Analysis on Supporting Privacy Protection in Personalized Web Search

#**B.Upender**[1], Asst. Professor, CSE Department,E mail: E mail: uppi989@gmail.com
#**Bathula Revathi**[2],M.Tech(CSE) E mail: brevathi25@gmail.com.
Christu Jyothi Institute of Science and Technology, Warangal, T.S, INDIA

Abstract - Web search engines help the users to find useful information on the World Wide Web (WWW). But it has become increasingly difficult to get the expected results from the search engine as a large number of topics are being discussed on the web. Generally, each user has different information needs for his/her query. Therefore, the search results should be adapted to users with different information needs. Personalized web search is a way which provides customized search results for people with individual information goals. But this approach has private issues since it usually requires users to disclose their personal information during search and the users are reluctant to disclose their information. Here we study, how we can provide privacy in PWS applications. To attend this problem we provide a framework called UPS (User customizable Privacy-preserving Search) which provides the search results by adapting to the user's information needs and also provides privacy according to the user specified privacy requirements which help the user to choose content and degree of detail of the profile information that is exposed to the search engine. While generalizing a profile it should strike balance between two predictive metrics that evaluate the utility of personalization and the privacy risk of exposing the generalized profile. Our runtime generalization aims at striking this balance. To provide runtime generalization of profiles we present two algorithms, GreedyDP, GreedyIL. An online prediction mechanism is provided for deciding whether personalizing a query is beneficial or not. Experimental results show that GreedyIL considerably performs well than GreedyDP in terms of efficiency.

Index Terms − Personalised web search, Privacy preserving, personalisation utility, privacy risk, user profile

———————— ✦ ————————

1. INTRODUCTION

Web search engine has become the most important portal for users finding useful information on the web. As the amount of information on the web continuously grows, it has become increasingly difficult for the web search engines to find the information what a user is looking for. For example, for the query "office" some users may be searching for a vacant office space, while other users may be searching for popular Microsoft productivity software.

Therefore, Web search results should adapt to users with different information needs. Personalized web search (PWS) is a promising way aiming at providing better search results, which are tailored for individual user needs.

The click-log based methods are straight forward methods because they simply impose bias to the clicked pages in the user's query history. This strategy performs consistently and considerably well but it can work only on repeated queries from the same user, so this approach has limitations. The Profile-based methods improve the search quality by generating complicated user-interest models by using user profiling techniques.

Profile-based methods prove to be effective for almost all sorts of queries, but they may become unstable under some circumstances. Though there are limitations in this approach, it has demonstrated more effectiveness in improving the quality of web search.

Generally while creating profile of a user the information is gathered implicitly from query history, browsing history, click-through data, bookmarks, user documents and so forth. But unfortunately this leads to privacy issues since such implicitly collected personal data can easily reveal the user's private life. In fact, privacy concerns have become the major barrier for wide proliferation of PWS services.

In this paper we will study how to provide privacy for the personalized web search applications that model the user preferences as hierarichal user profiles. We provide a framework called UPS (User customizable Privacy-preserving Search) which provides the search results by adapting to the user's information needs and also provides privacy according to the user specified privacy requirements which help the user to choose content and degree of detail of the profile information that is exposed to the search engine. An online prediction mechanism is provided for deciding whether to personalize the query (by exposing the profile) and what to expose in the user profile at runtime.

2. LITERATURE SURVEY

Here we focus on the literature of profile-based personalization and privacy protection in Personalized Web Search system.

2.1 Profile-Based Personalization:

The previous works on Profile-based Personalized Web Search mainly focuses on improving the search utility. Generally the Profile-based Personalized Web Search provides the search results by referring to the user profile that reveals an individual information need. Here we review the previous solutions to PWS on two aspects, namely the representation of profiles, and the measure of the effectiveness of personalization.

To facilitate different personalization strategies many profile representations are available in the literature. However in most recent the user profiles are built in hierarchical structures due to their stronger descriptive ability, better scalability, and higher access efficiency. Mostly the hierarchical representations are constructed with existing weighted topic hierarchy/graph, such as ODP, Wikipedia and so on. Another technique is to build the hierarchical profile automatically via term-frequency analysis on the user data. In our proposed UPS framework, we do not focus on the implementation of the user profiles. Actually, our framework can potentially adopt any hierarchical representation based on taxonomy of knowledge.

