Adaptive Query Recommendation Techniques for Log Files Mining to Analysis User's Session Pattern

Durga Choudhary
M.Tech Scholar
Department of CSE
RCEW, Jaipur

Subhash Chandra Jat
Assistant Professor
Department of Computer Application,
RCEW, Jaipur

Pankaj Kumar Sharma
Assistant Professor
Department of CSE
RCEW, Jaipur

ABSTRACT
System log files are very important part of any web application. System log files serves as the purpose of directory in various aspect of knowledge mining. There is a wide variety of logs to stock knowledge about the search patterns of the users. There might be lots of formats of availability of logs, each of web application can develop format of its own logs. Generally, IP, date and time of the request, result for the request (with code), transaction size, protocol, request description, browser and operating system used by the user are some of the important attributes of every request that get into the record of the log file. This paper presents the user's behavioral search pattern by the query log files.

General Terms
Log Files, User Session Pattern, Web Mining, Query Recommendation Techniques.

Keywords
Log Files, Web Mining, Query Recommendation Techniques.

1. INTRODUCTION
Each single access of the web page through the web queries given in the search engine enriches the query log files about the access information of every individual user. The automatic storage of the access information in the query log files of the search engines help to cull out relevant and appropriate information from the web. In every search through the search engine, the user submits a query that leads to the consolidation of the navigational information in the query log file. By this log, the queries can be personalized, based on the user's behavioral search pattern. Indeed, this study aims an effective comprehension of the search engine users‟ temperament of search, their interest through the implicit profile of use that may help to retrieve quick and appropriate information in every search.

Web Mining is technique in data mining to extract knowledge from web data, including web documents, hyperlinks between documents, usage logs of web sites, etc. In Web Mining, data can be collected at the server side, client-side, proxy servers, or obtained from an organization's database (which contains business data or consolidated Web data). There are many kinds of data that can be used in Web Mining. According to data analysis objective, web mining can be divided into three different types, which are web usage mining, web content mining and web structure mining. Web content mining describes the automatic explore of information resources available online, and involves mining of web data content. Web structure mining is the process of analyzing hyperlink and tree-like structure of a web site using graph theory. Web usage mining is the process of extracting effective information from web server logs. Users show different interests when looking for internet. Some users might be looking at only documentary data, whereas some others might be engaged in multimedia data. Web usage mining (WUM) involves the automatic detection of user access patterns from one or more web servers. Organizations rely on internet for their business work which often generate and collect bulk size data in their daily practices. Most of this information is generated automatically by web servers and collected in server access logs. The companies can establish better customer manager relationship by giving them exactly what they require. Companies can understand the requirements and serve them accordingly. They can also increase profitability and productivity based on the profiles generated [11].

2. LITERATURE REVIEW
WWW has become the source for most of the existing information. Search engines have become the gateway of retrieving the requisite information from the web in terms of web snippets with reference to the query logged in by the user. A query is an array of keywords employed to secure information from the web resources even though desirable results are not retrieved at all times. During certain times inappropriate and redundant results are retrieved from the web.

2.1 Review of Search Log and Search User Behavior analysis
The commonly used technique to augment the users ‟search experience is the utilization of the knowledge contained within past queries in the query log. A query log, typically, contains information about users, issued queries, clicked results, etc. From this information, knowledge can be extracted to improve the quality in terms of effectiveness and efficiency of their system. For instance, the search histories in any search engine are organized under the following attributes.

< AnonID, Query, QueryTime, ClickURL>

The query made in the searching process normally belongs to one of the following categories namely - Informational query, Navigational query and Transactional query [13]. When the user issues the query and views the content in the web snippet, then such a query is categorized as Informational query. Navigational query refers to the user issuing the query and clicking the sequence of URLs in a session and getting the information where the user intent is to see the web sites (e.g. "Inheritance in Java") and the recommended queries come into play here. Online tasks such as railway ticket reservation are performed through Transactional query. Tests conducted take account of the investigation of the query sessions for each user and of the connection among the terms of the queries. Their results show that above 85% of the users visit the first page of results only and 77% of the users session end up just after the first query.

