Web 关键资源发现中的链接分析技术[1]

刘悦　王斌　杨志峰　张鑫

中国科学院计算技术研究所 软件研究室 北京 100080
E-mail: yliu@software.ict.ac.cn

摘 要：Web 关键资源发现是指在 Web 数据中发现与主题相关的关键资源(key resources)。研究表明，关键资源不仅与网页的内容有关，还与网页间的链接结构紧密相关。本文研究如何有效地利用链接分析算法来发现关键资源。在著名的 HITS 算法的基础上，本文给出了改进后的三个应用方案。在 TREC 的 WT10G 数据集合上进行的初步实验表明，改进的算法可以提高关键资源发现的准确性。
关键词：Web 关键资源发现，链接分析，HITS 算法，TREC，WT10G

Link Analysis in Web Key Resources Discovery

Liu Yue　Wang Bin　Yang Zhifeng　Zhang Xin

Software Division, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080
E-mail: yliu@software.ict.ac.cn

Abstract: Key resources discovery is to find the key resources about a given topic on the Web. The content and the linking structure of the resources are both important in this task. The paper focuses on how to use link analysis method to effectively find key resources. Based on HITS algorithm, three improved implementation methods are proposed. The preliminary experiments on the TREC WT10G show that our methods can improve the accuracy of key resource discovery.
Keywords: Web Key Resources Discovery, Link Analysis, HITS algorithm

1　问题提出的背景

随着互联网的信息组织趋向专业化以及 WEB 信息的爆炸性增长，如何从信息海洋中获取所需信息，已经成为 WEB 信息应用的关键问题，当前以 Google 搜索引擎为代表的通用性信息检索服务，大大提高了用户在 Internet 上获取信息的速度。

然而，在通用搜索引擎返回的众多结果中，并非所有的结果页面都是用户真正所需要的。为了返回更相关的结果，2002 年的 TREC (Text Retrieval Conference)会议中的 WEB 检索任务中定义了主题信息抽取(Topic Distillation)子任务。在这个子任务中，它要求返回的结果对于给定的查询(query)而言是一个关键资源(key resource)，而不是通常网页检索得到的结果页面。TREC 要求的关键资源主要包括两类页面：一类关键资源是和查询高度相关的网页；另一类关键资源包括的是这样一种情况：如果来自同一个站点的多个页面，都和主题高度相关，那

[1] 本文研究受国家 973 课题(G1998030413)及中科院计算所领域前沿青年基金(20016280-9)资助

么要将指向这些页面的那个页面作为关键资源提交给用户。例如，给定一个查询是“microsoft”，返回的应该是 www.microsoft.com/。而不应该是 www.microsoft.com/mscorp/ (是主页，而不是某一个具体的页面)；关键资源提取目标——即找到入口点的页面。

2 算法的渊源——相关研究

2.1 HITS 算法

链接分析的目的就是开发和利用网页之间的链接关系，挖掘深层隐藏信息，找到页面之间的关联关系，超链接表明的是页面之间的一种引用关系。Kleinberg 于 1997 年提出了基于 WWW 的链接分析算法 HITS(Hyperlink Induced Topic Search)。

在 HITS 的算法中，对某个主题，算法为某个网页集合中的每一个网页文档 p 计算两个权重值：authority 值和 hub 值，分别代表该文档作为该主题权威(Authority)或资源中心(Hub)的可靠性。在算法中，可以分别用 $A(p)$和 $H(p)$表示网页 p 的 authority 值和 hub 值，$A(p)$定义为所有指向 p 的页面 q 的中心权重 $H(q)$之和，$H(p)$定义为所有 p 所指向的页面 q 的权重 $A(q)$之和，这种迭代关系如下式所示：

$$A(p)=\sum_{q_i} H(q_i) \quad \text{(其中 } q_i \text{ 是所有链接到 } p \text{ 的页面)} \qquad \text{（公式 2.1）}$$

$$H(p)=\sum_{q_j} A(q_j) \quad \text{(其中 } q_j \text{ 是所有页面 } p \text{ 所链接到的页面)} \qquad \text{（公式 2.2）}$$

