排序; TF*IDF

Application for Feature and Weight in Search Engine of Literature

LI Guang-Min

(College of Computer Science and Technology, Hubei Normal University, Huangshi 435000, China)

Abstract: As the Web continues to grow, it has become increasingly obvious that information overload, low-efficiency using traditional search engines. In this paper, after researching the index sort algorithm applied in the Lucene.Net and considering the requirement from the end-users, we present an approach to address this issue. The experimental results showed that the proposed approach is practical.

Key words: vertical search; lucene .net engine; index sort; TF*IDF

1 引言

Internet 的出现,使得互联网的信息容量按指数规律飞速增长,用户通常面对的是与自己感兴趣领域无关的由通用搜索引擎所检索到的结果, 浪费不少精力和时间仍难以获取真正需要的信息. 垂直搜索引擎[1]则是针对专业特定的领域和用户检索的偏好而对抓取的信息进行分析、挖掘、筛选, 精准的定位, 从而确保了用户能够迅速准确地获取自己需要的信息.

Lucene 是一个高性能的 Java 全文检索工具包, 由于以其开放源代码的特性、 优异的索引结构、 良好的系统架构等深受各大行业二次开发使用[2]. 本文正是在分析研究 Lucene.net(Lucene 的.Net 版本)的系统结构、索引原理的基础上, 改进其索引的评分算法, 最后实验结果证明改进后的算法在针对期刊论文的垂直搜索系统中有一定的实用价值.

2 相关工作

针对 Lucene 全文索引、检索的特征, 苏潭英[3], 阳奇[4], 吴代文[5]等人设计实现了中文全文数据库的搜索系统. 李永春[6]通过实验发现 Lucene 比传统的检索方式有更快的响应速度. 在中小型搜索引擎搭建应用中, Lucene.net 也有更广阔的应用空间, 李文江[7]完成了具有权限控制的文档全文检索系统, 谭文堂[8]提出了针对海量数据的 Lucene.net 分布式检索系统实现思路. 对于 Lucene 中检索效率和精度问题, 不少同行也提出不同的改进意见, 索红光[9]通过实验分析验证加入中文分词模块和在索引预处理中采用提取特定数量的特征词方法可提高检索性能. 张贤[10]结合 Direct Hit、PageRank 算法, 综合考虑关键词词频与位置关系、用户的行为特征、网页链接关系和响应时间等因素来改进 Lucene 中的排序算法并应用至糖业专业搜索引擎中. 张瑜[11] 针对传统的 TFIDF 算法未能考虑

① 基金项目:湖北师范学院文理学院 2012 教学研究项目(XJ201219)
收稿时间:2013-11-12;收到修改稿时间:2013-12-03

特征项在类间和类内的分布情况, 提出了自己的基于 WA-DI-SI 的特征权重改进算法能获得较好的分类效果. 上述这些系统的实现思路和检索结果排序的改进策略对本文有一定的借鉴意义.

3 Lucene.net总体结构

Lucene.Net 是一个强有力的开源全文搜索引擎, 是从 Apache 的 Lucene(Java)项目移植到.Net(C#)上的. 它不是一个完整的全文检索引擎, 而是一个用 C#写的全文索引引擎工具包, 主要有两个功能: 一是建立索引库, 即将待检索的文本内容分词后建立索引并存储, 二是检索索引库, 即根据查询关键字从索引库中找出符合条件的文档. 它为数据访问和管理提供了函数调用接口, 凭借高性能、可伸缩、低开销的优势被二次开发应用于垂直搜索行业中, 其系统结构与源代码组织结构如图 1 所示:

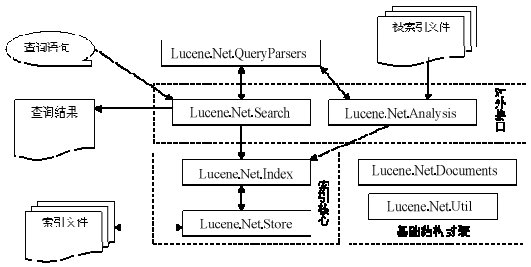

图 1 系统结构与源代码组织结构

由图 1 所知: Lucene.Net 系统由对外接口、索引核心、基础结构封装三个部分组成. 其中直接操作索引文件的索引核心又是系统的重点, 索引核心的功能就是对 Lucene.Net.Analysis 分词后的内容建立索引文件并存储, 最后用户使用查询语句通过 Lucene.Net.Search 检索索引库中的索引文件并将检索结果输出. 不难看出, 在索引文件的建立过程中
其实已经决定了最终检索记录输出的先后顺序(当然分词的准确度也是影响因素之一), 那么如何改进索引排序算法, 才能保证检索结果是最大程度地符合用户需求, 是该论文讨论的重点.

4 Lucene.Net索引算法原理及改进

设计一个高效检索系统的关键是建立一个倒排索引机制, 将数据源(如多篇期刊论文)排序顺序存储的同时, 建立一个排序好的关键词列表, 用来存储关键词(Keywords)和出现该关键词的记录(DocID)文件的关系. 并利用这种映射关系来建立如下结构的索引: 关键词 文章编号[出现频率] 出现位置(包括: 起始位置, 结束位置).

4.1 索引建立

Lucene.Net 的索引建立过程是先通过自定义的 KTDictSegAnalyzer()分析器组件来分词, 然后按照如上的索引结构来建立索引、存储索引, 其中 KTDictSeg 类继承自 Lucene.Net 的抽象类 Analyzer.

建立索引最重要的类是 IndexWriter,其构造函数的三个参数分别表示存放索引的目录、建立索引时使用的分析器、决定是重建索引还是更新原有索引.

3.2 索引的逻辑结构

在 Lucene.Net 中 Term(词)在倒排索引中是最小的单位, 由 N 个 Term 构成 Field(域), 而又有 N 个 Field 组成 Document(记录), N 个 Document 又会组成 Segment(段), N 个 Segment 会组成索引(index)被写到 Lucene.Net 的文件系统中. Lucene.Net 的索引结构类似数据库表的单记录结构. 用户的检索过程实质就是将查询关键词与索引中的 Field 精确匹配, 然后根据 Lucene.Net 内置的评分算法将检索结果按照最大相似度从高到低的顺序显示给用户的过程. 那么深入研究并合理改进其评分算法将对检索结果的显示顺序起着决定性的影响.

4.3 Lucene.Net 的评分机制

Lucene.net 的评分机制采用了信息检索中的空间向量模型和布尔模型相结合的方法, 用来决定给定的文档和查询项的关联程度. 其中用到的 TF*IDF 算法一般被认为能够产生高质量的搜索结果. TF*IDF 由 Salton 在文献[12]中提出, 此后 Salton 多次论证 TF*IDF 在信息检索中的有效性[13], 在 1988 年又详细阐述了多种词权重计算方法在文献检索中使用情况[14].
其中:

$$\sum_{t\,in\,q} tf(t\,in\,d)*idf(t)*boost(t\ field\,in\,d)*lengthNorm(t\ field\,in\,d)$$

(1) tf(t in d)即词条频率(Term Frequency), 表示检索词条 t 在文档 d 中出现的总次数. 在 Lucene.Net 中它

的值是词条出现总次数的平方根.

(2) idf(t)即反转文档频率(Inversed Document Frequency), 表示含有词条 t 的文档总数的反向影响值[3], 由 log 函数可知在文档中出现次数越少的词条, 其贡献的分数越高.

(3) boost(t field in d)在建立索引过程中对每个 field 设置的权重值(也称激励因子), 它的默认值是"1", 为了使某文档在检索结果中靠前显示, 即提高它的得分, 可通过增加相应 field 的权重值实现.