For the performance measures of PWS in the literature, Normalized Discounted Cumulative Gain (nDCG) is a common measure of the effectiveness of an information retrieval system. But there is a lot of human involvement in performance measuring and to reduce this researchers also propose other metrics of personalized web search that rely on clicking decisions, including Average Precision (AP), Rank Scoring, and Average Rank. In our framework we use the Average Precision metric, proposed by Dou et al., to measure the effectiveness of the personalization in UPS.

Our work also proposes two predictive metrics, namely personalization utility and privacy risk, on a profile instance without requesting for user feedback.

2.2 Privacy Protection in PWS System

There are two classes of privacy protection problems for PWS. One class includes which treat privacy as the identification of an individual. The other includes which consider the sensitivity of the data, particularly the user profiles, exposed to the PWS server. In the literature of protecting user identifications (class one) we try to solve the privacy problem on different levels, including the pseudo identity, the group identity, no identity, and no personal information. The Solution for the first level is proved too fragile. The third and fourth levels are impractical due to high cost in communication and cryptography. Therefore, the existing efforts focus on the second level. The solutions in class two do not require third-party assistance or collaborations between social network entries. In these solutions, users only trust themselves and do not allow the exposure of their complete profiles to an anonymity server.

Krause and Horvitz and Xu et al. proposed a privacy protection solution for PWS but unfortunately, this work does not address the query utility, which is crucial for the service quality of PWS. But our approach takes both the privacy requirement and the query utility into account. We also provide personalized privacy

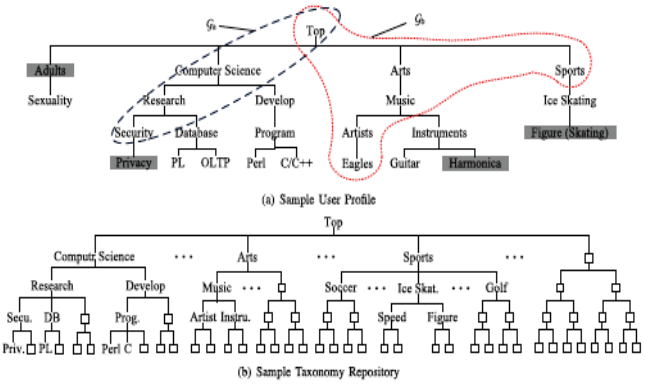


Fig 1: Taxonomy Based user profile

protection in PWS. In this approach we allow users to customize privacy needs in their hierarchical user profiles. Another problem that concerns the privacy protection in PWS is that personalization may have different effects on different queries. Queries with smaller click-entropies, namely distinct queries, are expected to benefit more from personalization, while those with larger values (ambiguous ones) are not and this may even cause privacy disclosure. In our UPS framework, we differentiate distinct queries from ambiguous ones based on a client-side solution using the predictive query utility metric. In this paper, we extend and detail the implementation of UPS and also the metric of personalization utility to capture our three new observations and they are: 1. The existing profile-based PWS do not support runtime profiling. 2. The existing methods do not take into account the customization of privacy requirements. 3. Many personalization techniques require iterative user interactions when creating personalized search results. We also propose a new profile generalization algorithm called GreedyIL. Based on three observations newly added in the extensions, the efficiency and stability of the new algorithm outperforms the old one significantly.

3. PROBLEM DEFINITION

3.1 Preliminaries

Here we first introduce the structure of the user profile in UPS then the customized privacy requirements on a user profile. Then we present the Attack model and finally we define the problem of privacy preserving profile generalization.

3.1.1. User Profile

The user profile in UPS adopts a hierarchical structure. Moreover, our profile is constructed based on the availability of a public accessible taxonomy called repository denoted by R which is a huge topic hierarchy covering the entire topic domain of human knowledge and a repository support is provided by R itself for each leaf topic.