HITS 算法常常和已有的文本检索系统配合使用：假设某个文本检索系统(例如搜索引擎)收到查询请求后返回一个按照相关度排序的相关页面集合，HITS 算法取该集合中的前 r(比如 r=200)个页面作为算法的根页面集合(root set) R，然后将 Web 中指向这 r 个页面和从这 r 个页面指出的页面都扩展进来，得到算法迭代所需的封闭网页集合 R'，将 R'中的每个网页视为图中一个顶点，则这些页面之间的超链接就可看成图的边。

Kleinberg 已经从代数上证明了上述算法收敛，可以利用 hub 值和 authority 值对页面进行排序。HITS 算法虽然不能找出所有的相关页面，但是 hub 和 authority 之间是一种互增强关系，一个好的 hub 必然指向许多好的 authority，同样一个好的 authority 必然被许多好的 hub 链接。受到该算法的启发，我们发现可以把这种互增强的思想用到主题信息提取(Topic Distillation)的关键资源(key resource)寻找上面，而且在关键资源的寻找的算法中，还可以避免由于邻接矩阵规模太大而导致的计算效率低下的问题。为表述简单起见，这里我们把 HITS 算法称为基本算法。

2.2 对 HITS 算法已有的改进

围绕着 HITS 算法，很多研究机构进行了改进，改进的目的都是为了让权威分数和枢纽分数能够更客观地反映 web 的超链接属性，这其中作的比较好的是由 Bharat 和 Henzinger 领导的 DEC 的一个研究小组，他们指出了 HITS 算法的不足之处，并且提出了改进的算法。

Bharat 和 Henzinger 在研究中发现 HITS 算法在许多情况下得到的结果并不能让人满意，主要是由于下面 3 个原因：

1) 不同的主机之间的互增强关系。这种互增强关系主要表现为，有些时候同一个主机上面的许多页面可能会同时指向第二个主机上的同一个页面，这将会导致第一个主机上的页面的枢纽分数和第二个主机上的权威分数被抬高；反之亦然。由于我们假设每一个主机上的页面都是属于同一个作者或者组织，而前面所说的这种情形就无形中加大了一个作者在迭代计算中所起到的作用。

2) 自动产生的链接。在 WWW 上为了商业或者一些其他的目的，一些自动的链接生成工具往往被用来产生大量的超链接，，这就破坏了超链接本身的客观性，由这种自动产生的链接计算出来的权威和枢纽值肯定是不能反映实际情况的。

3) 无关结点。在有些情况下，HITS 算法的扩展后的根集合中包含许多与查询主题无关的页面，如果这样的页面在 web 子图中的链接稠密的话，迭代运算导致的直接结果就是主题漂移，使得一些权威分数和枢纽分数很高的页面是与查询无关的。

Bharat 和 Henzinger 在他们的研究中提出了一种能够有效控制主题漂移的方法。具体的实现策略是 1）将与查询主题无关的结点从 web 子图中去掉，不让其参加迭代运算。2）根据关联度修正不同的页面结点的权值即对于公式 2.1 和公式 2.2 作如下修正。

$$A(p)=\sum_{q_i} H(q_i)\times auth\_wt(q_i,p) \quad \text{(其中 } q_i \text{ 是所有链接到 } p \text{ 的页面)} \qquad \text{（公式 2.3）}$$

$$H(p)=\sum_{q_j} A(q_j)\times hub(p,q_j) \quad \text{(其中 } q_j \text{ 是所有页面 } p \text{ 所链接到的页面)} \qquad \text{（公式 2.4）}$$

修正后的算法与原算法比起来，效率有了很大的提高。而修正的目的就是有效地降低噪音页面和噪音链接的影响。我们也本着这个原则，提出了我们的改进方案，将 HITS 算法应用与我们的关键资源的提取中。

3 应用于关键资源提取的链接分析算法

主题信息的提取任务不仅要找到相关联的页面，而且还要从这些相关联的页面中找到真正的资源中心。这些资源中心(key resource)可能从内容上讲不是最好的(相关但不一定是排名最靠前)，但却是和查询最相关的关键资源的入口点。通过分析我们发现，如果从 HITS 算法角度来衡量的话，关键资源页面从它的定义上来看，应该是 hub 值较高的枢纽页面，因为它是指向与查询相关联的资源中心。如何将 HITS 的算法有效地应用于主题信息提取的关键资源(key resource)的查找呢？经过分析我们提出了如下三种可能的方案：

