The Electronic Library

SAFQuery: a simple and flexible advanced Web search interface

Wei-Chao Lin Shih-Wen Ke Chih-Fong Tsai

SAFQuery: a simple and flexible advanced Web search interface

Wei-Chao Lin
Hwa Hsia University of Technology, New Taipei City, Taiwan
Shih-Wen Ke
Chung Yuan Christian University, Zhongli, Taiwan, and
Chih-Fong Tsai
National Central University, Jhongli, Taiwan

Received 2 December 2014
Revised 2 December 2014
Accepted 3 February 2015

Abstract

Purpose – This paper aims to introduce a prototype system called SAFQuery (Simple And Flexible Query interface). In many existing Web search interfaces, simple and advanced query processes are treated separately that cannot be issued interchangeably. In addition, after several rounds of queries for specific information need(s), it is possible that users might wish to re-examine the retrieval results corresponding to some previous queries or to slightly modify some of the specific queries issued before. However, it is often hard to remember what queries have been issued. These factors make the current Web search process not very simple or flexible.

Design/methodology/approach – In SAFQuery, the simple and advanced query strategies are integrated into a single interface, which can easily formulate query specifications when needed in the same interface. Moreover, query history information is provided that displays the past query specifications, which can help with the memory load.

Findings – The authors' experiments by user evaluation show that most users had a positive experience when using SAFQuery. Specifically, it is easy to use and can simplify the Web search task.

Originality/value – The proposed prototype system provides simple and flexible Web search strategies. Particularly, it allows users to easily issue simple and advanced queries based on one single query interface, interchangeably. In addition, users can easily input previously issued queries without spending time to recall what the queries are and/or to re-type previous queries.

Keywords Information retrieval, Interfaces

Paper type Technical paper

Introduction

The advancement of the Internet and World Wide Web technologies has led to the rapid growth of the number of websites and web pages. They are increasingly being used by all portions of society. Web search engines are the tool that enables Web users to discover and locate information (Bar-Ilan, 2004). In general, Web users spend a lot of their time using Web search engines (such as Google, Yahoo, etc.) to locate material on the vast and unorganized Web. More specifically, according to Lawrence and Giles (1999), about 85 per cent of users use Web search engines to locate information. However, the dramatic increase in the volume of Internet pages has led to the problem of information overload (Berghel, 1997). Simply speaking, it is often difficult for Web users to find the desired information quickly and easily from the vast amount of web pages available.

The Electronic Library
Vol. 34 No. 1, 2016
pp. 155-168
© Emerald Group Publishing Limited
0264-0473
DOI 10.1108/EL-12-2014-0207

To decrease information overload, many techniques have been proposed for boosting search results, such as query refinement by classification, clustering of the results, filtering out non-valuable documents (or web pages) from the set of retrieved results, question and answering and so forth (Bar-Ilan, 2004). No matter what techniques are used to enhance the retrieval results generated by user queries, a Web search interface must be provided for users to perform their searches and then display the retrieval results. Therefore, user satisfaction with Web search engines during the information access process is heavily dependent on their experience with the search interface, including aspects of learnability, usability, usefulness and aesthetic appeal (Hassenzahl and Tractinsky, 2006; Heimonen *et al.*, 2008).

Essentially, the information access process is an interaction cycle that consists of query specification based on an information need, receipt and examination of retrieval results, and then either stopping or reformulating the query and repeating the process until a desired result set is found (Baeza-Yates and Ribeiro-Neto, 1999). In general, there are two approaches to formulating a query, the simple keyword-based query and the advanced query. The simple keyword-based query, as the basic search function, is always provided by Web search engines. The advanced query, on the other hand, is based on query specifications. When users have a specific need in mind, for example to retrieve "information retrieval"-related documents in accordance with the "PDF format" that are registered as "the UK domain name" and were uploaded in "the past two years", some sort of advanced query interface is needed to perform these query specifications. For the search engines that provide the advanced query interface, such as Google, users must first go to the specific advanced query interface. Note that it is assumed that users who have very clear information needs can specify their queries. The query specifications are then sent to the system and finally the retrieval results are displayed. However, most of the time it is unlikely that users will be satisfied with the initial retrieval results and end the information access process. Users may change their initial query specifications to a greater or lesser degree after evaluating the initial retrieval results. In this case, users must go again to the advanced query interface, reformulate their query specifications, send the query to the system and so on. In other words, this kind of query specification is not a trivial task. Moreover, when the search session(s) take some time, some users may fully/partially forget his/her previous specifications, which can leave them frustrated. As this advanced query process depends solely on the design of the advanced query interface, the usefulness of current systems is questionable. Furthermore, as information access is an iterative process, the current interfaces cannot reduce the working memory load, which is one important principle of interface design, as discussed by Baeza-Yates and Ribeiro-Neto (1999). In other words, the interfaces should be able to keep track of choices made during the search process. By doing this, users are allowed to return to temporarily abandoned strategies, jump from one strategy to the next and retain information and context across search sessions.

