*IOSR Journal of Computer Engineering (IOSR-JCE)*
*e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 16, Issue 3, Ver. IV (May-Jun. 2014), PP 48-54*
*www.iosrjournals.org*

# Efficient Information Retrieval System with Higher Priority for User Preference

T.GnanaJanani[1], M.SivaKumar[2]
[1(]*M.E.II Year, Department of Computer Science and Engineering, K.Ramakrishnan College of Technology Tiruchchirappalli, India)*
[2]*(Assistant Professor, Department of Computer Science and Engineering, K.Ramakrishnan College of Technology Tiruchchirappalli, India)*

***Abstract:*** *A search engine is a software system or a tool that is designed to search for information. Web search engines return lists of web pages ranked by the page's relevance to the user query and by number of times a web page viewed. A personalization technique is adopted to retrieve the user preferred result by re-ranking the user's clickthrough data. The re-ranking is performed by evaluating the weights of the user's clickthrough data. The web search is classified as content based and location based, so as to obtain the most relevant data. The content based search returns the ontology results where as the location based searches returns the location information of the geo-web query. The system can be used in hand -held devices where individual user is focused and can also be used in desktops where multiple users are considered. This improves the search engine robustness and also reduces the waiting time of the user.*

***Keywords:*** *click through data, geo-web, ontology, re-ranking, web search engine, re-ranking*

## I. Introduction

A search engine is an information retrieval system designed to find information stored on a computer system. A search engine can either be a manual system or an automated system. Automated search engine systems are used to reduce what has been called "information overload". The first tool for searching the Internet, created in 1990, was called "Archie". It downloaded directory listings of all files located on public anonymous FTP servers; creating a searchable database of filenames. A year later "Gopher" was created. It indexed plain text documents. "Veronica" and "Jug head" came along to search Gopher's index systems. The first actual Ib search engine was developed by Matthew Gray in 1993 and was called "Windex". The search results are usually presented in a list and are commonly called hits. Search engines help to minimize the time required to find information and the amount of information which must be consulted, akin to other techniques for managing information overload. It provide an interface to a group of items that enables users to specify criteria about an item of interest and have the engine find the matching items. The criteria are referred to as a search query. In the case of text search engines, the search query is typically expressed as a set of words that identify the desired concept that one or more documents may contain. The problem behind search engines is that it is said to be impersonal and do not consider user expertise level. It provides ambiguous results and it is not tuned to individual environments. The search accuracy can be improved by retrieving the category, observing the user behaviour, matching the user interest and also considering the user expertise level. Another major problem in mobile search is that the interactions between the users and search engines are limited by the small form factors of the mobile devices. As a result, mobile users tend to submit shorter, hence, more ambiguous queries compared to their web search counterparts [1]. In order to return highly relevant results to the users, mobile search engines must be able to profile the users' interests and personalize the search results according to the users' profiles. A practical approach to capturing a user's interests for personalization is done by analysing the user's click through data. A search engine personalization method is developed based on users' concept preferences and it is more effective than methods that are based on page preferences.

## II. Information Behaviour

Information Behaviour is the totality of human behaviour in relation to sources and channels of information, including both active and passive information seeking, and information use. Thus, it includes face-to-face communication with others, as well as the passive reception of information as in, for example, watching TV advertisements, without any intention to act on the information given. Information Seeking Behaviour is the purposive seeking for information as a consequence of a need to satisfy some goal. In the course of seeking, the individual may interact with manual information systems (such as a newspaper

