

Automatically Sharing Web Experiences through a Hyperdocument Recommender System

Alessandra Alaniz
Macedo
SCE-ICMC
Univ. de São Paulo
Caixa Postal 668
São Carlos-SP, Brazil
ale@icmc.usp.br

Khai N. Truong
College of Computing & GVU
Center Georgia Institute of
Technology
801 Atlantic Drive 30332-0280
Atlanta/GA - USA
khai@cc.gatech.edu

José Antonio
Camacho-Guerrero,
Maria da Graça Pimentel
SCE-ICMC
Univ. de São Paulo- CP 668
São Carlos-SP, Brazil
jcamacho,mgp@icmc.usp.br

ABSTRACT

As an approach that applies not only to support user navigation on the Web, recommender systems have been built to assist and augment the natural social process of asking for recommendations from other people. In a typical recommender system, people provide suggestions as inputs, which the system aggregates and directs to appropriate recipients. In some cases, the primary computation is in the aggregation; in others, the value of the system lies in its ability to make good matches between the recommenders and those seeking recommendations.

In this paper, we discuss the architectural and design features of WebMemex, a system that (a) provides recommended information based on the captured history of navigation from a list of people well-known to the users — including the users themselves, (b) allows users to have access from any networked machine, (c) demands user authentication to access the repository of recommendations and (d) allows users to specify when the capture of their history should be performed.

Categories and Subject Descriptors

H.3 [Information Systems]: Information Storage and Retrieval; H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval—*Search process*; H.5.4 [Interfaces and Presentation]: Hypertext/Hypermedia—*Navigation*.

General Terms

Design, Algorithms, Theory.

Keywords

Web, Recommender Systems, Open Hypermedia, Information Retrieval, Semantic Structures.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

HT'03, August 26–30, 2003, Nottingham, United Kingdom.
Copyright 2003 ACM 1-58113-704-4/03/0007 ...\$5.00.

1. INTRODUCTION

The effort of finding information over the Web has been greatly facilitated over the recent years with significant improvements in the quality of the results returned by search engines. Two issues arise because search engines depend on a set of words given by users. First, the number of keywords provided by users is often suggested to be small [1]. Second, users must formulate an appropriate query for search engines to return completely satisfactory results. Thus, the challenge lies in providing search platforms with enough context about what users need. The use of local browsing context can help refine queries [31]; this typically involves the monitoring of the users' browsing activity. However, what defines the scope of browsing context is a difficult challenge.

Even after coming across potentially useful information, another hard task for users is to know if that information will be valuable again in the future. Too often, when we want to retrieve previously viewed information, we have either forgotten to bookmark the relevant URL or we cannot use the history mechanism stored on the browser machines. Existing history mechanisms do not provide appropriate searching services on the recorded data. Moreover, the existence of bookmarks and browsing histories tied to specific browsers or specific machines makes them less useful to people who use more than one computer or different browsers.

Recommendation systems are another category of application aimed at supporting users when searching for information. They are based on the idea that users often face the problem of having to make choices without sufficient experience and can use other people's recommendations. Recommendation systems leverage the notion that people are better at recognizing information needed that they see than at handling keywords over search engines [29]. Whereas the algorithms used by recommendation systems are not usually intuitive to the users, it can be assumed that friends have common understandings and interests. Sinha et al.'s study indicates that users still perceive friends as the best source of good and useful recommendations and have very high trust in their recommended information [35].

In this paper, we present WebMemex, a system geared towards making recommendations based on those pages the users themselves have previously seen. This is achieved by continuously capturing users' Web surfing activity. Moreover, the captured information may be shared with other people users know, in a manner of suggesting related in-

formation during their Web experience. In order to preserve privacy, users can explicitly disable the capture of their browsing activity. The WebMemex prototyped assists the users through a set of different infrastructures and applications supported by an open architecture. This system: (a) captures navigation using an extensible capture and access infrastructure [36]; (b) identifies semantic relationships between Web pages browsed by users using a linking server that manipulates semantic as similarity of terms according to Latent Semantic Indexing theory [27] [28]; (c) stores the associations identified in an open linkbase [8]; (d) handles the groups of people each user wants to share information with using *The Yahoo! MessengerTM* [5]. These characteristics make WebMemex an example of applying open hypermedia technology on the Web in this case, specially to create a recommender system.

The remainder of this paper is organized as follows. In Section 2, we briefly discuss the approaches of information retrieval and open hypermedia systems that we explored in WebMemex. In Section 3, we present the architecture and implementation details of the WebMemex. We also show how the system is used and discuss general issues such as privacy and social conventions. In Section 4, we describe an initial study of the WebMemex recommendation system used by two small groups of users. In Section 5, we discuss related work. We conclude with some discussion and some future directions of this work in Section 6.

2. BACKGROUND

Navigation on the World Wide Web relies on two main technical basis: selection of hypertext links or queries on search engines. While each technique constitutes a different paradigm, both are interesting in defining links to support user navigations over information.

Elaborated linking models are usually supported by open hypermedia systems (OHS); an open hypermedia system is a middleware that provides applications with hypermedia linking functionality orthogonally to their storage and display functionalities [14]. In open hypermedia systems, links are managed and stored in special databases called linkbases: the idea is to abstract links from documents and to store them in a linkbase. This approach incorporates flexibility to the documents since it allows the addition of hypermedia functionality to any document without changing the original document's format or embedding mark-up information within it. Several works have demonstrated the appropriateness of the open hypermedia approach [11] [16] [21], and a protocol [14] has been proposed towards allowing the interchange of information among applications. In this paper, we propose an application which is continuously recording users' Web surfing activity and using it to automatically define links between related documents visited. After that, the application sends the created links to a open hypermedia service called Web Linkbase Service (WLS). WLS is an XML-based open linkbase service for the Web that aims at (a) providing hypermedia functionalities to a set of non-hypermedia XML applications and (b) allowing the integration of these applications [8].