2.2 Review of Query Recommendation Techniques

The properties of the competent query recommendation system mentioned in [2] are

Relevancy: Recommended queries should be semantically relevant to the user search query.

Redundancy Free: The recommendation must refrain from redundant queries that repeat analogous search intents. ***Diversity:*** The recommendation should encompass search intents of different interpretations of the keywords given in the input query.

Ranking: Highly relevant queries should be ranked first ahead of less relevant ones in the recommendation list.

Efficiency: Query recommendation provides online helps. Therefore, recommendation algorithms should achieve fast response times.

Some of the criteria considered in query recommendation approach [3] are

Recommendation type: The recommendation system may either be content-based or collaborative. Within the content-based context, the user is considered individually with reference to his/her requirements. In the social or collaborative context, it further enfolds the context of other users who may represent comparable preoccupations. Recommendation input data: Recommendation source which can be a user profile, a query history (log file) or an external source (ontologies, web pages etc.).

Recommendation Time: Time of recommendation: before querying, while querying or after querying.

Recommendation features: Recommendation object which can be a set of entire queries or a set of query fragments.

Research field: Domain of application of the recommendation approach. It can be database field (DB) or data warehouse field (DW).

2.3 Query Recommendations through Graphs

Clustering of similar queries by means of distance is influenced by (i) query keywords (ii) string matching of keywords (iii) common clicked URLs and (iv) Distance of the clicked documents. Chains of queries stored in query logs form the basis to discover related queries [50] based on association rules. The query log is observed to be a set of transactions, where each transaction denotes a session in which a single user submits a sequence of related queries within a time frame. The basic idea here is to reformulate the query such that it gets closer to the term-weight vector of the documents the user is looking for. The patterns are inserted in a query relation graph which permits the identification and suggestion of "concepts".

Baeza Yates et al. [13] identified and clustered semantically similar queries. The content of historical preferences of users registered in the query log is used by the clustering process which apart from identifying the related queries and ranks them according to a relevance criterion. The similarity between the two queries is calculated by weighing each term according to the number of occurrences and the number of clicks of the documents in which the term appears. Given a query q, and a URL u, let POP(q, u) be the popularity of u (fraction of clicks) in the answers of q. Let TF(t, u) be the number of occurrences of term t in URL u. Vector representation for q, where q[i] is the i-th component of the vector associated to the i-th term of the vocabulary (all different words), as follows. The similar queries are clustered by successive calls to a k-means algorithm, using the CLUTO software package.

In order to facilitate a user to redefine a query, Dupret et al. [14] suggested a method by recommending a list of similar queries which are traced from click-through data. Recognition of better queries is valuable for query disambiguation and query specialization. A query qa is a recommendation for a query qb if a significant number of sessions of qa are consistent with qb and are ranked better by qa than by qb. The recommendation algorithm induces a directed graph between queries. While the original query is akin to the root of a tree, the recommendations are analogous to leaves. Each branch of the tree symbolizes a different specialization or sub-topic of the original query. In order to improve the search engine ranking of documents, alternate queries are ascertained. The documents selected during past sessions of a query are ordered in accordance with the ranking of other past queries [15]. Associated query is recommended if the resulting ranking is better than the original position.

Baeza Yates et al. [13] ascertained that the queries are characterized as query-click bipartite graph followed by cover graph (CG). Semantic relations perceives multi topical URLs. A number of features of these graphs are analyzed which relates to information both on how users query, how the user reacts after the query and the content distribution of what they observe. The presence of a path between two queries connects them in the bipartite graph. Different edge colors on the basis of URLs that connect two queries are studied. Each query in a node of the graph is also identified. Two nodes that include the queries are connected by an edge if they share at least one URL u. Edges are weighted according to the cosine similarity of the queries they connect. Hence, if e = {q, q‟} and the weight of e is given by, Baeza-Yates defined 5 types of query graphs [16]: word, session, URL cover, URL link and URL terms. Three types of information form the basis to construct query graphs: the query terms, the searching and clicking behaviors during the query session and the content of clicked URLs. Query terms, search result snippets and other simple text information.