1） 在内容检索的基础上直接应用基本算法，根据 hub 值的高低来确定关键资源

2） 对内容检索的结果进行分类，再对于每一类利用基本算法。

3） 将结构信息与内容信息相结合

3.1 算法描述

对于第一种方案，我们是基于这样一种想法：基本算法对于根集合的扩展是在整个 web 页面（在我们的实验中就是 WT10G 数据集合或者是.GOV 的数据集合）上进行的，所以噪音页面就会比较多，为了解决这个问题，我们在基本算法的基础上，在根集合的扩展上进行了改进，具体的算法如下：

算法 1：key-resource-find1

(1) 对于某一个查询，取内容检索返回的前 m 个返回结果作为初始集合，记为 M；从 M 中取前 r ($r<m$)个结果构成根集合 R；

(2) 对于 R 中的每个顶点，可以根据超链接的关系按照如下规则在 M 中进行扩展：

规则 1：对于 $\forall p \in R$,如果 $\exists q,(q,p)$是超链接，且 $q \in M$，则将 q 扩展进根集合 R

规则 2: 对于 $\forall p \in R$,如果 $\exists q,(p,q)$是超链接，且 $q \in M$，则将 q 扩展进根集合 R

(3) 对于集合 R，用向量 H、A 分别记录其中所有网页的 hub 值和 authority 值。

(4) 利用基本算法计算 R 中每个元素的 hub 值和 authority 值。

(5) 取 hub 值前 k 名的页面放入集合 K 作为关键资源输出。

在改进的算法中对于原有的根集合的扩展算法上提出了我们自己的一些想法，一定程度上降低了扩展后的根集合中的噪音页面的数量，这样使得我们计算出来的权威和中心的分数就能够更加客观地反映页面的质量。使得我们能够利用他来更好地进行关键资源的寻找。

在分析了基于内容的检索结果之后，我们发现很多排名比较好的页面往往来自于同一类，而这些页面之间又有着稠密的链接关系(可以看成一类页面)。所以我们考虑在关键资源(key resource)提取的过程中，可以针对每一类页面来选取根集合，然后对每一类页面都利用 HITS 算法，输出 hub 值最高的那些页面作为主题信息提取的关键资源(key resource)。

算法 2：key-resource-find2

(1) 对于某一个查询，基于内容的检索结果取前 m 个返回结果作为初始集合，记为 M。将 M 按照页面性质分成 t 个集合 $C_1, C_2, C_3, \cdots\cdots, C_t$；

(2) 对于 $\forall C_i \in C_1, C_2, C_3, \cdots\cdots, C_t$ 中的每一类做如下计算

取 C_i 的前 r 个作为根集合 R_i 执行算法 key-resource-find1 在 M 中对 R_i 扩展，得到扩展集合 R_i'。对于扩展后的集合 R_i' 应用基本算法计算 hub 值，并取 hub 值最大的 k_i 页面加入集合 K_i

(3) 返回集合 $K=\bigcup_{i=1}^{t} K_i$，并将 K 中元素按 hub 值排序，输出前 k 个结果。

在方案 2 中我们的想法是这样的：我们希望在同一类（关于类别的划分我们在后面还要详细说明）页面中利用链接之间的这种互增强关系找到其中的关键资源。从上面的算法我们已经看到，在我们的改进算法中，对于根集合的扩展无论发生在那里都是在考虑了内容检索的基础之上的，也就从一定程度上降低了扩展后的根页面集合的噪音页面的数量。从关键资源的定义我们可以看到，实质上我们所找的关键资源其实是能够标引这一类资源的入口点页面，从内容上讲，它是与查询高度相关的，从结构上讲，它又是许多权威页面的入口点。所以在对基于内容的前 m 个页面进行分类的时候我们考虑采用了 2 种策略：

策略 1：为了叙述上的方便我们将这 m 个页面构成的 web 子图记为 $G=(V_m,E_m)$,其中表示 V_m 表示顶点页面的集合，E_m 表示这些页面之间的有向边的集合。我们在这个策略中分类的原则是将拓扑结构上连接紧密的页面按照[11]中给出的方法聚成一类，在计算的过程中我们只考虑链接的有无而没有考虑链接的方向，所以我们把 web 子图 G 构成忽略了方向的无向图对应的邻接矩阵 M 中的元素是按照如下方式定义的：

$$m_{ij}=\begin{cases}1 & if <i,j> or <j,i> \in E_m \\ 0 & otherwise\end{cases}$$

