International Journal of Computer and Communication Technology

Volume 6 | Issue 4 Article 11

October 2015

ONLINE LOG ANALYSIS FOR DETECTING USER BEHAVIOR

ZINGADE A.M
Department of M.E.(C.S.E.), T.P.C.T.'s College Of Engineering, Osmanabad, India, abhizingade@gmail.com

P.P. KALYANKAR
Department of M.E.(C.S.E.), T.P.C.T.'s College Of Engineering, Osmanabad, India.,
kalyankarpravin@rediffmail.com

ONLINE LOG ANALYSIS FOR DETECTING USER BEHAVIOR

ZINGADE A.M.[1] &P.P.KALYANKAR[2]

[1,2]Department of M.E.(C.S.E.), T.P.C.T.'s College Of Engineering, Osmanabad, India.
Email:abhizingade@gmail.com, kalyankarpravin@rediffmail.com

Abstract-Most search engines are not provided the facility of expected results every time. Because two words can have same meaning e.g. if query "block size" is entered in search engine then search engine will display the records related to operating system as well as the living apartment block size. But search engine don't know about what actually user expect. That is data related to operating system or data related to living apartment. So generally it will display both type of links

In this paper we are managing the search engines on the fly. That is the paper is related to personalize the search of user on the fly without asking users to tagging or rating for the page.

Keywords- *query, page, links,behavior,on the fly,search logs,access logs,user clicks*

I. INTRODUCTION

Various techniques and applications are implemented in data mining field to mine the data and outcomes the result which are most expected according to the user's perspective. In case of Web based applications also the data mining field gets emerging. In day to day life users or the developers mostly uses the search engines. But they don't get their expected results in a one shot. They have to spend much more time to work on the same topic to achieve the expected result. If they are not getting the expected results in one shot then the process gets very time confusing. And user gets bored on the same task. So we can develop a technique which will reduce the work of user. Also which will shorten the time of user's searching process by personalizing the users search engine according to his behavior on the search engine. So he even doesn't get the knowledge of our technique or the process which is happening at the background. Means without the communication with user (e.g. ask for the rating) our search engine automatically gets personalized. And user get into wonder that the search engine is showing what he wants.

Here we are using a different technique to find out the human behavior on the fly. Here we are doing the following things. If user enters "block size" query in search engine then normally search engine displays both type of links related to operating system and living apartment and may be other ones. So this is first time result. So if user wants the result from operating system concepts then he can choose respective link and open respective page. And if user wants the result from living apartment concepts then he can choose respective link again and open respective page. After that our algorithm will check out which type of link is opened by the user. And try to find out the page type means the page belongs to which category i.e. either operating system or living apartment. If it is related to operating system then it will increase the count or rank related to operating system, for e.g. operating system keyword or the variable like cout_for_os increases by one.

So we can also use another algorithm to find out the page category. That might be text counting in a page or other technologies to find out the category of page. So if user first click is related to operating system then our rank of operating system is 'one'. And no other rank we are got here. So automatically when user goes to the next page then mostly links are related to the respective fields. First page may not get modified but the next pages are get modified according to users perspective. And automatically it gets personalized search engine without disturbing or asking questions to the user.

If two fields are getting same value of count variable then it will display approximately fifty percent of result from one filed and remaining from other field. And accordingly it will work for all queries. If a word having more than two meaning then also it will find out the respective meanings of word (we can use here dictionary facility to find out the word's meanings). So it will get lexically as well as semantically personalized.

It means if user enter a word in search engine's text box then our dictionary facility will find out the different meanings of the entered word. And try to pop out respective pages. And create the count variables for every different meanings of the word. And among user will try to choose preferable link that he wants to search and then finally our previous algorithm to track behavior of user will work.

II. RELATED WORK

The Number of techniques devised to create a search engine familiar with user or we can tell it as personalized search engine. One of the techniques is to ask the user to tag the page. So after using the respective link user will get the number of questions to give feedback about the link. Like whether the link is useful or not. So user can enter their respective information. In such a manner user are asked by the

software about his feelings. And store the respective tagging information into the database. So that algorithms might store the results at the database.

