The Emerald Research Register for this journal is available at
www.emeraldinsight.com/researchregister

The current issue and full text archive of this journal is available at
www.emeraldinsight.com/1066-2243.htm

Web page prediction from metasearch results

Lin-Chih Chen
Department of Information Management, National Taiwan University of Science and Technology, Taipei, Taiwan, and
Cheng-Jye Luh
Department of Information Management, Yuan-Ze University, Chung-Li, Taiwan

Abstract

Purpose – This study aims to present a new web page recommendation system that can help users to reduce navigational time on the internet.

Design/methodology/approach – The proposed design is based on the primacy effect of browsing behavior, that users prefer top ranking items in search results. This approach is intuitive and requires no training data at all.

Findings – A user study showed that users are more satisfied with the proposed search methods than with general search engines using hot keywords. Moreover, two performance measures confirmed that the proposed search methods out-perform other metasearch and search engines.

Research limitations/implications – The research has limitations and future work is planned along several directions. First, the search methods implemented are primarily based on the keyword match between the contents of web pages and the user query items. Using the semantic web to recommend concepts and items relevant to the user query might be very helpful in finding the exact contents that users want, particularly when the users do not have enough knowledge about the domains in which they are searching. Second, offering a mechanism that groups search results to improve the way search results are segmented and displayed also assists users to locate the contents they need. Finally, more user feedback is needed to fine-tune the search parameters including α and β to improve the performance.

Practical implications – The proposed model can be used to improve the search performance of any search engine.

Originality/value – First, compared with the democratic voting procedure used by metasearch engines, search engine vector voting (SVV) enables a specific combination of search parameters, denoted as α and β, to be applied to a voted search engine, so that users can either narrow or expand their search results to meet their search preferences. Second, unlike page quality analysis, the hyperlink prediction (HLP) determines qualified pages by simply measuring their user behavior function (UBF) values, and thus takes less computing power. Finally, the advantages of HLP over statistical analysis are that it does not need training data, and it can target both multi-site and site-specific analysis.

Keywords Search engines, Individual behaviour, Worldwide web, Information retrieval

Paper type Research paper

1. Introduction

Internet users generally hope that search engines can locate the exact required information. The primary aim of search engine design lies in finding data relevant to user queries. This task is generally recognized as difficult (Jansen *et al.*, 1998) because the inputs supplied by the users are generally insufficient for collecting suitable data. A search of previous literature showed that a user on an average enters 2.3 terms for a

Internet Research
Vol. 15 No. 4, 2005
pp. 421-446

1066-2243
DOI 10.1108/10662240510615182

given query (Spink *et al.*, 2001). Such a query generally produces thousands of results, so every search engine ranks the results by ranking algorithms (Li, 1998). Often the users are only concerned with the top ranking results, and ignore the rest (Sougné, 2000).

Search engines generally use information retrieval (IR) techniques to find web pages or documents relevant to a user query in a database of web pages. Traditional IR methods measure the similarity between user query and document contents by models including vector space models, probability models, and fuzzy logic models (Yates and Neto, 1999). These methods are primarily based on the keyword matches and their frequency in documents, and therefore easily result in "keyword spamming". That is, a document's rank can be manipulated by duplicating the same set of keywords several times in a document (Yates and Neto, 1999). However, search engines can now detect and penalize keyword spamming (Mall-Net, 2001).

To solve the keyword spamming problem, Li (1998) developed the HVV method. HVV uses a ranking scheme similar to that of science citation index (SCI), which ranks each web page by the number of times other web pages contain links to it. Thus, a web page would have a high weight if several hyperlinks point to it. However, this method does not distinguish between the ratings given by high-quality web pages from those given by low-quality web pages. Henzinger (2001) discussed the quality of hyperlinks in a PageRank algorithm, which is recursively applied to an arbitrary set of initial web pages and therefore needs considerable computing power to analyze the hyperlinks and page ranks (Sobek, 2002). As the web continues to grow at millions of pages per day (Chakrabarti *et al.*, 1999), many source web pages need to be collected, and many experiments need to be performed to demonstrate the effectiveness of this method.

Collecting search results from several search engines is an effective way to collect and rate relevant web pages (Selberg and Etzioni, 1997). This method is called metasearch (Dreilinger and Howe, 1996; Selberg and Etzioni, 1997). The metasearch approach saves a lot of time by searching only in one place and eliminating the need to use and learn several separate search engines. The quality of metasearch results depends on the search engines used and how they organize the results.

In this study, a metasearch method, called search engine vector voting (SVV), was developed to rate the search results obtained from several well-known search engines. The relevance of a web page to a given query is determined by its frequency and rankings in the search results returned by each search engine. A web page wins a vote from a search engine if it is listed on the top 50 items returned by a given query. The SVV method can solve the two problems encountered in using HVV stated above, since well-known search engines are generally recognized as high-quality web sites, and SVV only considers the top 50 matched web pages from each search engine. More importantly, the adoption of SVV would eliminate possible bias from any single search engine.

Web search engines generally respond to user queries with web page URLs and short descriptions. Users may browse through the URLs to reach the pages most relevant to their queries. Therefore, a search engine can significantly save users' time by directly providing the web pages and their embedded links most relevant to user queries. Nick and Themis (2001) implemented an intelligent agent system that uses a genetic algorithm to recommend web pages directly to users. To help the intelligent agent in learning a user's interests, the user is requested to provide some web pages

examples of interest in advance. This task is very hard for novice users, who may give inappropriate examples resulting in unacceptable outcomes. Thus, a better approach is required to solve the example input problem.

A hyperlink prediction (HLP) method was then developed to recommend web pages referred from the URLs listed in SVV results to the users. HLP is based on the assumption that the users would find more interesting pages by following many successive hyperlinks on high-ranking web pages in SVV results.

The rest of this study is organized as follows. Section 2 introduces some related work. Section 3 discusses SVV and HLP in detail. Section 4 presents three experiments using the proposed search methods. The first experiment was a user study on the quality of search results for some hot keywords. The second experiment compared SVV with the four best metasearch search engines. The third experiment compared SVV and HLP with six well-known search engines. Finally, Section 5 concludes this study and discusses future research.

2. Related work

This section reviews some important literature related to this study. First, we provide a literature review on metasearch. Second, we discuss some related work on hyperlink exploration.

2.1 Metasearch

Unlike general search engines (Dreilinger and Howe, 1996; Selberg and Etzioni, 1997; Meng *et al.*, 2002; Zacharis and Panayiotopoulos, 2002; Hai *et al.*, 2004), metasearch engines transmit the keywords submitted in its search box simultaneously to several search engines and present the search results in an integrated format. The format lets the users see at a glance which particular search engine returned the most relevant web pages retrieved for a query without having to search each engine individually. Such finding could be used to adjust the rank order of results. Thus, metasearch can significantly save searching time and eliminate the need to use and learn several separate search engines (Hu *et al.*, 2001). The quality of metasearch results depends on which search engines they search and how they integrate the results.

Several related studies have been conducted on metasearch engines. Lawrence and Giles (1998) proposed a NECI metasearch engine, which can analyze each downloaded document and then display results with the query term shown in a specific context. This format helps users readily determine whether the document is relevant without having to download each document. Glover *et al.* (1999a) described a metasearch engine architecture which customizes the searching and results ranking strategies based on the user's information need. Compared with a regular metasearch engine, which sends a query to a pre-defined list of search engines and ranks the results in a pre-defined order, this method allows much greater personalization by providing customized ranking of search results from a tailored list of search engines. Svidzinska (2001) proposed a two-tier metasearch engine. The first tier collects the required information about topic-specific search engines in advance. The second tier then expands and routes the user queries to a subset of the selected topic-specific search engines through a routing mechanism. Zacharis and Panayiotopoulos (2002) proposed a Webnaut metasearch system, which can learn the user's interests and adapt them appropriately as these interests change over time. The learning process uses a genetic

algorithm along with the user feedback to an intelligent agent's filtered selections. Osdin *et al.* (2002) presented a metasearch method, called HuddleSearch, using a newly developed clustering algorithm, which dynamically organizes the relevant documents into a traversable hierarchy from general to specific cluster categories. KartOO (2004) is a commercial metasearch engine with visual display interfaces. KartOO shows the results with sites interconnected by keywords, and also presents a thematic map showing the most important sites and the linkage relationships among the various results. Braslavski *et al.* (2004) presented ProThes, which consists of a metasearch engine, a graphical user interface for query specification, and a thesaurus-based query customization system. ProThes also provides simple heuristics for result merging and partial re-ranking. Hai *et al.* (2004) provide an overview of techniques for extracting information from the web search interfaces of e-commerce search engines, which is useful for constructing e-commerce metasearch engines. Hai *et al.* (2004) also presented a tool that can automatically build a unified search interface over multiple heterogeneous e-commerce search engines in the same product domain.

