基于领域模型的网页搜索排序算法①

潘 澄, 吴共庆, 李 磊, 胡学钢

(合肥工业大学 计算机与信息学院, 合肥 230009)

摘 要: 通用搜索引擎在检索过程中会出现查询结果与关键词所属领域无关的主题漂移现象. 本文提出了面向特定领域的网页重排序算法—TSRR(Topic Sensitive Re-Ranking)算法, 从一个新的视角对主题漂移问题加以解决. TSRR 算法设计一种独立于网页排序的模型, 用来表示领域, 然后建立网页信息模型, 在用户检索过程中结合领域向量模型和网页信息模型对网页搜索结果进行重排序. 在爬取的特定领域的数据集上, 以用户满意度和准确率为标准进行评估, 实验结果表明, 本文中提出的 TSRR 算法性能优异, 比经典的基于 Lucene 的排序算法在用户满意度上平均提高 17.3%, 在准确率上平均提高 41.9%.

关键词: 领域模型; 网页信息模型; 网页重排序

Web Page Re-Ranking Algorithm for Specific Domain Based on Domain Model

PAN Cheng, WU Gong-Qing, LI Lei, HU Xue-Gang

(School of Computer and Information, Hefei University of Technology, Hefei 230009, China)

Abstract: General search engines often cause the topic-drift problem, which means that during the retrieval process, some of the retrieval results are independent to the domain keywords. We propose a web page re-ranking algorithm for a specific domain—the TSRR(Topic Sensitive Re-Ranking) algorithm to solve the problem from a specific perspective. TSRR establishes a vector model which is independent to page rank for a specific domain and a web page information model; then it combines the vector model and the web page information model to re-rank the search results in the retrieval process. TSRR's performance is evaluated based on the criteria of customer satisfaction and precision. Experiment results on the dataset crawled for specific domains show that TSRR is excellent in performance. Compared with the ranking algorithm from Lucene, TSRR can promote the customer satisfaction performance by 17.3% and the precision performance by 41.9% on average.

Key words: domain model; web information model; re-ranking

1 引言

随着互联网技术的迅猛发展, 网络成为人们获取信息的重要渠道, 搜索引擎成为人们获取信息时使用最多的互联网工具之一. 从最早的商业搜索引擎 Archie 开始, 互联网包含的网页数量以每天百万数量级的速度爆炸式增长. 根据中国互联网信息中心的《第 34 次中国互联网发展状况统计报告》, 截至 2014 年 6 月, 我国搜索引擎用户规模达到 5.07 亿, 使用率为 80.3%, 用户规模较 2013 年 12 月增长 1783 万人, 增长率为 3.6%. 搜索引擎如何快速、高效、正确地给用户返回所查询结果, 成为目前搜索引擎所面临的最大挑战之一.

现阶段, 绝大部分搜索引擎都是通用搜索引擎, 网页排序结果与检索的关键词之间是相对独立的. 从网页重要性角度来看, 此类排序方式会将较为重要的网页放到排序结果的前面. 但从另一个角度来看, 由

① 基金项目:国家高技术研究发展计划(863)(2012AA011005)
收稿时间:2015-03-11;收到修改稿时间:2015-04-15

于网页排序结果与检索关键词所在领域无关, 那么对于任意领域来说, 网页的排序结果都相同, 就会出现主题漂移现象. 这样会带来一个问题: 如果用户只是对某个领域的内容比较感兴趣, 但是由于通用搜索引擎的领域无关性, 会导致检索返回的结果中, 用户感兴趣的领域的网页并没有被放在前 10 至 20 条结果中, 而大部分用户的检索习惯是对前 10 至 20 条结果比较关心[1], 这样会降低用户的体验, 甚至导致对特定领域较为关注的用户流失.

在这样的背景下, 面向特定领域的网页排序就显得尤为重要. 第一, 面向特定领域的网页排序可以为用户提供更加准确的搜索结果; 第二, 面向特定领域的搜索引擎的开发可以使用户的检索更具有针对性.

本文主要贡献有: 1)提出了一种领域向量模型的设计与构建方法; 2)设计并实现了一种面向特定领域的网页重排序算法. 本文在二章介绍相关工作; 在第三章描述网页排序算法; 在第四章给出实验结果; 在第五章作出总结.

