Mining Web Site's Clusters from Link Topology and Site Hierarchy

Kwok-Wai Cheung and Yuxiang Sun

Department of Computer Science
Hong Kong Baptist University
Kowloon Tong, Hong Kong

COMP-03-016

Release Date: July 11, 2003

Abstract

Foraging information in large and complex web sites simply using keyword search usually results in unpleasant experience due to the relatively low precision of existing search engines. To support more precise information search, some abstractions of the web sites, like sitemaps, are needed. However, their creation normally requires manual effort. In this paper, we extend the HITS algorithm and integrate hyperlink topology and web site hierarchy to identify a hierarchy of web page clusters as web site abstractions. As the algorithm is based on HITS, each cluster follows the bipartite graph structure and is computed with an authority and hub pair as the cluster summary. The effectiveness of the algorithm has been evaluated using three different web sites (containing ~ 6000-14000 web pages) with promising results. Detailed interpretation of the experimental results as well as qualitative comparison with other related works are also included.

Mining Web Site's Clusters from Link Topology and Site Hierarchy

Kwok-Wai Cheung and Yuxiang Sun
Department of Computer Science
Hong Kong Baptist University
Kowloon Tong, Hong Kong
{william, xiangzi}@comp.hkbu.edu.hk

Abstract

Foraging information in large and complex web sites simply using keyword search usually results in unpleasant experience due to the relatively low precision of existing search engines. To support more precise information search, some abstractions of the web sites, like sitemaps, are needed. However, their creation normally requires manual effort. In this paper, we extend the HITS algorithm and integrate hyperlink topology and web site hierarchy to identify a hierarchy of web page clusters as web site abstractions. As the algorithm is based on HITS, each cluster follows the bipartite graph structure and is computed with an authority and hub pair as the cluster summary. The effectiveness of the algorithm has been evaluated using three different web sites (containing ∼6000-14000 web pages) with promising results. Detailed interpretation of the experimental results as well as qualitative comparison with other related works are also included.

Keywords: Cluster mining, web site analysis, HITS algorithm, information search and retrieval

1. Introduction

Foraging information in a web site can be a time-consuming task, especially when the web site is both large and complex. To support information search in a web site, we need either a search engine or some sort of web site abstraction, like a sitemap. The existing search engines are mostly keyword-based and it is a common experience that the results they can return are far more than expected. For sitemaps, its quality depends very much on the web master, and yet has to be regularly updated due to the fast changing web contents. Therefore, it will be useful if a sitemap truely refecting the contents of the web site can be generated automatically.

A web site normally contains a collection of on-line documents interconnected by hyperlinks which are explicitly created by different authors for showing logical relationships between documents. Various algorithms have been proposed in the literature to analyze hyperlink structures for useful applications, like computing authorities and hubs for topic distillation [8], estimating document associations [3], identifying web communities [6], etc. With a similar motivation, the random graph approach has recently drawn a lot of researchers' attention for web modelling [10] where the use of power law has shown to be effective in characterizing the web connectivity. While most of the link analysis algorithms found in the literature attempt to identify emerging communities in the global Web, the connectivity structure of web pages within a *web site* can also be used for identifying logically coherent clusters. Besides, the directory hierarchy of a web site, in most cases, is carefully created by the web master and thus provides useful information about the relationship of web pages within a site [11]. It is obvious that effectively integrating the two sources of information — link topology and site hierarchy can result in effective web site mining algorithms.

Table 1 shows various research problems related to web site mining grouped under three commonly adopted categories of web mining [9], with the possible applications and representative references included. The focus of this paper is under the category of web structure mining and we focus on *web subgraph identification*.[1] Here, we define web subgraph identification as the task of identifying *clusters* of web pages of a certain subgraph structure within a web site. In the following, we will simply called it *web cluster identification* to highlight its intuitive meaning. As we do not assume the union of the clusters to form the whole web site, the approach is also called *cluster mining* [14].

[1] It should be noted that there are many applications where people find combining techniques under different categories effective in performance boosting. E.g., web page categorization can be enhanced by using snippets [7]. Also, topic distillation can be improved by weighting hyperlinks with some content similarity measures [2].

Web content mining	**Mining the web pages in web sites:** - Web page categorization [1] (support web site catalog generation) - Web site content comparison [12] (understand competitors' web sites)
Web usage mining	**Mining the web log of the web site:** - User access sequence clustering [16] (user profiling) - User access sequence prediction [13] (personalized recommendation/prefetching) - User access sequence mining [14] (support sitemap generation)
Web structure mining	**Mining the web site connectivity:** - **Web subgraph identification** [3] (support sitemap generation) - Web site classification [5] (support global web catalog generation)

Table 1. Research problems related to web site mining.

Among different subgraph structures, densely connected bipartite graphs are mostly interested in hyperlink analysis as a good representation of communities emerged in the Web. In this paper, we made the conjecture that bipartite graphs also form good approximates of logically coherent clusters in web sites and proposed to extend a link analysis algorithm called the HITS algorithm for mining the clusters [8].