Definition1 (USER PROFILE/H) *A user profile H, as a hierarchical representation of user interests, is a rooted subtree of R. The notion rooted subtree is given in Definition 2*

Definition2 (ROOTED SUBTREE) *Given two trees S and T, S is a rooted subtree of T if can be generated from T by removing a node set X ⊂ T (together with subtrees) from T , i.e., S = rsbtr(X,T).*

A diagram of a sample user profile is illustrated in Fig.a, which is constructed based on the sample taxonomy repository in Fig.b. Each topic t Є H is labelled with a user support, denoted by $sup_H(t)$, which describes the user's preference on the respective topic t. Similar to its repository counterpart, the user support can be recursively aggregated from those specified on the leaf topics:

$$sup_H(t) = \sum_{t' \in C} sup_H(t') \qquad (1)$$

3.1.2 Customized Privacy Requirements

Customized privacy requirements can be specified with a number of sensitive-nodes (topics) in the user profile, whose disclosure (to the server) introduces privacy risk to the user.

Definition3 (SENSITIVE NODES/S).
Given a user profile H, the sensitive nodes are a set of user specified sensitive topics S⊂H, whose subtrees are nonoverlapping, i.e.,∀ s1, s2∈ S (s1≠s2), s2 ∉ subtr (s1, H).

In the sample profile shown in Fig.a, the sensitive nodes S = {Adults, Privacy, Harmonica, Figure (Skating)} are shaded in gray colour in *H*. It must be noted that user's privacy concern differs from one sensitive topic to another. In the above example, the user may hesitate to share her personal interests (e.g., Harmonica, Figure Skating) only to avoid various advertisements. Thus, the user might still tolerate the exposure of such interests to trade for better personalization utility. However, the user may never allow another interest in topic Adults to be disclosed.

Definition4 (SENSITIVITY/*sen(s)*).
Given a sensitive-node s, its sensitivity, i.e., sen(s), is a positive value that quantifies the severity of the privacy leakage caused by disclosing s.

As the sensitivity values explicitly indicate the user's privacy concerns, the most straightforward privacy preserving method is to remove subtrees rooted at all sensitive-nodes whose sensitivity values are greater than a threshold. Such method is referred to as forbidding. However, forbidding is far from enough against a more sophisticated adversary.

3.1.3. Attack Model:

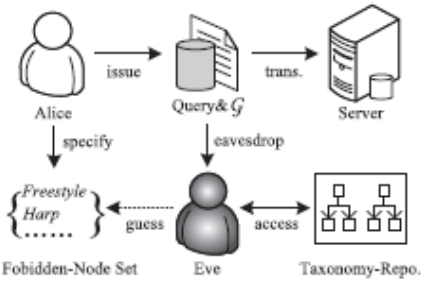


Fig 2: Attack model of personalised web search

Here we illustrate the limitation of forbidding. Our work aims at providing protection against a typical model of privacy attack, namely eavesdropping. As shown in Fig., to corrupt Alice's privacy, the eavesdropper Eve successfully intercepts the communication between Alice and the PWS-server via some measures, such as man-in-the middle attack, invading the server, and so on. Consequently, whenever Alice issues a query q, the entire copy of q together with a runtime profile G will be captured by Eve. Based on G, Eve will attempt to touch the sensitive nodes of Alice by recovering the segments hidden from the original H and computing a confidence for each recovered topic, relying on the background knowledge in the publicly available taxonomy repository R.

Note that in our attack model, Eve is regarded as an adversary satisfying the following assumptions:
Knowledge bounded: The background knowledge of the adversary is limited to the taxonomy repository R. Both the
profile H and privacy are defined based on R.
Session bounded: None of previously captured information is available for tracing the same victim in a long duration. In other words, the eavesdropping will be started and ended within a single query session.

Our approach to privacy protection of personalized web search has to keep this privacy risk under control.