2 相关工作

目前, 最为流行的网页排序算法是由斯坦福大学 Sergey Brin 和 Lawrence Page[2]在 1998 年提出的 PageRank 算法, 该算法于 2005 年被评为数据挖掘领域十大经典算法之一[3]. 以该算法为核心开发的 Google 搜索引擎在商业方面取得了极大成功. 这也引发了人们对于网页排序算法研究的兴趣.

但 PageRank 算法并不是完美的, 该算法存在主题漂移(Topic-drift)现象. 因为 PageRank 算法仅仅对网页的链接结构进行了分析, 难以区分网页中的超链接与该网页的主题是否相关, 这样会导致在最终的网页排序结果中出现很多与查询主题无关但是排名很靠前的网页, 出现主题漂移现象. 针对 PageRank 算法存在的这个问题, 一些国外学者提出了相应的改进算法. 其中, 文献[3]提出了一种主题敏感(Topic-sensitive)的 PageRank 改进算法(TH-PageRank 算法), 文献[4]提出了一个结合链接分析和文本内容的 PageRank 改进算法(MP-PageRank 算法). 这两种算法都需要利用查询主题以外的信息来提高对网页的辨识能力, 以减少主题漂移现象的发生. 但这两种算法仍然存在缺陷, TH-PageRank 算法需要利用查询关键词的上下文才能有效地进行主题类别判断. 而 MP-PageRank 算法, 不仅需要文本内容支撑, 而且时间复杂度和空间复杂度都比 PageRank 算法扩大了 N 倍(N 为搜索引擎涵盖的网页数量), 对于网页量十分巨大的互联网络, MP-PageRank 算法对空间和时间的需求都难以满足实际应用的需要[5].

同时, 国内学者也在网页排序算法方面提出了很多创新的算法. 文献[6-12]提出了在传统网页排序算法基础上改进的排序算法, 但是这些算法都没有解决主题漂移问题. 在此基础上, 国内另一些学者致力于研究基于领域主题的 PageRank 算法改进技术. 文献[5]在分析 PageRank 算法及其有关改进算法的基础上, 提出了基于虚拟文档的主题相似度模型和基于主题相似度模型 TS-PageRank 算法框架. 文献[13]从改进计算模型的传递概率和跳转概率的角度, 分析已有的网页排序算法的特点, 给出了一种面向主题的网页排序算法. 文献[14]提出了基于链接分析的网页排序算法. 然而这些算法依然存在一些问题, 文献[5,14]中的算法效率相对较低, 而文献[13]中的算法并没有考虑到链接与特定领域之间的相关性度量.

已有工作从不同的视角引入领域知识, 以解决网页排序算法的主题漂移问题. 然而, 这些工作使用的领域知识和算法有较强的耦合性, 均未形成一个基于领域知识解决该问题的框架性模型. 为此, 本文探索研究在特定领域模型的基础上开展网页排序算法研究, 从一个新的视角解决主题漂移问题.

3 面向特定领域的网页重排序算法

本文研究的主要内容是面向特定领域的网页重排序算法, 其中领域的具体表示方法以及网页重排序算法是研究的重点.

3.1 领域的表示方法

3.1.1 领域概述

领域一词来源于人工智能学科[15]. 在人工智能领域, 它主要应用在基于知识的专家系统和自然语言理解系统中. 本文设计了领域向量模型来表示领域知识. 领域向量模型是判断一个网页是否属于某个领域的依据, 也是网页重排序算法的基础. 领域向量模型应该满足下面几个特征: ①能够表示该领域的基本特征, 并且其中的元素不能再分割为更细的元素; ②能够最大限度的表示该领域的大部分特征, 其中每个元素都可以表示领域某一方面的特征; ③领域模型中的元素

不能有二义性. 使用中文词汇来表示领域特征的一个挑战就是同一个中文词汇在不同领域有不同的含义, 这一点在模型构建阶段是需要避免的.

3.1.2 领域的表示方式

通过对大量网页的分析, 我们发现, 属于同一领域的网页文本中往往包含着一些类似的领域特征词. 例如在学术领域中, “教授”、“报告”、“学院”等领域特征词都会有较高的出现频率, 而相对应的, 网页文本中如果出现了“教授”、“报告”、“学院”等词, 那么网页就可能属于学术领域. 于是我们使用领域特征词向量的形式来表示领域, 称为领域向量模型, 具体的表现形式下:

$$\boldsymbol{Topic} = (t_1, t_2, ..., t_n)$$

其中, $\boldsymbol{Topic} = (t_1, t_2, ..., t_n)$ 表示某个领域的向量模型, $t_i (i = 1, 2, ..., n)$ 是其中的元素, 描述的是特征, 每一个特征的具体表现形式为一组关键词, 这组关键词的提取过程将在第 3.2 节领域向量模型构建算法中详细说明. 利用向量表示的好处如下: ①表示方法较为简单; ②可以很明确的看出领域的特征; ③领域向量模型是为之后的网页重排序工作打基础, 这样的表现形式有利于实现相似度计算, 并且计算效率很高. 通过以上的分析, 特征词向量是一种合适的领域表现形式.

