基于内容的中文文本检索方法①

王 婧, 王新房

(西安理工大学 自动化与信息工程学院, 西安 710048)

摘 要: 随着信息量的急剧增加, 检索技术显得尤为关键. 目前很多检索技术都是基于索引的检索技术. 文中借助了 Lucene 的索引技术与检索机制, 通过对索引信息的改进以及使用基于内容的改进方法, 对 Lucene 结果与查询语句在向量空间中重新计算相似度, 实现了对长段查询语句检索结果排序位置的提高.

关键词: Lucene; 向量空间模型; 索引; 相似度

Chinese Text Retrieval Method Based on Content

WANG Jing, WANG Xin-Fang

(School of Automatic and Information Engineering, Xi'an University of Technology, Xi'an 710048, China)

Abstract: With the rapid increase of information, retrieval technology is becoming more and more important. At present, many retrieval techniques are based on index retrieval techniques. This paper studies the search and index technology of lucene and recalculated the similarity of Lucene result and query in the Vector Space Model to improve the sort location of long query search result through the improved index information and the improved method based on content.

Key words: vector space model; lucene; index; similarity

随着科技的飞速发展, 文本信息量越来越多, 如何在海量的信息中获取自己真正想要的信息成为一个巨大挑战. 基于内容的检索是根据媒体对象的语义和上下文联系进行检索. 这种检索技术突破了传统的基于关键字的检索的局限, 直接对对象的内容进行分析, 抽取特征和语义, 并建立索引进行检索. 如果返回的检索结果是按顺序检索的响应时间给出, 那么检索的过程会变的乏味冗长. 为了解决这种问题, 文本文档库采用建立索引的技术缩短检索时间. 基于内容的检索技术往往也都是基于索引的[1], 它非常适用于大规模、稳定的或周期性变化的文本文档的检索. 文中将引入 Lucene 平台, 在其已有的基础上进行相关检索结果改进, 以便获得更好的检索效果.

1 Lucene概述

Lucene 是用 Java 编写的全文检索引擎工具包, 它可以方便地嵌入到各种应用中实现全文索引和检索功能. Lucene 有一套自己的索引、检索机制与结果排序方法, 可以较为方便快捷地进行检索. 但索引与检索两者是相互独立的, 这使得开发人员可以根据需要对它们进行二次扩展.

Lucene 源码中共包括 7 个子包, 每个包完成特定的功能, 具体内容如表 1 所示:

表 1 Lucene 源码包对应功能表

Lucene 源码包名	功能
Org.apache.lucene.analysis	语言分析器, 主要用于分词
Org.apache.lucene.document	索引存储时的文档结构管理
Org.apache.lucene.index	索引管理, 提供库的读写接口
Org.apache.lucene.queryParser	查询分析器, 实现查询关键词间的运算
Org.apache.lucene.search	检索管理, 根据查询条件得到结果
Org.apache.lucene.store	数据存储管理
Org.apache.lucene.util	公共类

① 收稿时间:2012-01-10;收到修改稿时间:2012-02-29

在 Lucene 中, 一般分为以下几个步骤运作:

通过添加一系列多个字段(Fields)来创建一批文档(Documents)对象; 创建一个索引器(IndexWriter)对象, 并且调用它的 AddDocument( )方法来添加进 Documents; 调用 QueryParser.parse( )处理一段文本(string)来建造一个查询对象; 创建一个 IndexReader 对象并将查询对象传入到它的 search( )方法中进行检索; 调用 Hits( )类, 取得 Lucene 搜索的结果.

需要说明的是, Lucene 使用倒排索引技术, 这种索引技术可以通过文档中的特征词来查找到有哪些文档包含它们, 从而更快的检索到与查询相关的文档, 这是本文选择 Lucene 作为检索平台的一个原因.

另一点要说明的是 Lucene 可以使用 QueryParser 来处理一段文本, 而不仅仅处理一句话或者几个关键词, 这对于要从内容上满足用户查询需求是很必要的.

