LinkSelector: A Web Mining Approach to Hyperlink Selection for Web Portals

Xiao Fang and Olivia R. Liu Sheng
Department of Management Information Systems
University of Arizona, AZ 85721
{xfang,sheng}@bpa.arizona.edu

Submitted to ACM Transactions on Internet Technology.

Abstract

As the size and complexity of websites expands dramatically, it has become increasingly challenging to design websites on which web surfers can easily find the information they seek. In this paper, we address the design of the portal page of a website, which serves as the homepage of a website or a default web portal. We define a new and important research problem – hyperlink selection: selecting from a large set of hyperlinks in a given website a limited number of hyperlinks for inclusion in a portal page. The objective of hyperlink selection is to maximize the efficiency, effectiveness and usage of a web site's portal page.

We propose a heuristic approach to hyperlink selection, LinkSelector, which is based on relationships among hyperlinks – structural relationships that can be extracted from an existing website and access relationships that can be discovered from a web log. LinkSelector calculates preferences of hyperlinks and preferences of hyperlink sets from these relationships. Using these preferences, we develop and incorporate a clustering algorithm into LinkSelector to extract a limited number of hyperlinks from a large set of hyperlinks. We compared the performance of LinkSelector with that of the current practice of hyperlink selection (i.e., manual hyperlink selection by domain experts) and with data mining methods – classical hierarchical clustering and association rule mining, using data obtained from the University of Arizona website. Results showed that LinkSelector outperformed all of these. Specifically, the three major contributions of this paper are:

- We have introduced and formally defined a new and important research problem -- hyperlink selection.
- We have proposed and shown that a web mining based hyperlink selection approach, named LinkSelector, outperformed other hyperlink selection approaches
- We have developed a new clustering algorithm and applied it to hyperlink selection. Applications of the algorithm are not limited to hyperlink selection.

1. INTRODUCTION

As the size and complexity of websites expands dramatically, it has become increasingly challenging to design websites on which web surfers can easily find the information they seek. To address this challenge, we introduce a new research problem in this area, hyperlink selection, and present a web mining based approach, LinkSelector, as a solution.

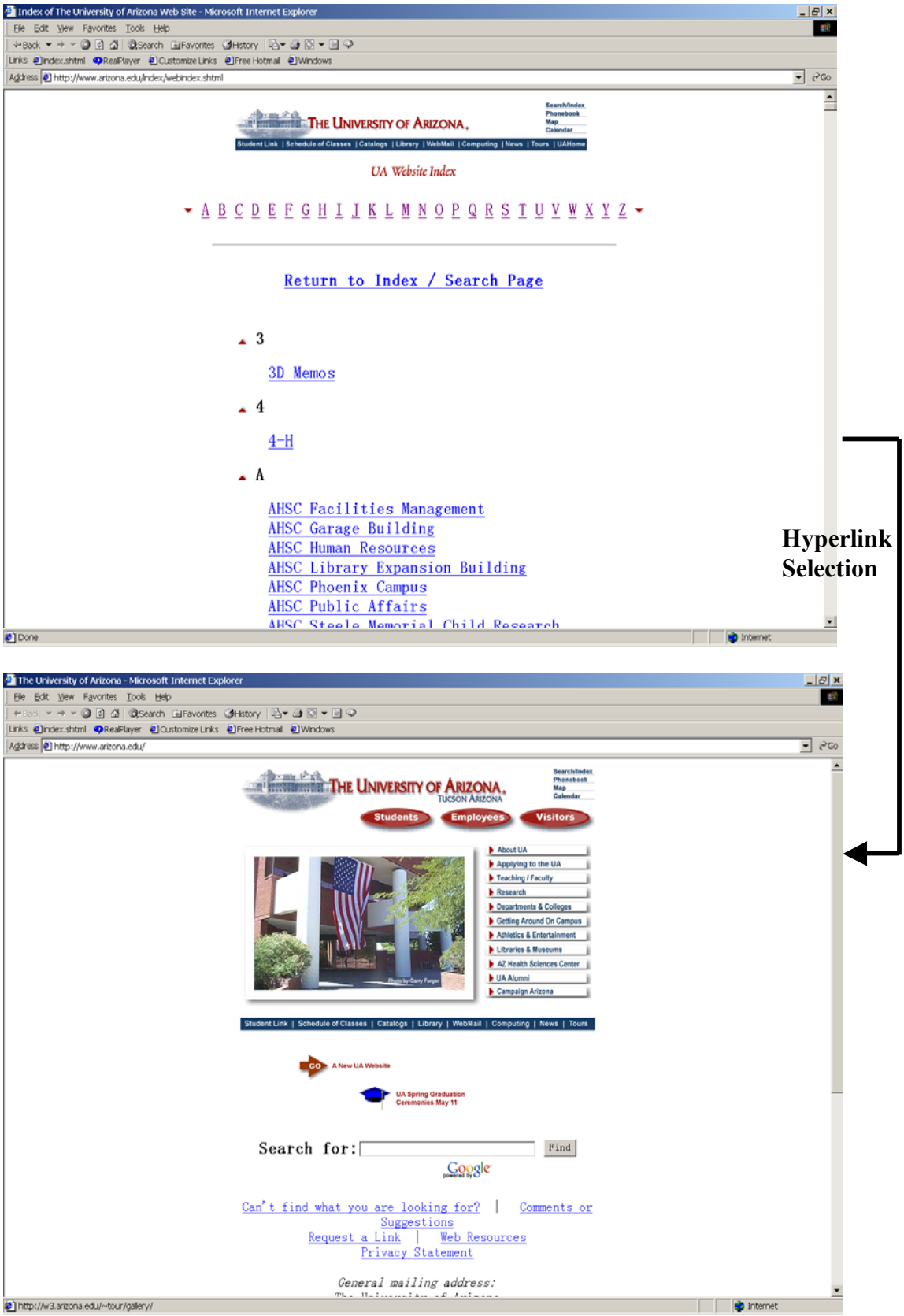
There are two dominant ways through which web surfers find the information they seek [Chakrabarti 2000]: using search engines and clicking on hyperlinks. Research on the former is concerned with improving recall and precision [Lawrence and Giles 1998;1999; Chakrabarti 2000] of search engines. Our research, however, concentrates on improving the efficiency of the second way of web information searching. As web surfers click on a group of hyperlinks to find the information they seek, placing appropriate hyperlinks in web pages is critical to improving their web information searching efficiency. In particular, this paper focuses on placing appropriate hyperlinks in the portal page of a website, which is the entrance to a website.

The homepage of a website is one type of portal page. Homepages which guide users to locate the information they seek easily create a good first impression and attract more users, while homepages which make information searching difficult result in a bad first impression and corresponding user loss [Nielson and Wagner 1996]. A default web portal is another type of portal page. Recently, web portals that serve as a personal entrances to websites have attracted more and more attention. Universities such as UCLA have built educational web portals (e.g., My UCLA, http://my.ucla.edu); corporations such as Yahoo! have developed commercial web portals (e.g., My Yahoo!, http://my.yahoo.com). For practical purposes, portal service providers (e.g., Yahoo!) provide portal users with a standard default web portal, which the users can personalize (e.g., add or remove hyperlinks from the default web portal). As the first version of a web portal encountered by portal users, the default web portal plays an important role in the

success of a web portal. Moreover, according to My Yahoo!, most users never customize their default web portals [Manber et al. 2000]. This finding makes the default web portal even more critical.

A portal page consists of hyperlinks selected from a hyperlink pool, which is a set of hyperlinks pointing to top-level web pages[1]. Usually, the hyperlink pool of a website consists of hyperlinks listed in the site-index page or the site-directory page. As shown in Fig. 1, hyperlinks in the portal page of the University of Arizona website (http://www.arizona.edu) are selected from its hyperlink pool. The hyperlink pool consists of hyperlinks in its site-index page (http://www.arizona.edu/ index/webindex.shtml). Hyperlinks in the portal page of My Yahoo! (http://my.yahoo.com) are also selected from its hyperlink pool. The pool, in this case, consists of hyperlinks in its site-directory page.

[1] Web pages in a website are organized in a hierarchy in which a high level web page is an aggregation of its low level web pages [Nielson 1999]. For example, the web page of faculty list is one level higher than its corresponding faculty homepages and it is an aggregation of its corresponding faculty homepages. For a university website, top level web pages include the web page of department list and the web page of computing resources etc..



(a)

Fig. 1. (a) The hyperlink pool (top) and the portal page (bottom) of the website of the University of Arizona;

(b)

Fig. 1 (b) The hyperlink pool (top) and the portal page (bottom) of My Yahoo!.

Given the web design principle that scrolling must be avoided in portal pages [Nielson 1999], a well-designed portal page normally contains several dozen (i.e., usually less than 4 dozen) hyperlinks[2]. However, the hyperlink pool of a typical website has at least several hundred hyperlinks. For example, the portal page of the University of Arizona website consists of 32 hyperlinks while the hyperlink pool has 743 hyperlinks. It is computationally too expensive to exhaust all combinations of several dozen hyperlinks from a hyperlink pool with several hundred hyperlinks and find the one that is the most efficient in guiding web surfers to find the information they seek. In this particular, for example, the number of combinations of selecting 32 hyperlinks from 743 hyperlinks is 1.44E+56 (i.e., C_{743}^{32}). Current practice of hyperlink selection relies on domain experts' (e.g., website designers) experiences. Obviously, such selection is subjective. In addition, it reflects only website designers' perspectives on what hyperlinks should be selected, not web surfers' perspectives. The second perspective should be emphasized as the purpose of hyperlink selection is to reduce web surfers' information searching efforts, not web designers'.

In comparison, our hyperlink selection method, LinkSelector, incorporates both patterns extracted from the structure of a website and those discovered from a web log, which records web surfers' behaviors of information searching. LinkSelector first employs web mining techniques [Kosala and Blockeel 2000] to extract the above-mentioned patterns and calculates preferences of hyperlinks and hyperlink sets defined in this paper from these patterns. LinkSelector then selects hyperlinks from a given hyperlink pool by running the calculated preferences through a clustering algorithm developed in this paper. There are three major contributions of this paper.

– We have proposed and formally defined a new and important research problem – hyperlink selection.

[2] Placing too many hyperlinks in a portal page will cause some hyperlinks to be visible only when scrolling down the window of the page. Unfortunately, according to Neilson's research [Nielson 1999], web surfers rarely scroll down the window of a portal page.

- We have proposed and shown that a web mining based hyperlink selection approach outperformed other hyperlink selection approaches.
- We have developed a new clustering algorithm and applied it to hyperlink selection. Applications of this algorithm are not limited to hyperlink selection. Currently, we are adapting this algorithm to clustering large itemsets discovered via association rule mining.

