一种元搜索主题偏好的排序算法

林 欣,温传林,韩立新
(河海大学 计算机与信息学院,江苏 南京 211100)

摘 要:元搜索引擎并行地向各个成员搜索引擎发出请求,合并及处理所有成员引擎的返回结果。相对于传统搜索引擎,元搜索引擎具有更好的查全率但在结果相关度排序及查准率方面仍需要改善。就相关度排序及查准率方面的问题元搜索成员引擎对于各个不同主题具有不同的检索质量并就此提出一种基于主题偏好的排序方法。利用 Beeferman 聚类方法对检索主题划分,通过 Borda 排序算法对元搜索引擎获得条目进行基于主题的分类排序,以此来提高元搜索查询质量和改善用户体验。

关键词:元搜索引擎;主题偏好;排序算法;聚类
中图分类号:TP301.6 **文献标识码**:A **文章编号**:1673-629X(2013)02-0041-03
doi:10.3969/j.issn.1673-629X.2013.02.010

A Ranking Algorithm Based on Topic Preference for Meta-search

LIN Xin, WEN Chuan-lin, HAN Li-xin
(College of Computer and Information, Hohai University, Nanjing 211100, China)

Abstract: Meta-search engine launches query simultaneously to its member search engines and shows a combined and ranked results list. Compared with the traditional search engine, meta-search engine has a better recall rate. However with the large amount of return items from its member engines, the precision rate and MMR still need to be perfected. Each member engine performs differently in the searching tasks with different topics in view of precision rate and MMR. In this paper, present a topic preference based ranking algorithm. Using Beeferman clustering method divides the search topic, with Borda ranking algorithm classify and rank the entries obtained by meta-search engine based on topic, improving the meta-search query quality and enhancing user experience.

Key words: meta-search engine; topic preference; ranking algorithm; clustering

0 引 言

在互联网飞速发展的今天,搜索引擎技术逐渐趋于成熟,但面对互联网中的海量数据,一般的搜索引擎检索到的结果仅仅是互联网中相关资源的一小部分,查全率较低。在这样的环境中,Andrew Berman 和 Erik Selberg 等人[1]提出并开发了第一个元搜索引擎。元搜索引擎向多个子搜索引擎同时发出请求,搜集各个搜索引擎返回的结果并加以后续处理,实现了一次请求多处检索的功能。由于多个搜索引擎返回结果数目众多,简单地合并[2]各个子搜索引擎返回的结果使得用户仍然需要花费大量的时间浏览查询所需要的信息。文中在研究已有结果合并排序算法的基础上,提出一种基于主题偏好的排序算法优化策略。

1 相关工作

1.1 传统元搜索分析

在传统的元搜索排序算法中,认为所有的成员搜索引擎都是相互平等的,即认为各个成员引擎的检索的质量都是相同的,但是事实上成员引擎对于不同主题的检索质量是不尽相同的。以“计算机”主题检索为例,在表 1 中列出各成员引擎前 20 个条目的相关度。对于此类主题的查询,Sogou 检索结果优于 Baidu 和 Yahoo。

表 1 “计算机”主题检索准确率表

查询词	Yahoo	Baidu	Sogou
元搜索分类	40%	55%	85%
Linux 多线程	45%	65%	65%
K 均值	55%	45%	55%
多进程编程	60%	75%	75%
	50%	60%	70%

因此猜想不同的成员引擎对于不同主题检索质量是不一样的,用户倾向于选择那些质量较好的成员引

收稿日期:2012-06-12;**修回日期**:2012-09-16
基金项目:国家自然科学基金资助项目(60673186);江苏省高校“青蓝工程”中青年学术带头人培养对象资助项目
作者简介:林 欣(1988-),男,江苏人,硕士研究生,主研领域为数据挖掘、信息检索;韩立新,教授,博士生导师,主研领域为信息检索、模式识别、数据挖掘。

擎返回的检索结果[3],基于这样的需求,文中提出了一种基于成员引擎主题偏好的排序算法。这里的偏好实际上就是指某个成员引擎对于某一主题检索的质量,它是一种在传统元搜索算法基础上的改进算法。