第三点要说明的是, Lucene 中带有的 IndexReader 对象和 Hits( )类, 对于调用文本信息非常方便. 通过 IndexReader 可以获取 Lucene 检索结果的词项、词频等信息, Hits( )类可以读出指定序号的 Document、或者得到 Searcher 检索出来的文档每个文档的分值, 这个分值可以作为自然排序的依据. 它是根据 Lucene 内部自有的一套计算得分公式得出, 具体公式是:

$$\sum tf(t)\cdot idf(t)\cdot boost(t.field)\cdot lengthNorm(t.field) \quad (1)$$

其中的参数解释如下:

tf(t)——检索词条在某个文档中总共出现的次数;

idf(t)——表示反转文档频率;

boost(t.field)——对每个 field 设置的一种激励因子, 增加某个 field 的重要性;

lengthNorm(t.field)——长度因子, 由词条所在 field 的总长度来决定. 一个 field 内词条越多, 则其长度因子越小.

2 检索结果排序优化方法

目前, 国内对 Lucene 的研究和应用多是集中在对各个模块功能的研究和分词方法的改进上, 以及直接将其应用到检索系统中[2]. 检索的效果多是由包含相关文档的数量是否满足用户需求来衡量的. 本文在 Lucene 框架之上扩展了 Lucene 的分词器, 使用盘古分词优化了分词效果. 在此基础上进行文本预处理, 对出现频率超过 80%的词项(即停用词)进行删除. 这样处理后的语料再应用 Lucene 的检索机制就得到了基础的检索结果. 通过对以上过程结果的分析发现, 即使扩展了 Lucene 的分词器, 对文档进行了预处理, 基于内容的检索结果也不尽如人意. 所以本文加入了以下的优化步骤: 一、在索引文档时进行预处理改进; 二、进行基于内容的检索算法改进.

2.1 索引文档的预处理改进

中文的词数量大于字的数量. 若为文本中所有词项创建索引, 将会造成索引中的词过度拥挤, 降低检索的精度, 所以本文采用的改进步骤如下:

Step1: 对文档分词, 删除停用词;

Step2: 统计剩余词的词频. 如果索引的文档数量大于 10 篇跳转到 step4, 小于 10 篇下一步;

Step3: 选取词频排前 50-100 的词作文档特征项;

Step4: 计算每个词的 TF-IDF 值, 根据值的大小降序排列, 选取排名前 50-100 的词作为文档特征项;

Step5: 根据以上步骤, 将选取出的特征项代替文档内容来建立索引.

2.2 基于内容的检索算法改进

Stpe1: 对于用户输入的查询语句使用 QueryParse 进行解析;

Step2: 利用 Lucene 在索引文件中找出包含特征项的文档集合;

Step3: 通过计算检索表达式中特征项的 TF-IDF 权重以及文档集合中每篇文档中特征项的 TF-IDF 权重, 可以分别构成查询特征向量和文档特征向量;

Step4: 根据向量空间模型[3]计算文档与查询表达式之间的相似度, 得分越高说明二者相似度越大;

Step5: 对相似度最大的前 N 个文档进行输出.

另外, 在计算文档与查询的相似度之前, 本文采用了两种方法对文本向量进行表示.

(1) 词频法 TF. 向量由每个词在文档中出现的次数构成.

(2) TF-IDF. 向量通过 TF-IDF 权重进行表示, TF-IDF 方法需要考虑三个因素[4,5]:

① tf_{ij} ——特征项 t_j 在文档 D_i 中出现的次数, 也就是词频;

② idf_j —— $\lg\left(\frac{d}{df_j}+0.01\right)$, 其中 d 表示文档集中所有的文档篇数; df_j 为包含特征项 t_j 的文档的篇数. 这就是逆文档频率(IDF);

③归一化因子——对各分量进行标准化. 根据以上三个因素, TF-IDF 可以用式(2)表示:

$$w_{ij}=\frac{tf_{ij}\times\log(d/df_j+0.01)}{\sqrt{\sum_{t_j\in D_i}[tf_{ij}\times\log(d/df_j+0.01)]^2}} \quad (2)$$

如果文档出现固定的结构, 比如含有标题、摘要、关键词、正文这些部分, 考虑引入位置加权方法.