For some years, researchers have presented studies that support the definition of links using Information Retrieval approaches [7] [19] [33]. Some of those studies explore (a) the use of models of representation of information such as the vector model and the probabilistic model to represent

information to be navigated and (b) the manipulation of matching mechanisms such as lexical chains [20] and Latent Semantic Indexing (LSI) [17] to identify similarities between text elements in order to define hypertext links. The similarities defined by LSI are based on closeness of terms in a semantic space built according to co-occurrence of all terms of collection of documents manipulated. There are other approaches for describing semantic content and similarity; for instance, ontological reasoning services are used to represent conceptual models of document, terms and their relationships. By using predefined ontology based on a thesaurus, resource information can be manipulated for navigational purposes [12].

We are interested in the construction of services that automatically identify links among homogeneous repositories. We first adopted the use of a simple matching algorithm to identify lexical links. This service was experimented over two complementary repositories associated to the same material in a graduate course [30]. In another opportunity, we explored Latent Semantic Indexing [17] to overcome the polysemy¹ and synonymy² problems related to the use of lexical based analysis [28], manipulated the links in an OHS system [27] and supported feedback from users [26]. We have integrated the results of the research in a Web-based linking service called LinkDigger [10].

3. THE WEBMEMEX SYSTEM

To capture user navigations on the Web, the WebMemex system records the visited URLs into the LinkDigger system which supports the creation of links automatically over those web pages browsed. All links defined are managed by the open linkbase WLS. In this section, we first discuss the architecture of WebMemex and its components. We then illustrate how the system is used and conclude by discussing general issues including privacy and social conventions.

3.1 The Architecture

The WebMemex system was developed on top of a high level architecture that provides capture, linking, user authentication, storage, retrieval and access capabilities, as shown in Figure 1. In this section we discuss each component of the architecture.

3.1.1 User Authentication Component

The WebMemex application captures and recommends Web pages for groups of users. To identify users and group memberships, the application relies on the fact that the Web browser client must establish a connection to the Web proxy server for each HTTP request (see Figures 1a and 1c). The Web proxy server maintains a list of IP addresses it has heard from, the user ID at that IP address (if the user has successfully logged in or null if the user has not yet authenticated), and the last time there was activity from that machine (see Figure 1b). If a user is idle for more than an extended period of time, re-authentication is required. While the user ID is null, the Web proxy server only delivers back to the browser the sign-in page or the help page.

¹The polysemy problem occurs when a word has multiple meanings.

²The synonymy problem occurs when two or more words have the same or nearly the same meaning.

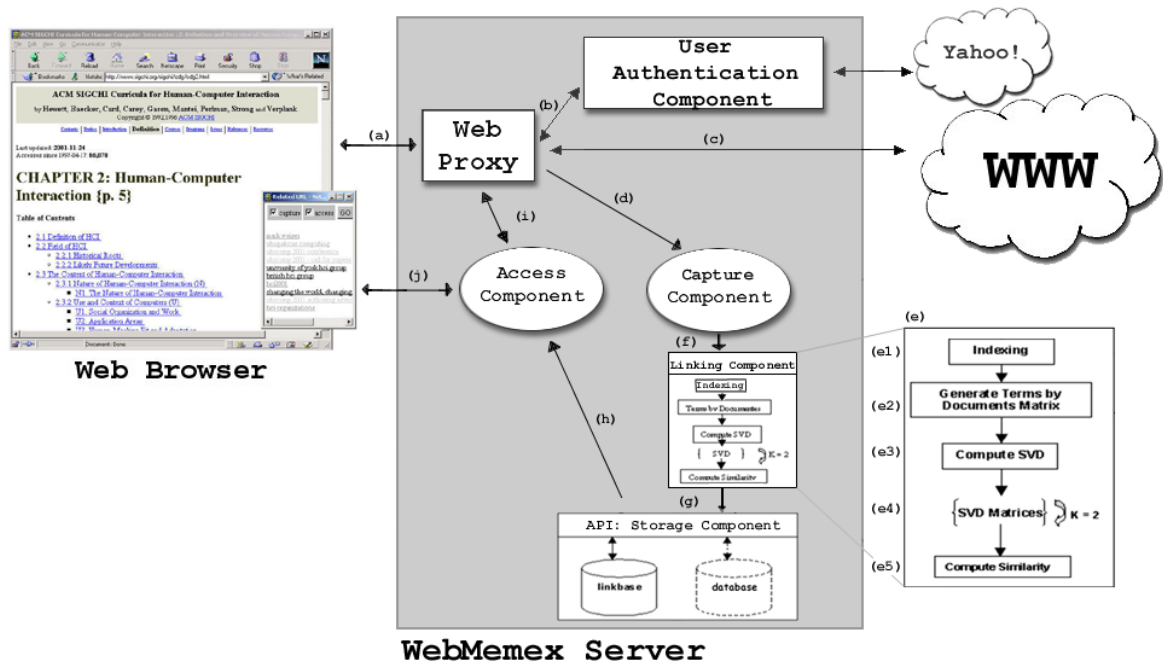


Figure 1: The WebMemex Architecture: (a) communication between the Web proxy server and a browser client, (b) handling of user authentication, (c) communication for each HTTP request, (d) capture component startup, (e) the underlying processing of the linking component, (f) linking component, (g) storage component, (h) access component startup, (i) a list of URLs to similar documents, (j) an access thread created for presenting recommendations.

