THE INTERNATIONAL JOURNAL OF SCIENCE & TECHNOLEDGE

Re-Ranking Clickthrough Event in Search Engine Using Ontology

R. Thenmozhi
M.E, Department of Computer Science, Arunai Engineering College, Tiruvannamalai, India
S. Bhuvaneshwari
M.E, Department of Computer Science, Arunai Engineering College, Tiruvannamalai, India
T. Chellatamilan
Professor & HOD, Department CSE, Arunai Engineering College, Tiruvannamalai, India

Abstract:
In Mobile based search engine the major problem is that the interaction between mobile users and search results are limited .In order to manage these problem collect user query and their relevant result to satisfy the user profile according to the interest. To perform this observe the different types of concepts in the personalized mobile search engine, it captures the user preferences concepts by mining click through data. In Personalized mobile search engine preferences of each user are ordered in ontology based model and each user profiles are ranked with the use of multi-facet for future search results. The search result can be classified into location and content based concepts based on their importance information. Improve the search engine result by investigate methods to develop normal query travel patterns from the location and click through data to enhance the personalization effectiveness of search engine. By introducing an association rule mining algorithm collect the different travel patterns by original search engine result in each and every query of user from the original personal mobile search engine profile. Association rule learning is used for finding the interesting query travel pattern results from each user query in search engine. From this query related patterns of the user to identify strong rules discovered in databases using different measures of interestingness. They introduced association rules for discovering regularities between normal patterns and query related patterns in the personalized mobile search engine result.

Keywords*: profiling, association rule, clicked data*

1. Introduction

In computer science field search is the major problem. The ability to index and search for information increases the ease of access to information which implies lower cost of access and greater spread of the information. However, most of the mobile phones in developing countries like India are basic phones devoid of access to rich information and compute resources. Despite having a focus towards mobile technology, the bigger international technology companies have been unable to provide access to rich information to people
with basic phones in countries like India. We intend to Develop a contextual search engine which is customized based on a user's profile and location of the query and implement it in a way so that people with basic phones can use it. Earlier personalization techniques were based solely on the computational behavior of the user (visited URL, viewed documents) to model his interests regardless of his surrounding environment. The main limitation of such approaches is that they do not take into account the dynamicity of the user interests regarding his environment context. A mobile device has limited power and memory that is not suitable to handle complex program. And, normally, agent needs to handle complex task, such as matching user request, communicating with other agents or finding personalized service. Agent is some kind of thread, which consumes a lot of resource such as memory. Further, when an agent communicates with another, it needs to keep connected on the Network that consumes additional power. Therefore, we figure that as a user uses mobile device, such as mobile phone, with limited power and memory, dynamic mobile agent is definitely needed. In this paper, we address the above issues to present a mobile user agent, which provides personalized access to web services for users according to user's preference. general process of our approach, which consists of two major activities: 1) Reranking and 2) Profile Updating.

- **Reranking**: When a user gives a query, the search results are obtained from the backend search engines. Then the search results are combined and reranked according to the user's profile trained from the user's previous search activities.
- **Profile Updating**: After the search results are obtained from the backend search engines, the two concepts (i.e. important terms and phrases) and their relationships are mined online from the search results and stored, respectively, as contextual ontology and location ontology. When the user clicks on a search result, the clicked items together with its associated contextual and location concepts are stored in the user's clicked data. The content and location ontologies, along with the clicked data, are then employed in ranking training to obtain a content weight vector and a location weight vector for reranking the search results for the user. Association regulation generation is typically opening up interested in two

divide steps: Primary, minimum support is practical to discover all frequent query patterns in a user clickthrough data. Next frequent query itemsets and the minimum confidence restraint be second-hand to appearance rules. To discover a query traveler's interest extract beginning search based user click all the way through files whilst the personal user search the consequences on or after mobile .

Whilst user enter query base path or traversal patterns are recognized initially and after that we make frequent item set with the intention of number of instance the user click thorough files and find the majority significant travel patterns in the click through files. This investigate focus on the travelers who use mobile search contain the majority frequent based links in together location and concept based ontology ,previous to so as to we discover the frequent item set that is additional numeral of period user look for the comparable web pages or concept and location.