对于词频法, 权重设置标题得 4 分; 关键字得 3 分; 摘要得 2 分; 正文得 1 分.

对于 TF-IDF, 设置 k_1, k_2, k_3, k_4 分别为标题、摘要、关键词、正文的权重系数, 数值大小分别为 4、2、3、1. 这样设置的原因是标题中出现的特征项最能代表文档内容, 其次是关键词, 最后才是正文内容. 改进后的 TF-IDF 计算公式为

$$d_{ij}=(k_1\cdot tf_{1j}+k_2\cdot tf_{2j}+\ldots+k_4\cdot tf_{4j})\times idf_j \quad (3)$$

3 实验设置与结果

3.1 语料设置

本文使用的语料集包括 500 篇 TXT 格式的文档. 针对指定文档的内容, 专门设定了 10 条长文本的查询语句作为测试样本, 即每输入其中一个查询语句, 就已知检索结果最好的那一篇文档. 这 10 条查询语句的内容符合以下要求: 1)专门针对某一篇文档的某一段内容进行改写; 2)尽量用同义词代替原文档中词项, 尽量不出现重复词项; 3)可以变换句型, 尽量体现出原文档内容的含义.

3.2 实验结果

用 Lucene、改进内容的 TF、TF-IDF 进行相同查询语句的测试, 得到的最佳文档在总检索结果中的位置如表 2 所示, 其中 Q1-Q10 代表 10 条查询语句.

表 2　检索出的最佳文档排序位置

	Q1	Q2	Q3	Q4	Q5
Lucene	14	6	6	8	6
TF	8	5	6	6	3
TF-IDF	3	1	1	2	1
	Q6	Q7	Q8	Q9	Q10
Lucene	7	3	12	2	13
TF	5	3	8	2	6
TF-IDF	1	1	3	1	2

结果表明, Lucene 对于长段查询的检索能力不能令人满意; 采用改进内容的 TF-IDF 方法的检索结构优于其他两种方法.

对以上三种方法进行系统性能指标检测, 分别测试相同的 100 篇文章得到的查准率、查全率结果如表 3 所示.

表 3　系统评价指标

系统使用方法	Lucene	改进基于内容 TF	改进基于内容 TF-IDF
查全率%	89	99	100
查准率%	53	65	85

从这两个指标也显示出了同样的结论, 经过改进基于内容的 TF-IDF 方法得到的效果是这三种方法中最好的.

4 结语

本文提出了一种基于内容的中文文本检索方法. 利用 Lucene 的索引模块和检索机制, 先通过索引文件预处理提高索引质量, 再将查询语句与 Lucene 得到的检索结果根据向量空间模型进行相似度计算. 经过与 Lucene 原有结果和用词频 TF、TF-IDF 表示的模型计算结果比较, 得到 TF-IDF 法再经过二次检索后能很好的体现特征项权重, 区分文档, 从而得到一个更优的排序结果. 相比于传统基于关键词的文本检索, 改进基于内容的 TF-IDF 方法可以有效地应用于长篇文章检索.

参考文献

1 肖明.基于内容的多媒体信息索引与检索概论.北京:人民邮电出版社,2009.63–94.

2 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,2009,25(6):175–177.

3 王晓黎,王文杰.基于向量空间模型的文本检索系统.微电子学与计算机,2003,23(6):188–190.

4 Grossman DA, Frieder O. Information Retrieval: Algorithms and Heuristics(Second Edition).北京:人民邮电出版社,2010. 10–20.

5 郭庆琳,李艳梅,唐琦.基于 VSM 的文本相似度计算的研究. 计算机应用研究,2008,25(11):3256–3258.