3.2 领域向量模型的构建算法

领域向量模型的表示形式是特征词向量的形式. 其中的关键技术有网页文本提取、中文分词、统计词频、添加同义词、赋予权重等. 由于中文具有一词多义的特殊性, 机器剔除的效果往往达不到要求, 于是在领域向量模型的构建过程中, 我们加入了人工操作. 构造方法如下:

步骤 1: 利用网络爬虫爬取特定领域的网页, 提取网页内容文本, 并对内容文本做中文分词;

步骤 2: 对分词结果进行词频统计, 并且剔除掉其中的停用词和对于领域没有贡献的词. 停用词指的是类似“的”、“啊”的常见词, 对于停用词, 我们利用停止词列表, 将分词结果中的停止词剔除, 而对领域没有贡献的词, 我们使用的是人工剔除的方式. 词频统计用的是平滑的方法, 如公式(1):

$$W_{tf} = 1 + \log(tf) \quad (1)$$

其中 tf 指的是词频. 使用对数是为了避免词频差距太大而造成贡献值的差距过大, 数字 1 是为了提供一种平滑机制, 避免出现了一次的词被过滤掉. 这样做可以满足领域模型的特征 1;

步骤 3: 对余下的高频词进行人工处理, 选取那些符合条件的词条, 并根据词频来考虑是否加入到领域向量模型中, 这里的人工处理是为了消除领域模型中特征的二义性. 到这一步, 初步建立领域向量模型;

步骤 4: 对于领域向量模型中的每一个特征加入其与领域相关的近义词, 近义词的数量一般不止一个. 该步骤主要方法是利用哈工大信息检索研究中心同义词词林扩展版 1 对领域中的词条加入近义词, 并人工的除掉与领域无关的近义词, 这样做可以满足领域模型的特征 2;

步骤 5: 为领域向量模型中的关键词赋于权重. 到这一步, 领域向量模型建立完毕.

领域向量模型容量较小, 本文设计其维度为 100 维, 存储载体可以选择数据库、XML 文件、文本文件等. 在之后的网页排序算法中需要频繁的利用领域向量模型进行计算, 因此将领域向量模型在系统运行时装入内存, I/O 读取效率会更高.

3.3 面向特定领域的网页重排序算法

本节首先介绍网页信息模型的建立方法, 并给出网页与用户检索词之间相似度的度量方式, 网页与领域向量模型之间相似度的度量方式, 最后给出面向特定领域的网页重排序算法.

3.3.1 网页信息模型的建立

网页信息通常都是由 HTML 语言描述的, 是基于 HTML 语言的半结构化文本[16]. 这类文本的特点是结构清晰, 内容较为丰富. 本文中网页信息模型的操作对象是网页标题文本以及网页内容文本, 这两类文本能够对网页的内容起到非常明确的表示作用. 为了方便之后的相似度计算, 并且根据 3.2 节的讨论, 我们使用关键词向量的形式来表示网页信息模型. 具体步骤如下:

步骤 1: 利用网络爬虫爬取领域相关网页, 提取网页内容文本, 对其做中文分词.

步骤 2: 建立一个与领域向量模型维度相同的网页信息向量, 向量中每一项初始化为 0. 将网页标题和网页文本的分词结果与领域向量模型进行比对, 如果领域向量模型包含分词结果中的某个词, 就将网页信息向量中这个位置的值设置为非零项.

步骤 3: 根据网页文本中词出现的频率, 对向量中的非零项进行加权, 出现频率较高的词拥有较高的权

重. 到这一步, 网页信息模型建立完成.

3.3.2 相似度度量方法

结合领域向量模型和网页信息模型, 通过下面两种方式分别计算网页与用户检索关键词之间的相似度, 网页与领域向量模型之间的相似度.