3.1.4 Generalizing User Profile

Now, we exemplify the inadequacy of forbidding operation. In the sample profile in Fig. a, Figure is specified as a sensitive node. Thus, *rsbtr*(S; *H*) only releases its parent *Ice Skating*. Unfortunately, an adversary can recover the subtree of *Ice Skating* relying on the repository shown in Fig. b, where *Figure* is a main branch of *Ice Skating* besides *Speed*. If the probability of touching both branches is equal, the adversary can have 50 percent confidence on *Figure*. This may lead to high privacy risk if *sen(Figure)* is high. A safer solution would remove node *Ice Skating* in such case for privacy protection. In contrast, it might be unnecessary to remove sensitive nodes with low sensitivity. Therefore, simply forbidding the sensitive topics does not protect the user's privacy needs precisely.

To address the problem with forbidding, we propose a technique, which detects and removes a set of nodes *X* from *H*, such that the privacy risk introduced by exposing *G= rsbtr(X, H)* is always under control. Set *X* is typically different from *S*.

We now define the problem of privacy-preserving generalization in UPS as follows, based on two notions named utility and risk. The former measures the personalization utility of the generalized profile, while the latter measures the privacy risk of exposing the profile.

3.2 Problem (δ-RISK PROFILE GENERALIZATION/δ-RPG)

Given a user profile H with sensitive-nodes S being specified, a query q, metric of privacy risk (q, G), metric of utility util (q, G), and a user specified thresholdδ, the δ-risk profile generalization is to find an optimal instance of G (denoted as G), which satisfies*

$G^* = \arg_G \max (\text{util}(q, G)), \text{risk}(q, G) < \delta$ (2)

In the above definition, δ represents the user's tolerance to the privacy risk (expense rate) of exposing the profile.

4. PROPOSED SOLUTION IMPLEMENTATION:

The above problems are addressed in our UPS (literally for User customizable Privacy-preserving Search) framework. UPS is distinguished from conventional PWS in that it 1) 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.

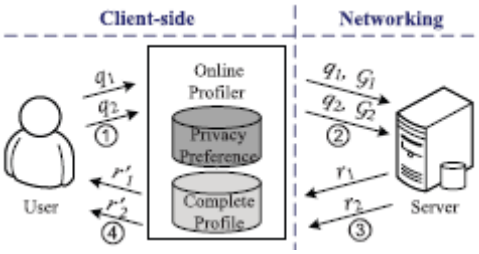


Fig 3: System Architecture of UPS.

As illustrated in Fig., UPS consists of a nontrusty search engine server and a number of clients. Each client (user) accessing the search service trusts no one but himself/herself.

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:

1. 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.
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 reranks them with the complete user profile.

Specifically, each user has to undertake the following procedures in our solution:

1. offline profile construction,
2. offline privacy requirement customization
3. online query-topic mapping, and
4. online generalization.

Offline-1: Profile Construction.

The first step of the offline processing is to build the original user profile in a topic hierarchy *H* that reveals user interests. We assume that the user's preferences are represented in a set of plain text documents, denoted by *D*. To construct the profile, we take the following steps:

1. Detect the respective topic in *R* for every document d ∈ *D*. Thus, the preference document set *D* is transformed into a topic set *T*.
2. Construct the profile *H* as a topic-path trie with *T*, i.e., *H = trie(T)*.
3. Initialize the user support $sup_H(t)$ for each topic t ∈ T with its document support from D, then compute $sup_H(t)$ of other nodes of *H* with $sup_H(t) = \sum_{t' \in C(t,H)} sup_H(t')$.

There is one open question in the above process—how to detect the respective topic for each document d ∈ D. A naive method is to compute for each pair of d and t ∈ R their relevance with a discriminative naıve Bayesian classifier as defined below:

$$dnb(d,t) = \sum_{w \in d} N_{d,w} \frac{N_{t,w} + e}{\sum_{t' \in R} N_{t',w} + e} \quad (3)$$

where $N_{t,w}$ is the frequency of word w in topic t, $N_{d,w}$ is the frequency of w in d, and e is a smoothing factor. The topic with the largest *dnb* value is considered the result. Unfortunately, the naive method is inefficient as many of the topics in R are not relevant to the documents in D. A more efficient way (and the one used in our implementation) is to exploit the user's click log to be the set D. Then, the dnb value for each t ∈ T (q_i) is computed as

$$dnb(d,t) = \sum_{w \in d} N_{d,w} \frac{N_{t,w} + e}{\sum_{t' \in T(qi)} N_{t',w} + e} \quad (4)$$

Offline-2: Privacy Requirement Customization.