The rest of the paper is organized as follows. We review related work in section 2. In section 3, we propose metrics to measure the quality of a portal page and formally define the hyperlink selection problem. A web mining based approach for hyperlink selection, LinkSelector, is presented in section 4. In section 5, we evaluate the performance of LinkSelector using data obtained from the University of Arizona website and the metrics proposed in section 3. We conclude the paper in section 6.

2. RALATED WORK

In this section, we review works on web mining on which LinkSelector is based. In [Cooley et al. 1997], web mining is defined as the process of discovering and analyzing useful information from the Web. A good survey on web mining research can be found in [Kosala and Blockeel 2000].

Srivastava et al. classified web data into content, structure and usage [Srivastava et al 2000]:

- Content is the data in web pages. It usually consists of texts and graphics.
- Structure is the data describing the organization of the Web, such as hyperlinks.
- Usage is the data that describe web surfers' information searching behaviors. Web usage data can be found in web logs.

For different types of web data, corresponding web mining methods are developed. Web content mining is the process of automatically retrieving, filtering and categorizing web documents, a

good survey on which can be found in [Chakrabarti 2000]. As it typically makes use of only texts on web pages, valuable information implicitly contained in hyperlinks is overlooked. As a complement to web content mining, web structure mining [Kleinberg 1998; Brin and Page 1998; Chakrabarti et al. 1999] infers useful patterns from the Web's link topology to help retrieve high quality documents from the Web. Web usage mining [Srivastava et al. 2000] is the process of applying data mining techniques to discover web access patterns from a web log. Due to its greater relevance to this research, we review works on web usage mining in more detail.

The data used for web usage mining are web logs. A web log is a collection of data that explicitly records web surfers' behaviors of information searching in a website. Fig. 2 shows a sample web log collected by a web server at the University of Arizona. Useful attributes for web usage mining in a web log include IP address, time and URL, which explicitly describe who at what time accessed which web page. Additional attributes include status of a HTTP request and the count of bytes returned by a web server.

IP Address[3]	Time	Method/URL/Protocol	Status	Size
123.456.789.001	[01/Sep/2001:05:38:33 -0700]	"GET /working/index.shtml HTTP/1.0"	200	7134
123.456.789.001	[01/Sep/2001:05:38:34 -0700]	"GET /working/images/head-employ.gif HTTP/1.0"	200	765
123.456.789.001	[01/Sep/2001:05:38:34 -0700]	"GET /working/images/work.jpg HTTP/1.0"	200	8864
123.456.789.001	[01/Sep/2001:05:38:34 -0700]	"GET /working/images/staff-quicklinks1.gif HTTP/1.0"	200	1618

Fig. 2. A sample web log collected by a web server at the University of Arizona

Projects of web usage mining are classified into two groups: general-purpose projects and specific-purpose projects [Srivastava et al. 2000]. General-purpose projects, such as [Chen et al. 1996; Cooley et al. 1999], focused on web usage mining in general. Cooley et al. proposed an architecture and specific steps for web usage mining and presented a method to identify

[3] To protect privacy of web users, IP addresses in this table are artificial IP addresses.

potentially interesting patterns from mining results (e.g., patterns in which unlinked web pages are visited together frequently) [Cooley et al. 1999]. Chen et al. explored a new data mining capability to mine path traversal patterns from web logs [Chen el al. 1996]. Specific-purpose projects focused on applications of web usage mining. Web usage mining can be used to improve organizations of websites. Adaptive website project [Perkowitz and Etzioni 2000] used web visiting patterns learned from web logs to automatically improve organizations and presentations of websites. Spiliopoulou and Pohle exploited web usage mining to measure and improve the success of websites [Spiliopoulou and Pohle 2001]. Lee and Podlaseck presented an interactive visualization system that provides users with abilities to actively interpret and explore web log data of online stores to evaluate the effectiveness of web merchandising [Lee and Podlaseck 2001]. Web usage mining can also be used to personalize users' web surfing experience. In [Yan et al. 1996], clusters of visitors who exhibited similar information needs (e.g., visitors who accessed similar web pages) were discovered via web usage mining. These clusters could be used to classify new visitors and dynamically suggest hyperlinks for them. Mobasher et al. [Mobasher 2001] presented techniques to learn user preferences from web usage data using data mining techniques, such as association rule mining. Based on the learned preferences, dynamic hyperlinks could be recommended for active visiting sessions. Anderson et al. developed MINPATH, an algorithm that automatically suggests useful shortcut links in real time to improve wireless web navigations [Anderson et al. 2001]. A complete survey of web usage mining research by year 2000 can be found in [Srivastava et al. 2000].

Research employing only web usage mining, such as work described in [Chen el al. 1996], extracted web surfers' web visiting patterns from a web log. However, theses studies did not consider the information contained in the structure of a website, which is an important complement to web visiting patterns. In [Cooley 1999; Perkowitz and Etzioni 2000], the structure of a website was used to filter out uninteresting web visiting patterns (i.e., patterns in which directly linked web pages are visited together frequently). However, these excluded uninteresting

web visiting patterns provide valuable information for hyperlink selection. We will discuss this in section 4.1. The hyperlink selection approach proposed in this paper is based on both web usage mining and web structure mining and considers both interesting and uninteresting web visiting patterns.

3. PROBLEM DEFINITION – HYPERLINK SELECTION

To define the hyperlink selection problem, we propose three metrics to measure the quality of a portal page – effectiveness, efficiency and usage. All three are calculated from web logs. A web log can be broken down into sessions with each session representing a sequence of consecutive web accesses by the same visitor. For the convenience of readers, important notations used in this article are summarized in Table I.

Table I. Notation Summary

Notation	Description
w	a website
wl	a web log of w
s	the number of sessions in wl
S_j	a session of wl, for $j = 1,2,\cdots s$
$SHL(S_j)$	the set of hyperlinks clicked in S_j, for $j = 1,2,\cdots s$
HP	the hyperlink pool of w
$UHL(S_j)$	$UHL(S_j) = SHL(S_j) \cap HP$, web pages pointed to by hyperlinks in $UHL(S_j)$ are user-sought top-level web pages in S_j
l	the number of hyperlinks in w
L_j	a hyperlink in w, for $j = 1,2,\cdots l$
P_{L_j}	the set of hyperlinks in the web page pointed to by L_j
PHL	the set of hyperlinks in the portal page of w
EHL	$EHL = \bigcup_{\forall L_j \in PHL} P_{L_j}$, the set of hyperlinks that are contained in web pages directly pointed to by the portal page of w
HL	$HL = PHL \cup EHL$, web pages pointed to by hyperlinks in HL can be easily found from the portal page of w
N	the number of hyperlinks to be placed in the portal page of w
W	the set of web pages in w
k-HS	a set of k hyperlinks L_i, where $L_i \in HP$
σ (k-HS)	the support of a k-HS
SR	the set of structure relationships between hyperlinks in HP
AR	the set of access relationships among hyperlinks in HP
PRE_{L_i}	the preference of a hyperlink L_i, where $L_i \in HP$
PHP	a set of pairs, in which, each pair consists of a hyperlink pair with group II relationship and its preference
PHS	a set of pairs, in which, each pair consists of a hyperlink set with group II relationship and its preference
C_i	a hyperlink cluster
sim_{C_i,C_j}	the similarity between hyperlink clusters C_i and C_j
SIM	the similarity matrix

The effectiveness of a portal page is measured as the degree of easiness to find user-sought top-level web pages[4] (Definition 1) from the portal page, as hyperlinks in a portal page are selected

[4] We believe that a levelwise approach is appropriate for the design of a website. In this approach, the portal page is designed to find user-sought top-level web pages easily. Top-level web pages then are designed to locate user-sought web pages one level below easily. As the hyperlink selection approach proposed in this paper can be applied to every subsequent level, consequently, websites designed in this approach will be easy to navigate to find user-sought information. In this paper, we concentrate on designing the portal page to facilitate the search of top-level web pages.

from a pool of hyperlinks pointing to top-level web pages. We denote *w* as a website, *wl* as a web log of *w*, *s* as the number of sessions in *wl*, S_j as a session in *wl*, for $j = 1,2,\cdots s$, and $SHL(S_j)$ as the set of hyperlinks clicked in S_j. We denote the hyperlink pool of *w* as *HP*.

Definition 1: For a session S_j, web pages pointed to by hyperlinks in $UHL(S_j)$ are user-sought top-level web pages in S_j, where,

$$UHL(S_j) = SHL(S_j) \cap HP \qquad j = 1,2,\cdots s \qquad (1)$$

Usually, web pages that are 1-2 clicks away[5] from a portal page can be easily found from the portal page. We denote *l* as the number of hyperlinks in *w*, L_j as a hyperlink in *w*, for $j = 1,2,\cdots l$, P_{L_j} as the set of hyperlinks in the web page pointed to by L_j and *PHL* as the set of hyperlinks in the portal page of *w*.

Definition 2: *EHL* is the set of hyperlinks, where

$$EHL = \bigcup_{\forall L_j \in PHL} P_{L_j} \qquad (2)$$

EHL consists of hyperlinks that are contained in web pages directly pointed to by the portal page of *w*.

Web pages pointed to by hyperlinks in *HL* are 1-2 clicks away from the portal page of *w* and can be easily found from the portal page of *w*, where

$$HL = PHL \cup EHL \qquad (3)$$

---

[5] Web pages that are 1 click away from a portal page refer to web pages that are directly pointed to by hyperlinks in the portal page. Web pages that are 2 clicks away from a portal page are web pages that are pointed to by hyperlinks in web pages 1 click away.

Example 1: As shown in Fig. 3, the portal page of a web site contains hyperlinks L_1, L_2 and L_3. Web pages pointed to by hyperlinks L_1, L_2 and L_3 contain hyperlinks L_2 and L_4; hyperlink L_3; and hyperlinks L_5 and L_8 respectively.

