Privacy Protection in Personalized Search

Xuehua Shen, Bin Tan, ChengXiang Zhai
Department of Computer Science
University of Illinois at Urbana-Champaign

Abstract

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 raise 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 privacy-preserving personalization is a great challenge. In this paper, we systematically examine the issue of privacy preservation in personalized search. We distinguish and define four levels of privacy protection, and analyze various software architectures for personalized search. We show that client-side personalization has advantages over the existing server-side personalized search services in preserving privacy, and envision possible future strategies to fully protect user privacy.

1 Introduction

Although search engines have been successfully deployed to serve users' information needs, they are far from optimal. A major deficiency of existing search engines is that they follow the model of "one size fits all" and are not adaptive to individual users. This causes inherent non-optimality as is seen clearly in the following two cases: (1) Different users may use exactly the same query (e.g., "Java") to search for different information (e.g., the Java island in Indonesia or the Java programming language), but existing search engines return the same results for these users. (2) A user's information needs may change over time. The same user may use "Java" sometimes to mean the Java island in Indonesia and sometimes to mean the programming language. Existing search engines are unable to distinguish such cases.

Clearly, without using more user information and/or the search context of a user it is impossible for a search engine to know which sense "Java" refers to in a query. In order to optimize search accuracy, we must use more user information and personalize search results according to each individual user [10]. To see how personalized search may help improve search accuracy, consider the query "Java" again. The intended meaning of "Java" can often be easily determined by exploiting some naturally available information about a user. Indeed, any of the following additional information about the user could help determine the intended meaning of "Java" in the query: (1) The user is a computer science student as opposed to a travel agent. (2) Before entering this query, the user had just bookmarked or viewed a web page with many words related to the Java programming language, such as "programming" and "applet". (3) The previous query that the user entered is “object-oriented programming” as opposed to “cheap flight ticket”. Exploiting such user information to optimize the ranking of search results for a particular user is very appealing because it does not require any extra effort from the user. In general, personalized search is considered as one of the most promising techniques to break the limitation of current search engines and improve the quality of search results.

Despite the attractiveness of personalized search, we have not yet seen large scale uses of personalized search services. This is not because such services are not available, but likely because users are not comfortable with the lack of protection of user privacy [8, 11]. Google, for example, has deployed a personalized search system [1]. However, to the best of our knowledge, it has not been widely adopted by users yet.

Indeed, there is an inherent tension between providing personalized search and privacy preservation since personalized search requires collecting and aggregating a lot of user information. Specifically, in order to personalize search, a user profile or user model must be constructed to accurately represent a user’s information need. To build a precise user profile, a lot of user information including query and clickthrough history is often aggregated. However, from a user’s privacy perspective, such a user profile can reveal a gamut of user’s private life such as political inclination, family life, and hobbies, which is clearly a serious concern for users. Thus there appears to be a dilemma: high-accuracy Web search requires accurate user modeling which increases the risk of privacy infringement. Indeed, the privacy concern is one of the major barriers in deploying serious personalized search applications, and how to achieve personalized search while preserving users’ privacy is

In this paper, we systematically examine the issue of privacy preservation in personalized search. We distinguish and define four levels of privacy protection, and analyze various software architectures for personalized search. We show that client-side personalization has advantages over the existing server-side personalized search services in preserving privacy, and envision possible future strategies to fully protect user privacy. We also propose several research questions related with the “identifiability” in privacy-preserving personalized search.

The remaining sections are organized as follows. In Section 2, the search privacy problem is defined in a formal way. In Section 3, different levels of privacy are defined and analyzed. In Section 4, possible software architectures of personalized web search system are described and analyzed from the privacy protection perspectives. In Section 6, some research questions related with privacy-preserving personalized search is defined. Section 7 is a summary of our vision of privacy preservation in personalized search.

2 Privacy Concern in Web Search

Search involves interactions between two parties, a user (U) and a search engine (S).

There are two basic interaction cycles between a user and a search engine: (1) Search: A user U composes and submits a query q to search engine S, and the search engine S would return some search results $R = \{R_1, ..., R_n\}$ to the user. (2) Browse: A user U chooses to view a result $R_i \in R$, and the search engine would bring the user the content of R_i.

[1] http://www.google.com/psearch

In a search process involving many such interaction cycles, a user thus potentially reveals the following three kinds of personal information:

1. User identity: This could be a personal user ID in the case when the user has to register an account, or the IP address of the machine that the user is using.
2. Queries: This includes all the queries the user has submitted to the search engine.
3. Viewed results: This includes all the viewed web pages by the user.