Beginning this compute the support and confidence standards of the click through files and the majority relevant regular query patterns results are considered as consumer the majority important concepts and location then yet again go on the concept to rank the feature for both content and location ontology.

Association Rule Mining (ARM) query travel pattern to explore for go target that is user concept consequences ,practical data mining and association rules method to investigate the association among travelers' profile and their transactions in the data .After this examine the identify majority important pattern to investigate the outcome and can amplify opportunity for the competitive operations of tourism firm to respond the travelers' demand effectively.

Users may have very diversified preferences in the pages they target through a search engine. It is therefore a challenging task to adapt a search engine to suit the needs of a particular community of users who share similar interests. Some previous approaches for optimizing search engines require training data generated from users' explicit relevance judgments on search results. Users are usually unwilling to give such feedback because of privacy concerns and the extra effort required by them. To overcome this problem, we propose using click through data, which is a log kept by the search engine on the queries submitted by users, the returned result items, and the items, if any, clicked on by the users, as an implicit relevance feedback on the search results. Formally, click through data can be denoted as a triplet $(q; r; c)$, where q is them input query consisting of a set of keywords, r is a list of ranked links, $(l1; : : : ; ln)$, and c is the set of links that the user has clicked on.

The main contributions of this paper are as follows: 1.This paper studies the unique characteristics of contextual and location concepts, and provides a coherent strategy using client-server architecture to integrate them into a uniform solution for the mobile environment. 2.The proposed personalized mobile search engine is an innovative approach for personalizing web search results. 3.By mining contextual and location based concepts for profiling of user, it utilizes both the content and location preferences to personalize search results for a user. Personalization incorporates a user's physical locations in the personalization process.

2. Related Work

Most commercial search engines return roughly the same results to all users. However, different users may have different information needs even for the same query. For example, a user who is looking for a mobile phones may issue a query berry to find products from black berry mobile, while a housewife may use the same query berry fruit to find apple recipes.

2.1. Existing Search Engines

The Internet started of with a directory listing of all the web pages. But as the size of the network and the content hosted on it grew, information retrieval became a challenge. Archie was the first search engine for finding and retrieving computer files. Others being Gopher and Wais. All had the following common characteristics.

- They had a spider which traversed the network and
- Retrieved documents from different servers.
- Built up databases of directories or web pages.
- Ran ranking algorithms

2.2. Existing Location Based Services

Location based services delivered through mobile devices have been a subject of interest in both the academic and application development community for a decade. The central idea is to know the location of the user via GPS/ cell triangulation or explicitly being told by the user and then provide services to the user based on his/her location.

Click through data is important for tracking user actions on a search engine. Many personalized web search systems are based on analyzing users' clickthroughs. The Internet started of with a directory listing of all the web pages. But as the size of the network and the content hosted on it grew, information retrieval became a challenge.

The objective of personalized search is to disambiguate the queries according to the users' interests and to return relevant results to the users. Click through data in search engines can be thought of as triplets (q, r, c) consisting of the query q, the ranking r presented to the user, and the set c of links the user clicked on. Mobile devices, such as mobile phones, and server applications often run under different platforms, which cause an integration problem. We thus proposed an agent communication layers framework, in which agents coordinate web services through several layers. We now further adapt this concept to mobile user agent. In addition, the user ontology is used to provide personalized search of web services for users

Varun Misra at al [2] user behavior data can significantly improve ordering of top results in real web search setting. We examine alternatives for incorporating feedback into the ranking process and explore the contributions of user feedback compared to other common web search features. We report results of a large scale evaluation over 3,000 queries and 12 million user interactions with a popular web search engine. We show that incorporating implicit feedback can augment other features, improving the accuracy of a competitive web search ranking algorithms by as much as 31% relative to the original performance. some of the problems are as follows:

In significant work on merging multiple rankings, we adapt a simple and robust approach of ignoring the original rankers' scores, and instead simply merge the rank orders. The main reason for ignoring the original scores is that since the feature spaces and learning algorithms are different, the scores are not directly comparable, and re-normalization tends to remove the benefit of incorporating classifier scores**.** Solution to the above problem is as follows:

In this paper we explored the utility of incorporating noisy implicit feedback obtained in a real web search setting to improve web search ranking. We performed a large-scale evaluation over 3,000 queries and more than 12 million user interactions with a major search engine, establishing the utility of incorporating "noisy" implicit feedback to improve web search relevance. We compared two alternatives of incorporating implicit feedback into the search process, namely re-ranking with implicit feedback and incorporating implicit feedback features directly into the trained ranking function. Our experiments showed significant improvement over methods that do not consider implicit feedback. The gains are particularly dramatic for the top K=1 result in the final ranking, with precision improvements as high as 31%, and the gains are substantial for all values of K. Our experiments showed that implicit user feedback can further improve web search performance, when incorporated directly with popular content- and link-based features.

Ourdia Boui dhagen, at al [3] In user behavior data on a web search engine can significantly improve the ranking accuracy of top results in real web search engine. A large click log dataset and a set of query URL relevance ags previously labelled by juries was used to train and evaluate a classifier, as well as to build a re-ranking alternative based on a cascade click model. Each one of these implicit feedback based ranking methods can improve the precision of a web search ranking algorithms by as much as 17% compared to the its original rankings.solution to the above problem as follows:

In this paper we explored the utility of incorporating implicit feedback obtained from Yandex search engine to improve its web search ranking. We performed a large-scale evaluation over the click log, using 71.930 manually labelled queries, establishing the utility of incorporating implicit feedback to improve web search relevance. We compared two alternatives of incorporating implicit feedback into the search process, namely re-ranking with implicit feedback and incorporating implicit feedback features directly into the trained ranking function on a ma-chine learning classifiers.

K.W.-T. Leung at al [4] A natural approach is to incorporate implicit feedback features directly as features for the ranking algorithm. During training or tuning, the ranker can be tuned as before but with additional features. At runtime, the search engine would fetch the implicit feedback features associated with each query-result URL pair. This model requires a ranking algorithm to be robust to missing values: more than 50% of queries to web search engines are unique, with no previous implicit feedback available. We now describe such a ranker that we used to learn over the combined feature sets including implicit feedback.

In this user feedback obtained as by tracing user interactions with the search engine. Interpreting implicit feedback in real web search setting is not an easy task. We characterize this problem in detail, where we motivate and evaluate a wide variety of models of implicit user activities. The general approach is to represent user actions for each search result as a vector of features, and then train a ranker on these features to discover feature values indicative of relevant (and non-relevant) search results. We first briefly summarize our features and model, and the learning approach in order to provide sufficient information to replicate our ranking methods and the subsequent experiment

H. Li at al [5] To perform context aware search of tagged data using a tag ontology(Tagged keywords + Context information).As more web services are offered on the Web, it is becoming increasingly difficult for users to manage and search for online content, using only flat keyword searching. Users often forget how they tagged their data but may remember generic information such as the location they were in when they took the picture.GPS locations are not integrated in personalizing the search results

We describe a framework for personalized context-aware search of ontology-based tagged data. The tag ontology leverages additional information on data coming from a user and a resource, besides the tagged keyword, in order to augment search information. The framework uses general concepts taken from PeCMan, a personal content manager and GloServ, an ontology-based global service discovery system to implement the front-end and backend of the overall system.It uses WEB ontology language description logic (OWL DL) to classify services in an ontology.

Xu Yingchen at al [6] AMPR collects users historical preferences dynamically and integrates historical preference with domain Ontology.

GPS Locations are not considered for personalization. Web service is a popular standard to publish services for users. However, diversified users need to access web service according to their particular preferences. Using agent and user ontology seems to provide such badly-needed personalized web service search. Further, mobile agent can be used in the search. However, how to deploy it in a mobile device with limited power and memory is an issue.

we present a mobile user agent architecture, in which a mobile agent migrates from one location server to another to execute a user task. Most importantly, user ontology with both static and dynamic information is defined by web ontology language (OWL), through which the location server can do personalized search of services. In addition, mobile agent reduces network load. Further, it can be dynamically deployed on a location server that connects to a mobile device. This reduces power and memory consumption.

