Memex: A browsing assistant for collaborative archiving and mining of surf trails

Soumen Chakrabarti Sandeep Srivastava Mallela Subramanyam Mitul Tiwari

Indian Institute of Technology Bombay

soumen,sandy,manyam,mits@cse.iitb.ernet.in

Abstract

Keyword indices, topic directories, and link-based rankings are used to search and structure the rapidly growing Web today. Surprisingly little use is made of years of browsing experience of millions of people. Indeed, this information is routinely discarded by browsers. Even deliberate bookmarks are stored in a passive and isolated manner. All this goes against Vannevar Bush's dream of the *Memex*: an enhanced supplement to personal and community memory.

We propose to demonstrate the beginnings of a 'Memex' for the Web: a browsing assistant for individuals and groups with focused interests. Memex blurs the artificial distinction between browsing history and deliberate bookmarks. The resulting glut of data is analyzed in a number of ways at the individual and community levels. Memex constructs a topic directory customized to the community, mapping their interests naturally to nodes in this directory. This lets the user recall topic-based browsing contexts by asking questions like "What trails was I following when I was last surfing about *classical music*?" and "What are some popular pages in or near my community's recent trail graph related to *music*?"

1 Motivation

Three paradigms have emerged for exploring the Web: keyword search, directory browsing, and following links. Popular search engine and directory sites are visited tens of millions of times per day. We speculate that the total number of clicks per day is orders of magnitude larger. This third source of information, the browsing history of millions of Web users over several years, an information source that dwarfs the scale of the Web itself, is almost entirely discarded by browsers as 'history'. Deliberate 'bookmarks' are preserved, but passively, in browser-dependent formats; this separates them from the dominant world of HTML hypermedia, even if their owners were willing to share them (as they are, in our experience, with all but a small section of their browsing activity).

Proceedings of the 26th VLDB Conference, Cairo, Egypt, 2000.

In 1945, Vannevar Bush dreamt of *Memex*: an enhanced, intimate supplement to personal and community memory [2]. Assisted by a Memex for the Web, a surfer can ask:

- What was the URL I visited about six months back regarding compiler optimization at Rice University?
- What was the Web neighborhood I was surfing the last time I was looking for resources on classical music?
- Are their any popular sites, related to my (Web) experience on classical music, that have appeared in the last six months?
- How is my ISP bill divided into access for work, travel, news, hobby and entertainment?
- What are the major topics relevant to my workplace? Where and how do I fit into that map? How does my bookmark folder structure map on to my organization?
- In a hierarchy of organizations (by region, say) who are the people who share my interest in recreational cycling most closely and are not likely to be computer professionals?

Since Bush proposed Memex, the theme of a 'living' hypermedia into which we "weave ourselves" has been emphasized often, e.g., by Douglas Engelbart[1] and Ted Nelson[2], and of late by Tim Berners-Lee[3] and Jim Gray[4]. Indeed, the current cost/volume ratio of storage makes it unnecessary to delete *anything* from one's Web surfing experience, provided we can make fruitful use of it.

We propose an architecture of a 'Memex' for the Web which can answer the above questions. Memex is a large project involving hypertext data mining, browser plug-in and applet design, servlets and associated distributed database architecture, and user interfaces. We have validated the design using a prototype implementation that we describe here. Memex is currently implemented on Netscape 4.5+. We are currently testing Memex with the help of local volunteers. The Memex service will be made

[1] http://jefferson.village.virginia.edu/elab/hfl0035.html

[2] http://www.sfc.keio.ac.jp/~ted/

[3] http://www.w3.org/1999/04/13-tbl.html

[4] http://research.microsoft.com/~gray/papers/MS_TR_99_50_TuringTalk.pdf

publicly accessible[5]. Further details about Memex have been reported elsewhere [4].

