A PRIVACY-PRESERVING PERSONALIZED WEB SEARCH (PWS) FRAMEWORK UPS

V.Aruna1, K.Palanivel2
1Department of Computer Science and Engineering, Pondicherry University, India
2 Department of Network and Internet Engineering, Pondicherry University, India

Abstract: Personalized Web Search has established to improve the quality of various search services on the Internet. Due to tremendous data opportunities in the internet, the privacy protection is very important to preserve user search behaviors and their profiles. In the existing system the generalized algorithm namely Greedy DP algorithm were applied to protect private data’s in personalized search engine. The existing systems failed to resist sequential and background knowledge adversaries who has the broader background knowledge such as richer relationship among topics. The proposed framework introduces vector quantization approach piecewise on the datasets which segmentize each row of datasets and that quantization approach is performed on each segment, using the proposed approach which later is again united to form a transformed data set. The proposed work is implemented and is analyzed using certain parameters such as Precision, Recall, Frequency Measure, Distortion and Computational Delay.

Keywords: Privacy Protection, Personalized Web Search, Profile, Vector, Quantization.

I. INTRODUCTION

The web search engine is the most important portal for ordinary people looking for useful information on the web. However, users generally experience failure and get improper results when search engines return irrelevant results that do not meet their real intentions. A typical search engine provides similar set of results without considering of who submitted the query. Therefore, the requirement arises to have personalized web search system which gives outputs appropriate to the user as highly ranked pages.

Personalized Web Search (PWS) [1] is a general category of search techniques which aims to provide better search results, according to the individual user needs. So, for this user information has to be collected and analyzed so that the perfect search results required for the user behind the issued query is to be given to the user. The solution to this is Personalized Web Search (PWS).

It can generally be categorized into two types namely click-log-based methods and profile-based ones. The click-log based methods are simple and straightforward [2]. This method performs the search based upon clicked pages in the user’s query history. Although this method has been demonstrated to perform consistently and considerably well, it can only work on repeated queries from the same user, which is a strong limitation and restricted for certain applications. In contrast, profile-based methods improve the search experience with complicated user-interest models generated from user profiling techniques. Profile-based methods can be proved more effective for almost all sorts of queries, but are reported to be improper under some situations. Although there are reasons and considerations for both types of PWS techniques, the profile-based PWS has proved its more effectiveness in improving the quality of web search recently, with increasing usage of one’s personal and behavioral information to profile its users, which is usually gathered implicitly with the help of query history, browsing history, click-through data, bookmarks, user documents, and so on. Unfortunately, such type of collected personal data can easily reveal a entire scope of user’s private life.

The existing profile-based Personalized Web Search does not support runtime profiling. A user profile is typically generalized for only once offline, and used to personalize all queries from a same user indiscriminately. Such “one profile fits all” strategy certainly has drawbacks given the variety of queries. One evidence reported in 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.

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, 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 “status,” the surprisal of this topic may lead to a conclusion that “status” is very general and not sensitive, despite the truth which is opposite. Unfortunately, little prior work can effectively address individual privacy needs during the generalization.

The algorithm of existing design consists of two heuristic rules by assuming two terms tA and tB. The two heuristic rules used in existing design are

www.gjaet.com
II. BACKGROUND
The existing profile-based Personalized Web Search does not support runtime profiling. A user profile is typically generalized for only once offline, and used to personalize all queries from a same user indiscriminately. Such “one profile fits all” strategy certainly has drawbacks given the variety of queries. One evidence reported in 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.

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, all the sensitive topics are detected using an absolute metric called surprisal [4] 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 “status,” the surprisal of this topic may lead to a conclusion that “status” is very general and not sensitive, despite the truth which is opposite. Unfortunately, little prior work can effectively address individual privacy needs during the generalization.

