International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 05 Issue: 04 | Apr-2018 www.irjet.net p-ISSN: 2395-0072

Multi-user Recommendation System for Improving Web Revisitaion Based on Access Context and Content Keywords

Uma .M[1], Vipin K.M[2]

[1]*M. Tech CSE Student, NCERC Pampady, Kerala, India*
[2]*Asst. Prof. CSE, NCERC Pampady, Kerala, India*

Abstract - *As the increasing amount of web searches nowadays, the need for information retrieval has become important. Web Revisitation is a technique involved in it, for visiting a webpage that is previously visited by particular user. Various efforts have been carried out to improve web revisitaion technique. Here in this paper, a novel method is proposed to help web revisitaion by combining techniques for web recommendation, and both web page access context and content keywords. Multiple users can register and browse web pages, those can be saved and retrieved separately for revisiting and also can be revisit based on recommendations from other users.*

Key Words: **revisit, access context, page content, refind, recommendation system**

1. INTRODUCTION

Nowadays, for information delivery web is playing a significant role for users. Among web searches, re-finding a previously viewed web page, that is the process of visiting a web page which has been already visited before has become one of the most important topic. This process is called as web revisit. The process of retrieving particular information that to satisfy user need has become difficult due to the wide range of information storage in web. So that made the web revisitation task difficult.

For both getting updated information and also to re-visiting information those they have seen before, most users revisit webpages on the Internet more often. It is founded that users revisit nearly half of all webpages they seen before. But, such "re-finding" becomes one of difficult task using today's web browsers. So many users have reported the issue for not being able to return to a page that they once visited, as one of the serious issues using the Web.

There are different previous studies held according to web revisitaton. In those studies, various demonstrations had been involved with use of page access context keywords and page content keywords to improve web revisitation. Among these, the technique that combines usage of both page access context and content keywords has become predominant.

2. RELATED WORK

Different types of methods and tools are developed, to support personal web revisitation, including bookmarks, history tools, search engines, metadata annotation and exploitation, and contextual recall systems. Contextual recall systems have become more user preferred technique. A survey based on these techniques can give a clear idea for improving web revisitation.

2.1 Stuff I've Seen: A System for Personal Information Retrieval and Re-Use

As there are different types of information that includes file system hierarchy for files, E-mail folder hierarchy for E-mail, favorites or history for web pages etc., a system that unifying these different types of information has needed. *Stuff I've Seen (SIS)* is such a system that facilitate information re-use by two key aspects developed by Susan Dumais, Edward Cutrell *et al*. First, whether it was seen as email, web page, document, media file, calendar appointment, etc. the system provides a unified index of information for all. Second one is, for supporting web revisitation rich contextual cues can be used in the search interface, because the user has seen the information before. Here contextual keywords for search interface include time of web search, author, thumbnails, previews etc. These filters are specified in interface are coupled with the fast client-side processing to support iterative refinement strategy. Even standard ranking functions are less important in the context of personal information, date and people names provide rich contextual keyword support for web page retrieval.

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

Other than a system that help users with providing basic information such as the date of the last visit and title of the page visited, Chinmay kulkarni *et al*. developed a system *Memento* that to provide users with descriptive topic-phrases to help re-finding. Here it retrieves topic-phrases for a webpage based on both page content and the page context in which the page visited before. Memento preserves user privacy and is able to index pages which are not publicly accessible because the system is completely client-side. Here the page's access context constitutes the browsed pages which followed and preceded the accessed web page. Identifying these page-contexts has an important role in extracting topic-phrases for a target page. Session boundaries are calculated to identify page contexts. Then content of each web page is parsed to obtain a term-frequency vector. If two pages have non-null term-frequency vectors and their cosine similarity is below a threshold value then that pair of consecutive pages is considered a session boundary. Then we can mine topic-phrases using a pool of pages identified as its context and content from the page.