2.2 Hyperlink exploration

Hyperlink exploration is a way of analyzing the hyperlink structures among web pages on the internet. The hyperlink analysis results can in turn be used to measure the quality of a web page. That is, pages pointed to by many other pages are recognized as high quality. Several researchers have studied new methods to resolve this issue (Brin and Page, 1998; Henzinger, 2001). These methods recursively analyze the hyperlinks and compute the ranks for all the pages starting from a set of source web pages. These methods need huge computing power to be effective (Sobek, 2002).

Several related studies have used stochastic methods, such as Markov chains, for hyperlink structure analysis and prediction. Lempel and Moran (2000) presented a new stochastic method for hyperlink structure analysis, and claimed that it can discover the most authoritative sites effectively and efficiently. Sarukkai (2000) used Markov chains to predict the probability of seeing a link in the future given navigation logs. Chen *et al.* (2002) proposed a two-phase method to predict a user's next access at the category level by analyzing user access patterns. Chi *et al.* (2001) used information sent to infer a user's information needs from the user's traversal history and then to predict the user's expected surfing patterns. Glover *et al.* (1999b) developed a metasearch system, Inquirus 2, which can produce search results tailored to a personalized need using utility functions. These studies have two limitations. First, all the proposed methods need training data, either for statistical analysis or to improve the utility functions (Jones *et al.*, 2000; Sarukkai, 2000; Nanopoulos *et al.*, 2001). Second, most of the proposed methods handle site-specific transition models only, and are not appropriate for multi-site analysis (Sarukkai, 2000; Gündüz and Özsu, 2003).

3. Search methods

This section describes the proposed search methods in detail. First, the underlying user behavior function is presented. Then, the formulation and implementation of SVV and HLP are described.

3.1 User behavior function

A search engine generally returns a list of URLs to a user query in decreasing order of relevance; that is, the most relevant answers are on the top. Consequently, users

generally prefer top-ranking web pages over others. This phenomenon is called the primacy effect in psychology (Morris *et al.*, 2002). Several researchers (Lempel and Moran, 2000; Paepcke *et al.*, 2000) have similar observations as ours. Based on the primacy effect, the user behavior function (UBF) for the *i-th* item l_i within an ordered item list l is defined as follows:

$$UBF(l, l_i) = \alpha i^{\beta} (\text{where } \beta < 0) \quad (1)$$

where α denotes the user's preference of the first item, which represents the user's first impressions on the item list, and β denotes the user preference decay factor. Figure 1 shows the effect of the user preference decay factor for $\beta = -0.3$ and $\beta = -0.9$. When $|\beta|$ is small, the UBF value decreases slowly.

3.2 Search engine vector voting

SVV is based on the voting concept that a web page's ranking is dependent on how several selected search engines rank it, rather than its own contents. A web page wins a vote from a particular search engine if it is listed in the search engine's results to a given query.

SVV currently gathers top 50 items of search results from each of the following six well-known search engines, Google, Yahoo, AltaVista, LookSmart, Overture and Lycos. Advertisements and paid placements, which generally show up on the top, are removed before the search results are collected and processed by SVV.

The SVV search method rearranges the returned web pages based on their weights. The weight of a particular web page considers both the voting tendencies of the six

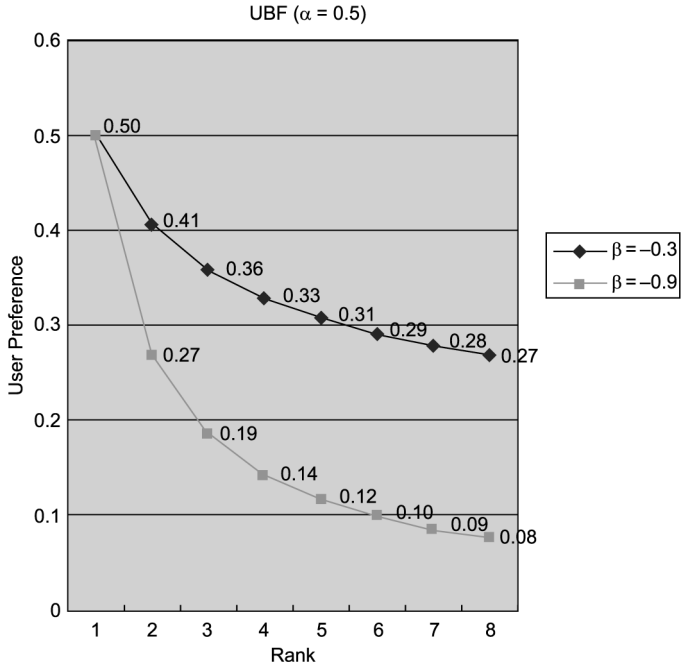

Figure 1.
Examples of UBF trajectory

search engines mentioned above and the user behavior function (UBF). The flow chart of SVV is shown in Figure 2.

Figure 3 shows the results of a sample run of the SVV method for the query term "php". The URLs are sorted in a decreasing order of their SVV weights. The SVV method presents the results with the vote distribution in filled rectangles and the degree of relevance in color balls.

Formally, the weight of a web page p for a user query q is defined as follows[1]:

$$w_{p,q} = \sum_{i=1}^{6} \alpha_{i,q} x_{i,p,q} \tag{2}$$

where $\alpha_{i,q}$ denotes the user's preference on the first match (which is generally the most relevant to a user query recommended by the search engine) to a given query q returned from search engine i; $x_{i,p,q}$ denotes the ranking of web page p in search engine i's results for query q, and β denotes the user preference of web pages (where $\beta < 0$). According to this definition, a web page clearly has larger weight if it either wins votes

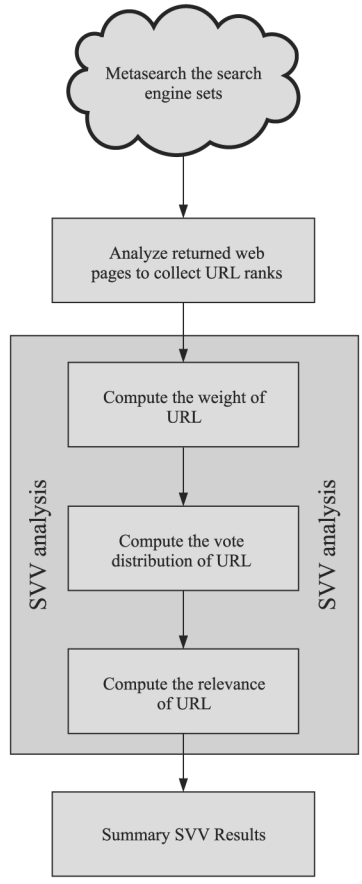

Figure 2.
Flow chart of SVV

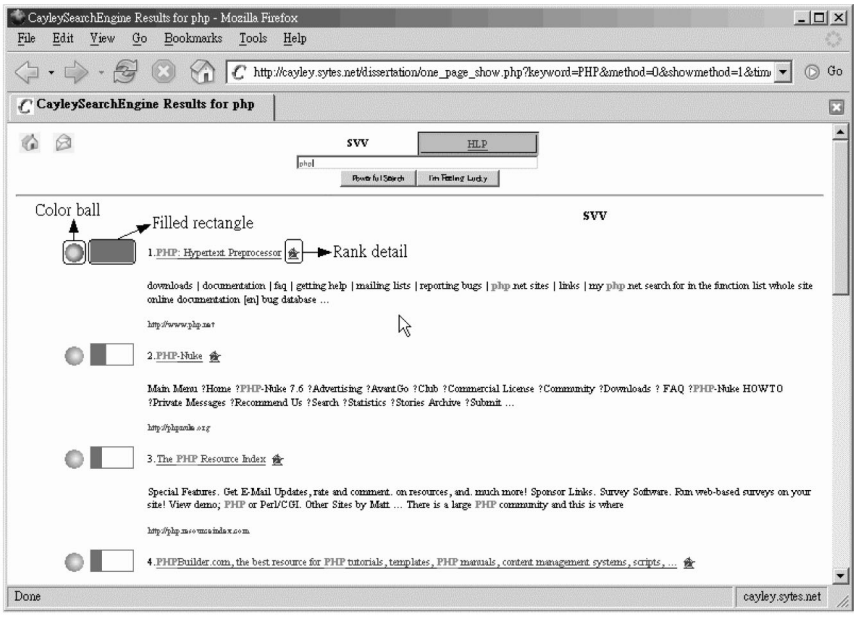

Figure 3.
A sample run of SVV

from more search engines or is ranked high in the results of at least one search engine. The web page URLs listed in the sample run of SVV shown in Figure 3 are sorted in decreasing order of weight. The "rank detail" button in Figure 3 shows the rank distribution of search engines on a particular URL.

To show the operations of the formulas related to SVV and HLP, a series of examples, based on the search results of the query "php", as shown in the sample run of SVV in Figure 3 and the sample run of HLP in Figure 4, appears below.

