ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

Enhancing Privacy Search in Web Search Engine Using Greedy Algorithm

N.Sumathi[1], V.Karuppuchamy[2]

P.G. Student, Dept of Computer Engineering, Muthayammal Engineering College, Rasipuram, India[1]

Assistant Professor, Dept of Computer Engineering, Muthayammal Engineering College, Rasipuram, India[2]

ABSTRACT: In web search, there are many techniques used extensively for effective retrieval of information from the web server. A user query entered into the search engine may return large number of web page results and thus, it becomes extremely important to rank these results in such a manner that it returns accurate and more relevant web results. These tasks of prioritizing the results are performed by ranking algorithms. But the search interest of every user differs with every other user uniquely. In existing system, the search engine that return results for a given user query is not uniquely based on their earlier searching behavior. Hence there may be return of web page results that are least used or accessed by the user. These results need to be filtered or less prioritized by the ranking algorithm. In our proposed system, the personalization of user search activities is taken to find search interest of each user. This personalization of user search behavior is done by storing user activities and analyzing web log by clustering frequently accessed tasks by semantic task clustering algorithm and dynamically ranks web results. In this project we can also extend our work include web image search with personalization and remove duplication keywords. It can be achieved using image features approach and top k answering techniques. Implement this approach in real time search engine environment.

I.INTRODUCTION

Data mining (the analysis step of the "Knowledge Discovery in Databases" process, or KDD), a field at the intersection of computer science and statistics, is the process that attempts to discover patterns in large data sets. It utilizes methods at the intersection of artificial intelligence, machine learning, statistics and database system. The overall goal of the data mining process is to extract information from a data set and transform it into an understandable structure for further use aside from the raw analysis step, it involves database and data management aspects, data processing, model and interface consideration, interestingness metrics, complexity consideration, post-processing of discovered structure, visualization and online updating.

Generally, data mining (sometimes called data or knowledge discovery) is the process of analyzing data from different perspectives and summarizing it into useful information that can be used to increase revenue, cuts costs, or both. Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases.

II.EXISTING SYSTEM

Many existing search engines use keyword matching as the search mechanism, which usually causes the situation that a large number of non relevant documents containing query terms are founded out, and the user will make strenuous efforts to browse these non-relevant documents. Moreover, the average length of queries is about two words. Thus, it is not simple to find out the real user goal from such short queries.

To achieve the user specific information needs many uncertain queries may cover a broad topic and dissimilar users may want to get information on different point of view when they submit the same query. User information need is to desire and obtain the information to satisfy the needs of each user. To satisfy the user information needs by considering the search goals with user given query. We cluster the user information needs with different search goal

ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

.Because the interference and examination of user search goals with query might have a numeral of advantages by improving the search engine significance and user knowledge. So it is necessary to collect the different user goal and retrieve the efficient information on different aspects of a query. Capture different user search goals in information retrieval outcome becomes changes than the normal query based information retrieval.Personalized web search faces several challenges that retard its real-world large-scale applications:

1. It is really hard to infer user information needs accurately. Users are not static. They may randomly search for something which they are not interested in. They even search for other people sometimes. User search histories inevitably contain noise that is irrelevant or even harmful to current search. This may make personalization strategies unstable.
2. Queries should not be handled in the same manner with regard to personalization. Personalized search may have little effect on some queries. Some work investigates whether current web search ranking might be sufficient for clear/unambiguous queries and thus personalization is unnecessary. A specific personalized search also has different effectiveness for different queries. It even hurts search accuracy under some situations.

Disadvantages:

- The existing methods do not take into account the customization of user requirements.
- Many personalization techniques require iterative user interactions when creating personalized search results.
-

III.PROPOSED SYSTEM

Personalized search is a promising way to improve the accuracy of web search, and has been attracting much attention recently. However, effective personalized search requires collecting and aggregating user information, which often raises serious concerns of privacy infringement for many users. Indeed, these concerns have become one of the main barriers for deploying personalized search applications, and how to do personalization is a great challenge. For a given query, a personalized Web search can provide different search results for different users or organize search results differently for each user, based upon their interests, preferences, and information needs. Personalized web search differs from generic web search, which returns identical research results to all users for identical queries, regardless of varied user interests and information needs. We provide an inexpensive mechanism for the client to decide whether to personalize a query in data base system. This decision can be made before each runtime profiling to enhance the stability of the search results while avoid the unnecessary exposure of the profile. To provide personalized search results to users, personalized web search maintains a user profile for each individual. A user profile stores approximations of user tastes, interests and preferences. It is generated and updated by exploiting user-related information. Such information may include:

1. Demographic and geographical information, including age, gender, education, language, country, address, interest areas, and other information;
2. Search history, including previous queries and clicked documents. User browsing behavior when viewing a page, such as dwelling time, mouse click, mouse movement, scrolling, printing, and bookmarking, is another important element of user interest.
3. Other user documents, such as bookmarks, favorite web sites, visited pages, and emails. A user profile can usually aggregate a user's history information and represent the user's long-term interests (information needs). Some work has investigated whether such a long-term user profile is ineffective in some cases. Consider the second case that was described in the historical background section: a user will have different needs at different times based on circumstances. In such situations, personalization based on a user's long-term interests may not provide a satisfying performance, because similar results could be returned. Some work has considered the use of a user's active context to represent short-term information needs. Search context is incorporated into the user profile, or is constructed as a separate short-term user model/profile and is used in helping infer a user's information needs. Our extensive experiments demonstrate the efficiency and effectiveness of our UPS framework. In personalized web services, each user profile in database adopts a hierarchical structure. Moreover, our profile is constructed based on the availability of a public accessible taxonomy.

ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

Advantages:

- Generalize profiles for each query according to user-specified privacy requirements.
- We can minimize the information loss in personalized web search system.
- Providing protection against a typical model of privacy attack, namely eavesdropping

Search query:

web search query is a query that a user enters into a web search engine to satisfy his or her information needs. Web search queries are distinctive in that they are often plain text or hypertext with optional search-directives (such as "and"/"or" with "-" to exclude). They vary greatly from standard query languages, which are governed by strict syntax rules as command languages with keyword or positional parameters. A search query, the actual word or string of words that a search engine user types into the search box, is the real-world application of a keyword – it may be misspelled, out of order or have other words tacked on to it, or conversely it might be identical to the keyword. There are three broad categories that cover most web search queries: informational, navigational, and transactional. These are often called "do, know, go." Informational queries – Queries that cover a broad topic (e.g., colorado or trucks) for which there may be thousands of relevant results. Navigational queries – Queries that seek a single website or web page of a single entity (e.g., youtube or delta air lines). Transactional queries – Queries that reflect the intent of the user to perform a particular action, like purchasing a car or downloading a screen saver. Search engines often support a fourth type of query that is used far less frequently: Connectivity queries – Queries that report on the connectivity of the indexed web graph (e.g., Which links point to this URL?, and How many pages are indexed from this domain name?).

Relevant web documents:

This module we implement the key component for privacy protection is an online profiler implemented as a search proxy running on the client machine itself. The proxy maintains both the complete user profile, in a hierarchy of nodes with semantics, and the user-specified (customized) privacy requirements represented as a set of sensitive-nodes. Then proposed the prototype of UPS, together with a greedy algorithm GreedyDP (named as GreedyUtility to support online profiling based on predictive metrics of personalization utility and privacy risk. Greedy algorithm GreedyDP works in a bottomup manner. The main problem of GreedyDP is that it requires recomputation of all candidate profiles (together with their discriminating power and privacy risk) generated from attempts of prune-leaf manner. And we proposed a new profile generalization algorithm called GreedyIL. The GreedyIL algorithm improves the efficiency of the generalization using heuristics based on several findings. One important finding is that any prune-leaf operation reduces the discriminating power of the profile. In other words, the DP displays monotonicity by prune-leaf. GreedyIL further reduces this measure with Heuristic . The greater the privacy threshold, the fewer iterations the algorithm requires.

