基于 Shark－Search 和 Hits 算法的主题爬虫研究

罗林波[1]，陈 绮[1]，吴清秀[2]
（1.海南大学 信息科学技术学院，海南 海口 570228；
2.海南软件职业技术学院，海南 琼海 571400）

摘 要：主题爬虫是实现垂直搜索引擎的核心技术。介绍主题爬虫的两个重要爬行算法：基于网页内容评价的 Shark－Search 算法和基于网页链接关系的 Hits 算法，并分析了各自的优缺点，提出了一种新的主题爬行策略：将上述两种算法的优点结合起来即将基于网页内容评价和基于网页链接关系算法结合起来判断待下载 url 的优劣，并实现了一个主题爬虫。这种新策略正好弥补了两个算法各自的不足。通过与 Shark－Search 算法和 Hits 算法实现的主题爬虫对比，发现用新算法实现的主题爬虫查准率比这两种算法高。

关键词：主题爬虫；爬行策略；垂直搜索引擎
中图分号：TP393 **文献标识码**：A **文章编号**：1673－629X(2010)11－0076－04

Research on Topical Crawler of Shark－Search Algorithm and Hits Algorithm

LUO Lin-bo[1], CHEN Qi[1], WU Qing-xiu[2]
(1. College of Information Science and Technology, Hainan University, Haikou 570228, China;
2. Hainan Software Profession Institute, Qionghai 571400, China)

Abstract: Topical crawler is the core technology to achieve vertical search engine. There are two important crawling algorithms to be introduced: content－based evaluation of Shark－Search algorithm and link－based relationships Hits algorithms. It analyzed their respective advantages and disadvantages and proposed a new topical crawling strategy that is to combine the two algorithms which include content－based evaluation and link－based relationships, to judge whether url to be downloaded is good or bad, and implements a topical crawler. This new crawling strategy can make up for the deficiencies of the two algorithms. With the Shark－Search algorithm and the algorithm of the Hits contrast, it is inferred that the effect of using the new topical crawling algorithm which reaches the degree of accuracy is better than those two algorithms.

Key words: topical crawler; crawling strategy; vertical search engine

0 引 言

当前互联网正以惊人的速度不断发展，据 2010 年第二十五次中国互联网报告显示：2009 年我国网页总数已达 336 亿，较上一年增长 108%，73.3% 网民通过搜索引擎获取信息[1]。Web 信息的急速膨胀，在给人们提供丰富信息的同时，又使人们面临挑战，一方面网上的信息多种多样、丰富多彩，而另一方面用户通过传统搜索引擎来获取信息却越来越困难。因此，人们迫切需要一种更专业的搜索技术，将网上的信息更好地展现出来。于是垂直搜索引擎便诞生了，它被视为解决传统搜索引擎局限性的一种潜在方案，垂直搜索引擎已成为研究的热点[2]。

垂直搜索是面向特定主题（领域）的搜索引擎，是搜索引擎的细分和延伸，其特点就是“专、精、深”。主题爬虫（Topical Crawler）又称聚焦爬虫（Focused Crawler）[3]，是垂直搜索引擎中核心的部分，就是根据一定的网页内容和链接分析算法过滤与预定主题无关的链接，保留与主题相关的链接并将其放入待抓取的 URL 队列中；然后根据一定的策略从队列中选取下一步要抓取的网页 URL，并重复上述过程，直到满足系统的某一条件时停止。

主题爬虫以何种策略抓取 Web 信息，成为近年来主题爬虫研究的焦点之一[4]。

收稿日期：2010－03－09；**修回日期**：2010－06－12
基金项目：海南省自然科学基金资助项目(609003)；海南大学科研项目(hd09xm84)
作者简介：罗林波(1982－)，男，湖北黄冈人，硕士研究生，研究方向为数据挖掘；陈 绮，副教授，博士，硕士生导师，研究方向为数据挖掘。

1　主题爬行策略

目前常用的主题爬行策略主要分两大类[5]:一种是基于内容评价的爬行策略[6],以 De Bra、Herseovici 等人的研究 Fish－Search[7]及 Shark－Search[8]等算法为代表;另一种是基于 Web 链接评价的策略,以 PageRank[9]和 Hits[10]等算法为代表。

