HotRank:热度敏感的非结构化数据检索排名算法*

韩 晶, 宋美娜, 鄂海红, 宋俊德
(北京邮电大学 计算机学院, 北京 100876)

摘 要: 为满足用户对非结构化数据检索的需求,分析用户对数据的操作行为,提出一种新型的数据热度敏感的非结构化数据检索排名算法 HotRank。通过对数据操作情况(任务、访问次数、编辑时长等)进行日志记录,形成非结构化数据检索数据集。在此基础上,定义数据的任务相似度和数据热度计算方法实现该算法。结合实例仿真,对算法进行评估,并将仿真结果与其他算法进行比较,证明了该排名算法的准确率优于其他算法。

关键词: 非结构化数据; 检索; 排名; 热度
中图分类号: TP311.13 **文献标志码**: A **文章编号**: 1001-3695(2013)05-1306-03
doi:10.3969/j.issn.1001-3695.2013.05.006

HotRank:heat-sensitive ranking algorithm for unstructured data search

HAN Jing, SONG Mei-na, E Hai-hong, SONG Jun-de
(*School of Computer Science, Beijing University of Posts & Telecommunications, Beijing* 100876, *China*)

Abstract: To satisfy users' retrieval needs for unstructured data, by analyzing data operations of users, this paper proposed a novel heat-sensitive ranking algorithm for unstructured data search which was named HotRank. By logging the operators of data, such as task, access time, edit time etc., it created unstructured dataset. After that, it calculated similarity between task attribute of data and recent task, so as data heat, then established HotRank. Finally, it used unstructured data search to verify this algorithm and the result of simulation. Compared with the other well-known algorithms, the results indicate that this algorithm is better than several other algorithms in precision.

Key words: unstructured data; search; rank; heat

大数据时代,企业中积累了大量办公文档、PDF、视频等非结构化数据,这些数据来自于企业员工工作过程的多种操作,有的由员工自己创建,有的来自于邮件,有的则来自网络下载。想要在积累起来的大量非结构化数据中搜索需要的文件,需要经过多次搜索尝试,也需要花费相当长的时间。为加强非结构化数据检索效果,涌现出很多针对搜索排名方法的研究,有的研究通过为文件之间建立链接以提高检索排名效果,有的研究通过记录用户搜索历史来提高再次检索的效率,还有的研究通过让用户添加自己对目标数据的记忆来帮助提高检索效率。现有研究对检索结果的排名基本是在"数据是平等的"前提下进行的。实际上,对数据的任何操作总是处于用户的某个任务中,如果对数据的任务上下文进行标志,检索结果中将任务属性与用户检索行为的当前任务相似数据的排名提前,显然更能够让用户满意。此外,用户在完成一项工作或者任务的过程中,有一部分数据会被频繁操作(如一个项目中的项目需求文档),而另一部分数据仅操作过数次(可能是一篇来自网络的技术文章),也就是属于同一任务的不同数据其重要性不同。基于以上分析,对非结构化数据检索时,通过计算非结构化数据的任务相似度、在任务中的重要程度、关键字匹配程度等因素,从而达到优化检索排名的效果。本文通过对用户在工作过程的行为(检索行为、数据访问情况、任务等)进行记录,定义数据热度计算方法,提出基于数据热度的非结构化数据检索排名算法,并将该算法与 Windows 操作系统检索排名算法进行对比。实验结果显示,基于数据热度的非结构化数据检索排名算法效率优于 Windows 操作系统的检索排名算法。

1 相关研究

非结构化数据管理包括建立数据模型、数据存储、数据应用等多个方面,其中,与用户直接交互的是各类数据应用,而数据检索应用使用最为频繁。因此,设计高效率的数据检索排名算法在非结构化数据管理中尤为重要,同时也能更大程度地满足用户对数据的需求。

目前国内外对非结构化数据检索排名的研究主要集中在以下方面。文献[1]提出基于任务来定位数据资源,并建立原型系统 Haystack;文献[2]在个人信息检索领域较早提出将用户浏览情况和快速检索结合在一起,然而以上文献并未明确阐述任务与检索的关系。文献[3]通过挖掘文件的单次访问情况和访问频率,提出新建桌面资源链接的算法;文献[4]提出了以学习为基础的排名算法,其效率远优于基于文件基本属性的排名算法;文献[5]提出基于用户记忆的桌面搜索方法,在搜索时由用户完善对目标文件的记忆(如文件名、上次访问时间)的方式来提高检索排名效率;文献[6]提出基于用户活动

收稿日期: 2012-08-22; **修回日期**: 2012-10-15 **基金项目**: 国家科技支撑计划资助项目(2009BAH39B03);国家自然科学基金资助项目(61072060);国家"863"计划资助项目(2011AA100706);高等学校博士学科点专项科研基金资助项目(20110005120007);中央高校基本科研业务费专项资金资助项目(2012RC0205)

作者简介:韩晶(1984-),女,山西偏关人,博士研究生,主要研究方向为数据服务和数据管理(babyblue110128@hotmail.com);宋美娜(1974-),女,教授,博士,主要研究方向为服务科学与服务工程;鄂海红(1983-),女,讲师,博士,主要研究方向为服务科学和服务网络;宋俊德(1938-),男,教授,博导,博士,主要研究方向为并行计算、服务科学与服务工程.