or a library), or with computer-based systems (such as the World Wide Web). Information Searching Behaviour [2] is the 'micro-level' of behaviour employed by the searcher in interacting with information systems of all kinds. It consists of all the interactions with the system, whether at the level of human computer interaction (for example, use of the mouse and clicks on links) or at the intellectual level (for example, adopting a Boolean search strategy or determining the criteria for deciding which of two books selected from adjacent places on a library shelf is most useful), which will also involve mental acts, such as judging the relevance of data or information retrieved. Information Use Behavior consists of the physical and mental acts involved in incorporating the information found into the person's existing knowledge base. It may involve, therefore, physical acts such as marking sections in a text to note their importance or significance, as well as mental acts that involve, for example, comparison of new information with existing knowledge. The idea of adapting a retrieval system to particular groups of users and particular collections of documents promises further improvements in retrieval quality for at least two reasons. First, a one-size-fits-all retrieval function is necessarily a compromise in environments with heterogeneous users and is therefore likely to act sub optimally for many users. Second, as evident from the TREC evaluations, differences between document collections make it necessary to tune retrieval functions with respect to the collection for optimum retrieval performance. Since manually adapting retrieval function is time consuming or even impractical, research on automatic adaptation using machine learning is receiving much attention. Click through data [4] can be recorded with little overhead and without compromising the functionality and usefulness of the search engine. In particular, compared to explicit user feedback, it does not add any overhead for the user. For recording the clicks, a simple proxy system can keep a log file.

## III. Web Search Engine

There are basically three types of search engines: Those that are powered by robots (called crawlers; ants or spiders) and those that are powered by human submissions; and those that are a hybrid of the two.Crawler-based search engines are those that use automated software agents (called crawlers) that visit a web site, read the information on the actual site, read the site's meta tags and also follow the links that the site connects to performing indexing on all linked web sites as well. The crawler returns all that information back to a central depository, where the data is indexed. The crawler will periodically return to the sites to check for any information that has changed. The frequency with which this happens is determined by the administrators of the search engine.

Human-powered search engines rely on humans to submit information that is subsequently indexed and catalogued. Only information that is submitted is put into the index.The algorithm is what the search engines use to determine the relevance of the information in the index to what the user is searching for. One of the elements that a search engine algorithm scans for is the frequency and location of keywords on a web page. Those with higher frequency are typically considered more relevant. But search engine technology is becoming sophisticated in its attempt to discourage what is known as keyword stuffing, or spamdexing.

Another common element that algorithms analyse is the way that pages link to other pages in the web. By analysing how pages link to each other, an engine can both determine what a page is about (if the keywords of the linked pages are similar to the keywords on the original page) and whether that page is considered "important" and deserving of a boost in ranking. Just as the technology is becoming increasingly sophisticated to ignore keyword stuffing, it is also becoming savvier to web masters who build artificial links into their sites in order to build an artificial ranking.

## IV. System Design

The system for providing a personalized search was designed with two integral parts a client and a server as in figure 1. This proposal was made to use in the personalized hand held devices. When a need arises to use this system in a common desktop, a user identity is generated for providing personalization to the search.

The client is responsible for receiving the user's requests, submitting the requests to the server, displaying the returned results, and collecting users click through in order to derive user personal preferences. The server, on the other hand, is responsible for handling heavy tasks such as forwarding the requests to a commercial search engine, as Ill as training and re-ranking of search results before they are returned to the client. The user profiles for specific users are stored on the clients, thus preserving privacy to the users. It has been prototyped with PMSE clients on the Google Android platform and the server on a PC server to validate the proposed ideas.