基于内容评价的爬行策略,主要是计算网页内容以及锚文本等与预定主题的相似度来评价待下载链接价值的高低,并依此决定其爬行策略,相似度的评价通常采用如下公式:

$$\mathrm{Sim}(\boldsymbol{d}_i,\boldsymbol{d}_j)=\frac{\sum_{k=1}^{n}(w_{ik}\times W_{jk})}{\sqrt{(\sum_{k=1}^{n}w_{ik}^2)(\sum_{k=1}^{n}W_{jk}^2)}} \tag{1}$$

其中 $\boldsymbol{d}_i$ 为新文本的特征向量,$\boldsymbol{d}_j$ 为第 j 类主题的中心向量,n 则为特征向量的维数,W_k 为向量的第 K 维。

基于 Web 链接评价的策略主要是依据网页之间的链接引用关系来判断网页之间的重要程度。目前的 Web 链接分析大多基于以下两个条件:

(1)从网页 A 指向网页 B 的超级链接是网页 A 作者对网页 B 的推荐;

(2)如果一条超链接将网页 A 和网页 B 相互连接起来,则网页 A 和网页 B 一般有共同的主题[11]。

下面分别介绍 Shark－Search 算法和 Hits 算法。

1.1　Shark－Search 算法

在 Fish－Search 算法的基础上,Hersovici 提出了 Shark－Search 算法。Shark－Search 算法对 Fish－Search 的一个重要改进就是利用所谓的"相似性引擎"对网页与主题的相关性进行模糊评分。子结点的主题相关性评分受 3 个因素影响:锚文本、锚文本附近的文字以及对父结点相关性的继承。Shark－Search 算法中对主题相关性的计算利用向量空间模型,取 0～1 之间的实数,URL 列表中的每一个 URL 的得分由(1)式计算。

$$\text{Potential\_score(child\_url)}=\gamma*\text{inherited(child\_url)}+(1-\gamma)*\text{neighborhood(child\_url)}\quad(\text{系数}\gamma<1) \tag{2}$$

当父结点与主题相关时,从父结点继承到的相关性评分 inherited (child_url) 由预定主题 q 和父结点网页的相似性计算得到,其中 current_url 为 child_url 结点的父节点,δ 为衰减因子且小于 1:

$$\text{inherited(child\_url)}=\begin{cases}\delta.\,\mathrm{sim}(q,\text{current\_url}),\ \text{if}(\mathrm{sim}(q,\text{current\_url}))>\varphi\\ \delta.\,\text{inherited(current\_url)},\ \text{otherwise}\end{cases} \tag{3}$$

邻近链接 neighborhood(child_url) 的评分与锚文本及锚文本附近的文字有关。根据锚文本,以及锚文本附近的文字与主题 q 的相似性 sim(q,anchor)和 sim(q,anchor_text)可以简单地计算出邻近链接的主题相关性得分:

$$\text{neighborhood(child\_url)}=\beta\cdot\mathrm{sim}(q,\text{anchor})+(1-\beta)\cdot\mathrm{sim}(q,\text{anchor\_text}) \tag{4}$$

1.2　Hits 算法

Hits 算法是由 Kleinberg 提出的基于超链接关系判断网页重要性的算法,目前主要用于搜索结果排序方面,引入了 Authority(权威)页面和 Hub(中心)页面两个重要的概念。通常好的 Hub 页面指向许多好的 Authority 页面;好的 Authority 页面总是被许多好的 Hub 页面所指向,这种 Hub 与 Authority 页面的相互加强关系,可用于 Authority 页面的发现,这就是 Hits 算法的基本思想。

Hits 首先根据查询的关键词确定一网络子图 G(V,E)(V 为网路子图的结点集,E 为边集),然后通过迭代计算得出每一个网页的权威值和中心值,具体步骤可分为三步:

(1)通过搜索引擎获得与主题最相关的 K 个网页(K =200)的集合,称之为 root 集。

(2)通过链接分析扩展 root 集,扩展后得到的集合称之为 base 集,扩展方法是对于 root 集中任一网页 p,加入所有 p 中所包含的链接到 root 集,加入最多 d(d =50)个指向 p 的链接到 base 集。