So according to that when next user or the same user comes next time for the query suggestion then the algorithm find out the query entered and its values stored in respective database. And if the values are higher for the respective queries then the respective links are comes in front of the user.

So again user will access the same page and again tag or rank the page according to the algorithm. So the tagging may be a question or it may be a rating in the form of stars. And star depicting the valuation of page or link. If it having the value of for e.g. one it means that is less useful for the user. And if it is for e.g. five then it means that it is more useful for the user than previous one.

But the drawback of this technique is we need to ask to the user for the rating of the page. Which is sometimes boring task for the user. Sometimes user is busy with his personal routine like accessing his bank account or doing the emergency air ticket reservation or train ticket reservation. And we are asking for rating so he may not be answered our query. And it is useless. Sometimes if user is not in good mood then he can again enter any values like if the page is not useful then also he can enter five rating for the page. And our algorithm will blindly accept that and work accordingly. So it might not work in long run.

One another technique is introduced the query-flow graph, where a graph structure depicts the behavior of query. Here graph which is used is directed graph if direction is from q1 to q2 then it means that the both links are belonging to the same search mission. Here each edge can have the strengths which explain its likelihood and the path explains the searching behavior of user [3]. So this can be used to find out the related queries. The use of query flow graph is in two applications, first is finding logical sessions and second one is query recommendation. But our work is differs from these prior work because we want the result should be expected according to the view of user. So this technique defined in [3] is for query suggestions and session managing but our technique is for the further processing of the user after taking suggestion from the user.

Another technique is devised in [5] is used to understand the web users search pattern. Sometimes users submit few queries and may he search number of different topics. Mostly user will not stick on a single search mission. So he can change his mind and start to search different information so the identification of topic, changes within a search session is an important branch of search engine user behavior analysis. So this topics is mainly devised to find out the topic identification

In technique explained in [2], the query grouping is done. Also query click graph is created. But both may not work properly if used individually. So, According to [2], here both graphs that are query graph and query click graph are combined to form new graph called as query fusion graph. And by merging both techniques it can organize the user search history.

Another technique explained is based on the temporal data. That is if user is searching for the computers history. But not mentioning the year by which user wants to search history. Then generally search engine shows all histories. But temporal data can be used to manage the respective data that user want [4]. For example, if user is searching for the storage device history. Then by default it will open current trends in the field. But if user is willing to see the history in some previous back years then automatically search engine will not display the required result, so the temporal data technique is used here. Generally here timestamp is used to store time information about the pages like in email and so on. Also it will search a timestamp for each and every pages so according to it, it will display the most popular pages on top and so on.

In [6] also they have mentioned ways to make result which are most personalize. They have referred a technique defined in a page rank. Also they have referred a technique called as topic sensitive page rank. If there are two pages and one page's link is mentioned in another page then it may be called as page rank. If the link is present in first page for the second page then it might be called as the link is important according to the authors' point of view in first page for the second page. Also another technique is mentioned called as topic sensitive page rank mention in [7]. The Topic-Sensitive PageRank scheme proposed is an another technique which is somewhat next version of PageRank that can actually provide different rankings for different queries, while at the same time it will remain the basic advantage of the standard PageRank. In the Topic sensitive page rank scheme, multiple scores, instead of just one, are computed for each page, one for every topic that we consider. So therefore it is not limited to only one topic but it is refers to the multiple topics at the same time and maintain the multiple scores for the multiple topics.

III. METHODOLOGY

A. How user work on serach engine

When we observer the behavior of user with the search engine then we found that all the users are not behaving in the same manner. One user may search with one page only and work on single topic. But in some situation a single user can work on more topics at the same time so it leads to opens more pages in different tabs of the search engine. So our algorithm should firstly able to find out that whether user is working on a single topic or he is searching for more than one topic at the same time. Also another fact is there which is, When we think about the users on the internet and try to segregate them into categories then

there are one type of users which are may be called as subject sensitive users , which are mostly belongs to only topics, their searches are only belongs to the respective topic. Another type of users may be surfing oriented means they are not actually searching for the respective subject. But only follow the link and jump from one link to another link and try to find out the information. And these types of working are executed for more than one page or by using more than one browser at the same time. So the algorithm should be work for every browser and for every tab which are created newly at the time of working in a same browser. So the application may be like the plug-in which will embed into the browser and works for the search engine. So we can divide user's behavior into the clusters that may be one or more according to the searching situation of the respective user.