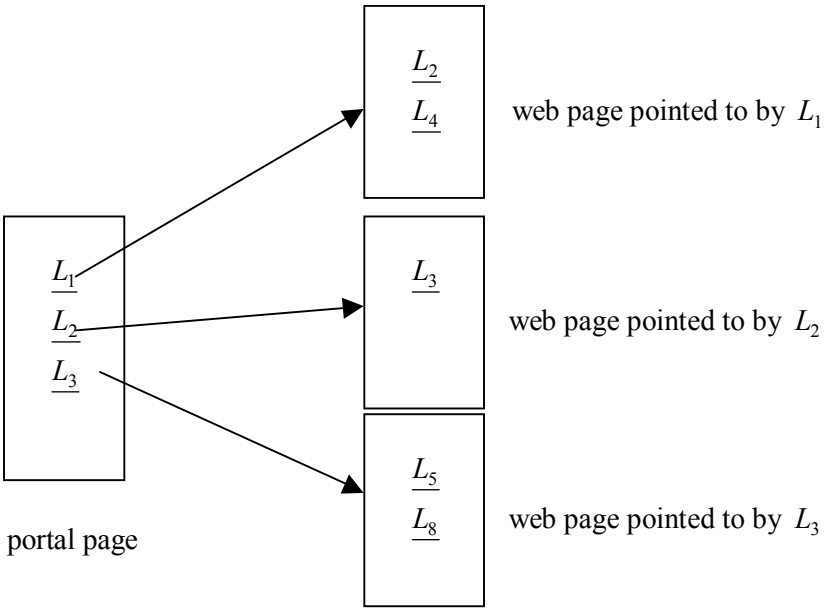


Fig. 3. *PHL* and *EHL*

In this example, $PHL=\{L_1, L_2, L_3\}$, $P_{L_1}=\{L_2, L_4\}$, $P_{L_2}=\{L_3\}$ and $P_{L_3}=\{L_5, L_8\}$. According to (2), $EHL = \bigcup_{\forall L_j \in PHL} P_{L_j} = (P_{L_1} \cup P_{L_2} \cup P_{L_3}) = \{L_2, L_3, L_4, L_5, L_8\}$; according to (3), $HL = PHL \cup EHL = \{L_1, L_2, L_3, L_4, L_5, L_8\}$.

The effectiveness of a portal page can be measured in terms of the recall rate of the portal page at two different levels – session level and web log level. For a session S_j, the more hyperlinks in $UHL(S_j)$ found in *HL*, the more user-sought top-level web pages easily found from the portal page of *w*; hence, the higher the effectiveness of the portal page of *w*.

Definition 3: For a session S_j, the session level effectiveness of the portal page of *w* is defined as:

$$effectiveness(S_j) = \frac{|UHL(S_j) \cap HL|}{|UHL(S_j)|} \qquad (4)$$

where $j = 1,2,\cdots s$ and $|X|$ denotes the cardinality of a set *X*.

Example 2: In a session S_1, web pages pointed to by hyperlinks L_1, L_{10}, L_{11}, L_2, L_{13}, L_{14}, L_5, L_9, L_7 and L_{12} were accessed. Hyperlinks L_1, L_2, L_5 and L_7 are also elements of the hyperlink pool *HP*. Hence, $UHL(S_1)$ is $\{L_1, L_2, L_5, L_7\}$, which points to user-sought top-level web pages. For the portal page given in Example 1, its session level effectiveness is,

$$effectiveness(S_1) = \frac{|UHL(S_1) \cap HL|}{|UHL(S_1)|} = \frac{|\{L_1, L_2, L_5\}|}{|\{L_1, L_2, L_5, L_7\}|} = \frac{3}{4} = 0.75$$

The result states that 75% of the user-sought top-level web pages in session S_1 can be easily found from the portal page.

Definition 4: For a web log *wl*, the log level effectiveness of the portal page of *w* is defined as:

$$effectiveness(wl) = \frac{\sum_{j=1}^{s} effectiveness(S_j)}{s} \qquad (5)$$

Given the limited number of hyperlinks that can be placed in a portal page, it is desirable to have more user-sought top-level web pages easily found from the portal page (i.e., more hyperlinks in $UHL(S_j) \cap HL$) with fewer hyperlinks placed in the portal page (i.e., fewer hyperlinks in *PHL*).

Definition 5: For a session S_j, the session level efficiency of the portal page of *w* is defined as:

$$efficiency(S_j) = \frac{\left|UHL(S_j) \cap HL\right|}{\left|PHL\right|} \qquad (6)$$

where $j = 1,2,\cdots s$ and $\left|X\right|$ denotes the cardinality of a set *X*.

Example 3: For the portal page given in Example 1 and the session S_1 given in Example 2, the session level efficiency of the portal page is,

$$efficiency(S_1) = \frac{\left|UHL(S_1) \cap HL\right|}{\left|PHL\right|} = \frac{\left|\{L_1, L_2, L_5\}\right|}{\left|\{L_1, L_2, L_3\}\right|} = \frac{3}{3} = 1$$

Definition 6: For a web log *wl*, the log level efficiency of the portal page of *w* is defined as:

$$efficiency(wl) = \frac{\sum_{j=1}^{s} efficiency(S_j)}{s} \qquad (7)$$

Usage of a portal page measures how often a portal page is visited. As the portal page constructed by LinkSelector has not been used by web surfers, we measure its usage by counting the number of user-sought top-level web pages that can be easily found from the portal page (i.e., the number of hyperlinks in $UHL(S_j) \cap HL$). It is a proper approximation because ease of finding these web pages from the portal page will attract users to visit the portal page. We define usage measured at the web log level as below.

Definition 7: For a web log *wl*, the usage of the portal page of *w* is defined as:

$$usage(wl) = \sum_{j=1}^{s} \left|UHL(S_j) \cap HL\right| \qquad (8)$$

where $\left|X\right|$ denotes the cardinality of a set *X*.

Example 4: Given a web log *wl* with 5 sessions (i.e., S_1, S_2, S_3, S_4 and S_5) and *HL* of a portal page, we found that $|UHL(S_1) \cap HL| = 1$, $|UHL(S_2) \cap HL| = 2$, $|UHL(S_3) \cap HL| = 0$, $|UHL(S_4) \cap HL| = 2$ and $|UHL(S_5) \cap HL| = 3$. According to (8), the usage of the portal page is,

$$usage(wl) = \sum_{j=1}^{5} |UHL(S_j) \cap HL| = 1+2+0+2+3=8$$

Based on the three metrics presented above, we formally define the hyperlink selection problem as below.

Definition 8: Given a website *w*, its hyperlink pool *HP* and the number of hyperlinks to be placed in the portal page of *w* – *N*, where $N < |HP|$, the hyperlink selection problem is to construct the portal page by selecting *N* hyperlinks from the hyperlink pool *HP* to maximize the effectiveness, efficiency and usage of the resulting portal page w.r.t. a web log *wl* of *w* (i.e., all metrics are measured at the web log level).

4. THE LINKSELECTOR APPROACH

As discussed in section 1, it is computationally too expensive to find the optimal solution for the hyperlink selection problem. In this section, we present a heuristic solution method named LinkSelector. Compared with the current practice of hyperlink selection based on domain experts' experience, our method incorporates patterns extracted from the structure of a website and those extracted from a web log, which records web surfers' behaviors of information searching in the website. Hence, LinkSelector is more objective and reflects web surfers' perspectives on which hyperlinks should be selected while current practice of hyperlink selection is subjective and reflects only domain experts' perspectives.

For a web site *w*, the input of LinkSelector includes *wl* -- a web log of *w*, *W* -- the set of web pages in *w*, *HP* -- the hyperlink pool of *w*, and *N* -- the number of hyperlinks to be placed in the portal page of *w*. The output of LinkSelector is the set of *N* hyperlinks selected from the hyperlink pool *HP*. We introduce LinkSelector in section 4.1. A detailed description of LinkSelector is illustrated step by step in section 4.2 through section 4.6. We discuss time complexity of LinkSelector in section 4.7.

4.1 Overview of LinkSelector

Hyperlinks in a hyperlink pool may be directly connected to each other (i.e., one hyperlink is contained in a web page pointed to by another hyperlink) or accessed together in a session. Accordingly, we categorize relationships among hyperlinks in a hyperlink pool into two types – structure relationship and access relationship.

Definition 9: For any pair of hyperlinks L_i and L_j, where $L_i \in HP, L_j \in HP$ and $L_i \neq L_j$, L_i has a structure relationship with L_j, denoted as $L_i \rightarrow L_j$, if and only if $L_j \in P_{L_i}$. L_i is the initial hyperlink and L_j is the terminal hyperlink in this structure relationship.

Example 5: As shown in Fig. 4, web page 1 contains hyperlinks L_1 and L_3; web page 2, which is pointed to by hyperlink L_1, contains hyperlinks L_2, L_4, L_6 and L_8; and web page 3, which is pointed to by hyperlink L_3, contains hyperlink L_5 and L_7. All hyperlinks are elements of the hyperlink pool *HP*.

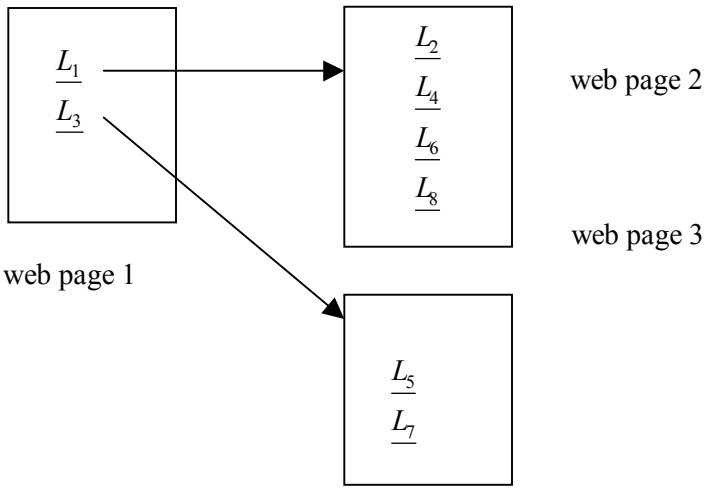


Fig. 4. Structure relationship

In this example, $P_{L_1} = \{L_2, L_4, L_6, L_8\}$ and $P_{L_3} = \{L_5, L_7\}$. According to Definition 9, structure relationship $L_1 \to L_2$ exists, in which L_1 is the initial hyperlink and L_2 is the terminal hyperlink. Similarly, structure relationships $L_1 \to L_4$, $L_1 \to L_6$, $L_1 \to L_8$, $L_3 \to L_5$ and $L_3 \to L_7$ also hold.

For a structure relationship $L_i \to L_j$, putting the initial hyperlink L_i in a portal page makes the initial hyperlink L_i an element of *PHL* and the terminal hyperlink L_j an element of *EHL* (i.e., according to Definition 1). If a hyperlink acts as the initial hyperlink in *M* structure relationships, placing it in a portal page makes it an element of *PHL* and the *M* terminal hyperlinks in all the structure relationships elements of *EHL*. As discussed in section 3, web pages pointed to by hyperlinks in *PHL* or *EHL* can be easily found from a portal page. Therefore, the more structure relationships a hyperlink participates in as the initial hyperlink, the more top-level web pages can be easily found from a portal page if the hyperlink is placed in the portal page. In Example 5, L_1 participates in four structure relationships (i.e., $L_1 \to L_2$, $L_1 \to L_4$, $L_1 \to L_6$ and $L_1 \to L_8$) as the initial hyperlink. Hence, placing L_1 in a portal page enables five top-level web pages (i.e., pages pointed to by L_1, L_2, L_4, L_6 and L_8) to be easily found from the portal page. L_3 participates in two structure relationships (i.e., $L_3 \to L_5$ and $L_3 \to L_7$) as the initial hyperlink. Hence, placing

L_3 in a portal page enables three top-level web pages (i.e., pages pointed to by L_3, L_5 and L_7) to be easily found from the portal page.