Given the limitations inherent to current Web search interfaces, the authors present a prototype system, namely, SAFQuery (Simple And Flexible Query interface), which allows users to issue simple keyword-based queries and easily formulate query specifications when needed in the same interface. In addition, it can simplify the advanced query process based on a query history table that displays the past query specifications, which can help with the memory load.

The next section provides an overview of related literature, including Web search behaviour and related prototype interfaces. The proposed prototype system is then introduced. System evaluation and its results are presented. Finally, some conclusions are provided.

Literature review

It is very important to understand the search behaviour of Web users, as the findings have implications for designers in developing more effective interfaces. Therefore, the authors reviewed related studies of Web search behaviour and then describe some related interface systems and techniques for the Web search purpose.

Web search behaviour

Although many Internet-enabled applications and services are available today, the primary use of the Internet (other than e-mail) is searching for information (Gordon and Pathak, 1999). Web searches usually follow the classic information retrieval (IR) model assuming that the user is driven by an information need, which can be defined as the user uses an IR system (Web search engine, in our case) to accomplish his or her perceived need for information. Particularly, a query in some query language is constructed based on the information after which a query refinement process might be used to create new queries and/or refine the search results (Broder, 2002).

According to Broder (2002), Web queries can be classified into three classes: navigational, informational and transactional. Navigational queries aim at reaching a particular site that the user has in mind. The purpose of informational queries is to find information assumed to be available on the Web. Finally, the intent of transactional queries is to perform some Web-mediated activity, such as shopping, downloading various types of files, accessing certain databases, finding servers and so forth. In addition, the analysis of the query log at AltaVista shows that information and transactional queries are the two most prevalent types of queries.

While information access is an iterative process, each iteration in the search formulation and reformulation process can be considered a search move. Wildemuth (2004) found that the search tactics (sets of moves) of medical students changed over time as their domain knowledge changed. In addition, the searcher's domain knowledge can affect the process of search strategy formulation and reformulation, as well as retrieval success and the outcomes of the search. Similarly, user behaviour varies not only between search systems, online catalogues and digital libraries, but also depends on the type of information sought and the way users search (Markey, 2007). Another aspect that differentiates search behaviour is the user's demographics, such as age, gender, ethnicity, income and educational level (Weber and Castillo, 2010).

White *et al.* (2009) studied the effect of domain expertise on Web search behaviour in the domains of medicine, finance law, and computer science. Specifically, they characterized the nature of the queries, search sessions, websites visited and search success for users identified as experts and non-experts within these domains. They showed that domain experts use different search strategies and are more successful than non-experts in the four different domains. They suggested that any search system that takes advantage of domain expertise needs to be able to identify whether a user is an expert or a non-expert, and then modify the experience accordingly. Costa and Silva (2011) utilized query-level analysis to show that users tended to add more terms in the modified queries (second queries) rather than to remove them. In addition, users tend to go from broad to narrow queries in Web search engines. They examined the following four advanced search operators:

(1) NOT, to exclude all results with a term in their full-text queries;

(2) PHRASE, to match all results with a phrase in their queries;

(3) SITE, to match all results from a domain name; and

(4) TYPE, to match all results from a media type.

The authors showed that about one-fourth of the queries included operators, with PHRASE and SITE being the two most used operators. Finally, temporal-level analysis showed that users are more interested in old documents. In other words, the older the documents are, the more likely they are to being included in queries.

According to Spink *et al.* (2002) and Spink and Jansen (2004), general Web queries are still short, with most users entering two to three terms per query and two to three queries per search. In addition, Spink *et al.* (2002) showed that most users enter only one or two queries and conduct limited query reformulation. Yamin and Ramayah (2011) studied user search behaviour related to query formulation, which they divided into breadth and depth query strategies. In the breadth strategy, queries are formulated in a way that is general, wide and not focused on the domain, while depth queries are narrowed to the specific domain and the use of keywords for the search task is more specific. Their study showed that the mean for the breadth search query is slightly higher compared to the depth search query, which indicated that users mostly utilize the breadth search query in their Web searches. However, small differences between the breadth search and depth search show that users tend to improve their search with depth queries. That is, users switch their strategy from broad to narrow to achieve their information needs. Therefore, these related studies have shown the importance of designing a useful and effective system for a better user experience based on limited queries and allowing users to easily reformulate second or subsequent queries.