Example 1. Table I shows the rank distribution among six search engines for the query term "php". To calculate $w_{p,q}$, $\alpha_{i,q}$ and β first need to be defined. Initially, two random numbers[2] are generated, each from their own range respectively, as shown in Table II[3]. Then, the weight of URL "http://www.php.net" (the top URL listed in Figure 3), $w_{p,q} = 0.895259 \times 1^{-0.77304} + 0.844789 \times 1^{-0.77304} + 0.811069 \times 1^{-0.77304} + 0.93683 \times 1^{-0.77304} + 0.905779 \times 1^{-0.77304} + 0.889514 \times 1^{-0.77304} = 5.28324$.

To understand the voting tendency of the six search engines on a particular web page, a web page's vote distribution is also provided with the following formula:

$$P_{p,q} = w_{p,q} / \sum_{i=1}^{6} \alpha_{i,q} \tag{3}$$

where $P_{p,q}$ denotes the vote distribution of web page p for query q in all six search engines, which is represented as a filled rectangle in Figure 3. If a web page is listed at the top of all the six search engines for a given query, then the rectangle to the left of its URL is full with color.

Example 2. The vote distribution $P_{p,q}$ for the URL "http://www.php.net" is $5.28324/(0.895259 + 0.844789 + 0.811069 + 0.93683 + 0.905779 + 0.889514) = 1$. The rectangle to its left, as shown in Figure 3, is 100 percent filled with color.

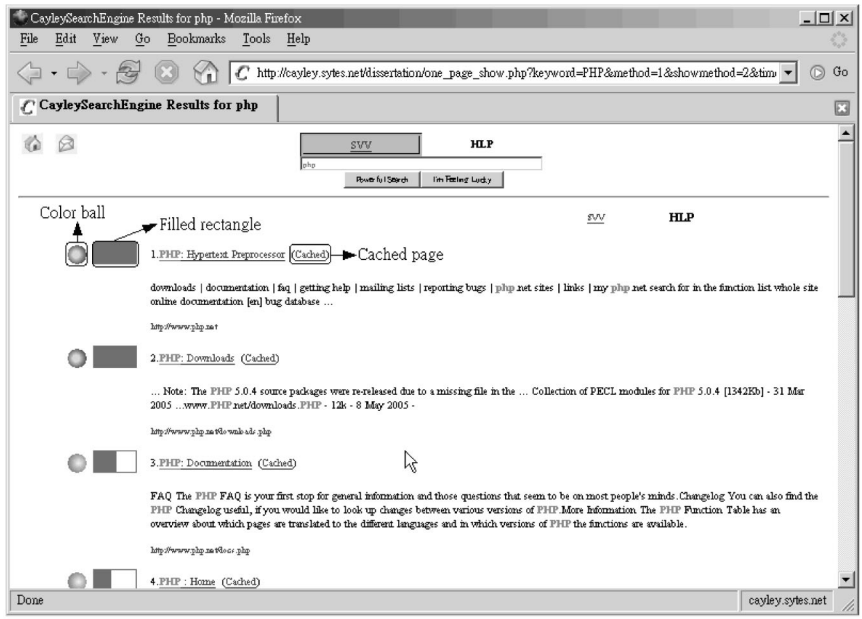

Figure 4.
A sample run of HLP

Table I.
Rank distribution among six search engines for query term "php"

URL	Search engine: Google	Yahoo	AltaVista	LookSmart	Overture	Lycos	$w_{p,q}$
http://www.php.net	1	1	1	1	1	1	5.28324
http://phpnuke.org	3	2	3	0	3	4	1.91623
http://php.resourceindex.com	6	5	8	0	5	2	1.41162

The vote distribution alone is insufficient for measuring the relevance of returned web pages to a given query. For instance, occasionally the votes on a web page are sparsely distributed among the six search engines, and the resulting almost empty rectangle may lead the user to misinterpret its relevance. Thus the relevance of a web page to a given query is also defined as follows:

$$R_{p,q} = \begin{cases} High, & w_{p,q} > \bar{w} + n\sigma_w \\ Middle, & \bar{w} < w_{p,q} \leq \bar{w} + n\sigma_w \\ Low, & otherwise. \end{cases} \quad (4)$$

where $R_{p,q}$ denotes the relevance of a web page p to a given query q; $\bar{w}$ denotes the average weight of all the web pages returned in this query, and σ_w denotes the standard deviation of the weights of all the returned web pages. Here, the probability of the condition $w_{p,q} > \bar{w} + n\sigma_w$, based on Chebyshev's theorem with n standard deviation for any population, is less than $1/n^2$ (we let $n = 3$ in the sample run shown in our results),

Parameter	Search engine											
	Google		Yahoo		AltaVista		LookSmart		Overture		Lycos	
$\alpha_{i,q}$ range	$0.8 \sim 0.95$		$0.8 \sim 0.95$		$0.8 \sim 0.95$		$0.8 \sim 0.95$		$0.8 \sim 0.95$		$0.8 \sim 0.95$	
β range	$-1 \sim -0.3$		$-1 \sim -0.3$		$-1 \sim -0.3$		$-1 \sim -0.3$		$-1 \sim -0.3$		$-1 \sim -0.3$	
$\alpha_{I,q}$	0.895259		0.844789		0.811069		0.93683		0.905779		0.889514	
β	−0.77304		−0.77304		−0.77304		−0.77304		−0.77304		−0.77304	

Table II.
The range of $\alpha_{i,q}$ and β, and their initial values

where n denotes any value greater than 1 (Walpole *et al.*, 1998). The web pages in this category are defined to have high relevance to this query. By contrast, the web pages whose weights fall below average are considered to have low relevance to the query. Other web pages between these two categories are considered to have medium relevance to the query. The three relevance categories are indicated by different color balls, namely green, yellow, and red balls for high, medium, and low relevance, respectively.

Example 3. The URL "http://www.php.net" is highly related to query "php", since $R_{p,q}$ is determined to be high by the following condition $5.28324 > 0.49174555266897 + 3 \times 0.84575553737867$, where $n = 3$, $\bar{w} = 0.49174555266897$ and $\sigma_w = 0.84575553737867$.

Notably, the color balls in Figure 3 not only help users judge the relevance of a web page to the query, but also help them compare web pages whose distribution rectangles are filled to a similar degree. That is, both the relevance indicator in the color ball and the vote distribution in the filled rectangle reinforce each other in helping users choose the most significant and relevant pages to their queries.

3.3 Hyperlink prediction

As we discussed above, users virtually always prefer high-ranked web pages. Users not only spend most time on the highest-ranked web pages, but also follow the hyperlinks on these pages to find more referred web pages (Koch, 2003). However, browsing through many hyperlinks is tedious and often distracts users. To recommend the most valuable hyperlinks to the users, this study developed the hyperlink prediction (HLP) method. The flow chart of HLP is shown in Figure 5.

HLP studies the source URLs listed in search results from SVV or any other search engines for discovering well-qualified hyperlinks, and then recommends them to users. For illustration, SVV search results are used as the inputs to HLP in the following sub-sections. To explore and collect qualified hyperlinks, three decisions need be repeatedly made on:

(1) Whether a web page is qualified to have the hyperlinks on it be visited.

(2) Which hyperlinks on a qualified page should be visited to fetch the referred web pages.

(3) The maximum number of successive hyperlinks being visited.

3.3.1 Finding qualified web pages. HLP first determines whether a web page is qualified for visiting its hyperlinks. Initially, the inputs to HLP, i.e. SVV search results pages are recognized as qualified pages by default. The qualifications $Q_{p,q}$ of other web pages except SVV result pages are computed individually by the following formula:

$$T_{pc,p} = \begin{cases} \log_2(T_{p,l}), & T_{p,l} > 2 \\ 1, & otherwise \end{cases} \tag{5.1}$$

$$Q_{p,q} = \begin{cases} 1, & if \left[\frac{\alpha_{p,q}}{T_{pc,p}}\right] \geq 1 \\ 0, & otherwise. \end{cases} \tag{5.2}$$

where $\alpha_{p,q}$ denotes the weight of web page p for a given query q, initially set to $w_{p,q}$ obtained from equation 2; $T_{pc,p}$ denotes the penalty cost for page p, and $T_{p,l}$ denotes the

