An Opportunity Cost Approach for Measuring Performance of An Intelligent Knowledge Based Search Assistant

Md. Mahbubul Alam Joarder, Khaled Mahmud, and Bulbul Ahamed

***Abstract*— The Web has myriad of useful information, but its dynamic, unstructured nature makes them difficult to locate the desired information. A general-purpose search engine, such as Google or AltaVista usually generates thousands of hits, many of them irrelevant to the user query. A knowledge based search assistant is developed which reduces the time and cost of information accumulating of common interest groups. When the users from a common network search on similar topics then an intelligent agent minimize the searching effort of a user by utilizing the previous experience of the users they have gathered from their surfing behaviours. Additionally the agent incrementally updates its database by analyzing its perception, which gradually increases its recall rate. A search assistant that accumulates knowledge from user activity and gathers information would provide a convenient searching environment with minimum effort within shortest possible time.**

***Index Terms*— Personalized Searching, Web User Satisfaction, Keyword Clustering, Computer Support Cooperative Work (CSCW), Search Assistant.**

I. Introduction

Searching on a specific topic in the Web is a difficult task because of having plenty of information throughout the whole world. Search engines help, but the number of Web pages now exceeds two billion, making it difficult for general purpose search engines to maintain comprehensive, up to date search indexes. Moreover, as the Web grows ever larger, so does information overload in query results. A surfer requires traversing unnecessary spots of information before finding out the specific field of interest. The procedure one passes is repeated by another user whenever a search for a similar type of topic or keyword has appeared. This searching would not be necessary if the user knew someone with similar search criteria whose search-request has been satisfied with a result that is hiding within the pages returned by some popular search engines. This situation is very frequent in an organization where many people have similar interests. Students, faculty members and researchers of a university, corporate personnel in a corporate office, and doctors in hospitals form some common user group that would result in common interest in searching. The searching experience of one surfer can be used for future users to eliminate some mundane repeated procedures for a group of people.

This paper introduces a knowledge based search assistant for a set of users that is implemented through a server agent and a client agent. The server agent scrutinizes the proxy log for search requests to the search engines like Google, Yahoo or MSN. The search keywords, criteria along with the returned URLs are stored in a database. The browsing activity of a surfer is monitored and the liking or disliking towards a page is also measured. The liking of a particular user toward a web page would increase the rank of the URL that corresponds to a specific search keyword. On the contrary, disliking toward a web page would decrease the rank of the URL corresponding against the search topic. A new user making search request with similar keywords would be served with the already searched materials so that the searching effort is minimized.

Several works have been done for improving performance of web searching. Cabri et al. [1] proposed Supporting Cooperative WWW Browsing which is a proxy-based Approach. In this work they have designed a new proxy server, which support cooperative www browsing. Several approaches have already been taken to utilize users' browsing information to help other users. Alexa [2] is such a tool that gathers information from the users who install that toolkit. Group Asynchronous Browsing proposed by Wittenburg et al. [3] on the World Wide Web is a technique to gather peoples' favorite pages by retrieving Bookmark or hot list information from popular browsers and serves the collected data to the users through HTML pages.

II. System Architecture

The accumulated knowledge based search assistant divided into two components: the client component and the server component which is depicted in Figure 1. The client component resides in user machine which monitors the pages browsing and saving for a particular topic. The client component periodically sends the collected information to the server component. The server component accumulates the information perceived from different client component and apply clustering algorithm proposed by Seung-Shik [4] to cluster the similar URL for a particular search keyword and persist them in the database. In addition the server component sends the list of URLs to the client component

Md. Mahbubul Alam Joarder is Associate Professor of the Institute of Information Technology, University of Dhaka, Dhaka-1000, Bangladesh. (email: joarder@univdhaka.edu).

Khaled Mahmud is Lecturer of the Institute of Business Administration, University of Dhaka, Dhaka-1000, Bangladesh. (Phone: +8801712536013; e-mail: khaled@iba-du.edu).