This procedure first requests the user to specify a sensitive-node set *S* *H*, and the respective sensitivity value *sen(s)* > 0 for each topic s ∈ S. Next, the cost layer of the profile is generated by computing the cost value of each node t ∈ *H* as follows:

1. For each *sensitive-node, cost(t) = sen(t);*
2. For each *nonsensitive leaf node, cost(t) = 0;*
3. For each *nonsensitive* internal node, *cost(t)* is recursively given by below equation in a bottom-up manner:

$$cost(t) = \sum_{t' \in C(t,H)} cost(t') \times Pr(t'|t). \quad (5)$$

When a query q is issued, this profile has to go through the following two online procedures:

Online-1: Query-topic Mapping:

Given a query q, the purposes of query-topic mapping are 1) to compute a rooted subtree of H; and 2) to obtain the preference values between q and all topics in H. This procedure is performed in the following steps:

1. Find the topics in *R* that are relevant to *q*. We develop an efficient method to compute the *relevances* of all topics in *R* with *q*. These values can be used to obtain a set of *nonoverlapping* relevant topics denoted by T(*q*), namely the *relevant* set. We require these topics to be nonoverlapping so that T(*q*), together with all their ancestor nodes in *R*, comprise a query-relevant trie denoted as *R(q)*. Apparently, *T(q)* are the leaf nodes of *R(q)*. Note that *R(q)* is usually a small fraction of *R*.
2. Overlap *R(q)* with *H* to obtain the *seed profile* G_0, which is also a rooted subtree of *H*.

The leaves of the seed profile G_0 (generated from the second step) form a particularly interesting node set—the overlap between set *T(q)* and *H*. We denote it by $T_H(q)$, and obviously we have $T_H(q)$ *T(q).*

Table: Contents of T (Eagles)

Topics in T(Eagles)	Rel
Top/Arts/Music/Artists/Eagles	23
Top/Sports/American football/NFL/Philadelphia Eagles	14
Top/Science/Biology/Animals/Birds/Raptors/Eagles	7
Top/Society/Military/Aviation/Aircraft/Fighters/F-15	4

Then, the preference value of a topic t ∈ H is computed as following:

1. If t is a leaf node and t ∈ $T_H(q)$, its preference $pref_H(t, q)$ is set to the long-term user support $sup_H(q)$[3],which can be obtained directly from the user profile.
2. If t is a leaf node and $t \notin T_H(q)$, $pref_H(t, q) = 0$.
3. Otherwise, t is not a leaf node. The preference value of topic t is recursively aggregated from its child topics as

$$pref_H(t,q) = \sum_{t' \in C(t,H)} \text{pref}_H(t', H). \quad (6)$$

Finally, it is easy to obtain the normalized preference for each t ∈ H as

$$\Pr(t \mid q, H) = \frac{\text{pref}_H(t,q)}{\sum_{t' \in C(t,H)} \text{pref}_H(t',q)} \quad (7)$$

Note that the first step computes for each $t \in T(q)$ a relevance value with the query, denoted by $rel_R(q)$. These values can be used to model a conditional probability that indicates how frequently topic t is covered by *q*:

$$\Pr(t \mid q) = \Pr(t \mid q, R) = \frac{\text{rel}_R(t,q)}{\sum_{t' \in T(q)} \text{rel}_R(t',q)} \quad (8)$$

Though this probability is not used in this procedure, it is needed to evaluate the discriminating power of q , and to decide whether to personalize a query or not.

Online-2: Profile Generalization:

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

and utility metrics. In addition, this procedure computes the discriminating power for online decision on whether personalization should be employed.

4.1. GENERALIZATION TECHNIQUES:

We use two critical metrics for our generalization problem: 1. Metric of Utility 2. Metric of privacy. In this section we also present our method of online decision on personalization.