1.2 Borda 排序方法

Borda 排序法是元搜索引擎中比较常用的一种排序合成算法[4],它最初是由法国数学家 Jena-Charles de Borda 于 1770 年提出[5],由于 Borda 排序法的计算分值过程和元搜索引擎排序具有一定的相似性,因此这种方法广泛地应用于元搜索引擎排序,并且取得了良好的效果。Borda 排序算法首先对每个成员搜索结果根据位置关系给予分配一定的相关分值,位置越靠前其相关分值就越大,反之就越小。传统 Borda 排序方法在实际中的应用描述如下:

1)对于查询串 q,每个搜索引擎 $S_i(i=1,2,\cdots,n)$,查询到的第 k 个结果,均给予分配一个相关分值

S_i _sim[k] = S_i _number+1-k

2)定义四个数组 totalUrl[m]、totalTitle[m]、totalText[m]、totalSim[m],其中 $m=1,2,\cdots,\sum_{i=1}^{n} S_i$_Number。将 S_i _Url[k]、S_i _Title[k]、S_i _Text[k]、S_i _Sim[k]的值分别赋给这四个数组,这样所有的查询结果就只需要用四个数组来表示,为后面的排序带来了很大的方便。

3)对 totalUrl[m]依次进行比较,若出现相同的网址,则认为是同一个结果,前一个结果的相关分值为这两个相同结果的相关分值相加,并将后一个结果的相关分值重置为-1。

4)将相关分值大于 0 的结果提取出来,重新按照相关分值从大到小进行排序。排序时,网址、标题和摘要也应随其相关分值的变化而变化。

5)将排序后的结果输出。

2 基于主题偏好的元搜索排序方法

鉴于 1.1 节中不同成员引擎对于不同主题有着不同检索质量的猜想,建立如下检索流程模型:建立主题分类模型->分析用户查询意图->根据用户查询主题调用元搜索并排序->根据用户隐反馈优化成员引擎权重。

基于主题偏好的元搜索排序方法具体步骤如下:

步骤 1:用户输入查询串 $Q=(q_1,q_2,\cdots,q_n)$,查询串分词之后映射到相应的主题 k_i。即 $Q\rightarrow k_i$,表示属于主题 k_i 的查询。

步骤 2:每个成员引擎按照 Borda 排序方法给自己的检索结果分配相应的分值,即按照相对位置分配分值 SE_i _sim(k)= SE_iNum- k + 1。

步骤 3:按照成员引擎对于主题 k_i 的权重,对步骤 2 中的分值进行加权计算。即 SE_i _sim(k) = (SE_iNum- k + 1) * W_i^j。

步骤 4:去重并且合并相同结果,按照分值从高到低依次排序返回。

文中的主题偏好排序是基于 Borda 排序算法的一种改进方法。相较于 Borda 排序算法,其优点是将成员引擎对不同主题的检索质量纳入排序参考依据,更加合理地计算链接的相关权重。算法实现中的关键的问题有如下几点:

(1)为互联网主题的分类,为整个互联网建立一个分类模型并且能够随着互联网的发展定期更新[6]。

(2)分析用户查询意图,将用户查询串映射到主题中的某一个类别。

(3)计算及更新成员引擎对于主题的偏好的权重。

将在 2.1,2.2 和 2.3 节中对这些问题详细讨论。

2.1 主题分类模型

在开始检索之前,使用类似于 ODP(Open Directory Project)[7]的数据结构进行网页主题分类。如图 1 所示,根节点一般为模型入口,对于多用户的系统,root 节点为用户名。在模型中,第一层主题为主题大类,第二层为主题大类下的详细分类,叶子节点为主题对应的一个个特征词。对于更加庞大的系统,主题分类模型的层数可适当增加。这里需要指出的是,特征词的数量并不是一成不变的,会根据用户的查询不断地更新,更新方法将在后续章节中做详细描述。