The HITS algorithm is an iterative algorithm which was originally proposed to identify authorities and hubs of bipartite subgraphs in web search results. To the best of our knowledge, it has not been applied to analyze web sites. In this paper, we first investigate the intrinsic limitation of the HITS algorithm in web cluster identification. Then, we extend to an algorithm called ReHITS which integrates repeated applications of HITS with the path information provided by the web site hierarchy to identify logical coherent clusters in a hierchical manner. To demonstrate its effectiveness, we have applied the proposed algorithm to three real web sites and evaluted in detail the identified clusters.

In the literature, there exist some related works in computing representations of web sites, though their formulations are quite different from ours. Perkowitz *et al.* [14] applied statistical clustering as well as conceptual learning algorithms to user access log and web page contents to identify conceptual clusters in a web site. Besides, Li and Candan have proposed to first make use of some carefully designed rules to characterize the entry points of a web site's *logical* domains (or partitions), and the web pages are merged according to the hyperlink structure to form the logical domains [11]. Then, graphs with key web pages describing document associations are computed as the domain summaries [3].

The rest of this paper is organized as follows. Section 2.2 describes the characteristics of the web site summarization problem. Section 2.1 gives the background of the HITS algorithm and discusses its limitations for analyzing web sites. Then, we propose to model a web site using a set of bipartite subgraphs and describe our proposed algorithm for summarizing web sites in Section 3. Experimental results are presented and discussed in Section 4. Section 5 concludes our contributions and lists out possible future extensions.

2. HITS & Web Cluster Identification

2.1. The HITS algorithm

One of link anlysis algorithms have been proposed where The HITS algorithm is one of the link analysis algorithms which was proposed to identify *authorities* and *hubs* in web search results [8]. Authorities are defined as the web pages which are referred or cited many times by hubs, while hubs are the web pages which contain many links to authorities. These co-definitions of authorities and hubs imply that the two sets of web pages should form a structure similar to a bipartite graph, i.e., majorities of the directed links are going from hubs to authorities.[2]

Let $\mathbf{G} = \{\mathbf{V}, \mathbf{E}\}$ denote the graph representing the hyperlink connectivity structure of a web site, where $\mathbf{V}$ is the vertex set containing all the web pages in the site as its elements, and $\mathbf{E} := \{(\mathbf{V}_s, \mathbf{V}_d)\}$ is the edge set containing ordered pairs of vertices, each formed by recording the source $\mathbf{V}_s$ and the destination $\mathbf{V}_d$ of all the hyperlinks. Also, let $\mathbf{A}_i$ and $\mathbf{H}_i$ denote the authority score and the hub score of the vertex $\mathbf{V}_i$.

The HITS algorithm contains two main steps:

Step 1: Computing Authority Scores

$$\mathbf{A}_i = \sum_{(\mathbf{V}_i, \mathbf{V}_j) \in \mathbf{E}} \mathbf{H}_j$$

$$\mathbf{A}_i = \frac{\mathbf{A}_i}{\sum_{\mathbf{V}_j \in \mathbf{V}} |\mathbf{A}_j|^2}$$

Step 2: Computing Hub Scores:

$$\mathbf{H}_i = \sum_{(\mathbf{V}_i, \mathbf{V}_j) \in \mathbf{E}} \mathbf{A}_j$$

$$\mathbf{H}_i = \frac{\mathbf{H}_i}{\sum_{\mathbf{V}_j \in \mathbf{V}} |\mathbf{H}_j|^2}$$

[2] Note that the bipartite graph assumption is only a sufficient but not necessary condition for defining authorities and hubs.

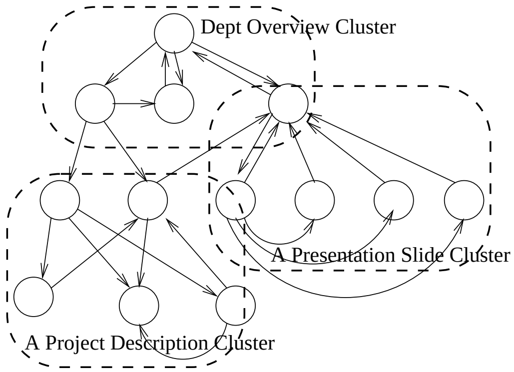

Figure 1. A website with several dense bipartite subgraphs as cluster candidates.

The algorithm iterates Step 1 and 2 until the authority scores and the hub scores converge. The vertices with high enough values of $\mathbf{A}_i$ and $\mathbf{H}_i$ are computed as the authorities and hubs respectively. It has been shown that the set of converged values forms the principle eigenvector of an adjacency related matrix. The resulting set of web pages with high authority and hub scores corresponds to the bipartite subgraph (or the cluster) of the web site with the highest degree of inter-connectivity.