4.1.1. Metric of Utility:

The purpose of the utility metric is to predict the search quality (in revealing the user's intention) of the query q on a generalized profile G.

To propose our model of utility, we introduce the notion of Information Content (IC), which estimates how specific a given topic t is. Formally, the IC of a topic t is given by

$$IC(t) = \log^{-1} \Pr(t) \qquad (9)$$

There are Three Components in the utility metric:

1. Profile Granularity: It is the *KL-Divergence* between the probability distributions of the topic domain with and without $\langle q, G\rangle$ exposed. That is

$$PG(q,G) = \sum_{t \in T_G(q)} \Pr(t \mid q, G) \log \frac{\Pr(t \mid q, G)}{\Pr(t)} \qquad (10)$$

$$= \sum_{t \in T_G(q)} Pr(t \mid q, G) IC(t) - Ht\ q,G)$$

2. Topic Similarity: It measures the semantic similarity among the topics in T_G *(q).* This can be computed as the Information Content of the Least Common Ancestor of T_G *(q)* as follows:

$$TS(q,G) = IC(lca(T_G(q))) \qquad (11)$$

3. Discriminating Power: It can be expressed as a normalized combination of *PG (q, G)* and *TS(q, G)* as follows:

$$DP(q,G) = \frac{\mathrm{PG(q,G)} + \mathrm{TS(q,G)}}{2\sum_{t \in T_{H(q)}} \Pr(t \mid q, H) IC(t)} \qquad (12)$$

Where $\sum_{t \in T_{H(q)}} \Pr(t \mid q, H) IC(t)$ is the expected IC of topics in T_H(q), given the profile G is generalized from H. It is easy to demonstrate that the value of DP (q, G) is bounded within (0, 1].

Then, the personalization utility is defined as the gain of discriminating power achieved by exposing profile G together with query q, i.e.,

$$util(q,G) = DP(q,G) - DP(q,R) \qquad (13)$$

where *DP(q, R)* quantifies the discriminating power of the query q without exposing any profile, which can be obtained by simply replacing all occurrences of *Pr(t | q, G)* with *Pr(t | q)* Note that *util(q, G)* can be negative.

That is, personalization with a profile *G* may generate poorer discriminating power. This may happen when *G* does not reduce the uncertainty of *Pr(t | q)* effectively, i.e., *TG(q)=T(q),* and describes the related topics in coarser granularity. Since *DP(q, R)* is fixed whenever q is specified, the profile generalization simply *take DP(q, G)* (instead of *util(q, G)*) to be the optimization target.

4.1.2 Metric of Privacy:

The privacy risk when exposing G is defined as the total sensitivity contained in it, given in normalized form. In the worst case, the original profile is exposed, and the risk of exposing all sensitive nodes reaches its maximum, namely 1. However, if a sensitive node is pruned and its ancestor nodes are retained during the generalization, we still have to evaluate the risk of exposing the ancestors. This can be done using the cost layer computed during Offline-2. Given a generalized profile G, the unnormalized risk of exposing it is recursively given by

$$Risk(t,G) = \begin{cases} cost(t) & if\ t\ is\ leaf, \\ \sum_{t' \in C(t,G)} Risk(t',G) & otherwise \end{cases} \qquad (14)$$

However, in some cases, the cost of a nonleaf node might even be greater than the total risk aggregated from its children. For instance, in the profile G_b (Fig.a), the cost of Music is greater than that of Artist since Music has sensitivity propagation from its sensitive descendent Harmonica. Therefore, it might underestimate the real risk. So we amend the equation for nonleaf node as

$$Risk(t,G) = \max\{cost(t), \sum_{t' \in C(t,G)} Risk(t',G)\} \quad (15)$$

Then, the normalized risk can be obtained by dividing the unnormalized risk of the root node with the total sensitivity in H, namely

$$\text{risk}(q,G) = \frac{\text{Risk (root ,G)}}{\sum_{s \in S} \text{sen (s)}} \quad (16)$$

We can see that risk (q, G) is always in the interval [0, 1].