Bulbul Ahamed is Senior Lecturer of the department of Computer Science and Engineering, Northern University Bangladesh. Dhaka- 1205, Bangladesh. (e-mail: bulbul2767@gmail.com).

and the client component is responsible for displaying the URLs when the user searches on the search engines.

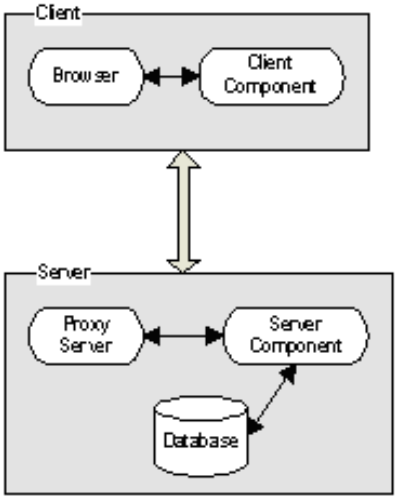

Fig. 1. High level system architecture.

A. Server Component

The server component is the core of the system, which manages the decision-making system. It consists of three modules as shown in Figure 2.

Keywords clustering module

Jussara and Cao [5] proposed the keywords clustering module which monitors the browsing log in the proxy server such as Squid [6]. Any URL users browse is checked if that belongs to any of the popular search engine query (Google, Yahoo, and MSN); the searching is analyzed to extract the keywords. In our implementation we use the squid proxy server and its access log file is monitored for that purpose. The extracted information conform the following data structure:

i) The originating IP address
ii) The time of the request
iii) The search keyword
iv) URLs returned by the search engine

The originating IP and the time of the requests are stored to distinguish the individual user and send them feedback of query. The keywords are periodically clustered using a clustering algorithm [7]. The clustering algorithm creates a cluster by the words of each search pattern (Term).

If n is the number of clusters in C, then

$C = \{ C1, C2, \ldots, Cn \}$

Each cluster C_i is initialized by Term, T that is not assigned to the existing cluster, and T is a seed Term or keyword of C_i . The center of each cluster is C_{center}.When a new cluster is created; expansion and reduction steps are repeated until it reaches a stable state from the start state. Word set, W_t of a Term T is a set of words $w_1, w_2 \ldots w_n$ that are extracted form the search pattern that is proxy log file.

$W_t = \{w|\ w$ is a word that is extracted from $T\}$

The clustering is necessary as many search keywords actually mean almost the same thing. Searching with "Artificial Intelligence" and "Artificial Intelligence Tutorial" actually represent almost the same query. One who looks for "Artificial Intelligence Tutorial" may be satisfied with a good page from search result of "Artificial Intelligence"; and vice versa. For this reason, resulting URLs are stored with respect to their originating search keyword cluster center, not the entire search topic.

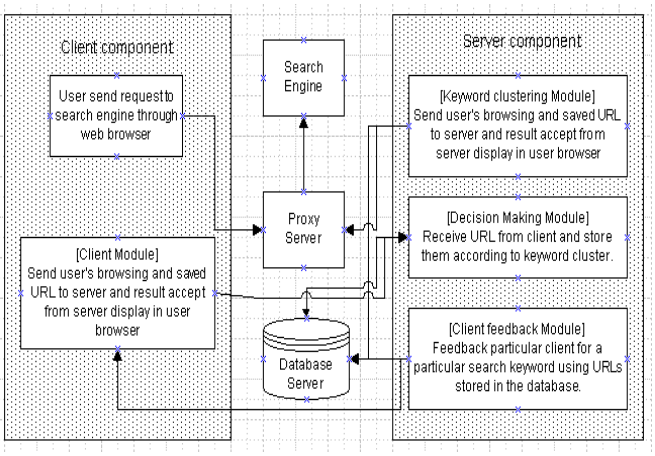

Fig. 2. Information flow in the system.

The clustering algorithm incrementally builds the different cluster at the time of users browsing. In this algorithm we always compare the similarity of the new keyword and center keyword of the existing cluster. The new keyword put into the highest similarity cluster and updates the center of the cluster. If the similarity of the new keyword is less than the threshold then it create a new cluster, put the new keyword into the new cluster and assign the center of this new cluster by this new keyword. Figure 3 illustrates the keyword-clustering algorithm.