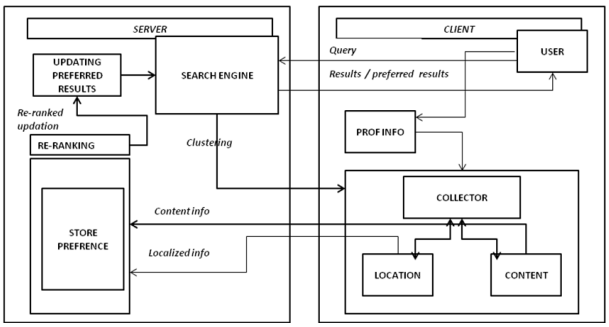


Figure 1 Working Model of the System

## 2.1 Query Processing

Exploratory queries could return too many answers, a phenomenon commonly referred to as "information overload". Thus, categorization and ranking present two complementary techniques to manage information overload (Chakrabarti, K. et al 2004). After browsing the categorization hierarchy and/or examining the ranked results, users often reformulate the query into a more focused narrower query. Therefore, categorization and ranking are indirectly useful even for subsequent reformulation of the queries. Default search engines like Yahoo, Google etc. provide the results by ranking the web pages with respect to the level of frequency. The frequency is estimated with the number of users views of that particular page. This ranking gives the same result when the user searches for the same query for n number of times. When the same user is looking for the results of the previously searched query then it leads to missing of previously viewed preferences. This issue takes much time for user to search their preferences among those obtained result. A ranking mechanism is used in such a way to solve this issue.

The user query has to be categorized in such a way that two different category arises. The categorization is done to provide varied results for varied user query. When the user searches for a random query based on some ontology then the obtained results will provide the results with respect to the previously viewed pages. When the user provides a query to identify the location or path of travel and if the query is to find the objects that lie in the requested path, then the result of the corresponding query is obtained through GPS system with a longitudinal and latitudinal value alone the path in which the user travels.

Categorization may also require the user profile identity if there exist more than one user of the system. The user profile identity is provided with unique identity number generated to identify the specific user preferences .A training data set is being used to work out this process such that a clear idea is being obtained.

The training set consists of both ontology search data set and a longitudinal data set with a priority value and subject value that provides the number of views. The author has given that a meaningful solution has to take into account the impact on query processing. The author has done implementation of the ranking function that exploits indexed access by drawing on insights from Fagin's Threshold Algorithm. The ranking is extremely domain and/or user specific, solving this issue is a difficult task. Thus it was not focused by the author

## 2.2 Clients

The clients are responsible for storing the user click throughs and the ontology derived from the server. Simple tasks, such as updating click throughs and ontology, creating feature vectors, and displaying re-ranked search results are handled by the clients with limited computational power. Moreover, in order to minimize the data transmission between client and server, the client would only need to submit a query together with the feature vectors to the server, and the server would automatically return a set of re-ranked search results according to the preferences stated in the feature vectors. The data transmission cost is minimized, because only the essential data (i.e., query, feature vectors, ontology and search results) are transmitted between client and

server during the personalization process. The client contains an observer phase that keeps track on users search history (logger) and the applications opened. These statics are sent to the server for further processes. The client collects the information about the user's interest from the log. It then checks for the user identification and observes the interest according to the user.The client has to check for users explicit information, users context, user browsing histories, user desktops/mobile applications (device profiles) and also user search histories. The Two Paradigms [12]: - Explicit vs. Implicit essentially, there is a need to have a model for collecting feedback from the user on various items. In Explicit User Modelling, the user is explicitly asked to rate the likeness for various items. The system records the ratings given by the users and analyses then to deduce future likeness for new items. In Implicit User Modelling, the system automatically gathers information about a user's interests and needs.

Data for user profiling may be gathered from the surfing history and surfing behaviour of the user. This includes time of visits, last visits, frequency of visits, number of out links followed, scrolling patterns, dwelling time, mouse clicks, mouse focuses etc. 'Curious browsers' have been designed to collect such data automatically and implicitly while the user surfs the net. Human-Computer Interaction systems may be employed to gather more of such data, like eye movements and eye-focus. Snippets gathered from page title, text contents also give valuable clues. Surfing history for a user may also include query history and output URLs selected by the user in the past. A device profile comprises the set of attributes (services and/or features) that are associated with a particular device. Device profiles include name, description, phone template, add-on modules, soft key templates, feature settings, multilevel precedence and pre-emption (MLPP) information, directory numbers, subscribed services, and speed-dial information. A user device profile can be assigned to a user, so, when the user logs in to a device, the user device profile that you have assigned to that user loads onto that device as a default login device profile. After a user device profile is loaded onto the phone, the phone picks up the attributes of that device profile.