Related prototype interfaces

In the literature, many domain-specific prototypes have been developed. For example, in the field of e-commerce, He *et al.* (2003) presented the WISE-Integrator tool to provide unified access to multiple e-commerce search engines to allow users to search for and compare products from multiple sites. In particular, it performs automatic integration of the Web interfaces of different search engines by using sophisticated techniques to identify matching attributes. The existing general purpose search engines or existing code search engines cannot meet the current needs of programmers who regularly search out relevant information about an appropriate API for a problem as part of the development process. To remedy this lack, Hoffmann *et al.* (2007) presented Assieme, a Web search interface that effectively supports common programming search tasks by combining information from Web-accessible Java Archive files, API documentation and pages that include explanatory text and sample code.

When users have poorly defined or complex goals, general search interfaces offering only keyword searching facilities are unlikely to provide adequate support to help them reach their information-seeking objectives. To this end, more advanced capabilities such as faceted browsing and result clustering can be integrated into the search interfaces (Wilson *et al.*, 2009). Wilson *et al.* (2010) reviewed various search systems in terms of visualization and exploration techniques and specific individual features that can enhance the usability of search interfaces. They also pointed out the importance of developing more effective search interfaces by incorporating novel interactions and features (or functions) that enable users to conveniently visualize, parse, manipulate and organize their Web search results.

Recently, Alhenshiri *et al.* (2010) presented an interactive visual search engine (VSE) for visualizing both processes of query reformulation and search results. In VSE, an alternate query is generated based on the user's query. That is, a query term is augmented with additional terms to create an alternate query using the semantic network WordNet. Moreover, VSE infers keywords and complete phrases from the search results. These query components are presented along with the research results. As a result, the user can select terms and phrases in the reformulation area to build further queries. On the other hand, Kaczmarek (2011) proposed a clustering-by-directions algorithm for interactive query expansion. It aims at supporting users of search engines in forming Web search queries. Particularly, when a user executes a query, the algorithm shows potential directions in which the search can be continued.

For Web personalization, as users often store, tag and organize useful URLs inside their information spaces or to social bookmarking sites, these collections of Web-related data can be combined to produce a view of the Web from the user's perspective. Consequently, Papagelis and Zaroliagis (2012) proposed a peer-to-peer, bottom-up search engine that can provide search results by combining the user's preferences regarding web pages. Srivastava *et al.* (2010) introduced a novel interface, namely, share-ken, to improve the accuracy of search engines for a particular search string (or query) which has been used before by other users. It is based on the idea of social collaboration among users to share their knowledge and ken.

Boydell and Smyth (2010) designed a novel recommendation interface for collaborative web search. A community-based approach is introduced, which is to recommend the results that are especially relevant to a community of users including community-focused result snippets and composite result summaries. Specifically, the screen of the interface is divided into three parts, which are the Web results returned from a particular search engine, the promoted results as the results returned from the community recommendations and the promoted summary, which is a composite social summary based on a concise overview of the recommended results and the community preferences and interests for the query.

SAFQuery: the prototype

From the literature review, the authors realized that the development of simple and flexible tools for advanced Web searching is necessary to better support human information-seeking behaviour. In this study, the Google search interface is targeted for further improvement, as it is the largest and most popular search engine today and advanced queries are available using the Google Advanced Search interface. Therefore, the Google API was used to develop the prototype, SAFQuery.

Interface design

The main goal of designing SAFQuery is to provide users with a simple and flexible (advanced) query interface without a large memory load. Therefore, the graphical user interface (GUI) has the following two important features:

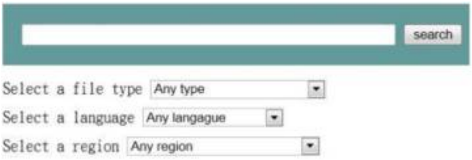
(1) The integration of simple and advanced query strategies. This integration design is much more flexible than designing two individual tools for simple and advanced queries, respectively. As a result, users can easily issue simple and advanced queries based on one single query interface, interchangeably. For instance, users may issue a simple query q_1 and then after looking at the initial retrieval results, they may have a more specific information need corresponding to q_1. In this case, users generally need to go to another interface for advanced search where they need to re-type q_1 as the query and input other query specifications for q_1. This advanced query process seems tedious. However, in the current integration design, users simply need to issue an advanced query based on q_1 by providing some query specifications. Consequently, the search process can be simplified.

(2) The support of query history for efficient re-querying. It is common that after several query iterations for an information need related to q_1, users may have already issued numbers of queries related to and/or query specifications for q_1 (represented by q_{1_1}, q_{1_2} …, q_{1_n}) or even some other queries (represented by q_2, q_3 and q_4). At this time, users have received a lot of information from q_{1_i} ($i = 1, 2 \ldots, n$) and q_j ($j = 2$, 3 and 4). In some circumstances, they may want to re-evaluate the retrieval results of q_{1_i} and/or q_j as they may come to be identified as the key queries after several rounds of queries. In this case, we specifically design a table listing the user's query history to be included in the query interface. Users can easily input previously issued queries without spending time to recall what the queries are and/or to re-type previous queries.