3. Motivation

In mobile IR, users' interests may change anytime due to change in their environment (location, time, near persons, …). Static approaches for building the user profile are therefore poorly useful, so we rather focus on more flexible techniques, any time capable of adjusting the user interests to the current search situation. Our general approach for search personalization relies on building a user profile in a specific search situation.

By mining content and location concepts of user profiling, it utilizes both the content and location preferences to personalize search results for a user. It studies the unique characteristics of content and location concepts, and provides a coherent strategy using a client-server architecture to integrate them into a uniform solution for the mobile environment.

4. System Design

4.1. Architecture:

- Weight Vector: content weight vector and a location weight vector describes the user interests based on the user's content and location preferences extracted from the user clickthroughs, respectively.
- Feature Vector; Feature vector is an n-dimensional vector of numerical features that represent some object Feature vectors are often combined with weights using a dot product in order to construct a linear predictor function that is used to determine a score for making a prediction.

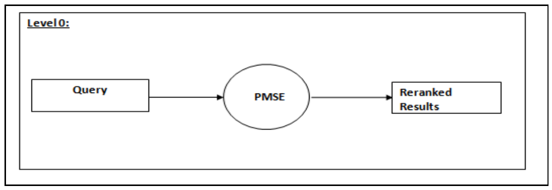

Figure 1

4.2. Modules

4.2.1.Content Ontology

If a keyword/phrase exists frequently in the web-snippets arising from the query, we would treat it as an important concept related to the query.

4.2.2. Location Ontology

Extract location concepts from the full documents. The predefined location ontology is used to associate location information with the search results. All of the keywords and key-phrases from the documents returned for query are extracted. If a keyword or key-phrase in a retrieved document matches a location name in our predefined location ontology, it will be treated as a location concept.

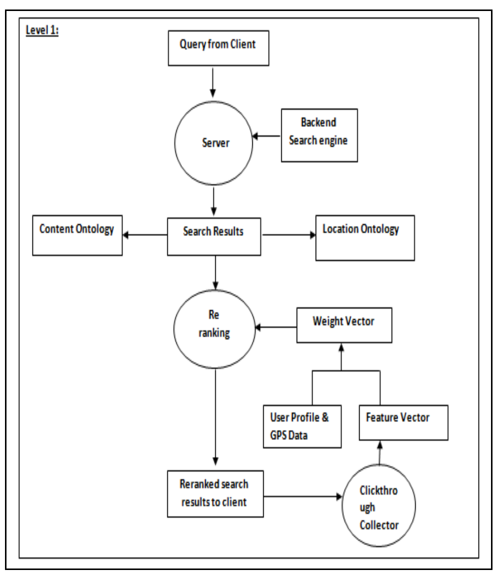

Figure 2

4.2.3. User Interest Profiling

The User Interests from their profiles are extracted to provide more personalized results. More weight is given to the user interests that helps in Reranking based on their interests.

4.2.4. Reranking

The Document preferences obtained from the clicks through collection are served as input to RSVM training to obtain the content weight vector and location weight vector. Location weight vector is incremented for frequently visited locations tracked by GPS. Based on these weight vectors, results are reranked and displayed to the user.

4.2.5. Click through
Clicks are the special case of user interaction with web engine. Click through data in search engines can be thought of as triplets (q, r, c) consisting of the query q, the ranking r presented to the user, and the set c of links the user clicked on.

4.2.6. GPS Data
Users are possibly interested in locations where they have interested. If a user has visited the GPS location l, the weight of the location concept is incremented. Set of location concepts that are closely related to the GPS location in the location ontology are possible candidates indicating user interests.

