PRESERVING USER'S PRIVACY IN PERSONALIZED WEB SEARCH

B. Mahesh Reddy[1], T. Lakshmi Prasanna[2]

[1]*M.Tech (CSE)., Dept of CSE, Sri Venkateswara Institute of Science and Technology, Kadapa*

[2]*Assistant Professor, Dept of CSE, Sri Venkateswara Institute of Science and Technology, Kadapa*

Abstract: 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. We study privacy protection in PWS applications that model user preferences as hierarchical user profiles. We propose a PWS framework called UPS that can adaptively generalize profiles by queries while respecting user specified privacy requirements. Our 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. We present two greedy algorithms, namely Greedy DP and Greedy IL, for runtime generalization. We also provide an online prediction mechanism for deciding whether personalizing a query is beneficial. Extensive experiments demonstrate the effectiveness of our framework. The experimental results also reveal that Greedy IL significantly outperforms Greedy DP in terms of efficiency.

Keywords: Privacy Protection, Personalized Web Search, Utility, Risk, Profile.

I. INTRODUCTION

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. The solutions to PWS can generally be categorized into two types, namely click-log-based methods and profile-based ones. The click-log based methods are straightforward they simply impose bias to clicked pages in the user's query history. Although this strategy has been demonstrated to perform consistently and considerably well [2], it can only work on repeated queries from the same user, which is a strong limitation confining its applicability. 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 unstable under some circumstances [2]. Although there are pros and cons for both types of PWS techniques, the profile-based PWS has demonstrated more effectiveness in improving the quality of web search recently, with increasing usage of personal and behavior information to profile its users, which is usually gathered implicitly 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. Unfortunately, such implicitly collected personal data can easily reveal a gamut of user's private life. Privacy issues rising from the lack of protection for such data, for instance the AOL query logs scandal [12], not only raise panic among individual users, but also dampen the data-publisher's enthusiasm in offering personalized service. In fact, privacy concerns have become the major barrier for wide proliferation of PWS services.

A. Motivation s

To protect user privacy in profile-based PWS, researchers have to consider two contradicting effects during the search process. On the one hand, they attempt to improve the search quality with the personalization utility of the user profile. On the other hand, they need to hide the privacy contents existing in the user profile to place the privacy risk under control. A few previous studies [11] suggest that people are willing to compromise privacy if the personalize ation by supplying user profile to the search engine yields better search quality. In an ideal case, significant gain can be obtained by personalization at the expense of only a small (and less-sensitive) portion of the user profile, namely a generalized profile. Thus, user privacy can be protected without compromising the personalized search quality. In general, there is a tradeoff between the search quality and the level of privacy protection achieved from generalization.

Unfortunately, the previous works of privacy preserving PWS are far from optimal. The problems with the existing methods are explained in the following observations:

- The existing profile-based PWS do not support runtime profiling. A user profile is typically generalized for only once offline, and used to personalize all queries from a same user indiscriminatingly. Such "one profile fits all" strategy certainly has drawbacks given the variety of queries. One evidence reported in [2] is that profile-based personalization may not even help to improve the search quality for some ad hoc queries, though exposing user profile to a server has put the user's privacy at risk. A better approach is to make an online decision on
  - whether to personalize the query (by exposing the profile) and
  - what to expose in the user profile at runtime to the best of our knowledge, no previous work has supported such feature.

- The existing methods do not take into account the customization of privacy requirements. This probably makes some user privacy to be overprotected while others insufficiently protected. For example, in [11], all the sensitive topics are detected using an absolute metric called surprisal based on the information theory, assuming that the interests with less user document support are more sensitive. However, this assumption can be doubted with a simple counterexample: If a user has a large number of documents about "sex," the surprisal of this topic may lead to a conclusion that "sex" is very general and not sensitive, despite the truth which is opposite. Unfortunately, few prior works can effectively address individual privacy needs during the generalization.

- Many personalization techniques require iterative user interactions when creating personalized search results. They usually refine the search results with some metrics which require multiple user interactions, such as rank scoring, average rank [9], and so on. This paradigm is, however, infeasible for runtime profiling, as it will not only pose too much risk of privacy breach, but also demand prohibitive processing time for profiling. Thus, we need predictive metrics to measure the search quality and breach risk after personalization, without incurring iterative user interaction.

