Privacy Preserving Web Search by Client Side Generalization of User Profile
Dhanalakshmi College of Engineering, Chennai, Tamilnadu, India

ABSTRACT

Personalized online search (PWS) has incontestable its effectiveness in up the standard of assorted search services on the web. However, evidences show that user’s reluctance to disclose their personal data throughout search has become a serious barrier for the wide proliferation of PWS. We have a tendency to study privacy protection in PWS applications that model user preferences as ranked user profiles. we have a tendency to propose a PWS framework referred to as UPS which will adaptively generalize profiles by queries whereas respecting user such privacy necessities. Our runtime generalization aims at placing a balance between 2 prognostic metrics that valuate the utility of personalization and also the privacy risk of exposing the generalized profile. We are going to use Resource Description Frame Work, for runtime generalization. Where privacy requirements represented as a set of sensitive-nodes. We use to conjoinly offer an internet prediction mechanism for deciding whether personalization is required or not. The decision depends on users wish. When decision is made by the user that particular nodes along with all sub nodes will be removed, in depth experiments demonstrate the effectiveness of our framework.

Keywords: Personalized online search; PWS framework ;Offline Profiling; Generalization; rdf;

I. INTRODUCTION

The web search engine has long become the most important part for ordinary people who look for useful information on the web. But there are some cases where, users might experience failure when search engines return irrelevant results that do not meet their requirements. Such irrelevance is largely due to the enormous variety of users’ contexts and backgrounds, as well as the ambiguity of texts.

The solutions to personalize this search can generally be categorized into two types, namely click-log-based methods and profile-based ones. Among the two types click-log based methods are mostly straightforward. They simply impose bias to clicked pages in the user’s query history. Though this strategy has been demonstrated to perform consistently and considerably well, it will work well on repeated queries from the same user. In contrast, profile-based methods improve the search experience with complicated user-interest models generated from user profiling techniques. Profile-based methods can be potentially effective for almost all sorts of queries, but are reported to be not effective under some circumstances. The abundant amount of data available on the web has been increasing rapidly, especially RDF data.

The Linking Open Data project alone maintains tens of billions of RDF triples in more than 100 interlinked data sources. Besides strong (Semantic Web) community support, this proliferation of RDF data can also be related to the generality of the underlying graph-structured model, i.e., many types of data can be expressed in terms of format including relational and XML data. Even though data representation is flexible, it also has the potential for serious scalability issues. Another problem is that schema information given by user is often unavailable or incomplete, and evolves rapidly for the kind of Resource Description Framework data on the web. Thus, web applications built to exploit RDF data cannot rely on a fixed and complete schema of a single user but, in general, must assume the data to be semi structured. For a Personalized Semantic Web Search the semi structured data should be indexed with RDF.
II. METHODS AND MATERIAL

2. MODULES

1. User Profile and Semantic Data Building
2. Rdf for User Uploaded Data.
3. Search over Indexed Data and Offline Profiling.
4. PSWS with UPS Framework.

2.1. User Profile and Semantic Data Building:

Consistent with many previous works in personalized web services, profile for a particular user in UPS adopts a hierarchical structure. Each users profile is built by considering the availability of a public accessible taxonomy, denoted as R, which satisfies the following assumption. User profile is constructed based on the sample taxonomy repository.

The Resource Description Framework (RDF) is constructed for semantic data on a Relational Database containing Structured as well as Unstructured data. A Schema is identified for the relational database and a RDF representing the schema of the database is constructed through model provided by the jena api. The Model contains all the informations about the data linkages in the schema. In this process the schema can also be altered based on admin requirement so that the search process can be effective.