1) 针对所有网页, 计算用户检索关键词和网页信息之间的相似度. 计算的方法是利用 BM25 检索模型, 如公式(2)和公式(3):

$$score_1=\sum_{i=1}^{n}IDF(q_i)*\frac{f(q_i,D)*(k+1)}{f(q_i,D)+k*(1-b+b*\frac{|D|}{avgdl})} \tag{2}$$

其中

$$IDF(q_i)=\lg\frac{N-n(q_i)+0.5}{n(q_i)+0.5} \tag{3}$$

公式(2)中, q_i 是检索的某个关键词, $f(q_i,D)$ 是在网页 D 中, q_i 出现的频率, $|D|$ 是网页 D 的长度, $avgdl$ 是网页 D 的平均长度, k 和 b 用来调整精度, 一般分别取为 2 和 0.75. 公式(3)中, N 表示网页的总数目, $n(q_i)$ 表示包含 q_i 的网页的总数目.

2) 针对所有网页, 计算网页信息模型和领域向量模型的相似度. 网页模型和领域向量模型可以表示为如下形式:

$$\boldsymbol{Topic}=(t_1,t_2,...,t_n)$$

$$\boldsymbol{Web}=(s_1,s_2,...s_n)$$

其中 n 是模型维度, $t_i,s_i(i=1,2,...,n)$ 可以是 0, 若为非零项就是相应权值, 网页信息模型与领域向量之间相似度计算利用余弦相似度方法, 如公式(4):

$$score_2=\frac{\sum_{i=1}^{n}t_i*s_i}{\sqrt{\sum_{i=1}^{n}t_i^2}*\sqrt{\sum_{i=1}^{n}s_i^2}} \tag{4}$$

3.3.3 面向特定领域的网页重排序算法

面向特定领域的网页重排序算法具体步骤如下:

步骤 1: 利用公式(3)和(4)对每个网页计算其与用户检索关键词的相似度, 可以得到所有网页的第一步的得分 $score_1$. 具体在图 1 的第 2 行;

步骤 2: 利用公式(5)对每个网页计算其与领域向量模型的相似度, 可以得到所有网页的第二步的得分 $score_2$. 具体在图 1 的第 3 行;

步骤 3: 网页时间也是我们需要考虑的因素之一, 更新的网页应当在排序结果中排序更靠前. 因此我们对网页的 HTML 代码进行分析, 从中抽取出网页的发布日期, 并对日期赋予一个分数 $score_3$. 这个分数是按照日期顺序从大到小, 离当前日期越近, 则分值越高. 具体在图 1 的第 4 行;

步骤4: 对所有的三个分数进行加权求和, 得到最后的分数计算公式如公式(5):

$$score=\alpha*score_1+\beta*score_2+score_3 \tag{5}$$

其中 $\alpha+\beta=1$, 在图 1 的第 5 行到第 8 行.

对最后的得分 $score$ 进行排序, 就可以得到最终的排序结果, 其中的参数 α,β 的值将在 4.3 节的实验中给出具体取值. 具体的算法如图 1.

```
Input: query, pageList, model
Output: result
1: for each page in pageList{
2:   score1 =BM25(query, page)
3:   score2 =cosine-relative(page, model)
4:   score3
5:   score=α*score1+β*score2+score3
6:   result.add( score , url)
7:   result.sortby( score )
8:   return result
9: }
```

图 1 网页重排序代码

4 实验及实验结果

4.1 实验数据集

本文研究的是面向特定领域的网页排序算法, 在本次实验中, 我们的领域选择为学术、娱乐、体育三个领域. 考虑到信息的权威性, 我们的实验数据选择如下:

学术领域的数据为从合肥工业大学, 中国科技大学以及科学网爬取的新闻网页;

娱乐领域的数据为从新浪网、腾讯网等门户网站爬取的娱乐版块新闻网页;

体育领域的数据为从新浪网、腾讯网等门户网站爬取的体育版块新闻网页;

4.2 评价标准

本文所研究的算法是为了后续的搜索引擎搭建做的准备工作, 所以搜索引擎的评估指标在本文中同样适用. 根据文献[1,2]中的评价标准, 本文以下几个标准作为评价指标:

1)用户满意度:

$$satisfication=\frac{1}{n}\sum_{i=1}^{n}F_i \tag{6}$$

其中, n 表示用户的数量, F_i 表示用户的反馈得分, 具体的数值是用户对排名前 10 的网页打分的均值, 用户的反馈得分使用的是 5 分制, 1 表示十分不满意, 2 表示不满意, 3 表示一般, 4 表示满意, 5 表示非常满意.