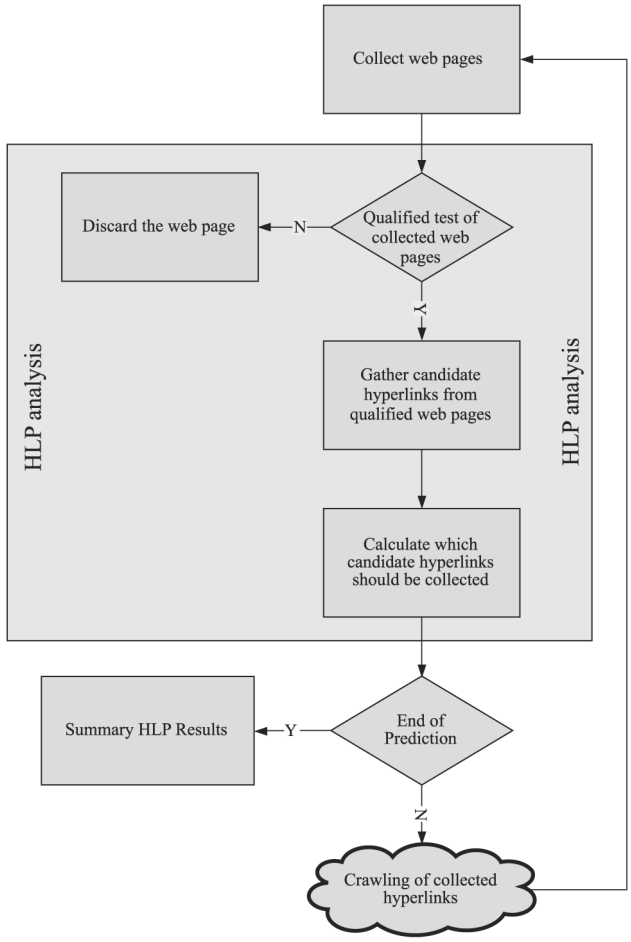

Figure 5.
HLP flow chart

total number of successive hyperlinks traversed from a starting URL listed in SVV results to page p, and determines the penalty cost. Users often get lost whenever they traverse a large number of successive hyperlinks (Catledge and Pitkow, 1995). Therefore, the penalty cost applies to a web page that is far from the starting URL. According to equation (5.1, 5.2), the penalty cost applies when $T_{p,l} > 2$. Finally, qualification value of 1 means that a web page is qualified for having its hyperlinks visited, and 0 means that it is not qualified.

Example 4. Consider the URL http://www.php.net listed in the SVV result page for query term "php" as shown in Figure 3. Partial of the detailed link traversal structure is shown in Figure 6.

Four cases are examined herein:

(1) The PHP home page (http://www.php.net): $T_{p,l} = 1$, $T_{pc,p} = 1$ and its qualification $Q_{p,q} = 1$ since $5.28324/1 \geq 1$, where $\alpha_{p,q} = 5.28324$.

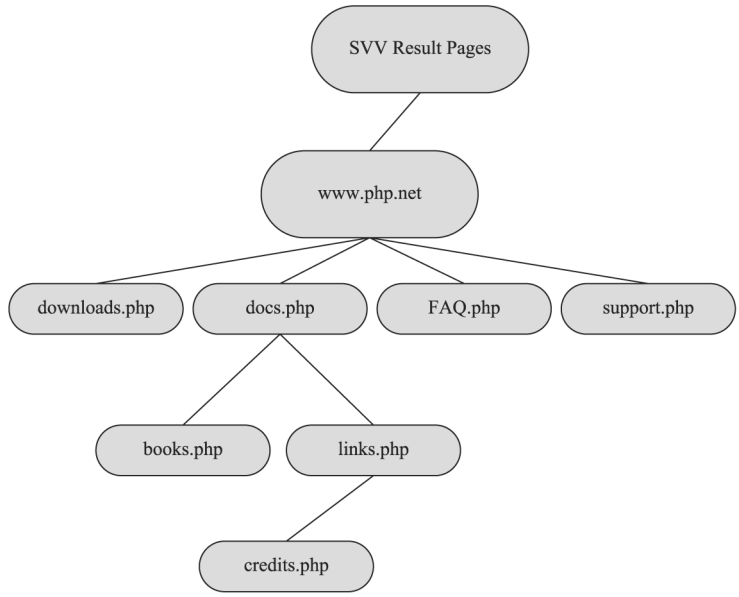

Figure 6.
Part of the link traversal structure for http://www.php.net

(2) The PHP downloads page (http://www.php.net/downloads.php): $T_{p,l} = 2$, and $T_{pc,p} = 1$ and its qualification $Q_{p,q} = 1$ since $5.28324/1 \geq 1$, where $\alpha_{p,q} = 5.28324$.

(3) The PHP books page (http://www.php.net/books.php): $T_{p,l} = 3$, and $T_{pc,p} = \log_2(3) = 1.58496$ and its qualification $Q_{p,q} = 1$ since $3.09166/1.58496 \geq 1$, where $\alpha_{p,q} = 3.09166$.

(4) The PHP credits page (http://www.php.net/credits.php): $T_{p,l} = 4$, and $T_{pc,p} = \log_2(4) = 2$ and its qualification $Q_{p,q} = 0$ since $1.80919/2 < 1$, where $\alpha_{p,q} = 1.80919$. $\alpha_{p,q}$ is discussed later in example 5.

The top three cases are qualified pages and the fourth page is not a qualified page.

Once a web page is evaluated as qualified, the HLP analysis process is invoked to fetch all the hyperlinks on the page, compute their individual weights and append these hyperlinks into candidate hyperlinks. Then, the candidate hyperlinks are analyzed to determine which hyperlinks should be collected.

3.3.2 Collecting qualified hyperlinks. A tournament competition is run to collect qualified hyperlinks from the candidate hyperlinks on a web page. At the start of the collection process, the URLs listed in the SVV results are set as the initial candidate hyperlinks. Then, the candidate hyperlinks are divided into "winner" and "loser" sets on the basis of their weights. The hyperlinks with high relevance as defined in equation (7.1) are classified into the "winner" set, and others are placed into the "loser" set. The weight of a starting web page URL is defined in equation 2. For other web pages, the weight of hyperlink l on web page p for given query q is given by the following equation:

$$w_{p,l,q} = \alpha_{p,q} x_{p,l} \tag{6}$$

where $x_{p,l}$ denotes the rank of hyperlink l on page p. A hyperlink is assumed to propagate its weight to the referred page. In other words, if the hyperlink l links page p to page p', then $w_{p,l,q}$ becomes $\alpha_{p',q}$, i.e. the weight of page p',q.

Example 5. Tables III and IV show the parameters of the top hyperlinks on URL "http://www.php.net" and its hyperlink http://www.php.net/docs.php, respectively. The value of $\alpha_{p,q}$ for URL http://www.php.net/docs.php used in Table IV is inherited from its corresponding $w_{p,l,q}$ obtained in Table III.

The hyperlinks of page p for a given query q to be collected into the winner set is specified as follows:

$$V_{p,l,q} = \left\{ l \text{ if } w_{p,l,q} > \overline{w_{p,l,q}} + n\sigma_w, \text{ for } \forall_{p,l,q} \right\} \tag{7.1}$$

$$V_{p,q} = \left| V_{p,l,q} \right| \tag{7.2}$$

where $V_{p,l,q}$ denotes the set of winning hyperlinks on page p for a given query q; $V_{p,q}$ denotes the number of winning hyperlinks on page p for a given query q; $\forall_{p,l,q}$ denotes all the links on page p for a given q; $\overline{w_{p,l,q}}$ denotes the average weight of all the hyperlinks on page p for a given query q, and $\sigma_{w_{p,l,q}}$ denotes the standard deviation of the weights of all hyperlinks on page p for a given query q. Then, the remaining hyperlinks on the page fall into the loser set.

Example 6. Table V shows the weights of hyperlinks on URL "http://www.php.net", and indicates which are in the winner set $V_{p,l,q}$, where $n = 3$, $\overline{w_{p,l,q}} = 0.472864$, and $\sigma_{w_{p,l,q}} = 0.384756$. Thus, the number of winning hyperlinks ($V_{p,q}$) is 4.

HLP then begins to build up the current hyperlink set once all hyperlinks have been successfully divided. The HLP analytical process first chooses from the winner set the hyperlinks that have not yet been gathered as part of the current hyperlink set. To avoid "link spamming", not all the winning hyperlinks are chosen. For example, in Figure 7, two qualified web pages, P_1 and page P_2, contain some identical hyperlinks as $P_{1,1}$, $P_{1,2}$ and $P_{1,3}$ links. If P_1 is an ancestor web page of P_2, then these duplicated hyperlinks, $P_{1,1}$, $P_{1,2}$ and $P_{1,3}$, should be collected when P_1 is processed. Page P_2 then gives the collection opportunity to other high-weighted hyperlinks that have not yet

Table III. Parameters of the child hyperlinks on URL "http://www.php.net"

URL	Parameter $\alpha_{p,q}$	$x_{p,l}$	β	$w_{p,l,q}$
http://www.php.net/downloads.php	5.28324	1	−0.77304	5.28324
http://www.php.net/docs.php	5.28324	2	−0.77304	3.09166

Table IV. Parameters of the child hyperlinks on URL "http://www.php.net/docs.php"

URL	Parameter $\alpha_{p,q}$	$x_{p,l}$	β	$w_{p,l,q}$
http://www.php.net/books.php	3.09166	1	−0.77304	3.09166
http://www.php.net/links.php	3.09166	2	−0.77304	1.80919

been collected. This approach ensures that all high-weighted hyperlinks would have chance to be collected, and are only collected once.

The hyperlinks are then collected from the loser set. Two decisions need to be made in this step:

(1) Determine the collection size, i.e. the number of hyperlinks to collect.

(2) Determine which hyperlinks to collect.

