Support For Privacy Protection In Personalized Web Search

Abstract

Personalized web search (PWS) used for improving the quality of various search services on the Internet. Users might experience failure when search results are inappropriate. User not satisfied with such results. These irrelevances are due to variety of contexts, backgrounds and ambiguous texts. However, private information of user during search has known publicly due to proliferation of PWS. This is main barrier. We propose a UPS framework, the PWS framework which generalizes profiles when there is user required privacy. The framework assumes that user given query do not contain any sensitive information. The aim of framework is to protect privacy of user profile and also preserve usefulness of user profile for PWS. For runtime generalization, we present greedy algorithm, Greedy IL. Also we provide mechanism for client to decide whether to personalize a query or not. The decision can be made before each runtime profiling. It helps to avoid unnecessary exposure of the profile. The experimental results also tell that Greedy IL performs significantly in terms of efficiency.

1. INTRODUCTION

The ordinary people searching for information on the web. Users might experience failure when search results are not appropriate. User not satisfied with such results. These irrelevances are due to variety of contexts, backgrounds and ambiguous texts. Personalized web search (PWS) is a common group of search techniques aiming at providing better results of search that modified for user needs [1]. For PWS, user
information is collected and analyzed to know real intention of user from given query. The PWS is achieved by two methods, one of them is click-log-based methods and another is profile-based. The profile based methods are better than click log based methods. These methods are straightforward. In this method, user’s history is gathered using clicked pages by user. This method is used for only repeated queries which are limitation of the method. This method has been demonstrated to perform consistently and considerably well [3]. But in profile-based method, users general profile is created and user interest is get from that profile. The profile based method improves quality of search and effective for all types of queries. Both methods have pros and cons. The profile-based PWS has demonstrated more effectiveness in improving the quality of search newly, with increasing handling of personal and behavior information of users, that typically gathered completely from query history [3],[4], [5], browsing history[6], [7], click-through data [8], [9], [2] bookmarks [10], user documents [3], [11], and so forth. The attacker can easily attack on personal information of user profile. The protection of private data is one of the privacy issue. The PWS has major barrier for privacy of user profile.

In Profile based PWS, privacy of user data become main concern. Researchers have to protect user data. During search process, they consider two effects as search quality and privacy. They have to improve search quality with the user profile in personalization and hide the contents in user profile to achieve privacy. A few previous studies [11] suggest that people are willing to compromise privacy if the personalization by supplying user profile to the search engine yields better search quality. There are some problems with existing system. They are as follows.

The existing profile-based PWS do not support runtime pro-filing [1]. A user profile is created and generalized only once, and same profile is used with all queries to personalize results of queries. Such one profile fits all method has drawbacks. One evidence reported in [2] is that profile-based personalization may not even help to improve the search quality for some informal queries, while revealing user profile to a server has put the users privacy at risk. Another solution is to decide online whether to personalize the query or not. It is better method and also decides what to expose in the user profile at runtime.

The existing methods do not take into account the customization of privacy requirements [1]. For example, in [11], all the sensitive topics are detected using an absolute metric called surprisal based on the information assumption, that assumptions with less user document support are more sensitive. If there is a large number of documents about any topic, then it is conclude that that topic is a general topic and not sensitive, but that is not true. Some previous works can address ones needs of privacy effectively during the generalization. Many personalization techniques require iterative user interactions when creating personalized search results [1]. The Search results are refined using some metrics which require multiple user interactions, such as rank scoring, average rank [9], and so on. The method is not feasible for runtime profiling. Thus, we need predictive metrics to measure the search quality and violate risk after personalization,
with no incurring repeated user interaction.

All these problems are required to address. The framework is designed to address these problems. The framework is called UPS that is User customizable Privacy preserving Search. The query is given to the search engine. Those queries do not contain any sensitive information. The UPS framework works with a server and a number of clients. The key component is online profiler which is running on client machine itself. It works as proxy.

The PWS framework works in two phases as offline and online. In offline phase, user profile is created at once and that used with every search. But in online phase, user profile is created every time when query is issued by user. When user issues a query, the online profiler at client construct user profile. The query, user profile and privacy requirements are sent to the server. The server search for result for query and personalize the result using privacy requirements received from user. The personalized results are sent back to the proxy which is working on client machine. Finally proxy gives result to the user.