Procedure Keyword Clustering(Term T_{new})
begin
for all C_i in C
find maxSim=sim (C_{center},T_{new})
[0<=s<=100]
if(maxSim>threshold) then
insert T_{new} into C_i and update center of C_i (C_{center})
else
create a new cluster C_{n+1} and insert T_{new} in C_{n+1}
and C_{center} = T_{new}
end

Fig. 3. keyword clustering algorithm.

Decision Making Module

The decision making module find out the URLs, which fall in a particular search keyword. For a given instant it is possible to send several queries to the server by the particular user and browse the desired URLs for the respective search pattern. This module generates a query to the respective search engine by the keyword that the user used and retrieves the URL list and stores them. When the client traverses the web pages, then this module matches the stored URLs and the URLs the client actually surfed. The matched URLs of the two lists are the original URLs the client actually browses for a particular keyword.

This module also determines rank of a particular URL. The rank of a URL indicates its relative importance and predilection of the user. Importance of a URL judges from the number of users visiting it, amount of time each user spends in the page or the tendency toward preserving the URL for future use. The search assistant formulates the

following rule to calculate the rank of a URL:

$$R(k, u) = \sum_{i} \{ L(k, u, i) + T(k, u, i) + S(k, u, i) \}$$

Here, $R(k, u)$ = The Rank of a URL u associated with a particular keyword k

$L(k, u, i)$=Whether a URL u has ever been browsed by an individual user i for the keyword k: returns 1 for yes, 0 for no

$T(k, u, i)$= Time in minute a URL u has been browsed by an individual user i for the keyword k

$S(k, u, i)$ = Whether a page having URL u has ever been saved by an individual user i for the keyword k: returns 1 for yes, 0 for no.

If the page is saved then

$S(k, u, i) / T(k, u, i) = 5$

In experiment we provide more importance on page saving event and consider it 5 times of browsing time.

Client feedback Module

The client feedback module finds the highest-ranking URLs from the database for a particular term or search pattern and send it to the client when user initiate search to the search engine through the proxy server. First it finds the highest similar cluster for the keyword stored in the database. Now it finds the URL list for the keyword from the cluster and sorts them in descending order according to their ranking and sends back to client. The client feedback module algorithm works according to the Figure 4.

```
Procedure Client Feedback (Term Tx)
Begin
  For ∀ Ci in database
    Find the highest similar cluster for Tx
        Find the highest ranking list, L from Ti
        for which similarity greater than
        threshold. After
        Sorting the list in descending order
        according to ranking and return the
        URL list
End
```

Fig. 4. Procedure for client feedback module

B. *Client Component*

The client agent continuously monitors the user activities and sends gathered information like current active browser URL and the URL of the pages that the user save to the disk.. Active browser URL is necessary for the server to measure the time a user spends on a particular page. This URL in turn could be associated with a keyword that was used when the search engine was employed. If the URL can be associated with such a keyword, calculation for ranking the page is performed by the server. The saved pages represents higher importance as users usually save pages that are important and may be needed in future use.

III. Performance Cost Model

For measuring the search cost, we designed a cost function for both search assistant and traditional search engine which will measure the performance cost in time.

During the traditional search query execution process, for each query term Q_T, user will get a set of URLs, $\boldsymbol{U_t = \{u_1, u_2, u_3 \dots u_n\}}$

Beside, the knowledge base search assistant query execution process, for the query term $\boldsymbol{T}$, user will get a set of URLs, $\boldsymbol{U_k = \{u_1, u_2, u_3 \dots u_r\}}$

And the user desired URLs set is $\boldsymbol{U_d = \{u_1, u_2, u_3 \dots u_j\}}$

The URLs list of $\boldsymbol{U_k}$ is populated in such a way that $\boldsymbol{U_k}$ is a subset of $\boldsymbol{U_t}$ so that $\boldsymbol{U_k \subseteq U_t}$.