分析的桌面资源定位,除自动抽取用户任务外,还通过挖掘桌面资源链接来支持模糊检索。由于面向桌面非结构化数据检索还没有通用的实验数据集,Kim 等人[7]创建了 pseudo-desktop collections 数据集,并在此基础上通过考虑对文档类型预测的影响因素,提出一种预测检索结果文档类型的方法[8]。可以看出,目前对非结构化数据检索排名的研究主要着眼于数据本身,并未考虑数据与用户行为的关系对检索排名的影响。同时,现有研究在数据重要性问题上涉及较少,个别文献虽提到了文件具有不同重要性,但未给出具体的算法[9]。本文在现有研究的基础上,进一步结合用户与数据交互过程中的上下文和数据操作情况,以期改善检索结果的排名。

此外,非结构化数据应用的基础是非结构化数据模型的构建,笔者在文献[10]中提出为非结构化数据标志任务、环境等属性构建星系数据模型,也为本文基于数据热度的检索排名算法奠定了研究基础。

2　基于数据热度的非结构化数据检索排名算法

2.1　算法和定义

人们在工作过程中常常遇到类似情景:正在做某项目 A 的验收答辩 PPT,此时想要参考项目执行过程中的其他 PPT,因此检索“项目 A,PPT”。结果通常是仅检索出文件名包含关键字“项目 A”的 PPT 类型文件,并且很可能当前正在做的 PPT 排名最为靠前,而真正希望检索到的大部分项目执行期间的 PPT 却并未被检索出来。究其原因,目前针对非结构化数据的检索:a)仅依靠关键字匹配,丢失关键字不完全匹配但却是用户需要的数据;b)默认按照修改日期排序,并不一定与当前用户需求不符。因此,为了改善非结构化数据检索中的这些问题,应该从新的角度进行考虑。

人类行为动力学研究表明,人的行为可以视做处理一系列的任务,且在一段时间集中完成某一项任务[11],可见用户的搜索行为与近期进行的任务非常相关。本文提出基于数据热度的非结构化数据检索排名算法 HotRank,通过计算检索结果数据的任务属性和用户当前任务的相似度,根据任务相似度、访问频率、访问时长等因素来计算热度分值,为检索结果排名。

HotRank 算法的基本思想是:用户提交查询后,使用关键字检索结果的同时得到初始检索结果;获取用户近期任务列表,将检索结果的任务属性与近期任务列表的任务属性进行相似度分值计算;然后计算访问次数分和编辑时长分,并为检索结果计算数据热度,调整检索结果的排序。描述算法之前,对算法中提到的术语进行定义:

定义 1　属性(attribute)。根据已提出的非结构化数据星系模型[7],非结构化数据 f_i 可以用属性来标志其特征。典型属性如表 1 所示。

表 1　典型数据属性表

名称	标志	描述
任务	f_i. taskAttr	文件所属任务
访问次数	f_i. accessAttr	文件访问次数
编辑时长	f_i. editTimeAttr	文件编辑时长
…	…	…

定义 2　数据热度。一个文件的数据热度分值表示该数据在所属任务中的重要程度,该分值由文件的访问次数、编辑时长、任务匹配度等综合计算。

2.2　算法描述

为了了解检索结果文件与用户近期进行的任务的相关度,需要对用户日志进行记录和分析,并计算出近期任务向量。首先对近期任务(recent task)算法进行描述。

```
algorithm 1:recent task
require:recent file list F
1 for all recentFile ∈ F do
2    if recentFile. taskAttr is not NULL then
3       add recentFile. taskAttr to recentTask
4    end if
5 end for
6 return recentTask
```

该算法通过记录用户近期访问过的文件集合 F,由 F 中文件的任务属性构建近期任务向量 recentTask = ($rtask_1$, $rtask_2$, …)。