2.2. Web cluster identification

As mentioned in [3], although it seems natural to apply the HITS algorithm to analyze and summarize Web sites, the applicability is limited. The main reason is that the HITS algorithm assumes the existence of only one densely connected bipartite subgraph in the web site where you can find all the information. This scenario, however, can seldom be found in real web sites. For example, it is common for a computer science department's web site to contain a large numbers of programming or systems documentation archives (e.g., java documentation, unix commands' glossary, etc.). At the same time, the relatively smaller but important information like staff or projects also exist. Direct application of the HITS algorithm to such a web site will result in all the computed authorities and hubs biased to and only to the most densely connected information clusters, which are mostly document archives like, for example, a mixture of HOW-TO documents. This is good as some documentation archives can be identified. But obviously, the web site visitors expect also the other types of information in order to have an overview of the website. See Figure 1 for a pictorial illustration.

In [8], the issue about multiple sets of authorities and hubs in web search results has been pointed out, with the suggested causes mainly related to the nature of the query string (e.g. its polysemy, its references to multiple communities). For multiple clusters in a web site, we note that due to the differences in link semantics and page context, they have their specific characteristics as compared to communities emerged in the Web.

Link semantics in web sites

Web search results mostly contain cross-domain links. The main objective for an author to create links crossing the domain boundaries is to cite logically related Web pages. However, for links within a web site (intra-domain), the objectives are far more diverse, from providing citations of related topics (e.g., a tutorial page pointing to many important references) to assisting web site navigation (e.g., links to next pages, previous pages and home pages). So, it is much more common for web pages within a web site to form a large number of densely connected subgraphs. Also, the underlying "clustering" structure, e.g., the number of clusters, their sizes and their degrees of inter-connectivity, depends very much on the web site designers. In order words, the degree of multiple cluster problem is intensified in web cluster identification.

Page context in web sites

Due to the cross-domain requirement, we do expect the application of HITS to Web search results to be sensitive to the existence of polysemy, resulting in the well-known *Topic Drift Problem*. On the contrary, web pages in a web site have more or less the same context. For example, a university web site can contain course information, student information, job vacancies, mission statements, press releases, etc., but all are under the university context. So, we believe that applying HITS to summarize web sites is less affected by topic drift and related techniques may not be relevant.

To summarize, the main causes of multiple clusters within a web sites include

- existence of self-referenced and indexed hypertext document archives (e.g., Java SDK documentation, presentation slides),
- existence of self-referenced topics' description (e.g., project/people home pages),
- links showing the inter-relationship between those self-contained archives and topics (e.g., the navigation aids near the root of the Web site).

Note that the first two causes should result in clusters found in deeper levels of Web sites, but normally with a higher degree of inter-connectivity. The last one should form clusters at a higher level, where people used to reveal in the sitemap. This hints a notion of cluster hierarchy and motivates the recursive algorithm we are going to propose.

3. ReHITS - A Recursive HITS Algorithm

In [8], Kleinberg has studied the possibility of extracting not only the principle eigenvector of an adjacency related matrix but also the non-principle ones as logical clusters. However, to the best of our knowledge, we have not seen any work that apply or analyze this approach intensively. Also, a web site normally contains thousands to ten thousands of web pages. The size of the adjacency related matrix could be too large for eigenvalue decomposition.

3.1. Repeated applications of HITS

In this section, we demonstrate how the HITS algorithm can be extended to extract a hierarchy of clusters from a web site, with each cluster summarized by a set of corresponding authorities and hubs. In order to relax the single bipartite graph assumption, we first apply the HITS algorithm to the whole web site. The authorities and hubs of the densest cluster(s) can be identified. Then, we remove the authorities and hubs as well as other vertices in the cluster, and apply the HITS algorithm again to the remainings to further explore the presence of more clusters. Also, for each of the identified clusters, the same procedure can be applied recursively until no more clusters can be identified. We call such a **re**cursive application of **HITS** — the **ReHITS** algorithm. We consider such an iterative procedure as an approximation to make the extraction of non-principle eigenvectors tractable. We are currently studying the theoretical link between them.

To realize the proposed idea, there are a number of implementation issues needed to be carefully considered. First of all, the identified set of authorities and hubs is just an implicit representation of the clusters without explicitly defining the cluster boundaries. So, we need a way to infer the boundaries so as to make hierarchical clustering possible. Second, the authorities and hubs obtained in each round of HITS can come from a *mixture* of multiple clusters with the similiar degree of inter-connectivity. It is essential to separate them as different logical clusters. Lastly, it would be important for the extracted clusters to be properly organized for subsequent visualization or processing needs.