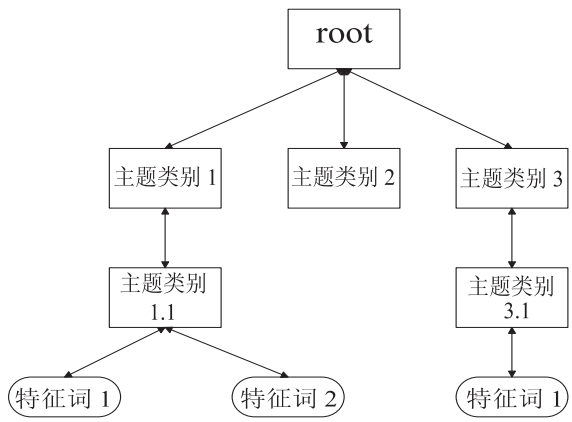

图 1 主题分类模型

在主题分类完成后,系统将查询映射为某一相关主题,然后依据主题偏好调用相对应的权重参数配置,对返回结果进行重排序。

2.2 用户查询意图分析

如上一章节所述,主题偏好排序算法是基于各类主题的,因而正确理解用户的查询意图,并将查询映射为相关主题是问题的关键。此处使用基于 Beeferman

聚类的方法对用户查询进行分析[8],并记录查询以便后续调整成员搜索引擎权重并更新系统。此聚类方法是由 Beeferman 等人[9]所提出,其中聚类相似度计算函数如公式 1 所示:

$$Dis(x,y)=\begin{cases}\dfrac{|N(x)\cap N(y)|}{|N(x)\cup N(y)|}, & |N(x)\cup N(y)|>0\\ 0, & \text{其他}\end{cases} \quad (1)$$

公式 1 中 $N(x)$ 表示所有与节点 x 相关联的节点,$|N(x)|$表示与节点 x 相关联的节点数目,$|N(x)\cap N(y)|$表示同时与节点 x 与节点 y 相连接的节点数目,$|N(x)\cup N(y)|$表示与节点 x 和节点 y 相连接的节点的数目的总和。

分析查询时 Beeferman 聚类的算法描述如下:

步骤 1:对每一个 $k_i\in K$ 的主题中的特征词 $WD(k_j)=\{wd_1,wd_2,\cdots,wd_n\}$ 分别进行检索并解析得到数据的 url 库 $\text{Url}k_j=\{\text{url}_1,\text{url}_2,\cdots,\text{url}_n\}$,$\text{Url}k_j$ 表示参考模型中主题 k_j 中的所有特征词使用成员引擎检索的结果网页链接的集合。

步骤 2:对步骤 1 中得到的 url 库进行去重处理。

步骤 3:新的查询词 nw_i 被当作一个特征词节点,将 k_j 主题中的所有特征词看成另外一个节点,把 url 库中的链接作为链接节点构建 Beeferman 图。

步骤 4:由公式 1 计算查询词 nw_i 与主题 k_j 的相关度(nw_i,k_j)。

步骤 5:依次重复步骤 2 ~ 4,计算所有主题与 nw_i 的相关度,返回相关度最大的二元组(nw_i,k_j)。

步骤 6:如果有新词没有处理完,则继续重复步骤 2 ~ 5。最终返回结果(nw_i,k_j)。

步骤 7:上面的过程处理完之后,即可将 $NW=\{nw_1,nw_2,\cdots,nw_n\}$ 分别添加到 ODP 模型与用户关注模型中。

返回结果(nw_i,k_j)表示查询项 nw_i 是一个关于主题 k_j 的查询。接下来调用关于 k_j 的权重配置对元搜索引擎返回的链接进行重排序。k_j 主题的权重计算会在下一章节中进行详细描述。

2.3　成员搜索引擎偏好计算

成员搜索引擎偏好权重的计算基于用户的点击隐反馈[10]。利用自适应方法,调整各个成员搜索引擎对于不同主题的偏好权重[11]。

成员搜索引擎主题偏好权重的计算方法详细说明如下:

步骤 1:将用户输入的查询串按照 2.2 节中的方法映射到相应的主题类别 k_j,表明这一次的查询是关于 k_j 主题的查询。

步骤 2:记录查询结果中用户对成员引擎 SE_i 提供链接的点击 Click (url_i^j),根据点击隐反馈计算相对应 W_i^j 的变化。重复这个过程直到这一次的检索结束。此处自增权重计算使用公式 2。

$$W_i^j=\frac{\text{click}(\text{url}_i^j)+1}{\sum_{i=1}^{n}\text{click}(\text{url}_i^j)+1} \quad (2)$$

步骤 3:将各个成员搜索引擎 SE_i 对于主题 k_j 的权重更新为 W_i^j。

计算获得的权重 W_{ij} 会对最终的结果归并排序产生影响,使擅长于相应主题的成员引擎返回的结果排名更加靠前。

2.4　结果归并算法

根据 Borda 排序算法,结合成员搜索引擎主题偏好权值得出链接的最终分值。

$$\text{Score}(\text{url}_j)=\sum_{i=1}^{n}(W_i^j*\text{Sim}(k)) \quad (3)$$

其中 url_j 表示查询主题 k_j 返回的链接的最终评分,n 为成员搜索引擎的数目,k 为链接在成员引擎 SE_i 返回结果中的相对位置,并按照最终的分数从高到低进行排序。这样的计算方式一方面可以让被多个搜索引擎同时检索到的链接排名靠前,另外一方面充分考虑了各个成员引擎对于不同主题的检索质量以及用户的点击反馈和关注兴趣等特点[12]。

3　结束语

文中重点介绍了元搜索的结果合成排序方法——一种基于主题的成员引擎偏好排序算法。提出该算法的依据是 Internet 上数以亿计的网页属于特定的主题,而每个成员引擎因其算法和实现上的不同,对于不同主题的检索质量是不一样的,该算法尽量让检索质量高的成员引擎的检索结果相对排在前面,文中对于成员引擎偏好的评价是来自用户查询点击的统计结果,通过用户的反馈获取各个成员引擎对于特定主题的权重。其中主要介绍了相关的主题模型以及用户主题模型的建立与更新并且提出了一种与内容无关的聚类算法——Beeferman 对新词聚类,可以将模型中未出现的特征词映射到相应的主题中。

参考文献:

[1] Selberg E, Etzioni O. Multi-service Search and Comparison Using the MetaCrawler[C]//Proceedings of the 4th International World Wide Web Conference. Boston, Massachusetts, USA: [s. n.], 1995: 195-208.

[2] Shreedhar M, Varghese G. Efficient fair queueing using deficit round-robin[J]. IEEE/ACM Trans. on Netw., 1996, 4(3): 375-385.

(下转第 48 页)

泛应用,而在这些系统中,通常需要与嵌入式系统板卡进行各种高精度定时数据采集、处理等操作。因此,需要在 VC 中方便地实现高精度定时器。

因此,文中对 VC 平台下各种定时器的设计进行了总结,并对其精度进行了详细测试。结果证实:通常的 VC 定时器方案定时误差均在数十毫秒以上,根本无法满足高精度定时要求。而如果对定时精度要求较高,则需要采用 CPU 自带的硬件寄存器完成定时功能,如文中 2.1、2.2 所示,此时其精度可达微秒级甚至更好。在某雷达信号处理机测试设备中,也正是采用 2.2 所示的时间戳定时器方案才达到了设计要求,并最终通过系统验收。

总体上,Windows 系统常规定时器使用简单方便,但定时时间越短误差越大,适合于定时时间长,精度要求低的场合。sleep、GetTickCount 和 timeGetTime 定时器精度与常规定时器性能相当。多媒体定时器可获得最高 1 ms 的定时精度,但需占用较多资源。基于系统计数定时器和 CPU 时间戳的定时器具有微秒级的定时精度,且具有很高的稳定性,相应地,其缺点是资源占用率高,比较适用于对定时精度要求高的程序场合,如实时仿真和数据采集等。而且,事实上以上各种方法并不仅局限于 VC 平台本身,因而在不同的软件开发平台下,各种定时方法均有其各自的优越性,具体应用时应根据实际的需要合理地进行选择,在定时器精度和系统开销之间取得较好的平衡。

参考文献:

[1] 卓红艳,赵 平. 基于 VC++的实时数据采集系统中定时器的使用与比较[J]. 现代电子技术,2007,18(2):129-132.