2.3 YouPivot: Improving Recall with Contextual Search

Human memory can predicate more on contextual cues according to cognitive science literature. Since computer systems do not depend the natural process of using contextual cues for recalling, Joshua Hailpern *et al.* proposed a new interaction technique, Pivoting, to search for contextually related activities often not semantically related, to find target web information. Also a new personal annotation method, called TimeMarks, is presented to support more contextual recall and the pivoting process. This YouPivot system allow users to search for contextual activities through digital history based on the context, users do remember and find a target. The main part of this system is to allow users to Pivot, and changes their point of view to that of a file, website, or other activity. That is users can search using everything else those were active during that period of time. That is, this system allow user to search in terms of the context of a file rather than just a meta-data title, or keyword piece of information.

2.4 ReFinder: A Context-Based Information Refinding System

A context-based information refinding system called ReFinder is proposed by Tangjian Deng and Liang Zhao, It depends human's natural recall characteristics and also allow users to refind Web pages according to the previous access context. A query-by-context model over a context memory snapshot that linking to the accessed information contents is used to refind information contents. Clustered and associated manner organization of context instances in the memory snapshot is used. Then the matched context instances are found and linking to the recalled information, in a context memory snapshot. There are two approaches for refinding, those are Context Degradation and Context Annotation mechanisms. In Context Degradation, very old and useless contexts can be removed from the search space so system can easily find desirable contexts. In Context Annotaion, users need to manually annotate access context like place and activity, for retrieving their encountered interesting files and Web pages.

2.5 Enhancing Web Revisitation by Contextual Keywords

Since users have to manually annotate access context, it make users very difficult to achieve retrieval for web revisit. So Tangjian Deng, Liang Zhao, and Ling Feng developed a system to enhance web revisitation by contextual keywords. Taking advantages of access context like time, location, concurrent activity, context-based search are more preferred by users. That is the context under which information is accessed tends to be more easily to remember than content. To improve users' memory recall, this system presented a way to automatically capture user's access context from user's concurrent activities. Access context is managed by a probabilistic context tree for each accessed web page stored with corresponding URL. The tree contains of contextual keywords, stored automatically from user's running computer programs. The context memory generated as the elapsing time and adjusts in accordance with the user's revisit feedbacks.

3. WEB REVISITATION SYSTEM

The system consists of 2 main parts; one is web page access that is to prepare for web revisit and another one is web revisitation itself. Both parts include different acquisition and management strategy.

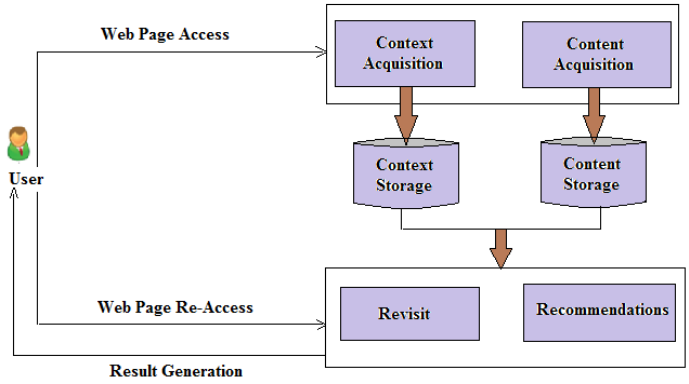

Fig-1: System Architecture

3.1 Web Page Access For Prepare Revisit

Users can browse web pages according to their needs and can select important portions from those pages for later revisit. Since both previous access context and content-related keywords are used to retrieve previously accessed web pages, those context and content related keywords have to be stored and managed.

Context Storage and management: Access context of web page access includes time, location and activities. Users can browse web pages and also can revisit them according to user needs based on these context keywords. While a registered user login into the system, the access time and access location will be stored automatically. The user can give context keyword activity as input for web browsing. Access activity is used to represent category of browsing that is study related, entertainment related etc. There is a table maintained for storing activities in the database and the access time maintenance includes storage of the current month of web access. Access location is stored automatically by fetching current location of web access. By use of geo location, the latitude and longitude of location can be mapped and stored in a table of database.