Query interface and retrieval results

Taking into account the interface design issues described previously, the prototype is based on Microsoft ASP.NET. Figure 1 shows the default query interface of SAFQuery, which contains a query block and three query specifications. In other words, it provides both the simple and advanced query strategies. Note that, for simplicity, only three important query specifications provided in Google Advanced Search were selected to embed in SAFQuery. The researchers believe that three are enough to assess the usefulness of SAFQuery.

Figure 2 shows an example of the retrieval results by SAFQuery. Block A contains the information of the query and query specifications. Block B shows a table listing the query history. In particular, 13 queries and query specifications were issued in this

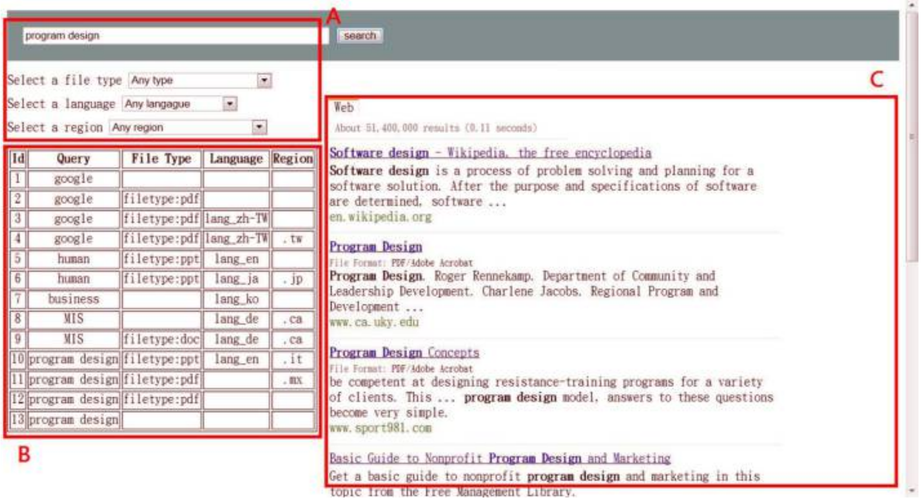


Select a file type Any type

Select a language Any langague

Select a region Any region

Figure 1.
Query interface on SAFQuery



program design

Select a file type Any type

Select a language Any langague

Select a region Any region

Id	Query	File Type	Language	Region
1	google			
2	google	filetype:pdf		
3	google	filetype:pdf	lang_zh-TW	
4	google	filetype:pdf	lang_zh-TW	.tw
5	human	filetype:ppt	lang_en	
6	human	filetype:ppt	lang_ja	.jp
7	business		lang_ko	
8	MIS		lang_de	.ca
9	MIS	filetype:doc	lang_de	.ca
10	program design	filetype:ppt	lang_en	.it
11	program design	filetype:pdf		.mx
12	program design	filetype:pdf		
13	program design			

Web

About 51,400,000 results (0.11 seconds)

Software design - Wikipedia, the free encyclopedia
Software design is a process of problem solving and planning for a software solution. After the purpose and specifications of software are determined, software ...
en.wikipedia.org

Program Design
File Format: PDF/Adobe Acrobat
Program Design. Roger Rennekamp. Department of Community and Leadership Development. Charlene Jacobs. Regional Program and Development ...
www.ca.uky.edu

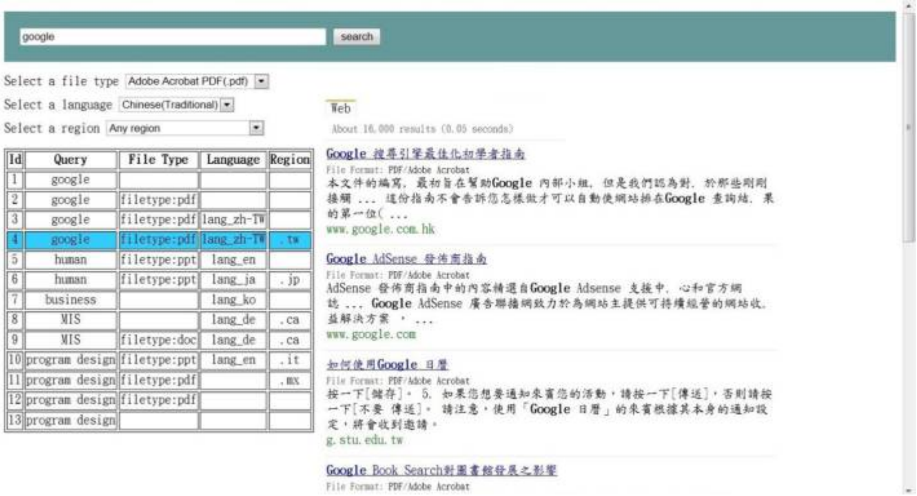
Program Design Concepts
File Format: PDF/Adobe Acrobat
be competent at designing resistance-training programs for a variety of clients. This ... **program design** model, answers to these questions become very simple.
www.sport981.com

Basic Guide to Nonprofit Program Design and Marketing
Get a basic guide to nonprofit **program design** and marketing in this topic from the Free Management Library.