(3)计算 base 集中所有页面的中心值和权威值:若 G 中有 n 个结点,设 n 维向量 $\boldsymbol{a}$、$\boldsymbol{h}$,其中 $\boldsymbol{a}(i)$、$\boldsymbol{h}(i)$ 分别表示结点 i 的权威值和中心值。算法如下:用 1 初始化向量 $\boldsymbol{a}$、$\boldsymbol{h}$,$\boldsymbol{a}0=1$,$\boldsymbol{h}0=1$,然后进行 I,O 操作:

I 操作:$\boldsymbol{a}_i(v)=\sum_{(w,v)\in E}\boldsymbol{h}_{i-1}(w)$ (5)

O 操作:$\boldsymbol{h}_i(v)=\sum_{(v,w)\in E}\boldsymbol{a}_{i-1}(w)$ (6)

(4)规范化 $\boldsymbol{a}(u)$,$\boldsymbol{h}(v)$,$\boldsymbol{a}_i(v)=\dfrac{\boldsymbol{a}_i(v)}{\sqrt{\sum_{q\in n}[\boldsymbol{a}(q)]^2}}$,

$$\boldsymbol{h}_i(v)=\frac{\boldsymbol{h}(v)}{\sqrt{\sum_{q\in n}[\boldsymbol{h}(q)]^2}} \tag{7}$$

重复计算上面的 I、O 操作和规范化操作,直到 $\boldsymbol{a}(u)$,$\boldsymbol{h}(v)$ 收敛。

2　算法分析及改进

以 Fish－Search 为基础的爬行策略其优点是具有较好的理论基础,而且计算比较简单,但是这类方法忽略了链接结构信息,这类在距离相关页面集较近的地

方搜索时表现出良好的性能[12]；但由于页面中的文本信息缺乏“全局性”，很难反映 Web 的整体情况，使得这类网络蜘蛛普遍存在“近视”的缺点[13]。

Hits算法是一种依赖于查询(query - dependent)的主题提取算法。它首先利用搜索引擎从整个 Web 中选取与用户查询相关的部分网页来构成 Web 链接结构子图，然后在此链接结构子图上进行相应分析计算，由于 Web 链接结构具有自组织性，在互联网中具有相同或相关的主题内容的网页之间往往通过超链接相互连接形成一个个 Web 社区(communits)[14]，因此链接结构子图代表了互联网上某一主题的 Web 社区，当用户查询的主题较宽(甚至是多个主题)时，链接结构子图可能因多个子主题形成多个相对紧密的 Web 社区。因为 Hits 算法是一种基于迭代的算法，紧密链接区域中的页面的权值必定增加，从而影响了结果，这种现象通常被称为“主题漂移”(Topicdrift)[15]现象。

针对此，文中将 Shark - Search 算法与 Hits 算法结合，在计算待下载 url 的价值时除了依据网页内容、锚文本和锚文本附近的文字外还引入了依据 Hits 算法计算出的网页的权威值，既弥补了前者缺乏 Web 全局性之不足，又消除了后者容易产生“主题漂移”的现象。则计算待下载 url 值的公式变为：

$$\text{Potential\_score(child\_url)} = A * \text{inherited(child\_url)} + B \cdot \text{neighborhood(child\_url)} + C \cdot a_i(v) \quad (8)$$

其中系数 A、B、C 为正数且满足 $A + B + C = 1$，其他参数的意义同前。

如果将扩充后的所有连接都加入下载队列必导致下载队列过于臃肿从而影响爬虫的性能，故设定一个阈值 β，只有潜在价值大于 β 的链接才加入下载队列中。β 按如下公式计算得出：

$$\beta = \sum_{i=1}^{n} \text{url}_i / n \quad (9)$$

新算法描述如下：

(1)通过关键字匹配从搜索引擎获取前 k 个链接，并给链接赋初始值，然后将链接加入待下载队列 WateQueue 中。

(2)While(WateQueue 不为空){

While(保存的网页数量没有达到 m){

从 WateQueue 中取出得分最高的链接 current_url 下载该链接的网页 current_page 并计算出该网主题相关度值 inherited(child_url)；