4.1.3. Online Decision: To Personalize or Not

The results reported in previous woks demonstrate that there exist a fair amount of queries called distinct queries, to which the profile-based personalization contributes little or even reduces the search quality, while exposing the profile to a server would for sure risk the user's privacy. To address this problem, we develop an online mechanism to decide whether to personalize a query. The basic idea is straightforward— if a distinct query is identified during generalization, the entire runtime profiling will be aborted and the query will be sent to the server without a user profile.

We identify distinct queries using the *discriminating power*. The benefits of making the above runtime decision are twofold:

1. It enhances the stability of the search quality;
2. It avoids the unnecessary exposure of the user profile.

4.2 THE GENERALIZATION ALGORITHMS:

In our Method we propose two greedy algorithms, namely the GreedyDP and GreedyIL.

1. The GreedyDP Algorithm:

Given the complexity of our problem, a more practical solution would be a near-optimal greedy algorithm. As preliminary, here we introduce an operator $\xrightarrow{-t}$ called prune-leaf, which indicates the removal of a leaf topic t from a profile. Formally, we denote by $G_i \xrightarrow{-t} G_{i+1}$ the process of pruning leaf t from G_i to obtain G_{i+1}. Obviously, the optimal profile G^* can be generated with a finite-length transitive closure of *prune-leaf*.

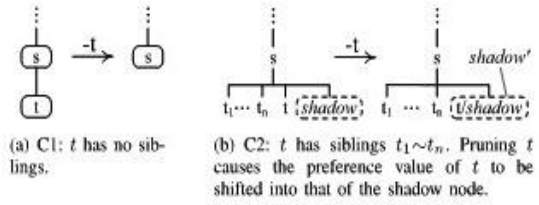
The first greedy algorithm GreedyDP works in a bottomup manner. Starting from G_0, in every ith iteration, GreedyDP chooses a leaf topic t ∈ T_{Gi} (q) for pruning, trying to maximize the utility of the output of the current iteration, namely G_{i+1}. The main problem of GreedyDP is that it requires recomputation of all candidate profiles (together with their discriminating power and privacy risk) generated from attempts of prune-leaf on all t ∈ T_{Gi} (q). This causes significant memory requirements and computational cost.

2. The GreedyIL Algorithm:

The GreedyIL algorithm improves the efficiency of the generalization using heuristics based on several findings. One important finding is that any prune-leaf operation reduces the discriminating power of the profile. In other words, the DP displays monotonicity by prune-leaf. Formally, we have the following theorem:

Theorem 2: *If G' is a profile obtained by applying a prune-leaf operation on G, then DP(q, G)≥ DP(q, G').*

Considering operation $G_i \xrightarrow{-t} G_{i+1}$ in the ith iteration, maximizing $DP(q,G_{i+1})$ is equivalent to minimizing the incurred *information loss*, which is defined as $DP(q,G_i)$-$DP(q, G_{i+1})$. Theorem 2 also leads to the following heuristic, which reduces the total computational cost significantly.



(a) C1: t has no siblings. (b) C2: t has siblings $t_1 \sim t_n$. Pruning t causes the preference value of t to be shifted into that of the shadow node.

Fig 4 : Two cases of Prune-leaf on a leaf t

Heuristic1.

The iterative process can terminate whenever δ-risk is satisfied.

The second finding is that the computation of IL can be simplified to the evaluation of

$$\Delta PG(q,G) = PG(q,G_i) - PG(q,G_{i+1}) \quad (17)$$

Furthermore, consider two possible cases as being illustrated in Fig.: (C1) t is a node with no siblings, and (C2) t is a node with siblings. The case C1 is easy to handle. However, the evaluation of IL in case C2

requires introducing a shadow sibling4 of t. Each time if we attempt to prune t, we actually merge t into shadow to obtain a new shadow leaf shadow', together with the preference of
t, i.e.,

$$Pr(shadow'|q,G) = Pr(shadow|q,G) + Pr(t|q,G) \quad (18)$$

Finally, we have the following heuristic, which significantly eases the computation of IL(t). It can be seen that all terms in following eq. can be computed efficiently.