Figure 2.
An example of the retrieval results on the interface

example. Block C displays the retrieval results. The results in this example are based on the 13th query listed, "program design".

When users want to re-issue previous queries and query specifications, they simply need to click on a specific query ID in Block B (Figure 2) for this search. Figure 3 shows an example of issuing the fourth query and its retrieval results. In this case, SAFQuery automatically shows the query term and query specifications of the fourth query in Block A (Figure 2). This can facilitate the issuing of new queries or query specifications related to the fourth query when necessary.



google

Select a file type Adobe Acrobat PDF(.pdf)

Select a language Chinese(Traditional)

Select a region Any region

Id	Query	File Type	Language	Region
1	google			
2	google	filetype:pdf		
3	google	filetype:pdf	lang_zh-TW	
4	google	filetype:pdf	lang_zh-TW	.tw
5	human	filetype:ppt	lang_en	
6	human	filetype:ppt	lang_ja	.jp
7	business		lang_ko	
8	MIS		lang_de	.ca
9	MIS	filetype:doc	lang_de	.ca
10	program design	filetype:ppt	lang_en	.it
11	program design	filetype:pdf		.mx
12	program design	filetype:pdf		
13	program design			

Web

About 16,000 results (0.05 seconds)

Google 搜尋引擎最佳化初學者指南
File Format: PDF/Adobe Acrobat
本文件的編寫，最初旨在幫助**Google** 內部小組，但是我們認為對，於那些剛剛接觸 ... 這份指南不會告訴您怎樣做才可以自動使網站排在**Google** 查詢結，果的第一位(...
www.google.com.hk

Google AdSense 發佈商指南
File Format: PDF/Adobe Acrobat
AdSense 發佈商指南中的內容精選自**Google** Adsense 支援中，心和官方網誌 ... **Google** AdSense 廣告聯播網致力於為網站主提供可持續經營的網站收，益解決方案 。 ...
www.google.com

如何使用Google 日曆
File Format: PDF/Adobe Acrobat
按一下[儲存]。 5. 如果您想要通知來賓您的活動，請按一下[傳送]，否則請按一下[不要 傳送]。 請注意，使用「**Google** 日曆」的來賓根據其本身的通知設定，將會收到邀請。
g.stu.edu.tw

Google Book Search對圖書館發展之影響
File Format: PDF/Adobe Acrobat

Figure 3.
An example of retrieving the fourth previous query

Discussion

Using the current interface tools to issue simple and advanced queries is not simple and flexible enough. Taking the Google search engine as an example, the search process is as follows:

- *For simple queries*: Go to the default query interface → type a query → search → receive the results → type the second query and so on.
- *For advanced queries*: Go to the advanced query interface → type a query → input query specifications → search → receive the results → type the second query → input query specifications and so on.

After issuing several simple and/or advanced queries, the user might want to re-issue previous (advanced) queries or want to check on what they have queried before, but to do this, they must repeat the search process by re-typing a query and so on. This may not be possible due to the problem of remembering previous queries. However, in SAFQuery, the search process makes the query task simpler and more flexible as follows:

- *For simple and advanced queries*: Go to the default query interface → type a query → input query specifications if needed → search → receive the results → type the second query and so on.

For re-issuing previous queries, users just need to view the query history table in the query interface and then click on the issued query (without re-typing the query). In addition, users can modify the query or query specifications related to the clicked query in the same query interface. Therefore, this makes the tasks of issuing simple and advanced queries much simpler and more flexible.

System evaluation

Experimental setup

To evaluate the usefulness of SAFQuery, the authors designed a questionnaire utilizing the QUIS (Questionnaire for User Interface Satisfaction) methodology (Chin *et al.*, 1988). The questions can be categorized into five groups, namely, "overall reactions to the system", "screen", "terminology and system information", "learning" and "system capabilities". Note that some questions which are not suitable for the Web search scenario are not considered.

Ten human subjects, including six males and four females, were asked to participate in the system evaluation. They were graduate students at the Department of Management Information Systems. They were familiar with Web searching using Google, normally spending over four hours per week searching for information with Google. Before performing system evaluations, respondents were allowed to use SAFQuery for one week.

Experimental results

Overall reactions to the system. Table I shows the results of overall reactions to SAFQuery. These indicate that most users had a good experience using SAFQuery (94 per cent). In contrast, 4 per cent of users thought that using SAFQuery was not as flexible and it made them feel some degree of frustration.

Screen. Table II shows the human evaluation of several screen-related issues for SAFQuery. As we can see, 97.5 per cent of users thought the interface to be easy to read and information organization very clear. Moreover, the most important judgement is that it could simplify the query tasks.