[2] Zhou Xuejun. Applied Mechanics and Materials[J]. Applied Mechanics and Materials,2011,93(9):1795-1800.

[3] Zhu Yongguang, Sun Zhengshun, Zhao Nanyuan. Video Capture Program Based on Visual C++ and the Application of Timer[J]. Computer Engineering and Applications, 2002, 20(12):128-132.

[4] Tang Hongzhong, Huang Huixian, Yin Lin. Application of VC++ DLL Timer in Design of Industrial Control Software[J]. Ordnance Industry Automation, 2003, 18(6):781-786.

[5] 郭占社,孟永钢,苏才钧. 基于 Windows 的精确定时技术及其在工程中的应用[J]. 哈尔滨工业大学学报,2005,18(12):38-41.

[6] 姚 晔,胡益雄. VC++应用程序精确定时方法的实现[J]. 计算机系统应用,2001,22(9):192-195.

[7] 何 斌,韦 工. 基于多媒体时钟的定时控制[J]. 舰船电子工程,2006,26(4):4-8.

[8] 洪锡军,李从心. Windows 下高精度定时的实现[J]. 计算机应用研究,2008,14(3):147-150.

[9] 何 斌,耿春萍. 多媒体时钟解决实时控制系统的定时[J]. 飞行器测控学报,2006,25(6):189-191.

[10] 李福华,李悦丽,段巧雄. Windows 环境下多串口控制软件设计中精确定时的实现[J]. 电子技术,2010,14(6):213-216.

[11] 刘春凤,田延岭. Windows 操作系统下的软件定时器的设计与应用[J]. 机电一体化,2004,10(5):39-42.

[12] 毕 业,史忠科. Windows2000 下高精度定时器设计与实现[J]. 工业仪表与自动化装置,2007(1):53-56.

[13] 韩志勇,李先国. Windows 操作系统下高精度计时研究[J]. 计算机工程与设计,2005,24(9):236-239.

(上接第 43 页)

[3] Page L, Brin S, Motwani R, et al. The PageRank Citation Ranking:Bringing Order to the Web[R]. Stanford:Stanford InfoLab. ,1999.

[4] 曹 林,韩立新,吴胜利. 元搜索引擎排序技术综述[J]. 计算机应用研究,2009,26(2):411-414.

[5] Regenwetter M, Tsetlin I. Approval voting and positional voting methods:inference, relationship, examples[J]. Social Choice and Welfare,2004,22(3):539-566.

[6] 盛宪锋,山 岚. 基于元搜索引擎的专业式智能网络信息检索系统[J]. 计算机工程与设计,2004(1):69-73.

[7] Weiss O S. Conceptual Clustering Using Lingo Algorithm:Evaluation on Open Directory Project Data[C]//Proc of IIPWM. [s. l.]:[s. n.],2004:369-377.

[8] 严莉莉,王倩倩,孟 杰,等. 基于聚类的个性化元搜索引擎设计[J]. 计算机技术与发展,2007,17(4):186-188.

[9] Bartell B T, Cottrell G W, Belew R K. Automatic Combination of Multiple Ranked Retrieval Systems[C]//Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. Dublin, Ireland:[s. n.],1994:173-181.

[10] 胡升泽. 个性化元搜索引擎若干关键技术研究[D]. 长沙:国防科学技术大学,2008.

[11] 孟 星. 基于 Agent 的自适应信息检索系统技术研究[D]. 西安:西安电子科技大学,2009.

[12] 王 忠,程 磊. 基于元搜索引擎的个性化 Web 信息采集[J]. 计算机工程与设计,2009(13):3117-3119.

一种元搜索主题偏好的排序算法

作者： 林欣， 温传林， 韩立新
作者单位： 河海大学 计算机与信息学院,江苏 南京211100
刊名： 计算机技术与发展
英文刊名： Computer Technology and Development
年，卷(期)： 2013(2)