A hyperlink set *k-HS* is a set of *k* hyperlinks L_i, where $L_i \in HP$. A *k-HS* is accessed in a session S_j if and only if $k\text{-}HS \subseteq SHL(S_j)$. For a web log *wl*, the support of a *k-HS* (denoted as $\sigma(k\text{-}HS)$) is the ratio of the number of sessions in which the *k-HS* is accessed over the total number of sessions in *wl*.

Definition 10: For a *k-HS*, where $k \geq 2$, there exists an access relationship among elements in the *k-HS* if and only if its support $\sigma(k\text{-}HS)$ is greater than a pre-defined threshold. $\sigma(k\text{-}HS)$ is called the support of the access relationship.

Example 6: For a web log with 100 sessions, a 3-*HS* $\{L_1, L_2, L_3\}$ is accessed in 20 sessions. Hence, the support of the 3-*HS*, $\sigma(\{L_1, L_2, L_3\})$, is 0.2. If the threshold is set at 0.15, then there exists an access relationship among hyperlinks L_1, L_2 and L_3 and the support of the access relationship is 0.2.

In practice, structure relationships are discovered by parsing an existing website while access relationships are extracted from a web log using data mining algorithms, such as association rule mining [Agrawal et al. 1993]. For hyperlinks in a hyperlink pool, their pairwise relationships can be categorized into four groups as shown in Fig. 5.

Structure Relationship / **Access Relationship**	**YES**	**NO**
YES	I	II
NO	III	IV

Fig. 5. Pairwise relationships for hyperlinks in a hyperlink pool

– Group I relationship indicates that both structure relationship and access relationship hold between two hyperlinks. As we have discussed, hyperlinks participating in more structure relationships as initial hyperlinks will be selected for the portal page over other hyperlinks with respect to increasing the number of top-level web pages easily found from the portal page. For a structure relationship $L_i \rightarrow L_j$, the support of the access relationship between L_i and its terminal hyperlink L_j -- $\sigma(\{L_i, L_j\})$, reflects the quality of the structure relationship in hyperlink selection. The higher the support $\sigma(\{L_i, L_j\})$, the higher the possibility that top-level web pages pointed to by L_i and L_j will be accessed together in future visits[6]. In this regard, group I relationship which was regarded as uninteresting in previous research [Perkowitz and Etzioni 2000; Cooley 1999], provides us with two indicators of hyperlink preference in hyperlink selection: the number of structure relationships a hyperlink participates in as the initial hyperlink and the qualities of these structure relationships measured as supports of access relationships between the hyperlink and its terminal hyperlinks. We describe how to calculate the preference of a hyperlink using these two indicators in section 4.4.

– Group II relationship indicates that an access relationship but not a structure relationship exists between two hyperlinks. As hyperlinks with group II relationship are structurally unlinked, in order to navigate from the web page pointed to by one hyperlink to the web page pointed to by the other, web visitors have to explore the website to find the path. This creates inconvenience for web surfing and the situation becomes worse as these two hyperlinks are accessed together frequently (i.e., access relationship between the hyperlinks). In contrast, if these two hyperlinks were placed together in a portal page, users could navigate between these two hyperlinks without searching the website for a path as they could easily be found from the portal page. Hence, hyperlink pairs with group II relationship are preferred to hyperlink pairs without in hyperlink selection. For a hyperlink pair with group II

[6] This is based on an assumption that web visiting patterns are coherent in past and future visits [Perkowitz and Etzioni 2000].

relationship, the support of the access relationship between its hyperlinks is the determining factor for its preference over other hyperlink pairs with group II relationship in hyperlink selection. We describe how to calculate the preference of a hyperlink pair for group II relationship in section 4.5.

- Group III relationship indicates that a structure relationship but not an access relationship exists between two hyperlinks. This relationship reveals that the web page pointed to by the initial hyperlink in a structure relationship contains a rarely visited hyperlink which is the terminal hyperlink in the structure relationship. As hyperlink selection focuses on choosing hyperlinks for a portal page, we do not discuss group III relationship in this paper.
- Group IV relationship does not reveal any patterns between hyperlinks; thus, is not considered in this research.

LinkSelector employs group I and group II relationships to calculate preferences of hyperlinks and preferences of hyperlink pairs respectively. Based on preferences calculated, a clustering algorithm is developed to extract N hyperlinks from the given hyperlink pool. We outline LinkSelector in Fig. 6. Steps in this algorithm are described in section 4.2 through section 4.6.

Input: *wl*: a web log of a website *w*
W : the set of web pages in a website *w*
HP : the hyperlink pool of a website *w*
N : the number of hyperlinks to be placed in the portal page of a website *w*
Output: *PHL*: the set of hyperlinks in the portal page of a website *w*

Discover structure relationships;
Discover access relationships;
Calculate preferences of hyperlinks based on group I relationship;
Calculate preferences of hyperlink pairs based on group II relationship;
Cluster hyperlinks.

Fig. 6. The sketch of LinkSelector

4.2 Discover Structure Relationships

LinkSelector discovers structure relationships between hyperlinks in a hyperlink pool by parsing the web pages the hyperlinks point to. As shown in Fig. 7, the web page pointed to by each hyperlink L_i in *HP* is retrieved and parsed. A structure relationship $L_i \rightarrow L_j$ is added to structure relationships *SR* if L_j appears in the web page L_i points to, L_j and L_i are different and L_j is an element of *HP*.

Input: *HP*: the hyperlink pool of a website *w*
W : the set of web pages in a website *w*
Output: *SR*: the set of structure relationships between hyperlinks in *HP*

$SR = \phi$;
For each hyperlink L_i, where $L_i \in HP$
retrieve the web page pointed to by L_i ;
parse the web page pointed to by L_i ;
For every hyperlink L_j, where $L_j \in P_{L_i}$
If ($L_j \in HP$) $\wedge$ ($L_j \neq L_i$)
add the structure relationship $L_i \rightarrow L_j$ to *SR*;
End if
End for
End for

Fig. 7. An algorithm to discover structure relationships between hyperlinks in *HP*

4.3 Discover Access Relationships

4.3.1 Web Log Preprocessing

Access relationships among hyperlinks can be extracted from a web log. A raw web log collected from a web server needs to be preprocessed before meaningful access relationships can be

extracted [Cooley et al. 1999]. In this research, we divide the preprocessing task into two steps – web log cleaning and session identification.

In web log cleaning, two types of web log records are removed. First, web log records with the value of the status attribute greater than 400 are deleted because they record failed web access. Second, web log records recording accessory requests to a web page request, such as picture requests, are also removed. As a raw web log records every file request sent to a web server, one web access could result in several web log records. For example, an access to a web page with two pictures will result in three web log records, one for the web page and the other two for the pictures. Web log records recording accesses to web pages are sufficient to describe web surfers' information searching behaviors.

The basic processing unit for extracting access relationships is a session. A web log needs to be divided into sessions before the extraction of access relationships. However, in HTTP protocol, as connections between web clients and a web server are stateless, there is no notion of session in a web log. One method of dividing a web log into sessions is based on timeout. If the time between page requests from the same user exceeds a certain limit, it is assumed that the user has started another session [Cooley et al. 1999]. Some commercial tools use 30 minutes as a default timeout. Catledge and Pitkow calculated a timeout of 25.5 minutes based on empirical data [Catledge and Pitkow 1995]. Another method is to modify a web server to encode session identifiers in web pages transferred between clients and a web server [Yan et al. 1996]. As the second method requires modification of a web server, it is not convenient and practical for many websites. We adopt the timeout method to identify sessions. Web log records are first sorted by client IP address then by access time. We use a 30-minute timeout to divide web log records generated by the same IP address into sessions.

4.3.2 Mine Access Relationships From The Preprocessed Web Log

Once a raw web log has been cleaned and divided into sessions, association rule mining [Agrawal et al. 1993] can be applied to extract access relationships from it. Association rule mining is defined on a set of items $L = \{i_1, i_2, \cdots, i_k\}$. Let *D* be a set of transactions, where each transaction *T* is a set of items such that $T \subseteq L$. The support of an itemset (i.e., a set of items) in *D* is the fraction of all transactions containing the itemset. An itemset is called large if its support is greater than a user-specified support threshold. The most important step in association rule mining is to find large itemsets and their supports. Various algorithms for finding large itemsets, such as Apriori [Agrawal and Srikant 1994], are in use.

In the case of mining access relationships, sessions correspond to transactions, hyperlinks correspond to items and hyperlink sets correspond to itemsets. Applying the Apriori algorithm on the preprocessed web log, all hyperlink sets that have access relationships among their elements can be found and their corresponding supports can be calculated. The Apriori algorithm can be found in [Agrawal and Srikant 1994] and is skipped in this paper. The output of mining access relationships is a set of pairs denoted as *AR*, in which, each pair contains a hyperlink set and its corresponding support.

4.4 Calculate Preferences of Hyperlinks

As discussed in section 4.1, in hyperlink selection, the preference of a hyperlink is determined by two factors: the number of structure relationships it participates in as the initial hyperlink and the qualities of these structure relationships measured as supports of access relationships between it and its terminal hyperlinks. Based on these two factors, we define the preference of a hyperlink as below.

