ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

# A Large Scale Analysis Of Information Re-Finding System

Prakash.M[1], Karthika.D[2], Sophia.J[3]

M.E. Computer Science and Engineering, Arunai Enginnering College, Thiruvannamalai, India[1]

M.E. Computer Science and Engineering, Arunai Enginnering College, Thiruvannamalai, India[2]

M.E. Computer Science and Engineering, Arunai Enginnering College, Thiruvannamalai, India[3]

**ABSTRACT:** We present a context-based information re-finding is known as Re-Finder. It power of human's natural recall characteristics and grant users to re-find files and Web pages based on their previous access context. The Re-Finder re-finds the information based on an objection by context model over the context memory snapshot and linking to the accessed information contents. The Context occurrence in the memory snapshots are arranged in a clustered and associated aspect and dynamically emerges in life cycles. Here we evaluate the scalability of Re-Finder on a large synthetic data set. The results of experiments shows the persistent degradation of context occurrence in the context memory in users refinding requests can lead to the best re-finding precision and recall. Initial finding shows that the time, activity and, place it helps to re-find the information and could serve as useful recall clues. It improves the re-finding requests compare to the existing methods.

**Keywords:** Re-finding information; context-based; context-annotation; context degradation; memory organize.

## I. INTRODUCTION

Nowadays people are experiencing unprecedentedly data explosion, reading, writing, and collecting different kinds of information from local computer and the global Web. Once in a while, people revisit information that have ever been come across occasionally or intentionally. The explosion in the amount of personally accessed information has made re-finding certain targets time consuming. It faces grand challenges just as arduous as information finding itself.

Information refinding is different from information finding. There is uncertainty in the latter process because users do not know enough information, while refinding is a more directed process as users have already seen the information before. A general way to support information refinding is to maintain access logs recording what users have ever seen and it based on their revisit frequencies, say, an hour ago, one day ago, one month ago, and so on. As the logs grow with time, users commonly prefer searching to browsing the logs for the information which was accessed particularly a long time ago. However, because of human users' dim memories of the past sometimes it is a difficult and time-consuming task for them to refind what they want by simply entering keywords of the previous accessed information contents.

### *A. Inspiration from Human Memory*

Psychological studies show that context under which information accessed before can serve as 1a powerful cue for information recall, as it is always easier to remember than detailed information content itself. For example, it may be hard to recall a recipe's detail encountered one year ago, but the time, place, and concurrent activity associated with the happening of that access event may leave a deeper impression, this could serve as useful cues to refind the target recipe.

## II. RELATED WORK

The topic of information refinding is explored extensively by two major communities: Web search and personal information management communities.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

*A. Web Search*

On the Web, a diversity of methods have been devised to organize Web information for reaccess and reuse. Typical techniques include bookmarks, history lists, search engines, and so on. MacKay et al. proposed landmark which is an extension to the traditional bookmarks. It is a user-directed technique that aids users in returning to specific content within a previously visited Web page. Contextual Web history tool improves the visual appearance of the history by combining thumbnails of Web sites and snippets of contents, assisting users to easily browse or search the history by time. Google's Web history keeps users' search requests and clicked pages and classifies them into Different topics such as images, news, and so on, and allows users to navigate or search accessed Web pages by keywords from accessed page titles and contents. The Search Bar tool allows users to organize their search keywords and clicked pages under different topics. Users can make notes on the topics for easy navigation. Teevan built a Research system supporting simultaneous finding and refinding on the Web. When a user's query is similar to a previous query, it obtains the current results from an existing search engine, and fetches relevant viewed results from its cache. The newly available results are then merged with the previously viewed results to create a list that supports intuitive refinding and contains new information.

## III. PROPOSED METHODOLOGY

*A. Context-Based Refinding*

Context-based refinding differs from the traditional database query conceptually in three aspects. First, request formulation is based on contextual attributes rather than database contents. Second, query target is context memory snapshot rather than database. Third, an intermediate query result is a ranked list of context instances, with their linked information as the final query result. At the implementation level, the query target (i.e., context memory snapshot) is organized in a hierarchical, cluster and associated manner, and dynamically evolves in life cycles according to query user's memorization strength.

For the final result generation via the context instances is quite straightforward, we focus on the intermediate result computation in the following discussion.