Content Storage and management: Page content keywords can be used to retrieve previously viewed web pages. While web page access, user can highlight important portions of web page content. Depends on the highlighted portions of web access by different users, corresponding text data, title and URL will be retrieved and stored into database automatically. Since the selected portions of webpage may contain many more sentences with different words, the data have to be shortened based on its importance. Data are

preprocessed and graph based keyword extraction algorithm is used to retrieve important keywords from whole preprocessed text and those will be stored. Text preprocessing and keyword extraction includes following steps:

1) Tokenization of text document
2) Normalization includes case normalization
3) Punctuation removal
4) Stop words removal
5) Porter-stemmer Algorithm
6) Graph based keyword extraction algorithm

3.1.1 Tokenization of Document

Tokenization is used to divide longer strings of text into smaller segments, or tokens. This is done by tokenizing larger collection of text into sentences, then sentences can be tokenized into words, etc. After tokenizing a piece of text appropriately, further processing is generally performed. So the process tokenization is also referred to as text segmentation or lexical analysis. Generally breakdown of a large chunk of text into paragraphs, sentences or pieces larger than words is referred as segmentation, while the breakdown process which results words is called as tokenization.

3.1.2 Normalization of tokens

Text normalization has to done before further processing. Generally normalization is a series of related process to put all text on same level field; that is converting all text into same case format (upper or lower), removing punctuation, converting numbers to their word equivalents, stop words removal, and so on. So normalization is used to put all words on equal format, and to allow processing for proceed uniformly.

Here First of all, tokens are converted into lowercase for uniform storage purpose. Then punctuations are removed from those tokens. Punctuation removal is generally a part of tokenization but still need to be considered on this stage. Then after punctuation removal, stop words have to be removed. Stop words are those most common words in a language which have to be filtered out before further processing of text, since these words contribute little to overall meaning of a passage. For instance, "the," "and," and "a," even though, all required words in a particular passage, but don't generally contribute greatly to reader's understanding of passage.

3.1.3 Porter Stemmer Algorithm

The Porter stemming algorithm (or 'Porter stemmer') is a process used to remove the commoner morphological and inflexional endings from words. Main use of this algorithm is as part of a term normalization process that is usually done when setting up Information Retrieval systems. Over the past 20 years, the Algorithm has been widely used. Unfortunately many variants of this claim to be true implementations, and this can cause confusion. The original stemmer was coded in BCPL, a language no longer in trend. All the languages ANSI C, Java and Perl versions which are exactly equivalent to the original BCPL version are, tested on a large corpus of English text. By automatic means, removing suffixes is an operation which is especially useful in the field of information retrieval. Each document described by the words in the document title and possibly by words in the document, abstract in a typical IR environment. Since we can say that a document is represented by a vector of words, or terms, ignores the issue of precisely where the words originate. Those terms with a common stem will usually have similar meanings, for example:

CONNECT

CONNECTED

CONNECTING

CONNECTION

CONNECTIONS

3.1.4 Graph-Based Keyword Extraction Algorithm

Most dominating words ("keywords") are retrieved in order to generate a summary, after all preprocessing tasks. Since in a text each different word can be represented by a node in the document graph, the keywords extraction algorithm is used to reduce the prior nodes extraction in graphs. From very simple co-occurrence relation of syntactic ones like words connected by edges to more complex ones like concepts connected by semantic relations are represented by this graph structure. Its language independency is the main advantage of a syntactic representation, while the semantic graphs representation provides new characteristics of text such as its captured semantic structure that itself can serve as a document surrogate and provide means for document navigation. Here performs experiments with directed graphs, where the nodes stand for words/phrases and the edges represent syntactic relationships between them.

To extract the summary keywords, apply a ranking algorithm called HITS to directed graphs representing source documents. Since here works with directed graphs, HITS is the most appropriate algorithm for this task as it takes into accounts both in-degree and out-degree of nodes. Also by running HITS till convergence is not necessary and initial weights that get after the first iteration of algorithm are good enough for rank-based extraction of summary keywords. Currently, use the "simple" graph representation defined in that holds unlabeled edges representing order-relationship between the words represented by nodes. The stemming and stop-word removal operations of basic text preprocessing are done before graph building. Only a single vertex for each distinct word is created even if it appears more than once in the text. Thus each vertex label in the graph is unique. If a word 'a' immediately precedes a word

'b' in the same sentence somewhere in the document, then there is a directed edge from the vertex corresponding to term 'a' to the vertex corresponding to term 'b'. Sentence terminating punctuation marks (periods, question marks, and exclamation points) are taken by us into account and an edge is not created when these are present between two words.