4.3. Algorithms

4.3.1. Joachims Method
User scans the search result list from top to bottom. If a user skips a document dj at rank j but clicks on document di at rank i where j < i, user must have read dj 's web snippet and decided to skip it. Thus, Joachims method concludes that the user prefers di to document dj.

4.3.2. OMF Profiling
An ontology-based, multi-facet (OMF) user profiling strategy used to capture both of the users' content and location preferences (i.e., .multi-facets.) for building a personalized search engine for mobile users.

4.3.3. RSVM
Ranking SVM is an application of SVM to solve certain ranking problems. It's purpose is to improve the performance of the internet search engine.

4.3.3.1. Input
User Query,User Profiling,Clickthrough collector
Process Steps:
- Maps the similarities between queries and the clicked pages onto certain space.
- Calculates the weights between any two of the vectors obtained in Step 1.
- Reranks the search results based on the weights.

4.3.3.2. Output
Content ontology,location ontology.

4.4. Tools
1.Protege:Protégé is a free, Open source ontology editor and knowledge base framework. Supports creation, visualization, Manipulation of Ontologies.Can be extended by way of a plug-in Architecture and Java bases APIs for building knowledge based applications.

5. Results
If users considered only the relevance of a result to their query,they would click on the topmost relevant results. Unfortunately, as Joachims and others have shown, presentation also influences which results users click on quite dramatically. Users often click on results above the relevant one presumably because the short summaries do not provide enough information to make accurate relevance assessments and they have learned that on average top ranked items are relevant. Figure 3 shows relative clickthrough frequencies for queries with known relevant items at positions other than the first position; the position of the top relevant result (PTR) ranges from 2-10 in the figure. For example, for queries with first relevant result at position 5 (PTR=5), there are more clicks on the non-relevant results in higher ranked positions than on the first relevant result at position 5. As we will see, learning over a richer behavior feature set, results in substantial accuracy improvement over clickthrough alone.
We evaluate the ranking algorithms over a range of accepted information retrieval metrics, namely *Precision at K* (P(K)), *Normalized Discounted Cumulative Gain* (NDCG), and *Mean Average Precision* (MAP). Each metric focuses on a deferent aspect of system performance, as we describe below.
- **Precision at K**: As the most intuitive metric, P(K) reports the fraction of documents ranked in the top K results that are labeled as relevant. In our setting, we require a relevant document to be labeled "Good" or higher. The position of relevant documents within the top K is irrelevant, and hence this metric measure overall user satisfaction with the top K results.
- **NDCG at K:** NDCG is a retrieval measure devised specifically for web search evaluation . For a given query *q*, the ranked results are examined from the top ranked down
- **MAP:** Average precision for each query is defined as the mean of the precision at K values computed after each relevant document was retrieved. The final MAP value is defined as the mean of average precisions of all queries in the test set. This metric is the most commonly used single-value summary of a run over a set of queries.
- Recall that our goal is to quantify the effectiveness of implicit behavior for real web search. One dimension is to compare the utility of implicit feedback with other information available to a web search engine. Specifically, we

compare effectiveness of implicit user behaviors with content-based matching, static page quality features, and combinations of all features.