Figure 1: Personalized Web Search

The figure (1) provides an overview of the whole system. An algorithm is provided for the user to automatically build a hierarchical user profile that represents the user’s implicit personal interests. General interests are put on a higher level and specific interests are put on a lower level. Only some portions of the user profile will be exposed to the search engine in accordance with a user’s own privacy settings [5]. A search engine wrapper is developed on the server side to incorporate a partial user profile with the results returned from a search engine. Rankings from both partial user profiles and search engine results are combined. The customized results are delivered to the user by the wrapper.
**Existing Algorithm**

The existing design algorithm consists of two stages namely called Split process and Buildup process. The following steps describe the Split process of User profile:

**Step 1:** The user sends a query and the partial user profile to the search engine wrapper, where the partial user profile is represented by a set of $\langle t, w_t \rangle$ pairs.

**Step 2:** The list of user profile entries is ordered using ascending or descending based on the value of the user.

**Step 3:** The wrapper calls the search engine to retrieve the search result from the web. Each result comprises of a set of links related to the query, where each link is given a rank from search, called Search Rank. These links are passed to the partial user profile.

**Step 4:** For each of the returned link $l$, a score called UP Score is calculated by the partial user profile as follows:

$$\sum_{t \in \text{tfwl}} \text{UPScore}$$

where $t$ is any term in the partial user profile, and $f$ is the frequency of the term $t$ in the webpage of the link $l$. An UP Rank is assigned to each link according to its UP Score, and the link with the highest UP Score will be ranked first.

**Step 5:** The similarity of user terms can be identified and that covers the document sets with overlap of the user profile.

**Step 6:** The specific terms often appear together with general terms of the user profile and it can be split based on the rank of the user list.

**Step 7:** Re-ranking results by combining ranks from both MSN search and the partial user profile. The final rank, PP Rank (Privacy-enhancing Personalized Rank), is calculated as

$$\text{PP Rank} = \alpha \times \text{UP Rank} + (1 - \alpha) \times \text{MSN Rank}$$

where the parameter $\alpha \in [0, 1]$ indicates the weight assigned to the rank from the partial user profile. If $\alpha = 0$, the user profile is ignored, and the final rank is decided by the user profile instead of the search engine when $\alpha = 1$.

The following step describes the Buildup process of User profile:

**Step 1:** “interest” and “term” are indistinguishable in the context of the user profile. The support of an interest or a term $t$ is $\text{Sup}(t)$, and $S(t)$ represents all the supporting documents for term $t$.

**Step 2:** $\sum \text{Sup}(t) = |D|$ is for all terms $t$ on the leave node, where $|D|$ represents the total number of supports received from personal data.

**Step 3:** According to probability theories, the possibility of one interest (or a term) can be calculated as $P(t) = \text{Sup}(t)/|D|

The drawbacks in the existing design are:

- The existing profile-based PWS do not support runtime profiling.
- The existing methods do not take into account the customization of privacy requirements.

- The existing system suffers from the customized privacy policy maintenance.
- Privacy protection domain requires iterative user interactions for personalization. This produced ineffective results.
- Failed to protect data from sequential and background attackers.

**III. REVIEWS OF EXISTING WORK**

Search personalization [6] is based on the fact that individual users tend to have different preferences and that knowing the user’s preference can be used to improve the relevance of the results the search engine returns. There have been many attempts to personalize web search. These attempts usually differ in:

- How to infer the user preference, whether explicitly by requiring the user to indicate information about herself or implicitly from the user’s interactions,
- What kind of information is used to infer the user’s preference,
- Where this information is collected or stored, whether on the client side or the server side, and
- How this user preference is used to improve.

Lidan Shou, et.al, 2014, [7] 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. It also tells about where the information is collected or stored, whether on the client side or the server side.

Personalized search is a promising way to improve the accuracy of web search [8], and has been attracting much attention recently. However, effective personalized search requires collecting and aggregating user information, which often raises 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. Here we systematically examine the issue of privacy preservation in personalized search. We distinguish the 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.
models of user interests built from both search-related information such as previously issued queries and previously visited web pages and other information about the user such as documents and email the user has read and created. The research suggests that rich representations of the user and the corpus are important for personalization but that it is possible to approximate these representations.