2) $P@N$ 指标:

$P@N$ 指标是对搜索引擎精度评估的一个重要标准, 它更加注重搜索排名的结果质量, 该指标的含义是在前 N 个搜索结果中与查询相关的结果的个数. 在前 N 个查询结果中, 有 N_1 个结果与查询词相关, 那么 $P@N$ 的值就为:

$$P@N=\frac{N_1}{N} \tag{7}$$

根据文献[9]的评价标准, 我们这里使用的是 $P@10$ 指标. 在搜索出的结果中, 要求用户指出每一项排序结果是否与检索关键词相关, 并使用投票的方式, 按照少数服从多数的原则来最终判定每一项排序结果是否与检索关键词相关.

4.3 实验步骤

步骤 1: 建立领域向量模型. 根据 3.2 节叙述的步骤, 首先利用网络爬虫 crawler4j 对新浪、腾讯等门户网站进行不同领域的爬取; 之后利用利用 Java 中的 Jsoup 库对网页根据标签提取内容文本, 并利用 Python 中的 jieba 中文分词库对提取的内容文本进行分词; 然后统计词频, 并根据停止词列表去除停止词; 再利用哈工大信息检索研究中心同义词词林扩展版加入近义词, 从而建成合理的领域向量模型. 每个领域为 100 维, 每一维是领域的一个特征. 针对学术、娱乐、体育三个领域, 我们分别构建了三个领域模型, 具体存储方式是 TXT 文件.

步骤 2: 网页爬取:

学术领域, 使用爬虫对合肥工业大学, 中国科技大学校园新闻网页以及科学网的新闻进行爬取. 总共爬取网页数据 11999 条, 其中合肥工业大学校园新闻网页数据 4076 条, 中国科技大学校园新闻网页数据 741 条, 科学网的新闻网页数据 7182 条;

娱乐领域, 使用爬虫对新浪网、凤凰网、腾讯网的娱乐版块进行爬取. 总共爬取网页数据 6294 条;

体育领域, 使用爬虫对新浪网、凤凰网、腾讯网的体育版块进行爬取. 总共爬取网页数据 5705 条.

步骤 3: 网页内容分析:

建立数据库, 设计数据库表, 对每一条网页数据进行分析, 并将对分析的结果存入表中, 表结构如表 1.

表 1 数据表结构

列名	数据类型	长度	注释
id	int	16	网页 id
content	longtext		网页内容
content_fenci	longtext		网页内容分词
title	text		网页标题
title_fenci	text		网页标题分词
url	varchar	128	网页 url
date	varchar	64	网页时间
source	varchar	128	网页来源
model_similarity	double		网页模型相似度

如果数据中有空值, 均置为空;

步骤 4: 领域向量模型赋权值: 为了之后计算网页的权重, 需要对领域向量模型进行赋权值, 当前的赋权方法是将领域向量模型中的所有的特征词的权值都赋为 1;

步骤 5: 网页内容赋权值: 同样, 为了计算网页的权重, 需要对网页内容进行赋权值. 赋权方法是建立与领域向量模型维度相同的网页信息向量, 考虑到标题中出现的词比正文中出现的词更加重要, 那么, 如果该词来自标题, 权重为 2, 如果该词来自内容, 权重为 1, 如果该词在标题和内容中都有, 权重为 3;

步骤 6: 网页与模型相似度计算: 虽然当前的网页数量不是很多, 但是考虑到今后会有扩充, 故将网页与模型相似度离线计算好. 利用公式(4)计算网页信息与领域向量模型之间的相似度, 计算的结果存放在数据库表的最后一列, 如果有计算出来的值为 NaN, 将其数据库表中的值指定为空;

步骤 7: 建立 Lucene 索引: 对于爬取的所有网页数据, 使用 Lucene 对网页标题, 网页内容以及网页 url 建立索引. 需要说明的一点是, Lucene 底层实现的是利用倒排索引和链接分析做的经典的网页排序算法. 根据文献[5,13]的实验步骤, 本文的算法对比实验也选择经典的网页排序算法. 具体的选择是基于 Lucene 的网页排序算法.