B. Contributions

The above problems are addressed in our UPS (literally for User customizable Privacy-preserving Search) framework. The framework assumes that the queries do not contain any sensitive information, and aims at protecting the privacy in individual user profiles while retaining their usefulness for PWS. As illustrated in Fig. 1, 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 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.

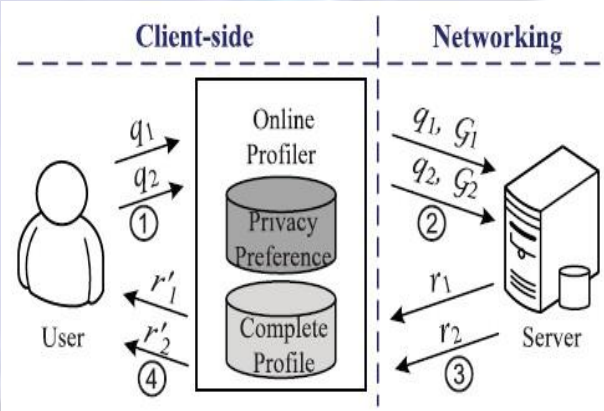

Fig.1. System architecture of UPS.

The framework works in two phases, namely the offline and online phase, for each user. During the offline phase, a hierarchical user profile is constructed and customized with the user-specified privacy requirements. The online phase handles queries as follows:

- When a user issues a query q_i on the client, the proxy generates a user profile in runtime in the light of query terms. The output of this step is a generalized user profile G_i satisfying the privacy requirements. The generalization process is guided by considering two conflicting metrics, namely the personalization utility and the privacy risk, both defined for user profiles.
- Subsequently, the query and the generalized user profile are sent together to the PWS server for personalized search.
- The search results are personalized with the profile and delivered back to the query proxy.
- Finally, the proxy either presents the raw results to the user, or re-ranks them with the complete user profile.

UPS is distinguished from conventional PWS in that it1) provides runtime profiling, which in effect optimizes the personalization utility while respecting user's privacy requirements; 2) allows for customization of privacy needs; and 3) does not require iterative user interaction. Our main contributions are summarized as following:

- We propose a privacy-preserving personalized web search framework UPS, which can generalize profiles for each query according to user-specified privacy requirements.
- Relying on the definition of two conflicting metrics, namely personalization utility and privacy risk, for hierarchical user profile, we formulate the problem of privacy-preserving personalized search as δ-Risk Profile Generalization, with its NP-hardness proved.

- We develop two simple but effective generalization algorithms, Greedy DP and Greedy IL, to support runtime profiling. While the former tries to maximize the discriminating power (DP), the latter attempts to minimize the information loss (IL). By exploiting a number of heuristics, Greedy IL outperforms Greedy DP significantly.
- We provide an inexpensive mechanism for the client to decide whether to personalize a query in UPS. 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.
- Our extensive experiments demonstrate the efficiency and effectiveness of our UPS framework.

The rest of this paper is organized as follows: SectionII Software Architecture for Personalized Search .SectionIII Privacy Protection in Current Web Search Systems. The experimental results and findings are reported in SectionIV. Finally, SectionV concludes the paper.

II. SOFTWARE ARCHITECTURE FOR PERSONALIZED SEARCH

For Web search applications, server-client architecture, as shown in Fig.2 (a), is commonly adopted, where a client (often the web browser) sends queries to a server (the search engine). The search engine analyzes the user information need, looks up its index structure of documents, and returns a ranked list of search results to the client for a user to view. A search engine generally stores user search logs for various kinds of purposes including personalization and anti-spam. Thus it is to the interest of search engines not to remove the search engine logs automatically. Indeed, they tend to keep the search engine logs indefinitely. There are three kinds of software architectures that expand the basic server-client model of Web search to support personalized search. Their main differences lie in where personally identifiable information P (U) is stored and how it is exploited for personalization. In this section, we describe these three kinds of software architectures and analyze what levels of privacy preservation can be achieved with these different architectures.