3.1.2 Capture Component

The WebMemex service is supported through an augmented Web proxy server. When information is requested, the proxy server retrieves the information and immediately delivers it back to the requesting client (see Figure 1a). If users enable capture, then the retrieved document is also passed to the capture component (see Figure 1d). The proxy only logs information returned to the Web browser when the content type is text/html. HTML documents can be processed for additional meta-information, for instance to define relationships between those documents as hypertext links.

When an HTML document is served to the Web browser, the proxy server's capture component records the visit by tagging it with the URL, the time that Web page was visited, the IP address of the browser machine, and the user IDs. This information is necessary to initialize the linking component's process (see Figure 1f). The links created between the Web pages visited are sent to the storage component. When users want to visit the related pages using WebMemex, the access component will retrieve the links from the storage component.

3.1.3 Linking Component

Using the information sent by the access component, the process of word extraction from Web pages is initialized in the linking component by the indexing process. Figure 1e illustrates the processes of the linking component of the WebMemex architecture. This component automatically generates hypertext links between Web pages browsed by users and recorded by the capture component. After receiving the URLs of each page navigated and captured by the capture

component, the linking component performs two main tasks: indexes all Web pages and establishes the links among pages according to the LSI [17] approach.

The underlying processing of the linking component is as follows:

- Initially all Web pages captured by the capture component are indexed (see Figure 1e1). We used the mnoGoSearch [3] general public license search engine to index Web repositories. It was necessary to modify the engine itself so that it computes over links activated by JavaScript.
- The index resulting from the previous step is used to generate a term by document matrix (see Figure 1e2). This term by document matrix is called matrix X and it is exploited by Latent Semantic Indexing [17].
- The matrix X is decomposed into the product of three other component matrices T , S and D' using Single Value Decomposition (SVD) which is part of LSI theory (see Figure 1e3). Following the decomposition by SVD, the k most important dimensions (those with the highest values in the singular matrix S) are selected. All other factors are omitted. The reduction in the indexing space implies in less use of memory and computation. The amount of dimensionality reduction, i.e., the choice of k , is critical and is an open issue in the literature. Ideally, k should be large enough to fit the real structure in the data, but small enough such that noise, sampling errors and unimportant details are not modelled.

- A semantic matrix is generated by the combination of the component matrices of last step and a query column matrices by the computation of the inner-product among those matrices (see Figure 1e4). This query column matrix is generated after each Web page visited by a user with a capture component enabled.
- Given the semantic matrix generated in the previous step, relationships between the Web pages browsed by each group of users are identified by considering the cells that have the higher values of similarity (see Figure 1e5). A threshold level of similarity is used to filter the links created to generate a relevance semantic matrix which is used to identify semantic links between Web pages. The links generated are stored at the storage component (see Figure 1g).

We developed the linking modules to be reusable in other implementations. We use a configuration file to specify the context the linking modules will be employed. This configuration file describes information such as the vocabulary, stop list and source of URLs.

3.1.4 Storage Component

Links created by the linking component can be stored in databases or open hypermedia linkbases. In our WebMemex prototype, we have used a linkbase called WLS [8], which offers some facilities that common databases do not offer.

The WLS linkbase was developed as an API (Application Programming Interface), so that application developers can reuse and combine the available operations with their own building blocks. By means of its API, WLS can be reused in different contexts, reducing the authoring effort as shown in [8].

During the design of the WebMemex architecture, we defined a mapping between objects manipulated by the architecture and the classes of the WLS presented in Figure 2. For example, **Anchor** is a set of terms of each Web page captured during user navigation and **EndPoints** are equivalent to terms mapped to **Anchors**. **Link** is a set of two **EndPoints** identified to relationships defined with the linking component. **Context** is a collection of links generated automatically. **Nodes** corresponds to each Web page that has been found to be similar to another page. Finally, **Semantics** correspond to pairs of similar terms. These pairs of terms can be automatically defined by the linking component and stored in the WLS linkbase, independently of the documents identified holding the information.

Because of the relationships among the **Anchor**, **EndPoint** and **Link** classes, WLS provides support to n-ary multidirectional links and to the sharing of an anchor between several links endpoints to our architecture. The external linkbase defined by WLS supported an effective creation of links; these functionalities have been explored in the prototype.

Besides the WLS tables, the storage component manipulates additional tables that are used during the indexing process to manage the URLs, the stop words list, the weights of terms, and the list of users and groups. A table that relates users and their groups with the URLs they visit is updated as new URLs are captured by the system. The access component obtains links from these tables with matching user and group IDs.

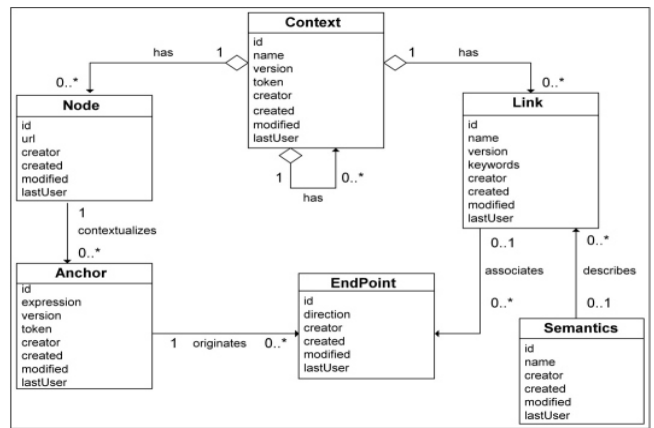


Figure 2: Conceptual Model of WLS linkbase [8].

