# REGAINER : A Context based Data Retrieval System

Priyanka G. Hakke
Department of CSE, Government College of Engineering, Amravati, India

Pushpanjali Chouragade
Professor, Department of CSE, Government College of Engineering, Amravati, India

## ABSTRACT
In searching, reading, writing and collecting lots of information from local computer or web. Retrieving information from the web is time consuming activity and most of the time the input queries from users are repeated.A context-based information refinding system called Regainer. It uses human's natural recall characteristics so that users can refind files and web pages according to the previous access context. Regainer refinds information based on query by context model over context memory snapshot. Context instances are clustered and required result is retrieved.

**Keywords**: Refinding information; context based; context degradation.

## 1. INTRODUCTION
People are experiencing data reading, writing and collecting lots of information from local computer or web. Retrieving information from the web is time consuming activity and most of the time the input queries from users are repeated i.e. Information refinding which is different from information finding. To support information refinding is to maintain access logs, recording what users revisit frequencies. 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.

## 2. HUMAN MEMORY ACTIVITY IN REFINDING PROCESS
Human memory activities are remembering and forgetting. In this activity context influences memory retrieval.

### 2.1Remembering and Forgetting
Remembering is the process of recollecting information from long-term storage, which can either be detailed information of the retrieving target or a judgment of whether the target exists in one's memory. These two types of outputs are referred to as remembered and known. The remember or know paradigm is widely used in studying the human memory, corresponding to recall and recognition tasks respectively.

According to psychology, Forgetting is caused by the inability to retrieve the item from memory rather than the lost or damage of storage. That is, the inability of tracing back to where that piece of memory is stored. Recalling related information usually results in better recall performance than with no provided cues nor subject to any specific order. Thus, we assume that a user can have better recollection of required information for searching (i.e.queries), if they are presented with relevant information as cues. As for free recall, most of the ideas that pop out are in fact either cued by the previous recalled piece of memory (thoughts) or triggered by external environment, which possesses certain elements that are associated with the piece of information in long term memory. Clustering is usually observed in free recall tasks. It can be considered as a more advanced version of chunking in retrieving, as it groups information according to some higher-level criterion. Forgetting has also been argued as an mechanism of filtering out unwanted information from memory, that means it is possible that people forget things because they do not think they want to remember them (i.e. unimportant, or cause too much pain remembering that) . Sometimes, it is also possible they might want to use this previously unwanted information, but are unable to retrieve it due to forgetting.

### 2.2 Context Factors that Influence Memory Retrieval
Retrieval of information and the popping out of associated (or clustered) ideas are triggered by the interaction of external information and internal context. In the physical world, context refers to the external physical environment, including temporally and specially surrounding information, which is assumed to be encoded together, associating with each other, and acting as cues at retrieval. These types of related information present at the time ofretrieval are called external context, while the internal contextis the activated information in human memory. When new information comes in, it broadcasts to the stored memory and the preexisting nodes which are active enough (above the 'to be perceived' threshold), these nodes react to the broadcast if they can be associated with the incoming information according the clustering rules. The threshold is determined by the effort or energy at the time of broadcasting. Those high priority links with best matched nodes pop out and construct the internal context which interacts with the input and leads to the searching target. It resides in short term storage waiting to associate or to reinforce association with the input information.

## 3. CONTEXT-BASED REFINDING
Context-based refinding differs from the traditional database query conceptually in three aspects. First, requestformulation is based on contextual attributes rather than database contents. Second, query target is context memorysnapshot rather than database. Third, an intermediate query result is a ranked list of context instances, with their linkedinformation as the final query result. At the implementation level, the query target (i.e., context memory snapshot) isorganized in a hierarchical, cluster and associated manner, and dynamically evolves in life cycles according to queryuser's memorization strength. For the final result generation through the context instances is quite straightforward [1].

## 3.1 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 queryrequest 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.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 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 wi€[0, 1] for every 1≤ j≤ n.[4]

Following equation used for calculating rank which is given in [1].