(4) lengthNorm(t field in d)是一个长度因子, 它的值由词条 t 所在相应 field 中的总数目来决定, 它的计算方法是: $1.0/\sqrt{numTerms}$, 其中 numTerms 参数为该 field 内词条的总书目, 在建立索引过程中计算出并存储在索引中. 所以如果 field 中的内容越少, 那么词条 t 的命中率越低, 从而其长度因子就越大, 导致影响该文档的分值就越高.

4.4 算法调整

(1) 调查数据表明,用户在检索期刊文献数据库时常关注检索关键词是否包含在如下四类内容中, 按照关注度高低顺序依次是: 关键词、主题、作者名、标题.

(2) 对检索的期刊文章 90%以上的用户关注与自己研究方向紧密相关的专业期刊, 不足 10%的用户会关注与自己研究方向关系不大的其他专业期刊, 据分析这部分用户可能在写交叉性学科论文时会参考其他专业的相关术语.

通过上面对影响文档排名的各因素因子的研究, 结合投票调查结果, 根据用户的搜索偏好和专业方向, 通过该期刊引擎来揣测用户搜索意图, 最先提供其相似度最高的期刊文章链接和实现个性化服务推送功能, 如同样输入检索关键词"数据挖掘", 那么靠前显示的检索结果对矿业工程方向的用户是有关采掘工业的论文; 对计算机信息科学方向的用户来说是有关该算法原理和应用等方面的论文. 这两个方面是本文的研究重点. 基于此, 对 Lucene.Net 评分算法提出如下改进方案.

(1) 根据调查结果, 通过 setBoost 函数提高用户所最关心的四个 Field(域)的权重值, 这样可以使得分值较高的检索记录较靠前显示.

(2) 在 Lucene.Net 本身的索引排序算法中, boost(t field in d)是一个被弱化的因子, 这个值被设置在 AbstractField 的对象属性中, 在索引建立时一旦生成就不再发生变化. 因此, 我们在 AbstractField 中增加一个专业方向 Major 域并设置它的 boost 值, 用来记录各用户的专业方向, 此时 boost(t field in d)变成了 boost(t field in d, major).

5 实验结果对比分析

我们以维普期刊网中随机采集的 6000 篇涉及三个专业方向的期刊论文为训练样本, 其中计算机应用方向占 2000 篇、矿业工程方向占 2000 篇和农业信息学方向占 2000 篇. 该算法中用户专业方向的相关度 Relation 定义为:

$$R=\sum_{i=1}^{n}P_iW_i$$

其中 $n=\left\lceil\frac{N}{M}\right\rceil$ 表示检索结果总页数, N 表示检索结果总条数, M 表示平均每页所含检索结果条数, $P_i=\frac{O_i}{M}$, O_i 表示第 i 页中出现检索用户相关专业的总记录条数. W_i 表示权值, 由此表示该页的检索结果对用户的重要程度, 并且 $\sum_{i=1}^{n}W_i=1$, 根据用户的检索偏好(一般只关注检索结果的前几页), 我们因此规定 $W_i>W_j$ (i<j), 即越靠前的页面对用户的重要性程度越高. 该实验中共有三页检索结果, 我们取 w_1 =0.6, w_2 =0.3, w_3 =0.1, 后期可通过对用户浏览的历史记录采用机器学习来修正各检索页的权值 w, 从公式可以看出最靠前的检索结果页对 R 的影响程度较大. 在实施改进方案一之前, 即关键词、主题、作者、标题的 boost 默认值均为 1.0 时, 检索结果与用户专业方向相关度都是相同的, 所以 R 的值为其平均值(33.3%), 实施改进方案一后的数据对比如下:

表 1　改进方案一后的专业方向相关度 R 平均值

Keywords	Subject	Author	Title	R 均值（%）
1.4	1.3	1.2	1.1	65.9
1.5	1.4	1.3	1.2	66.3
1.6	1.5	1.4	1.3	70.1
1.7	1.6	1.5	1.4	72.3