3.1.5 Access Component

The WebMemex server separates the handling of the capture, linking, storage and access concerns to not hinder the Web browsing activity as the system captures the Web visit and suggests relevant material. As a URL is being captured and subsequently linked and stored, Web proxy server creates an access component thread to query for recommendations from the system (see Figure 1i).

When users visit a Web page for the first time, the access component must wait for the linking and storage process to complete. Once related links are computed and stored, the storage component provides the access component with a list of URLs to similar documents (see Figure 1h). The access component formats these URLs into links presented to the users as recommendations displayed in a small pop-up browser window (see Figure 1j).

3.2 The Prototype

To build the WebMemex prototype, we augmented a standard Web browser to support a number of access features. One access feature supported, though not discussed in this paper, is the ability to perform searches over personal Web histories, allowing the user to revisit her previous navigation trails. In this paper, we detail the recommendation capability (an asynchronous collaborative access feature), where the system uses previous Web experiences to enhance user navigation session with suggestions of related URLs.

3.2.1 Using WebMemex

To use WebMemex, users need to configure a browser to talk to a proxy server that captures Web histories as indicated in Figure 3a. When they begin a Web surfing session (see Figure 3b), the browser automatically goes to a sign-in screen (see Figure 3c). Once authenticated, the Web browser is fully augmented to capture and suggest URLs from the Web surfing history of the users and their friends. The related URLs are shown in a small window outside of the Web browser (see Figure 3d).

3.2.2 Handling User Authentication

The WebMemex system unloads the user authentication responsibility to the “Yahoo! Messenger” service. Before using the actual prototype, users must register for “Yahoo! Messenger” accounts. When users sign in to use Web-

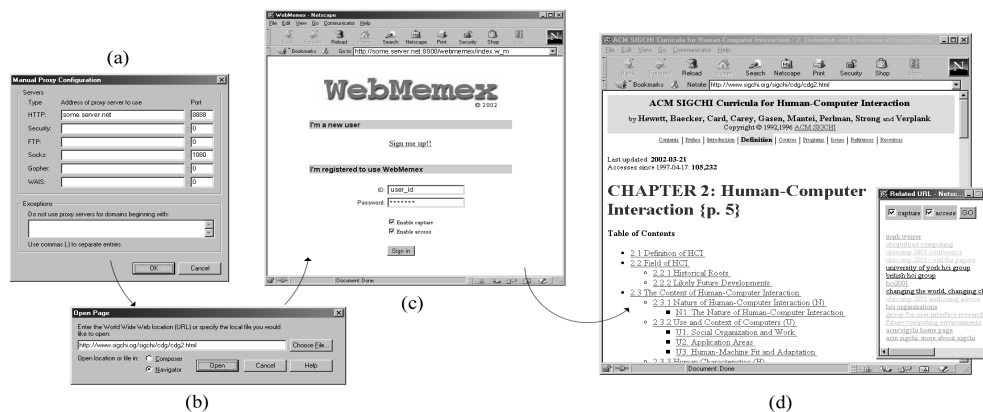


Figure 3: The WebMemex Prototype. Screens for: (a) proxy configuration, (b) browsing session startup, (c) user authentication, (d) presentation of the related URLs.

Memex, this information is confirmed through Yahoo! The instant messaging application, itself, does not need to be running or even installed. However, users must have Yahoo! accounts, because the “Yahoo! Messenger” system can verify correct ID and password combinations.

The decision to leverage Yahoo! accounts goes beyond the simple ID and password verification service. Instant messaging applications also store the user buddy lists, or their online circle of friends. These lists allow the WebMemex system to determine with whom the users’ captured histories should be shared. Thus, rather than needing to maintain our own user authentication system we used Yahoo! Messenger.

Implementation details. The user authentication component is implemented in Java to verify username and password pairs with Yahoo! Instant Messaging server. Through an HTTP connection with the Yahoo! server, if the correct username and password pairs are provided, a buddy list for that login is obtained by the user authentication component.

3.2.3 Recording Web Pages Continuously

To capture users’ Web surfing history, the system needs to be able to monitor the Web pages they visit. We initially explored the implementation of hooks or listeners for common Web browser’s such as Internet Explorer; however, we found this method was not ideal because client-based solutions are too platform-specific. In some situations, users may also work on more than one machine that individually logs Web visits. However, a Web viewing history should be accessible from any networked machine, regardless of the machine on which the URL was initially visited.

Hence, we leverage on existing Web browser’s ability to talk to an HTTP proxy. On existing Web browsers, users can quickly specify the location where the proxy server is running, as in Figure 3a. When users begin Web surfing, their Web browsers will talk to the Web proxy server. This proxy server initially checks to see if it knows the users (i.e., if users are registered and logged into the system). This step allows users to use the service while logged into different machines using the same ID (see Figure 3c); but the information will automatically be tied together with their previously captured information.

Implementation details. The Web proxy is a Java application written to block HTTP requests until the users success-

fully log in to the WebMemex. It spawns the capture and access component as users visit pages. The capture component is a Java thread spawned each time users view a page. This thread tags the Web visit with the timestamp and identity of the users before passing it to the linking component.