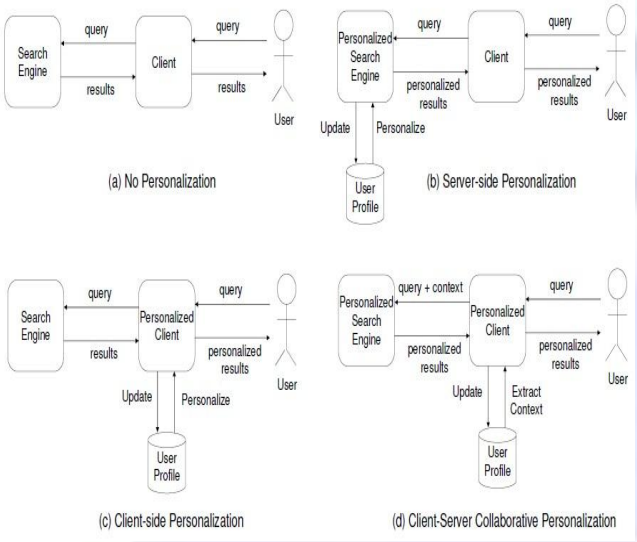

Fig.2. Software Architecture of Personalized Web Search.

A. Server-Side Personalization

For server-side personalization as shown in Fig.2 (b), the personally identifiable information P (U) is stored on the search engine side. The search engine builds and updates the user profile either through the user's explicit input (e.g., asking the user to specify personal interests) or by collecting the user's search history implicitly (e.g., query and click through history). Both approaches require the user to create an account to identify him. But the latter approach requires no additional effort from the user and contains richer description of user information need. The advantage of this architecture is that the search engine can use all of its resources (e.g, document index, common search patterns) in its personalization algorithm. Also, the client software generally requires no changes. This architecture is adopted by some general search engines such as Google Personalized. Currently most personalized search systems with server-side personalization architecture require the user to give consent before his/her search history can be collected and used for personalization. If the user gives the permission, the search engine will hold all the personally identifiable information possibly available on the server side. Thus from the user perspective, it even does not have level I privacy protection.

Since many users fear its potential privacy infringement by search engines, this has hindered the wide adoption of personalization with this architecture. However, if the search engine decides to voluntarily replace the user identity ID (U) with a pseudo user identity $ID^P(U)$, Level I privacy protection can be achieved. When the search engines release the search engine logs to the public or a group of researchers, they generally replace user identity ID (U) by a pseudo user identity $ID^P(U)$. To the third parties receiving these search engine logs, which may use it for personalized search purpose, the user will have Level I privacy protection. If the user decides to use a proxy to communicate with the search engine, all user information going through the same proxy will be combined in a user profile. Through this method, privacy protection can be achieved. However, this method does not always work: When the search engine uses the user login ID to collect user information, this method will not achieve privacy protection; when the search engine only uses the IP address to aggregate the user information, this method works. Sometimes, search engines group users randomly or according to some criteria before they release the search engine logs. Then the user will also have privacy protection to those third parties which receive the search engine logs. It is impossible to implement privacy protection if personalization is done on the server side.

B. Client-side Personalization

For client-side personalization as shown in Fig.2(c), the personally identifiable information is always stored on a user's personal computer. As in the case of server-side personalization, the user profile can be created from user specification explicitly or search history implicitly. The client sends queries to the search engine and receives results, which is the same as in the general web search scenario. But given a user's query, a client-side personalized search agent can do query expansion to generate a new query before sending the query to the search engine. The personalized search agent can also re-rank the search results to match the inferred user preferences after receiving the search results from the search engine. With this architecture, not only the user's search behavior but also his contextual activities(e.g., web pages viewed before) and personal information (e.g., emails, browser bookmarks) could be incorporated into the user profile, allowing for the construction of a much richer user model for personalization. The sensitive contextual information is generally not a major concern since it is strictly stored and used on the client side. Another benefit is that the overhead in computation and storage for personalization can be distributed among the clients.

A main drawback of personalization on the client side is that the personalization algorithm cannot use some knowledge that is only available on the server side (e.g., Page Rank score of a result document). UCAIR adopts the client-side personalization. With proxy functionality applied to the client side, Level II privacy protection can be achieved. If the client side uses an anonymous network such as Tor to communicate with the search engine, privacy protection can also be achieved. In order to achieve privacy protection, additional cooperation of the search engine would be needed as we described.