Heuristic2.

$$IL(t) = \begin{cases} \Pr(t|q,G)\left(IC(t) - IC\left(par(t,G)\right)\right), caseC1 \\ dp(t) + dp(shadow) - dp(shadow'), caseC2 \end{cases} \quad (19)$$

Where $dp(t) = \Pr(t|q,G)\log\frac{\Pr(t|q,G)}{\Pr(t)}$

Heuristic3.

Once a leaf topic t is pruned, only the candidate operators pruning t's sibling topics need to be updated in Q. In other words, we only need to recompute the IL values for operators attempting to prune t's sibling topics.

Algorithm 1 shows the pseudocode of the GreedyIL algorithm. In the worst case, all topics in the seed profile have sibling nodes, then GreedyIL has computational complexity of O ($|G_0|*|T_{G_0}(q)|$). However, this is extremely rare in practice. Therefore, GreedyIL is expected to significantly outperform GreedyDP.

Algorithm: GreedyIL(H, q, δ)
Input: Seed Profile G_0 ; Query q; Privacy threshold δ
Output: Generalized profile G_* satisfying δ-Risk
1 let Q be the IL-priority queue of prune-leaf decisions;
i be the iteration index, initialized to 0;
// Online decision whether personalize q or not
2 if *DP(q,R)* < μ **then**
3 Obtain the seed profile G_0 from *Online-1;*
4 Insert $\langle t, IL(t)\rangle$ into Q for all $t \in T_H(q)$;
5 **while** *risk(q, G_i)* > δ **do**
6 Pop a *prune-leaf* operation on *t* from Q;
7 Set *s*← *par(t,G_i);*
8 Process *prune-leaf* $G_i \xrightarrow{-t} G_{i+1}$;
9 **if** *t has no siblings* **then** // Case C1
10 Insert $\langle s, IL(s)\rangle$ to Q;
11 **else if** *t has siblings* **then** // Case C2
12 Merge *t* into *shadow sibling*;
13 **if** *No operations on t's siblings in* Q **then**
14 Insert $\langle s, IL(s)\rangle$ to Q;
15 **else**
16 Update the IL-values for all operations on t's siblings in O;
17 Update i ← i+1;
18 **return** G_i as G_*;
19 return *root(R)* as G_*;

5. Conclusion

In this paper we presented a client-side privacy protection framework called UPS (User Customisable Privacy Preserving Search) for personalized web search. UPS could likely be adopted by any PWS that captures user profiles in a hierarchical taxonomy. Our proposed framework provided customized privacy requirements via the hierarchical profiles to the users. Through this profile, users control what portion of their private information is exposed to the server and the users can specify to which degree the content should be protected.

In addition, UPS also performed online generalization on user profiles to protect the personal privacy without compromising the search quality. 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 proposed two simple but effective generalization algorithms, GreedyDP and GreedyIL, 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, GreedyIL outperforms GreedyDP significantly. We proposed 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 experimental results revealed that while preserving user's customized privacy requirements our proposed UPS framework could achieve quality search results. The results also confirmed the effectiveness and efficiency of our solution. There is a scope in future that we could 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 from the victim and would work in future. We will also find more advanced method to build the user profile, and better metrics to predict the performance, especially the utility of UPS.

6. REFERENCES:

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

[2] F. Qiu and J. Cho, "Automatic Identification of User Interest for Personalized Search," Proc. 15th Int'l Conf. World Wide Web (WWW), pp. 727-736, 2006.

[3] Y. Xu, K. Wang, B. Zhang, and Z. Chen,"Privacy-Enhancing Personalized Web Search," Proc. 16th Int'l Conf. World Wide Web (WWW), pp. 591-600, 2007.

[4] J. Teevan, S.T. Dumais, and D.J. Liebling, "To Personalize or Not to Personalize: Modeling Queries with Variation in User Intent," Proc. 31st Ann. Int'l ACM SIGIR Conf. Research and Development inInformation Retrieval (SIGIR), pp. 163-170, 2008.

[5] X. Shen, B. Tan, and C. Zhai, "Privacy Protection in Personalized Search," SIGIR Forum, vol. 41, no. 1, pp. 4-17, 2007.

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