2 Client architecture overview

Memex should run on popular browsers. It should be possible to distribute updates and new features effortlessly to users. Hence the Memex client has been designed as an applet. In view of secure firewalls, proxies, and ISPs' restrictions on browser setups, the client should communicate with the server over HTTP. The data transfered should be encrypted, if desired, to preserve privacy.

The user can log on to a Memex server at the level of a department, organization, interest group, ISP, nation or the world. The architecture makes no assumptions about the logical community level at which Memex might be deployed. At any time, the user can choose not to archive surfing actions, archive for private use, or archive for use by the community (Figure 1). If permitted, Memex taps the browser to get the current location and passes this on to the server, which then processes it in many ways.

Apart from a standard full-text search over all pages visited, the Memex client has several function tabs to assist topic-based mining. The editable **folder tab** (Figure 1) provides topic management: this is the means by which users exemplify their interests. Existing bookmarks from Netscape or Explorer can be imported into Memex's editable tree-structured topic view; conversely Memex can export back to these browsers. Apart from implicit history logging, bookmarks can be added to folders while surfing. A user will typically assign a bookmark explicitly to a topic. These assignments are analyzed by the server, which then **classifies** all surfed pages automatically into these folders. The folder tab can also be used to reinforce or correct the classifier. Memex also uses unsupervised **clustering** to propose a topic hierarchy [6] over a set of links that the user may want to reorganize. Periodically, the server consolidates all users' public folders and browse history into a topic directory tailored to the needs of that specific community (see §4 and Figure 4).

Users surf on many topics with diverse priorities. Because browsers have only a transient context (one-dimensional history list), surfers frequently lose context when browsing about a topic after a time lapse. Studies have shown that visiting Web pages is best expressed using spatial metaphors: your context is "where you are" and "where you are able to go" next [9]. Memex's topic classifier also helps render the topic-focused **trail tab** (Figure 2). In the trail tab, the left panel shows the user's topic folders. Selecting a folder replays the hypertext graph of recent pages publicly surfed by the community which are most likely to belong to the selected topic, and thus recreates the user's browsing context.

[5] `http://www.cse.iitb.ernet.in/~soumen/memex/`

3 Server architecture overview

On the server side, the system should be robust and scalable. It is important that the server recovers from network and programming errors quickly, even if it has to discard a few client events. The server consists of servlets that perform various archiving and mining functions as triggered by client action, or continually as demons. We prefer servlets to CGI scripts because the client-server interactions exchange complex objects and sometimes have state. We prefer HTTP tunneling also because direct JDBC connections may be refused by many firewalls.

Server state is managed by two storage mechanisms: a relational database (RDBMS) such as Oracle or DB2 for managing metadata about pages, links, users, and topics, and a lightweight Berkeley DB[6] storage manager to support fine-grained term-level data analysis for clustering, classification, and text search. Storing term-level statistics in an RDBMS would have overwhelming space and time overheads.

An interesting aspect of the Memex architecture is the division of labor between the RDBMS and the lightweight storage manager. Planning the architecture was made non-trivial by the need for asynchronous action from diverse modules. There are some user interface-related events that must be guaranteed immediate processing. Typically these are generated by a user visiting a page, or deliberately updating the folder structure. With many users concurrently using Memex, the server cannot analyze all visited pages, or update mined results, in real time. Background demons continually fetch pages, index them, and analyze them w.r.t. topics and folders. The data accesses made by these demons have to be carefully coordinated. This would not be a problem with the RDBMS alone, but maintaining some form of coherence between the metadata in the RDBMS and several text-related indices in Berkeley DB required us to implement a loosely-consistent versioning system on top of the RDBMS, with a single producer (crawler) and several consumers (indexer and statistical analyzers). Figure 3 shows a block diagram of the system.

4 Mining algorithms overview

The stream of data from surfers has to be analyzed in various ways. Some parts of the processing, such as keyword indexing, are mundane. Other parts constitute new algorithms or novel implementations.