Generally, consider that the whole data situated on World Wide Web as 100 % then, when user works on search engine then actually he needs only 10 % of data among the 100% of data on the web. So because of that reason when user prefers to enter particular word as search keyword in search engine then mostly he gets only 10 % of useful results or the links in front of him. And may be remaining are useless. So here search engine is giving every possible result to the user but user accessing only 10 to 20 percent among them. So the algorithm of the search engine who gives most results for user among 90% of result is useless. And this may be the unnecessary working of algorithm. If algorithm of search engine giving thousands of result for user and user only accessing nine or ten among them then according to the algorithm complexity it is doing some unnecessary working. So we have to reduce the working of algorithm to putting the results that mostly useful for user.

B. How effectiveness can be put in algorithm

In this paper we are using a technique where on the fly users choice is recorded in the database. So we can take the example of a query where a query can have two meanings. For e.g. "block size". Here we are taking the example of two words in a query. And when we enter this query in search engine then the algorithm of search engine generally displays the link from block size related to living apartments as well as block size of hardware storage devices that is from operating system concepts. So the search engine only flashes the pages where the respective term includes. So it may be not related with user. Also the term block size is related to the different fields and search engine flashes the pages from every possible fields. So user need to find out the specific link that he wants. And once he clicks on a link and if he thinks it is useful then he goes on to the same link. And our algorithm just scans his behavior. And algorithm execute accordingly to flashes most of the pages that user want.

Generally we are gathering data on the fly. Here data means the user behavior on the entered query. So after gathering that data we can take number of decisions through that data. It is also useful for the organizing and analyzing the user's nature. So data collection is very important phase in our scenario.

Generally it is like learning process. When user is working or surfing on the internet then our algorithm just trying to make search personalized. That is it is just identifying users behavior or the intension towards the link. If his intension towards link is positive, then algorithm just catches that behavior and store respective data in database. So, next time algorithm can work according to the created database to identify the next links or the pages to be flashed.

So this is the working of user click but here we not only going to use the user clicks to achieve our result, here we also use the *search logs*. Search logs are records of the user requests for information from your index. You can generate and export this information, and then use it as input to your preferred log analysis software or reporting software.

The various information that search logs provided are as follows,

- What types of different queries are users making in various periods of time?
- How fast the search engine provides the result to the user?
- Are users getting the results they need according to their query, lexically and semantically?

So, these types of information are stored in a search log.

And finally we are using here another mechanism called as *access log*. An access log is a list of all the requests for individual files that people have requested from a Web site. These files will include the HTML files and their embedded graphic images and any other associated files that get transmitted. The access log can be analyzed and summarized by another program.

In general, an access log can be analyzed for the following purposes:

- The number of visitors to a home page it includes only unique first-time requests
- The origin of the visitors in terms of their associated server's domain name (for example, visitors from .com, and .org sites and so on)
- How many requests for each page at the site, which can be presented with the pages with most requests listed first so it may be like in decreasing order of number of page requests count
- Usage patterns in terms of time of day, day of week, and seasonally. It means that some

sites user only access at particular time(for e.g. Some types of stocks or trading that can be only checked finally at the particular day of week)

Access log keepers and analyzers can be found as shareware on the Web or may come with a Web server.

So basically we are using here the user clicks records as well as search logs and finally the access logs. So we are combining all these techniques to find out the links which are more preferable for the user.