3.2. Directory subtrees as cluster boundaries

One obvious approach to address the aforementioned issues is to further analyze the local hyperlink structures around the extracted authorities and hubs. This, however, could be computationally expensive and yet the performance depends very much on how the navigation hyperlinks are designed. With the rationale that people tend to put related materials not too far apart in the site directory hierarchym, we propose to utilize the path information embedded in the URLs of the authorities and hubs with the hope of achieving the same goal. We first scan through the authorities and extract those with their authority scores higher than a fraction p of $\mathbf{A}$'s maximum value.[3] The extracted authorities will then be grouped if part of their paths have overlaps.[4] The longest common path of each group is considered to be a potential directory subtree containing logical clusters. For example, the following authorities

http://www.abc.edu/doc/java/classes/index.html
http://www.abc.edu/doc/java/index.html
http://www.abc.edu/doc/proj/Slides/index0.html
http://www.abc.edu/doc/proj/Slides/App/page.html

will be grouped and form two potential subtrees as

Potential Subtree 1: /doc/java/
Potential Subtree 2: /doc/proj/Slides/

After this grouping process, a set of potential subtrees, denoted as $\mathbf{T}^A$, is resulted. A similar procedure can be applied to the hubs to identify another set of potential subtrees, denoted as $\mathbf{T}^H$. The common elements in the two sets $\mathbf{T}^A$ and $\mathbf{T}^B$ forms the "confirmed" set of subtrees, $\mathbf{T}$ with the identified clusters embedded. We believe that this can make sure that all the authorities and hubs corresponding to the extracted clusters are logically coherent. The subtrees being identified, i.e., $\mathbf{T}$, will then be removed from $\mathbf{G}$ and this complete one application of HITS. The whole process will be repeated until there is no more cluster found. By applying the aforementioned procedures recursively to all the identified clusters, a hierarchy of clusters can be extracted. The situation is similar to the Scatter/Gather methodology [4]. The use of this URL-based approach makes our algorithm both effective and efficient. The detailed algorithm is shown in Figure 2.

4. Experimental Results and Discussion

In order to evaluate the performance of our proposed ReHITS algorithm, we have applied it to three web sites with their details shown in Table 2. The datasets used in our experiment contain all the web pages retrieved from the web sites, except for media files (e.g., *.jpg), dynamically generated pages (e.g., *.php) as well as other resources accessed with protocols other than http (e.g., ftp). The graph data structure of each of the web sites was then created by extracting the hyperlinks embedded in the <A> tags. Note that all the javascripts in the web pages are discarded.[5]

[3] In our experiments, we find that $p = 0.1$ turns out to be a good choice

[4] Note that the path in a URL refers to the sub-string starting from right after the domain name to just before the file name.

[5] Nowadays, more and more web pages have their drop-down menus implemented using Javascript. Removal of the javascripts results in unnecessary elimination of some hyperlinks. In this paper, we ignore this issue for simplicity.

1. Set $\mathbf{G} = \{\mathbf{V}, \mathbf{E}\}$ to be the input graph (the whole web site initially).
2. $\mathbf{R}$:= the root of the substree containing $\mathbf{V}$ (the whole web site hierarchy initially)
3. Do
   (a) Apply the HITS algorithm to $\mathbf{G}$ to compute the authority scores $\mathbf{A}$ and hub scores $\mathbf{H}$ for all the vertices $\mathbf{V}_i \in \mathbf{V}$
   (b) Extract the elements from $\mathbf{V}$ with $\mathbf{A}_i < p \times \max\{\mathbf{A}\}$ and form $\mathbf{V}_p^A$
   (c) Group elements in $\mathbf{V}_p^A$ based on their directory paths to identify potential subtrees (with clusters embedded) and store their roots as $\mathbf{T}^A$.
   (d) Extract the elements from $\mathbf{V}$ with $\mathbf{H}_i < p \times \max\{\mathbf{H}\}$ and form $\mathbf{V}_p^H$
   (e) Group elements in $\mathbf{V}_p^H$ based on their directory paths to identify potential subtrees (with clusters embedded) and store their roots as $\mathbf{T}^H$.
   (f) $\mathbf{T} := \{path : path \in \mathbf{T}^A \wedge path \in \mathbf{T}^H\}$
   (g) For each $\mathbf{T}_i$ in $\mathbf{T}$,
      - remove from $\mathbf{V}$ the elements $\mathbf{V}_p^A \cup \mathbf{V}_p^H$ which are under $\mathbf{T}_i$ in the site hierarchy as the identified cluster $\mathbf{C}_i$, and remove from $\mathbf{E}$ all the edges pointing to those elements.
      - remove from $\mathbf{V}$ the elements which are under $\mathbf{T}_i$ but not in $\mathbf{C}_i$ as $\mathbf{V}_{\mathbf{T}_i \setminus \mathbf{C}_i}$, and remove from $\mathbf{E}$ all the edges pointing to those elements.
      - $\mathbf{G}_i := \{\mathbf{V}_{\mathbf{T}_i \setminus \mathbf{C}_i}, \mathbf{E}_{\mathbf{T}_i \setminus \mathbf{C}_i}\}$ where $\mathbf{E}_{\mathbf{T}_i \setminus \mathbf{C}_i}$ are the edges corresponding to $\mathbf{V}_{\mathbf{T}_i \setminus \mathbf{C}_i}$.
   (h) Repeat until $\mathbf{T} = \mathbf{R}$ /* no more clusters found */
4. Output $\{\mathbf{C}_i\}$ as the logical clusters.
5. Treat each element in $\{\mathbf{G}_i\}$ as an individual input graph and repeat all the steps recursively.