Definition 11: For a hyperlink L_i ,where $L_i \in HP$, and L_i participates in M structure relationships as the initial hyperlink, $(L_i \to L_{j_m}) \in SR$, for $m = 1,2,\cdots M$, the preference of the hyperlink L_i is defined as:

$$PRE_{L_i} = \sum_{m=1}^{M} (\sigma(\{L_i, L_{j_m}\}) * coeff) \qquad (9)$$

where $coeff = \begin{cases} 1 & \text{if } (\{L_i, L_{j_m}\}, \sigma(\{L_i, L_{j_m}\})) \in AR \\ 0 & \text{otherwise} \end{cases}$

Example 7: Given a hyperlink pool $HP = \{L_1, L_2, L_3, L_4, L_5, L_6, L_7\}$, structure relationships SR and access relationships AR,

$$SR = \{L_1 \to L_2, L_1 \to L_3, L_1 \to L_4, L_2 \to L_5, L_3 \to L_4, L_3 \to L_6, L_4 \to L_5, L_5 \to L_6,$$
$$L_5 \to L_7, L_6 \to L_3, L_7 \to L_3, L_7 \to L_4\}$$
$$AR = \{ (\{L_1, L_2\}, 0.022), (\{L_1, L_3\}, 0.018), (\{L_1, L_4\}, 0.01), (\{L_1, L_5\}, 0.002), (\{L_1, L_6\}, 0.014),$$
$$(\{L_1, L_7\}, 0.014), (\{L_2, L_5\}, 0.007), (\{L_2, L_6\}, 0.006), (\{L_2, L_7\}, 0.008), (\{L_3, L_6\}, 0.008),$$
$$(\{L_3, L_7\}, 0.012), (\{L_4, L_5\}, 0.01), (\{L_5, L_6\}, 0.03), (\{L_6, L_7\}, 0.005), (\{L_1, L_6, L_7\}, 0.003)\}$$

Hyperlink L_1 participates in three structure relationships as the initial hyperlink. According to (9),

$$PRE_{L_1} = \sum_{m=1}^{3} (\sigma(\{L_1, L_{j_m}\}) * coeff) = \sigma(\{L_1, L_2\}) * 1 + \sigma(\{L_1, L_3\}) * 1 + \sigma(\{L_1, L_4\}) * 1$$

$$= 0.022 + 0.018 + 0.01 = 0.05$$

Similarly, we get, $PRE_{L_2} = 0.007, PRE_{L_3} = 0.008, PRE_{L_4} = 0.01, PRE_{L_5} = 0.03, PRE_{L_6} = 0.008$ and $PRE_{L_7} = 0.012$. In this example, hyperlinks with the highest and the second highest preference value represent two types of important hyperlinks – hub hyperlink and hot hyperlink. Hub hyperlink, such as hyperlink L_1, is a hyperlink that has structure relationships with many other hyperlinks. Hot hyperlink, such as hyperlink L_5, is the starting point of a popularly visited path. The algorithm to compute preferences of hyperlinks is trivial and skipped in this paper.

4.5 Calculate Preferences of Hyperlink Pairs

As discussed in section 4.1, hyperlink pairs with group II relationship are preferred to hyperlink pairs without in hyperlink selection. For a hyperlink pair with group II relationship, the support of the access relationship between its hyperlinks is the determining factor for its preference over other hyperlink pairs with group II relationship in hyperlink selection. We set preferences of hyperlink pairs without group II relationship to be 0 and those of hyperlink pairs with group II relationship to be the supports of the access relationships between its hyperlinks.

Both hyperlink pairs with group II relationship and their preferences can be extracted from access relationships *AR*. For a 2-*HS* in *AR*, if there is no structure relationship between its elements, then the 2-*HS* is a hyperlink pair with group II relationship and its support σ(*2-HS*) is the preference of the hyperlink pair. We denote *PHP* as a set of pairs, in which, each pair consists of a hyperlink pair with group II relationship and the preference of the hyperlink pair.

Example 8: For a 2-*HS* $\{L_1, L_5\}$ in access relationships *AR* given in Example 7, since there is no structure relationship between hyperlinks L_1 and L_5, $\{L_1, L_5\}$ and its support become an element of *PHP* shown below. Preferences of hyperlink pairs outside *PHP* are set to be 0.

$$PHP = \{\ (\{L_1, L_5\}, 0.002), (\{L_1, L_6\}, 0.014), (\{L_1, L_7\}, 0.014), (\{L_2, L_6\}, 0.006), (\{L_2, L_7\}, 0.008), (\{L_6, L_7\}, 0.005)\}$$

Discussions on hyperlink pairs with group II relationship can be extended to include hyperlink sets with more than 2 elements.

Definition 12: A hyperlink set *k-HS*, where $k \geq 2$, is a hyperlink set with group II relationship if and only if ,

- there exists an access relationship among its hyperlinks, and
- there is no structure relationship between any pair of its hyperlinks.

Similarly, we set preferences of hyperlink sets without group II relationship to be 0. And, the preference of a hyperlink set with group II relationship is set to be the support of the access relationship among its hyperlinks. We denote *PHS* as a set of pairs, in which, each pair consists of a hyperlink set with group II relationship and the preference of the hyperlink set.

Example 9: *PHS* extracted from access relationships *AR* in Example 7 is,

$$PHS = \{ (\{L_1, L_5\}, 0.002), (\{L_1, L_6\}, 0.014), (\{L_1, L_7\}, 0.014), (\{L_2, L_6\}, 0.006), (\{L_2, L_7\}, 0.008), (\{L_6, L_7\}, 0.005), (\{L_1, L_6, L_7\}, 0.003)\}$$

According to Definition 12, hyperlink set $\{L_1, L_6, L_7\}$ is added to *PHS*. Preferences of hyperlink sets outside *PHS* are set to be 0.

The algorithm to extract *PHS* from access relationships *AR* is trivial and skipped in this paper.

4.6 Cluster Hyperlinks

In this section, a clustering algorithm is developed to extract *N* hyperlinks from the given hyperlink pool based on preferences of individual hyperlinks (i.e., PRE_{L_i}) and those of hyperlink sets (i.e. *PHS*). Clustering is a well researched area. In [Jain and Dubes 1988; Jain et al. 1999], clustering algorithms are divided into two groups – hierarchical approaches and partitional approaches. As partitional approaches need to specify the number of output clusters, they are not suitable for hyperlink clustering. The hyperlink clustering we proposed is a hierarchical approach. However, classical hierarchical clustering algorithms [Jain and Dubes 1988; Jain et al. 1999], such as single-link and complete-link algorithms, are not suitable for hyperlink clustering because of the following limitations:

(1) Classical hierarchical clustering algorithms are based on a similarity matrix whose indexes are objects to be clustered and whose elements are pairwise similarities between these objects. In the case of hyperlink clustering, hyperlinks correspond to objects and preferences of hyperlink pairs correspond to similarities between objects. In this classical framework, two additional attributes essential to hyperlink clustering have not been considered:

(1a) weights of objects -- preferences of hyperlinks in hyperlink clustering;

(1b) similarities among three or more objects -- preferences of hyperlink sets with more than two hyperlinks in hyperlink clustering.

(2) Classical hierarchical clustering algorithms pick either the maximum of the similarities between all pairs of objects drawn from a pair of clusters (one element from each cluster) (i.e., single-link algorithm) or the minimum of the similarities between all pairs of objects drawn from a pair of clusters (i.e., complete-link algorithm) as the similarity between the two clusters. However, both minimum and maximum methods could bias the actual similarity between clusters.

To include preferences of hyperlink sets with more than two hyperlinks, our hyperlink clustering algorithm is based on a similarity matrix whose indexes are hyperlink clusters and whose elements are similarities between hyperlink clusters. Initially, each hyperlink in the given hyperlink pool is placed in a unique hyperlink cluster. As hyperlink clusters are merged, the similarity matrix is updated.

To address limitation (1a), we define the similarity between two hyperlink clusters $C_i = \{L_{i_1}, L_{i_2}, \cdots, L_{i_P}\}$ and $C_j = \{L_{j_1}, L_{j_2}, \cdots, L_{j_Q}\}$ to include two components: preferences of individual hyperlinks in C_i and C_j and preferences of hyperlink sets whose elements are selected from C_i and C_j. When calculating preferences of individual hyperlinks in C_i and C_j, we use

average preference of hyperlinks in C_i and C_j, $\frac{\sum_{p=1}^{P} PRE_{L_{i_p}} + \sum_{q=1}^{Q} PRE_{L_{j_q}}}{P+Q}$, instead of sum of preferences of hyperlinks in C_i and C_j, $\sum_{p=1}^{P} PRE_{L_{i_p}} + \sum_{q=1}^{Q} PRE_{L_{j_q}}$, to avoid the possibility that the number of hyperlinks in clusters could impact the similarity between them. To address limitation (1b), we consider preferences of hyperlink sets *k-HS*s (i.e., $k \geq 2$, u elements from C_i and (k-u) elements from C_j, where $u \geq 1$ and $(k-u) \geq 1$), when calculating the similarity between hyperlink clusters C_i and C_j. These hyperlink sets can be grouped into several categories based on the value of k. For each category, we use the average preference of hyperlink sets, average $(\sigma(k\text{-HS}))$, instead of the minimum or the maximum preference of hyperlink sets (i.e., limitation 2).

Definition 13: For any two hyperlink clusters $C_i = \{L_{i_1}, L_{i_2}, \cdots, L_{i_P}\}$ and $C_j = \{L_{j_1}, L_{j_2}, \cdots, L_{j_Q}\}$ where $P \geq 1$ and $Q \geq 1$,

$$sim_{C_i,C_j} = \frac{\sum_{p=1}^{P} PRE_{L_{i_p}} + \sum_{q=1}^{Q} PRE_{L_{j_q}}}{P+Q} + \sum(\text{average}(\sigma(k\text{-HS}))) \qquad (10)$$

where *k-HS* is a hyperlink set with u elements selected from C_i and (k-u) elements selected from C_j, $k \geq 2, u \geq 1$ and $(k-u) \geq 1$; and $(k\text{-HS}, \sigma(k\text{-HS})) \in PHS$.

Example 10: Given preferences of hyperlinks and hyperlink sets calculated in Example 7 and Example 9, for two hyperlink clusters $C_1 = \{L_1, L_2\}$ and $C_2 = \{L_6, L_7\}$, according to (10), sim_{C_1,C_2} includes two parts,

preferences of hyperlinks in C_1 and C_2: $\frac{PRE_{L_1} + PRE_{L_2} + PRE_{L_6} + PRE_{L_7}}{2+2} = 0.01925$

preferences of hyperlink sets whose elements are selected from C_1 and C_2:

$$\text{average } (\sigma(2\text{-HS})) = \frac{\sigma(\{L_1, L_6\}) + \sigma(\{L_1, L_7\}) + \sigma(\{L_2, L_6\}) + \sigma(\{L_2, L_7\})}{4} = 0.0105$$

$$\text{average } (\sigma(3\text{-HS})) = \frac{\sigma(\{L_1, L_6, L_7\})}{1} = 0.003$$

Hence, sim_{C_1,C_2} =0.01925+0.0105+0.003=0.03275.