If(inherited(child_url)>∂){

将 current_url 加入 G_nodes 中；

保存网页 current_page；

提取网页 current_page 中的每一条链接 child_url，按公式(4)计算每条链接的 neighborhood(child_url)，将边 current_url—>child_url 加入 G_Edges，将 child_url 作为节点加入 G_nodes；

}

通过搜索引擎将指向链接 current_url 的最多 d(d=50)个链接加入 G_nodes 中，并将相应的边加入 G_Edges；

}

If(下载网页数量达到 200){

按顺序重复执行上述公式(5)～(7)直到收敛，计算出每个连接的权威值 $a_i(v)$ 和中心值 h(i)；

}

利用公式(8)计算 G_nodes 中每个链接的得分，并利用公式(9)计算出 β；

将得分大于 β 的链接加入 WateQueue 中；

}

其中 G_nodes 为网络子图中的结点集，G_Edges 为边集，K 为首次通过搜索因为获取的连接数且必须 $K > m$，为了防止重链死链一般取 200 ～ 300 间，m 为 root 集中 url 数量，一般取 150 ～ 200 间，∂ 为主题判断的阈值。主题相关性判断采用的是向量空间模型方法。

新算法流程图如图 1 所示。

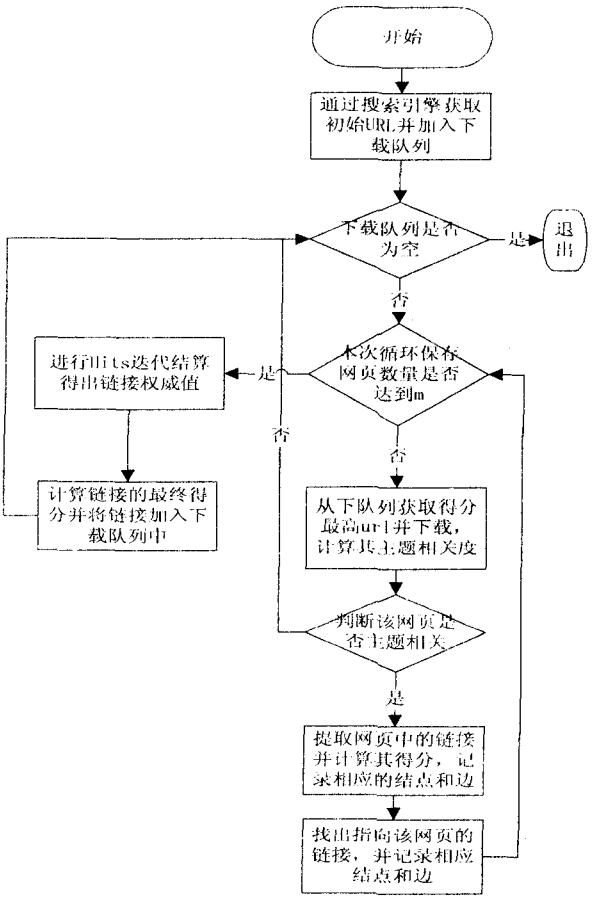

图 1 算法流程图

3　试验及分析

根据上述思想,设计了一个主题爬虫。评价聚焦爬虫系统性能指标主要有查准率、查全率[8]。这里主要计算爬虫系统抓取网页的查准率,图 2 是对比结果。

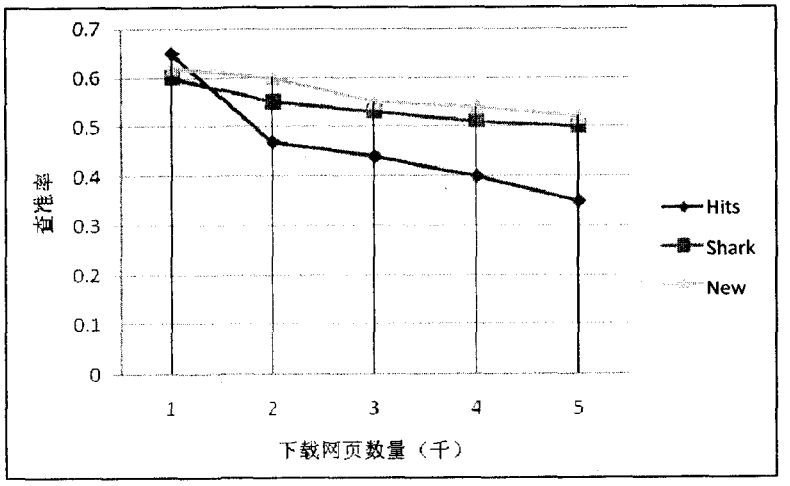

图 2　测试结果

上图表明,Hits 算法随着下载的网页数量的增加,查准率一直下降,因为随着抓取网页的增加"主题漂移"现象越来越重,Shark－Search 随着下载网页增加趋于稳定,但没考虑网页的全局性查准率不高,而新算法随着下载网页增多趋于稳定的同时也保持了较高的查准率,且效果明显好于 Hits 和 Shark－Search 算法。

4　结束语

文章在深入研究 Shark－Search 算法和 Hits 算法后,针对前者没有考虑链接关系缺乏 Web 全局性之不足和后者没考虑网页内容容易产生"主题漂移"的现象,提出了将两种算法相结合的思路即文本和链接相结合的爬行策略,结果表明新策略效果明显,新算法在提高查准率的同时也增加了算法的复杂性。如何在提高查准率的同时降低复杂度,将是下步研究的重点。

参考文献:

[1] CCNIC. 第 25 次中国互联网络发展状况统计报告[EB/OL]. 2010. http://www.cnnic.cn/uploadfiles/pdf/2010/1/15/101600.pdf. CCNIC.

[2] Panidis A, Poulos G K C, Pitas I. Combining Text and Link Analysis for Focused Crawling－an Application for Vertical Search Engines[J]. Information System, 2007, 32(6): 886－908.

[3] Menczer F, Pant G, Srinivasan P. Topical web crawlers: evaluating adaptive algorithms[J]. ACM Transactions on Internet Technology, 2004, 4(4): 378－419.

[4] Menczer F, Pant G. Evaluating Topic－Driven Web Crawlers[C]//Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New York, NY, USA: [s.n.], 2001: 9－12.