其表1中行记录的boost值设置标准是按照用户对其所属字段的关注程度来依次递减(步长值设为 0.1), 由表 1 看出在提高了各字段 boost 值(列值)后, 其专业方向相关度的 R 均值有了不同程度的提高, 这意味着用户较关注的四个检索条件(关键词、主题、作者、标题)中含有与检索内容更相关的论文记录在检索结果中靠前显示. 但是用户可能对靠前显示的较多的非专业方向论文不大感兴趣, 为此在方案一的改进基础上, 增加专业方向域(Major)并设置 boost 的值后重新建立索引, 其数据对比如下:

表 2 改进方案二后的专业方向相关度 R 平均值

Keywords	Subject	Author	Title	Major	R 均值（%）
1.4	1.3	1.2	1.1	1.1	73.1
1.5	1.4	1.3	1.2	1.2	74.3
1.6	1.5	1.4	1.3	1.3	76.1
1.7	1.6	1.5	1.4	1.4	82.6

由表 2 看出在原改进方案基础上, 引入专业方向域(Major)并递增其 boost 值后, R 均值较之以前有了明显的提高, 这说明在首页的检索结果中, 用户感兴趣的自己专业方向的论文记录占较大比例(82.6%), 因此可获知与自己专业方向相关的更多信息. 从实际出发, 在保证用户检索结果的准确性和全面性以及系统建立索引效率的前提下, 我们在方案二中采纳专业方向(Major)的 boost 值为 1.4 的组合应用在该期刊检索引擎中.

5 结语

本文通过分析研究基于 Lucene.net 开发的文献搜索引擎中的所用索引排序算法 TF*IDF, 结合用户实际需求改进其算法使检索结果更具有针对性、个性化. 实验证明改进后的算法可行并具有一定的实用价值. 但在用户所属专业方向的分类上目前还是采用用户注册系统后手工填写专业方向, 后期提高训练样本数量, 采用对搜索结果的用户访问日志进行聚类, 以提高检索结果的准确度和相关度.

参考文献

1 Zhou K, Cummins R, Lalmas M, et al. Which vertical search engines are relevant. Proc. of the 22nd international conference on World Wide Web. International World Wide Web Conferences Steering Committee. 2013. 1557−1568.

2 McCandless M, Hatcher E, Gospodneti O. Lucene in action, 2nd edition. Manning Publications Co. 2010.

3 苏潭英,郭宪勇,金鑫.一种基于 Lucene 的中文全文检索系统.计算机工程,2007,33(23):94–96.

4 阳奇,林镇蚰,黄帆.基于 Hibernate 搜索的数据库全文检索系统.计算机工程,2010,36(4):74–76.

5 吴代文,杨方琦.Lucene 在数据库全文检索中的性能研究.微计算机应用,2011,32(6):53–59.

6 李永春,丁华福.Lucene 的全文检索的研究与应用.计算机技术与发展,2010,20(2):12–15.

7 李文江,陈诗琴.基于 Lucene. net 全文检索在文档管理中的应用.现代图书情报技术,2010,21(11):84–89.

8 谭文堂,贺明科,李阜.基于 Lucene. Net 的分布式全文检索系统.计算机应用与软件,2009,26(9):142–145.

9 索红光,孙鑫.针对中文检索的 Lucene 改进策略.计算机应用与软件,2009,26(6):175–177.

10 张贤,周娅.基于 Lucene 网页排序算法的改进.计算机系统应用,2009,18(2):155–158.

11 张瑜,张德贤.一种改进的特征权重算法.计算机工程,2011,37(5):210–212.

12 Salton G, Yu CT. On the construction of effective vocabularies for information retrieval. ACM SIGPLAN Notices. ACM,1973, 10(1): 48–60.

13 Salton G, Fox EA, Wu H. Extended Boolean information retrieval. Communications of the ACM, 1983, 26(11): 1022–1036.

14 Salton G, Buckley C. Term-weighting approaches in automatic text retrieval. Information Processing & Management, 1988, 24(5): 513–523.