Figure 2. ReHITS algorithm

We implemented the ReHITS algorithm in Java and applied it to the prepared datasets. Using a Pentinum II machine, the computational time for the whole cluster identification process is within a few minutes for the scale of the Web sites we considered. In the following, we use the web site of www-db.stanford.edu as the major brenchmark dataset as it was also used in [11]. The results of applying the ReHITS algorithm to this web site are shown in Table 3 and 4. A set of logical clusters is successfully identified, each being summarized by their sizes as well as their own authorities and hubs which lead to the cluster formation.

4.1 Characteristics of identified clusters

After carefully evaluating the logical clusters identified, there are a number of observations worth discussing:

URL	No. of pages
http://www.eee.hku.hk/	14711
http://www-db.stanford.edu/	12342
http://www.ox.ac.uk/	6091

Table 2. Web sites tested.

1. There were mainly three types of clusters identified, namely system documentations, research project description, and personal web sites. The system documentation clusters were identified first and then followed by the other two types. Also, among the large number of project descriptions and personal web sites, the identified ones were those with more contents and hyperlinks in them. We believe that these are the targets to look for in information foraging.

2. Most of the best authority and hubs were either home or index pages of the identified clusters. Some even reveals the authoritative research area of an individual. For example, for the cluster /∼burback/ found in the seventh iteration of ReHITS (see Table 3), it is interesting to find that Theorem 4 of Ron Burback's PhD thesis was identified as the most authoritative page of his personal home and the theorem summary page of this thesis was identified as the strongest hub. By refering back to the web site, the "Theorem 4" turns out to be the formal foundations of the thesis research.

3. It is interesting to observe that index pages, which intuitively should serve as hubs, sometimes appear as authorities of some clusters (e.g., in /∼testbed/cos/) while sometimes as hubs (e.g., in /∼LIC/). Also, there are also cases that both the most authoritative and hub pages are in fact the same page (e.g., /db_pages/). The discrepancy is mainly due to the different design of hyperlinks as navigation aids.

4. It is also observed that the cluster /pub/ does not contain any strong hubs or authorities. By carefully examining it, it is noted that it basically contains one web page pointing to a directory structure, instead of specific documents or pages. So, /pub/ possesses a giant tree structure without any obvious authorities or hubs. This also implies that our algorithm is not effectively in extracting groups of logically related web pages which follow tree structure.

4.2 Hierarchical structure of clusters

As described in Section 3, the cluster identification process can be repeated recursively. For example, by applying it to the cluster /dbseminar/, seminars archives of different years were identified as shown in Table 5. In fact, if the recursion continues, individuals' presentations within each archives can be further extracted as shown in Table 6. As noted in Table 6, there were only six presentations while there are eleven in the original web sites. The fact is that only seven of the eleven presentation got hyperlinks (but one of them is broken) and the others are either in postscript or pdf formats. So, as a whole, all the html slides are successfully detected. Similar hierarchical structures also present in other clusters. However, due to the limitation of space, they are not illustrated in the paper.

In addition to www-db.stanford.edu, we have also applied the ReHITS algorithm to two more web sites — http://www.eee.hku.hk and http://www.ox.ac.uk. Similar results were observed and tabulated in Table 7 and 8. In order to visualize those Web site summaries, we have also built a visualization tool with a GUI. Figure 3 is a snap shot of the visualization tool.

4.3. Qualitative comparison with related algorithms

A related project on web site mining done by Li and Candan was reported in [11, 3]. In terms of the methodology adopted, Li&Candan system involves three major steps, namely entry point identification using a set of heuristic rules, logical domain identification by merging Web pages, and finally representing the domain using a simpler graph via random walks. The methods used for each steps are independent to each other. For our system, we have all three steps performed in an integrated manner as a recursive application of HITS. All the clusters' summaries are computed at the same time when the clusters are identified (c.f. segmentation by recognition paradigm in pattern recognition). Regarding the cluster representation, with the use of HITS as its core, the ReHITS algorithm can guarantee that each of the identified clusters contains not only an obvious home page but also a densely connected bipartite graph under it. The existence of such bipartite graphs have been considered as good indications of informative self-contained resources in many other works [8]. Besides, there have been quite some works on enhancing the performance of HITS, e.g., weighting the links by contents similarity between Web pages. We believe all such research efforts can easily be incorporated into the ReHITS algorithm. Similar advantages are not shared by Li&Candan system. However, our proposed algorithm cannot result in detailed logical partititions which is the strength of Li&Candan system.