Actually, the user also reveals some context information such as the time stamp, but such information is not central to the issue of privacy, thus we do not consider it in this paper.

Since such personal information can potentially reveal a gamut of user's private life such as political inclination, family life, and hobbies, disclosing such information, especially in an aggregated fashion, would clearly raise serious concerns for users.

One may notice that there is a remarkable difference between user's queries and clicked search results. since queries are composed by users themselves, thus directly reveal the user's information need, while the search results are composed by the Web page publishers. Thus in general, queries may contain much more personally identifiable information (PII) than viewed search results. However, from the viewpoint of privacy protection, both queries and viewed results can cause concerns for users and the difference appears to be not crucial. Thus in the rest of the paper, we do not distinguish the queries and viewed search results; instead we refer to them together as *descriptions of user's information needs.*

Thus in any search activity, the information a user U potentially reveals when attempting to satisfy an information need N can be represented as $(ID(U), TEXT(N))$, where $ID(U)$ is some ID revealed about the identity of the user (e.g., a user ID or an IP address), and $TEXT(N)$ is a text description of the information need N (e.g., a set of related queries and/or viewed results). When a user conducts a series of k search activities, the sensitive personal information that the user may reveal can be represented as $P(U) = \{(ID(U,i), TEXT(N,i))\}$ where $i = 1, ..., k$.

The privacy concern of a user is that all or some of the information in $P(U)$ may be captured by some other people in the world. The concern may be less if $P(U)$ is revealed to some "trustable" party (e.g., a search engine company that has a clearly written policy on privacy protection) than to some "untrustable" parties (e.g., any third party who has access to the web search log).

Note that $P(U)$ is precisely what is needed to help a search engine better understand the user's information need. Thus performing personalized search in some sense "requires" a user to release $P(U)$. Such tension has created a barrier for deploying personalized search applications, and the main challenge of privacy-preservation personalized search is to exploit $P(U)$ to help improve the search service for U while protecting $P(U)$ as much as we can from being known by anyone else in the world.

3 Levels of Privacy Protection in Personalized Search

Different users have different requirements of privacy protection. While some users may not want anyone else to know or hold any of their personal information, others may be willing to share some personal information for better search results or services. Thus the level of privacy protection may need to be tuned for different users to accommodate different preferences for the tradeoff of personalization and privacy protection. In this section, we define and analyze four levels of privacy protection in personalized search.

3.1 Level I: Pseudo Identity

A personalized web search system has Level I privacy protection (Pseudo Identity) if:

a The user identity $ID(U)$ is replaced by a pseudo identity $ID^p(U)$ which contains less personally identifiable information than $ID(U)$ does.

b The description of user information needs $TEXT(N, i)$ can be aggregated according to $ID^p(U)$ at the search engine side.

$ID(U)$ can generally be mapped to a single or a small group of users (e.g., family members) with the help of public databases. For example, given an IP address, geographic information such as city and state can be known through the *whois* service. With a pseudo identity $ID^p(U)$, such mapping is not available and some personal information such as the location of the user is protected.

From the viewpoint of personalized search, a pseudo identity $ID^p(U)$ can still be used to group all the descriptions of user information needs to build a user profile without needing $ID(U)$. The content of user profile such as queries and clickthrough is intact at the search engine side. This complete and clean descriptions of user information need can then be exploited to support personalize web search. For example, when AOL released their search engine log in August, 2006, they replaced IP addresses with pseudo identities [4].

Level I is the lowest level of privacy protection. Because of the removal of $ID(U)$, which may otherwise be used to directly identify a user, some people who do not care much about privacy may accept this level of privacy protection. Unfortunately, this level is not enough to protect a user's privacy because it allows aggregation of all the information need descriptions of a user, which can in turn facilitate identification of the user. Since queries directly indicate a user's interests, being able to group many queries from the same user makes it quite possible to identify a user. For example, a New York Times reporter identified a lady in Lilburn, Georgia according to the released AOL query logs.

3.2 Level II: Group Identity

A personalized web search system has Level II privacy protection (Group Identity) if:

a A group of users share a single user identity $ID(U)$.

b The description of user information needs $TEXT(N, i)$ is aggregated at the group level according to $ID(U)$.