*Context-Based Refinding Model*

A context-based refinding query can be denoted as a function RF (Q, CM)=(C1,C2, ; Cm), where Q is the query request formulated in the form of a context instance, CM is the query target that is the context memory snapshot, and the intermediate query result of Q upon CM is a ranked list of context instances in CM, (C1; C2; . . . ; Cm), whose ranking is determined by a ranking function. In the study, we consider three different ranking methods based on simple similarity, weighted similarity, and negative dissimilarity between Q and C. Let Q =(q1, q2, . ; qn), and C =(c1, c2; . ; cn) without loss of generality. Ranking by simple similarity. A straightforward way is to use the similarity function (see (1)) to rank context instances in the memory snapshot against Q:

Ranking by weighted similarity. Considering that a user's query request Q may be vague as well due to the vague memory along with time, and some contextual attribute values like activity may leave a deeper impression than others such as time, we incorporate a weight vector (w1, w2,…, wn) for different contextual attribute values in Q to state their precise degrees, where $w_i$ €[0, 1]_ for every 1≤ i≤ n,

$$Rank(Q,C) = \sqrt{\sum_{i=1}^{n} w_i \cdot sim^2(A_i, q_i, c_i)}.$$

*Association of Context Instances*

For each contextual attribute Ai, and every value in its hierarchy, we build an association chain Chain ($A_i$, v), which consists of all the context instances with the same attribute value of Ai. That is, for any context instance C 2 Chain (Ai, v), ($c_i$ = v). Fig. 2 illustrates six 3D context instances in the memory snapshot. A few chains are illustrated in the left of Fig. 2. To facilitate exactly and specifically matching between the query and context instances, we extend association chains to include all the descendants based on the contextual attribute hierarchies, and obtain EChain ($A_i$, v), so that for

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

any context instance C 2 EChain ($A_i$, v), ($c_i$ = v) or ($ci_i$<a v). For example, since 2010-09, 2010-10 <a 2010, Chain (A1, 2010) is extended to include {2010, 2010-09, 2010- 10}, as shown in the right of Fig. 2.

### *B. Cluster-Association-Based Refinding Algorithm*

We explore context cluster and association relationships in refinding. Given a query Q, the cluster-based approach checks every context in the candidate clusters. If we build association chains for the context instances within each cluster on the right attribute, we probably could skim the irrelevant chains immediately. The time cost of association-based refinding approach depends on the length of the selected extended association chain, which guides us to choose an appropriate attribute to build association chains for context clusters, i.e., select the attribute that can scatter the context instances by the greatest extent.

Fig. 7 shows an example of context cluster association relationships. The context instances are clustered on the time attribute There are three clusters with 2010, 2010-09 and 2010-10 as their representative attribute values respectively. The first two are built association chains on the location attribute (Asso-dim = 2) and the third one is built on the activity attribute (Asso-dim = 3).

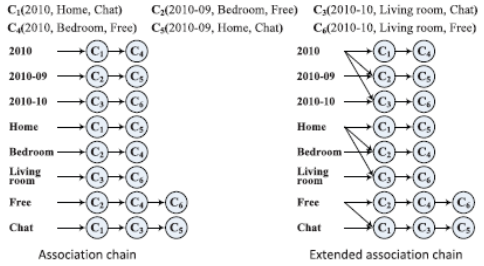


Fig. 1. Context Assosiation Relationships.

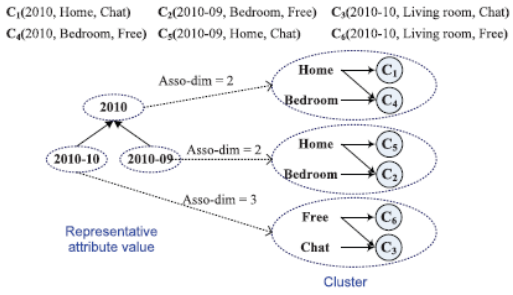


Fig. 2. Context Cluster-Association Relationships.

## IV. SYSTEM DESING

### *A. Implementation of Refinder System*

We implement a context-based information refinding system called ReFinder. It facilitates users to annotate any interesting Web pages or local files encountered with access contextual information, and enables users to refind them later by the previous access context.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

*Organize local files and web pages*

This process Organize local files and Webpage link's based on previously visited Web page and files. It keeps users' search requests and clicked pages and classifies them into different topics and allows users to navigate or search accessed Web pages by keywords from accessed page titles and content. A general way to support information re-finding is to maintain access logs, recording what users have ever seen based on their revisit frequencies, say, an hour ago, one day ago, one month ago, and so on. As the logs grow with time, users commonly prefer searching to browsing the logs for the information which was accessed particularly a long time ago. However, because of human users' dim memories of the past; sometimes it is a difficult and time-consuming task for them to re-find what they want by simply entering keywords of the previous accessed information contents. The Search Bar tool allows users to organize their search keywords and clicked pages under different topics.