3.1.5 HITS Algorithm

A page that is linked to by many important pages (with high rank) receives a high rank itself. A similar idea can be applied to lexical or semantic graphs extracted from text documents, in order to extract the most significant blocks (words, phrases, sentences, etc.) for the summary.

In this paper, HITS (Hyperlink-Induced Topic Search; also known as hubs and authorities) algorithm is applied to document graphs and evaluate its performance on automatic unsupervised text unit extraction in the context of the text summarization task. The HITS algorithm distinguishes between "authorities" (pages with a large number of incoming links) and "hubs" (pages with a large number of outgoing links). HITS is a link analysis algorithm that rates Web pages, developed by Jon Kleinberg. The idea behind Hubs and Authorities stemmed from a particular insight into the creation of web pages when the Internet was originally forming; that is, certain web pages, known as hubs, served as large directories that were not actually authoritative in the information that they held, but were used as compilations of a broad catalog of information that led users direct to other authoritative pages. In other words, a good hub represented a page that pointed to many other pages, and a good authority represented a page that was linked by many different hubs. For each node, HITS produces two sets of scores- an "authority" score, and a "hub" score.

The algorithm performs a series of iterations, each consisting of two basic steps:

- **Authority Update:** Update each node's Authority score to be equal to the sum of the Hub Scores of each node that points to it. That is, a node is given a high authority score by being linked from pages that are recognized as Hubs for information.

- **Hub Update:** Update each node's Hub Score to be equal to the sum of the Authority Scores of each node that it points to. That is, a node is given a high hub score by linking to nodes that are considered to be authorities on the subject.

The Hub score and Authority score for a node is calculated with the following algorithm:

1. Start with each node having a hub score and authority score of 1.
2. Run the Authority Update Rule
3. Run the Hub Update Rule
4. Normalize the values by dividing each Hub score by square root of the sum of the squares of all Hub scores, and dividing each Authority score by square root of the sum of the squares of all Authority scores.
5. Repeat from the second step as necessary.

3.2 Web Revisit

After web page access users can revisit them at any time according to their need. Highlighted portion of web content with its title and URL will be displayed as result for users. For aiding web revisit, users can use context keywords and/or content keywords. There are two types of web revisit possible with this system.

I. One is with use of context and/or content keywords of web access by particular user

II. Another one is recommended web pages by other users of the system

By giving context and content keywords as input to aid web revisit, web pages with corresponding keywords will be retrieved. User can give context keywords that he remembers about corresponding web page access or content keywords or both. So that user can retrieve web pages according to their own previous access history. The

3.2.1 Recommendation System

Also user can visit web pages those are previously visited by other user of the system. Currently, recommendation systems are used in a daily basis in several domains such as hotel and flight booking, learning, and purchasing or renting a property. The main idea of the system is to recommend to users web pages relevant to specific topics in a digital library. The recommendations here go through two main phases: collection of items (i.e. dataset creation) and selection of items from the created dataset.

With inputting a content term related to user need, corresponding web pages which are previously visited by other users will be resulted as output of this recommendation system. For this purpose, the stored content term list will be checked and web pages will be displayed based on below two attributes:

1. According to the score of terms contained in each related web page

2. Based on visitCount attribute

The total score calculation is based on authority score and hub score of HITS algorithm. The calculations of both scores are given below:

$$auth(V_i) = \sum_{V_j=1}^{n} hub(V_j) \quad and \quad hub(V_j) = \sum_{V_i=1}^{n} auth(V_i)$$

For the total rank (R) calculation we used the following four functions:

1. Rank equals to authority score:

$$R(V_i) = auth(V_i)$$

2. Rank equals to hub score:

$$R(V_j) = hub(V_i)$$

3. Rank equals to the average between two scores:

$$R(V_i) = avg\{auth(V_i), hub(V_i)\}$$

4. Rank equals to maximum between two scores:

$$R(V_i) = \max\{auth(V_i), hub(V_i)\}$$

4. RESULTS AND USER STUDY

The system uses NetBeans IDE 8.2 tool for its implementation, which is an integrated development environment (IDE) for JAVA.