This level of protection is achieved when a group of users send their profiles to the search engine in such a way that the search engine can only build a group user profile for the group instead of a user profile for each single user.

In this case, personalized web search can not be done at the individual user level, but is possible at the group level. This may reduce the effectiveness of personalization because a group's information need description is used to model an individual user's information need. However, if the group is appropriately constructed so that people with similar interests are grouped together, we may have much richer user information to offset the sparse description of individual user information needs. Thus the search performance may actually be improved because of the availability of more information from the group profile.

Level II has higher privacy protection than Level I. At this level, one cannot construct an individual user profile. Instead, only an aggregated profile for a group of users can be constructed. Since the identity information of an individual user $ID(U)$ is lost in a group of identity, and the description of user information needs $TEXT(N, i)$ is also mixed with those of other users, it is difficult to infer true information needs of any individual user if the group is appropriately constructed.

A common way to implement the Level II privacy protection is to set up a proxy for a group of users and all the users would communicate with the search engine through the proxy. Currently, there are many public proxy servers available on the Internet.

The obfuscation of query terms can also be considered as an indirect way to achieve Level II privacy protection. For example, TrackMeNot [2], a Firefox web browser plug-in, protects web searchers' identities by periodically issuing randomized search queries to search engines. This method can effectively mix the description of real user information needs with other people's description of user information needs if the noisy queries are carefully constructed so as to resemble common queries. Thus the goal of sending noisy queries is to make a single user profile look like a group of user profiles, i.e., a user profile is undistinguished from a group of other user profiles. This method can also be considered as a way to realize k-anonymity [13].

3.3 Level III: No Identity

A personalized web search system has Level III privacy protection (No Identity) if:

a The user identity $ID(U)$ is not available to the search engine.

b The description of user information needs $TEXT(N, i)$ can not be aggregated on the search engine side, even at the group level.

At Level III, a search engine can not know $ID(U)$ of individual users at all, thus it has no way to aggregate the description of user information needs. At this level, however, it would be impossible to build a user profile on the search engine side, even at the group level. Since the search engine does not have the user profile, personalized search must be supported on a user's own computer. Specifically, the user profile $P(U)$ can be kept on the personal computer of the user U. Personalized search can be achieved by combining general Web search with a local, personalized reranking of results.

A possible way to implement Level III privacy protection is through the anonymous network. For example, The web browser Torpark [3], which is based on Tor (The Onion Router) [4], enables

[2] http://mrl.nyu.edu/˜dhowe/trackmenot/

[3] http://www.torrify.com/

[4] http://tor.eff.org/

the user to communicate anonymously on the Internet. When the user searches the Web using Torpark, the search engine would not be able to decide where the search originally comes from, but the search results can still be returned to the correct user through Tor network.

Level III has a higher privacy protection than Level II. At Level III, it is impossible for the search engine to aggregate any information about the individual user, even at the group level. However, some user information is still kept at the search engine side. For example, the original user queries may be kept at the search engine side. Although a user's query generally does not explicitly contain personal identity $ID(U)$, it sometimes contains quite sensitive information (It is known that some queries contain social security numbers.) It is thus still possible to infer a user's identity just from a query. We will further discuss this issue in Section 6.

3.4 Level IV: No Personal Information

A personalized web search system has Level IV privacy protection (No Personal Information) if:

a Neither the user identity $ID(U)$ nor the description of user information need $TEXT(N)$ is available to the search engine.

At Level IV, a search engine does not know $ID(U)$ of an individual user or the description of user information need $TEXT(N)$ at all. However, the search engine can still return the normal search results to the correct user. Thus the user privacy is fully protected.

On the surface, it appears to be impossible to achieve this level of privacy protection. However, cryptography methodology may be applied to realize this ultimate level of privacy protection. For example, the search engine can release the index to a trusted third party; the user sends the query to the trusted third party and the third party does the search and returns the results to the user. Nevertheless, it is a challenge to design a communication protocol to make sure the ultimate privacy is guaranteed on both the search engine side and the third party side.