Another related work is done by Kumar *et al.* [10] and later further enhanced by Reddy *et al.* [15] on trawling the global Web for emerging cyber-communities. The approach represents a community as a bipartite graph of hyperlinks, where it is called a *core*. To extract those cores, an iterative edge pruning step is first applied and the cores are then

k	Clusters found	Remarks
1	/~melnik/rdf/api-doc/ (182)	RDF java doc. **A**: UML core class **H**: The class index
2	/~testbed/ilu/ ilu20a7doc/ (25)	ILU alpha 7 manual **A**: Using ILU in ANSI C **H**: Table of content
3	/~testbed/ilu/ ilu20doc (24)	ILU manual **A**: Using ILU in ANSI C **H**: Table of content
4	/CHAIMS/Code/ Wrappers/ RMI_Wrapper/ Vers_2_0/Doc/ (54)	A java package doc. **A**: Class Invocation **H**: The class index
	/cs446/Assignments/ Assignment3/Doc/	*A clone of the above*
5	/~burback/dcg/ (44)	The distributed computing group - two-year plan **A**: Appendix (list of tables) **H**: Index to the tables
6	/pub/gio/CS99I/ (89)	A seminar course **A**: Course references **H**: A topic with many ref.
7	/~burback/ (476)	A personal home **A**: WaterSluice S/W Eng. Methodology **H**: Main Theorem summary
8	/people/ (22)	Old personal homes **A**: Gio Wiederhold's home **H**: Gio Wiederhold's home
	/CHAIMS/ (515)	The CHAIMS project **A**: Project home **H**: *no strong hubs*
	/pub/ (3768)	Publication directory **A**: *no strong authority* **H**: *no strong hub*
	/LIC/ (357)	The LIC project **A**: A slide presentation home **H**: A slide presentation index
9	/dbseminar/Archive/ SpringY96/ (62)	Seminars in Spring 96 **A**: A slide presentation index **H**: One particular slide
10	/~testbed/cos/ (57)	Common Obj. Services **A**: A slide presentation index **H**: One particular slide
11	**/dbseminar/** (1342)	**DB Seminars from 97** **A**: Archive year index **H**: Index page of seminars in 1999

Table 3. Results of applying ReHITS to http://www-db.stanford.edu, where k is the number of HITS iterations (Part I).

k	Clusters found	Remarks
12	/∼hector/photos/ (2117)	A photo archive **A**: A photo index page **H**: Another photo index page
	/∼vassalos/ (22)	A personal home **A**: Publication abstracts **H**: Publication index page
	/∼ullman/ (399)	A personal home **A**: Ullman's home **H**: Ullman's home
	/db_pages/ (15)	Web site overview **A**: Index to members **H**: Index to members
	/∼widom/ (43)	A personal home **A**: XML Management Research **H**: Jennifer Widom's home
13	/lore/ (33)	The LORE project **A**: Publication index **H**: Research overview
14	/SKC/ (84)	The SKC project **A**: Project home **H**: A slide presentation index
	/∼jan/ (43)	A personal home **A**: Whale watching **H**: Jan Jannink's home
15	/∼crespo/ (163)	A personal home **A**: The first presentation slide **H**: A slide presentation index

Table 4. Results of applying ReHITS to http://www-db.stanford.edu, where k is the number of HITS iterations (Part II).

/dbseminar/		
k	Clusters found	Remarks
1	/Archive/ SpringY99/ (563)	Spring 99 Seminars **A**: Manasse's slides home **H**: Manasse's slides index
2	**/Archive/ FallY99/** (447)	**Fall 99 Seminars** **A**: Russakovskii's slides home **H**: Russakovskii's slides index
3	/Archive/ FallY97/ (94)	Fall 97 Seminars **A**: Jan Jannink's slides home **H**: Jan Jannink's slides index
4	/Archive/ (225)	Other Seminar Archives **A**: Genome Databases Seminar Abstracts **H**: Genome Databases Seminar home

Table 5. Results of applying ReHITS to the cluster /dbseminar/.

identified using an algorithm similar to *a priori* algorithm. One major limitation of this algorithm is that the size of the cores has to be predetermined. In a web site, the size of those cores can vary greatly.

The use of graph theoretics methods is yet another approach which could be useful for web site mining. Based on the maxflow/minimum cut framework, Flake *et al.* [6] proposed an algorithm together with a focused crawler to identify communities in the global Web. We believe that similar algorithms can also be applied to web sites for clustering. But again, the algorithm does not provide a way to represent the contents of the clusters and we did that using the authority and hub pairs.

In addition to more extensive performance comparison with other approaches, there are still a number ways to extend our work. The clusters extracted from a web site are of various content types, e.g., in our case, research projects, personal homes, documentations, etc. We believe that it will be useful if we can automatically categorize the clusters, which however has to rely on further information from the web page contents. Besides, it will be interested to integrate also text summarization techniques so that not only a summarized structure, but also a textual summary, can be obtained for each cluster.

From the perspective of *Web site summarization* which includes *content summarization* [17] and *structure summarization* [3], our work definitely belongs to the latter one. Web site content summarization aims at extracting keywords, key-terms and key phrases to summarize the entire web site. Information retrieval and natural language processing techniques are commonly used to generate the "summary". Instead of generating a content summary, Web site structure summarization emphasizes on converting the graph structure of a Web site into a simpler data structure with representative web pages of the Web site stored in it. In fact, one can first perform Web site clustering and then extract representation subgraphs from each cluster forming the summary of the Web site structure.

