IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 17, Issue 4, Ver. I (July – Aug. 2015), PP 54-65
www.iosrjournals.org

Privacy Protection in Personalized Web Search Via Taxonomy Structure

Syed Arif Ahmed
(M.Tech, Cse)[1,] Mrs. Krishna Keerthi Chennam (Asst. Professor, Cse)[2,]
(Cse Dept, Muffakham Jah College Of Engineering And Technology, Telangana, India)

***Abstract**: Web search engine has long become the most important portal for ordinary people looking for useful information on the web. User might experience failure when search engine return irrelevance information due to enormous variety of user's context and ambiguity of text. The Existing System failed to resist ambiguity of text. Our Proposed System aim at removing ambiguity of text and provide the relevance information to the User. We learn privacy protection in PWS applications that model user preferences as hierarchical user profiles (via taxonomy Structure). We propose a PWS framework called UPS that can adaptively generalize profiles by queries while respecting user-specified privacy requirements via taxonomy structure. Our runtime generalization aims at striking a balance between two predictive metrics that evaluate the utility of personalization and the privacy risk of exposing the generalized profile. For runtime generalization greedy algorithms GreedyDP and GreedyIL are used. For deciding whether to personalizing a query is beneficial online mechanism is provided.*

***Key Words**: Privacy protection, personalized web search, utility, risk, profile, taxonomy*

I. Introduction

Personalized web search (PWS) is a general category of search techniques aiming at providing better search results, which are provided for individual user needs. User information has to be collected and analyzed to figure out the user intention behind the given query. The solutions to PWS can generally be divided into two types, namely click-log-based and profile-based [1] ones. The click-log based methods are straightforward- they simply impose bias to clicked pages in the user's query history logs. Although this strategy has been demonstrated to perform consistently and considerably well [2], it can only work on repeated queries from the same user, which is a big limitation confining its applicability. In contrast, profile-based methods increase the search experience with complicated user-interest models generated from user profiling techniques. Profile-based technique can be potentially effective for almost all forms of queries, but are unstable under some circumstances [2].

Although there are pros and cons for both types of PWS techniques, the profile-based method for PWS has demonstrated more effectiveness in improving the quality of various web search results recently, with increasing usage of personal as well as behavior information to profile its users, which is usually collected implicitly from query history [3], [4], [5], browsing history [6], [7], click-through data [8], [9], [1] bookmarks [10], user documents [3], [11], and so on. Unfortunately, such implicitly gathered personal data can easily reveal a gamut of user's private life. Privacy concerns rising from the lack of protection for such personal data, for example the AOL query logs scandal [12], not only raise panic among individual users, but also dampen the data-publisher's confidence in offering personalized service. In fact, privacy issues have become the major barrier for wide proliferation of PWS services.

II. Motivation

Researchers have to consider two main contradicting effects during the search process, for protecting user's privacy in profile-based PWS. On the one side, they try to improve the quality of search with the personalization utility of the user profile. On the other side, to place the privacy risk under control, they need to hide the privacy contents existing in the user profile. A few prior studies suggest that people are wishing to compromise privacy if the personalization is done by supplying user profile to the search engine yields better search quality. In an ideal, significant growth can be obtained by personalization at the expense of only a small, non-sensitive portion of the user profile, namely a generalized user profile. Consider the example 1) Different users may use exactly the same query (e.g., "Washington") to search for different information (e.g., the Washington DC city in America or the George Washington first president of America), but existing search engines gives the same results for these users. (2) Information needs of user's may change over time. The same user may use "Washington" as a Washington DC city in America and sometimes as the president of America. Existing search engines are unable to detect such cases. So it is clear that without knowing more user information and the search interest of a user it is impossible for a search engine to know which sentence

"Washington" refers in a query. So in order to get results, must use more user information and personalize search results according to each discrete user. Again, consider the query "Washington" to see how personalized search may help improve search accuracy. The intended meaning of "Washington " can often be easily determined by exploiting some naturally available information about a user. Any of the following additional information about the user could help to know the intended meaning of "Washington" in the query: (1) the user is a history student as opposed to a travel agent company. (2) Before typing this query, the user had just viewed or bookmarked a web page with many words related to the Washington. 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 web 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. Thus, without compromising the personalized search quality user privacy can be protected. In general, there is a connection between the level of privacy protection and search quality which is achieved by generalization. Unfortunately, the prior works of privacy preserving PWS are far from optimal. Given below observations shows the problems with the existing methods:

1. The existing profile-based PWS does not support runtime profiling.
2. The existing methods do not take into account the customization of privacy requirements.
3. The existing system failed to resists ambiguity of text.