C. Client-Server Cooperative Personalization

For the client-server cooperative personalization as shown in Fig.2 (d), it is a compromise between the previous two kinds of architectures. The user profile is still stored on the client side, but the server also participates in personalization. At query time, the client extracts contextual information from the user profile, and sends it to the search engine along with the query. The search engine then does personalization with the received context. Compared with client-side personalization, this architecture has an advantage of allowing for the use of a search engine's internal resources. The contextual information sent to the server specifies the user's search preferences (e.g., query expansion terms, topic weight vector). It is extracted from the user profile (e.g., the weight vector can be learned from search history), and is only relevant to a particular query. Therefore, it is a condensed version of the whole user profile (generally a few terms or a weight vector from a user's search history), thus the architecture can minimize the personal information obtained by the search engine. A main drawback is that the condensed contextual information may not be as powerful as the whole user profile. We have not yet seen any personalization products in this category, probably due to the relatively complex architecture. This architecture provides the same level of privacy protection as server-side personalization. However, the personally identifiable information collectable on the server side is less than in the case of pure server-side personalization.

III. PRIVACY PROTECTION IN CURRENT WEB SEARCH SYSTEMS

Currently, there are a variety of search engines on WWW-general search engines such as Google and Yahoo!, meta-search engines such as dogpile and ixquick, special search engines such as cluster search engine vivisimo, and personalized search systems such as UCAIR. In this section, we analyze privacy protection for some of these typical search paradigms.

A. Autonomous Search Engines

When people do web search with an autonomous search engine such as Google, Yahoo, or MSN, both the IP address and query terms are stored on the search engine side unless the user uses a proxy or anonymous communication system additionally. Although Google has a strict and clear privacy policy, the personally identifiable information P (U) is stored on Google severs and the users have no full control of their personal information. According to the levels of privacy protection described, it does not even satisfy privacy protection unless the user applies some privacy protection measures to strengthen the privacy protection themselves. Users are generally not comfortable with counting on others to protect their privacy. Recent history has witnessed several privacy infringement incidents when some companies accidentally or willingly had violated such trust and were facing bankruptcy courts, civil subpoenas or lucrative mergers.

B. Meta Search Engines

There are quite a few Meta search engines on the Web such as Dogpile, Look smart and ix quick. A meta-search engine sends user requests to several autonomous search engines and re-ranks search results returned from each one. When people use the Meta search engines, autonomous search engines only receive all user queries from the single Meta search engine. Thus there is the privacy protection to those underlying autonomous search engines. However, there is no automatic privacy protection for the users of these Meta search engines, which is the same as the scenario when people directly use autonomous search engines. Interestingly, the Meta search engine ixquick guarantees that it removes the IP addresses of users and keep no other unique identity. Thus ID (U) of personally identifiable information is not stored on the server side although TEXT(N) still is. It provides Level III privacy protection for the users of this Meta search engine, but ixquick has no personalization functionality.

C. Client-side Personalized Search Tools

There are also some client-side personalized search tools such as UCAIR]. These client-side personalized search tools are installed on a personal computer and build rich user profiles for individual users. They communicate with autonomous search engines when they do web search. Authors have designed and developed a privacy-preserving personalized search system (UCAIR),which resides on the client side and greatly alleviates the privacy concerns while doing personalized search. A user's personal information including user queries and click through history resides on the user's personal computer, and is exploited to better infer the user' information need and provide more accurate search results. UCAIR is implemented as a web browser plug-in 8.

The software architecture of the system is as Fig.3 As shown in Fig.3; the UCAIR personalized search system has three major components: (1) the implicit user modeling module captures a user's search context and history information, including the submitted queries and any clicked search results and infers search session boundaries. (2) The query modification module selectively improves the query formulation according to the current user model. (3) The result re-ranking module immediately re-ranks any unseen search results when-ever the user model is updated. For example, when the user clicks on a search result to view the corresponding web page, UCAIR would assume that the clicked result summary is appealing to the user and thus reflect the user's information need. It would immediately re-rank the not-yet-viewed results based on the viewed summaries and attempt to pull up results that match the clicked summaries well while pushing down those results that are originally ranked high, but do not match the clicked summaries well.