And $\boldsymbol{U_d \subseteq U_t}$ and is also $\boldsymbol{U_d \subseteq U_k}$ is also true.

Using traditional search engine, for a query term Q_T, the total time required to find the desired URLs set $\boldsymbol{U_d}$ from URLs set $\boldsymbol{U_t}$ is calculated by the following equation

$$F_t(Q_T) = CT(Q_T) + ET(Q_T) + \sum_{u \in Ut} t_u$$

Where, CT is communication time for the query measure by as follows:

$CT(Q_T) = T_{req}(Q_T) + T_{res}(Q_T)$

Here, $T_{req}(Q_T)$ and $T_{res}(Q_T)$ are request and response time respectively.

Again, ET is query execution time for the search engine for query term, Q_T.

Finally, t_u is time spent in each element of URLs set U_t.

Similarly, for our knowledge base search assistance, for query term Q_T, the total time required to find the desired URLs set $\boldsymbol{U_d}$ from URLs set $\boldsymbol{U_k}$ is calculated by the following equation

$$F_k(Q_T) = CT(Q_T) + ET(Q_T) + \sum_{u \in Uk} t_u$$

Finally, the Opportunity Cost [8] of using traditional search engine instead of our knowledge base search agent is calculated by the following equation:

$OC(Q_T) = F_t(Q_T) - F_k(Q_T)$

IV. Performance Measurement

The performance of any type of search engine depends on user satisfaction. As there is no benchmarking dataset for measure the performance of these kind of search agent. So a data set is populated to measure the performance of the agent.

Initially some user groups are created, where the members of each group always search for similar type of subjects. After that, the members of each group are divided in to two parts, where each part contains 50% of the group members. First 50% of group members are used to train up the search agent knowledge base. Finally the rest of the members are used to test the performance of the agent.

For performance testing, the remaining group members are divided in 1:2 ratios. First half is tested using traditional search engine and the rest of the members are tested using our search agent for similar type of key word for each group.

Table 1 depicts the sample query term and the time required finding desired information with and without using the search assistant. From Figure 5 we see that when a user uses the search assistant, it takes less time to find desire information and the opportunity cost is high if use general purpose search engine instead of search agent.

TABLE I
TIME REQUIRE TO SEARCH IN DIFFERENT CLUSTER

Group no	Query Term	Time required without agent	Time required using agent
1	machine learning	1h:20min	0h:17min
2	Java	0h:35min	0h:08min
3	microarray	0h:50min	0h:10min
4	neural network	0h:40min	0h:05min
5	J2EE tutorial	0h:30min	0h:04min
6	microcontroller project	0h:55min	0h:07min

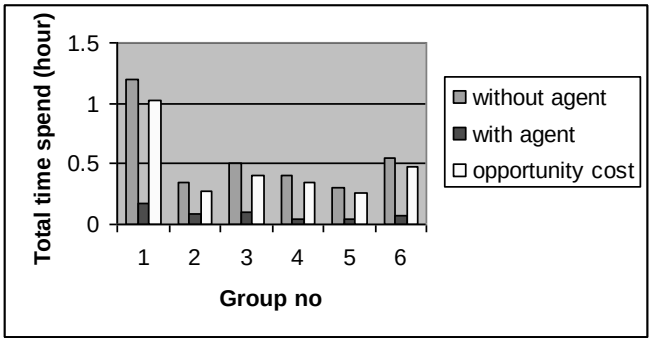

Fig. 5. Comparative performance of search assistance with general purpose search engines with opportunity cost.

V. CONCLUSION

Accumulated knowledge based search assistant gathers knowledge from user searching behaviors that provides a convenient searching environment with minimum effort within a shortest possible time. Introduction of the server agent has centralized the searching capability of a large group of users. Search assistant provides a global depository of knowledge for future use. Users of common interest need not go through a monotonous and mundane search procedure.

REFERENCES

[1] G. Cabri, L. Leonardi and F. Zambonelli. "Supporting Cooperative WWW Browsing: a Proxy-based Approach." Proceedings of the 7th Euromicro Workshop on Parallel and Distributed Processing. Madeira (P), IEEE CS Press, pp. 138–145.1999.