4.4 参数调优

在给出最终网页排序实验结果之前, 首先需要通过实验确定在计算最后得分的公式中参数 α 与 β 的最优值. 参数调优方法是找三名用户, 利用用户满意度公式, 对由不同的参数值计算出的排序结果进行打

分, 最后确定参数的最优值. 由公式(5)有 $\alpha+\beta=1$, 在参数调优的实验中, 我们做了九组实验, α 的值由 0.1 逐渐增加到 0.9, 增加的步长为 0.1, β 的值相应的减小, 参数调优的实验结果如图 2:

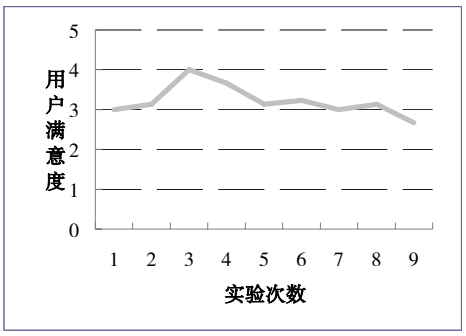


图 2　参数调整结果

通过图可以看出, 在 $\alpha=0.3,\beta=0.7$ 时用户满意度最高.

另外, 对于公式(6)中的 $score_3$, 通过实验我们发现, 同一领域网页的得分之间差距通常在 0.2 之内, 于是我们将 2014 年的网页 $score_3$ 值设置为 0.3, 2013 年的网页 $score_3$ 值设置为 0.1, 2012 的网页 $score_3$ 值设置为 0.05. 这样可以让最新的网页能够得到更加高的得分, 从而被排序到前面.

4.5 实验结果

步骤 1: 实验设置. 由于关于学术领域查询量相对较少, 学术领域由我们自定义关键词, 而用户对于娱乐领域和体育领域的近期发生的大事件会更加关心, 我们根据百度统计的 2014 年热词进行实验. 针对学术领域, 娱乐领域以及体育领域分别设置如下五个关键词:

① 学术领域: 北大, 计算机, 院士, 诺贝尔奖, 转基因;

② 娱乐领域: 高仓健, 锋菲恋, 冰桶挑战, 金马奖, 奶茶恋;

③ 体育领域: 德国国家足球队, 巴西世界杯, 欧洲冠军杯, 仁川亚运会, 砸砸.

利用 Lucene 检索出前 100 个查询结果, 利用一个列表存储排名前十的查询结果;

步骤 2: 对于有 Lucene 检索出的前 100 个查询结果, 我们利用在数据库中计算的网页内容和领域向量模型的相似度、网页的时间和网页标题, 对排序结果进行调整, 利用一个列表存储重排序后排名前十的查询结果;

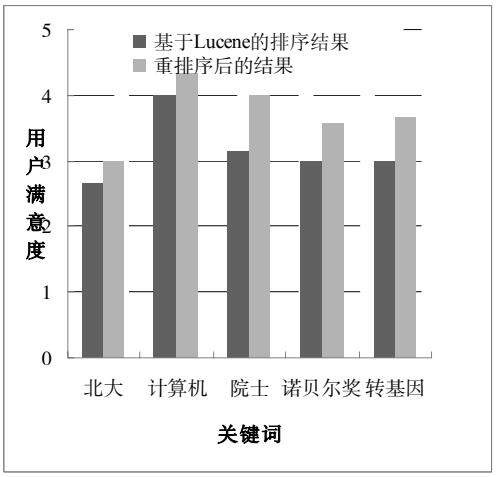
步骤 3: 制作一个简单的 JSP 页面, 将两个列表中的排序结果显示在网页中;

步骤 4: 找三名用户, 评分之前不告知用户两种排序分别属于哪种结果, 让用户根据自己对于搜索结果的满意程度对两种排序结果进行打分. 打分为 5 分制, 从 1 到 5 分别表示非常不满意、不满意、一般、满意、非常满意. 利用公式(7), 得到用户满意度的评分结果. 利用公式(8), 得到 $P@10$ 的结果. 用户满意度的评分结果如表 2.