Here we are saving the user clicks record in database to find out that, after entering the query which respective links are clicked by the user, so this record is stored in a part of database for future use. Again Search logs are used to put the information about user search's that is what type of queries users are raising and how the results are popping and at the same time how records are useful for them, So these may be combined with user click to find out that whether the pages are useful for the user or not. And finally the access logs it will store the information about a page that is if a page having highest rank then it means that it is most usable by the user. And most of the users prefer for the particular page. So we can conclude that, users prefer the respective links for the respective queries. So we again combine this technique with the previous techniques of user clicks and search logs.

So when algorithms works as a whole then all the time search logs are keeping the record on users behavior as well as user clicks maintain the user behavior record for the usage of pages and finally access log stores the most popular pages and whatever user are using. So when we combine all these features then automatically our search engine gets familiar with user. Because if user is entering a particular query then firstly the first page of search engine is default page. But the user can work on that page so by using user clicks feature we stores user's usage information. As well as at the same time we maintain the search logs to find out the users query styles and the percentage of usage on the respective links that are popped for the user and which are considered as useful. And if number of users prefer a page for the respective query. Then automatically next time it will open the pages which are stored in a database as a history. So mostly search logs and access logs are saved in a database to maintain the history and which will be used at the next time when user prefers to search on the same query. And when user is surfing then users search record and users clicks are used to make search engine personalize. So first time as we told first page is default page but next time when user clicks on the next page link then automatically our algorithm find out the page category according to the previous click of user and find out that, was it useful for user or not? If it finds out that the link was useful and user prefers that then it will save the category of page and when next page link opens then it will flashes most of the links which are according to the category of previous linked page. So automatically it will flash the most user friendly pages and our search engine gets more personalized.

So user clicks records are used to make search engine more personalized and the search logs and access logs are used to create a history of user's behavior. So by using user click record algorithm will open personalized pages on the fly so it will work at runtime but if the same record is present in our history through search logs or through access logs then instead of run time calculation it will directly use our history to open more personalized pages. So this paper mostly used to make search engine personalized at the first time when users accessing the search engine through click records and may be at the next times it will use the history to again make the records personalized though search logs and access logs. Because when we have personalized records in history then there is no need to again do the calculations on the fly. So we use user clicks only when user attempting the queries first time. If the same query is used by the user again then we go through the history and if the record is not present in history then we do the on the fly calculations. So at both times that means on the fly calculation time and the history based calculation time that is both times our search gets more personalized.

C. Algorithm

Input: users click record and the created respective variables.

Output: the respective pages to be flashed.

(1) Display the first search page as it is.
(2) Input users click record in the database.
(3) For next pages check the database and change the users entered query in another form according to the value of variable which is stored in database. (So here at background side we are changing the users entered query. For e.g. if query is like "block size" then we convert it into "block size for operating system". It is only at background user can't see that changes.)
(4) And finally search the next page according to the query created by using database values.

(Note: If users are searching the records first time then do on the fly calculation. And when data is already getting in database then use history through search logs and access logs.)

Sometime user may suddenly chage his intent and tries to capture the new context. So at that time our algorithm should find out the changes happening in search query tag and work accordingly. [1]

We can gather data according to the users click information. That is we can follow through the user click. It means that how user is working for the respective query. If after entering query user is clicking on some link then it means that it would be useful for him so we can do the matching or create a graph according to its working nature. [2].

Our algorithm task is started here. Here it will check the user click information of the user. If user is clicking the link related to the living apartment then it will create a count or rate variable on the fly related to the word "block size" of living apartment and it will convert the query to "block size of living apartment". (But this conversion is at the background side so the end user cannot get information about this. And this newly formatted query is only stored until working progress of same query goes on. After that automatically that field get vanishes or deleted) and set the value of count or rate variable to one. So here user can use that link which opened and then close and comes back to the main search engines page. Here if he will not get more useful links then he can go to the next page. So at the next time our algorithm just finds out the value of count or rank variable. And the respective query conversion. And according to the respective query conversion our algorithm tell to the search engine to pop out the pages which are according to the new query which is created by our algorithm. And immediately when next page appears automatically it will shows the links related to the new formatted query like "block size related to the living apartment". So we can segregate them.