Rank

$$(Q,C)=\sqrt{\sum_{j=1}^{n} w_j . sim(A_j, q_j, a_j)^2} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

## 3.2 Association of Context Instances

For each contextual attribute Ai, and every value in its hierarchy, An association chain Chain (Ai, v), whichconsists of all the context instances with the same attribute value of Ai. For any context instance C 2 Chain (Ai,v), (ci = v). Fig. 1 illustrates six 3D context instances in the memory snapshot. A few chains are illustrated in the Fig. 1. 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 gainEChain (Ai, v), so that for any context instance C 2 EChain (Ai, v), (ci = v) or (cii<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 Fig. 2, which is from paper [4].

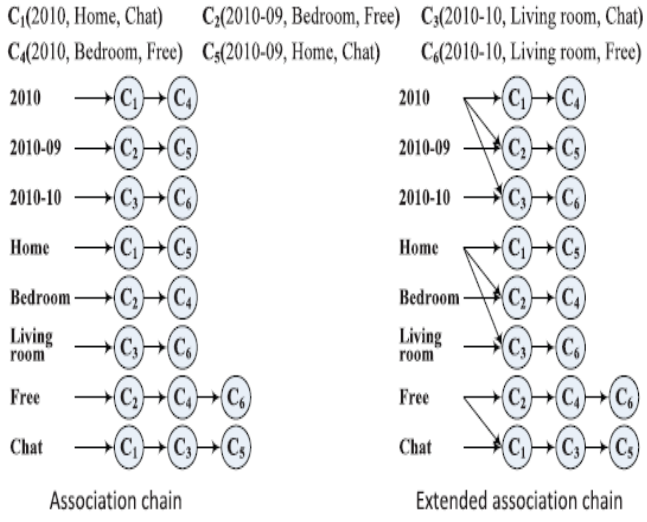


**Figure 1: Context Association Relationships [1]**

### 3.2.1Cluster-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 thecontext instances by the greatest extent.The context instances are clustered on the timeattribute. 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 theactivity attribute (Asso-dim = 3).[1]

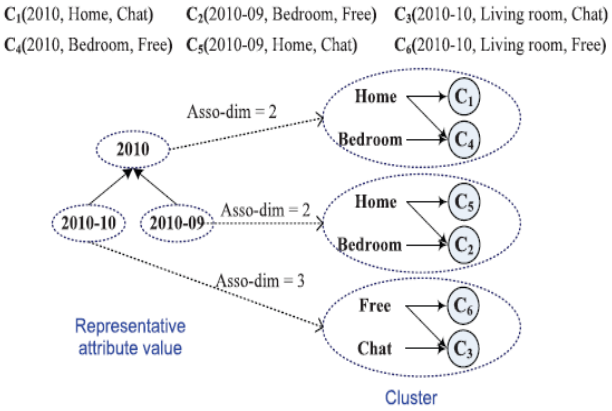


**Figure 2: Context Cluster-Association Relationships [4]**

## 4. REVIEW OFINFORMATION RE-GAIN

Regainer system accepts user's context-based re-finding requests, and returns the result files/web pages. Re-gainer accepts users' re-finding requests by their previous access context. It then finds matched context instances, linking tothe recalled information, in a context Memory snapshot. Fig. 3 illustrates the overall architecture of Regainer system.Its main components include information access, information refind, context memory management, and a database ofcontextually accessed file paths and URLs.

- 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.