The following two cases are studied. First, if the current web page is just a starting SVV result page, then the collection size is determined by the following equation:

$$F_{p,q} = \left[\frac{V_{p,q} \times (1+\beta)}{\log(T_f)}\right] \tag{8}$$

where $F_{p,q}$ denotes the number of loser hyperlinks on page p for a given query q to be collected; T_f denotes the total number of hyperlinks in the loser set, and β denotes user preference parameter, which influence strong or weakness user preference. A large value of T_f indicates that the user cannot easily make a decision. Hence, $\log(T_f)$ is defined as the penalty cost. The equation clearly shows that the number of hyperlinks to be collected from the loser set is proportional to the number of hyperlinks in the winner set, depending on the user preference. The stronger the user preference, the

Table V.
Weights of hyperlinks on URL "http://www.php.net"

URL	$w_{p,l,q}$	Parameter $\overline{w_{p,l,q}} + n^* \sigma_{w_{p,l,q}}$	*in* $V_{p,l,q}$
http://www.php.net/downloads.php	5.28324	1.62713	✓
http://www.php.net/docs.php	3.09166	1.62713	✓
http://www.php.net/FAQ.php	2.25978	1.62713	✓
http://www.php.net/support.php	1.80919	1.62713	✓
http://www.php.net/mailing-lists.php	1.52254	1.62713	✗

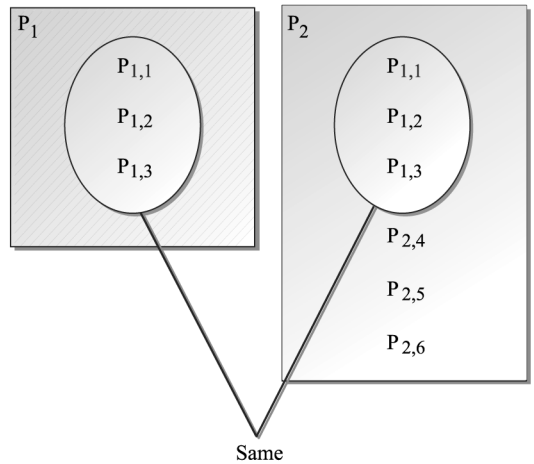

Figure 7.
Two web pages have the partially identical hyperlinks

larger the number of hyperlinks collected from the loser set at the beginning of the HLP process.

Example 7. The collection size from the loser set on the URL "http://www.php.net":

$$F_{p,q} = \left\lceil \frac{4 \times (1 + -0.77304)}{\log_{10}(94)} \right\rceil = 1,$$

where $V_{p,q} = 4$, $\beta = -0.77304$, and $T_f = 94$.

Second, if the current web page is not a starting SVV result page, then the collection size is calculated by the following equation:

$$F_{p,q} = \left\lceil \frac{\frac{\alpha_{p,q}}{\sum \forall_{p,l,q}(w_{p,l,q})} \times T_f \times (1 + \beta)}{\log(T_f)} \right\rceil \tag{9}$$

where $\sum \forall_{p,l,q}(w_{p,l,q})$ denotes the sum of the weights of all hyperlinks on a web page p for a given query q. The chance for a hyperlink in the loser set to be collected increases with the related weight of the current web page, i.e. $\alpha_{p,q} / \sum \forall_{p,l,q}(w_{p,l,q})$, as showed in this formula.

Example 8. The collection size from the loser set on URL "http://www.php.net/downloads.php":

$$F_{p,q} = \left\lceil \frac{\frac{5.28324}{221.48511872068} \times 136 \times (1 + -0.77304)}{\log_{10}(136)} \right\rceil = 1,$$

where $\alpha_{p,q} = 5.28324$, $\sum \forall_{p,l,q}(w_{p,l,q}) = 221.48511872068$, $\beta = -0.77304$, and $T_f = 136$.

Once the collection size is determined, the roulette wheel selection method from genetic algorithms (Goldberg, 1999; Sullivan, 2003; Obitko, 2004) is used to select hyperlinks from the loser set. The roulette wheel selection offers a higher probability of being selected to those hyperlinks with larger weights even though they fall into the loser set. Once the selection is finished, the selected hyperlinks are appended to the current hyperlink set. The HLP method then simultaneously fetches the referred web pages pointed to by the hyperlinks in the current hyperlink set for the next iteration.

3.3.3 Termination condition. HLP continues the collection process until the maximum number of successive hyperlinks is reached for each starting URL collected from the SVV search results. The number of successive hyperlinks to be visited is limited to let the HLP method respond quickly to user queries. Thus, the maximum number of successive hyperlinks to be collected is defined as follows:

$$ML_{p,q} = \lceil w_{p,q} \rceil \tag{10}$$

That is, the maximum number of successive hyperlinks for a given query q to be collected from a starting web page p listed in SVV results is determined by the limit of its own weight.

Example 9. The maximum number of successive hyperlinks to be collected from the URL "http://www.php.net", $ML_{p,q} = \lceil 5.28324 \rceil = 6$, where $w_{p,q} = 5.28324$.

3.3.4 A Sample run of HLP. Figure 4 shows the analytical results of a sample run of the HLP method for the query term "php". The hyperlinks are sorted in decreasing

order of weight. HLP, similar to SVV, presents the results with the degree of relevance in color balls and the voting distribution in filled rectangles. Notably, the second (http://www.php.net/downloads.php) and third (http://www.php.net/docs.php) items are hyperlinks explored from the first ranked item (http://www.php.net), because HLP has explored the inside hyperlinks of top ranked URLs in SVV results. Additionally, HLP provides a cached tag to those pages, which have been cached for later use, as shown in Figure 4.

4. Experiments

Three performance tests were performed on the proposed search methods. First, 23 volunteers were recruited from local bulletin boards to judge the quality of search results for some hot keywords. Second, the SVV metasearch engine was compared with the four best metasearch engines (Sherman, 2002a, b). Third, the SVV and HLP were compared with six well-known search engines.

4.1 User study

This experiment was performed to determine how users rate the proposed search methods compared with other search engines.

A total of 23 volunteers were recruited from local bulletin boards to judge the quality of search results returned by Google, Yahoo, AltaVista, LookSmart, Overture, Lycos and the proposed search methods[4], SVV and HLP. In this experiment, the search engines adopted for SVV were the preceding six popular search engines. The popularity of Google and Yahoo was emphasized by adjusting their $\alpha_{i,q}$ and β ranges so that the search results would be consistent with general preferences of most users. Additionally, sponsored and banner links at the top of search results from AltaVista, LookSmart, Lycos and Overture were dropped to make SVV rank reasonably without any distortion. HLP, in principle, can be applied to the search results of any search engine. In this experiment, the input search engine results for HLP were obtained from SVV. SVV was used to eliminate possible bias from any single search engine.

The testers were asked to perform queries on each search engine with popular query terms obtained from Lycos 50 (2004) including "Dragonball", "Pamela Anderson", "Britney Spears", "Las Vegas", "Harry Potter", "Kazaa", "WWE", "Jennifer Lopez", "Final Fantasy", "Yu-Gi-Oh!", "The Bible" and "NBA". The testers independently rated the search results of each search engine for a given query term on a score ranging from 1 to 5.

Table VI shows the experimental results. The number in each cell (except the last row) is the average score of a search engine on a given query term. For example, Google has an average score of 4.18 for the query term "Dragonball". The last row in Table VI shows the average score of each search engine on all testing query terms. Generally, the results were similar to those expected of the proposed search methods. That is, HLP scored higher than SVV and other search engines. The scores of SVV and HLP, 3.88 and 4.03 respectively, were significantly better than those of other methods. The SVV score, 3.88, was close to that of Google (3.81) and Yahoo (3.72), which is consistent with the SVV parameter configuration. The experiment confirmed that users are more satisfied with the proposed search methods than with general search engines, because of the depth of the information provided.

Considering the hot keywords performance of all eight search engines, an analysis of variance (ANOVA) analysis shows that $F = 4.656887$ (Table VII) is larger than

	Google	Yahoo	AltaVista	LookSmart	Overture	Lycos	SVV	SVV_HLP
Dragonball	4.18	4.23	3.08	3.57	2.25	3.51	4.18	4.33
Pamela Anderson	3.84	3.51	2.11	3.74	2.90	3.56	3.79	4.04
Britney Spears	4.09	3.43	2.78	4.02	1.77	3.33	3.93	4.34
Las Vegas	3.81	4.23	3.36	3.55	1.32	3.74	4.02	4.18
Harry Potter	3.15	2.69	3.70	3.77	1.33	3.27	3.86	3.77
Kazaa	4.11	3.86	3.71	3.95	2.34	3.86	4.03	4.07
WWE	3.97	3.74	3.38	3.73	2.25	3.51	3.95	4.22
Jennifer Lopez	4.29	3.95	3.02	3.16	2.67	3.66	4.02	3.45
Final Fantasy	3.51	4.01	3.52	3.66	2.28	3.37	3.83	4.33
Yu-Gi-Oh!	4.30	4.52	2.41	3.33	2.71	3.29	4.33	4.46
The Bible	2.73	2.63	2.09	2.86	2.21	2.90	2.57	2.79
NBA	3.71	3.82	3.77	3.79	1.08	3.35	4.04	4.35
Average	3.81	3.72	3.08	3.59	2.09	3.45	3.88	4.03

Note: HLP is not an extension of SVV. Rather, it is a post-search method that can augment search results of any search engines including SVV as stated above. However, in Tables VI and XI, HLP does mean SVV with HLP because it takes SVV results as its input

Table VI.
User study results of SVV, SVV with HLP and other search engines

	SS	DF	MS	F
Treatments	42.26177	7	6.037396	4.656887
Error	347.4471	268	1.296444	
Total	389.7089	275*		

Note: *Total degree freedom is $n - 1$, where n is obtained from 23*12 (23 users and 12 queries)

Table VII.
ANOVA Analysis of hot keyword queries

$F^{7}_{268}(0.05) = 2.043836$ (F distribution), which is strong evidence against the null hypothesis, implying a significant difference in the performance of the search engines on this task.