For clustering we started with a bottom-up hierarchical agglomerative approach [6]. For classification we started with a Bayesian classifier [3]. Although these simple text-based techniques work reasonably well for

[6] `http://www.sleepycat.com`

average Web pages, bookmarked URLs offer special challenges: people tend to bookmark many "front pages" with less text and more graphics compared to typical Web documents. Surfers may also place two URLs in the same folder for functional reasons, even if the corresponding documents are syntactically dissimilar.

We have implemented two new learning algorithms for Memex. For classification we use a new technique that combines features from text, hyperlink and folder placement to offer significantly boosted accuracy, increasing from a mere 40% accuracy for text-only learners to about 80% with our more elaborate model.

We generalize clustering to finding a new notion of **themes** among the bookmarks. In principle, each user need not design his/her own topic hierarchy, given there are 'standard' ones like Yahoo![7] and the Open Directory[8]. In practice, these 'universal' hierarchies are neither necessary nor sufficient for individual surfers and focused communities, they are too specialized in most topics, and not sufficiently specialized in the areas in which the community is deeply interested. We propose a new formulation for discovering a topic hierarchy specifically expressing and addressing the interests of the community, refining topics where needed and coarsening where possible. Details of the new classification and theme discovery algorithms are reported elsewhere [4] (also see Figure 4).

Once topic hierarchies for the user community are determined, automatic resource discovery is undertaken by demons to update users about recent and/or authoritative sources, organized by topic [5]. 'Normalizing' all members of the community to themes also lets us represent surfers' interests in a *canonical form*: roughly speaking, a user profile is a set of weights associated with each node of a theme hierarchy; this gives us a means of comparing profiles that is far superior to overlap in sets of URLs. We intend to use this for better collaborative recommendation [10].

5 Related work

Our work is closest in spirit to two well-known systems, PowerBookmarks[9] and the Bookmark Organizer [8].

PowerBookmarks is a semi-structured database application for archiving and searching bookmark files via explicit CGI programs. PowerBookmarks uses Yahoo! for classifying the bookmarks of all users. In contrast, Memex preserves each user's view of their topic space, and reconciles these diverse views at the community level. Furthermore, PowerBookmarks does not use hyperlink information for classification or for synthesizing themes. The Bookmark Organizer is a client-side solution for personal organization, but does not provide community-level themes or topical surfing contexts. Purple Yogi[10] is a client-side software which logs pages visited and clusters them into folders. Then it tunes in on the Purple Yogi server to collect additional related material. No community-level mining is involved; Purple Yogi explicitly guarantees that user-specific data is stored locally on the user's desktop and never shipped out. Thus scope for valuable collaboration is lost and surfing history becomes inaccessible from other places from which the user might browse.

Other Internet start-ups have been quick to discover the annoyance of surfers maintaining multiple bookmark files and the opportunity of a central, networked bookmark server. We can list several sites which, using Javascript or a plugin, import existing Netscape or Explorer bookmarks and thereafter lets the surfer visit their Web site and maintain it using CGI and Javascript: Yahoo Companion[11], YaBoo[12], Baboo[13], Bookmark Tracker[14], and Backflip[15] are some examples. Some services like Third Voice[16] enables surfers to attach public or private annotations to any page they visit. These are essentially glorified FTP services with none of our extensive server-side analysis.

Several visualization tools have been designed recently that explore a limited radius neighborhood and draw clickable graphs. These are often used for site maintenance and elimination of dead links. Mapuccino and Fetuccino from IBM Haifa are well known examples [7, 1]. Our context viewer could benefit from better hypertext rendering techniques.

References

[1] I. Ben-Shaul, M. Herscovici, M. Jacovi, Y. S. Maarek, D. Pelleg, M. Shtalheim, V. Soroka, and S. Ur. Adding support for dynamic and focused search with Fetuccino. In *8th World Wide Web Conference*. Toronto, May 1999.

