ISSN:2348-2079

Volume-6 Issue-2

International Journal of Intellectual Advancements and Research in Engineering Computations

Privacy Protected Personalized Web Search using Cache Model

[1]Mrs. K.E.Eswari, M.C.A., M.Phil.,M..E., Associate Professor/MCA
[2]Ms. S. VishwaPriya, III MCA
Department of MCA, Nandha Engineering College(Autonomous), Erode-52.
E-Mail ID: eswari.eswaramoorthy@nandhaengg.org , riyagi23794@gmail.com

***Abstract-* The web search engine has long become the most important portal for ordinary people looking for useful information on the web. However, users might experience failure when search engines return irrelevant results that do not meet their real intentions. Such irrelevance is largely due to the enormous variety of users' contexts and backgrounds, as well as the ambiguity of texts. Personalized web search (PWS) is a general category of search techniques aiming at providing better search results, which are tailored for individual user needs. As the expense, user information has to be collected and analyzed to figure out the user intention behind the issued query. Personalized web search (PWS) has demonstrated its effectiveness in improving the quality of various search services on the Internet. However, evidences show that users' reluctance to disclose their private information during search has become a major barrier for the wide proliferation of PWS. This research studies privacy protection in PWS applications that model user preferences as hierarchical user profiles. This project proposes a PWS framework called UPS that can adaptively generalize profiles by queries while respecting user-specified privacy requirements. The proposed runtime generalization aims at striking a balance between two predictive metrics that evaluate the utility of personalization and the privacy risk of exposing the generalized profile. The study presents two greedy algorithms, namely Greedy DP and Greedy IL, for runtime generalization. It also provides an online prediction mechanism for deciding whether personalizing a query is beneficial.**

Index term-Privacy Protection, Personalized web-search

I. INTRODUCTION

Data mining, the extraction of hidden predictive information from large databases, is a powerful new technology with great potential to help companies focus on the most important information in their data warehouses. Data mining tools predict future trends and behaviors, allowing businesses to make proactive, knowledge-driven decisions.The automated, prospective analyses offered by data mining move beyond the analyses of past events provided by retrospective tools typical of decision support systems.

Data mining tools can answer business questions that traditionally were too time consuming to resolve. They scour databases for hidden patterns, finding predictive information that experts may miss because it lies outside their expectations. Most companies already collect and refine massive quantities of data. Data mining techniques can be implemented rapidly on existing software and hardware platforms to enhance the value of existing information resources, and can be integrated with new products and systems as they are brought on-line.The white paper provides an introduction to the basic technologies of data mining. Examples of profitable applications illustrate its relevance to today's business environment as well as a basic description of how data warehouse architectures can evolve to deliver the value of data mining to end users.

II. RELATED WORKS

Zhicheng Dou, Ruihua Song and Ji-Rong Wen [1] studies this problem and provides some preliminary conclusions. They present a large-scale evaluation framework for personalized search based on query logs, and then evaluate five personalized search strategies (including two click-based and three profile-based ones) using MSN query logs. By analyzing the results, they reveal that personalized search has significant improvement over common web search on some queries but it has little effect on other queries (e.g., queries with small click entropy). It even harms search accuracy under some situations. Furthermore, they show that straight- forward click-based personalization strategies perform consistently and considerably well, while profile-based ones are unstable in their experiments. They also reveal that both long- term and short-term contexts are very important in improving search performance for profile-based personalized search strategies.

One criticism of search engines is that when queries are issued, most return the same results to users. In fact, the vast majority of queries to search engines are short and ambiguous and different users may have completely different information needs and goals under the same query. Personalized search is considered a solution to this problem since different search results based on preferences of users are provided. Various personalization strategies including have been proposed, and personalized web search systems have been developed, but they are far from optimal. One problem of current personalized search is that most proposed methods are uniformly applied to all users and queries. In fact, they think that queries should not be handled in the same manner because they find:

- Personalization may lack effectiveness on some queries, and there is no need for personalization on such queries.
- Different strategies may have variant effects on different queries. In such a case, simply leveraging pages visited by this user in the past may achieve better performance.
- Personalization strategies may provide different effectiveness based on different search histories and under variant contexts. Furthermore, as Shen et al. [5] noted, users often search for documents to satisfy short-term information needs, which may be inconsistent with general user interests. In such cases, long-term user profiles may be useless and short-term query context may be more useful.

