The Re:Search Engine
Helping People Return to Information on the Web

Jaime Teevan
Massachusetts Institute of Technology
32 Vassar Street, G472
Cambridge, MA 02139 USA
+1 (617) 253-1611
teevan@mit.edu

ABSTRACT
Re-finding information is commonly cited as a problem on the Web. One reason re-finding on the Web is difficult is that while people rely on a considerable amount of context to return to information (*e.g.*, the original path taken to it), the Web makes no guarantee that the context will remain static. The *Re:Search Engine* is designed to help people return to information in the dynamic environment of the Web by maintaining consistency in the search results it returns across time. For example, if Connie, while looking to purchase a Global Positioning System, found several systems she liked via a search for "GPS", she would expect to be able to use the same query to locate the exact same systems again. However, simply returning the original result list when she re-issues the query might omit newly available GPS systems that she would like to see. The ideal result list would contain both the systems Connie remembers having seen and high quality new systems. Because people tend to remember little of what is presented in a result list, when a person repeats a query, the Re:Search Engine can preserve what is remembered about the original result set while still presenting new information.

Categories and Subject Descriptors
H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *Search process*, *Relevance feedback.* H.5.2 [**Information Interfaces and Presentation (e.g., HCI)**]: User Interfaces.

General Terms
Design, Experimentation, Human Factors.

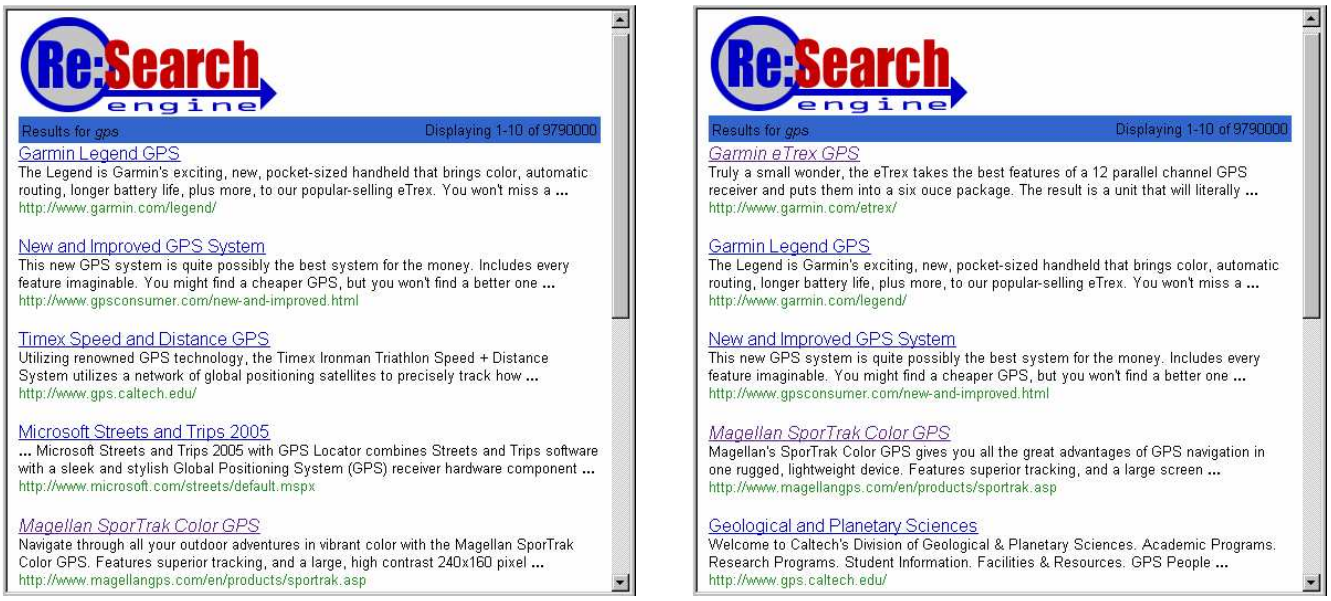
Keywords
Re-finding, personalization, information management, implicit feedback, user profiling, dynamic information.