[18] Suggested a two-level query suggestion model by mining click through data, in the form of user-query and query-URL bipartite graphs. Joint matrix factorization method employs two bipartite graphs to study the low rank query latent feature space followed by the construction of a query similarity graph based on the features. This is followed by designing an online ranking algorithm to disseminate similarities on the query similarity graph and finally suggest the users about semantically relevant queries.

QUBIC [17] is Query-URL bipartite based query recommendation approach has two distinctive characteristics. First, it extracts an affinity graph of queries from the initial query-URL bipartite graph instead of operating on the original bipartite graph. Only queries are included in the affinity graph as its vertices and its edges are weighted according to a query-URL vector based similarity (distance) measure. With the query affinity graph, by inducing an implicit topical relatedness between queries, the propagation of similarity from query to query is captured.

A query-flow graph [16] or probabilistic reformulation graph [14] relates to user querying behavior where the sequence of

queries submitted by a user can be seen as a path on this graph. Assignment of score values to queries permits to perform a random walk on the query-flow graph. Furthermore, offering recommendations is observed as adding shortcuts in the query-flow graph that "nudge" the reformulation paths of users, such that the users are more likely to follow paths with larger expected utility. NP-hard and Greedy algorithm are used to optimize the recommended queries.

Query logs and social annotation data are combined which capture two types of interests form the basis for the construction of Query relation graph. Query suggestions are provided by combining hitting time analysis on the query-URL bipartite graph and Markov random walk. The relationships between queries q and web pages l are made use for the construction of bipartite graphs containing two types of vertices. Portrays the adaptation of bipartite graph, every undirected edge in the original bipartite graph is transformed into two directed edges. The weight on a directed query-URL edge is normalized by the number of times the query is issued. On the contrary, the weight on a directed URL-query edge is normalized by the number of times the URL is clicked.

The query-flow graph [16] witnesses the concepts of template rules. At a high level, the concept is general and further used to enhance other query-log constructs.

2.4 Review of Re-Ranking Of Recommended Queries

The recommended queries are well-organized based on the significance allocated to the queries. The first ranking functions derived from the vectorial model of IR [10] were based on document terms. But, such techniques posed several problems that influenced the precision of the answer lists, among them the following:

Synonymy: Many documents or queries related to a topic might not share the same words

Polysemic queries: Queries with the same descriptive text may have different meanings in the same or in different languages.

Spamdexing: It is regular for documents to include artificial repetition of words, which permit them to appear higher in the answer lists.

Quality: The text itself does not always reveal the quality of the document [7].

Filtering (ACF) techniques to assemble personalised query-less job recommendations. The recommendable items are ranked based only on their frequency in the community. Ranked list approach is one type of the recommendation systems, the user indicates requisite with one or more keywords and the system usually presents a long list of results, ordered by their predicted relevance to the user. On the other hand, the inquirer has series of conversations with the advisor to get the different choices and recommendations in conversational system.

The recommended URLs and queries that combine the similarity of the query to the input query and the support of the query in the cluster are ranked by a criterion offered by [46]. The support assesses the extent to which the suggested query has attracted the attention of users. The rank measures, the interest of the query to the user that submitted the input query. It is important to include the extent of support for the recommended queries. Baeza Yates et al. [10] trace and cluster semantically similar queries. This method, apart from tracing the related queries, ranks them according to a relevance criterion. On the basis of similarity and support of the query, rank score is computed. Support is the unit to measure the query as the fraction of the documents returned by the query that captured the attention of users. It is also estimated from the query log.

3. PROPOSED FRAMEWORK

3.1 Architecture for Processing the Log Files

Following Figure 1 explicitly describes the general architecture for processing the log files. Here, the log entries are stored into log file and the unnecessary data in the file is removed. User actions and their sessions are identified from the cleaned log file. If the format of the log files is different then, the retrieval of the user sequence and generation of frequent pattern of URLs is a difficult process.