However, HLP had lower scores than other methods in some query terms, such as "Jennifer Lopez" and "Harry Potter", because it performs badly on web pages without description texts or with many non-textual elements such as Flash, photos, audio and movies. Thus, HLP has a hard time predicting hyperlinks from the top returned URL "http://www.jenniferlopez.com", which has no descriptive text, in the case of "Jennifer Lopez".

Notably, the average score of Overture was 2.09, the lowest among all search engines. The reason for Overture's low ranking is that it places very many banner links at the top of search results since it is a pay-per-click search engine for hot keywords. Meanwhile, no search engines scored well on the query term "The Bible", because some testers expected to find some "bible" books on programming topics.

4.2 Comparisons with metasearch engines

SVV was compared with four well-known metasearch engines, Dogpile, Excite, MetaCrawler and WebCrawler (Sherman, 2002a, b). The mean reciprocal rank (MRR) was used to measure the effectiveness of the metasearch engines. The 500 queries provided in the TREC 2002 (2003) QA data were applied to each metasearch engine. The MRR of each individual query is the reciprocal of the rank at which the first correct answer occurred, or zero if none of the top ten results contains a correct answer.

The score for the 500 queries is the mean of each individual query's reciprocal ranks. Table VIII lists the ranks at which the metasearch engines returned correct answers. For instance, Dogpile returned the correct page at rank #1 in 218 out of 500 cases, and returned no answer[5] in 116 out of 500.

Considering the MRR performance of Dogpile, Excite, MetaCrawler, WebCrawler and SVV, an ANOVA shows that $F = 14.9381005$ (Table IX) is larger than $F^4_{495}(0.05) = 2.389948$ (F distribution). Again, this finding is strong evidence against the null hypothesis, indicating that a significant difference exists in the performance of the metasearch engines on this task.

Multiple pairwise comparisons were then performed using the least significant difference (LSD) test with $p < 0.05$. As shown in Table X, SVV was found to be significantly better than the other four metasearch engines.

Finally, Figure 8 shows the MRR values achieved by the five metasearch search engines. SVV was found to out-perform the other metasearch search engines, because it rearranges the metasearch results based on the weights obtained from the user behavior function (UBF), but does not simply group the results together. Therefore, correct answers in the top ten results of any metasearch engine are very likely to appear in the top ten results of SVV.

Consequently, SVV outperforms the other metasearch engines by 19 percent ($(67.3\% - 56.6\%)/56.6\% \approx 19\%$) on average for the mean reciprocal rank studied.

4.3 Comparisons with general search engines

MRR was used to measure the effectiveness of SVV, HLP and some well-known search engines. The 500 queries provided in the TREC 2002 (2003) QA data were applied individually to Google[6], Yahoo, AltaVista, LookSmart, Overture, Lycos, SVV and

Table VIII.
Distribution of ranks at which the first correct answer was returned by metasearch engines out of the 500 queries

Rank	Dogpile	Excite	MetaCrawler	WebCrawler	SVV
1	218	220	219	220	294
2	96	82	90	90	60
3	28	37	23	23	18
4	10	18	15	19	8
5	11	5	14	12	8
6	5	8	9	11	10
7	3	8	7	3	2
8	11	5	4	2	5
9	1	0	3	3	2
10	1	5	1	6	1
No	116	112	115	111	92

Table IX.
ANOVA Analysis of metasearch engines

	SS	DF	MS	F
Treatments	51.93086	4	12.98271583	14.9381005
Error	430.2049	495	0.869100851	
Total	482.1358	499		

SE comparison	Low confidence limit	Difference between means	Upper confidence limit
Dog-Exci	−0.952	0.083	1.118
Dog-Meta	−0.790	0.245	1.280
Dog-Web	−1.109	−0.074	0.961
Dog-SVV	−2.493	−1.458	−0.423*
Exci-Dog	−1.118	−0.083	0.952
Exci-Meta	−0.848	0.187	1.222
Exci-Web	−1.139	−0.104	0.931
Exci-SVV	−2.508	−1.473	−0.438*
Meta-Dog	−1.280	−0.245	0.790
Meta-Exci	−1.222	−0.187	0.848
Meta-Web	−1.211	−0.176	0.859
Meta-SVV	−2.528	−1.493	−0.458*
Web-Dog	−0.961	0.074	1.109
Web-Exci	−0.931	0.104	1.139
Web-Meta	−0.859	0.176	1.211
Web-SVV	−2.477	−1.422	−0.407*
SVV-Dog	0.423	1.458	2.493*
SVV-Exci	0.438	1.473	2.508*
SVV-Meta	0.458	1.493	2.528*
SVV-Web	0.407	1.442	2.477*

Notes: The SAS system. Analysis of variance procedure. T-tests (LSD) for variable: OBS. This test controls the type I comparisonwise error rate not the experimentwise error rate. Alpha=0.05; Confidence=0.95; df=495; MSE=2.84; Critical value of t=1.96481; Least significant difference=1.0353; *Comparisons significant at the 0.05 level

Table X.
Results of LSD test on metasearch engines

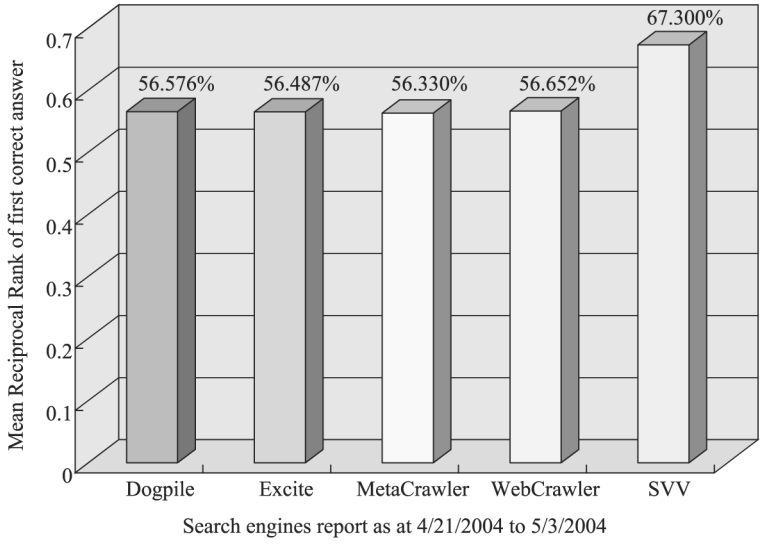

Figure 8.
Mean reciprocal rank achieved by the metasearch engines

HLP. Table XI shows the ranks at which the first correct answers were returned by the search engines.

Considering the MRR performance of all eight search engines, an ANOVA analysis shows that $F = 29.79031556$ (Table XII) is larger than $F_{492}^{7}(0.05) = 2.028182926$ (F distribution). Thus, the null hypothesis is rejected, indicating that a significant difference was found in the performance of the search engines on this task.

Figure 9 shows the MRR values achieved by the eight search engines. Google, Yahoo, SVV and HLP out-performed AltaVista, LookSmart, Overture and Lycos. Most significantly, the proposed search methods, SVV and HLP, performed better than most other search engines. As stated previously, SVV collectively expresses the voting behavior of the six other search engines' in terms of correct answers. Therefore, correct answers appearing in any search engine's top ten results are highly likely to also appear in SVV top ten results, but not necessarily the other way around. HLP further expanded the inside links of top ranked URLs in SVV results into its top ten, results and possibly pushed correct answers out of the top ten. Thus, HLP performed slightly worse than SVV. However, HLP performed better than SVV in some cases, in which the correct answer appears inside a hyperlink of a top ranked URL in SVV results, but does not appear in the URL itself. For example, SVV could not find a correct answer for Qid = 1562 (Q: where did the US civil war begin? Answer: Fort Sumter). However, HLP found the correct answer at rank 7 URL http://www.plainfield.k12.in.us/hschool/webq/webq44/HISTORY.HTM, which is a referred link from http://www.plainfield.k12.in.us/hschool/webq/webq44/civwar.htm (a URL which appeared at rank 3 in SVV).