1. PEOPLE RELY ON CONSISTENCY WHEN SEARCHING …
People rely on consistency in their information environment when searching for information, and are particularly likely to expect consistency when searching for previously encountered information. Consider the example search shown in Figure 1. If Connie, while looking to purchase a Global Positioning System, found several systems she liked through a search for "GPS", she would expect to be able to use the same query to locate the exact same systems again.

The importance of consistency has been emphasized through two studies conducted to gain insight into how people return to information. One, a modified diary study of fifteen computer science graduate students performing personally motivated searches in their email, in their files, and on the Web, found that even among this technically savvy population, participants preferred to navigate to what they were looking for along known paths over jumping directly to it [11]. This preferred search strategy will fail if any part along the known path changes. Similarly, a naturalistic study analyzing instances of re-finding mined from Web pages found that when people expressed difficulty re-finding information, they were relatively unlikely to describe the temporal aspects of their original encounter with the information, and instead commonly described it using the path they took to originally find the information [13]. The results of these studies are consistent with the findings of others [2].

2. … BUT THE WEB CHANGES
Despite the importance of consistency in re-finding, information on the Web frequently changes. Search results, often an

Figure 1. Connie's initial results for the query "GPS".



(a) Current Web results

(b) Results show to user by Re:Search Engine

Figure 2. An example of the Re:Search Engine in action. Figure 1 shows the search results when Connie first searched for "GPS" (visited links are *italicized*). Figure 2(a) shows the results when the query is next performed, and Figure 2(b) shows how the Re:Search Engine combines what Connie is likely to remember from Figure 1 with what is new in Figure 2(a).

important step in the original access path to a piece of information, change regularly as search engines update their indices to reflect the current state of the Web. The high rate of change to search engine results can be illustrated through analysis of the top ten results for ten queries issued to Google and tracked over the course of a year. On average, only 2.7 of the results remained in the top ten after a year; three results disappeared from the list within the first month. Thus, if Connie wanted to revisit two of the top ten GPS systems she found during her original search for "GPS", she would have a 51% chance of not being able to locate one of them after a month. Attempts to improve retrieval quality through personalization or collaborative filtering are likely to increase the frequency of search result list changes.

Although Web search engines have traditionally sought to return the search results that are the most relevant to a query without consideration of past user context, some recent search systems, such as A9 [1], allow users to mark pages of interest to return to later. However, people are unlikely to employ keeping strategies that require active involvement [6]. Some search engines also allow people to explicitly search within information they have seen before [1, 4], but these systems do not maintain consistency in result presentation, requiring the user to take a different path to the same information. Information management systems that preserve consistency in dynamic environments permit their users choose to interact with a cached version of their information space [5, 10]. While employing similar methods to keep the results for repeat queries static would make re-finding simpler, it would deny users the opportunity to discover new information. For example, if Connie re-issues her "GPS" search, in addition to re-finding the systems she liked before, it is possible she would also be interested in learning about newly available systems. Even though changes to search results associated with a query can potentially hinder returning to previously viewed information, they benefit users by providing new information.

3. SOLUTION–THE RE:SEARCH ENGINE

The *Re:Search Engine* addresses the dual goals of maintaining search result consistency and providing new information by seamlessly integrating old relevant information with new. The engine interfaces with a preexisting search engine (*e.g.*, Google or Yahoo!). When a person issues a query that has been issued before, the Re:Search Engine first fetches the current results for that query from the underlying search engine. It then merges this newly available information with a cached copy of the results that were previously presented to the user, leaving unchanged what the user *remembers* about the initial result set. Because people tend to remember very little about the search result list they originally saw, it is possible to preserve the salient features of the old results while still presenting new information.