The UPS framework is defined as follows:

We propose a PWS framework called UPS. The UPS generalizes the user profile, when query is given to the system. The user profile is personalized using privacy risk parameter that is how many of risk is lowered by personalizing user profile. For generalization of profile, we develop a Greedy algorithm Greedy IL. Greedy IL means greedy information loss. This algorithm minimizes the information loss. Greedy IL performs significantly.

![Figure1: System architecture of UPS.](image)

2. LITERATURE SURVEY

The web search engine is one of existing retrieval systems, which experiences the problem of one size fits all: means in this result of query which documents to return is only depend on query, and not on user’s preferences. If given query (e.g. Python) is ambiguous, the results are undoubtedly mixed with content (e.g., containing documents on the snake and on the programming language), which is definitely not optimal for the user. Therefore, instead of depending only on the query, which is typically just a set of keywords, retrieval systems should make use of the users search background; from we can learn more about
the user’s proper information requirement. Certainly, background knowledge has been known as a major confront in information retrieval research [5].

The search engines facilitate users to search useful information on the World Wide Web (WWW). On the other hand, as same query is given by different users, usually search engines revisit the same result in spite of who submitted the query. Normally, each user has different information requirements for his/her query. Therefore, the search results should be modified to users with different information requirements. In this paper, it first suggest several approaches to adapting search results according to each users requirement for appropriate information not including any user effort, and then calculate the usefulness of proposed approaches. Tentative results illustrate that search systems that adapt to each users preferences can be achieved by constructing user profiles based on customized mutual filtering with complete analysis of users browsing history in one day [6].

The numerous topics are being discussed on the web and our terms remains quite secure, it is not easy to allow the search engine recognize what we want. Working with ambiguous queries has been an important part in Information Retrieval. Still it remains challenging task. Personalized search has now got major attention to address this challenge, which is based on the basis of users preference may facilitate the search engine disambiguate intention. But, research tells that users are not interested to give any clear input on personal preference. In this paper, we study that how users preference learn by search engine automatically depending on past click history and how it user preference used by search engine to personalize search results.[9]

To improve search quality, personalized web search is best way. But, users are sore with revealing information with private preference to search engines. In contrast, privacy is limited; if there is a gain in service then privacy is compromised. In this paper, it gives a way for users which construct well user profiles automatically. User’s interests are summarized as hierarchy of organization with respect to interests. Privacy requirements are specified by two parameters. These parameters are help to select the content and profile information level which shows to the search engine [11].

3. SCOPE OF WORK

The main goal of the data mining is to extract information from a data set and transform it into an understandable structure for additional use. The data mining task is automatic or semi-automatic analysis of large data, in which it extracts earlier unidentified remarkable patterns like data record group, odd records dependencies. It introduces potential privacy problems in which a user may not be aware that their search results are personalized, and surprise why things become so relevant. An interesting point about personalization that often gets overlooked is the privacy vs. personalization battle.

4. PROPOSED SYSTEM

A. Architectural Design

User profiles are generalized using greedy IL algorithm. The finding motivates us to maintain a priority queue of candidate prune-leaf operators in descending order of the
information loss caused by the operator. This queue, denoted by Q, enables fast retrieval of the best so-far candidate operator. Filtering results based on UPS and results are shown to user. Figure 2 shows Architecture diagram of our proposed system. The diagram shows that when user gives query to search required result, generalized profile is created using greedy algorithm. Query with user profile is given for personalized search, the personalized web search framework then searches the result and personalized that result and final result is given to the user.

![Architecture Diagram]

Figure 2: Architecture Diagram
B. Random Algorithm
A random algorithm takes source of random numbers and makes random choices during execution.

C. Modules
   a. User Interface Design
      To connect with server user must give their username and password then only they can able to connect the server. If the user already exits directly can login into the server else user must register their details such as username, password and Email id, into the server. Server will create the account for the entire user to maintain upload and download rate.

   b. Query Processing
      In this module, the data is given by customer requests goes to server, When a user issues a query, the proxy running on same system generates a user profile at run-time. The output is a generalized user profile satisfying the privacy needs. The generalization process considers two contradictory metrics, as the personalization utility and the privacy risk, both defined for user profiles were administrator maintains all files and responsible for storing that files into cloud.

   c. Combining User Profile And Query:
      In this module, user given query and the generalized user profile are sent together to the PWS server for personalized search. Query with related user preferences stored in a user profile with the aim of providing better search results.
d. Online Generalization
In this module, user given query based on privacy requirements and cost of profiling search results are checked whether to personalize or not.