有了这样一个矩阵之后,我们利用聚类算法得到在拓扑结构上连接紧密的一个个的团，这里也就是将 M 分成 t 个集合 $C_1, C_2, C_3, \cdots\cdots, C_t$。

策略 2：这个分类的原则比较简单，我们按照 URL 的属性对 M 中的页面进行分类，来自与同一个站点的页面就分成同一类别，将 M 分成 t 个集合 $C_1, C_2, C_3, \cdots\cdots, C_t$。

方案 3 是结合结构和内容两个方面进行主题信息提取中的关键资源的寻找，具体的算法我们是在基本算法的基础之上作了如下的改进(前 3 步和基本算法一样，第(4)步开始实现上进行了改进)：

算法 3：

(4) while(当向量 H 和 A 都不收敛)

{对所有的 $p\in R'$ $A(p)=\sum_{p_i} H(p_i)*\alpha$ (其中 p_i 是链接到页面 p 的页面)

对所有的 $p\in R'$, $H(p)=\sum_{p_j} A(p_j)*\beta$ (其中 p_j 是页面 p 链接到的页面)

标准化向量 H 和 A.

}

(5) 取 hub 值前 k 名的页面放入集合 K 作为关键资源输出。

其中 α 是页面 p_i 通过内容检索之后得到的一个与查询相关程度的权值，β 是页面 p_j 通过内容检索之后得到的一个与查询相关程度的权值。α 和 β 可以通过训练得到，在我们的实验中，假定页面 p_i,p_j 基于内容的得分为 a,b 我们取 $\alpha = a/in\_degree(p)$，$\beta = b/out\_degree(p)$。最后在计算关键资源时，由于要考虑到内容的因素，对于扩展后根集合里的某个页面 p，我们按照如下公式来计算关键资源的重要程度：

$$score(p) = \frac{k_1 \bullet hub(p) + k_2 \bullet authority(p)}{rank(p)}$$

其中 k_1 和 k_2 是两个可以调节的参数(例如对于 query1，$k_1=0.79$，$k_2=0.21$)，$rank(p)$是页面 p 在内容检索结果中的排名。

3.2 算法分析

对于算法 key-resource-find1 由于在初始集合的选取上面，我们是从比较相关的页面中来进行根集合的扩展，通过指定相应的阈值，既控制了根集合的规模，又保证了扩展进根集合的页面从内容检索的角度讲是与主题相关的，在一定程度上控制了噪音页面的数量。与 HITS 的算法比较起来，降低了算法迭代的规模，减少了迭代次数。

算法 key-resource-find2 则是利用了主题信息在结构上显现出来的 Linkage Locality 特性（主题页面有较大可能链接同主题的页面）。对于来自于同一类页面找出其中的关键资源，由于对于内容检索之后的页面进行了分类，所以算法的迭代规模在最坏的情况下也只是算法 key-resource-find1 叠代规模的 $1/k$（因为我们在该算法中将页面分成了 k 类），对每一类页面我们通过它内部的链接结构计算出该类别中最重要的关键资源，从本质上讲，通过分类，我们就在通过结构信息提取关键资源的同时也考虑了内容的因素在里面。这样保证了提取出来的关键资源不会是重复的。

算法 3 是在算法 1 的基础上的一个改进，因为考虑到内容检索直接影响到检索的质量，所以该算法在基本算法的 hub 和 authority 值的计算过程中就考虑了内容的因素在里面，在计算关键资源的重要性时，考虑到 authority 的值越大，说明该页面是资源中心的可能性就越大，页面对于相关的查询来讲质量越好。所以将它作为反映页面是否关键资源的因素之一考虑到重要性的计算中。$rank(p)$的引入也是为了调节结构和内容在最后结果中所占的比例。