Terminology and system information. Table III shows the results related to terminology and system information. These indicate that most users (87.5 per cent) agree that the terms used throughout the interface, the position of messages on the interface and so forth are consistent. Furthermore, most users think that the system keeps them informed about what it is doing.

Learning. Table IV shows the results of the learning issues for SAFQuery. Again, most (92.5 per cent) felt that the interface was easy to use for Web search purposes. This

Negative experience	0	1	2	3	4	5	6	7	8	9	Positive experience
Terrible	0	0	0	0	0	2	5	3	0	0	Wonderful
Difficult	0	0	0	0	0	1	2	5	0	2	Easy
Frustrating	0	0	0	0	1	0	6	3	0	0	Satisfying
Inadequate support	0	0	0	0	0	2	4	3	1	0	Adequate support
Rigid	0	0	0	1	1	1	2	4	1	0	Flexible
Average (%)	0	0	0	2	4	12	38	36	4	4	

Table I. Results showing overall reactions to the system

Negative experience	0	1	2	3	4	5	6	7	8	9	Positive experience
Characters on the computer screen											
Hard to read	0	1	0	0	0	2	1	3	2	1	Easy to read
Highlighting on the screen simplifies the task											
Not at all	0	0	0	0	0	0	5	5	0	0	Very much
Organisation of information on the screen											
Confusing	0	0	0	0	0	1	5	1	2	1	Very clear
Sequence of screens											
Confusing	0	0	0	0	0	0	4	3	3	0	Very clear
Average (%)	0	2.5	0	0	0	7.5	37.5	30	17.5	5	

Table II. Results on the screen interface

Negative experience	0	1	2	3	4	5	6	7	8	9	Positive experience
Use of terms throughout the system											
Inconsistent	0	1	0	0	1	5	3	1	0	0	Consistent
Computer terminology is related to the task at hand											
Never	0	0	1	0	0	1	5	1	2	0	Always
Position of messages on the screen											
Inconsistent	0	0	0	1	1	2	4	1	1	0	Consistent
Computer keeps you informed about what it is doing											
Never	0	0	0	0	1	3	2	3	1	0	Always
Average (%)	0	2.5	2.5	0	7.5	27.5	35	15	10	0	

Table III. Results related to terminology and system information

Table IV. Results related to learning

Negative experience	0	1	2	3	4	5	6	7	8	9	Positive experience
Learning to operate the system											
Difficult	0	0	0	1	0	1	4	1	1	2	Easy
Exploring new features by trial and error											
Difficult	1	0	1	0	0	2	1	4	1	0	Easy
Remembering names and use of commands											
Difficult	0	0	0	0	0	1	1	3	2	3	Easy
Tasks can be performed in a straight-forward manner											
Never	0	0	0	0	0	1	1	5	1	2	Always
Average (%)	2.5	0	2.5	2.5	0	12.5	17.5	32.5	12.5	17.5	

may be because these human subjects are familiar with the Google search engine and they have an IT-related background.

System capabilities. Table V shows the results related to system capabilities, in which 95 per cent of users think that SAFQuery is fast in response to their queries and is reliable during Web searching. This suggests that the implementation of Google API in SAFQuery is successful.

Summary

From the Cronbach α test, we obtain a value of 0.826, which means that the evaluation results are highly reliable. We further analyse the strengths and weaknesses of SAFQuery. Table VI shows the average scores from the questionnaire. On average, the score of SAFQuery is 6.4, which is an encouraging result. Note that higher than average scores are underlined.

The results in Table VI confirm the usability of SAFQuery, which allows users to easily search the Web for information. In other words, SAFQuery can achieve our research objective, making the Web search simpler and more flexible.

On the other hand, by looking at the scores that are certainly below the average, smaller than 6, we find some issues of concern for future improvement. For example, the use of term and message positions on the interface should be carefully designed. This may be an important factor for non-English speakers or for some users who prefer looking at the retrieval results in a specific area or a larger area, and so on. This means that a better arrangement of Blocks A, B and C and better use of terms for each block are necessary.

Table V. Results for system capabilities

Negative experience	0	1	2	3	4	5	6	7	8	9	Positive experience
System speed											
Too slow	0	0	0	0	0	0	2	3	4	1	Fast enough
System reliability											
Unreliable	0	0	0	1	0	1	1	3	3	1	Reliable
Average (%)	0	0	0	5	0	5	15	30	35	10	