![Figure 6: Online Generalization](image)

e. Search Personalization:
In this module, user given query search results are personalized according to user profile and delivered back to the query proxy. After results are shown to user.

![Figure 7: Search Personalization](image)

5. MATHEMATICAL MODEL

Execution:
\[ S (O, SW); \]
Here S represents the Data Selection Process,
O represents the Ontology’s & SW represents Semantic web.
Initial Condition
1. The user must have load proper input Data Set.
Functional dependency graph:
1. The user must load proper input Credentials.
Set Theory Analysis
A. Identify the User A= fa1, a2, a3.g
Where A is main set of Users

B. Identify the Attribute AT= fat1, at2, at3.g
Where AT is main set of registered Attribute like at1, at2, at3

C. Identify the User requested For another Attribute RAA= fraa1, raa2, raa3g
Where RAA is main set of Request for another Attribute raa1, raa2, raa3
D. Identify the User Requested All Attribute in Emergency. RAE= rae1, rae2, rae3
   Where RAE is main set of requested all attribute in emergency rae1, rae2, rae3

E. Identify Ontology in data set. IO= io1, io2, io3
   Where IO is main set of Ontology of users like io1, io2, io3

F. Identify Real World Patterns RWP= rwp1, rwp2, rwp3
   Where RWP is main set of Real World Patterns of users rwp1, rwp2, rwp3

G. Identify Scenario Execution:-SE= se1, se2, se3
   Where SE is the main set of Scenario Execution.

H. Identify the processes as P. P= f Set of processes g
   P = fP1, P2, P3, P4 g
   P1 = fe1, e2, e3, e4, e5 g Where,
   fe1= upload data Set g
   fe2= make the entry in database using different encryption algorithm g
   fe3= Benchmark with real world patterns g fe4= Data Selection g
   fe5= Scenario Execution g

I. Identify failure cases as FL Failure occurs when -
   FL= fF1,F2, F3 g
   F1= ffj f if error in uploading data set due to improper input g Identify success case
   SS:-Success is defined as-SS=fS1, S2, S3, S4 g
   a. S1= fsj s if fast and no interrupted in inputs. g
   b. S2=fsj s if data is added into database g

J. Initial conditions as I0
   a. User have proper inputs format.

6. RESULTS

The results of system are presented in this section.

The above figure is registration page, where user register him in system at the first time.
The above figure shows client user interface, where user give username and password to login in to the system.

The above figure shows that login is successful.

The above figure shows that result of search on it.
A. Experimental Setup
The proposed system is implemented on a personal computer with a Pentium IV 2.6 GHz, Intel core 2 Duo, 512-MB DD RAM, 15 color monitor and 40 GB Hard Disk running Microsoft Windows XP. All the modules are implemented in Java (JSP Servlet).

B. Experimental Results
The number of experiments is done with system. The scalability of system depends upon profile size and data set size. The database in our system stores the user profile and queries given by that user. The database consist information in the form of uid, query. Figure 8 shows the relationship between Profile size and Average time.

![Figure 8: Scalability with different profile size](image)

The efficiency is calculated by the number of iterations and response time of generalization. The efficiency is achieved by selecting single query number of times and calculating response time for each iteration. It is also clear from experiments that the quality of result is different when given privacy risk is different. As privacy risk is increases, search quality also increases. The below figure 9 shows the relationship between search quality and privacy risk.

![Figure 9: Search quality versus Privacy risk](image)

7. CONCLUSION
The system gives a privacy protection framework called UPS for personalized web search. In this framework, users specify privacy requirements via the hierarchical profiles. To protect the personal privacy without compromising the search quality UPS performed online generalization on user profiles. The experimental results discovered that UPS could achieve quality search results while preserving users customized privacy requirements.
The results conform effectiveness and efficiency of solution. In future work, we will try to resist adversaries with broader environment knowledge, as better relationship between topics, or capability to confine a sequence of queries from the dupe. Also try to find more refined method to construct the user profile, and superior metrics to guess the presentation and action of UPS.

8. REFERENCES