As can be seen in figure.2, Google search history is shown in chronological order. Google can also show the search history in terms of various categories such as Ib, images, news, shopping, Ads, videos, maps, blogs, books, visual search, travel and finance. However, it does not organize the search history based on related similarity of the searches. Query groups help search engines in many applications. The key features of search engine can be improved by making query groups meaningfully. The utilities of query groups include collaborative search, sessionization, query alterations, result ranking and query suggestions the dynamically categorize SQL query results by inferring a hierarchy based on the characteristics of the result tuples. Their domain is the tuple attributes and their problem is how to organize them hierarchically in order to minimize the navigation cost.

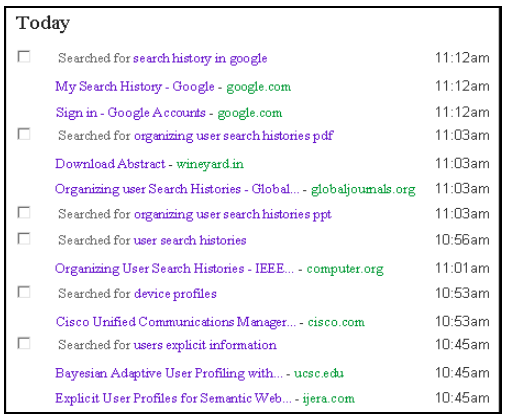

Figure 2 Search history of user organized by Google

1) Click through Data collection: The servers contain the concept space that models the relationships between the concepts extracted from the search results. They are stored in the ontology database on the client. When the user clicks on a search result, the clickthrough data together with the associated content and location concepts are stored in the clickthrough database on the client. The clickthroughs are stored on the clients, so the server does not know the exact set of documents that the user has clicked on. This design allows user privacy to be preserved in certain degree. If the user is concerned with user own privacy, the privacy level can be set to high so that only limited personal information will be included in the feature vectors and passed along to the server for the personalisation . on the other hand, if a user wants more accurate results according to user preferences, the privacy level can be set to low so that the server can user the full feature vectors to maximize the personalisation effect. The results are updated using a formula that updates the weights in user profile using a relevant document $d_{N+1}$ from the relevance feedback. $W^u_i = W^u_i + \Pi * W_i{}^d{}_{N+1} * (1/N * \Sigma^N_{k=1} W_i{}^d{}_k)$ where $W^u_i$ is

the weight of category $C_i$ in a user profile u , N is the total number of past relevant documents , $d_{N+1}$ is the current relevant document , η is the learning rate which ranges from 0 to 1,$w_i d_{N+1}$ is the weight of document $d_{N+1}$ in $C_i$ ,$w_i^{dk}$ is the weight of past relevant document $d_k$ in C.

### 2.3 Server

Heavy tasks, such as re-ranking of search results, are handled by the server. Server's design addressed the issues: Limited computational power on mobile devices, Data transfer minimization. Server performs of three major activities: 1) Classification of user's search query, 2) Re-ranking the search results at the server, 3) Ontology updates and click through collection at a mobile client. The server also maintains user profiles and device profiles. It includes crawler, indexer, and personalization filter. The indirect proxy server is used to track the user from the server side and also creates a query log for future analysis.

**2.3.1 Classification**: Classification module at first collects information about the user's interests from logs and creates a relevant document into category hierarchy. The user interests are captured with the use of the determined category weights. The weights are evaluated by two factors as a probabilistic distribution of weights over terms and as a probabilistic distribution of weights over categories.Text classification performs the task of automatically sorting documents into pre-defined categories which are widely used in personalization systems. Text classification is carried out in two phases: training and classification. In training the system is trained on a set of pre-labeled documents and the system is made to learn features that represent each of the categories. In classification, system receives a new document and assigns it to a particular category.Text classification generally has two variations with the classifiers: Flat classifier and Hierarchical classifier. Flat classifier has no relationship between the categories but they have good accuracy. Flat classifiers can produce results in single classification; a best example can be said as the top 100 Yahoo! Search results are obtained in ~500ms after classification. Hierarchical classifiers provide a parent-child relationship between categories and they are used with hierarchical knowledge bases. It does not have much accuracy as flat classifier but an improvement in accuracy can be found. It requires one classifier for every node in hierarchy and document must go through multiple classifications before being assigned to a category. For example the top 100 Yahoo! Search results are obtained in ~ 2sec after classification.