3.2.4 Linking Web Pages Captured

Figure 1e shows an instance of the linking component used in the WebMemex prototype. The “Compute Similarity” module uses a level of similarity filter to generate a relevance semantic matrix which identifies and creates relationships between repositories. In WebMemex, we are using 85% as the level of similarity.

Because the computation involved with LSI is expensive, the matrices are computed periodically (such as once a day). During the access phase, the keywords computed for the page being visited are used to instantly perform a query in the semantic matrix.

The query column matrix will consider all navigations performed until the last computation of the semantic matrix; the most recent navigation will be included the next time the semantic matrix is computed as we explained before.

After creating the links in the last module of the linking component, those links are stored in the WLS linkbase which manages and provides the information to the WebMemex.

Implementation details. We are using a matrix-oriented programming language, OX [15], augmented with C++ code to implement the modules supporting the linking process.

3.2.5 Accessing Web Pages Suggested

Hypertext links are used to present suggestions to the users. They are displayed in a small pop-up window, as shown in Figure 4, when users navigate the Web. For each page users view, WebMemex relates it with pages visited by the users and their friends.

Implementation details. The access component is a Java thread which talks to the WLS linkbase to retrieve a list of Web pages relevant to the current Web visit. It formats the list of related URLs into an HTML interface (see Figure 1h).

3.2.6 Handling Privacy and Social Conventions

The fact that browsing information is continuously captured and shared leads to important issues in terms of pri-

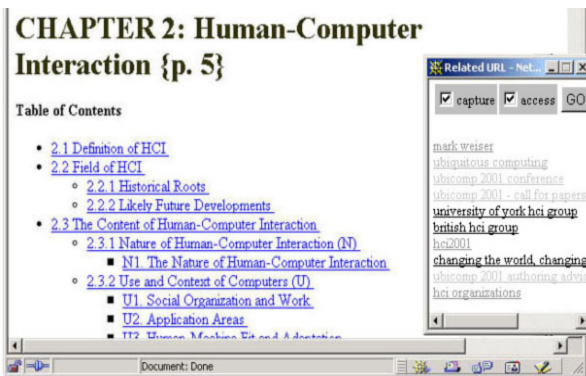


Figure 4: The links to the suggested Web pages are presented in a small pop-up window – different colors are used to indicate several levels of relevance. In this interface, users can enable and disable the capture and access components.

vacy and social conventions. In some situations, users may not want to have their Web histories preserved; or users may view pages that they do not want to share with others.

In order to handle privacy, the WebMemex interface gives users opportunities to specify the services they want. At the sign-in time, users can tell the system to “enable capture” and “enable access” (top of pop-up window in Figure 4).

To preserve the spirit of sharing, the option to capture information and the option to share what is captured were reduced to a single option to enable capture. Thus, all information that is captured is shared.

The default is that both capture and access are enabled. During the course of a surfing session, it is also possible for users to change their mind about whether or not to have the content captured and shared or not.

While privacy may lead to the desire for being able to modify retroactively the captured history of more than just the currently viewed page, it is not completely necessary for users to manually specify that they did not view a page. The same effect is achieved by protecting anonymity of the source of the suggested materials.

It is possible that some of the people whom users have on their list of friends do not have them on theirs. There may be people who do not want to share captured histories with other users. In normal social conventions, when a person has a question, she would ask people she considers her friends for advice. If the person asked considers her to be a friend, a response is probably returned.

This is not to say that if the person asked does not consider her to be a friend, a response will not be returned. However, in such scenarios, how much trust to put into the response is questionable. To reduce this problem, the protocol that WebMemex uses to support the asynchronous collaboration is to share information only between reciprocating friends. With the list of people users consider their friends, the system checks for the subset that has them also on their list of friends. WebMemex uses this subset of reciprocating friends to check their captured histories for related Web pages to suggest back to the users.

This design decision also helps to resolve a second problem that arises in the domain of information sharing when a person’s social circle changes. In such scenarios, should old

friends still be able to see information that they could have seen when they were friends with the users, and should new friends be allowed to see something users captured some time ago? Each time users sign in, the system verifies IDs and passwords with the authentication component. A matching ID and password allows the system to retrieve the users’ list of friends. The system caches this list for security reasons rather than keeping passwords which the users might change later. Using this cached list, it is possible to determine the set of reciprocating friendships for every user. This real-time method for determining reciprocating friends resolves the issue of changing group dynamics.

Another social convention was defined to protect privacy when users do not have more than one reciprocating friend. In this scenario, suggestions from friends are not included in what is returned to the users. This mechanism protects privacy because users can easily determine when they have not visited a Web page and therefore, the suggestion must have come from their sole friend’s viewing history.

4. EVALUATION

In this section, we report two experiences with the Web-Memex prototype. We split the section into two subsection. The first part focuses on experience of using WebMemex among four people at the Georgia Institute of Technology (GATECH) in Atlanta/GA — USA and the second one is based on experiences with WebMemex among four people at the São Paulo University (USP) in São Carlos/SP — Brazil.

The first experiment was conducted among a group of four postgraduate school colleagues. These subjects provided us with Web surfing history for the span of a month. Their histories included themes such as computer hardware and software, e-commerce sites, online banking, summer internships, news and tourist attractions.

The second group of subjects is composed of three Brazilian postgraduate students and one professor. They are a homogeneous group of people who collaborate in the Hypermedia Group of their university. During the span of a month, they searched for information related to ontology, XML approaches, Web services, agents, Java, systems for televising presentation, computer hardware, online banking, sport site and tourist attractions.