[2] V. Bush. As we may think. *The Atlantic Monthly*, July 1945. Online at `http://www.theatlantic.com/unbound/flashbks/computer/bushf.htm`.

[3] S. Chakrabarti, B. Dom, R. Agrawal, and P. Raghavan. Scalable feature selection, classification and signature generation for organizing large text databases into hierarchical topic taxonomies. *VLDB Journal*, Aug. 1998. Invited paper, online at `http://www.cs.berkeley.edu/~soumen/VLDB54_3.PDF`.

[4] S. Chakrabarti, S. Srivastava, M. Subramanyam, and M. Tiwari. Archiving and mining community web browsing experience using Memex. In *9th International World Wide Web Conference*, Amsterdam, May 2000.

[5] S. Chakrabarti, M. van den Berg, and B. Dom. Focused crawling: a new approach to topic-specific web resource discovery. *Computer Networks*, 31:1623–1640, 1999. First appeared in the 8th International World Wide Web Conference[17],

[7] `http://www.yahoo.com`
[8] `http://dmoz.org`
[9] `http://www.ccrl.neclab.com/webdb/`
[10] `http://www.purpleyogi.com`
[11] `http://www.yahoo.com/r/cm`
[12] `http://www.yaboo.dk`
[13] `http://www.baboo.com`
[14] `http://www.bookmarktracker.com`
[15] `http://www.backflip.com`
[16] `http://www.thirdvoice.com`
[17] `http://www8.org`

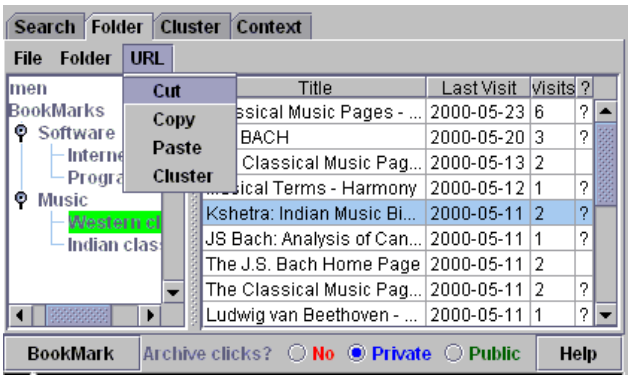

Figure 1: Each user has a personal folder/topic space, which is usually initialized by importing existing browser-specific bookmark folders. The classification demon then classifies all subsequent history elements, marking its guesses by '?'. The user can correct or reinforce the classifier using cut/paste, thus continually improving Memex's models for the user's topics of interest.

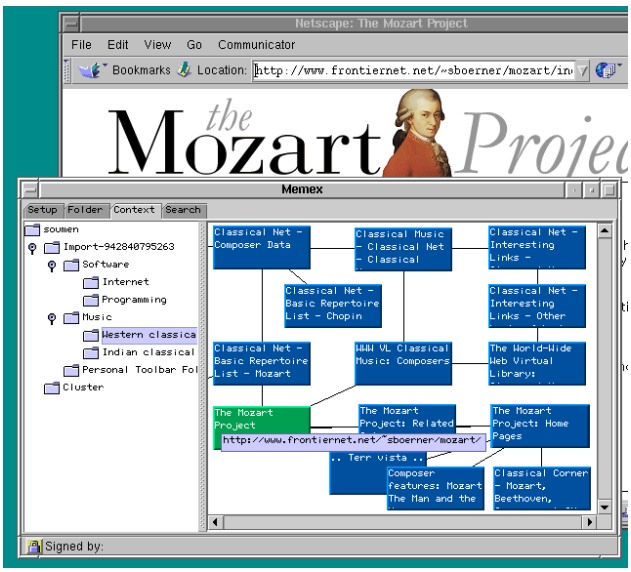

Figure 2: The trail tab shows a read-only view of the user's current folder structure. When the user selects a folder, Memex replays recently browsed pages which belong to the selected (or contained) topic(s), reminding the user of the latest topical context. In the screen-shot above, the chosen folder is */Music/Western Classical*. The user can now resume browsing and the display is updated with additional resources related to the topic.