Next, the search results of the most popular search engines, Google and Yahoo, were taken instead of SVV to HLP to demonstrate that HLP can be used to improve the search results of any search engines. The combined search engines were designated Google_HLP and Yahoo_HLP respectively. Figure 10 shows the MRRs achieved by

Table XI.
Distribution of ranks of the first correct answer returned by search engines out of the 500 queries

Rank	Google	Yahoo	AltaVista	LookSmart	Overture	Lycos	SVV	SVV_HLP
1	262	266	209	115	224	172	277	278
2	54	58	71	55	84	74	64	42
3	31	34	34	35	33	47	20	27
4	19	15	24	33	16	31	11	12
5	17	12	22	27	14	17	9	6
6	8	8	10	24	8	11	9	3
7	5	4	8	9	6	7	2	4
8	5	5	7	7	9	10	5	5
9	0	1	5	4	4	6	2	6
10	4	3	6	4	2	4	2	4
No	95	94	104	187	100	121	99	113

Table XII.
ANOVA Analysis of general search engines

	SS	DF	MS	F
Treatments	291.7441	7	41.67773369	29.79031556
Error	688.3259	492	1.399036328	
Total	980.0701	499		

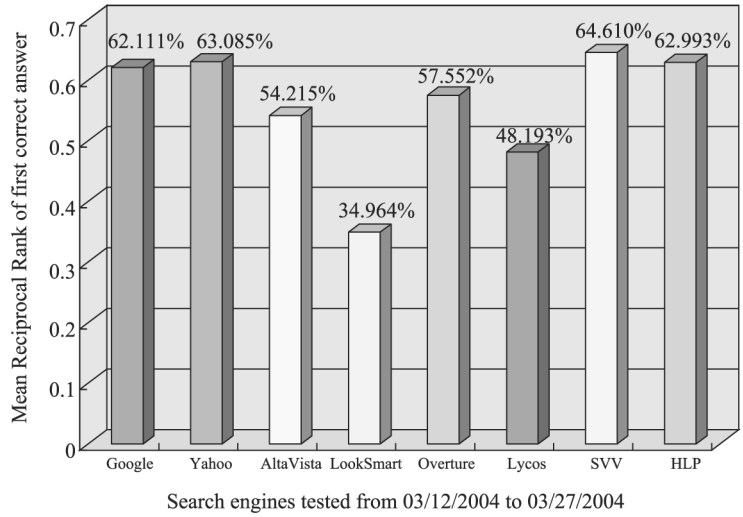

Figure 9.
Mean reciprocal ranks achieved by the general search engines

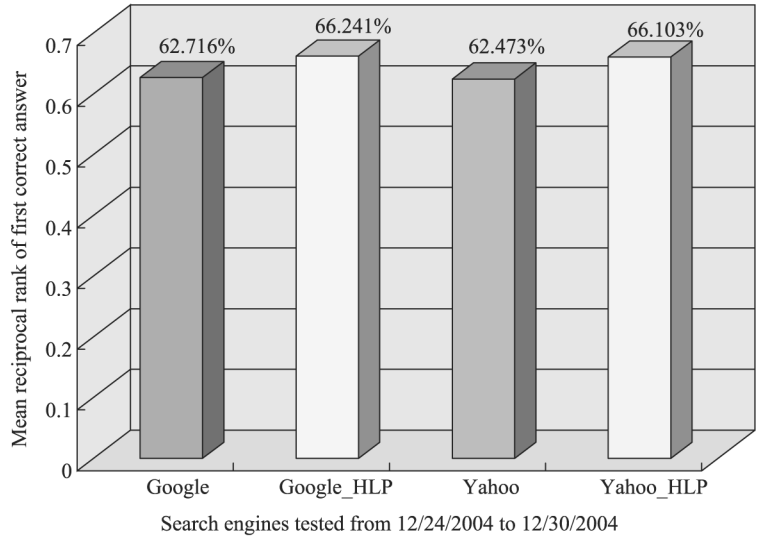

Figure 10.
Mean reciprocal rank achieved by Google, Google_HLP, Yahoo, and Yahoo_HLP

Google (62.716 per cent), Google_HLP (66.241 per cent), Yahoo (62.473 per cent), and Yahoo_HLP (66.103 per cent). Obviously, Google_HLP and Yahoo_HLP out-performed their respective originating search engines, Google and Yahoo. More specifically, HLP, when applied to Google and Yahoo, can improve their performance by 5.62 per cent $((66.241\% - 62.716\%)/62.716\% \approx 5.62\%)$ on average, for the mean reciprocal rank studied. Similarly, HLP could be used to improve the results of AltaVista, LookSmart, Overture and Lycos.

5. Conclusions and future work

This study developed SVV, a metasearch and ranking method and HLP, a post-search and ranking method. These methods working together can recommend a user's next hyperlink access based on metasearch results. The proposed design is based on the primacy effect of browsing behavior, that users favor top ranking items in search results. This approach is intuitive and needs no training data at all. A user study showed that users are more satisfied with the proposed search methods than general search engines on hot keywords. Moreover, two performance measures confirmed that the proposed search methods out-perform other metasearch and search engines.

Future work is planned along several directions. First, the search methods implemented are primarily based on the keyword match between the contents of web pages and the user query items. Using the semantic web in recommending concepts or items relevant to the user query might help finding the exact contents required by users. This approach is particularly helpful when the users lack sufficient knowledge about the domains in which they are searching. Second, a mechanism that groups search results for improving the way search results are segmented and displayed also helps users locate the required information. Finally, further user feedback is needed to fine-tune the search parameters, including α and β, to improve the performance of the proposed methods.

Notes

1. Inverse document frequency (IDF) (Jones, 1972) is a popular measure of a word's importance used in information retrieval. A drawback of IDF is that it treats all texts containing a certain term equally. According to the primacy effect, users prefer terms in the front to those at the back. Therefore, this study uses user behavior function (UBF) instead of IDF to redefine term weighting.
2. According to literature, human behavior is a random process (Zipf, 1949; Malerba *et al.*, 2002). Moreover, human preferences are values ranging from lower bound to upper bound (Yao, 1995). Therefore, this study uses random numbers to simulate user behavior. The initial ranges of α and β were obtained from our own evaluation experiments and by mining the user access log.
3. The initial ranges of $\alpha_{i,q}$ and β were formulated from performance evaluation experiments in this study. Both $\alpha_{i,q}$ and β would be updated regularly according to performance evaluation results such as the user study on hot keywords and the MRR measure.
4. The metasearch engine used in this study is available at: http://cayley.sytes.net/dissertation/index_home.php
5. A "no" answer was defined as no response from the TREC-2002 provided, no correct answer or a dead link (404 not found, 403 forbidden or timeout).
6. Google has been blocking queries from metasearch engines for about two years. In order to access Google by SVV, the crawler was made to emulate Opera 7.11 by changing the user-agent of the http-header to "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0) Opera 7.11".

References

Braslavski, P., Alshanski, G. and Shishkin, A. (2004), "ProThes: thesaurus-based meta-search engine for a specific application domain", *Proceedings of the 13th International World Wide Web Conference*, *New York, NY*, pp. 222-3.

Brin, S. and Page, L. (1998), "The anatomy of a large-scale hypertextual web search engines", *Proceedings of the 7th World Wide Web Conference*, Brisbane, pp. 107-17.

Catledge, L.D. and Pitkow, J.E. (1995), "Characterizing browsing strategies in the world-wide web", *Computer Networks and ISDN Systems*, Vol. 27 No. 6, pp. 1065-73.

Chakrabarti, S., Dom, B.E., Kumar, S.R., Raghavan, P., Rajagopalan, S., Tomkins, A., Gibson, D. and Kleinberg, J. (1999), "Mining the web's link structure", *IEEE Computing*, Vol. 32 No. 8, pp. 60-7.

Chen, M., LaPaugh, A.S. and Singh, J.P. (2002), "Predicting category accesses for a user in a structured information space", *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Tampere*, pp. 65-72.

Chi, E.H., Pirolli, P., Chen, K. and Pitkow, J. (2001), "Using information scent to model user information needs and actions on the web", *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Seattle, WA, pp. 490-7.

Dreilinger, D. and Howe, A. (1996), "An information-gathering agent for querying web search engine", *Tech. Report CS-96-111*, Computer Science Department, Colorado State University, Fort Collins, CO.

Glover, E.J., Lawrence, S., Birmingham, W.P. and Giles, C.L. (1999a), "Architecture of a metasearch engine that supports user information needs", *Proceedings of the 8th International Conference on Information Knowledge Management*, pp. 210-16.

Glover, E.J., Lawrence, S., Gordon, M.D., Birmingham, W.P. and Giles, C.L. (1999b), "Recommending web documents based on user preferences", *SIGIR 99 Workshop on Recommender Systems*, available at: www.neci.nec.com/~ lawrence/papers/search-sigir99/search-sigir99.pdf

Goldberg, D.E. (1999), *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Redwood City, CA.