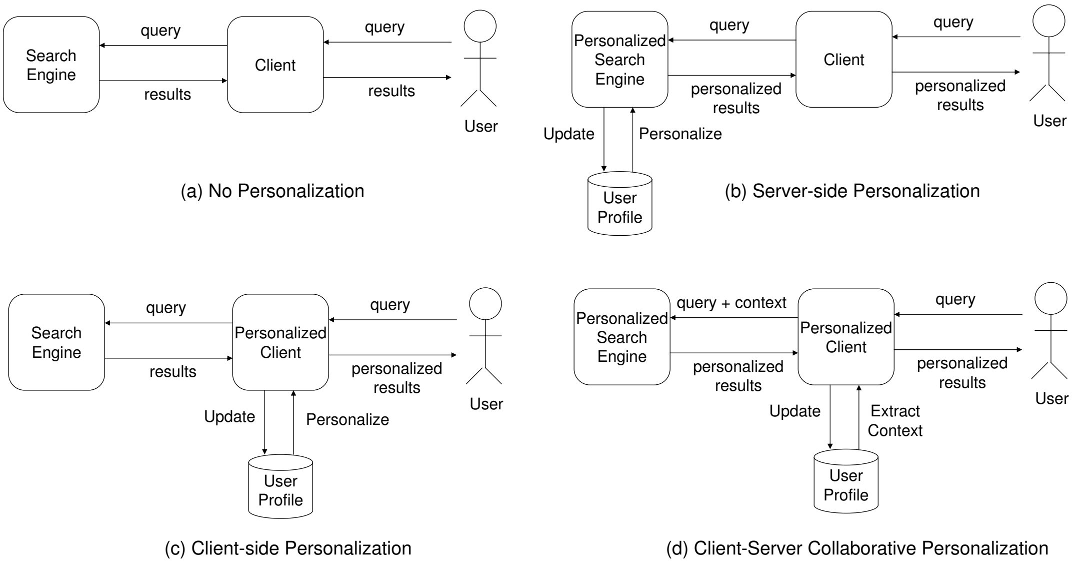
Another possibility for achieving the Level IV privacy protection is that a search engine would be required by law to guarantee that it does not store any user information ($ID(U)$ or $TEXT(N)$). That is, the search engine will have no memory of any activity of a user, even though it would still respond to a user search request directly. This scenario can be considered to be equal to the scenario that the search engine does not know any information about the user. As in the case of Level III privacy protection, since a search engine cannot construct any kind of user profile, personalized search must be supported on the user's computer.

Level IV has the highest level of privacy protection for personalized search. However, it may also have the highest cost due to higher communication cost and encryption/decryption cost, which will delay real-time response. In another form, the cost is that the search engine gives up the logging of any user information, which could otherwise be useful for other purposes such as anti-spam or detection of attacks.

4 Software Architecture for Personalized Search

For Web search applications, server-client architecture, as shown in Figure 1(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.



(a) No Personalization

(b) Server-side Personalization

(c) Client-side Personalization

(d) Client-Server Collaborative Personalization

Figure 1: Software Architecture of Personalized Web Search

4.1 Server-side Personalization

For server-side personalization as shown in Figure 1(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 clickthrough history). Both approaches require the user to create an account to identify himself. 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 [5].

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, Level II 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 Level II 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 Level II privacy protection to those third parties which receive the search engine logs.

It is impossible to implement Level III or Level IV privacy protection if personalization is done on the server side.

4.2 Client-side Personalization

For client-side personalization as shown in Figure 1(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 rerank 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., PageRank score of a result document). UCAIR [12] adopts the client-side personalization.

[5] http://www.google.com/psearch

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, Level III privacy protection can also be achieved. In order to achieve Level IV privacy protection, additional cooperation of the search engine would be needed as we described in Section 3.

4.3 Client-Server Cooperative Personalization

For the client-server cooperative personalization as shown in Figure 1(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.

5 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 [12]. In this section, we analyze privacy protection for some of these typical search paradigms.

5.1 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 [6], 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

[6] http://www.google.com/privacy.html

protection described in Section 3, it does not even satisfy Level I 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 [1].

5.2 Meta Search Engines

There are quite a few meta search engines on the Web such as Dogpile, Looksmart and ixquick [7]. A meta-search engine sends user requests to several autonomous search engines and reranks 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 Level III privacy protection to those underlying autonomous search engine. 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.

5.3 Client-side Personalized Search Tools

There are also some client-side personalized search tools such as Stuff I've Seen [7], Phlat [5] and UCAIR [12]. 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. (See [9, 6] for two related systems.) A user's personal information including user queries and clickthrough 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 Figure 2. As shown in Figure 2, 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 reranking module immediately reranks any unseen search results whenever 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 rerank the not-yet-viewed results based on the viewed summaries and attempt to pull up results that match the

[7] http://www.ixquick.com/

[8] UCAIR is available at: http://sifaka.cs.uiuc.edu/ir/ucair/download.html

clicked summaries well while pushing down those results that are originally ranked high, but do not match the clicked summaries well. 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.