The format of the first log file used in the data set is

<Client IP,Date and Time,URL visited,Status code,Browser >

The format of the second log file is

<Client IP,Http status code,URL visited,Date and Time, Browser >

The format of the third log file is

< Date and time,Client IP,Pages/directories,URL,Method, Username,Http status code,Referrer, Browser Mime type, Filter name,Filter reason,Profile,Interface IP,Interface port, Events,Page views,Bytes transferred,Elapsed time >

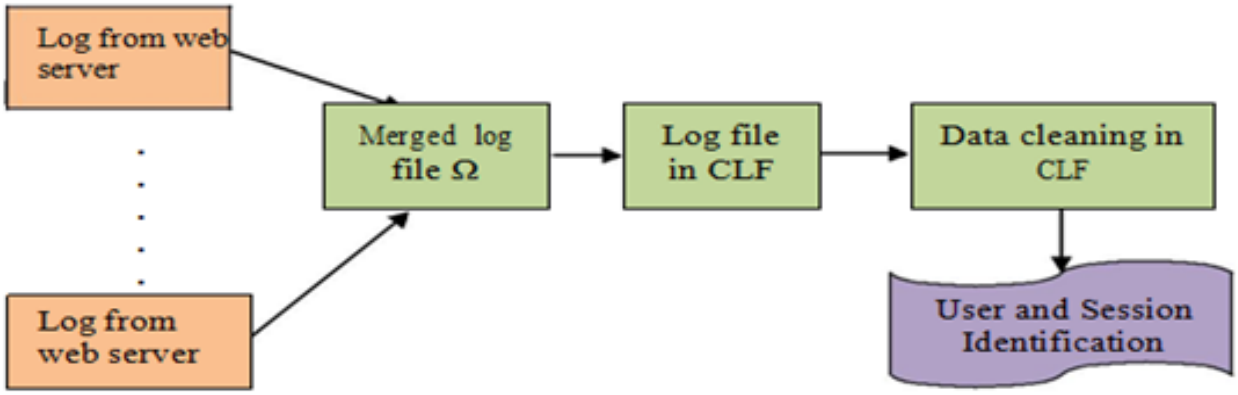

Figure 1. Process of Log Files CLF Mining

The discrepancies exhibited in the format of the log files set a predicament to access the user sequence. So it becomes mandatory to generate the common log file in which the log entries are organized under the attributes

< *Client IP,Date and Time,URL visited,Status code, Browser,OS* >

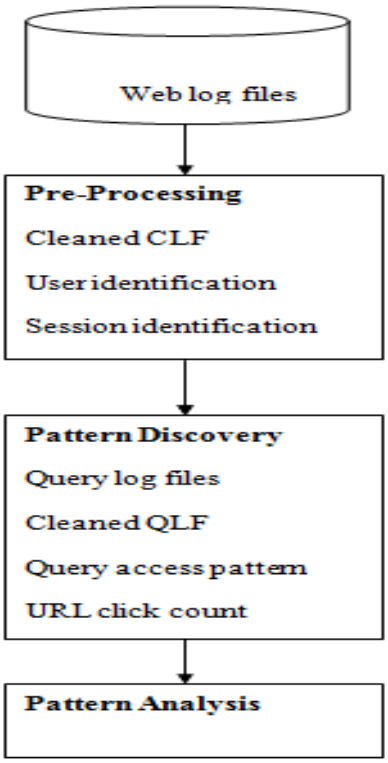

Figure 2. Mining flow of User's session Pattern

4. IMPLEMENTATION AND ALGORITHM

Implementation of proposed scheme performed using Java programming language. The implemented proposed scheme using Java development kit (jdk 1.7.0) on windows platform. System configuration consists of Intel Pentium dual core (1.86 GHz), CPU with graphic and 2 GB RAM. System has Windows 7 ultimate operating system installed.