Evaluation items	Scores
Overall reactions to the system	
Terrible/wonderful	6.1
Difficult/easy	7
Frustrating/satisfying	6.1
Inadequate/adequate support for advanced Web search	6.3
Rigid/flexible	6
Screen	
Characters on the computer screen: hard to read/easy to read	6.2
Highlighting on the screen simplifies tasks: not at all/very much	6.5
Organisation of information on screen: confusing/very clear	6.7
Sequence of screens: confusing/very clear	6.9
Terminology and system information	
Use of terms throughout system: inconsistent/consistent	5.4
Computer terminology is related to the task you are doing: never/always	6
Position of messages on the screen: inconsistent/consistent	5.6
Computer keeps you informed about what it is doing: never/always	6
Learning	
Learning to operate the system: difficult/easy	6.5
Exploring new features by trial and error: difficult/easy	5.2
Remembering names and use of commands: difficult/easy	7.6
Tasks can be performed in a straight-forward manner: never/always	7.2
System capabilities	
System speed: too slow/fast enough	7.4
System reliability: unreliable/reliable	6.8
Average	6.4

Note: the underline level of significance is $p < 0.05$ based on the t-test

Table VI. Average scores on the questionnaire

For the question about “exploring new features by trial and error”, although the score was 5.2, it does not mean that SAFQuery cannot easily provide users with some new features. Instead, as the main features of SAFQuery are based on the integration of simple and advanced queries and the query history support for re-queries in a single interface, there is no new feature(s) of SAFQuery that can be explored. However, one implicit weakness of SAFQuery in relation to this issue is that first-time users may not realize that the queries listed in the query history table can be clicked on.

Comparison with Google

Furthermore, participants were asked to compare SAFQuery with Google in terms of Web search usability. Table VII shows the average scores from the questionnaire. Note that scores higher than the average score are underlined.

Overall, users agree that using SAFQuery to search the Web is more useful and flexible than using the Google search interface. In addition, SAFQuery provides additional search supports, including advanced query and query history functionalities, which allow users to search the Web more efficiently than using Google.

Table VII. Average scores for the comparison with Google

Evaluation items	Scores
Overall reactions to the system	
I am more satisfied with the search functionalities provided in SAFQuery	6.2
I think that SAFQuery is more useful in searching the Web	7.8
I think that using SAFQuery to search the Web is more flexible	7.6
I think that SAFQuery provides more supports for efficient Web search	7.5
Search functionalities	
I am more satisfied with integrating the simple and advanced query functionalities into a single Web search interface	5.7
I am more satisfied with integrating the query history functionality into a single Web search interface	6.1
I think that the Web search interface providing both simple and advanced query functionalities is more useful	7.3
I think that the Web search interface providing the query history functionality is more useful	7.5

Note: the underline level of significance is $p < 0.05$ based on the t-test

Despite the fact that SAFQuery can make Web searching simpler and more flexible than Google does, some evaluation results are lower than the average score. That is, most participants have the same comment on SAFQuery about the arrangement of the three blocks in the GUI of SAFQuery. Particularly, they suggest that it would be better if the query specifications in the advanced query functionality could be hidden when users do not want to issue the advanced query. This is because users do not perform advanced Web searches all the time. Consequently, this makes the GUI simpler and cleaner, allowing more space to display search results.

On the other hand, although the query history functionality is very useful, it was suggested to provide a better structured table with more information, such as the dates and frequencies of issuing specific queries. In addition, similar to the advanced query functionality, the query history should be closed as the default setting, and users can open it when needed. This comparison indicates that most users are used to the Google search interface with its arrangement of the query and result display blocks. However, users would have a better experience using the same interface, plus the supporting functionalities proposed in this paper. This reveals that these supporting functionalities should be carefully integrated into the interface. This issue is discussed further in the conclusion.

Conclusion

The increasing popularity of using Web search engines to discover and locate information has made the user experience with the search interface important for successful and/or satisfied information access. Successful information access by Web searching involves an iterative process, which can contain many steps to fulfil a user's information needs. This process is not made simple in current interfaces. Moreover, they require a certain memory load that it would be good to reduce, such that it would alleviate the need to remember information over a period.

To this end, the authors introduce a prototype system, namely, SAFQuery (Simple and Flexible Query interface). This prototype integrates both simple and advanced query strategies into a single interface and provides query history information that allows users to reuse previous queries easily. The user evaluation experiments show that most users had a positive experience using SAFQuery. Specifically, users found it easy to use and able to simplify the Web search task.

Although SAFQuery allows users to search the Web in a simple and flexible manner, some important GUI design issues can be further improved upon in the future. The first one is that the query specification area and the query history table should be able to be hidden if so desired, or even moved dynamically to any position the users want. The second issue is to modify the way query specifications are input. For example, for the region field, the general layout displays all possible regions in a list box. However, this is not very useful because there are many regions which can be selected and the user is currently limited to a single choice. Therefore, one possible solution is to design a pop-up window as a visualization-based query specification that displays a map from which users can choose single or multiple regions for this kind of query specification.

References

Alhenshiri, A., Watters, C., Brooks, S. and Shepherd, M. (2010), "Augmenting the visual presentation of web search results", *International Conference on Digital Information Management*, Thunder Bay, ON, pp. 101-107.