Consider as an example Connie's search. Recall that Figure 1 shows the results returned when Connie first searched for "GPS". Later, when she re-performed the same query, the results had changed to include several new GPS systems (Figure 2a). Instead of directly returning the new results, which could be disorienting, or the original results, which might omit items Connie would want to see, the Re:Search Engine merged the two (Figure 2b). The merging preserves memorable aspects of the original results, such as followed links (*italicized*), anomalous results ("Geological and Planetary Sciences"), and the ordering of the first and last results, while including new results and an updated result summary.

An exploratory paper prototype study of people interacting with lists of document summaries suggests that many changes, such as changes to the summary wording and to the document ordering, go unnoticed, even when the changes occur as the person interacts with the information [12]. A challenge in designing a system that takes advantage of the fact that people don't notice all changes is to identify which aspects of the information a person interacts with are memorable (and thus should only be changed with care), and which are not (and can change as needed).

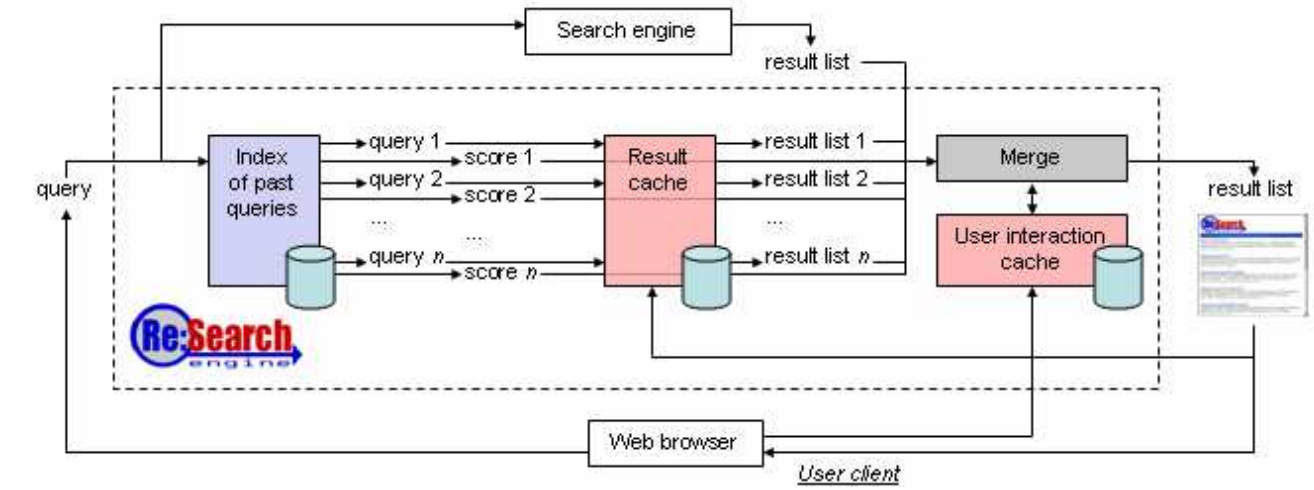


Figure 3. The architecture of the Re:Search Engine. The user's current query is matched to past queries, and the results for the past queries are retrieved from a cache. These results are then merged with the live search engine results based on how memorable the results are, and the resulting result list is presented to the user.

4. RE:SEARCH ENGINE ARCHITECTURE

The architecture of the Re:Search Engine is shown in Figure 3. When a person performs a search via the Re:Search Engine, the query is passed through an *index of past queries* that returns similar previously issued queries. These matching queries are used to retrieve the associated previously viewed search results from a *result cache*. Using information stored in the *user interaction cache*, the past results are then *merged* with the live results for the current query and the merged list is returned to the user. Finally, the current query is added to the index of past queries, the merged result list is added to the result cache, and the user interactions with the returned result list are logged. Each of these components is described in greater detail below.