There is some contrast in the focus of the Web navigations of the two groups used in our experiments. The first group primarily works on mobile laptops as their primary computers. The second group works on different fixed computers at work and at home and we observed different access patterns from machines at work and at home. As a result, there is more work context found in the Web history gathered for the second group. The data collected for the first group has many themes beyond only the work context.

4.1 The First Experiment

In the first experiment, a set of 2556 Web pages was captured for the four subjects participating in this experiment, an average of 639 Web pages from each person as shown in Table 1. Those Web pages were related to various topics including computer hardware and software, e-commerce sites, online banking, summer internships, news and tourist attractions. However, a manual analysis of each page revealed that several Web visits were to images file, pdf files or text files with images. The first step in this experiment was to include, in the configuration file of linking service, a filter

Table 1: Characteristics of the two experiments

General Characteristics	Experiments	
	1st	2nd
Number of Web pages	2556	2555
Number of users	4	4
Average of pages by user	639	638.7
Number of Web pages after filtering	454	106
Average of pages by user	113.5	26.5
Number of distinct terms	13376	2466
Number of stop words used	562	1180
Number of hypertext links created	71	154

Table 2: For the first experiment: (a) Number of Web visits for which recommendations were made & (b) number of recommendations pulled from user’s Web history.

	(a)	(b)
User 1	53	37
User 2	12	19
User 3	5	9
User 4	1	6

to ignore image files and compressed files. We required at least 10 words in a document before it is considered as a useful document. From this manual analysis, we have since included this filter in the WebMemex prototype to automatically look for text/html files that meet the above criteria. After the filtering step, we had 454 Web pages indexed.

The indexing process over those 454 Web pages generated a matrix of 454 columns by 13376 lines to be handled using the Latent Semantic Analysis approach. At the end of linking process we had 71 hypertext links defined for the first experiment (see Table 1).

We anticipated that there would be a large number of context that exist collectively for all the subjects. Analyzing the relationships created among the Web pages visited by the four subjects during last weeks of their browsing history, we observed that there is a small number of context under which different subsets of the users’ Web visits related. With respect to the 454 Web pages our system considered to be useful documents, a reasonable 71 recommendations was generated.

Analyzing information from Table 2, we can verify how numerically useful the WebMemex recommendations are for each user. For example, User 1 has 53 Web pages for which the system provided recommendations. From her captured Web history, 37 recommendations were formed (for her and others). However, User 2 has 12 pages where she received relevant links and provided 19 recommendations. So we can see WebMemex was numerically more helpful for User 1.

In this experiment, 48 recommendations came from the same user’s history and 23 recommendation came from a different user’s history. The 23 recommendations account for almost a third of the total number of links, demonstrating the potential value of having other people’s information accessible to the group.

A qualitative analysis of the first experiment gave us the following information. A manual analysis of the total 71 links generated as WebMemex recommendations revealed that 44 (62%) links were actually related to the Web page

Table 3: For the second experiment: (a) number of Web visits for which recommendations were made & (b) number of recommendations pulled from user’s Web history.

	(a)	(b)
User 1	30	135
User 2	10	16
User 3	4	5
User 4	1	2

which they were anchored. This is a very satisfactory number by information retrieval standards [13].

4.2 The Second Experiment

A set of 2555 Web pages was recorded during navigation of 4 subjects in the second experiment, or an average of 639 Web pages per person (see Table 1). The 2555 Web pages were based on the computer science themes previously mentioned. We carried out a manual analysis over those pages and defined a filter and a numerical threshold to ignore the same kind of files that we are eliminating in the first experiment. After the filtering step we had 106 Web pages to be indexed as presented at Table 1.

The indexing process resulted in a matrix with 2466 distinct terms for 106 documents. The LSI modules in the linking component operated on this initial matrix to create 154 hypertext links. Those relationships were used as recommended pages for subjects in the second group, and the number of links created for each user is indicated in Table 3. The high number of links is related to the fact the browsing space in this experiment is more homogenous when compared with the first experiment.

Since the total of 154 links created in this experiment is too high for practical purposes, alternatives such as adjusting process of link creation with respect to the level similarity must be investigated (we are currently using 85% as a fixed level of similarity, as discussed in Section 3.2.4).

While we are concluding a qualitative analysis over the second experiment results, a couple of points can be extracted. First, there is a large number of URLs viewed by more than one of the subjects. As a result, when recommendations are computed, links to the anchor have multiple endpoint sources (sometimes including from the same user herself). This suggests that in WebMemex, even if users have not previously seen a related page, other users’ Web history can still lead them to the same content.

5. RELATED WORK

Recommender systems is a category of systems that assist and augment the natural social process of asking for recommendations from other people [2]. Sometimes, recommender systems are also referred to as collaborative filtering or social filtering systems according to Goldberg et al. [18]. In a typical recommender system, people provide recommendations as inputs, which the system aggregates and directs to appropriate recipients. In some cases, the primary computation is in the aggregation; in others, the system’s value lies in its ability to make good matches between the recommenders and those seeking recommendations.

Recommender systems can use different mechanisms to provide information to the systems. SIFT is a kind of rec-

ommender system where the users need to provide keywords to describe their interest [37]. Thus, this system still has the same challenges of search engine: it requires a lot of effort on the part of user.

On the other hand, there are recommender systems which learn users' preferences by eliciting explicitly some kind of user feedback instead of formulation of queries. For instance, when users move from one page to another, users rate how interesting the current page is. Examples of recommender systems in this category are: Ringo, Fab, Alexa and PHOAKS. Ringo is a system for personalized recommendations for music albums and artists [34]. Fab is a recommender system based on user votes and content-based filtering techniques [6]. Alexa is a Web navigation assistant which provides information about the registered site owner, ratings and reviews of the site, site statistics and it can also recommend related sites [23]. Finally, PHOAKS (People Helping One Another Know Stuff) is a filtering site for Usenet Netnews, used to gather people's opinions of Web resources in Netnews [24].