MircoSperetta [6] describe to creating user profiles collect user information through proxy servers (to capture browsing histories) or desktop bots (to capture activities on a personal computer). Both these techniques require participation of the user to install the proxy server or the bot. In this study, they explore the use of a less-invasive means of gathering user information for personalized search. In particular, they build user profiles based on activity at the search site itself and study the use of these profiles to provide personalized search results. By implementing a wrapper around the Google search engine, they were able to collect information about individual user search activities.

In particular, they collected the queries for which at least one search result was examined, and the snippets (titles and summaries) for each examined result. User profiles were created by classifying the collected information (queries or snippets) into concepts in a reference concept hierarchy. These profiles were then used to re-rank the search results and the rank-order of the user-examined results before and after re-ranking were compared. Their study found that user profiles based on queries were as effective as those based on snippets. They also found that their personalized re-ranking resulted in a 34% improvement in the rank-order of the user-selected results.Many approaches create user profiles by capturing browsing histories through proxy servers or desktop activities through the installation of bots on a personal computer. These require the participation of the user in order to install the proxy server or the bot.

In this study, they explore the use of a less-invasive means of gathering user information for personalized search. Their goal is to show that user profiles can be implicitly created out of short phrases such as queries and snippets collected by the search engine itself. They demonstrate that profiles created from this information can be used to identify, and promote, relevant results for individual users. In general, personalization can be applied to search in two different ways. To providing tools that help users organizing their own past searches, preferences, and visited URLs. To creating and maintaining sets of user's interests, stored in profiles, that can be used by retrieval process of a search engine to provide better results.The first approach is applied by many new toolbars and browser add-ons. The Seruku Toolbar and the Surf Saver are examples of tools that try to help users to organize their search histories in a repository of URLs and web pages visited. Furl is another personalization tool that stores web pages including topics which users are interested in, however it was developed as a server-side technology rather than a desktop toolbar.

Bin Tan, Xuehua Shen, ChengXiangZhai [4]describes theLong-term search history contains rich information about a user’s search preferences, which can be used as search context to improve retrieval performance. In this paper, they study statistical language modeling based methods to mine contextual information from long term search history and exploit it for a more accurate estimate of the query language model. Experiments on real web search data show that the algorithms are effective in improving search accuracy for both fresh and recurring queries. The best performance is achieved when using click through data of past searches that are related to the current query.

In this paper, authors systematically study how to exploit a user’s long-term search history to improve retrieval accuracy. They propose mixture models to represent a user’s information need and apply statistical language modeling techniques to discover relevant context from the search history, and exploit it to obtain improved estimates of the query model. They then evaluate the methods on a test set of Web search histories collected from some real users. They find that mined search history information, can substantially improve retrieval performance for both recurring and fresh queries, and works best when click through data is used with a discriminative weighting scheme for past searches. They also find that although recent history tends to be much more useful than remote history (especially for fresh queries), all of the entire history is helpful for improving the search accuracy of recurring queries.

In this paper, authors systematically explored how to exploit long-term search history, which consists of past queries, result documents and click through, as useful search context that can improve retrieval performance. They emphasized the importance of discriminative use of search history, by concentrating on the most relevant past queries. They cast the search history mining problem as estimating a more accurate query model from evidence in the search history, and developed methods based on statistical language modeling for this task. They collected real web search data as their test set and shown in their experiments that the contextual methods can effectively improve search accuracy over the traditional, context less method for both fresh and recurring queries, with the EM-based discriminative weighting scheme achieving best performance. They also found through their study of different cutoffs in search history that although recent history is more important, remote history is also useful, especially for recurring queries.