Web search engines (e.g., Google, Yahoo, Microsoft Live Search, etc.) are widely used to find certain data among a huge amount of information in a minimal amount of time. However, these useful tools also pose a privacy threat to the users: web search engines profile their users by storing and analyzing past searches submitted by them. To address this privacy threat, current solutions propose new mechanisms that introduce a high cost in terms of computation and communication. In this paper we present a novel protocol specially designed to protect the users’ privacy in front of web search profiling. Our system provides a distorted user profile to the web search engine. We offer implementation details and computational and communication results that show that the proposed protocol improves the existing solutions in terms of query delay. Our scheme provides an affordable overhead while offering privacy benefits to the users.

**Merits**
- It is used to find certain data among a huge amount of information in a minimal amount of time.
- Proposed protocol improves the existing solutions in terms of query delay.

**Demerits**
- Some tools they pose a privacy threat to the users
- This system provides a distorted user profile to the web search engine

**IV. PROPOSED WORK**

The proposed design contains 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 Risk Profile Generalization, with its NP hardness proved.

It has simple and effective generalization algorithm namely 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 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 runtime profiling to
enhance the stability of search results while avoid the unnecessary exposure of the profile.

**A) Privacy protection in PWS System**

We propose a PWS framework called UPS that can generalize profiles in for each query according to user-specified privacy requirements. Two predictive metrics are proposed to evaluate the privacy breach risk and the query utility for hierarchical user profile. We develop two simple but effective generalization algorithms for user profiles allowing for query-level customization using our proposed metrics. We also provide an online prediction mechanism based on query utility for deciding whether to personalize a query in UPS. Extensive experiments demonstrate the efficiency and effectiveness of our framework. (See Figure 2)

**B) Generating User Profile**

The generalization process has to meet specific prerequisites to handle the user profile. This is achieved by preprocessing the user profile. At first, the process initializes the user profile by taking the indicated parent user profile into account. The process adds the inherited properties to the properties of the local user profile. Thereafter the process loads the data for the foreground and the background of the map according to the described selection in the user profile. Additionally, using references enables caching and is helpful when considering an implementation in a production environment. The reference to the user profile can be used as an identifier for already processed user profiles. It allows performing the customization process once, but reusing the result multiple times. However, it has to be made sure, that an update of the user profile is also propagated to the generalization process. This requires specific update strategies, which check after a specific timeout or a specific event, if the user profile has not changed yet. Additionally, as the generalization process involves remote data services, which might be updated frequently, the cached generalization results might become outdated. Thus selecting a specific caching strategy requires careful analysis. (See Figure 3)

**C) Coding and Encoding in Privacy Protection Technique**

The encoding and decoding process of the cryptography method is illustrated below:

Quantization is the procedure of constraining something from a relatively large or continuous set of values (such as the real numbers) to a relatively small discrete set (such as the integers). The discrete cosine transform (DCT) helps separate the text into parts (or spectral sub-bands) of differing importance (with respect to the image's visual quality). The DCT is similar to the discrete Fourier transform but using only real numbers. There are eight standard DCT variants, of which four are common. The most common variant of discrete cosine transform is the type-II DCT, which is often called simply "the DCT"; its inverse, the type-III DCT, is correspondingly often called simply "the inverse DCT" or "the IDCT". Two related transforms are the discrete sine transforms (DST), which is equivalent to a DFT of real and odd functions, and the modified
Discrete cosine transforms (MDCT), which is based on a DCT of overlapping data.

D) Algorithm of Proposed Design
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 monotony-city by prune-leaf. The benefits of making the above runtime decision are, it enhances the stability of the search quality and it avoids the unnecessary exposure of the user profile. Therefore, GreedyIL is expected to significantly outperform Greedy DP.