The design of the system is influenced by a study of what 119 people found memorable about search result lists. In the study, participants were initially asked to interact with a search result list and then later asked to recall their query and the results they interacted with without referring back to the original information.

4.1 Index of Past Queries

The purpose of the *index of past queries* is to identify repeat searches. The index functions similarly to a traditional document index, except that the "documents" indexed are query strings. An index was chosen for the query matching both for efficiency and because it accurately reflects how people remember their past queries. For example, word ordering, tense, capitalization and stop words are commonly forgotten when recalling search terms, and these features are removed when a query is indexed.

Not every query issued with similar text to a past query indicates a repeat search. During search session, people commonly explored variants of the same query, actively seeking new results with each variant. For example, if Connie thought the results she received for her query for "GPS" returned too many expensive systems, she might try searching for "GPS, cheap". The results of this search should not be merged with the results for the query issued immediately prior. For this reason, past queries that are similar but that occurred recently are ignored.

4.2 Result Cache

If the query the user issued is determined to be related to one or more previous searches run by the user, the results corresponding to the previous searches are fetched from a *result cache* using the pervious queries returned by the past query index as keys. Only the most recently viewed set of results for a particular query is stored in the cache. For example, when Connie issued the query "GPS" a second time, the results shown in Figure 2(b) replaced the results shown in Figure 1 in her result cache.

4.3 User Interaction Cache

Once past results of possible relevance to the current query are fetched, they are merged with the live search results to produce a list containing old and new results to return to the user. The merge algorithm is designed to help people take advantage of the context built during past searches, and thus requires an understanding of how memorable past results are. Implicit measures of attention, like those discussed by Kelly and Teevan [7], suggest what one might have noticed during a search. These measures are collected by instrumenting the user's browser to observe the user's interactions with previous result sets and are stored in a *user interaction cache*.

4.4 Merge Algorithm

In the merging of old and new result lists, each old result is given a *memorability* score. This score was developed through analysis of what people remember about search result lists, and is computed using past user interactions with the results (*e.g.*, whether or not the associated Web page was visited), static information about the result (*e.g.*, its rank in the result list), and the result's associated query (*e.g.*, the query's relevance to the current query and its recency).

Changing the presentation of a memorable result incurs a cognitive cost because it no longer occurs where expected. This cost is represented by assigning a *cost of change* to the types of changes a previously viewed result can undergo. For example, changing the rank of a search result incurs a small cost, while removing a search result from the search result list incurs a large

cost. Like the memorability score, the cost of change is based on actual user behavior. Because a change to a memorable results incurs a greater cognitive cost than a change to a result that is hardly remembered at all, the cost of any given result list is a function of result memorability and the cost of making the changes necessary to produce that list.

Additionally, each result in the new result list for the current query is given a *benefit of new information* score based on the expected benefit the as yet unseen result will provide to the user. If scoring information is available from the underlying search engine, the result's score can be used to represent the expected benefit. However, scoring information is often not available, so the Re:Search Engine uses the result's rank as a proxy. Beneficial results are more likely to be seen if they occur high in the returned result list, and the benefit of a result list is based both on each individual result's benefit and its location in the list.

During the merge process, all permutations of possible final result lists that include at least two old results and two new results are considered. The result list with the highest total benefit minus cost is selected and returned to the user. Although considering all permutations naively is expensive, the merge algorithm can be implemented efficiently by representing the problem as a min-cost network flow problem.

5. EVALUATION PLAN

An underlying principle of the Re:Search Engine is that search results should not merely contain the information most relevant to a searcher's immediate need. Instead, results should account for previous interactions with related information, making it easy to take advantage of past context. To test such a principle requires including the user directly in the evaluation. As such, the Re:Search Engine will be evaluated through two user studies.