- **BM25F:** As a strong web search baseline we used the BM25F scoring, which was used in one of the best performing systems in the TREC 2004 Web track. BM25F and its variants have been extensively described and evaluated in IR literature, and hence serve as a strong, reproducible baseline. The BM25F variant we used for our experiments computes separate match scores for each “field” for a result document (e.g., body text, title, and anchor text), and incorporates query-independent linkbased information (e.g., PageRank, ClickDistance, and URL depth). The scoring function and field-specific tuning is described in detail in . Note that BM25F does not directly consider explicit or implicit feedback for tuning.
- **RN:** The ranking produced by a neural net ranker that *learns* to rank web search results by incorporating BM25F and a large number of additional static and dynamic features describing each search result. This system automatically learns weights for all features (including the BM25F score for a document) based on *explicit* human labels for a large set of queries. A system incorporating an implementation of RankNet is currently in use by a major search engine and can be considered representative of the state of the art in web search.
- **BM25F-RerankCT**: The ranking produced by incorporating clickthrough statistics to reorder web search results ranked by BM25F above. Clickthrough is a particularly important special case of implicit feedback, and has been shown to correlate with result relevance. This is a special case of the ranking method with the weight ***wI*** set to 1000 and the ranking ***Id*** is simply the number of clicks on the result corresponding to ***d***. In effect, this ranking brings to the top all returned web search results with at least one click (and orders them in decreasing order by number of clicks). The relative ranking of the remainder of results is unchanged and they are inserted below all clicked results. This method serves as our baseline implicit feedback reranking method.
- **BM25F-RerankAll** The ranking produced by reordering the BM25F results using *all* user behavior features .This method learns a model of user preferences by correlating feature values with explicit relevance labels using the RankNet
- neural net algorithm (Section 4.2). At runtime, for a given query the implicit score ***Ir*** is computed for each result ***r*** with available user interaction features, and the implicit ranking is produced. The merged ranking is computed. Based on the experiments over the development set we fix the value of ***wI*** to 3 (the effect of the w***I*** parameter for this ranker turned out to be negligible).
- **BM25F+All:** Ranking derived by training the RankNet (Section 3.3) learner over the features set of the BM25F score as well as all implicit feedback features . We used the 2-layer implementation of RankNet [5] trained on the queries and labels in the training and validation sets.
- **RN+All:** Ranking derived by training the 2-layer RankNet ranking algorithm over the union of all content, dynamic, and implicit feedback features (i.e., all of the features described above as well as all of the new implicit feedback features we introduced).The ranking methods above span the range of the information used for ranking, from not using the implicit or explicit feedback at all (i.e., BM25F) to a modern web search engine using hundreds of features and tuned on explicit judgments (RN). As we will show next, incorporating user behavior into these ranking systems dramatically improves the relevance of the returned documents.

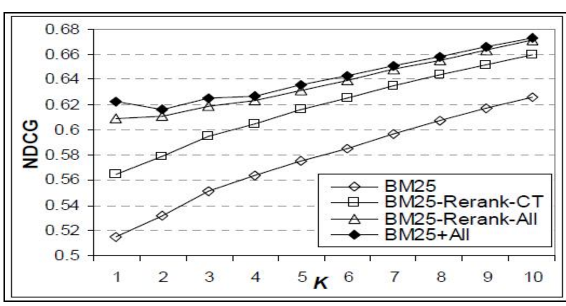

Figure 3

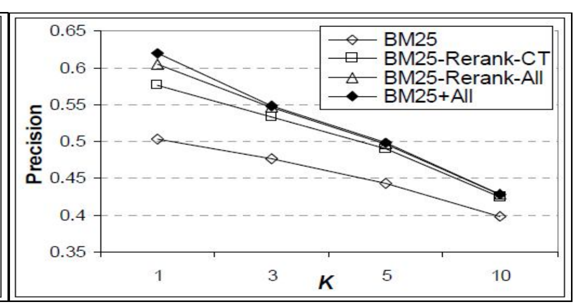

Figure 4

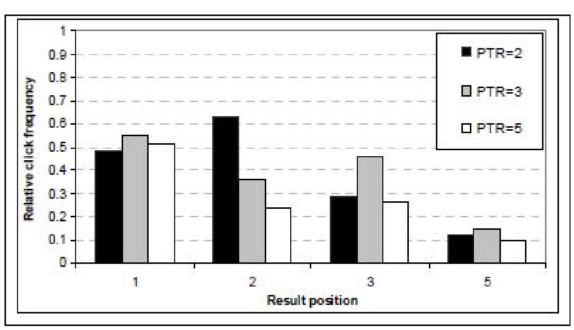

Figure 5

6. Discussion

The experimental results show that the Ranking SVM can successfully learn an improved retrieval function from clickthrough data. Without any explicit feedback or manual parameter tuning, it has automatically adapted to the particular preferences of a group of _ 20 users. This improvement is not only a verification that the Ranking SVM can learn using partial ranking feedback, but also an argument for personalizing retrieval functions. Unlike conventional search engines that have to “fit” their retrieval function to large and therefore heterogeneous groups of users due to the cost of manual tuning, machine learning techniques can improve retrieval substantially by tailoring the retrieval function to small and homogenous groups (or even individuals) without prohibitive costs.