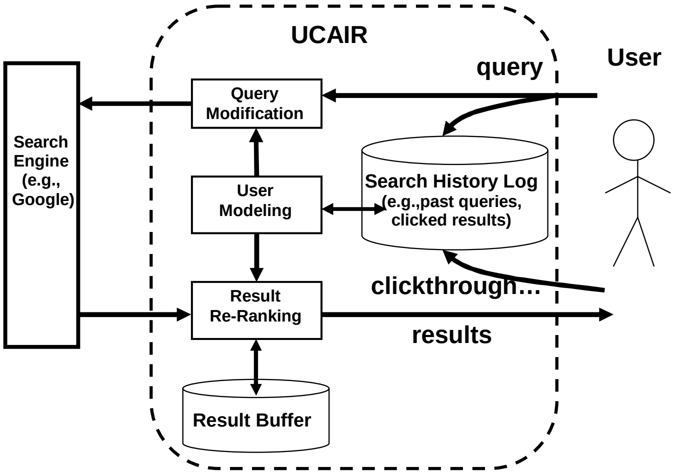


Figure 2: UCAIR architecture

6 Identifiability of Information Need Descriptions

In the previous sections, we have mostly discussed the protection of privacy from the perspective of *directly* identifying a user. However, we may also *indirectly* identify a user based on the description of the user's information needs $TEXT(N, i)$, especially if the descriptions are aggregated. In this section, we discuss the identifiability of information need descriptions. Since queries are far more likely to help infer the identity of a user than viewed results, our discussion will be focused on queries, but it can be easily generalized to cover viewed results as well.

Given queries $Q = \{q_1, q_2, ..., q_n\}$ submitted by a user, some information about the user can be disclosed. A set of queries may give a panorama of the user's information seeking activities, thus raise a lot of privacy issues. However, different queries q_i have different amounts of personally identifiable information. For example, the query "wikipedia" contains nearly no personally identifiable information while the query "landscapers in Lilburn, Ga" probably can reveal that the user is living in or around Lilburn, Georgia.

The difference of the personally identifiable information also holds for other types of user information in the user profile, which can directly affect the user selection of levels of privacy protection in personalized web search and even the general web search. For example, in Level II privacy protection (Group Identity), a set of queries from one group may reveal a lot of personal information while a set of queries from another group may reveal little personal information. Thus if the user is not satisfied with the Level II privacy protection according to the user profile, he/she should consider adopting the Level III privacy protection (No Identity).

In the following subsections, we propose some research questions about the measurement and identifiability of the personally identifiable information in queries.

6.1 Identifiability of a Single Query

Intuitively, different queries contain different personally identifiable information. For example, apparently the query “wikipedia” and the query “landscapers in Lilburn, Ga” contain different amounts of personally identifiable information. Thus we need a way to measure the identifiability of a single query. Here is the formal description of the problem.

Question 6.1 *Given a query q, how can the identifiability of q: $I(q)$ be measured?*

Currently, some researchers [3, 2] use different metrics to measure the privacy of the data. We think that the method proposed in [2], which uses the entropy in information theory to quantify the privacy of the data, is a sound way to measure the identifiability of a single query.

When we do personalized search, personalized search system will reformulate query. It is possible that the query reformulation process alters the value of identifiability. Here is the question about this *differential identifiability.*

Question 6.2 *Given the original query q and the reformulated query q', how can we measure the difference of identifiability: $\Delta(q, q')$?*

In [2], mutual information is proposed to measure the additional information in the perturbed value. This method can also be used to measure the additional personally identifiable information given two versions of queries (the original query q and reformulated query q').

6.2 Identifiability of a Set of queries

Search by search, click by click, the identity of web user becomes more easier to discern. For example, a single query “landscapers in Lilburn, Ga” may not identify a unique person. However, several queries including “landscapers in Lilburn, Ga”, “homes sold in shadow lake subdivision gwinnett county georgia” have been used to identify a lady in Lilburn Georgia by New York Times reporters.

Question 6.3 *Given a set of queries $Q = \{q_1, q_2, ..., q_n\}$, how can we measure the identifiability of Q: $I(Q)$?*

When we do personalized search, similar to the questions in Section 6.1, we have a differential privacy.