Toronto, May 1999. Available online at `http://www8.org/w8-papers/5a-search-query/crawling/index.html`.

[6] D. R. Cutting, D. R. Karger, and J. O. Pedersen. Constant interaction-time scatter/gather browsing of very large document collections. In *Annual International Conference on Research and Development in Information Retrieval*, 1993.

[7] M. Hersovici, M. Jacovi, Y. S. Maarek, D. Pelleg, M. Shtalheim, and S. Ur. The Shark-Search algorithm-an application: Tailored web site mapping. In *7th World-Wide Web Conference*, Brisbane, Australia, Apr. 1998. Online at `http://www7.scu.edu.au/programme/fullpapers/1849/com1849.htm`.

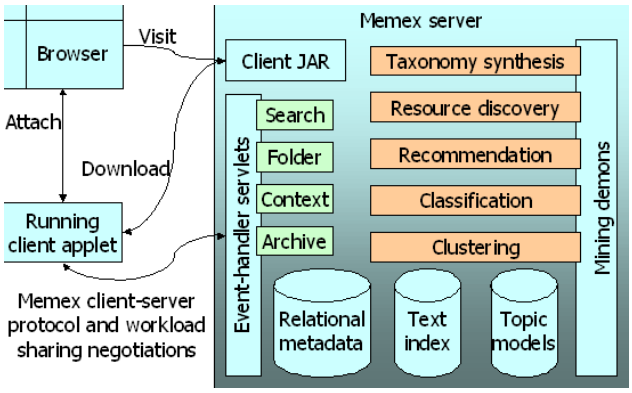

Figure 3: Block diagram of the Memex system, showing the client-server interface, the UI event handlers, the mining demons, and the loosely synchronized data repositories.

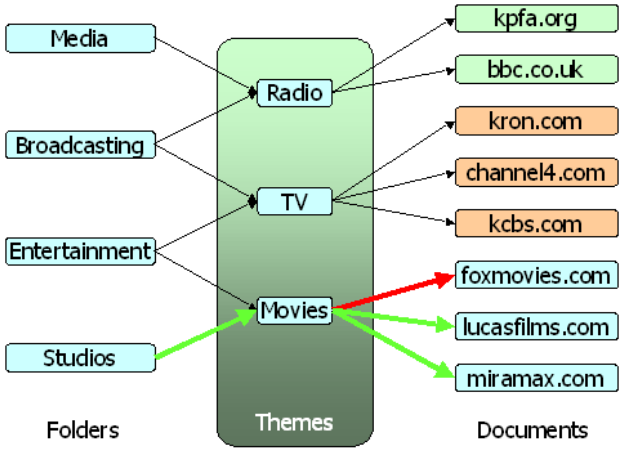

Figure 4: Memex computes, from the document-folder associations of multiple users, a topic taxonomy specifically tailored for the interests of that user population. The taxonomy consists of *themes* which capture common factors in people's interests when they can, while maintaining individuality when they must. Once computed, they can be used to guide resource discovery and collaborative recommendation.

[8] Y. S. Maarek and I. Z. Ben Shaul. Automatically organizing bookmarks per content. In *Fifth International World-Wide Web Conference*, Paris, May 1996.

[9] P. P. Maglio and T. Matlock. Metaphors we surf the Web by. In *Workshop on Personalized and Social Navigation in Information Space*, Stockholm, Sweden, 1998.

[10] L. H. Ungar and D. P. Foster. Clustering methods for collaborative filtering. In *AAAI Workshop on Recommendation Systems*, 1998. Online at `http://www.cis.upenn.edu/~ungar/papers/clust.ps`.