There are also recommender systems that explore user preferences transparently without any extra effort from the users. For example, Letizia [25], WebWatcher [22], Margin Notes [31] and QuickStep [29] infer user preferences from observing user-browsing behavior. The MEMOIR system manipulates a set of software agents that mine trails of users and it uses hypertext technologies such as open hypermedia link services to allow users to manage associations between different types of documents in an independent and flexible way [32].

Another kind of recommender systems are popular Web sites such as Amazon.com, MovieFinder.com, CDNow.com and Launch.com. They have placed collaborative filtering technology into authentic use settings. These systems have proven to be accurate enough in their specific entertainment domains. However, the success of these systems has not been met in other domains or more general experiences because collaborative filters compute predictive models based on heuristic approximations of human processes.

The WebMemex system mainly differs from these systems in the following aspects. First, recommended information comes from a list of people, well-known to the users, with common ground and interest and most importantly who the users trust. Second, the majority of systems discussed rely on local context or a short-term user profile and function in a manner similar to the Remembrance Agent [31]. Rather than trying to predict user profiles and creating social clusters based on dimensions of interest, our system uses what the users define as their social circle. Each Web page the users visit is captured and compared against all other Web pages they have previously seen and those that their friends have seen. From this point, traditional data clustering methods can be used.

6. CONCLUSION AND FUTURE WORK

Motivated by Bush's article *As We May Think* [9], Douglas Engelbart envisioned the use of computer-based tools to augment human intellect and improve our overall ability to tackle the problems. In his work at the Bootstrap Institute, Engelbart coined the term "Collective IQ" to describe how a group can "leverage its collective memory, perception, planning, reasoning, foresight, and experience into applicable knowledge" to solve problems of users [4]. We have built

a capture and access application to explore the visions of Vannevar Bush and Douglas Engelbart.

Our WebMemex continuously captures users' Web surfing history and uses this history to provide the users and their friends with suggestions of related Web pages to what they are currently viewing. This system acts as an instantiation of an architecture for capturing and asynchronously sharing experiences for the automated recommendation of related information.

Besides the quantitative evaluation presented, we are designing a qualitative evaluation of these recommendations towards answering questions like "how many pages were indeed used" and "how is the quality of these recommended pages". We are also interested in comparing homogeneous and heterogeneous groups of users in terms of browsing patterns to determine if more controlled and focused groups of people would be shared more effective suggestions.

The WebMemex application will be deployed to study how an automated capture and access Web application is adopted by users. Users are allowed to create many profiles. Thus, they can use this application in different ways: (a) as a personal application which only suggests information from her own personal Web history; or (b) as a collaborative application where information is shared between groups of friends and recommends related URLs from the collective Web history. We will study how it is used and examine if the system is found to be useful for individual users and/or a group of users.

In a different approach, we are investigating the recommending of related material coming from a long-term capture history authored by the user as she interacts with the Web application. The suggestion of related material can be considered an added feature to the Web surfing activity in the WebMemex.

ACKNOWLEDGMENTS

Alessandra Macedo is a PhD candidate supported by FAPESP (99/115270). Maria Pimentel currently holds individual research grants from CNPq and FAPESP. Maria Pimentel has international support to the InCA-SERVE Project by CNPq in Brazil jointly with Gregory Abowd who is supported by NSF in the U.S. The authors would like to thank the subjects of experiments, Gregory Abowd and Lonnie Harvel for their support.

7. REFERENCES

- [1] Google site. Internet. <http://www.google.com/help/basics.html>.
- [2] Recommender systems site. Internet, 1999. <http://www.iota.org/Winter99/recommend.html>.
- [3] Mnogosearch group. Internet, 2001. <http://www.mnogosearch.ru>.
- [4] Bootstrap Alliance — Douglas Carl Engelbart. Internet, 2002. <http://www.bootstrap.org/engelbart/index.jsp>.
- [5] Yahoo! Internet, 2002. <http://www.yahoo.com>.
- [6] M. Balabanovi and Y. Shoham. Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40(3), March 1997.
- [7] J. Blustein. Automatically generated hypertext versions of scholarly articles and their evaluation. In *Proceedings of the Eleventh ACM on Hypertext and Hypermedia*, pages 201–210, 2000.