在用户提交查询后,提取用户的查询关键字,记为 userQuery = ($keyw_1$, $keyw_2$, …),其中 $keyw_1$ 和 $keyw_2$ 代表查询关键字。将关键词向量 userQuery 提交给 Windows 系统的检索后,返回初始检索结果集 initF,每个结果文件均具有任务属性,可记为一个向量 fileTask = ($ftask_1$, $ftask_2$, …),其中 $ftask_1$ 和 $ftask_2$ 代表该文件具有的任务属性标记。根据任务分数 taskScore、访问次数分数 accessScore 和编辑时长分数 editTimeScore 综合计算数据热度分数,最后调整结果排序。HotRank 算法描述如下:

```
algorithm 2:HotRank
require:result file list R. recentTask
ensure:reranked R
1 for all resultFile ∈ R do
2    initScore←rank/count(R)
3    if task not NULL then
4       taskScore←sim(task,recentTask)
5    end if
6    simF←select resultFiles with same taskScore in R
7    accessSet←{access | resultFile ∈ simF}
8    editTimeSet←{editTime | resultFile ∈ simF}
9 end for
10 for all resultFile ∈ R do
11    accessScore←access/2 * max(accessSet)
12    editTimeScore←editTime/2 * max(editTimeSet)
13    heatScore←p * taskScore * (t1 + t2 * accessScore + t3 * editTimeScore) + q * initScore
14 end for
15 sort R with heatScore then create R′
16 return R′
```

具体地,设 sim(fileTask,recentTask)是文件 f_i 任务属性向量 fileTask 与近期任务向量 recentTask 的相似度,则

$$\text{taskScore} = \text{sim}(\text{fileTask},\text{recentTask}) \tag{1}$$

设文件 f_i 的访问次数为 a_i,$A = \{a_j \mid 0 < j < n$ 且 $f_i.\text{taskScore} = f_j.\text{taskScore}\}$,则

$$\text{accessScore} = a_i/2\max_A \tag{2}$$

设文件 f_i 的编辑时长为 et_i,$\text{ET} = \{et_j \mid 0 < j < n$ 且 $f_i.\text{taskScore} = f_j.\text{taskScore}\}$,则

$$\text{editTimeScore} = et_i/2\max_{\text{ET}} \tag{3}$$

则数据热度可计算为

$$\text{heat\_score} = p \times \text{taskScore} \times (t_1 + t_2 \times \text{accessScore} + t_3 \times \text{editTimeScore}) + q \times \text{initScore} \tag{4}$$

需要指出的是,由于访问次数和编辑时长是数值型,其最小最大值跨度大。为了减少过大过小属性值对排名分值的过度影响(例如由软件生成的日志文件访问次数非常大,但通常与用户任务关系较小),本算法采用聚类方式,首先将结果文件根据任务相关性分值分为三个等级,然后对任务级别相同的检索结果的 A 和 ET 取最大值,最终分别计算访问次数分值和编辑时长分值。另外,p、q、t_1、t_2、t_3 是各个分数在计算热度时的权重,经过多次实验表明,$p:q:t_1:t_2:t_3 = 95:5:0.9:0.07:$

0.03 时,实验效果最好。

3 实验过程及结果分析

3.1 实验过程和数据集

本文实验的基本思路是首先用 Windows Search 进行检索,得到原始检索结果;然后对检索结果使用 HotRank 算法进行排名,将结果与 Windows Search 的排名算法(本文用 WinRank 代称)进行对比。本文对四名不同领域(分别属于测绘、机械、计算机、电子)的志愿者在计算机上的日常操作进行了为期 30 天的监测,在日志中记录所有对文件的操作、检索词、检索结果及排序,并在监测结束后请志愿者协助描述每次检索的检索目的、标志检索结果相关度,并完善检索结果文件任务属性。需要说明的是,由于该类实验会记录用户的所有日常计算机操作,因此实验前都与志愿者签订了隐私保护协议。

首先选取 20 条检索,条件是每个检索的结果超过 20 个,构建检索词集 Q;然后选取每个检索的前 20 个检索结果,共 400 条数据,构建检索数据集 D。

3.2 实验结果及分析

判断结果集 D 中的每条结果与查询需求的相关性,根据其相关程度为每条结果打分,分值取自[0,1,2,3,4]集合;然后分别利用 11-point P-R、P@K 曲线对排序结果进行评估。结果如图 1、2 所示。从图中可以看出,HotRank 算法的准确率均比 Windows Search 有明显提高。这说明在 HotRank 算法中,使用任务、编辑时长、访问次数等因素来改进排序使其效果更优。