5. Conclusion

In this paper, we have demonstrated how to effectively integrate the hyperlink topology and the site hierarchy for logical cluster identification in a web site. By extending the HITS algorithm, we proposed a a novel recursive algorithm called ReHITS where a hierarchy of logical clusters can be identified. Besides, each computed cluster is associated with a authority and hub pair which appears naturally as a by-product of the algorithm. We have tested the effectiveness of the algorithm using three web sites, and show that the proposed method for cluster identification is both efficient and yet powerful.

Acknowledgment

We thank Charlie Wong for implementing the recursive

/dbseminar/Archive/FallY99/	
Clusters found	Remarks
/russakovskii-slides/ (117)	Russakovskii's Presentation **A**: Slide home **H**: Slide index
/meersman-slides/ (115)	Meersman's Presentation **A**: Slide home **H**: Slide index
/hector-slides/ (69)	Hector's Presentation **A**: Slide home **H**: Slide index
/fankhauser-slides/ (59)	Fankhauser's Presentation **A**: Slide home **H**: Slide index
/malhotra-slides/ (39)	Malhotra's Presentation **A**: Slide home **H**: Slide index
/sikka-slides/ (37)	Sikka's Presentation **A**: Slide home **H**: Slide index

Table 6. Results of applying ReHITS to the cluster /dbseminar/Archive/FallY99/.

version of the algorithm. This research is supported by UST AoE-IT Grant UST/AOE/01-02/1.

References

[1] Giuseppe Attardi, Antonio Gullí, and Fabrizio Sebastiani. Automatic Web page categorization by link and context analysis. In *Proceedings of European Symposium on Telematics, Hypermedia and Artificial Intelligence*, pages 105–119, Varese, IT, 1999.

[2] Krishna Bharat and Monika Rauch Henzinger. Improved algorithms for topic distillation in a hyperlinked environment. In *The 21st ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 104–111, 1998.

[3] K.S. Candan and Wen-Syan Li. Discovering web document associations for web site summarization. In *Proceedings of the Third International Conference on Data Warehousing and Knowledge Discovery*, pages 152–161, Munich, Germany, 2001.

[4] Douglass Cutting, David Karget, and Jan Pedersen. Constant interaction-time Scatter/Gather browsing of large document collections. In *Proceedings of the 16th Annual International ACM/SIGIR Conference*, Pittsburgh, PA, 1993.

[5] Martin Ester, Hans-Peter Kriegel, and Matthias Schubert. Web site mining: A new way to spot competitors, customers and suppliers in the world wide web. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Edmonton, Canada, 2002.

[6] Gary William Flake, Steve Lawrence, C. Lee Giles, and Frans Coetzee. Self-organization of the web and identification of communities. *IEEE Computer*, 35(3):66–71, 2002.

[7] Eric J. Glover, Kostas Tsioutsiouliklis, Steve Lawrence, David M. Pennock, and Gary W. Flake. Using web structure for classifying and describing web pages. In *Proceedings of the Eleventh International Conference on World Wide Web*, Hawaii, USA, 2002.

[8] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[9] Raymond Kosala. Web mining research: A survey. *SIGKDD Explorations*, 2(1), 2000.

[10] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and Eli Upfal. Stochastic models for the web graph. In *Proceedings of the 41th IEEE Symp. On Foundations of Computer Science*, pages 57–65, November 2000.

[11] Wen-Syan Li, Okan Kolak, Quoc Vu, and Hajime Takano. Defining logical domains in a Web site. In *Proceedings of the 11th ACM Conference on Hypertext and Hypermedia*, pages 123–132, San Antonio, TX, 2000.

[12] Bing Liu, Yiming Ma, and Philip S. Yu. Discovering unexpected information from your competitors's web sites. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, USA, 2001.

[13] B. Mobasher and R. Cooley. Automatic personalization based on Web usage mining. *Communications of the ACM*, 43(8):142–151, 2000.

[14] Mike Perkowitz and Oren Etzioni. Towards adaptive web sites: Conceptual framework and case study. *Artificial Intelligence*, 118:245–275, 2000.

[15] P. Krishna Reddy and Masaru Kitsuregawa. An approach to build a cyber-community hierarchy. In *Proceedings of SIAM ICDM Workshop on Web Analysis*, Arlington, VA, 2002.

[16] Jaideep Srivastava, Robert Cooley, and Mukund. Web usage mining: Discovery and applications of usage patterns from web data. *SIGKDD Explorations*, 1(2), 2000.