- [8] R. F. Bulcão Neto, C. A. Izeki, M. G. C. Pimentel, R. P. M. Pontin, and K. N. Truong. An open linking service supporting the authoring of web documents. In *Proceedings of the ACM Document Engineering Conference*, pages 66 – 73, Virginia, USA, 2002.
- [9] V. Bush. As we may think. *Atlantic Monthly*, 1945.
- [10] J. A. Camacho-Guerrero, A. A. Macedo, and R. P. M. Fortes. Uma infra-estrutura configurável para serviços de criação automática de ligações. In *Anais do VII Brazilian Symposium on Multimedia and Hypermedia System*, pages 298 – 305, Fortaleza/CE, 2002.
- [11] L. Carr, D. C. DeRoure, H. C. Davies, and W. Hall. The distributed link service: A tool for publishers, authors and readers. In *Proceedings of the fourth International World Wide Web*, pages 647–656. ACM Press, 1995.
- [12] L. Carr, W. Hall, S. Bechhofer, and C. Goble. Conceptual linking: Ontology-based open hypermedia. In *Proceedings of the 10th International World Wide Web*, pages 334–342. ACM Press, May 2001.
- [13] G. Cormack, C. Clarke, C. Palmer, and S. To. Passage based refinement. In *Proceedings of Sixth Text REtrieval Conference (TREC-6)*, pages 303 – 319, Maryland, USA, 1997.
- [14] H. Davis, A. Lewis, and A. Rizk. OHP: a draft proposal for a standard Open Hypermedia Protocol. In *Proceedings of the 2nd Workshop on Open Hypermedia Systems*, pages 27 – 53, 1996.
- [15] J. Doornik. Ox - an object-oriented matrix language, 2002.
<http://www.nuff.ox.ac.uk/users/doornik/doc/ox/>.
- [16] M. A. Fountain, W. Hall, I. Heath, and H. C. Davis. Microcosm: An open model for hypermedia with dyanmic linking. In *Proceedings of ECHT'90*, pages 298 – 311, 1990.
- [17] G. W. Furnas, S. Deerwester, S. T. Dumais, T. K. Landauer, R. A. Harshman, L. A. Streeter, and K. E. Lochbaum. Information retrieval using a singular value decomposition model of latent semantic structure. In *Proceedings of the Eleventh International Conference on Research & Development in Information Retrieval*, pages 465 – 480, 1988.
- [18] D. Goldberg, D. Nichols, B. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Comm. of the ACM*, 35(12):61–70, 1992.
- [19] G. Golovchinsky. What the query told the link: The integrations of hypertext and information retrieval. In *Proceedings of the ACM Conference on Hypertext 1997*, pages 30 – 39, 1997.
- [20] S. Green. Building hypertext links by computing semantic similarity. *IEEE Transactions on Knowledge and Data Engineering*, 11(5):713 – 730, September/October 1999.
- [21] K. Grønbaek, L. Sloth, and P. Orbaek. Webvise: browser and proxy support for open hypermedia structuring mechanisms on the WWW. In *Proceedings of the Eighth International World Wide Web Conference*, pages 253 – 267, Toronto, Canada, 1999.
- [22] T. Joachims, D. Freitag, and T. Mitchell. Webwatcher: A tour guide for the world wide web. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI'97)*, 1997.
- [23] B. Kahle and B. Gilliat. Alexa – navigate the Web smarter,faster, easier. Technical report, Alexa Internet, Presidio of San Francisco/CA, USA, 1997.
<http://www.alexa.com>.
- [24] R. Keller, S. Wolfe, J. Chen, J. Rabinowitz, and N. Mathe. A bookmarking service for organizing and sharing URLs. In *Proceedings of The Sixth International World Wide Web Conference*, Santa Clara-California/USA, 1997.
- [25] H. Lieberman. Autonomous interface agents. In *Proceedings of CHI'97*, 1997.
- [26] A. A. Macedo, J. A. Camacho-Guerrero, and M. G. C. Pimentel. Incluindo abordagens de recuperação de informação em serviços de criação de hiperligações. In *XXVIII Conferencia Latinoamericana de Informática*, Montevideu/Uruguai, Novembro 2002.
- [27] A. A. Macedo, M. G. C. Pimentel, and J. A. Camacho-Guerrero. An infrastructure for open latent semantic linking. In *Proceedings of ACM Hypertext 2002*, pages 107 – 116. ACM Press, 2002.
- [28] A. A. Macedo, M. G. C. Pimentel, and J. A. C. Guerrero. Latent semantic linking over homogeneous repositories. In *Proceedings of the ACM Symposium on Document Engineering*, pages 144 – 151. ACM Press, November 2001.
- [29] S. Middleton, D. D. Roure, and N. Shadbolt. Capturing knowledge of user preferences: Ontologies in recommender systems. In *Proceedings ACM of K-CAP'01*, pages 100 – 107. ACM Press, October 2001.
- [30] M. G. C. Pimentel, A. A. Macedo, and G. D. Abowd. Linking homogeneous web-based repositories. In *Proceedings of International Workshop on Information Integration on the Web*, pages 35 – 42, Rio de Janeiro-Brazil, 2001.
<http://www.cos.ufrj.br/wiww/schedule.html>.
- [31] B. Rhodes. *Just-In-Time Information Retrieval*. PhD thesis, Media Laboratory MIT, 2000.
- [32] D. D. Roure, W. Hall, S. Reich, G. Hill, A. Pikrakis, and M. Stairmand. MEMOIR – an open framework for enhanced navigation of distributed information. *Information Processing and Management*, 37:53 – 74, 2001.
- [33] G. Salton and J. Allan. Selective text utilization and text transversal. In *Proceedings of the ACM Conference on Hypertext 1993*, pages 131 – 144, 1993.
- [34] U. Shardanand and P. Maes. Social information filtering: Algorithms for automating “word of mouth”. In *Proceedings of Conference on Human Factors in Computing Systems*, pages 210 – 217, Denver/CO, USA, 1995.
- [35] R. Sinha and K. Swearingen. Comparing human recommenders to online systems. In *Proceedings of Delos-NSF Workshop on “Personalisation and Recommender Systems in Digital Libraries”*, 2001.
- [36] K. Truong and G. Abowd. Enabling the generation, preservation & use of records and memories of everyday life. Report GIT-GVU-02-02, Georgia Institute of Technology Technical, January 2002.
- [37] T. Yan and H. Garcia-Molina. SIFT - a toll for Wide Area Information Dissemination. In *USENIX Technical Conference*, 1995.