Question 6.4 *Given the original query set Q and the reformulated query set Q', how can we measure the difference of identifiability: $\Delta(Q, Q')$?*

6.3 Privacy Preservation through Obfuscation

Some researchers try to protect privacy through sending noisy queries to the search engine, e.g., TrackMeNot [9]. But it is a question whether this method can effectively enhance the user privacy. Here are two questions from oppositive perspectives, both of which are directly related with the effectiveness of this method.

Question 6.5 *Given a query set Q stored on a search engine, does there exist a boolean function $f : Q \longrightarrow B$, such that B is the boolean value, which is T for the original query and F for the noisy query?*

Question 6.6 *Given the user's original query set Q_T, how can the obfuscation method construct a noisy query set Q^*_F so that it maximizes the probability of failure of the boolean function f at the search engine?*

These two research questions are in some sense similar to the research questions about web spam and anti-spam.

7 Summary

Personalized search is a promising way to improve the accuracy of web search, and has been attracting much attention recently. Because effective personalized search requires collecting and aggregating user information, it raises serious concern of privacy infringement for many users. In this paper, we systematically examine the issue of privacy preservation in personalized Web search. We define and analyze four levels of privacy protection. We explore different kinds of software architectures of personalized search and their levels of privacy protection. We also investigate the privacy protection of current search systems.

From the above analysis, we show that client-side personalization has advantages over the existing server-side personalized search services in preserving privacy, and envision possible future strategies to fully protect user privacy. Applying client-side personalization paradigm, Level I, Level II and Level III privacy protection can be easily achieved using various existing technologies. For example, when we combine UCAIR, a client-side personalized search system with Tor, an anonymous communication system, we can achieve Level III privacy protection. When a search engine is willing to share the index with a trusted third party and an appropriate communication protocol is designed, client-side personalized search system can even be used to achieve Level IV privacy protection.

We further discuss identifiability of the description of user information needs, and define some relevant research questions. Privacy concern is a serious issue that has become a major barrier for deploying serious personalized search applications. There are many research challenges to be solved before we can achieve the ultimate Level IV privacy protection. We believe that in the future there will likely be different levels of privacy protection provided by search engines depending on a user's preference for the tradeoff between the privacy concern and the improved search service quality.

[9] http://mrl.nyu.edu/˜dhowe/trackmenot/

References

[1] G. Aggarwal, M. Bawa, P. Ganesan, et al. Vision paper: Enabling privacy for the paranoids. In *Proceedings of VLDB 2004*, 2004.

[2] D. Agrawal and C. C. Aggarwal. On the design and quantification of privacy preserving data mining algorithms. In *PODS*, 2001.

[3] R. Agrawal and R. Srikant. Privacy-preserving data mining. In *SIGMOD Conference*, pages 439–450, 2000.

[4] M. Barbaro and T. Zeller Jr. A face is exposed for AOL searcher No. 4417749. *New York Times*, August 2006.

[5] E. Cutrell, D. C. Robbins, S. T. Dumais, and R. Sarin. Fast, flexible filtering with phlat - personal search and organization made easy. In *Proceedings of SIGCHI 2006*, 2006.

[6] S. Dumais. PSearch: An interface for combining personal and general results.. In *Proceedings of SIGIR 2006 Personal Information Management (PIM) Workshop*, 2006.

[7] S. T. Dumais, E. Cutrell, J. J. Cadiz, G. Jancke, R. Sarin, and D. C. Robbins. Stuff I've seen: a system for personal information retrieval and re-use. In *Proceedings of SIGIR 2003*, pages 72–79, 2003.

[8] C.-M. Karat, C. Brodie, and J. Karat. Usable privacy and security for personal information management. *Communications of the ACM*, 49(1):56–57, 2006.

[9] Y. Lv, L. Sun, J.-Y. Nie, and W. Z. Wan Chen. An iterative implicit feedback approach to personalized search. pages 585–592, 2006.

[10] J. Pitkow, H. Schütze, T. Cass, et al. Personalized search. *Communications of the ACM*, 45(9):50–55, 2002.

[11] S. Sackmann, J. Strker, and R. Accorsi. Personalization in privacy-aware highly dynamic systems. *Communications of the ACM*, 49(9):32–38, 2006.

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

[13] L. Sweeney. K-anonymity: a model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5):557–570, 2002.