[2] Alexa. 2000. Available: url: http://www.alexa.com

[3] K. Wittenburg, D. Das, W. Hill and L. Stead. "Group asynchronous browsing on the World Wide Web". Proceedings of the 4th International WWW. 1995.

[4] K. Seung-Shik. "Keyword-based Document Clustering". Proceedings of the sixth international workshop on Information retrieval with Asian languages - Volume 11. Sappro, Japan, pp. 132 – 137. 2003.

[5] A. Jussara, and P. Cao. "Measuring proxy performance with the Wisconsin Proxy benchmark". Journal of Computer Networks and ISDN Systems. Volume 30, Issues 22-23, pp. 2179-2192. 1998.

[6] Squid. Open source proxy server. 1996. Available: http://www.squid-cache.org.

[7] D. Koller and M. Sahami. Hierarchically classifying documents using very few words. Proceedings of the Fourteenth International Conference on Machine Learning, Morgan Kaufmann Publishers Inc. San Francisco, CA, pp. 170 – 178. 1997.

[8] McConnell, C. Microeconomics: Principles, Problems, and Policies. McGraw-Hill Professional. ISBN 0072875615. 2005.

Md. Mahbubul Alam Joarder was born in 1968 at Kushtia, Bangladesh. He had his BSc (Hons) in Applied Physics and Electronics from University of Dhaka. He had his MSc from the same discipline. He completed his Phd from Ibaraki University, Japan from Dept of Computer Science and Engineering.

Now he is working as Director of the Institute of Information Technology, University of Dhaka. Prior to joining University of Dhaka he was working in Institute of Scientific Instrumentation (I.S.I), UGC, Agargoan, Sher-e-bangla Nagar, Dhaka. He also worked as PiL Advisor, Microsoft Bangladesh Ltd. He is also working as academic advisor of renowned private universities. He is also working jointly with JICA, KOICA and many other esteemed multinational companies. He has research interest in technology, business, e-learning, e-governance and social issues. He has published his articles in prestigious journals and conferences in home and abroad. He has two books published: "Introduction to C and X Window Programming - Guiding Very Beginners to Developer Level" and "Text Book on ICT for class IX & X" Approved by National Curriculum of Text Book (NCTB), Ministry of Education, Government of The Peoples Republic of Bangladesh.

He is member of Applied Synergetic Group (Japan)(since 1998), Bangladesh Computer Society (BCS). He is also member, Editorial Board, Journal of Computer Science. He is also House tutor of Salimullah Muslim Hall, Dhaka University, Bangladesh. He is working as moderator, Debating Club , University of Dhaka. He is the Chairman of Job placement, IIT, University of Dhaka.

Khaled Mahmud was born in 1984 at Pabna, Bangladesh. He was graduated from Bangladesh University of Engineering and Technology (BUET) in Computer Science and Engineering. He had his MBA (Marketing) from Institute of Business Administration, University of Dhaka. He was awarded gold medals both in his secondary and higher secondary school level for excellent academic performance.

He is now working as Lecturer at Institute of Business Administration, University of Dhaka. He previously worked as Assistant Manager in Standard Chartered Bank. Prior to Standard Chartered he was working in TwinMOS Technology Bangladesh. He has research interest business, technology, e-learning, e-governance, human resource management and social issues. He has his articles published in journals and conferences of USA, Canada, Australia, Malaysia, Thailand, South Korea and Bangladesh. He has one book published: "Strategic Alliances in the Software Industry" (LAP Lambert Academic Publishing AG & Co. KG, 2011).

Bulbul Ahamed was born in 1982 at Munshiganj district, Bangladesh. He has completed his Bachelor in Computer Science and Engineering from Northern University Bangladesh. He has completed his MBA from the same university (MIS & Markting). Now he is pursuing his MSc in Computer Science and Engineering in United International University, Bangladesh.

He is now working as senior lecturer in the Northern University Bangladesh. He has research interest in Business and Information Technology. He has published articles in the journal of Business and Technology (Dhaka), Northern University Bangladesh.