So meanwhile our click algorithm works and rate to the users query automatically when user goes on. The more personalized result he gets. And after some iteration he will get results which are mostly useful for the user. And our search engine gets personalized without intervention of the user or without disturbing the user.

So most time algorithm works on the fly but when user getting working on the same type of query for long duration then it will not do on the fly calculation all the time because it will use history. So both functions are used at the background side. And at front side user thinks that the search engine gets personalized. And the main thing here is that the search engine gets personalized without intervening with user even once.

IV. APPLICATION

Whatever we are doing with query and click all the information is going to store in the database. So the stored data we can use for the several purposes like query suggestions, query log Analysis, user profiling and personalization, and more.

Also we are creating history in database so it may be useful further to see the back records. So it may be used like history feature of search engine and same may be use for the query suggestions and other query related applications like query optimization and so on.

V. CONCLUSION

Mostly the developers need to take users decision when searching the behavior of user. So it is like we are asking to the user and according to the answer of user we are creating the scenario that user likes. But mostly users are not interested to share his feeling on site every time he is surfing. So may be user is not interested in those activities.

Therefore our new technique to identifying the user's behavior according to his working will more preferable without asking question to the user. This may be like that learning. So according to the users perspective user is feeling like that, when he goes ahead on searching then he is getting most expected result. And he will get interest in the same. So at most time when user works on the search engine then he prefers such type of working which is mentioned above.

REFERENCES

[1] Tomas Kramar And Maria Bielikova, "Detecting Search Sessions Using Document Metadata and Implicit Feedback"

[2] Heasoo Hwang, Hady W.Lauw, Lise Getoor and Alexandros Ntoulas "Organizing User Search Histories"

[3] Paolo Boldi , Francesco Bonchi , Carlos Castillo , "The query-flow graph : model and application"

[4] Rosie Jones , Fernando Diaz, "Temporal Profiles of Queries"

[5] H. Cenk Ozmutlu, Fatih Cavdur, "Application of automatic topic identification on Excite Web search engine data logs"

[6] Feng Qiu , Junghoo Cho "Automatic Identification of User Interest For Personalized Search"

[7] T. Haveliwala., "Topic-sensitive pagerank."

[8] Arindam Banerjee, John Langford, "An Objective Evaluation Criterion for Clustering"

[9] JI-RONG WEN, JIAN-YUN NIE, HONG-JIANG ZHANG, "Query Clustering Using User Logs"

[10] Mehran Sahami, Timothy D. Heilman, "A Web-based Kernel Function for Measuring the Similarity of Short Text Snippets"

[11] Wesam Barbakh and Colin Fyfe," Online Clustering Algorithms"

[12] Zhiyuan Liu, Yabin Zheng, and Maosong Sun, "Quantifying Asymmetric Semantic Relations from Query Logs by Resource Allocation"

[13] Doug Beeferman, Adam Berger , "Agglomerative clustering of a search engine query log"

[14] Nick Craswell and Martin Szummer , "Random Walks on the Click Graph"

[15] Kevyn Collins-Thompson Jamie Callan , "Query Expansion Using Random Walk Models"

[16] Oren Zamir and Oren Etzioni, "Web Document Clustering: A Feasibility Demonstration"

[17] Eldar Sadikov, Jayant Madhavan, Lu Wang, Alon Halevy , "Clustering Query Refinements by User Intent"

[18] Filip Radlinski, Thorsten Joachims, "Query Chains: Learning to Rank from Implicit Feedback"

[19] Tessa Lau and Eric Horvitz, "Patterns of Search: Analyzing and Modeling Web Query Refinement"

[20] Craig Silverstein, Monika Henzinger , "Analysis of a Very Large Web Search Engine Query Log"

[21] Alan L. Montgomery and Christos Faloutsos, "Identifying Web Browsing Trends and Patterns"

[22] Lara D. Catledge, James E. Pitkow, "Characterizing Browsing Strategies in the World-Wide Web"

[23] Paolo Boldi, Massimo Santini, Sebastiano Vigna, "PageRank as a Function of the Damping Factor"