[17] Yongzheng Zhang. *World Wide Web Site Summarization*. PhD thesis, Faculty of Computer Science, Dalhousie University, 2002.

k	Clusters found	Remarks
1	/LDP/LDP/ linuxcookbook/html/	Linux cookbook
2	/LDP/LDP/nag2/	Network admin. guide
3	/LDP/LDP/LG/	Linux Gazette
4	/LDP/HOWTO/ Antares-RAID- sparcLinux-HOWTO/	An how-to doc.
5	/LDP/LDP/gs/	Linux installation doc.
6	/LDP/	Linux doc. project
7	/eee/undergrad/ curricul/	Undergraduate curriculum
8	/computer/unix/	Unix commands
9	/∼mpsev/ /ITSAbstract/	The MPSEV project
10	/∼cea/	Comp. Eng. Assoc.
11	/∼ktchau/	Personal home
12	/∼ugsnews/	Undergraduate Forum
13	/eee/	Dept's overview
14	/∼ykwok/advice/	Readings for PGs
15	/computer/eeefaq /∼electsoc/ /seminars/	Computer FAQs Elect. Eng. Soc. Seminars listing
16	/news/	Dept's newsletter
17	/∼eese/	A degree program
18	/power_electronics_lab/ /∼ieeesb/ /∼kyeung/	Power Elect. Lab IEEE Student Branch Personal home
19	/∼ykwok/ /computer/	Personal home Computing facilities

Table 7. Results on http://www.eee.hku.hk.

k	Clusters found	Remarks
1	/gazette/1996-7/supps/	A journal supplements
2	/blueprint/ /gazette/	An official newsletter An official journal
3	/it/ /publicrelations/ /aboutoxford/ /visitors/ /contact/	IT resources PR office Oxford's overview Visitors' Information Contact Information
4	/services/ /associnst/ /departments/	On-campus services Associated institutions Academic departments
5	/innovation/ /business/	Oxford's leading fields Business acitivities
6	/year2000/	Year 2000 overview

Table 8. Results on http://www.ox.ac.uk.

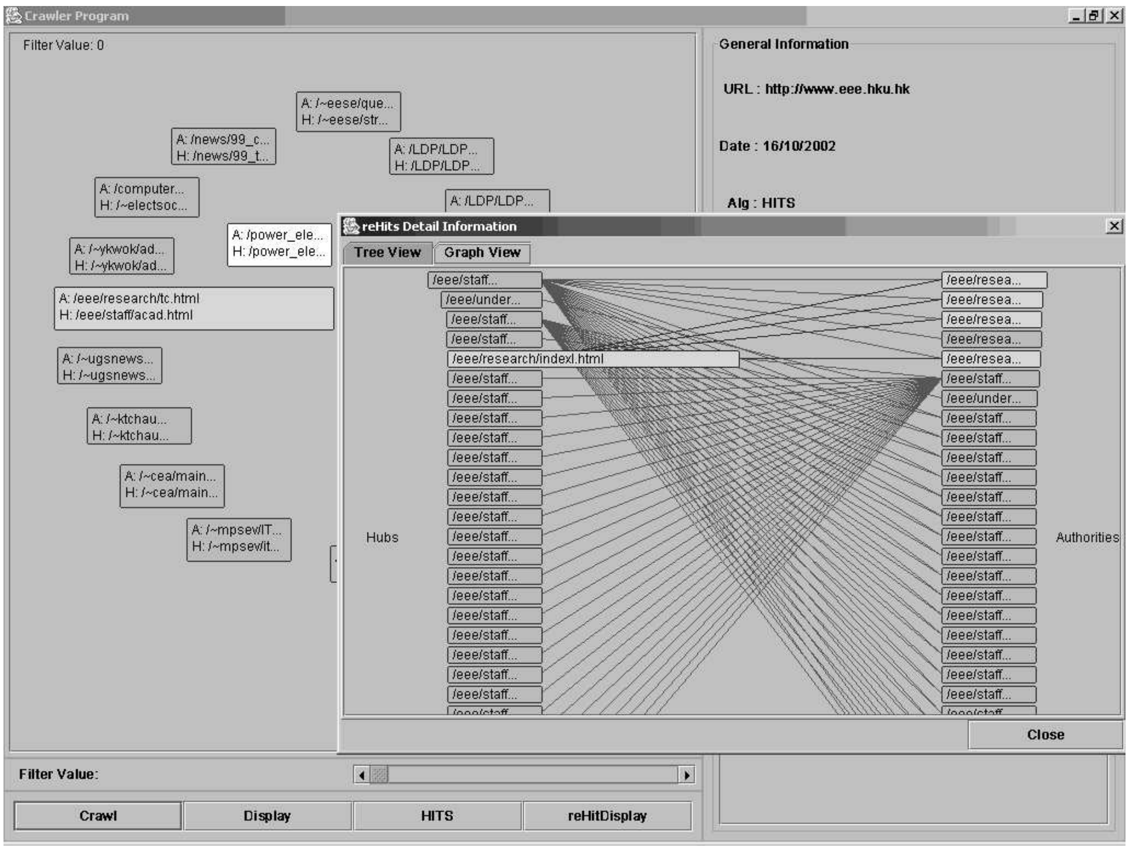

Figure 3. A snap shot of our visualization tool.