Duplication Top k answers

In this module, we have proposed the prototype of search engine, together with a greedy algorithm named as GreedyUtility to support online profiling based on predictive metrics of personalization utility. Greedy algorithm Greedy algorithm works in a bottomup manner. The main problem of Greedy is that it requires recomputation of all candidate profiles (together with their discriminating power and privacy risk) generated from attempts of prune-leaf manner. In this module, we can create taxonomy repository in tree structure. D represents the collection of all personal documents and each document is treated as a list of terms. D(t) denotes all documents covered by term t, i.e., all documents in which t appears, and $|D(t)|$ represents the number of documents covered by t. A term t is frequent if $|D(t)| \geq$ minsup, where minsup is a user-specified threshold, which represents the minimum number of documents in which a frequent term is required to occur. Each frequent term indicates a possible user interest. In order to organize all the frequent terms into a hierarchical structure, relationships between the frequent terms are defined below. Assuming two terms tA and tB., the two heuristic rules used in our approach are summarized as follows: 1. Similar terms: Two terms that cover the document sets with heavy overlaps might indicate the same interest. Here we use the Jaccard function to calculate the similarity between two terms: $Sim(tA, tB) = | D(tA) \cap D(tB) | / | D(tA) \cup D(tB) |$. If $Sim(tA , tB) > \delta$, where δ is another user-specified threshold, we take tA and tB as similar terms representing the same interest. 2. Parent-Child terms: Specific terms often appear together with general terms, but the reverse is not true. Thus, tB is taken as a child term of tA if the condition probability $P(tA | tB) > \delta$, where δ is the same threshold in Rule 1.

ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

Best answers:

The best answer in each subspace is found and used to produce the current best global answer. The sub-space that produces the best global answer is further divided into sub-subspaces and the best answer among its sub-subspaces is used to compete with the best answers in other sub-spaces in the previous level to find the next best global answer. Two main issues in this procedure are how to divide a space into subspaces and how to find the best answer within a (sub) space. To have duplication free answers, the procedure for dividing the search space into sub-spaces must produce **disjoint** sub-spaces so that the same answer cannot be generated from different sub-spaces.

Search results:

It is important to monitor the personalization utility during the generalization. Using the running example, profiles Ga and Gb might be generalized to smaller rooted subtrees. However, overgeneralization may cause ambiguity in the personalization, and eventually lead to rich search results. Monitoring the utility would be possible only if we perform the generalization at runtime. In this module, we improve the results based on search experience and profession based results. Our experimental results provide the efficient privacy based search results.

IV.IMPLEMENTATION

- **Profile Based approach**
- **Greedy DP algorithm:**

A greedy algorithm is an algorithm that follows the problem solving heuristic of making the locally optimal choice at each stage with the hope of finding a global optimum. In many problems, a greedy strategy does not in general produce an optimal solution, but nonetheless a greedy heuristic may yield locally optimal solutions that approximate a global optimal solution in a reasonable time.

A greedy algorithm is a mathematical process that recursively constructs a set Recursion of objects from the smallest possible constituent parts. is an approach to problem solving in which the solution to a particular problem depends on solutions to smaller instances of the same problem. Greedy algorithms look for simple, easy-to-implement solutions to complex, multi-step problems by deciding which next step will provide the most obvious benefit. Such algorithms are called greedy because while the optimal solution to each smaller instance will provide an immediate output, the algorithm doesn't consider the larger problem as a whole. Once a decision has been made, it is never reconsidered.

The advantage to using a greedy algorithm is that solutions to smaller instances of the problem can be straightforward and easy to understand. The disadvantage is that it is entirely possible that the most optimal short-term solutions may lead to the worst long-term outcome. In this project we proposed GreedyDP algorithm works in bottom up manner. Then graph starts with parent node to choose leaf topic and trying to maximize the utility of the current iteration, during the iterations, we also maintain a best profile-so-far, which indicates the graph having the highest discriminating power while satisfying the risk constraint. The iterative process terminates when the profile is generalized to a root-topic. The best-profile-so-far will be the final result of the algorithm

Greedy IL algorithm:

The GreedyIL algorithm provides the efficiency of the generalization using heuristics algorithms and performs prune leaf operation to reduce the risks. GreedyIL provide training by simulating the profile construction process at an intermediate time, with the set of candidate keywords collected over the preceding time interval yielding the set of training instances. True labels (actual utilities) are obtained based on the presence of the keyword candidate in subsequent behavior. The utility is 0 if the keyword was not matched to subsequent search contexts, or if it was matched and there were no ad clicks. If the keyword was matched to a future query leading to a corresponding ad impression, and a click was observed, utility is the corresponding bid increment value. The learning algorithm then attempts to identify the predictor parameters that minimize expected error (typically, by minimizing training set error under some regularization technique that prevents over fitting).

ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

Top-K answering algorithm:

Algorithm 1 Generate Duplication Free Top-k *Answers*

Input: the input graph G; the query $Q = \{k_1, k_2, \ldots, k_l\}$; k
Output: the set of top-k ordered *Answers* printed with polynomial delay

```
1: C ← an empty set for storing content nodes
2: for i ← 1 to l do
3:     add the nodes in G containing k_i to C
4: Queue ← an empty priority queue
5: A ← FindBestAnswer(G, Q, C, ∅, ∅)
6: if A ≠ NULL then
7:     insert ⟨A, ∅, ∅⟩ into Queue
8: while Queue ≠ ∅ do
9:     ⟨A, Inc, Exc⟩ ← top element of Queue
10:    print(A)
11:    k ← k − 1
12:    if k = 0 then
13:        return
14:    {n_1, n_2, ..., n_p} ← content nodes of A
15:    for i ← 1 to p do
16:        Inc_i ← Inc ∪ {n_1, ..., n_{p−i}}
17:        Exc_i ← Exc ∪ {n_{p−i+1}}
18:        if Inc_i ∩ Exc_i = ∅ then
19:            A_i ← FindBestAnswer(G, Q, C, Inc_i, Exc_i)
20:            if A_i ≠ NULL then
21:                insert ⟨A_i, Inc_i, Exc_i⟩ into the right place of Queue
                   according to A_i's weight
```

Algorithm is a general framework for generating top-k duplication-free answers by wisely dividing the search space. It calls Algorithm 2. The algorithm first computes the set C of nodes that contain at least one input keyword. This can be easily done using a pre-built inverted index. In line 5, procedure FindBestAnswer is called to find the best answer from the whole search space (i.e., C). It takes input graph G, query Q, C, an inclusion set and exclusion set as input, and returns the best answer in the search space specified by C. and takes an input graph G, a query Q, a set of content nodes C, and the inclusion and exclusion sets (Inc and Exc) as input and produces the best (approximate) Answer as output in polynomial time. The algorithm approximates the sumDistance of an answer using the sum of distances from each node in the answer to a center node within the answer. In the pseudo-code, set F is the search space, which consists of all the nodes in the inclusion set and the set of content nodes containing the query keywords not covered by the inclusion set and not belonging to the exclusion set. In the code, Di is the set of nodes that contain keyword ki (which is not covered by the inclusion set) but do not belong to the exclusion set. Then checks each node in the Answer to see if the input keywords the node contains are all covered by other nodes. If yes, it removes the node. The complexity of this algorithm is O (n^2) where n is the number of nodes in the input Answer. Algorithm 3 produces a minAnswer in which each node contains at least one unique input keyword. In addition, all the input keywords are covered in the minAnswer. And uses a greedy set-covering procedure to reduce the number of nodes in Answer while still covering all the input keywords. The procedure chooses nodes to form an answer A as follows: at each stage, choose the node that contains the largest number of uncovered keywords. However, A may not be a minAnswer because the above procedure is a greedy procedure for minimizing the number of nodes.