The first study will involve inviting people into the lab twice, once to perform an initial search, and a second time to re-search for information encountered during the initial search. Lab studies allow for the conduct controlled studies and the examination of participant's thought processes by having them think aloud as they search [7]. Such a framework will permit for the exploration of a number of result orderings, including, in addition to the Re:Search Engine's result ordering, a static ordering, a dynamic ordering, and an ordering where the most memorable results are presented first. The success of each ordering will be measured subjectively ("Does the user like the result list?") and objectively ("Does the user successfully and quickly complete the task?").

A drawback to a lab study is that it can introduce artificialities that bias behavior. This is particularly true for re-finding, because it difficult to motivate a repeat search without over-specifying the target. To gain a realistic understanding of how the Re:Search Engine will be used in practice, a large-scale deployment is planned. Because there are limits to the number of interfaces that can be explored through large-scale deployment, as well as to the quality of data that can be collected this way, such a study will be done to test a relatively mature version of the Re:Search Engine.

6. GENERALIZING THE SOLUTION

Although the Re:Search Engine focuses on maintaining consistency between old search result lists and new, other types of information people commonly interact with also change. For example, online news sites change when new stories are written, and Web sites change as their hosts edit them. The growing ease of electronic communication and collaboration, the rising availability of time dependent information, and even the introduction of automated agents, suggest information is becoming ever more dynamic. As stated by Levy, "[P]art of the social and technical work in the decades ahead will be to figure out how to provide the appropriate measure of fixity in the digital domain [8]." Understanding how people interact with search results on the Web while re-finding will shed light on how people return to information in dynamic environments in general, and I look forward to applying what I learn from the Re:Search Engine to other problems in the domain.

7. ACKNOWLEDGMENTS

I appreciate the support of my advisor, David Karger, and my committee, Mark Ackerman, Susan Dumais and Robert Miller.

8. REFERENCES

[1] A9, http://www.a9.com

[2] Capra, G. and Pérez-Quiñones, M. A. (2003). Re-finding found things: An exploratory study of how users re-find information. Technical Report cs.HC/0310011, Computing Research Repository (CoRR).

[3] Graphic, Visualization, and Usability Center (1998). GVU's tenth WWW user survey (conducted October 1998).

[4] Dumais, S., Cutrell, E., Cadiz, J. J., Jancke, G., Sarin, R. and Robbins, D. C. (2003). Stuff I've Seen: A system for personal information retrieval and re-use. In Proceedings of SIGIR '03, pp. 72-79.

[5] Hayashi, K., Normura, T., Hazama, T., Takeoka, M. Hashimoto, S. and Grudmundson, S. (1998). Temporally threaded workspace: A model for providing activity-based perspectives on document spaces. In Proceedings of HyperText '98, pp. 87-96.

[6] Jones, W., Bruce, H. and Dumais, S. (2003). How do people get back to information on the Web? How can they do it better? In Proceedings of INTERACT '03, pp. 793-796.

[7] Kelly, D. and Teevan, J. (2003). Implicit feedback for inferring user preference: A bibliography. SIGIR Forum, 37(2):18-28.

[8] Levy, D. (1994). Fixed or fluid? Document stability and new media. In Proceedings of European Conference on Hypertext, pp. 24-31.

[9] Murdock, B. B. (1962). The serial position effect of free recall. *Journal of Experimental Psychology*, 64:482-488.

[10] Rekimoto, J. (1999). Time-machine computing: A time-centric approach for the information environment. ACM Symposium on User Interface Software and Technology 1999, pp. 45-54.

[11] Teevan, J., Alvarado, C., Ackerman, M. S. and Karger, D. R. (2004). The perfect search engine is not enough: A study of orienteering behavior in directed search. In Proceedings of CHI '04, pp. 415-422.

[12] Teevan, J. (2001). Displaying dynamic information. In Proceedings of CHI '01 (Extended Abstract), pp. 417-418.

[13] Teevan, J. (2004). How people re-find information when the Web changes. MIT AI Memo AIM-2004-012.