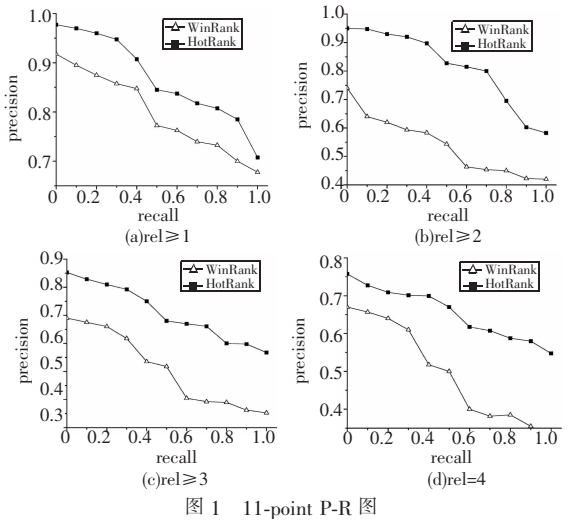

图 1 11-point P-R 图

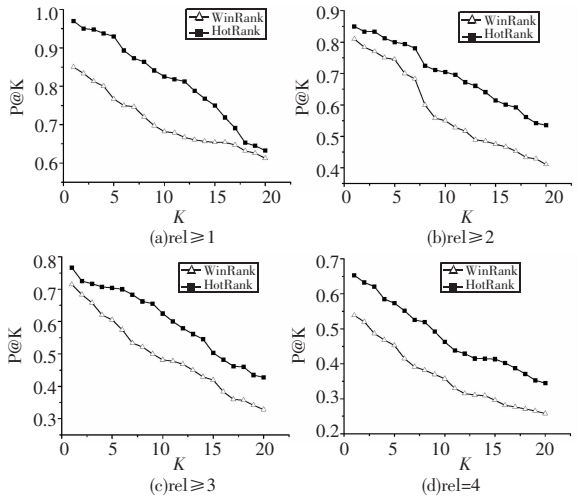

图 2 P@K 图

对比 Windows Search 和 HotRank,由于 Windows Search 仅考虑文件的关键字匹配情况和修改时间,使得其在计算排序时无法有效地将用户实际需要的数据检索出或者将其排名提前。而本文提出的 HotRank 则通过将数据与用户近期任务的相似程度、用户在该文件上所花费的编辑时间、用户对该文件的访问次数等因素纳入检索结果的排名分值计算中;同时,计算最终排名分值时对各个因素分数在最终分值中所占比例进行了合理的分配;此外,还将排序初始值融合到排序算法中,最终得到了较好的排序结果。

4 结束语

本文提出基于数据热度的非结构化数据检索排名算法 HotRank。该算法将数据所具有的任务属性、编辑时长和访问情况等因素纳入排名因素中,通过为各个因素赋予不同权重,综合计算最终排名分值,并对非结构化数据进行排序。经过实验对比分析,该算法的效率优于 Windows Search 排名算法。同时,由于 HotRank 算法充分结合数据的任务属性,对于已经积累起来的大量非结构化数据,在计算数据热度分值时需要人工补充任务属性,这也是本算法的局限性。因此,自动识别非结构化数据的任务上下文、对海量非结构化分析结果进行可视化以及架构高性能的非结构化数据管理框架等就成为下一步的研究方向。

参考文献:

[1] KARGER D R, BAKSHI K, HUYNH D, *et al*. Haystack: a customizable general-purpose information management tool for end users of semi-structured data[C]//Proc of the 2nd Biennial Conference on Innovative Data System Research. 2005.

[2] CUTRELL E, ROBBINS D, DUMAIS S, *et al*. Fast, flexible filtering with phlat[C]//Proc of Conference on Human Factors in Computing Systems. New York: ACM Press, 2006: 261-270.

[3] CHIRITA P P, NEJDL W. Analyzing user behavior to rank desktop items[C]//Proc of the 13th International Conference on String Processing and Information Retrieval. Berlin: Springer-Verlag, 2006: 86-97.

[4] COHEN S, DOMSHLAK C, ZWERDLING N. On ranking techniques for desktop search[J]. ACM Trans on Information System, 2008, 26(2): 1-24.

[5] CHEN Yi, KELLY L, JONES G J F. Memory support for desktop search[C]//Proc of SIGIR Workshop on Desktop Search. 2010.

[6] LI Yu-kun, ZHANG Xiang-yu, MENG Xiao-feng. Exploring desktop resources based on user activity analysis[C]//Proc of the 33rd International Conference on Research and Development in Information Retrieval. New York: ACM Press, 2010: 700.

[7] KIM J, CROFT W B. Retrieval experiments using pseudo-desktop collections[C]//Proc of the 18th Conference on Information and Knowledge Management. New York: ACM Press, 2009: 1297-1306.

[8] KIM J, CROFT W B. Ranking using multiple document types in desktop search[C]//Proc of the 33rd International Conference on Research and Development in Information Retrieval. New York: ACM Press, 2010: 50-57.

[9] JEWSEN C, LOWSDALE H, WYNN E, *et al*. The life and times of files and information: a study of desktop provenance[C]//Proc of Conference on Human Factors in Computing Systems. New York: ACM Press, 2010: 767-776.

[10] 韩晶,鄂海红,宋美娜,等. 基于主体行为的非结构化数据模型[J]. 计算机工程与设计,2013,33(3).

[11] GABRIELLI A, CALDARELLI G. Invasion percolation and critical transient in the Barabási model of human dynamics[J]. Physical Review Letters, 2007, 98(20): 208701.