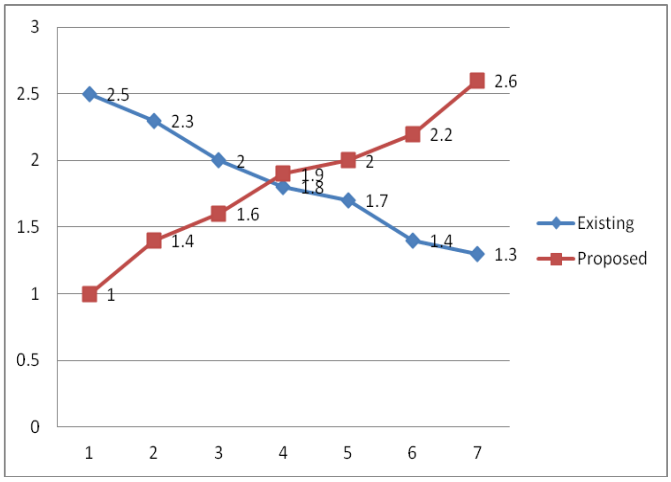

Chart-1: Comparison between exisiting and proposed systems

NetBeans develop applications from a set of modular software components called modules. It can run on different operating systems including Microsoft Windows, macOS, Linux, Solaris. Also runs extensions of different languages C, C++, PHP, Java, HTML etc.

After implementation of this novel system with recommendation the favorability towards the system got increased over time. By testing different user queries, it is noted that users have the interest of selecting different web sites using this novel method. They revisit those web pages at some point of time and also they prefer recommendations from other users both with context and content keywords. This is shown in Chart-1 above.

Table-1: Performance comparison in Revisit query

Keywords	Average Precision	Average Recall	Average F1-measure
Content	0.1322	0.8283	0.2280
Context	0.2501	0.8621	0.3877
Content+Context	0.3011	0.9121	0.4527

For revisit purpose users mainly choose access context keywords as they remember than content keywords. Likewise users also refer recommendations from others to satisfy their needs. For recommendations, users prefer content keywords than context keywords. The performance comparison of system is done based on these access context and content keywords.

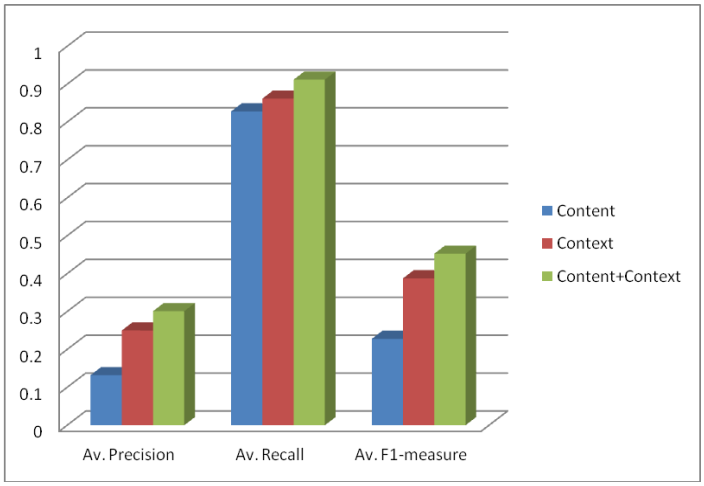

Chart-2: Performance comparison in Revisit

For refind certain web pages, users prefer to choose context keywords that includes access time (in the sense month), location and activity which them remember. It is shown in Table-1. The Precision, Recall and also F1-measure for refind with context keywords remain higher.