*Context Annotation And Clustering*

This component facilitates users to annotate any interesting web pages or local files encountered with access contextual information, and enables the users to re-find them later by the previous access context. The overall process of the context annotation and clustering method is illustrated in Fig.5.

Three contextual attributes are,

- Time.
- Place.
- Activity.

These are considered in *Re-Finder also* we explore context cluster and association relationships in re-finding. Re-finder provides a small translucent window at the right corner of the computer screen, by double clicking which users can do context annotation t any opened file or viewed web page. Time is the current date to be filled in automatically by the system. The user can either manually enter location and concurrent activity, or select appropriate ones from a predefined Location hierarchy and Activity hierarchy. Re-Finder allows the user to maintain the hierarchies of contextual attributes by adding, deleting, or renaming attribute values.

*Information Access Phase*

When a user accesses (read, write, or modify) files from the local computer, or Web pages from the global Internet, ReFinder system allows the user to annotate any of the encountered interesting files or Web pages with the access context. ReFinder designs an IE browser plug-in to acquire the titles and urls of the currently visited Web pages and provides an interface to record user-input contextual information (see Fig. 7).


Fig. 3. Context Based Information Refinding Interface.

Alternatively, users can double click the small translucent window at the right corner of the computer screen to do context annotation to any opened file or viewed Web page.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

Three contextual attributes (time, place, and activity) are considered in ReFinder. The main interface of the refinder shown in the Fig.4.Time is the current date to be filled in automatically by the system. The user can either manually enter location and concurrent activity, or select appropriate ones from a predefined Location hierarchy and Activity hierarchy by clicking the right-hand zoom button. ReFinder allows the user to maintain the hierarchies of contextual attributes by adding, deleting, or renaming attribute values. A context annotation example for an encountered Web page is as follows:[Time: [2012-02-17], Place: Lab, Activity: Do project]

*Information Re-find*

component accepts user's context-based re-finding requests, and returns the result files/web pages. *Re-Finder* accepts users' re-finding requests by their previous access context. It then finds matched context instances, linking to the recalled information, in a context Memory snapshot. Fig. 8 illustrates the overall architecture of ReFinder system. Its main components include information access, information refind, context memory management, and a database of contextually accessed file paths and URLs.

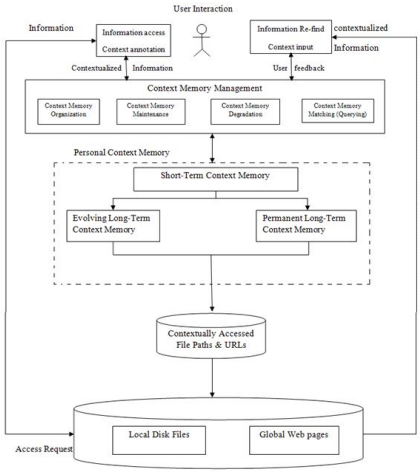


Fig. 4. The Overall Architecture Of Refinder.

- *Information access.* This component facilitates users to annotate their accessed interesting files/Web pages with the access context.
- *Information refind.* This component accepts users' context-based refinding requests, and returns the result files/Web pages.
- *Context memory management.* To process context based information refinding requests, the core context memory management component needs to do a bundle of work related to the organization, maintenance, degradation, reinforcement, and matching (i.e., querying) of the personal context memory.
- *Database of contextually accessed file paths and URLs.* Each context instance in the context memory links to the accessed files or Web pages, whose file paths and URLs as well as the titles are kept in the database of contextually accessed file paths and URLs

*Information Refinding Phase*

A user requests to refind previously accessed Web pages and files by indicating corresponding access context through the main interface of ReFinder (see Fig. 6). For example, the user can type in the following context instance:

[Time: 2012-02, Place: -, Activity: Do project]

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

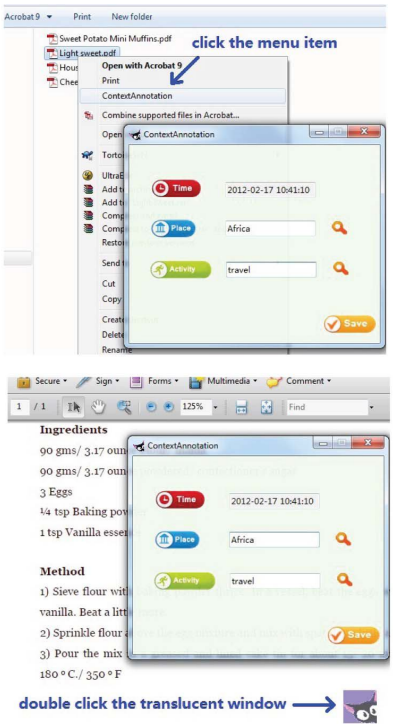


Fig. 5. Two Ways of Context Annotation for a File.