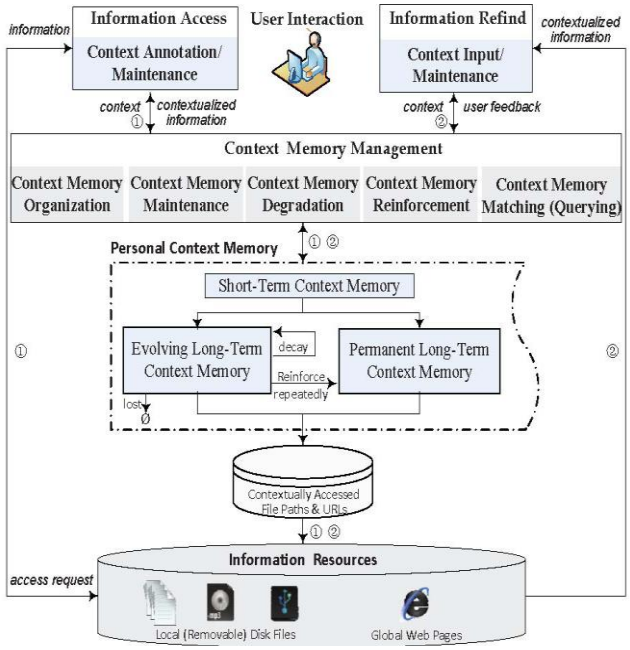


**Figure 3:Overall Architecture of Information Regainer [1]**

- 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

## 5. CONCLUSION

In this paper, we present information on a context-based datarefinding system called Regainer, which facilitate users in refinding their previously accessed files and Web pages based on access context. Regainerrefinds information based on a query-by-context model over a context memory snapshot, linking to the accessed information contents. Using the characteristics of human brain memory in organizing episodic events, context instances in the memory snapshot are organized in a clustered and associated way, and dynamically evolve by degradation and reinforcement in life cycles. On average, 15.53 seconds are needed to complete a refinding request with Regainer and 84.42 seconds with other existing methods.

## 6. 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] Yi Chen Gareth J F, Jones, "Integrating Memory Context into Personal Information Refinding", the 2nd BCS-IRSG Symposium on Future Directions in Information Access.

[3] Yeswanth.L, Nimala.K, "Activity Based Key Search Information Re-finder",Yeswanth.L et al,/ (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (1) , 2014, 648-650.

[4] Prakash.M, Karthika.D, Sophia.J, "A Large Scale Analysis Of InformationReFindingSystem", International Journal of Innovative Research in Computer and Commu- n icationEngineering (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).

[5] Goula. Vijay Kumar, A. Krishna Chaitanya, "User Priority Based Search on Organizing User Search Histories with Security", IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 12, Issue 6 (Jul. - Aug. 2013), PP 49-53.

[6] A.Revathi, A.Mekala, "User Search Histories Based On Query Relevance Using Incremental Algorithm", International Journal of Innovative Research in Computer and Communication Engineering (An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014.

[7] A.P. Nivethith, D. Kerana Hanirex, Dr.K.P. Kaliyamurthie, "A Comparative Study of Context-Based Information Refinding", COMPUSOFT, An international journal of advanced computer technology, 3 (4), April-2014 (Volume-III, Issue-IV).

[8] Yeswanth.L, Nimala.K, "Activity Based Key Search Information Re-finder" (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (1) , 2014, 648-650.

## 7. AUTHOR BIOGRAPHY

**Priyanka Gulabrao Hakke** has received her B.Tech. degree in Computer Science and Engineering from Shri Guru Gobind Singhji Institute of Engineering and Technology, Nanded, India in 2013 and persuing M.Tech. in Computer Science and Engineering from Government College of Engineering, Amravati, India. Her research interest includes Data Mining, Web Mining, Image Processing.

**Pushpanjali M. Chouragade** has received her Diploma in Computer Science and Engineering from Government Polytechnic, Amravati, India, in 2007, the B.Tech. degree in Computer Science and Engineering from Government College of Engineering, Amravati, India in 2010 and her M.Tech. in Computer Science and Engineering from Government College of Engineering, Amravati, India, in 2013. She was a Lecturer with Department of Computer Science & Engineering, in Government College of Engineering, Amravati, in 2010-11. Her research interest includes Data Mining, Web Mining, Image Processing. At present, she is an Assistant professor with department of Computer Science and Engineering at Government College of Engineering, Amravati, India, since 2011.