[5] 欧阳柳波,李学勇. 专业搜索引擎搜索策略综述[J]. 计算机工程, 2004(7): 32－33.

[6] 黄萱菁,吴立德,石崎洋之,等. 独立于语种的文本分类方法[J]. 中文信息学报, 2000, 14(6): 1－7.

[7] Bra D P, Post R. Searching for arbitrary information in the WWW: the fish－search for mosaic [C]//Second WWW Conference. Chicago: ACM Press, 1994: 45－51.

[8] Herseovici M, Jacov M, SMaarek Y. The Shark－Search Algorithm－An Application: Tailored Web Site Mapping[J]. Computer Networks and ISDN Systems, 1998, 30: 317－326.

[9] Page L, Brin S, Motwani R. The PageRank Citation Ranking: Bring Order to the Web[R]. Stanford, CA: Stanford University, 1998.

[10] Kleinberg J. Authoritative Sources in A Hyperlinked Environment[J]. Journal of the ACM, 1999, 46(5): 604－632.

[11] 康平波,田永鸿,黄铁军. 智能化网页资源收集工具的设计与实现[J]. 计算机工程, 2004, 30(4): 88－92.

[12] Menczer F. Complementing Search Engines with Online Web Mining Agents[J]. Decision Support Systems, 2003, 35(2): 195－212.

[13] Diligenti M, Coetzee F M, Lawrence S, et al. Focused crawling using context graphs[C]//Proceedings of the 26th International Conference on Very Large Databases (VLDB－2000). Cairo, Egypt: [s.n.], 2000.

[14] Flake G W, Lawrence S, Giles C L, et al. Self－Organization and Identification of Web Communities[J]. IEEE Computer, 2002, 35(3): 66－71.

[15] Menczer F, Pant G, Ruiz M E, et al. Evaluating topic－driven web crawlers[C]//Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New York, NY, USA: [s.n.], 2001: 241－249.

(上接第 75 页)

portunistic networks[C]//In: Proc. of the 2006 SIGCOMM Workshop on Challenged Networks. Pisa: ACM, 2006: 213－220.

[15] Crawdadproject [EB/OL]. 2008. http://crawdad.cs.dartmouth.edu/.

[16] UCSD wireless topology discovery project [EB/OL]. 2004. http://sysnet.ucsd.edu/wtd/.