4.1 Algorithm to Create a Merged File

Input : Different Log file

Output : The merged Log file CLF

begin
let n = number of log files for i from 1 to n do
for each log entry j in the ith log file do
Merge the log entry j in the file CLF
end
end
Arrange the log entries based on the access time.
Return (CLF)
end

4.2 Algorithm to Clean the CLF File

Input : CLF file which contains list of log entries

Output: Cleaned CLF file

begin
let inlog = set of input log entries
outlog = set of cleaned output log entries scode = status code of the log entry
ext = extension of the visited URL met = access method
Step 1: for each log entry i in inlog do
Tokenize the log entry i using StringTokenizer Store the tokens in outlog
Step 2: for each token i in the outlog do
if ((scode>=200 && scode<=299) && (met is either GET or POST) && (ext not equals to image formats and automated programs))
Write the token i in outlog
End

4.3 Algorithm to Identify Users from Pre-processed CLF

Input : Cleaned Pre-processed CLF log file.

Output: List of users begin

Let d1= Date object contains the user's navigation starting date and time d2= Date object contains the user's navigation ending date and time user = user information

Step 1: for i from 1 to number of log entries do Assign flag[i] is false

Step 2: for each log entry i in the CLF log file do if (flag[i] is false) k=0

Assign the URL from the log entry to user[x][k]

Assign user x starting time into d1; Change flag[i] is true; for each log entry j from i+1 to n do

if(IP address and browsing agent of ith and jth log entries are equal) Assign the jth log entry to the same user x; Change flag[j] is true Assign user x ending time into d2

Find the time taken by the user as dateDifference (d1, d2) Increment x for the next user
End

4.4 Algorithm to Identify the Sessions for each User from the Log File

Input : Log entries user-wise
Output: Sessions from each user and the visited URL begin
Let p = number of users
referrer = String contains the referrer URL for i from 1 to p do
for j from 1 to number of entries for ith user
if (referrer. equals(„-“) || timedifference(d1,d2)>30)
Store it in a new Session
else
backward: for k from 0 to j-1 // backward reference
if (referrer. equals (referrer of kth log entry))
The log entry j belongs to same user session and break backward
End

4.5 Algorithm QLF-cleaning

Input : Query log file
Output: Cleaned Query log file begin
Let inlog = Input query log entries outlog = Cleaned query log entries
for each query log entry i in inlog do
if an URL extension is an image format or URL is „-„ remove the query log entry from inlog
else
assign the query log entry to outlog
end
end

4.6 Algorithm QUERY_PATTERN

Input : Cleaned Query log file
Output : Query and access patterns begin
Step 1: for each user Ui do [where i=1 to number of users] Group the query log entries based on query

term
Step 2: for each Query term for user Ui do
Generate the access pattern for each session by concatenating the URLs
end
Finding URL Click Count

4.7 Algorithm SEQ_URLCOUNT

Input : User and Query-wise URL access pattern
Output: User and Query-wise URL click count begin
Step 1: for every user Ui do
Classify the query log entries based on the query term for every query term Qj do

Combine the URLs for each query term Qj Step 2: for each query cluster pattern Qj do

Find the URL count for the unique URL uk Store the userid, query term, URL and its count
End

Figure 3 shows the user's session behavior that describe the How may way to enter and stay into the website.

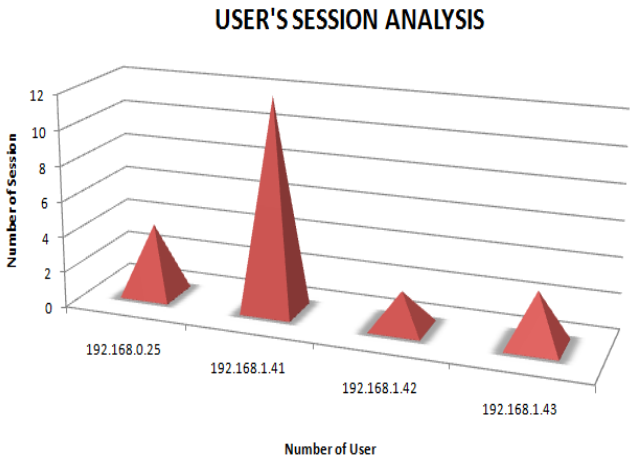

Figure 3: User's Session Behavior Analysis

5. CONCLUSION

As a result of lots of research and development, quality of web related services has improved significantly. But because of unprecedented growth of the Web and abundance of information, Web Users are not able to locate what they actually want within a reasonable amount time. This work has proposed one efficient tool towards an intelligent web. The proposed tool provides a good model for accessing the information related to particular log from the log files. The tool can be used in different type of mining areas of data mining applications. But due to the fast development in the technology and explosion in the number of users, Web Mining area still gives lots of research opportunities.