Baeza-Yates, R. and Ribeiro-Neto, B. (1999), *Modern Information Retrieval*, Addison Wesley, Boston, MA.

Bar-Ilan, J. (2004), "The use of web search engines in information science research", *Annual Review of Information Science and Technology*, Vol. 38 No. 1, pp. 231-288.

Berghel, H. (1997), "Cyberspace 2000: dealing with information overload", *Communications of the ACM*, Vol. 40 No. 2, pp. 19-24.

Boydell, O. and Smyth, B. (2010), "Social summarization in collaborative web search", *Information Processing & Management*, Vol. 46 No. 6, pp. 782-798.

Broder, A. (2002), "A taxonomy of web search", *ACM SIGIR Forum*, Vol. 36 No. 2, pp. 3-10.

Chin, J.P., Diehl, V.A. and Norman, K.L. (1988), "Development of an instrument measuring user satisfaction of the human-computer interface", *ACM SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, pp. 213-218.

Costa, M. and Silva, M.J. (2011), "Characterizing search behavior in web archives", *International Temporal Web Analytics Workshop*, Hyderabad, pp. 33-40.

Gordon, M. and Pathak, P. (1999), "Finding information on the World Wide Web: the retrieval effectiveness of search engines", *Information Processing & Management*, Vol. 35 No. 2, pp. 141-180.

Hassenzahl, M. and Tractinsky, N. (2006), "User experience – a research agenda", *Behaviour and Information Technology*, Vol. 25 No. 2, pp. 91-97.

He, H., Meng, W., Yu, C. and Wu, Z. (2003), "WISE-Integrator: an automatic integrator of web search interfaces for e-commerce", *International Conference on Very Large Data Bases, Trondheim*, pp. 357-368.

Heimonen, T., Aula, A., Hutchinson, H. and Granka, L. (2008), "Comparing the user experience of search user interface designs", *ACM Conference on Human Factors in Computing Systems, Florence*, pp. 1-4.

Hoffmann, R., Fogarty, J. and Weld, D.S. (2007), "Assieme: finding and leveraging implicit references in a web search interface for programmers", *ACM Symposium on User Interface Software and Technology*, New York, NY, pp. 13-22.

Kaczmarek, A.L. (2011), "Interactive query expansion with the use of clustering-by-directions algorithm", *IEEE Transactions on Industrial Electronics*, Vol. 58 No. 8, pp. 3168-3173.

Lawrence, S. and Giles, C.L. (1999), "Accessibility of information on the web", *Nature*, Vol. 400 No. 107, pp. 107-109.

Markey, K. (2007), "Twenty-five years of end-user searching, part 1: research findings", *Journal of American Society for Information Science and Technology*, Vol. 58 No. 8, pp. 1071-1081.

Papagelis, A. and Zaroliagis, C. (2012), "A collaborative decentralized approach to web search", *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, Vol. 42 No. 5, pp. 1271-1290.

Spink, A. and Jansen, B.J. (2004), "A study of web search trends", *Webology*, Vol. 1 No. 2, p. 4.

Spink, A., Jansen, B.J., Wolfram, D. and Saracevic, T. (2002), "From e-sex to e-commerce: web search changes", *IEEE Computer*, Vol. 35 No. 3, pp. 133-135.

Srivastava, V., Kollipara, S.V., Tyagi, V. and Jaiswal, R. (2010), "Share-ken: a way to improve web search", *International Conference on Recent Trends in Information, Telecommunication and Computing*, Kochi, pp. 327-329.

Weber, I. and Castillo, C. (2010), "The demographics of web search", *ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, NY, pp. 523-530.

White, R.W., Dumais, S.T. and Teevan, J. (2009), "Characterizing the influence of domain expertise on web search behaviour", *ACM Conference on Web Search and Web Data Mining*, New York, NY, pp. 132-141.

Wildemuth, B.M. (2004), "The effects of domain knowledge on search tactic formulation", *Journal of the American Society for Information Science and Technology*, Vol. 55 No. 3, pp. 246-258.

Wilson, M.L., Kules, B., Schraefel, M.C. and Shneiderman, B. (2010), "From keyword search to exploration: designing future search interfaces for the web", *Foundations and Trends in Web Science*, Vol. 2 No. 1, pp. 1-97.

Wilson, M.L., Kules, B., Schraefel, M.C. and White, R.W. (2009), "Evaluating advanced search interfaces using established information-seeking models", *Journal of the American Society for Information Science and Technology*, Vol. 60 No. 7, pp. 1407-1422.

Yamin, F.M. and Ramayah, T. (2011), "User web search behavior on query formulation", *International Conference on Semantic Technology and Information Retrieval*, Putrajaya, pp. 182-188.

Corresponding author
Chih-Fong Tsai can be contacted at: cftsai@mgt.ncu.edu.tw