Kazunari Sugiyama, Kenji Hatano and Masatoshi Yoshikawa[6]Web search engines help users find useful information on the World Wide Web WWW). However, when the same query is submitted by different users, typical search engines return the same result regardless of who submitted the query. Generally, each user has different information needs for his/her query. Therefore, the search results should be adapted to users with different information needs. In this paper, authors first propose several approaches to adapting search results according to each user's need for relevant information with-out any user effort, and then verify the effectiveness of their proposed approaches. Experimental results show that search systems that adapt to each user's preferences can be achieved by constructing user profiles based on modified collaborative filtering with detailed analysis of user's browsing history in one day.

Xuehua Shen, Bin Tan and ChengXiangZhai[3]describe the information retrieval systems and to exploit such immediate and short-term search context to improve search has so far not been well addressed in the previous work. In this paper, they study how to construct and update a user model based on the immediate search context and implicit feedback information and use the model to improve the accuracy of ad hoc retrieval. In order to maximally benefit the user of a retrieval system through implicit user modeling, they propose to perform "eager implicit feedback".

That is, as soon as they observe any new piece of evidence from the user, they would update the system's belief about the user's information need and respond with improved retrieval results based on the updated user model.

III. SYSTEM METHODOLOGY

A. PROFILE-BASED PERSONALIZATION

Previous works on profile-based PWS mainly focus on improving the search utility. The basic idea of these works is to tailor the search results by referring to, often implicitly, a user profile that reveals an individual information goal. In the remainder of this section, we review the previous solutions to PWS on two aspects, namely the representation of profiles, and the measure of the effectiveness of personalization. Many profile representations are available in the literature to facilitate different personalization strategies. Earlier techniques utilize term lists/vectors or bag of words to represent their profile.

B. PRIVACY PROTECTION IN PWS SYSTEM

Generally there are two classes of privacy protection problems for PWS. One class includes those treat privacy as the identification of an individual, as described. The other includes those consider the sensitivity of the data, particularly the user profiles, exposed to the PWS server.

Typical works in the literature of protecting user identifications (class one) try to solve the privacy problem on different levels, including the pseudo identity, the group identity, no identity, and no personal information. Solution to the first level is proved to fragile.

The third and fourth levels are impractical due to high cost in communication and cryptography. Therefore, the existing efforts focus on the second level. Both provide online anonymity on user profiles by generating a group profile of k users. Using this approach, the linkage between the query and a single user is broken.

In the useless user profile (UUP) protocol is proposed to shuffle queries among a group of users who issue them. As a result any entity cannot profile a certain individual. These works assume the existence of a trustworthy third-party anonymizer, which is not readily available over the Internet at large. Social networks instead of the third party to provide a distorted user profile to the web search engine. In the scheme, every user acts as a search agency of his or her neighbors. They can decide to submit the query on behalf of who issued it, or forward it to other neighbors.

The shortcomings of current solutions in class one is the high cost introduced due to the collaboration and communication. Krause and Horvitz[2] employ statistical techniques to learn a probabilistic model, and then use this model to generate the near-optimal partial profile. One main limitation in this work is that it builds the user profile as a finite set of attributes, and the probabilistic model is trained through predefined frequent queries. These assumptions are impractical in the context of PWS. Xu et al.It Proposed a privacy protection solution for PWS based on hierarchical profiles. Using a user-specified threshold, a generalized profile is obtained in effect as a rooted sub-tree of the complete profile. Unfortunately, this work does not address the query utility, which is crucial for the service quality of PWS. For comparison, our approach takes both the privacy requirement and the query utility into account.

A more important property that distinguishes our work from is that we provide personalized privacy protection in PWS. The concept of personalized privacy protection is first introduced by Xiao and Tao in Privacy-Preserving Data Publishing (PPDP). A person can specify the degree of privacy protection for her/his sensitive values by specifying "guarding nodes" in the taxonomy of the sensitive attribute.

Motivate by this, we allow users to customize privacy needs in their hierarchical user profiles.