Gündüz, Ş. and Özsu, M.T. (2003), "Recommendation models for user accesses to web pages", *Proceedings of Joint International Conference ICANN/ICONIP 2003*, Istanbul, pp. 1003-10.

Hai, H., Meng, W., Clement, Y. and Zonghuan, W. (2004), "Automatic extraction of web search interfaces", *Proceedings of the 13th International World Wide Web Conference*, *New York, NY*, pp. 414-5.

Henzinger, M.R. (2001), "Hyperlink analysis for the web", *IEEE Internet Computing*, Vol. 5 No. 1, pp. 45-50.

Hu, W.C., Chen, Y., Schmalz, M.S. and Ritter, G.X. (2001), "An overview of the world wide web search technologies", *Proceedings of the 5th World Multi-conference on Systems, Cybernetics and Informatics, SCI2001*, available at: http://webminer.mis.yzu.edu.tw/ref/hu01overview.pdf

Jansen, B., Spink, A., Bateman, J. and Saracevic, T. (1998), "Real life information retrieval: a study of user queries on the web", *ACM SIGIR Forum*, Vol. 32 No. 1, pp. 5-17.

Jones, K.S. (1972), "A statistical interpretation of term specificity and its application in retrieval", *Journal of Documentation*, Vol. 28 No. 1, pp. 11-21.

Jones, S., Cunningham, S.J., McNab, R.J. and Boddie, S. (2000), "A transaction log analysis of a digital library", *International Journal on Digital Libraries*, Vol. 3 No. 2, pp. 152-69.

KartOO (2004), "KartOO Visual Metasearch Engine", available at: www.kartoo.com/

Koch, R. (2003), *The 80/20 Individual: How to Build on the 20% of What You Do Best*, Doubleday & Company, London.

Lawrence, S. and Giles, C.L. (1998), "Context and page analysis for improved web search", *IEEE Internet Computing*, Vol. 2 No. 4, pp. 38-46.

Lempel, R. and Moran, S. (2000), "The stochastic approach for link-structure analysis (SALSA) and the TKC effect", *Computer Networks*, Vol. 33, pp. 387-401.

Li, Y. (1998), "Toward a qualitative search engine", *IEEE Internet Computing*, Vol. 2 No. 4, pp. 24-9.

Lycos 50 (2004), "Lycos 50 with Aaron Schatz – Lycos.com", available at: http://50.lycos.com/

Malerba, D., Esposito, F. and Ceci, M. (2002), "Mining HTML pages to support document sharing in a cooperative system", *Proceedings of International Conference on Extending Database Technology*, pp. 420-34.

Mall-Net (2001), "Web site design to sell", available at: www.mall-net.com/se_report/

Meng, W., Clement, T.Y. and Liu, K.L. (2002), "Building efficient and effective metasearch engines", *ACM Computing Surveys*, Vol. 34 No. 1, pp. 48-89.

Morris, C.G., Levine, A. and Maisto, A.A. (2002), *Psychology: An Introduction*, Prentice-Hall, Englewood Cliffs, NJ.

Nanopoulos, A., Katsaros, D. and Manolopoulos, Y. (2001), "Exploiting web log mining for web cache enhancement", *Proceedings of the 3rd International Workshop on Mining Web Log Data across All Customers Touch Points*, pp. 68-87.

Nick, Z.Z. and Themis, P. (2001), "Web search using a genetic algorithm", *IEEE Internet Computing*, Vol. 5 No. 2, pp. 18-26.

Obitko, M. (2004), "Introduction to genetic algorithms with Java applets", available at: http://cs.felk.cvut.cz/ ~ xobitko/ga/

Osdin, R., Ounis, I. and White, R.W. (2002), "Using hierarchical clustering and summarisation approaches for web retrieval: Glasgow at the TREC 2002 interactive track", *Proceedings of the 10th Text Retrieval Conference*.

Paepcke, A., Garcia-Molina, H., Rodriguez-Mula, G. and Cho, J. (2000), "Beyond document similarity: understanding value-based search and browsing technologies", *SIGMOD Record*, Vol. 29 No. 1, pp. 80-92.

Sarukkai, R.R. (2000), "Link prediction and path analysis using Markov chains", *Computer Networks*, Vol. 33, pp. 377-86.

Selberg, E. and Etzioni, O. (1997), "The metacrawler architecture for resource aggregation on the web", *IEEE Expert*, Vol. 12 No. 1, pp. 11-14.

Sherman, C. (2002a), "SearchDay – the big four meta search engines", available at: http://searchenginewatch.com/searchday/article.php/2160781

Sherman, C. (2002b), "SearchDay – the best and most popular meta search engines", available at: http://searchenginewatch.com/searchday/article.php/2160791

Sobek, M. (2002), "Additional factors influencing PageRank", available at: http://pr.efactory.de/e-further-factors.shtml

Sougné, J.P. (2000), "Short-term memory in a network of spiking neurons", *Tech. Report*, University of Liège, Liège, available at: www.ulg.ac.be/cogsci/jsougne/TR2000-1.pdf

Spink, A., Wolfram, D., Jansen, B.J. and Saracevic, T. (2001), "Searching the web: the public and their queries", *Journal of the American Society of Information Science*, Vol. 53 No. 2, pp. 226-34.

Sullivan, M. (2003), "An introduction to genetic algorithms", available at: www.cs.qub.ac.uk/ ~ M.Sullivan/ga/ga_index.html

Svidzinska, R. (2001), "A world wide web meta search engine using an automatic query routing algorithm", Master thesis at Auburn University, Auburn, AL, available at: ftp://ftp.eng.auburn.edu/pub/techreports/csse/01/CSSE01-06.ps.gz

TREC 2002 (2003), "TREC 2002 QA data", available at: http://trec.nist.gov/data/qa/t2002_qadata.html

Walpole, R.E., Myers, R.H. and Myers, S.L. (1998), *Probability and Statistics for Engineers and Scientists*, Prentice-Hall, Englewood Cliffs, NJ.

Yao, Y.Y. (1995), "Measuring retrieval effectiveness based on user preference of documents", *Journal of the American Society for Information Science*, Vol. 46 No. 2, pp. 133-45.

Yates, R.B. and Neto, B.R. (1999), *Modern Information Retrieval*, Addison-Wesley, Redwood City, CA.

Zacharis, N. and Panayiotopoulos, T. (2002), "SpiderServer: the metasearch engine of WebNaut", *Proceedings of the 2nd Hellenic Conference on Artificial Intelligence, Thessaloniki*, pp. 475-86.

Zipf, G.K. (1949), *Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology*, Addison-Wesley, Redwood City, CA.

Appendix 1

The pseudo code of SVV is listed as follows:

```
Algorithm search (query)
{
  Concurrently collect the search results to the query from six search engines;
  Does a page analysis to collect a URL rank from the above collection;
  Do a SVV analysis on the results of the page analysis;
  Set the SVV results to be the analysis results;
  Break;
  Do a Summary analysis on the analysis results
  {
    Sort the analysis results according to their weight;
    Layout the results;
    Combine the results to be final results;
    Show final results;
  } end of Do;
} end of Algorithm;
```

Appendix 2

The pseudo code of HLP is listed as follows:

```
Algorithm HLP (query)
{
  If the query has not yet been processed before then
  {
    If the corresponding SVV search result is NULL then
    {
      Do a SVV search on this query;
      Set the URLs listed in the SVV search result as candidate hyperlinks;
    } end of If;
    Do a HLP analysis on the candidate hyperlinks
    {
    1. Divide the candidate hyperlinks into winner set and loser set;
    2. Select a majority of hyperlinks from the winner set and a minority of URLs from the loser
set into the current hyperlink set;
    3. Append the current hyperlink set to the final hyperlink collection set;
```

```
      4. Concurrently fetch the referred web pages pointed to by the hyperlinks in the current
hyperlink set, append these pages to next-level web pages, and add these web pages to the final
web page collection;
      } end of Do;
      Set the number of hyperlink traversal to 1;
      While (the maximum number of successive hyperlinks is not reached)
      {
        Assign next-level web pages to current web pages;
        Set next-level web pages to NULL;
        For all pages of the current web pages
        {
        If (a web page is qualified to have its hyperlinks to be visited then)
        {
          Set candidate hyperlinks to NULL;
          Do a Page Analysis on this web page
          {
            Fetch all the hyperlinks on this web page;
            Compute the weights of these hyperlinks;
            Assign these hyperlinks to candidate hyperlinks;
          } end of Do;
          Do a HLP analysis on the candidate hyperlinks;
        } end of If;
        } end of For;
        Increase the number of hyperlink traversal by 1;
      } end of While;
      Do a HLP Summary on the final hyperlink collection set
      {
        Sort the hyperlinks in a decreasing order of their weights;
        Layout the hyperlinks in a predefined format;
        Show the cached tag to the hyperlink of a cached web page;
      } end of Do;
    } end of If;
    Return HLP search result;
} end of Algorithm;
```