For the hyperlink pool given in Example 7, seven hyperlinks are initially placed in seven unique hyperlink clusters. Similarities between hyperlink clusters can be computed and a 7×7 initial similarity matrix is created as below. Indexes of the matrix (i.e., 1,2,3,4,5,6,7) represent hyperlink clusters $\{L_1\},\{L_2\},\{L_3\},\{L_4\},\{L_5\},\{L_6\}$and $\{L_7\}$ respectively.

$$\begin{bmatrix} 0 & 0.0285 & 0.029 & 0.03 & 0.042 & 0.043 & 0.045 \\ 0.0285 & 0 & 0.0075 & 0.0085 & 0.0185 & 0.0135 & 0.0175 \\ 0.029 & 0.075 & 0 & 0.009 & 0.019 & 0.008 & 0.01 \\ 0.03 & 0.0085 & 0.009 & 0 & 0.02 & 0.009 & 0.011 \\ 0.042 & 0.0185 & 0.019 & 0.02 & 0 & 0.019 & 0.021 \\ 0.043 & 0.0135 & 0.008 & 0.009 & 0.019 & 0 & 0.015 \\ 0.045 & 0.0175 & 0.01 & 0.011 & 0.021 & 0.015 & 0 \end{bmatrix}$$

Once the initial similarity matrix has been created, hyperlink clusters with the highest similarity, $\{L_1\}$ and $\{L_7\}$, are merged into one hyperlink cluster $\{L_1, L_7\}$. A 7×7 similarity matrix changes to a 6×6 similarity matrix and similarities related to the merged hyperlink cluster (i.e., $\{L_1, L_7\}$) are re-computed. The updated similarity matrix is listed below and indexes of the matrix (i.e., 1,2,3,4,5,6) represent hyperlink clusters $\{L_1, L_7\},\{L_2\},\{L_3\},\{L_4\},\{L_5\}and\{L_6\}$ respectively.

$$\begin{bmatrix} 0 & 0.031 & 0.0233 & 0.024 & 0.0237 & 0.0328 \\ 0.031 & 0 & 0.0075 & 0.0085 & 0.0185 & 0.0135 \\ 0.0233 & 0.075 & 0 & 0.009 & 0.019 & 0.008 \\ 0.024 & 0.0085 & 0.009 & 0 & 0.02 & 0.009 \\ 0.0237 & 0.0185 & 0.019 & 0.02 & 0 & 0.019 \\ 0.0328 & 0.0135 & 0.008 & 0.009 & 0.019 & 0 \end{bmatrix}$$

Based on the updated similarity matrix above, hyperlink clusters $\{L_1, L_7\}$ and $\{L_6\}$ can be merged and the similarity matrix is updated again. This process is repeated until the number of hyperlinks in a hyperlink cluster is greater than or equal to the number of hyperlinks to be placed in the portal page *N*. This cluster contains all hyperlinks to be placed in the portal page[7]. We summarize the hyperlink clustering algorithm in Fig. 8.

```
Input:  HP: the hyperlink pool of a website w
        PRE_i : the preference of a hyperlink i in HP, for i = 1,2,...,|HP|
        PHS : hyperlink sets and their preferences
        N: the number of hyperlinks to be placed in the portal page of w
Output: PHL : the set of hyperlinks in the portal page of w

distribute each hyperlink in HP into an unique hyperlink cluster;
max_size = 1;  /*max_size: maximum number of hyperlinks in hyperlink clusters*/
initialize SIM;  /*SIM: similarity matrix*/
While (max_size < N)
        merge hyperlink clusters with the highest similarity into C;  /*C: merged cluster*/
        If (size(C) > max_size)  /*size(C): number of hyperlinks in C*/
                max_size = size(C);
        End if
        re-compute SIM ;  /*only need to re-compute similarities related to
                                the merged cluster*/
End while
PHL = C;
```

Fig. 8. The hyperlink clustering algorithm

4.7 TIME COMPLEXITY OF LINKSELECTOR

To compute time complexity of LinkSelector, time complexity for each step in the algorithm is considered first. We denote the number of hyperlinks in the given hyperlink pool *HP* as n_l, the number of sessions in a web log as n_s, the number of elements in access relationships *AR* as n_{AR}

[7] If the number of hyperlinks in the result hyperlink cluster is larger than the number of hyperlinks to be placed in a portal page (*N*), then *N* hyperlinks are selected from the cluster according to their individual preferences, from high to low.

and the number of elements in structure relationships *SR* as n_{SR}. Structure relationships are discovered in step 1. In this step, pages pointed to by all hyperlinks in *HP* are retrieved and parsed to extract structure relationships. If the average number of hyperlinks in all retrieved pages is β, time complexity for step 1 is O($n_l \times \beta$). Usually, β is much less than n_l. Hence, time complexity for step 1 is O(n_l). We employ the Apriori algorithm to discover access relationships *AR*[8] in step 2. Apriori goes through all sessions in a web log α rounds to discover access relationships *AR*. Therefore, time complexity for step 2 is O($n_s \times \alpha$). As α is much less than n_s (e.g., in our experiment discussed in Section 5, α equals to 6 while n_s is 262K), time complexity for step 2 is the order of O(n_s). Preferences of hyperlinks are computed in step 3. If the average number of structure relationships a hyperlink participates in as the initial hyperlink is γ, time complexity for step 3 is O($n_l \times \gamma$). Usually, γ is much less than n_l. Hence, time complexity for step 3 is O(n_l). In step 4, each hyperlink set in access relationships *AR* is checked to see whether there is structure relationship between its elements. Hence, time complexity for step 4 O(n_{AR}). One $n_l \times n_l$ similarity matrix is created in step 5, which needs O(n_l^2) time. At worst, it also needs O(n_l^2) time to re-compute the similarity matrix. Hence, time complexity for step 5 is the order of O(n_l^2). Combining time complexities of all 5 steps, we get the time complexity of LinkSelector as O($n_l^2 + n_s + n_{AR}$).

5. EXPERIMENT RESULTS

In this section, we compare LinkSelector with other hyperlink selection approaches. The data is described in section 5.1 and a portal page constructed using LinkSelector is presented in section 5.2. We compare LinkSelector with the current practice of hyperlink selection in section 5.3. In

[8] We assume that web log is cleaned and sessions are identified before applying LinkSelector.

section 5.4, we also compare LinkSelector with classical hierarchical clustering and with association rule mining -- a popular data mining method used in the web usage mining field.

5.1 DATA

We did our experiments on the University of Arizona website because it is large enough and generates enough web logs to permit comparisons of different hyperlink selection approaches. One month's web log recently collected from the university web server was used in our experiments. The raw web log contained 10 million records. After web log cleaning, the cleaned web log had 4.2 million records, enough to extract web visiting patterns from. We used a 30-minute timeout to divide the cleaned web log into 344K sessions. 23 days of the cleaned web log were used as the training data, which contained 262K sessions; 7 days of the cleaned web log were used as the testing data, which contained 82K sessions.

The hyperlink pool consists of 743 hyperlinks in the index page of the university website (http://www.arizona. edu/index/webindex.shtml). Among these, 110 hyperlinks pointed to web pages at the university web server and the remaining 633 hyperlinks pointed to web pages at other web servers. As the web log was collected from the university web server, access relationships among the 633 hyperlinks pointing to web pages at other web servers could not be mined from the collected web log. Therefore, the hyperlink pool used in our experiments consisted of the 110 hyperlinks pointing to web pages at the university web server.

5.2 RESULTS OF LINKSELECTOR

Applying LinkSelector to the hyperlink pool and the training data, preferences of hyperlinks and hyperlink sets were calculated. We list hyperlinks with top ten preference values in table II.

Table II. Hyperlinks in the Hyperlink Pool with Top 10 Preference Values

Hyperlink	Preference
/index/alldepts-index.shtml	0.075193358
/shared/sports-entertain.shtml	0.067539082
/working/teaching.shtml	0.032591667
/shared/aboutua.shtml	0.023351642
/newschedule/parse-schedule-new.cgi	0.015668777
/student_link	0.015668777
/spotlight/index.shtml	0.012788894
/shared/getting-around.shtml	0.008771161
/shared/athletics.shtml	0.008755914
/shared/tours.shtml	0.008081209

In table II, the hyperlink with the highest preference value(i.e., /index/alldepts-index.shtml) is a hub hyperlink that has structure relationships with hyperlinks pointing to homepages of departments, programs and colleges at the university. The hyperlink with the second highest preference value (i.e., /shared/sports-entertain.shtml) is a hot hyperlink which is the starting point of a popularly visited path leading to web pages on sports and entertainments at the university. Clustering hyperlinks based on preferences of hyperlinks and hyperlink sets resulted in the 10 hyperlinks selected for the university portal pages. These hyperlinks are listed in table III.

Table III. Result of LinkSelector (N=10)

No.	Hyperlink
1	/index/alldepts-index.shtml
2	/shared/sports-entertain.shtml
3	/working/teaching.shtml
4	/shared/aboutua.shtml
5	/shared/getting-around.shtml
6	/spotlight/index.shtml
7	/newschedule/parse-schedule-new.cgi
8	/student_link
9	/phonebook
10	/academic/oncourse/data/interface

5.3 PERFORMANCE COMPARISON WITH EXPERT SELECTION AND TOP-LINK SELECTION

Current practice of hyperlink selection, namely expert selection, relies on domain experts' (e.g., designers of a website) experiences. Hyperlinks contained in the current portal page

(http://www.arizona.edu) of the university web site are hyperlinks selected by domain experts. Another simple approach to hyperlink selection, namely top-link selection, is to select hyperlinks with top *N* access frequency, where *N* is the number of hyperlinks to be placed in a portal page. Table IV lists 6 hyperlinks selected by LinkSelector, domain experts[9] and top-link selection, respectively.