C. USER PROFILE

Consistent with many previous works in personalized web services, each user profile in UPS adopts a hierarchical structure. Moreover, our profile is constructed based on the availability of a public accessible taxonomy, denoted as R, which satisfies the following assumption. The repository R is a huge topic hierarchy covering the entire topic domain of human knowledge. That is, given any human recognizable topic t, a corresponding node (also referred to as t) can be found in R, with the sub-tree subtre (t, R) as the taxonomy accompanying.

The repository is regarded as publicly available and can be used by anyone as the background knowledge. In addition, each topic t 2R is associated with a repository support, denoted by subtre (t, R), which quantifies how often the respective topic is touched in human knowledge. If we consider each topic to be the result of a random walk from its parent topic in R, we have the following recursive equation used to calculate the repository support of all topics in R, relying on the following assumption that the support values of all leaf topics in R are available.

$$SUPR(t)= \Sigma_{t'\in C(t,R)} SUPR\,(t')$$

We can observe that the owner of this profile is mainly interested in Computer Science and Music, because the major portion of this profile is made up of fragments from taxonomies of these two topics in the sample repository. Some other taxonomies also serve in comprising the profile, for example, Sports and Adults.

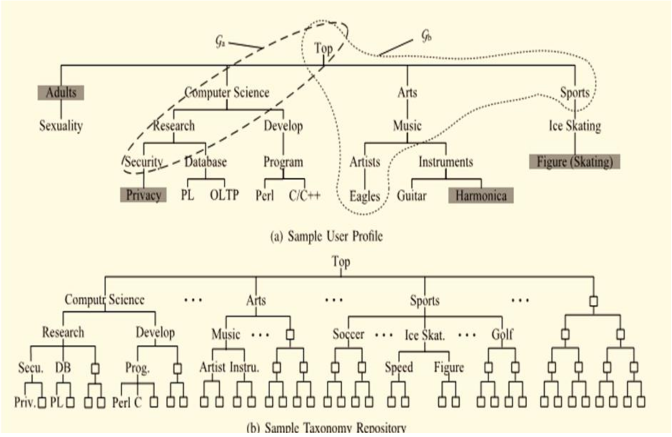

Fig 1. Sample user profile

D. ATTACK MODEL

The proposed work aims at providing protection against a typical model of privacy attack, namely eavesdropping. As shown in Fig. 3, to corrupt Alice's privacy, the eavesdropper Even successfully intercepts the communication between Alice and the PWS-server via some measures, such as man-in-the-middle attack, invading the server, and so on. Consequently, whenever Alice issues a query q, the entire copy of q together with a runtime profile G will be captured by Eve.

Based on G, Eve will attempt to touch the sensitive nodes of Alice by recovering the segments hidden from the original H and computing a confidence for each recovered topic, relying on the background knowledge in the publicly available taxonomy repository R. Alice by recovering the segments hidden from the original H and computing a confidence for each recovered topic, relying on the background knowledge in the publicly available taxonomy repository R.

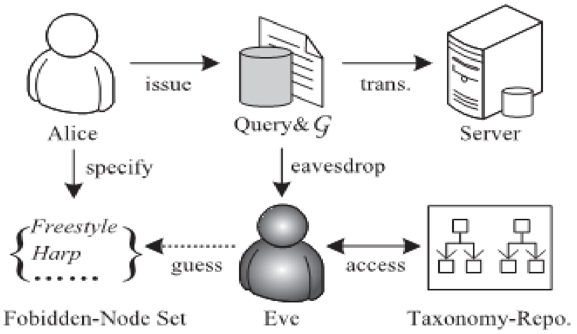

Fig 2. Attack model

E. PROFILE CONSTRUCTION

In this module, a hierarchical user profile is constructed and customized with the user-specified privacy requirements. The first step of the offline processing is to build the original user profile in a topic hierarchy H that reveals user interests. We assume that the user's preferences are represented in a set of plain text documents, denoted by D. To construct the profile, we take the following steps:

1. Parse the query.
2. Check the category names which contains query words.

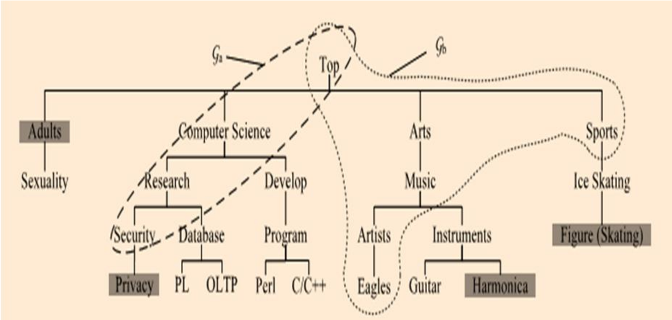

Fig 3. Profile Construction

F. GET PROFILE (Greedy IL)

```
INPUT Query Q, Privacy Threshold
OUTPUT Profile P
  Get Query Q.
  Privacy Threshold
  Split query into words QW.
  Find Seed Profile G. (Any category level contains the
  category name inside the query words).
  CID ={}.

      For i=1 to QW.Count
          a.  Find Category Ids which contains QW[i] in
              their category names.
          b.  Add Category Ids to CID

  Next
  If CID.Count>0
  Create Profile P.
  For j=1 to CID.Count
  While RiskFactor(Q,CID[j]) >Threshold
  P.CategoryId= CID[j]
  P.CategoryName  = Category Name of CID[j]
  End While
  Next
  Return P
  End If
```

IV. BLOCKING TECHNIQUES

A block pair consisting of two small blocks defines only few comparisons. Using such small blocks, the PB algorithm carefully selects the most promising comparisons and avoids many less promising comparisons from a wider neighborhood. However, block pairs based on small blocks cannot characterize the duplicate density in their neighborhood well, because they represent a too small sample. A block pair consisting of large blocks, in contrast, may define too many, less promising comparisons, but produces better samples for the extension step. The block size parameter S, therefore, trades off the execution of non-promising comparisons and the extension quality.

MagpieSort: To estimate the records' similarities, the PB algorithm uses an order of records. As in the PSNM algorithm, this order can be calculated using the progressive MagpieSort algorithm. Since each iteration of this algorithm delivers a perfectly sorted subset of records, the PB algorithm can directly use this to execute the initial comparisons.

A. Attribute Concurrent PSNM

The basic idea of AC-PSNM is to weight and re-weight all given keys at runtime and to dynamically switch between the keys based on intermediate results. Thereto, the algorithm pre-calculates the sorting for each key attribute. The pre-calculation also executes the first progressive iteration for every key to count the number of results. Afterwards, the algorithm ranks the different keys by their result counts. The best key is then selected to process its next iteration. The number of results of this iteration can change the ranking of the current key so that another key might be chosen to execute its next iteration.

V. CONCLUSION

The proposed system is client-side privacy protection framework called UPS for personalized web search. UPS could potentially be adopted by any PWS that captures user profiles in a hierarchical taxonomy. The framework allowed users to specify customized privacy requirements via the hierarchical profiles. In addition, UPS also performed online generalization on user profiles to protect the personal privacy without compromising the search quality. It proposed a greedy algorithm, namely GreedyIL, for the online generalization. The experimental results revealed that UPS could achieve quality search results while preserving user's customized privacy requirements. The results also confirmed the effectiveness and efficiency of our solution. The main benefits are capability to capture a series of queries, User profile is categorized into multiple nodes in the tree structure and past query based suggestion is given to user.

REFERENCES

[1] 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.

[2] R. Krovetz and W. B. Croft. Lexical ambiguity and information retrieval. Information Systems, 10(2):115–141, 1992.

[3] X. Shen, B. Tan, and C. Zhai. Implicit user modeling for personalized search. In Proceedings of CIKM '05, pages 824–831, 2005.

[4] 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.

[5] J. Allan et al. Challenges in information retrieval. In SIGIR Forum, volume 37, 2003.

[6] J. Rocchio. Relevance feedback information retrieval. In The Smart Retrieval System-Experiments in Automatic Document Processing, pages 313–323, Kansas City, MO, 1971. Prentice-Hall.