The steps for GreedyIL algorithm are

**Step 1:** If $G'$ is a profile obtained by applying a prune leaf operation on $G$, then $DP(q; G) \geq DP(q; G')$.

**Step 2:** Specifically, each candidate operator in the queue is a tuple like $op = (t, IL(t, Gi))$, where $t$ is the leaf to be pruned by $op$ and $IL(t, Gi)$ indicates the IL incurred by pruning $t$ from $Gi$.

**Step 3:** The iterative process can terminate whenever $\theta$-risk is satisfied.

**Step 4:** The second term ($TS(q, G)$) remains unchanged for any pruning operations until a single leaf is left (in such case the only choice for pruning is the single leaf itself).

**Step 5:** In C1, $t$ is a node with no siblings, and in 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 sibling of $t$.

**Step 6:** Each time if we attempt to prune $t$, we actually merge $t$ into shadow to obtain a new shadow leaf shadow0, together with the preference of $t$.

**Step 7:** Prune-leaf only operates on a single topic $t$. Thus, it does not impact the IL of other candidate operators in $Q$. While in case C2, pruning $t$ incurs re-computation of the preference values of its sibling nodes.

**Step 8:** Once a leaf topic $t$ is pruned, only the candidate operators pruning $t$’s sibling topics need to be updated in $Q$.

In general, GreedyIL traces the information loss instead of the discriminating power. This saves a lot of computational cost.

The advantages Enhanced Privacy Protection Framework is as follows:

- It enhances the stability of the search quality
- Improves the privacy protection against different type of attacks
- It avoids the unnecessary exposure of the user profile
- It provides runtime profiling

V. RESULTS & DISCUSSION
The following performance parameters are commonly used in privacy protection technique evaluation. The existing approach is compared with proposed approach using these evaluation parameters. The system is evaluated in terms of Precision, Recall, F-measure, Computational Delay and Distortion.

Precision - It is measure of correctly predicted documents by the system among all the predicted documents. It is defined as the number of relevant documents retrieved by a search divided by the total number of documents retrieved by that search.
Precision = number of correct results/ number of all returned results

### Precision Comparative

**Evaluation of Precision using GreedyIL Algorithm:**
The proposed approach accuracy level is high when compared with the existing one.

**Frequency-Measure:**
F-measure combines precision and recall and is the harmonic mean of precision and recall.

\[ F\text{-measure} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \]

<table>
<thead>
<tr>
<th>Categories</th>
<th>No. Of User Profiles</th>
<th>Precision Existing</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 NG</td>
<td>412</td>
<td>75%</td>
<td>98%</td>
</tr>
<tr>
<td>Sports</td>
<td>300</td>
<td>61%</td>
<td>96%</td>
</tr>
<tr>
<td>Health</td>
<td>669</td>
<td>58%</td>
<td>90%</td>
</tr>
<tr>
<td>Society</td>
<td>442</td>
<td>68%</td>
<td>91%</td>
</tr>
<tr>
<td>Local News</td>
<td>254</td>
<td>68%</td>
<td>73%</td>
</tr>
</tbody>
</table>

### VI. CONCLUSION AND FUTURE WORK

The remarkable development of information on the Web has forced new challenges for the construction of effective search engines. The proposed work provides information on user customizable privacy preserving search framework-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. Another important conclusion we revealed in this proposed work is that personalization does not work equally well under various situations. The click entropy is used to measure variation in information needs of users under a query. Experimental results showed that personalized Web search yields significant improvements over generic Web search for queries with a high click entropy. For the queries with low click entropy, personalization methods performed similarly or even worse than generic search. As personalized search had different effectiveness for different kinds of queries, we argued that queries should not be handled in the same manner with regard to personalization. The proposed click entropy can be used as a simple measurement on whether a query should be personalized. For future work, we try to resist adversaries with border background knowledge including exclusiveness, sequentiality and so on or the capability to capture a series of queries from the victim.

### REFERENCES