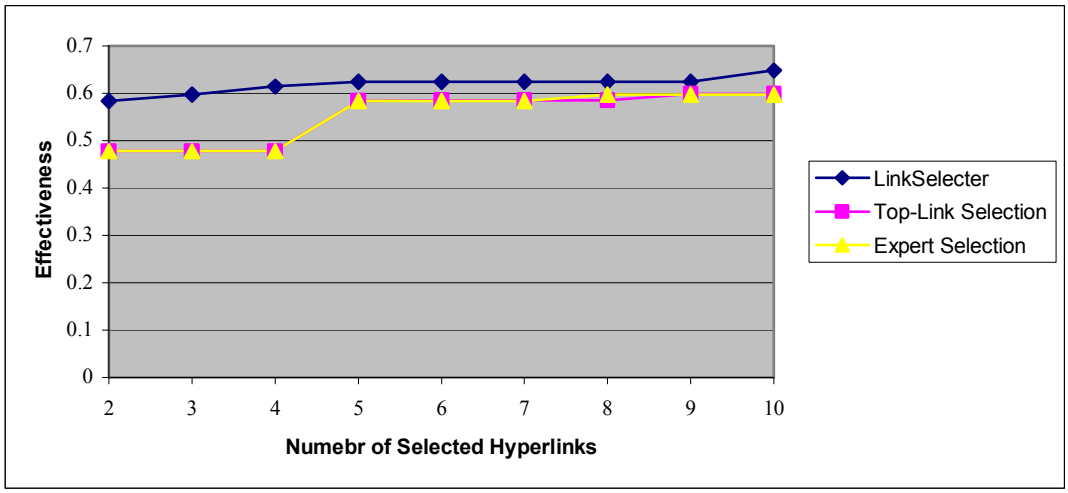
Table IV. Hyperlinks Selected by LinkSelector, Domain Experts and Top-link Selection (*N*=6)

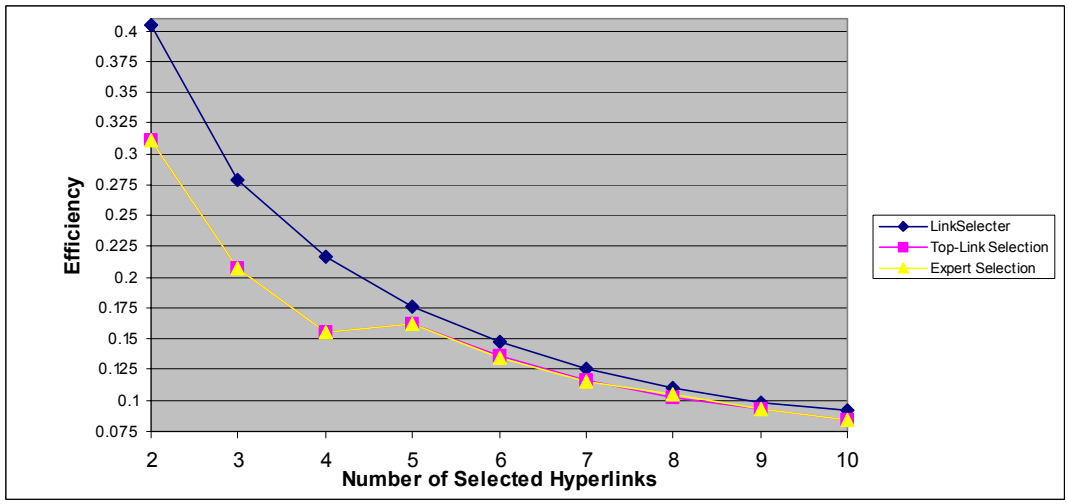
No.	LinkSelector	Expert Selection	Top-Link Selection
1	/index/alldepts-index.shtml	/student_link	/student_link
2	/shared/sports-entertain.shtml	/index/alldepts-index.shtml	/index/alldepts-index.shtml
3	/working/teaching.shtml	/newschedule/parse-schedule-new.cgi	/newschedule/parse-schedule-new.cgi
4	/shared/aboutua.shtml	/phonebook	/phonebook
5	/shared/getting-around.shtml	/shared/sports-entertain.shtml	/shared/sports-entertain.shtml
6	/spotlight/index.shtml	/shared/libraries.shtml	/shared/athletics.shtml

As shown in table IV, hyperlinks selected by domain experts and those selected by top-link selection overlapped significantly. This was the result of the cross reinforcement effect – hyperlinks placed in the portal page were likely to be visited and domain experts were likely to place hyperlinks with high access frequency in the portal page. However, hyperlinks selected by LinkSelector differed from those selected by domain experts and top-link selection significantly. In the example shown in Table IV, four of the six hyperlinks selected by LinkSelector differed from the expert/top-link selections.

Using the training data, 9 portal pages with the number of hyperlinks increasing from 2 to 10 were constructed by each of the three hyperlink selection approaches. Qualities of these portal pages were compared using the testing data, as shown in Fig. 9.

[9] k hyperlinks selected by domain experts are k hyperlinks chosen from the portal page designed by domain experts and these chosen hyperlinks are top-k frequently accessed hyperlinks among all hyperlinks in the portal page.





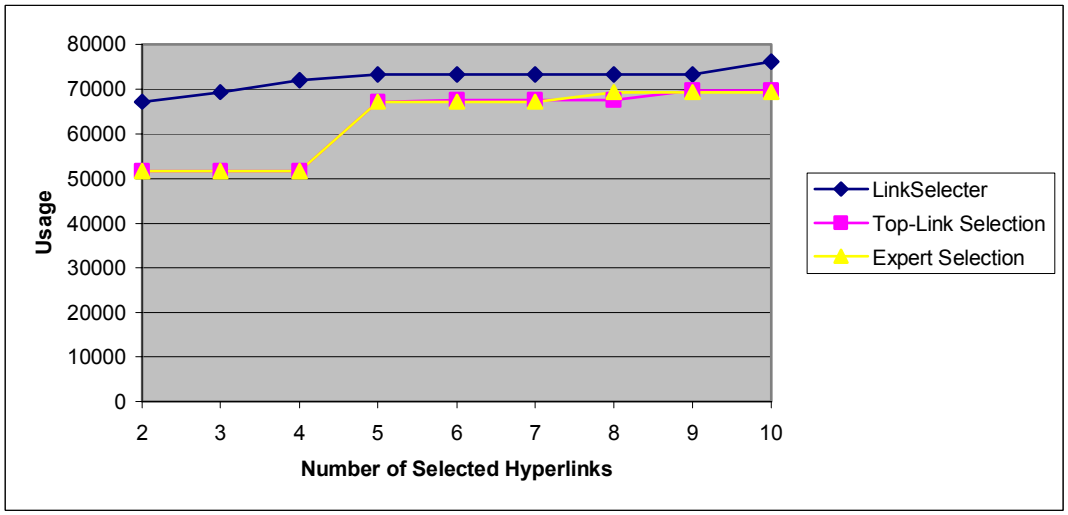


Fig. 9. Quality comparison among portal pages constructed by LinkSelector, expert selection and top-link selection

On average, LinkSelector outperformed both expert selection and top-link selection with a 12.7% increase in effectiveness. Given the large number of visiting sessions (e.g., 11.5k sessions per day at the website of the University of Arizona[10]), this is a big improvement in ease of finding user-sought top level web pages. The improvement decreased from 22.1% to 8.4% as the selection ratio (i.e., the ratio of the number of the selected hyperlinks over the total number of hyperlinks in a hyperlink pool) increased from 1.8% (i.e., 2/110) to 9.1% (i.e., 10/110). However, even at the selection ratio of 9.1%, which is more than two times of the selection ratio for the portal page of the University of Arizona website (i.e., 32/743=4.3%), the improvement in effectiveness (i.e., 8.4%) was still quite apparent. Compared with expert selection and top-link selection, LinkSelector improved efficiency by 16.9% on average. Similarly, the improvement for efficiency decreased from 30.2% to 9.3% as the selection ratio increased. Compared with expert selection and top-link selection, LinkSelector also improved usage by an average of 17.0%. The improvement decreased from 30.2% to 9.4% as the selection ratio increased. Improvement in usage indicated that the portal page constructed by LinkSelector had potential to attract more visits than portal pages generated by domain experts and top-link selection.

The improvements were attributed to the relationships among hyperlinks considered in LinkSelector but missed in the other two approaches. For example, hyperlinks "/shared/sports-entertain.shtml" and "/shared/athletics.shtml" were on a popularly visited path to web pages on sports and entertainments. The first hyperlink was the starting point of the path and the second one was the second link on this path. Therefore, both of them had high access frequency and were selected by top-link selection. However, top-link selection failed to consider that there was a structure relationship between these two hyperlinks. In this structure relationship, hyperlink "/shared/sports-entertain.shtml" was the initial hyperlink and hyperlink "/shared/athletics.shtml" was the terminal hyperlink. It was natural to navigate from "/shared/sports-entertain.shtml" to "/shared/athletics.shtml" and find web pages on sports and entertainments. Hence, it was

[10] This is a conservative estimation without considering web logs cached at proxy servers.

unnecessary to put both of them in a portal page. Applying group I relationship, LinkSelector selected only the starting point of the path -- "/shared/sports-entertain.shtml". Both top-link selection and expert selection failed to consider group II relationships among hyperlinks (i.e., hyperlinks that are structurally unrelated but access related). For example, hyperlinks "/shared/sports-entertain.shtml" and "/shared/aboutua.shtml" were structurally unrelated hyperlinks. However, a large number of sessions (i.e., 0.2% of the training sessions) looking for information regarding sports and entertainment at the university (i.e.,/shared/sports-entertain.shtml) also tried to learn something about the university (i.e., /shared/aboutua.shtml). Placing both hyperlinks in a portal page could save web surfers' efforts of finding the path from one topic to the other. Applying group II relationship, LinkSelector selected both hyperlinks.

5.4 PERFORMANCE COMPARISON WITH DATA MINING METHODS

To show the limitations of classical hierarchical clustering in hyperlink selection, we compared it with LinkSelector. We also compared LinkSelector with one popular data mining method used in the web log mining field – association rule mining. Classical hierarchical clustering algorithms [Jain and Dubes 1988; Jain et al. 1999] selected hyperlinks based only on preferences of hyperlink pairs. Apriori [Agrawal and Srikant 1994], an association rule mining algorithm, was applied to find the *N*-hyperlink set with the highest support among all *N*-hyperlink sets, where *N* is the number of hyperlinks to be placed in a portal page. And, the discovered *N*-hyperlink set was the result of hyperlink selection. Table V lists 6 hyperlinks selected by LinkSelector, classical hierarchical clustering and association rule mining, respectively.

Table V. Hyperlinks Selected by LinkSelector, Classical Hierarchical Clustering and Association Rule Mining (N=6)

No.	LinkSelector	Classical Hierarchical Clustering	Association Rule Mining
1	/index/alldepts-index.shtml	/phonebook	/index/alldepts-index.shtml
2	/shared/sports-entertain.shtml	/student_link	/shared/aboutua.shtml
3	/working/teaching.shtml	/newschedule/parse-schedule-new.cgi	/shared/getting-around.shtml
4	/shared/aboutua.shtml	/academic/oncourse/data/interface	/shared/libraries.shtml
5	/shared/getting-around.shtml	/index/alldepts-index.shtml	/working/research.shtml
6	/spotlight/index.shtml	/working/teaching.shtml	/working/teaching.shtml

Using the training data, 5 portal pages with the number of hyperlinks increasing from 2 to 6[11] were constructed by each of the three hyperlink selection approaches. Qualities of these portal pages were compared using the testing data, as shown in Fig. 10.

[11] As the largest hyperlink set discovered using association rule mining only has 6 hyperlinks, we use 6 as the maximal number of hyperlinks in comparing qualities of portal pages.