Algorithm: k-means clustering

K-Means Clustering algorithm is an idea, in which there is need to classify the given data set into K clusters; the value of K (Number of clusters) is defined by the user which is fixed. In this first the centroid of each cluster is selected for clustering and then according to the chosen centriod, the data points having minimum distance from the given cluster, is assigned to that particular cluster. Euclidean Distance is used for calculating the distance of data point from the particular centroid.

This algorithm consists of four steps:

1. Initialization -In this first step data set, number of clusters and the centroid that I defined for each cluster.
2. Classification-The distance is calculated for each data point from the centroid and the data point having minimum distance from the centriod of a cluster is assigned to that particular cluster.
3. Centroid Recalculation-Clusters generated previously, the centriod is again repeatedly calculated means recalculation of the centriod.
4. Convergence Condition

Some convergence conditions are given as below:

4.1 Stopping when reaching a given or defined number of iterations.
4.2 Stopping when there is no exchange of data points between the clusters.
4.3 Stopping when a threshold value is achieved.

5. If all of the above conditions are not satisfied, then go to step 2 and the whole process repeat again, until the given conditions are not satisfied [13]

Re-Ranking:--When a user submits a query on the client, the query together with the feature vectors containing the user's content and location preferences (i.e., filtered ontology according to the user's privacy setting) are forwarded to the server, which in turn obtains the search results from the backend search engine (i.e., Google). The content and location concepts are extracted from the search results and organized into ontologies to capture the relationships between the concepts. The server is used to perform ontology extraction for its speed. The feature vectors from the client are then used in RSVM training to obtain a content weight vector and a location weight vector, representing the user interests based on the user's content and location preferences for the re-ranking. Again, the training process is performed on the server for its speed. The search results are then re-ranked according to the weight vectors obtained from the RSVM training. Finally, the re-ranked results and the extracted ontologies for the personalization of future queries are returned to the clientThe personalization score of a result is based on similarity between the each result profile and user profile. The re-rank results are based on

personalization score and rank given by the search engine.The ranking algorithm has parameters like "R", the result, "S(R)" the original ranking given by search engine, "PS(R)" is the personalization score of the result R and "a" is the personalization factor which ranges from 0 to 1. The formula to evaluate is given here Rank(R) = a * (PS(R)) + (1-a) * (S(R)).

### 2.4 Performance Evaluation

A performance evaluation is done with five varied users who are made to use the application. The current location of each user is taken with the help of GPS system in the smart phone. The user preferred search results are obtained with an average rate of above 90 % when evaluated with the web search engine results. The average time taken to fetch standard search result, re-rank and display them is less than 2 seconds which is acceptable and almost real-time on a mobile device. The user interests can in fact improve the web search results and the effectiveness with respect to the user's waiting time , energy consumed per search query and the data transfer rate. The fig.3 shows the Rank results with energy variation per query. The search query in Google consumes 0.003 kwh every time but when the user preferred search is used the consumption values is decreased as shown in graph. This uses only the content-based features in personalization. The observation is that personalized content performs the best because it contains the relevance values among all methods.

.