III. Contributions

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

In order to provide relevance result to the user by eliminating ambiguity in text we are bringing the taxonomy table which is hierarchical structure on to the frontend of the web search Engine which could be visible to the user. Hierarchical taxonomy [1] structure could potentially be adopted by any search engine that captures user profiles in a hierarchical structure.

User could directly choose or select the topic directly from the taxonomy table as provided by the Search Engine.

For ex: User can select the topic from the taxonomy table as

" Main topic "which is the rooted node of the taxonomy table and child node of the rooted topic is "Topic "and the child node of the "Topic" is "Sub topic and child node of "Sub topic" is "Sub topic2".

We can confirms our work to four node of the Taxonomy structure as given above and the selection procedure as follows

Main topic->Topic->Subtopic->Subtopic2

For ex: If Main topic is Sports the selection of the topic from the Taxonomy Structure is given as

Sports->Cricket->Famous Batsman->Sachin

As a result of using the above taxonomy structure in the front end of the web search engine ambiguity of text is eradicated as for

Ex: If a user want to know about artist "Eagles"

normally if we type "Eagles" in the search engine, we will get all the results related to the name "Eagles" as Eagle is biological animal, foot ball player team Eagles and also Eagles as Artist or sometime irrelevance information due to ambiguity of text but by using taxonomy structure in search engine the above problem is eliminated as the user have to select the topic what he is about to search for by giving his own user customizable privacy requirements in order to maintain privacy by making the node sensitive.

If he considers some 'Music' is a privacy data then he can assign the sensitivity as '1' which internally means 'Music' is a privacy data and it should not be exposed to server. Whatever may be the topic he is searching for can be selected from the taxonomy structure from the search engine and get the exact result what he is searching for without getting any irrelevance results due to ambiguity of text.

Ex: Sports->football->famous teams->Eagles

Arts->Music->Artist->Eagles->

Biology->living things->animal->Eagles

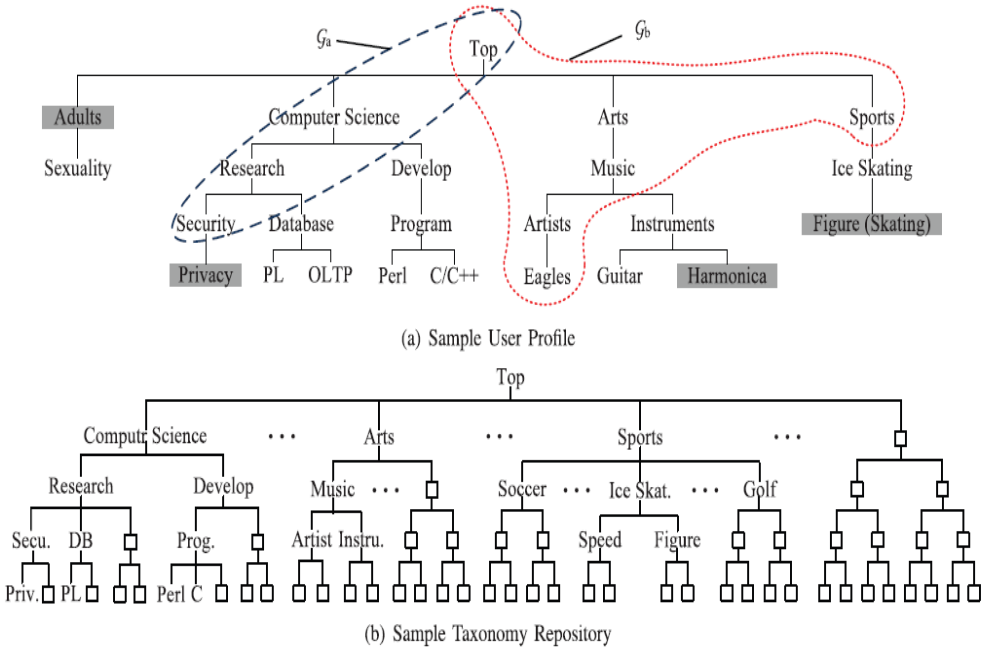


Fig.1.Taxonomy –based user profile

IV. Related Works

Profile-Based Personalization

Prior works on profile-based PWS mainly focus on improving the search utility. The basic proposal of these works is to direct the search results by referring to, often implicitly, a user profile that disclose an individual information goal. We review the prior solutions to PWS on two aspects, namely the presentation of profiles, and the measure of the usefulness of personalization.