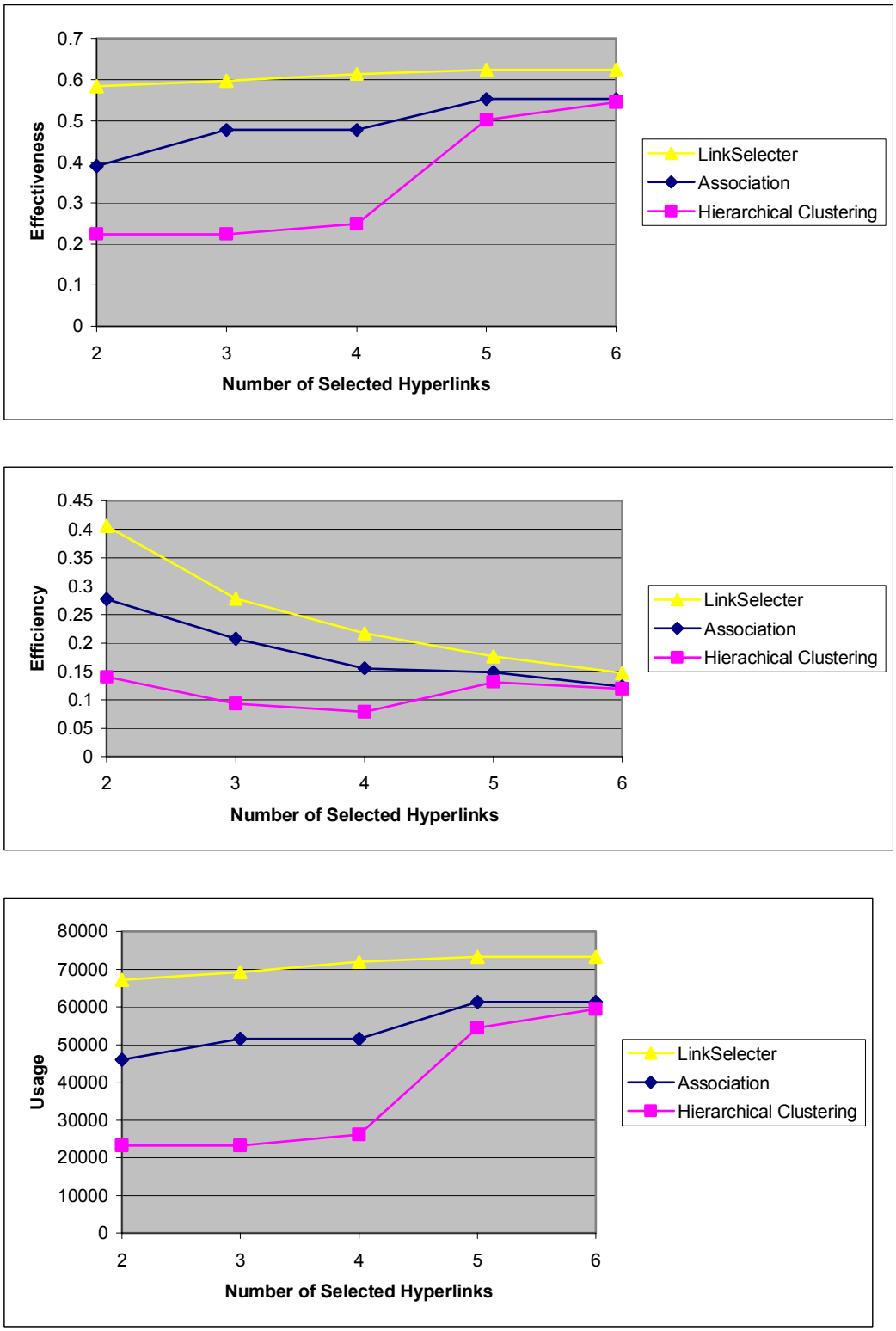


Fig. 10. Quality comparison among portal pages constructed by LinkSelector, classical hierarchical clustering and association rule mining

Compared with association rule mining and classical hierarchical clustering, LinkSelector improved effectiveness by an average of 25.8% and 102.0%, efficiency by an average of 31.7% and 124.0% and usage by an average of 31.6% and 123.0% respectively. Hyperlinks selected using association rule mining were hyperlinks with the highest co-occurrence frequency. However, association rule mining failed to address hyperlinks individually (i.e., preferences of hyperlinks). Moreover, structure relationships between hyperlinks were not considered in association rule mining. Therefore, hyperlinks with high preferences, such as "/shared/sports-entertain.shtml" , were missed in the result of association rule mining. These missing hyperlinks significantly decreased the quality of the portal page generated by association rule mining. Limitations of classical hierarchical clustering algorithms, as discussed in section 4.6, explain their bad performance in hyperlink selection. For example, failure to consider preferences of hyperlinks excluded hyperlinks with high preferences, such as hyperlinks "/index/alldepts-index.shtml" and "/shared/sports-entertain.shtml", from the results of hyperlink selection. This, in turn, reduced the quality of the generated portal page.

6. CONCLUSION AND FUTURE WORK

Hyperlink selection is an important but rarely researched problem. In this paper, we have formally defined the hyperlink selection problem and proposed a heuristic solution method named LinkSelector. The proposed method is based on relationships among hyperlinks – structure relationships extracted from an existing website and access relationships discovered from a web log. Preferences of hyperlinks and hyperlink sets are calculated from these relationships and a clustering algorithm is developed to extract hyperlinks from a given hyperlink pool using the preferences calculated. We compare LinkSelector with the current practice of hyperlink selection and top-link selection, using data obtained from the University of Arizona website. Using the same data, we also compare LinkSelector with two data mining methods – classical hierarchical

clustering and association rule mining. Results show that LinkSelector outperformed all these hyperlink selection approaches.

Future work is needed in two areas. First, we plan to conduct an empirical user study to examine the properties of different hyperlink selection approaches and compare their performances using the metrics proposed in section 3. Second, we plan to work on making LinkSelector adaptive to changes both in the structure of a website and in users' web visiting patterns. The former leads to changes in structure relationships and the latter causes changes in access relationships. As a result, hyperlinks selected by LinkSelector based on old structure relationships and access relationships could be out-of-date. To keep the selected hyperlinks up-to-date, an obvious solution is to re-run LinkSelector every time a change occurs. Apparently, for websites with frequent changes, the cost of frequent re-run is unbearable. An efficient solution needs to be developed to monitor both types of changes and to trigger the re-run of LinkSeletor only when necessary.

References:

Anderson, C., Domingos, P., AND Weld, D. 2001. Adaptive Web Navigation for Wireless Devices. *Proc. of the Seventeenth International Joint Conference on Artificial Intelligence.*

Agrawal, R., Imielinski, T., AND Swami, A. 1993. Mining Association Rules between Sets of Items in Large Database. *Proc. of the 1993 ACM SIGMOD.*

Agrawal, R., AND Srikant, R. 1994. Fast Algorithms for Mining Association Rules. *Proc. of the 20th VLDB Conference.*

Brin, S., AND Page, L. 1998. The Anatomy of a Large-scale Hypertextual Web Search Engines. *Proc. of the Seventh International World Wide Web Conference*.

Catledge, L. AND Pitkow, J. 1995. Characterizing Browsing Behaviors on the World Wide Web. *Computer Networks and ISDN Systems*, 27(6): 1065-1073.

Chakrabarti, J. et al. 1999. Mining the Web's Link Structure. *IEEE Computer*, 32(8): 60-67.

Chakrabarti, J. 2000. Data Mining for Hypertext: A Tutorial Survey. *ACM SIGKDD Explorations*, 1(2): 1-11.

Chen, M., Park, J., AND Yu, P. 1996. Data Mining for Path Traversal Patterns in A Web Environment. *Proc. 16th Intl. Conf. on Distributed Computing Systems*.

Cooley, R., Mobasher, B., AND Srivastava, J. 1997. Web Mining: Information and pattern discovery on the world wide web. *IEEE International Conference on Tools with AI*.

Cooley, R., Tan, P., AND Srivastava, J. 1999. WebSIFT: The Website Information Filter System. *Proc. of the Web Usage Analysis and User Profiling Workshop.*

Cooley, R., Mobasher, B., AND Srivastava, J. 1999. Data Preparation for Mining World Wide Web Browsing Patterns. *Knowledge and Information Systems* 1(1): 1-27.

Jain, A. K., AND Dubers, R. C. 1988. Algorithms for Clustering Data. Prentice-Hall Inc..

Jain, A. K., Murty, M. N., AND Flynn, P. J. 1999. Data Clustering: A Review. *ACM Computing Surveys*, 31(3): 264-323.

Kleinberg J. 1998. Authoritative Sources in a Hyperlinked Environment. *Proc. of ACM-SIAM Symposium on Discrete Algorithms.*

Kosala, R. AND Blockeel, H. 2000. Web Mining Research: A Survey. *ACM SIGKDD Explorations*, 2(1): 1-15.

Lawrence, S., AND Giles, C. L. 1998. Searching the World Wide Web. *Science* 280: 98-100.

Lawrence, S., AND Giles, C. L. 1999. Accessibility of Information on the Web. *Nature* 400: 107-109.

Lee, J., AND Podlaseck, M. 2001. Visualization and Analysis of Clickstream Data of Online Stores for Understanding Web Merchandising. *Data Mining and Knowledge Discovery* 5(1-2): 59-84.

Manber, U., Patel, A., AND Robison, J. 2000. Experience with Personalization on Yahoo!. *Communications of the ACM*, 43(8): 35-39.

Mobasher, B., Cooley, R., AND Srivastava J. 2001. Automatic Personalization Based on Web Usage Mining. *Communications of the ACM* 43(8): 142-151.

Nielson, J. 1999. User Interface Directions for the Web. *Communications of the ACM* 42(1): 65-72.

Nielson, J. and Wagner, A. 1996. User Interface Design for the WWW. *Proc. of ACM CHI 96.*

Perkowitz, M., AND Etzioni, O. 2000. Towards Adaptive Websites: Conceptual Framework and Case Study. *Artificial Intelligence,* 118(1-2): 245-275.

Spiliopoulou, M., AND Pohle, C. 2001. Data Mining to Measure and Improve the Success of Websites. *Data Mining and Knowledge Discovery* 5(1-2): 85-114.

Srivastava, J., Cooley, R., Deshpande, M., AND Tan, P. 2000. Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data. *SIGKDD Explorations*, 1(2), 1-12.

Yan, T., Jacobsen, M., Garcia-Molina, H., AND Dayal, U. 1996. From User Access Patterns to Dynamic Hypertext Linking. *Proc. of the 5th International World Wide Web Conference*.