V. RESULTS

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find

ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

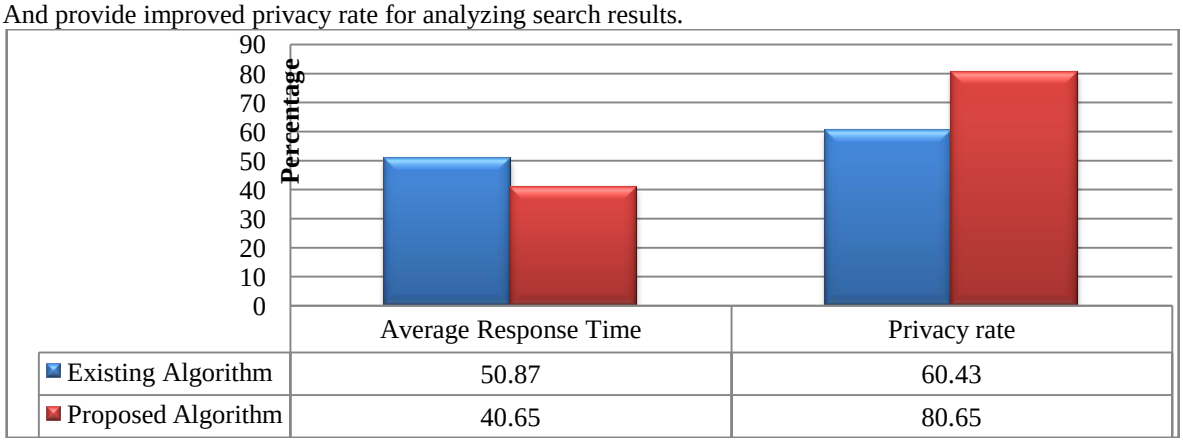
Vol. 4, Special Issue 6, May 2015

the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

- Select methods for presenting information.
- Create document, report, or other formats that contain information produced by the system.

Evaluation results:

- Our algorithm provide less response time for retrieving search results.
- And provide improved privacy rate for analyzing search results.



	Average Response Time	Privacy rate
Existing Algorithm	50.87	60.43
Proposed Algorithm	40.65	80.65

V.CONCLUSION

A major problem in mobile search is that the interactions between the users and search engines are limited by the small form factors of the search engines. As a result, web users tend to submit shorter, hence, more ambiguous queries compared to their web search counterparts. We proposed PWS to extract and learn a user's history and content preferences based on the user's profiles. To adapt to the user mobility, we incorporated the user's details in the personalization process. We observed that profile help to improve retrieval effectiveness, especially for search queries. We also proposed privacy parameters, DP and IL, to address privacy issues in PWS by allowing users to control the amount of personal information exposed to the web server. The privacy parameters facilitate smooth control of privacy exposure while maintaining good ranking quality.

REFERENCES

[1] K. Sugiyama, K. Hatano, and M. Yoshikawa, "Adaptive Web Search Based on User Profile Constructed without any Effort from Users," Proc. 13th Int'l Conf. World Wide Web (WWW), 2004.
[2] B. Tan, X. Shen, and C. Zhai, "Mining Long-Term Search History to Improve Search Accuracy," Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD), 2006.
[3] J. Teevan, S.T. Dumais, and E. Horvitz, "Personalizing Search via Automated Analysis of Interests and Activities," Proc. 28th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR), pp. 449-456, 2005.
[4] Z. Dou, R. Song, and J.-R. Wen, "A Large-Scale Evaluation and Analysis of Personalized Search Strategies," Proc. Int'l Conf. World Wide Web (WWW), pp. 581-590, 2007.
[5] Nicolaas Matthijs, "Personalizing Web Search using Long Term Browsing History" ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR), 2011.
[6] M. Spertta and S. Gach, "Personalizing Search Based on User Search Histories," Proc. IEEE/WIC/ACM Int'l Conf. Web Intelligence (WI), 2005.
[7] X. Shen, B. Tan, and C. Zhai, "Implicit User Modeling for Personalized Search," Proc. 14th ACM Int'l Conf. Information and Knowledge Management (CIKM), 2005.
[8] X. Shen, B. Tan, and C. Zhai, "Context-Sensitive Information Retrieval Using Implicit Feedback," Proc. 28th Ann. Int'l ACM SIGIR Conf. Research and Development Information Retrieval (SIGIR), 2005.
[9] F. Qiu and J. Cho, "Automatic Identification of User Interest for Personalized Search," Proc. 15th Int'l Conf. World Wide Web (WWW), pp. 727-736, 2006.
[10] J. Pitkow, H. Schu¨ tze, T. Cass, R. Cooley, D. Turnbull, A. Edmonds, E. Adar, and T. Breuel, "Personalized Search," Comm. ACM, vol. 45, no. 9, pp. 50-55, 2002.