Many profile presentations are available in the literature to facilitate different personalization techniques. Earlier techniques utilize term lists/vectors [6] or bag of words [3] to represent their profile. However, many recent works build user profiles in hierarchical structures due to their stronger detailed ability, better scalability, and higher access efficiency. The majority of the hierarchical structures are built with existing weighted topic hierarchy/graph, such as

ODP [2],[15],[4],[16],Wikipedia [17], [18], and soon. Another work in [11] builds the hierarchical profile automatically through term-frequency analysis on the user data. In our proposed UPS framework, we concentrate on the implementation of the user profiles. Actually, our UPS framework can potentially acquire any hierarchical representation based on taxonomy of knowledge.

As for the performance measures of PWS, Normalized Discounted Cumulative Gain (nDCG) [19] is a common metric of the effectiveness of an information retrieval system. It is ground on a human-graded relevance scale of item-positions in the result list set, and is, therefore, known for its high cost in explicit feedback collection. To reduce the human participation in performance measuring, researchers also present other metrics of personalized web search that depends on clicking decisions, including Average Precision (AP) [20], [11], Rank Scoring [14], and Average Rank [4], [9].Average Precision metric, proposed by Dou et al. [2],is used to measure the effectiveness of the personalization in UPS framework. Meanwhile, our work is distinguished from prior studies as it also proposes two predictive metrics, namely privacy risk and personalization utility, on a profile instance without requesting for user feedback.

Privacy protection in PWS System

We propose a PWS framework called UPS that can generalize profiles for each query according to user specified privacy requirements. Two predictive metrics are proposed to estimate the privacy risk and the query utility for hierarchical user profile taxonomy. We develop effective generalization algorithm for taxonomy user profiles allowing for query customization using our proposed metrics. Online prediction mechanism based on query utility for deciding whether to personalize a query in UPS is provided.

Table 1: Symbols and Descriptions

Symbol	Description
|T|	The count of nodes of the tree T
t∈T|N⊂T	t is a node(N is a node set) in the tree T
Subtr(t,T)	The subtree rooted on t within the tree T
rsbtr(N,T)	The rooted subtree of T by removing the set N
trie(N)	The topic-path prefix tree built with the set N
root(T)	The root of the tree T
part(t,T)	The parent of t in the tree T
lca(N,T)	The least common ancestor of the set N in T
C(t,T)	The children of t within the tree T

V. Existing System

The existing profile-based Personalized Web Search does not support runtime profiling. A user profile is consistently generalized for only once offline, and used to personalize all queries randomly from a same user. Such “one profile fits all" strategy certainly has drawbacks given the variety of queries. One evidence reported that profile based personalization may not even help to improve the search quality for some ad hoc queries, though showing user profile to a server has put the user's privacy at risk. The existing methods do not support customization of privacy requirements. This perhaps makes some user privacy to be more protected while others insufficiently protected. For example: all the sensitive topics are detected using an absolute metric called surprise based method on the information theory, thinking that the interests with less user document support are more sensitive. However, this assumption can be suspect with a simple counter example: If a user has a large number of documents about \sex,"the surprise of this topic may lead to a conclusion that \sex" is very general and non sensitive, despite the truth which is opposite. Unfortunately, few previous works can effectively address individual privacy needs during the generalization. Many personalization methods or techniques require iterative user interactions when creating personalized search results. They usually filter the search results with some metrics which require multiple user interactions, such as average rank, rank scoring and so on. This paradigm is, however, infeasible for runtime profiling, as it will not only cause too much risk of privacy, but also demand prohibitive processing time for user profiling. Thus, need predictive metrics to measure the search quality and risk after personalization, without involving iterative user interaction.

1. User sends a query 'q' online to the server.
2. Query is stored on server and it generates user's profile 'g', which is resided at the server side. Clients have no control over that profile.
3. Server gives online response 'r' to the client according to query.

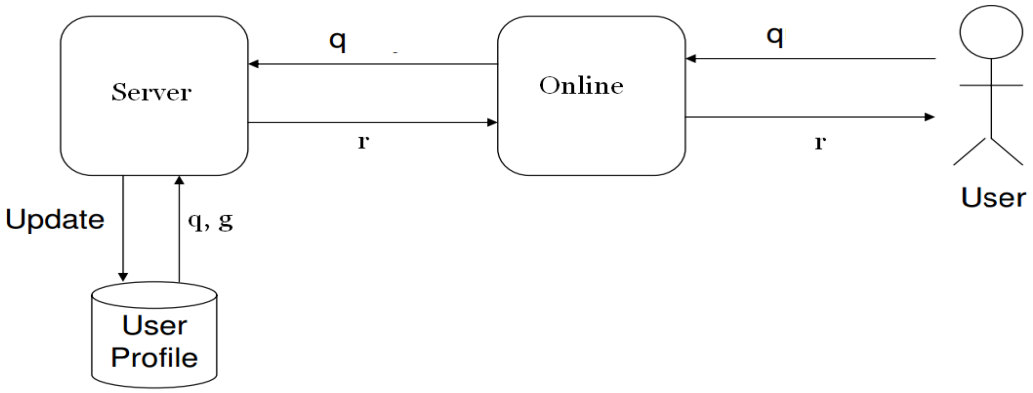


Fig.2 Existing System Architecture

Disadvantages:

1. Users might experience failure when search engines return irrelevant results that do not meet their real intentions.
2. Such irrelevance is largely due to the enormous variety of users' contexts and ambiguity of texts.
3. The existing profile-based PWS does not support runtime Profiling.
4. The existing methods do support the customization of privacy requirements.
5. Many personalization techniques or methods require iterative user interactions when creating personalized search results.
6. All the sensitive topics are found using an absolute metric called surprise based on the information theory.

VI. Proposed System

The proposed system contains a privacy-preserving personalized web search framework UPS, which can generalize profiles for each user query according to his specified privacy requirements. Relying on two conflicting metrics, privacy risk and personalization utility, for hierarchical user profile, we compose the problem of privacy preserving personalized search as Risk Profile Generalization, with its NP-hardness proved. It has simple but effective generalization algorithms, GreedyIL and GreedyDP [1], to support runtime profiling. While the former tries to minimize the information loss (IL), the latter attempts to maximize the discriminating power (DP). By considering a number of heuristics, significantly GreedyIL outperforms GreedyDP. We provide an inexpensive mechanism for the client to decide whether to personalize a query in UPS is beneficial or not. 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.

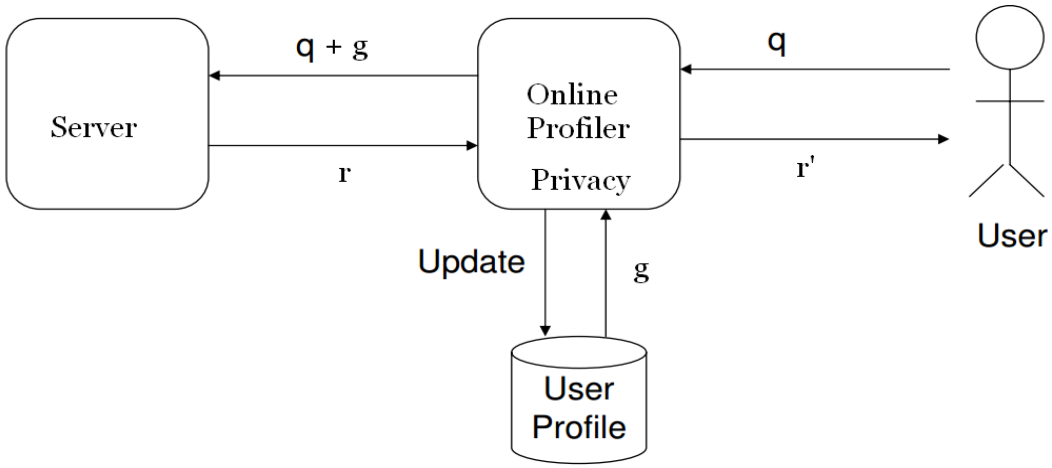


Fig.3 Proposed System Architecture

User Profile

Consistent with many previous works in personalized web services, User has to build his profile according to his interest which is hierarchical taxonomy structure. Moreover, our user profile is built based on the availability of a public accessible taxonomy structure, denoted as R.

Customized Privacy Requirements

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

Definition 1 (SENSITIVE NODES/S) Given a user profile H, the sensitive nodes are a set of user given sensitive topics S ⊂ H, whose sub trees are non overlapping, i.e., ∀s1 ,s2 ∈S(s1 ≠ s2); s2 ≠ subtr(s1, H).

In the sample profile shown in Fig.1, the sensitive nodes S = {Adults, Privacy, Harmonica, Figure {Skating}} are shaded in dark in H. It must be noted that user's privacy concern differs from one topic to another. Ex: In fig.1, the user may hesitate to share his/her personal interests (e.g., Harmonica, Figure Skating) only to avoid various recommendation/advertisements. Thus, the user might still compromise the exposure of such interests to trade for better personalization utility. However, the user may never wants some interest in topic Adults to be disclosed to server. To address the difference in privacy concerns, we allow the user to specify sensitivity for each node s ∈ S.

Definition 2 (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 direct privacy-preserving technique is to prune sub trees rooted at all sensitive-nodes whose sensitivity values are greater than a threshold. Such method is termed to as forbidding. However, forbidding of sub trees is far from enough against a more sophisticated adversary.