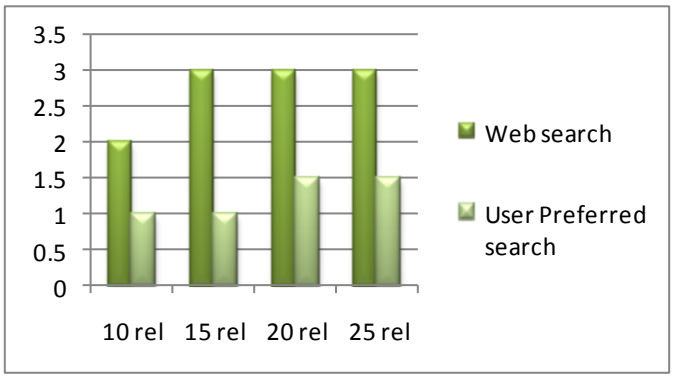


Figure 3. Rank results with energy variation per query

## V. Conclusions

The work studies the unique characteristics of content and location concepts, and provides a coherent strategy using client-server architecture to integrate them into a uniform solution for the mobile environment. The user preferred mobile search engine is an innovative approach for personalizing web search results. By mining content and location values for user profiling, it utilizes both the content and location preferences to personalize search results for a user. The results show that location helps improve retrieval effectiveness for queries in various locations. The design has been adopted with the server-client model in which user queries are forwarded to a server for processing the weighing and re-ranking quickly. This work is a phase 2 work done on the real device with android platform. The phase 1 work was studied with the training set values on a PC on a .NET environment. The training set where taken with location values of all the states in INDIA and the content results from the pre-stored data sets. The process of clustering , re-ranking , weighing has been performed with those training set in-order to avoid the conflicts during the implementation on the real device. The architecture of phase 1 work has been modified to bring out more efficiency in the system. Those changes are found to reduce the overall energy consumption of the mobile device when compared to normal web search engine. The future work is to enhance the same system with operating system compatibility such that this system should be compatible with all the basic and high operating systems of hand-held devices. The graphical user interface can further be improved to make it more clear for all kind of users.

## Acknowledgment

The authors would like to express their sincere thanks to the editors and the reviewers for giving very insightful and encouraging comments.

## References

[1]. Kenneth.W.L, Dik.L,(2013), "Personalized mobile search engine",IEEE Transaction on Knowledge And Data Engineering, Vol 25, No. 4.
[2]. Wilson D. (2000). "Human Information Behavior", informing science,vol. 3 No.2 ,pp.2-3.
[3]. A. MarkoItz, Y.-Y. Chen, T. Suel, X. Long, and B. Seeger. , (June 2000) "Design and implementation of a geographic search engine". In 8th Int. Workshop on the web and Databases (webDB).
[4]. Joachims, T. (2002) "Optimizing search engines using Click through data", ACM 1-58113-567-X/02/0007.
[5]. Comtet, L. (2005) "Advanced Combinatorics: The Art of Finite and Infinite Expansions", rev. enl. ed. Dordrecht, Netherlands: Reidel, pp. 176- 177.
[6]. Page.L and Brin.S,(1998)" The PageRank Citation Ranking: Bringing Order to the web "http://dbpubs.stanford.edu:8090/pub/1999-66/
[7]. Alessio.s,(2005), "A survey of ranking algorithms", asignori phd-report -archive/macros/latex/contrib/supported/IEEEtran/
[8]. J.Kleinberg,(1999)"Authoritative sources in a hyperlinked environment". Jmynal of ACM (JASM).
[9]. Boyan J., Freitag D., and Joachims,(1996) T. "A machine learning architecture for optimizing web search engines." In Proc. AAAIWorkshop on Internet Based Information Systems.
[10]. Culliss G., (2007),"The Direct Hit Popularity Engine Technology",. A White Paper, DirectHit.
[11]. Joachims T. and Radlinski F.(2000) "Search engines that learn from implicit feedback". IEEE Comp., 40(8):34–40.
[12]. Srinvas.C (2012)," Explicit User Profiles for Semantic web Search Using XML" , International Journal of Engineering Research and Applications (IJERA) ,Vol. 2, Issue 6, pp.234-241.
[13]. Navjot.K,Jaspreet.K.S,(2012),"Efficient K-Means Clustering Algorithm Using Ranking Method In Data Mining," International Jmynal of Advanced Research in Computer Engineering & Technology ,Vol.1, Issue 3.
[14]. Manpreet.K,Usvir.K,(2013) "A Survey on Clustering Principles with K-means Clustering Algorithm Using Different Methods in Detail",IJCSMC, Vol. 2, Issue. 5, pg.327 – 331.