## V. RESULTS

The user study with ReFinder returns both quantitative results, as well as some valuable qualitative feedbacks. Among the total 104 refinding activities with ReFinder and 103 without ReFinder, Web pages re-visitation occupied 57.97 percent, and local files' reaccess 42.03 percent.

The participates utilized time, place, and activity as refinding context cues at the percentage of 56.73, 44.23, and 61.54 percent, respectively. On average, the participants spent the mean time of 15.53 seconds (13:67) in refinding with ReFinder, and 84.42 seconds (85:68) with other methods. The participants could re-localize the targets for 98.08 percent refinding tasks by ReFinder, and 91.26 percent targets could be re-found by other methods.

The reason why ReFinder failed in some tasks is due to the decay of context memory: certain participants' refinding requests cannot find matched context instances in the memory snapshot any more. Failures of other refinding channels (e.g., through desktop search or file explorer) lie in the participants' misremembering of specific file names or locations, and difficulty in browsing through a long list of historically visited Web pages.

## VI. DISCUSSIONS

ReFinder accepts users' refinding requests by their previous access context. It then finds matched context instances, linking to the recalled information, in a context memory snapshot. Underlying the refinding approach are context degradation and context annotation mechanisms. We discuss these fundamental issues and further possible improvement in the section.

*Why Degrading Context?*

Astute readers may argue the necessity of context degradation in the context memory snapshot, since keeping every detailed access contextual information can help the system do a better context matching job. In the study, we choose to degrade context for the following reasons. Compared to the enormous files and Web pages one has ever accessed or

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

created, user's refinding targets constitute only a small portion at a certain period. As time goes by, information accessed long time ago has a small chance to be recalled by the user. Degrading very old access context shrinks the search space so that the system can localize the most desirable candidates based on user's context input.

## VII. CONCLUSION

We have designed and implemented context-based information re-finding is known as Re-Finder to facilitate users in re-finding their previously accessed files and Web pages based on access context. ReFinder refinds information based on an objection by context model over the context memory snapshot and linking to the accessed information contents. Drawing on the characteristics of human brain memory in organizing episodic events, context occurrence in the memory snapshot are arranged in a clustered and associated aspect, and dynamically evolve by degradation and reinforcement in life cycles. We evaluate the performance of ReFinder system in two aspects. We are now working on the automatic context recognition and annotation to make ReFinder more user-friendly and practical.

## REFERENCES

[1] Tangjian Deng, Liang Zhao, Hao Wang, Qingwei Liu, and Ling Feng "ReFinder: A Context-Based Information Refinding System," IEEE Transactions On Knowledge And Data Engineering, Vol. 25, No. 9, September 2013.
[2] E. Adar, J. Teevan, and S.T. Dumais, "Large Scale Analysis of Web Revisitation Patterns," Proc. SIGCHI Conf. Human Factors in Computing Systems (CHI), 2008.
[3] J. Chen, H. Guo, W. Wu, and W. Wang, "iMecho: An Associative Memory Based Desktop Search System," Proc. 18th ACM Conf. Information and Knowledge Management (CIKM), 2009.
[4] Y. Chen and G. Jones, "Integrating Memory Context into Personal Information Re-Finding," Proc. Second Symp. Future Directions in Information Access, 2008.
[5] J. Hailpern, N. Jitkoff, A. Warr, R. Karahalios, K. Sesek, and N. Shkrob, "You Pivot: Improving Recall with Contextual Search," Proc. SIGCHI Conf. Human Factors in Computing Systems (CHI), 2011.
[6] S.K. Tyler and J. Teevan, "Large Scale Query Log Analysis of Re-Finding," Proc. Third ACM Int'l Conf. Web Search and Data Mining (WSDM), 2010.
[7] S. Won, J. Jin, and J. Hong, "Contextual Web History: Using Visual and Contextual Cues to Improve Web Browser History," Proc. SIGCHI Conf. Human Factors in Computing Systems (CHI), 2009.
[8] Liadh Kelly, Yi Chen"A Study of Remembered Context for Information Access from Personal Digital Archives," ACM 2008.
[9] Sarah K Tyler, Jian Wang, Yi Zhang "Utilizing Re-finding for Personalized Information Retrieval," ACM 2010.
[10] M.J. Kahana, M.W. Howard, and S.M. Polyn, "Associative Retrieval Processes in Episodic Memory," Learning and Memory: A Comprehensive Reference, pp. 1-24, Academic Press, 2008.