Memento: Unifying Content and Context to Aid Webpage Re-visitation

Chinmay Kulkarni, Santosh Raju, Raghavendra Udupa
Microsoft Research India
196/36 2nd Main
Bangalore, 560 080, India
{v-chinku, t-sanv, raghavu}@microsoft.com

ABSTRACT
While users often revisit pages on the Web, tool support for such re-visitation is still lacking. Current tools (such as browser histories) only provide users with basic information such as the date of the last visit and title of the page visited. In this paper, we describe a system that provides users with descriptive topic-phrases that aid re-finding. Unlike prior work, our system considers both the content of a webpage and the context in which the page was visited. Preliminary evaluation of this system suggests users find this approach of combining content with context useful.

ACM Classification: H5.4 [Information interfaces and presentation]: Hypertext/Hypermedia- Navigation

General terms: Experimentation, Human Factors

Keywords: Browsing history, Topic phrases, Internet search

INTRODUCTION
Users often revisit webpages on the Internet, both to obtain updated information and to re-visit information they have seen before. It is conservatively estimated that users revisit nearly half of all webpages they see [2]. Unfortunately, such “re-finding” is often difficult using today’s web-browsers. Users have reported “Not being able to return to a page I once visited,” as one of “the biggest problems in using the Web” [3].

Aiding users to re-find pages is an actively studied area, and prior work primarily focuses on two approaches: the first approach focuses on adapting search engines to aid re-finding [5]. Using search engines, however, has its own limitations: users have reported being unable to “remember the terms that I used when I found a relevant site”. Furthermore, search engines do not index all pages on the Internet (the “deep web”)—deep-links a user follows are especially troublesome. [2]

The second approach uses client-side browser add-ons that enrich user-history with semantic information [4]. Such approaches are a definite improvement over traditional browser history and do not suffer from the privacy concerns of server-based systems. However, these systems summarize the content of a webpage in isolation, and disregard the context in which the page was visited. Disregarding context has two disadvantages: first, the target webpage could have little information that can be extracted (for instance, when the target page has a video). In such a case, considering the page from which a user navigated to the target page often yields more easily extractable information (such as a description of the video).

More importantly, as noted in [5], users are often unable to reliably remember the exact content of a webpage, but tend to remember the approximate context in which the page was seen. Users often then use this approximate context to “orient” them (possibly using a search engine); subsequently navigating to their target page in small, local steps.

The above reasons suggest that capturing context in addition to the content of a webpage would help users re-find it more successfully. In this paper, we describe a completely automatic system, Memento, that finds topic-phrases for a webpage based both on the page content and the context in which the page was visited. Memento is completely client-side, so 1) it preserves user privacy and 2) is able to index pages that are not publicly accessible (e.g. Intranet pages). The topic-phrases generated by Memento could be used to improve existing history functionality, or as part of a search engine intervention that suggests pages from a user’s history, related to the current search terms.

MEMENTO
We define the context of a web page as other pages in the browsing session that immediately precede or follow the current page. Table 1 compares topic-phrases generated by Memento for a URL visited in two different contexts (target page is shown in bold; pages visited immediately after this page are omitted for clarity). In the first session, the user is possibly looking for information about an automobile recall. In the second session, the user arrives at the same target page, but navigated to it along a different path, and is possibly looking for pricing. In the two cases, Memento extracts different topic-phrases for the same target page, reflecting the difference in the page context.

Identifying Page Context
Identifying page-context plays a key role in extracting topic-phrases for a target page. Memento identifies the page-
context by determining its session boundaries. The length of a session (i.e., the number of pages considered to be a page’s context) varies considerably based on the user’s information need. Prior work has used the time-interval between successive page loads or a page-window of fixed size to identify session boundaries [2]. However, due to the large variance in session-length, we find neither measure performs adequately and that using content similarity between successive pages yields better session boundaries for our purposes. First, we parse the content of each HTML page to obtain a term-frequency vector. A pair of consecutive pages is considered a session boundary if both pages have non-null term-frequency vectors and the cosine similarity between their vectors is below a set threshold.

Table 1: Topic-phrases for identical URL in different sessions

<table>
<thead>
<tr>
<th>Session</th>
<th>Topic-phrases</th>
</tr>
</thead>
</table>

**Topic Mining**

We mine topic-phrases using content from the page and a pool of pages identified as its context. Candidate topic-phrases are extracted from this set of pages using a filter based on titles from Wikipedia. This ensures phrases are well-formed and meaningful. Candidate phrases are then co-clustered to identify the subset of topic-phrases relevant to the page [1]. Relevance scores of these phrases are renormalized to account for the user’s general browsing pattern using a TFIDF measure and top-N topic-phrases are displayed. Since Memento runs on a user’s own machine, we use a space-efficient Bloom filter for our Wikipedia topic list. This allows us to use a large list of potential topics (~6 million Wikipedia topic titles) with a low footprint (~9 MB).

**EVALUATION**

We evaluated our system with 8 participants drawn from our organization. All users used Internet Explorer. Participants ran our stand-alone program which analyzed pages visited within the last 30 days to find topic-phrases. The program then chose five pages such that no two pages had any topic-phrases in common. Users then saw each page along with its associated topic-phrases, and rated the relevance of the phrases to the page on a scale of 1 to 7 (1: entirely irrelevant; 7: entirely relevant). Based on their ratings, four users were then chosen to participate in a semi-structured interview.

**Results**

Users rated topic-phrases for 36 pages in all (users were shown less than 5 pages when the program couldn’t find pages with non-intersecting topic-phrases). The median rating obtained was 4.5 (SD= 2.42). Fig. 1 shows the distribution of ratings. While the largest number of pages yielded “entirely relevant” topic-phrases, several others were reported to yield topic-phrases with little relevance.

**Figure 1: Distribution of reported relevance**

During the interviews, we discovered two main causes for lower ratings. First, for some pages (such as technical documentation), page-content was remarkably more important than the page-context, so topic-phrases that captured context were seen as irrelevant. Second, participants gave low ratings when they expected a particular topic-phrase and did not find an exact match (e.g. P7 was looking for “Cricket”, but we found the page was about “Fantasy Cricket”; P5 was disappointed we found “Event Log” as a topic-phrase, but not the exact API call he was using for event logging).

Lastly, we note that 6 of the 11 pages that resulted in “entirely relevant” topic-phrases had at least one topic-phrase that was not present in the target page but was present in the page-context. Systems that ignore page-context would miss these completely.

**FUTURE WORK**

Our current system can easily be modified to assign different weights to the page-content and page-context. We plan to conduct more detailed studies to understand when content is more important, and appropriately assign weights in our algorithm. We plan to integrate Memento into existing browsers. This will also help us evaluate how Memento performs in conjunction with other re-visititation support like exact-match search for browsing history.

**REFERENCES**

5. Teevan, J., Alvarado, C. and Ackerman, M.S. The perfect search engine is not enough: a study of orienteering behavior in directed search. In *Proc. CHI* 2004