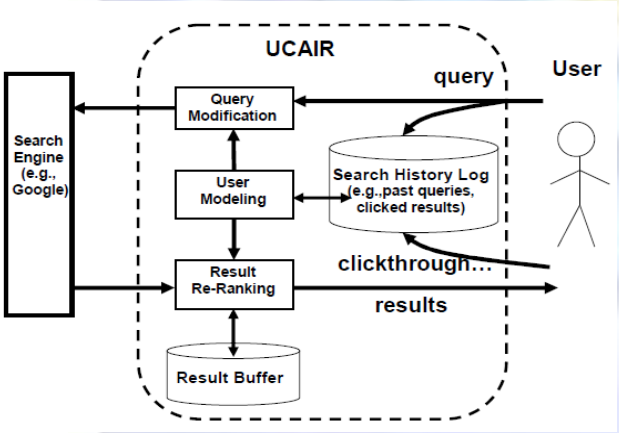

Fig.3. UCAIR architecture.

Thus when the user clicks on the \Back" button of the web browser or \Next" link of the search result page to view more results, the new results displayed would be different from the original results. When a user combines UCAIR with the Tor tool, it will be at the Level III privacy protection even though UCAIR communicates with a general search engine such as Google.

IV. CONCLUSION

This paper presented a 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. We proposed two greedy algorithms, namely Greedy DP and Greedy IL, for the online generalization. Our 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. For future work, we will try to resist adversaries with broader background knowledge, such as richer relationship among topics (e.g., exclusiveness, sequentiality, and so on),or capability to capture a series of queries (relaxing the second constraint of the adversary) from the victim. We will also seek more sophisticated method to build the user profile, and better metrics to predict the performance (especially the utility) of UPS.

V. REFERENCES

[1] Lidan Shou, He Bai, Ke Chen, and Gang Chen, "Supporting Privacy Protection in PersonalizedWeb Search", IEEE Transactions on Knowledge and Data Engineering, Vol. 26, No. 2, February 2014.

[2] Z. Dou, R. Song, and J.-R. Wen, "A Large-Scale Evaluation andAnalysis of Personalized Search Strategies," Proc. Int'l Conf. WorldWide Web (WWW), pp. 581-590, 2007

.

[3] J. Teevan, S.T. Dumais, and E. Horvitz, "Personalizing Search viaAutomated Analysis of Interests and Activities," Proc. 28th Ann.Int'l ACM SIGIR Conf. Research and Development in InformationRetrieval (SIGIR), pp. 449-456, 2005.

[4] M. Spertta and S. Gach, "Personalizing Search Based on UserSearch Histories," Proc. IEEE/WIC/ACM Int'l Conf. Web Intelligence(WI), 2005.

[5] B. Tan, X. Shen, and C. Zhai, "Mining Long-Term Search Historyto Improve Search Accuracy," Proc. ACM SIGKDD Int'l Conf.Knowledge Discovery and Data Mining (KDD), 2006.

[6] K. Sugiyama, K. Hatano, and M. Yoshikawa, "Adaptive WebSearch Based on User Profile Constructed without any Effortfrom Users," Proc. 13th Int'l Conf. World Wide Web (WWW),2004.

[7] X. Shen, B. Tan, and C. Zhai, "Implicit User Modeling forPersonalized Search," Proc. 14th ACM Int'l Conf. Information andKnowledge Management (CIKM), 2005.

[8] X. Shen, B. Tan, and C. Zhai, "Context-Sensitive InformationRetrieval Using Implicit Feedback," Proc. 28th Ann. Int'l ACMSIGIR Conf. Research and Development Information Retrieval (SIGIR),2005.

[9] F. Qiu and J. Cho, "Automatic Identification of User Interest forPersonalized Search," Proc. 15th Int'l Conf. World Wide Web(WWW), pp. 727-736, 2006.
[10] J. Pitkow, H. Schutze, 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.
[11] Y. Xu, K. Wang, B. Zhang, and Z. Chen, "Privacy-EnhancingPersonalized Web Search," Proc. 16th Int'l Conf. World Wide Web(WWW), pp. 591-600, 2007.
[12] K. Hafner, Researchers Yearn to Use AOL Logs, but They Hesitate,New York Times, Aug. 2006.