Table-2: Performance comparison in Recommend query

Keywords	Average Precision	Average Recall	Average F1-measure
Content	0.2803	0.8872	0.4261
Context	0.1822	0.8231	0.2983
Content+Context	0.3211	0.9411	0.4788

The performance for recommendation queries are showed in Table-2. There is more possibility to search web pages by content keywords according to recommendations from other users, since context keywords for other user's search are often unknown.

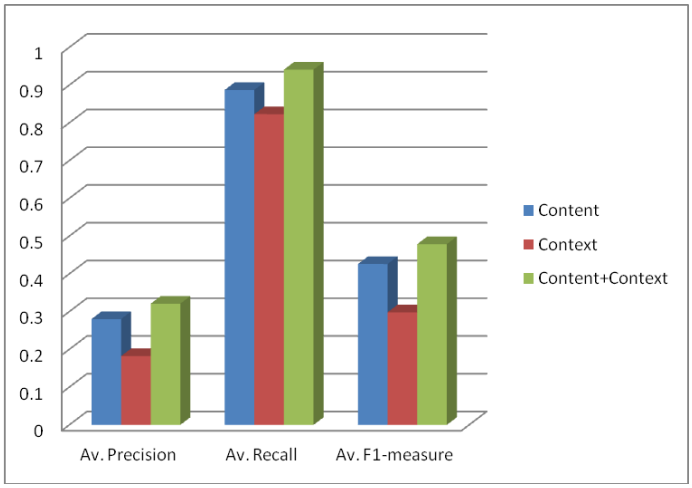

Chart-3: Performance comparison in Recommend

5. CONCLUSION

A novel method of web revisit is proposed in this paper. Web page revisit has become one of most popular technique for web users. That methodology facilitates visiting a web page that is previously visited by the user. There are different studies held according to web page revisit by using content and context keywords of web access as inputs and also by recommendations. Here in this paper it combines both techniques of revisit; access context and/or content keywords and recommendation. Access context includes access time, location and activity of web access. Content keyword includes important terms of highlighted web content. Users can give these access contexts or content keywords or both as input for retrieving web pages those are previously visited by that particular user. Also the recommendations from other users can help a user greatly to find out particular web page according to their need.

REFERENCES

[1] S. Dumais, E. Cutrell, J. Cadiz, G. Jancke, R. Sarin, and D. C. Robbins. "Stuff i've seen: a system for personal information retrieval and re-use". In SIGIR, 2003

[2] C. E. Kulkarni, S. Raju, and R. Udupa. "Memento: unifying content and context to aid webpage re-visitation". In UIST, pages 435–436, 2010

[3] J. Hailpern, N. Jitkoff, A. Warr, K. Karahalios, R. Sesek, and N. Shkrob. "Youpivot: improving recall with contextual search." In CHI, pages 1521–1530, 2011

[4] T. Deng, L. Zhao, H. Wang, Q. Liu, and L. Feng. "Refinder: a context-based information re-finding system." IEEE TKDE, 25(9):2119–2132, 2013

[5] T. Deng, L. Zhao, and L. Feng. "Enhancing web revisitation by contextual keywords." In ICWE, pages 323–337, 2013

[6] M. A. Soliman, I. F. Ilyas, D. Martinenghi, and M. Tagliasacchi. "Ranking with uncertain scoring functions: semantics and sensitivity measures." In SIGMOD, pages 805–816, 2011.

[7] Leskovec, J., Grobelnik, M. and Milic-Frayling, N. 2004. "Learning Semantic Graph Mapping for Document Summarization." In Proceedings of ECML/PKDD-2004 W

[8] Mihalcea R. 2004. "Graph-based ranking algorithms for sentence extraction, applied to text summarization." In Proceedings of the 42nd Annual Meeting of the Association for Computational Lingusitics, Barcelona, Spain.

[9] Alon, N., Awerbuch, B., Azar, Y. and Patt-Shamir, B. "Tell Me Who I Am: An Interactive Recommendation System." Theory of Computing Systems, 45, 2 2009, 261-279.

[10] I. Ruthven and M. Lalmas. "A survey on the use of relevance feedback for information access systems." Knowledge Engineering Review, 18(2):95–145, 2003.