7. Merits And Demerits

7.1. Demerits

- In an existing system, GPS location is in some difficulties.
- Some obstacles in the privacy.
- Most commercial search engines return roughly the same results to all users. However, different users may have different information needs even for the same query.

7.2. Merits

- This paper 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 proposed personalized mobile search engine is an innovative approach for personalizing web search results. By mining content and location concepts for user profiling, it utilizes both the content and location preferences to personalize search results for a user.
- The results show that GPS location helps improve retrieval effectiveness for location queries (i.e., queries that retrieve lots of location information).
- Our design adopts the server-client model in which user queries are forwarded to a PMSE server for processing the training and reranking quickly.
- We implement a working prototype of the PMSE clients on the Google Android platform, and the PMSE server on a PC to validate the proposed ideas.
- PMSE addresses the privacy issue by allowing users to control their privacy levels with two privacy parameters, minDistance and expRatio.

8. Conclusion

Personalization search engine extract the user preferences on both content and location based on the user click through data .To become accustomed to the user mobility, it also included the user’s GPS locations in the personalization procedure to examine the location and help to increase retrieval efficiency, mostly for location queries. Query patterns scheme contribute new information which gather more and more on folder to suit the user profiles consequences novel user is searching for travel information on mobile devices, the scheme determination study user performance transaction which user clicks.The scheme determinations to gather new data and examine them then interpret to user. The scheme will motivation to learn increasingly whilst numerous of users click further on mobile request. It will accumulate additional data and repeatedly examine the recently obtained data. If the travelers’ behavior changes, the pattern in database also change. The scheme will work more precise and work efficiently all along with the dynamics of the result.

9. Future Work

In addition through the huge expansion of the information obtainable on the Web, it is very complex for Web search engines to assure the user information obligation only by means of a short vague query. Dissimilar Query based results dissimilar on or after every user, query based recommendation system determination aid user to discover the vague query .In future work comprise the query based recommendation procedure to identify the user alike queries and their results

10. References

1. Kenneth Wai-Ting Leung,Dik Lun Lee and Wang-Chien Lee,”Personalized Mobile Search Engine”,IEEE Transaction on Knowledge and Data Engineering,2013
2. Varun Misra, Premarayan Arya, Manish Dixit, “Improving Mobile Search through Location Bases context and personalization”, IEEE Transactions on Communication Systems and Network Technologies, 2012
3. Ourdia Boui dhagen, Lynda Taime, Mohand Boughanem, “Coext-Aware user’s Interest for personalizing Mobile Search”, IEEE Transactions on Knowledge and Data Engineering, 2011
4. K.W.-T. Leung, D.L. Lee, and W.-C. Lee, “Personalized Web Search with Location Preferences,” IEEE Transactions on Data Mining , 2010
5. K. Arabshian, E. Brill, “A framework for Personalized Context –Aware search of Ontology based tagged data, IEEE Transactions on Services Computing, 2010.

6. H. Li, Z. Li, W.-C. Lee, and D.L. Lee, "A Probabilistic Topic-Based Ranking Framework for Location-Sensitive Domain Information Retrieval," Proc. International ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR), 2009.
7. Xu Yingchen, Yang Jing, Gu Junzhung, "An Ontology Based Approach for Mobile Personalized Recommendation", IEEE Transactions on Services Computing, 2009,
8. K.W.-T. Leung, W. Ng, and D.L. Lee, "Personalized Concept-Based Clustering of Search Engine Queries," IEEE Transaction Knowledge and Data Eng., vol. 20, no. 11, Nov. 2008.
9. Q. Gan, J. Attenberg, A. Markowetz, and T. Suel, "Analysis of Geographic Queries in a Search Engine Log," IEEE Transactions on Mobile Management, 2008