2.2. Rdf For User Uploaded Data.

The RDF is also generated by mining the text contents uploaded by the users in blogs and the contents of the file are analyzed and the meta contents are manipulated. The meta contents are the key for search process so that the file can be rendered on demand. The Text mining process analyses the text word by word and also picks up the literal meaning behind the group of words that constitute the sentence. The Words are analyzed in WordNet api so that the related terms can be found for use in the meta content in generation of RDF. Generally RDF runs in the web services of Servers in all over the world to provide the schematic datas that the server holds in db to the distribution in the web to access it. Hence this process is handled in real time servers .Hence the structure-oriented approach to RDF data management where data partitioning and query processing make use of structure patterns generated by the RDF. The framework works in two types of phases, the offline and online phase, for unique user. In offline phase, a tree type hierarchical user profile is constructed and customized with the user-specific privacy requirements. UPS consists of a non-trusty search engine server and a number of clients. Each client (user) accessing the search service trusts no one but himself/ herself. The Important component for privacy protection is an online profiler implemented as a search proxy running on the client machine itself. The created proxy maintains both the complete user profile, in a hierarchy of nodes with varying types of semantics, and the user-specified (customized) privacy requirements represented as a set of sensitive-nodes.

In this section, we present the procedures carried out for each user during two different execution Steps, namely the offline and online phases. Generally, the offline phase creates the original user profile and then performs privacy requirement customization according to user-specified topic sensitivity. The subsequent online phase finds the optimal _-Risk Generalization solution in the search space determined by the customized user profile. Specifically, each user has to undertake the following procedures in our solution:

1. Offline profile construction
2. Privacy requirement customization
1. Offline-Profile Construction. In this step, the original user profile is built in a topic hierarchy $H$ that reveals user interests.

2. Privacy Requirement Customization. This procedure first requests the user to specify a sensitive-node set, and the respective sensitivity value for each topic.

2.4. PSWS with UPS Framework.

The online phase handles queries in the following manner:

1. When Client issues a query, the proxy creates a user profile in runtime in the light of query terms. Final outcome of this step will be a generalized user profile satisfying the privacy requirements.

2. Subsequently, the query and the generalized user profile are sent together to the PWS server for personalized search.

3. The search results are personalized with the profile and delivered back to the query proxy.

4. Finally, the proxy either presents the raw results to the user, or re-ranks the results with the complete profile given by the user. As the sensitivity values explicitly indicate the user’s privacy requirements, the straightforward privacy preserving method is to remove subtrees rooted at all sensitive-nodes whose sensitivity values are greater than a threshold value. This method is referred to as forbidding.
   i. Online query-topic mapping, and
   ii. Online generalization.

2.4.1 Query-topic Mapping:

The purposes of online query-topic mapping are
1) To compute a rooted sub tree of $H$, which is called a seed profile, where all topics relevant to $q$ are contained in it; and
2) For obtaining the preference values between $q$ and all topics in hierarchy $H$

2.4.2 Profile Generalization:

This procedure generalizes the seed profile $G_0$ in a cost-based iterative manner relying on the privacy and utility

Algorithm
Algorithm: web semantic rdf algorithm for dynamic proxy profiler
Input: Rdf attributes text mining files and query $Q$
Output: query result set $Q^*$, with respect to profile $p$

If new user then
   Download profilertool; Invoke registration;
   Call proxyprofiler();
   else if
      Call proxyprofiler(); Call search();
   else
      Call admin();

Method proxyprofiler()
   {
      Get semanticdb input; Call dbrdf();
      Get semanticweb input; Call webrdf();
   }

Method admin()
   {
      Categorize userrdf; Call masterrdf();
   }

Method dbrdf()
   {
      Get dbattributes; Get userinput; generate dbrdf;
   }

Method webrdf()
   {
      Get textminingfiles; Call NLP();
   }

Method NLP()
   {
      Invoke chunker, tagger; Get processedresult;
      Invoke wordnet; generate webrdf;
   }

Method masterrdf()
CONCLUSION

This paper presented a client-side protection by generalizing user profile in personalized web search. UPS can be potentially be adopted by any PWS that captures user profiles in a hierarchical taxonomy. The Resource Description Framework allowed users to specify sensitive nodes the privacy requirements via the hierarchical profiles. In addition, UPS also performed online user profile generalization to protect the personal privacy without compromising the search quality. We proposed algorithm in Resource Description Framework, for the online generalization. Our results revealed that UPS could achieve quality search results while preserving clients’s customized privacy requirements. The results also confirmed the efficiency and effectiveness of our solution.

For future work, we will try to resist adversaries with broader knowledge of particular user, such as richer relationship among topics or capability to capture a series of queries from the client. We will also try more sophisticated method to get and create user profile, and better ideas to predict the performance of UPS.

REFERENCES