6. ACKNOWLEDGMENTS

This research paper is made possible through the help and support from everyone especially, please allow us to dedicate our acknowledgment of gratitude toward the Department of Computer Science And Engineering, R.C.E.W Bankrota, and Jaipur for providing necessary support and lab facility.

7. REFERENCES

[1] S. K. Pani Et El(2011), Web Usage Mining: A Survey On Pattern Extraction From Web Logs, Web Usage Mining: A Survey On Pattern Extraction From Web Logs, International Journal Of Instrumentation, Control & Automation (Ijica), Volume 1, Issue 1,

[2] Mehak(2013), Web Usage Mining: An Analysis, Journal Of Emerging Technologies In Web Intelligence, Vol. 5, No. 3,

[3] Govind Murari(2013), Web Content Mining: Its Techniques and Uses, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 11

[4] Priyanka Verma(2014), Web Usage Mining Framework For Data Cleaning And Ip Address Identification, International Journal Of Advanced Studies In Computer Science And Engineering Ijascse, Volume 3, Issue 8

[5] Shiva Asadianfam And Masoud Mohammadi(2014), Identify Navigational Patterns Of Web Users, International Journal Of Computer-Aided Technologies (Ijcax) Vol.1,No.1,

[6] Dr. Girish S. Katkar(2014), Use Of Log Data For Predictive Analytics Through Data Mining, Current Trends In Technology And Science Volume: 3, Issue: 3

[7] Pooja Kherwa (2015). Data Preprocessing: A Milestone Of Web Usage Mining , International Journal Of Engineering Science And Innovative Technology (Ijesit) Volume 4, Issue 2.

[8] Shivaprasad G.(2015), Knowledge Discovery From Web Usage Data: An Efficient Implementation Of Web Log Preprocessing Techniques, International Journal Of Computer Applications (0975 – 8887) Volume 111 – No 13

[9] V. Bharanipriya & V. Kamakshi Prasad WEB CONTENT MINING TOOLS: A COMPARATIVE STUDY.

[10] M.Santhanakumar(2015), Web Usage Based Analysis Of Web Pages Using Rapidminer, Wseas Transactions On Computers, Volume 14.

[11] Chandana S. Khatavkar(2015), A Hybrid Approach For Clustering Weblog(2015). International Journal Of Advanced Research In Computer Science And Software Engineering, Volume 5,Issue3.

[12] Liu Y., Miao J., Zhang M., Ma S. and Ru L., "How do users describe their information need: Query recommendation based on snippet click model," Expert Systems with Applications, vol. 38, no. 11, pp. 13847-13856,2011.

[13] Baeza-Yates R., Hurtado C. and Mendoza M., "Query recommendation using query logs in search engines," In Current Trends in Database Technology-EDBT 2004 Workshops, pp. 588-596, 2005.

[14] Dupret G. and Mendoza M., "Recommending better queries based on click-through data," In Proceedings of the 12th International Symposium on String Processing and Information Retrieval, pp. 41-44, 2005.

[15] Dupret G. and Mendoza M., "Automatic query recommendation using click-through data," In Professional Practice in Artificial Intelligence, pp. 303-312,2006.

[16] Burke R., "Hybrid web recommender systems," In The adaptive web, Springer Berlin Heidelberg, pp. 377-408, 2007

[17] Li L., Yang Z., Liu L. and Kitsuregawa M., "Query-URL Bipartite Based Approach to Personalized Query Recommendation," In AAAI, vol. 8, pp. 1189-1194, 2008.

[18] Ma H., Yang H., King I. and Lyu M. R, "Learning latent semantic relations from clickthrough data for query suggestion," In Proceedings of the 17th ACM conference on Information and knowledge management, pp.709-718,2008.