4 相关的实验结果和分析

我们所有的实验都是在我们自己的一个基于内容的检索系统上进行的，该检索系统采用了向量空间模型，它能够处理几十千兆的数据。数据集合我们采用了 TREC 的 Web Track Task 的 WT10G 数据集合，该集合数据规模为 10GB，共有 169 万网页。如何准确找到这些关键资源的入口点页面我们利用上述三个算法在 TREC WT10g 的数据集合上作了实验,以 TREC 中给定的 50 个查询为例，我们取定 $M=10000;R=200;$对于每一个查询，我们都用我们提出的三种改进的链接分析算法来做一下，和答案集合进行比较，计算出找到关键资源的比例，然后将 50 个查询的结果做平均（见下表 1）。对于只是内容检索分析一下实验的结果，我们是将我们基于内容的检索结果与答案集合比较计算出每一个查询找到关键资源的比例，然后将 50 个查询的结果作一个平均（见下表 1）。通过对照相关的页面分析，我们发现，指定了根集合的扩展范围之后，会过滤掉一些无关的页面，算法 3 中返回的结果比较好。算法 2 中由于分

类的缘故，有一些子类对应的子图是孤立点或者是平凡子图，这样的子类应用结构分析算法对于关键资源的抽取没有任何意义，所以导致分出来的子类和真正有用的关键资源在数量上相差比较大；但在相对稠密的子图中我们发现抽取出来的关键资源页面还是比较准确的。算法 1 返回的结果质量介于算法 2 和 3 之间，由此可见内容检索还是主题信息抽取的基础，结构化信息对于链接稠密的子图能够显示出它的优势。表 1 给出了对于所有查询三种算法和只考虑内容检索的结果比较。

	只考虑内容检索的结果，不考虑链接	算法 1	算法 2	算法 3
扩展前作为基准的页面数量	10000	10000	10000	10000
扩展后根集合中平均页面数量	10000	1247	4107	1176
提取出的可能作为入口点的平均页面数目（前 1000 个页面中）	23	27	34	39

表 1：三种考虑了链接的算法的结果与不考虑链接的算法的对照表。

5 结论

本文在对于 TREC 的子任务 Topic Distillation 完成的过程中尝试着利用链接分析的算法用于关键资源的查找。以 HITS 的算法为基础，给出了三种改进算法，并对算法作了相应的分析，在 WT10g 的数据集合上进行的实验表明:与没有采用结构信息的结果比较，结构化信息对于那些链接稠密的子图关键资源的寻找还是十分有效的。

在今后的研究中，我们希望能利用有效的分类模型，在第一遍内容检索之后的结果上，分离出稠密子图；以提高关键资源的抽取率，另外一方面，对于利用结构化信息计算出来的结果，如何利用预测模型参数的自适应调整方法来进行改进。

参 考 文 献

[1] Krshna Bharat and Monika Henzinger. "Improved algorithms for topic distillation in a hyper-linked environment." In 21st SIGIR Conference on Research and Development in Information Retrieval, 1998.

[2] Sergey Brin and Lawrenece Page. "The anatomy of a large-scale hypertextual web search engine." The Seventh International World Wide Web Conference, April 1998.

[3] J. Kleinberg."Authoritative sources in a hyperlinked environment. In Proceedings of the ACM-SIAM Symposium n Discrete Algorithms, 1998

[4]David Gibson, Jon HITS "Inferring web communities from link topology" In Proc. 9th ACM conference on hypertext and hypermedia 1998

[5] P. Indyk S. Chakrabarti, B. Dom. "Enhanced hypertext categorization using hyperlinks." In ACM SIGMOD 1998.

[6]B.Yuwono and D.L.Lee "Search and ranking algorithms for locating resources on World Wide Web." In proc. of the Int Conference on Data Engineering(ICDE),pages164-171, New Orleans,USA,1996

[7]Ramesh R.Sarukkai "link prediction and path analysis using Markov chains." In proc.The International World Wide Web Conference 2000.

[8]Ricardo Baeza-Yates Berthier Ribeiro-Neto "Modern Information Retrieval" ACM Press 1999.

[9]TREC-2002 Web Track Guidelines

[10]Soumen Chakrabarti,Martin van den Berg,Byron Dom,Focused Crawing:A New Approach to Topic-Specific Web Resource Discovery.Proceeding of the 8th World Wide Web Conference,May 1999,Toronto,Canada

[11] 中国科学院计算技术研究所博士学位研究生学位论文"聚类/分类理论研究及其在文本挖掘中的应用" 卜东波 2000 年 10 月