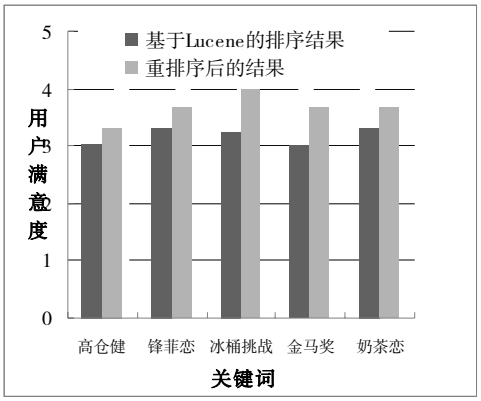
表 2　用户满意度打分结果

关键词	基于 Lucene	重排序后	评分提高(%)
北大	2.67	3.00	12.5
计算机	4.00	4.33	8.3
院士	3.13	4.00	27.7
诺贝尔奖	3.00	3.57	18.9
转基因	3.00	3.67	22.2
高仓健	3.03	3.33	9.8
锋菲恋	3.33	3.67	10.1
冰桶挑战	3.27	4.00	22.4
金马奖	3.00	3.67	22.2
奶茶恋	3.33	3.67	10.1
德国国家队	3.00	3.33	11.0
巴西世界杯	3.00	3.67	22.2
欧洲冠军杯	3.27	4.00	22.4
仁川亚运会	3.00	3.57	18.9
砸砸	3.33	4.03	21.1

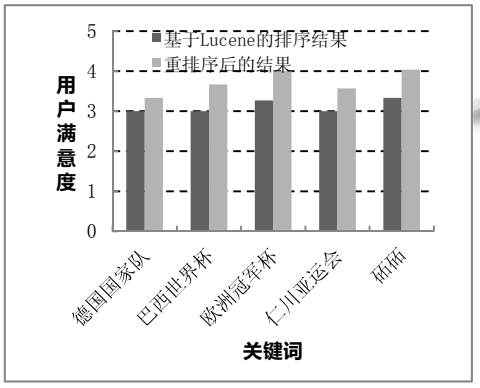
转换成图表形式后, 用户满意度的结果如图 3 所示, 图 3 中(a)表示学术领域的实验结果, (b)表示娱乐领域的实验结果, (c)表示体育领域的实验结果:



(a)学术领域实验结果



(b)娱乐领域实验结果



(c)体育领域实验结果

图 3 用户满意度实验结果

另一方面, 在这三个领域中, 对于所有 15 个关键词的 P@10 结果如表 3 所示.

表 3 结果

关键词	基于 Lucene	重排序后	评分提高(%)
北大	0.4	0.5	25.0
计算机	0.7	1.0	42.9
院士	0.3	0.7	130
诺贝尔奖	0.5	0.6	20.0
转基因	0.4	0.7	75.0
高仓健	0.6	0.6	0.0
锋菲恋	0.8	1.0	25.0
冰桶挑战	0.4	0.7	75.0
金马奖	0.6	0.9	50.0
奶茶恋	0.4	0.5	25.0
德国国家队	0.7	0.9	28.6
巴西世界杯	0.8	0.8	0.0
欧洲冠军杯	0.7	1.0	42.9
仁川亚运会	0.7	0.8	14.3
砸砸	0.4	0.7	75.0

将表中结果转换为图的表示形式之后, P@10 的实验结果如图 4 所示.

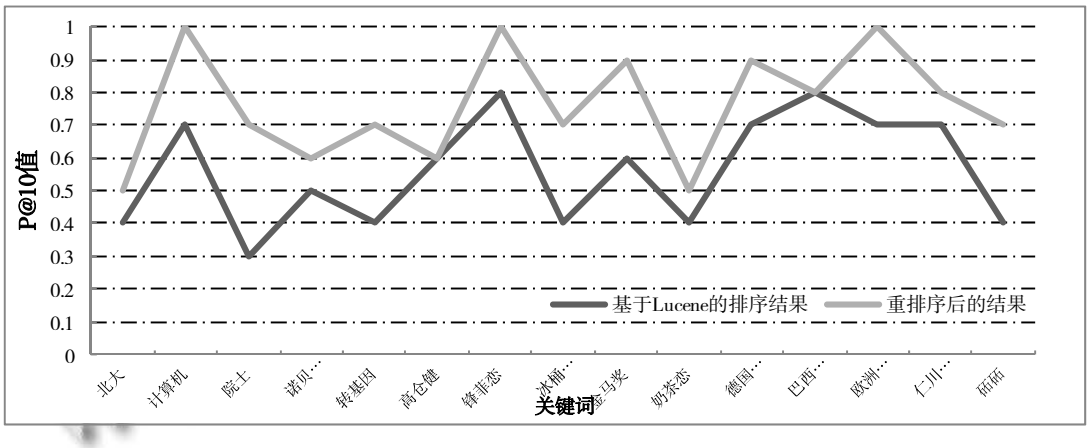


图 4 实验结果

4.5 结果分析

由实验的结果可以看出:

① 在学术领域, 重排序后用户满意度结果比基于 Lucene 的排序结果平均提高了 17.9%, P@10 结果平均提高了 58.6%;

② 在娱乐领域, 重排序后用户满意度结果比基于 Lucene 的排序结果平均提高了 14.9%, P@10 结果平均提高了 35%;

③ 在体育领域, 重排序后用户满意度结果比基于 Lucene 的排序结果平均提高了 19.1%, P@10 结果平均提高了 32.2%;

④ 在所有三个领域中, 重排序后用户满意度结果比基于 Lucene 的排序结果平均提高了 17.3%, P@10 结果平均提高了 41.9%.

总的来说, 本文提出的算法在不同评价指标上均比传统的网页排序算法有较大提高, 这表明, 通过在网页排序的过程中结合刻画领域特征的领域向量模型, 能够有效地提高网页排序结果的质量.

5 结语

本文研究了面向特定领域的网页重排序算法—TSRR 算法, 从一个角度来解决主题漂移问题. 利用向量模型对特定领域的特征进行刻画, 在网页排序的过程中结合领域向量模型, 对网页排序结果进行调整. 实验结果表明, TSRR 算法比传统网页排序算法在用户满意度和准确率上有较大提高. 本文将领域知识作为一种模型独立的抽象出来, 可以更加灵活的运用到其他的应用场景当中, 具有良好的可拓展性. 在后续的研究中, 将进一步基于特征的领域贡献度对特征赋权重, 以改进领域模型, 使网页排序算法达到更好的领域区分效果.

参考文献

1 Haveliwala TH. Topic-sensitive pagerank. Proc. of the 11th International Conference on World Wide Web. ACM, 2002: 517–526.

2 Brin S, Page L. The anatomy of a large-scale hypertextual Web search engine. Computer Networks and ISDN Systems, 1998, 30(1): 107–117.

3 Wu XD, Kumar V, Quinlan J R, Ghosh J, Yang Q, Motoda H, McLachlan GJ, Angus Ng, Liu B, Yu PS, Zhou ZH, Steinbach M, Hand DJ, Steinberg D. Top 10 algorithms in data mining. Knowledge and Information Systems, 2008, 14(1): 1–37.

4 Richardson M, Domingos P. The intelligent surfer: Probabilistic Combination of Link and Content Information in PageRank NIPS. 2001: 1441–1448.

5 黄德才,戚华春,钱能.基于主题相似度模型的 TS-PageRank 算法.小型微型计算机系统,2007,28(3):510-514.

6 张贤,周娅.基于 Lucene 网页排序算法的改进.计算机系统应用,2009,10:155-158.

7 刘菁菁,林鸿飞,赵晶.基于 PageRank 和锚文本的网页排序研究.计算机工程与应用,2007,43(10):170–173.

8 蒋建中,丁宝琼,吴琼,邱文武.基于页面分块的网页排序算法:BHITS.计算机工程,2010,36(11):64–69.

9 刘凯鹏,方滨兴.一种基于社会性标注的网页排序算法.计算机学报,2010,33(6):1014–1023.

10 龙文明,彭敦陆,姜兴隆.一种基于用户角色的综合网页排序算法.计算机工程,2011,37(7):53–55.

11 毕硕本,曾晓文,马燕.基于相似度的快速网页排序算法.科学技术与工程,2014,14(13):67–70.

12 王冲,曹姗姗.基于用户反馈与主题关联度的网页排序算法改进.计算机应用,2014,34(12):3502–3506.

13 闫泼,马军,陈竹敏.面向主题的网页排序算法研究.第三届全国信息检索与内容安全学术会议论文集.2007-11,江苏苏州.2007.521–527.

14 王晓宇,周傲英.万维网的链接结构分析及其应用综述. 软件学报,2003,14(10):1768–1780.

15 于楠,朱靖波,陈文亮.领域知识库的构建机制.第二届全国学生计算语言学研讨会论文集.北京,2004.

16 Glover EJ, Tsioutsiouliklis K, Lawrence S, Pennock DM, Flake GW. Using web structure for classifying and describing web pages. Proc. of the 11th international conference on World Wide Web. Honolulu, Hawaii. ACM Press. 2002.