Generating User Profile

The generalization process has to meet specific requirements to handle the user profile. This is achieved by processing the user profile. At first, the process creates the user profile by taking into account the indicated parent user profile. The process adds the inherited properties to the properties of the local user profile. After that 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 allows caching and is helpful when

considering an implementation in a production domain. The reference to the user profile can be used as an identity for already processed user profiles. It allows performing the customization of user profile once, but reuses the result many times. However, it has to be taken into account that an update of the user profile is also made during generalization process. This requires some update methods, which check after a specific timeout or event, if the user profile has not changed yet. Furthermore, as the generalization process requires remote data services, which might be changed / updated frequently, the cached generalization process results might become old. Thus selecting a specific caching strategy method involves careful analysis.

VII. Ups Procedures

In this section, we start the procedure for each user into two different execution phases, namely offline and online phases. Normally, the offline phase constructs the original user profile and then performs privacy requirement customization according to user given topic sensitivity. The subsequent online phase finds the Optimal δ-Risk Generalization solution in the search space determined by the customized user profile. As mentioned in the prior section, the online generalization procedure is directed by the global risk and utility metrics. The computation of these metrics depends on two intermediate data structures, namely a cost layer and a preference layer defined on the user profile. The cost layer defines for each node $t \in H$ a cost value cost (t) ≥ 0, which indicates the total sensitivity at risk caused by the disclosure of t. These cost values can be calculated offline from the user specified sensitivity values of the sensitive nodes. The preference layer is calculated online when a query q is issued. It contains for each node $t \in H$ a value indicating the user's related query preference on topic t. These preference values are computed rely on a procedure called query topic mapping.
Generally, each user has to undergo the following process in our solution:

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

Offline 1 **User Profile Construction**: The first step of the offline process is to build the original user profile in a topic hierarchy H that reveals user interests. We expect that the user's preferences are represented in a set of plain text documents, indicated by D. To construct the user profile, we consider the given below steps:

1. Detect the respective topic in R for every document $d \in D$. Hence, the preference document set D is converted into a topic set T.
2. Build the profile H as a topic path tri with T, i.e. H = tri (T).
3. Initialize the support of user $sup_H(t)$ for each topic $t \in T$ with its document support from D, then compute $sup_H(t)$ of other nodes of H with
4. There is one open question in the above process how to detect the respective topic for each document $d \in D$. We present our solution to this problem in our implementation.

Offline-2 **Privacy Customization:**

User has to specify a sensitive node-set to achieve privacy. For example, if he considers some 'Harmonica' is a privacy data then he can assign the sensitivity as '1' which internally means 'Harmonica' is a privacy data and it should not be exposed to server. And if he considers some 'Arts' is normal/not privacy one then he can assign the sensitivity as '0'which internally means 'Arts' can be exposed to server. We have to generate **cost layer** of the profile by computing the cost value of each node $t \in H$.
1. For each sensitive-node, Cost (t) = Sen (t); where t is topic;
2. For each non-sensitive node, Cost (t) =0;

For each non-sensitive internal node,

$$\text{Cost}(t) = \sum_{t' \in C(t,H)} \text{cost}(t') \times \Pr(t'|t) \quad (1)$$

Where $\Pr(t'|t) = \text{probability of } t' \text{ and } t, \text{and} t' = \text{immediate child of} t$

Till now, we have obtained the customized profile with its cost layer available.

Query-Topic Mapping

At runtime user will input's some query, and this query should be matched with the topics in topic-hierarchy (H). We have generated sub-topic-hierarchy which contains the matched topics with the query and this sub-topic-hierarchy is known as **Seed-Profile (G_o)**. After obtaining the seed profile we have to obtain the preference value between q and all topics in H. This is done by using following steps**.**

i. Obtaining Seed-Profile G_o

Find the set of topics in Repository R (Huge topic hierarchy, understandable by user) relevant to q (user inputted query).

These set of non-overlapping relevant topics are denoted by T (q), namely relevant set.

T (q) along with their parent nodes is known as R (q).

For better understanding consider an example.

If the user input's a query as "Eagles" then T (Eagles) returns

Contents of T(Eagles):-

a. Arts/Music/Artists/Eagles

b. Sports/ Football/Famous team/ Eagles

c. Biology/Living thing/Animal/Eagles

Leaf nodes (Eagles) are treated as $T_H(q)$and the tree from the root node to these leaf nodes is treated as R(q) and $T_H(q) \supset T(q)$

Overlap the R(q)with H to obtain the seed-Profile(G_o). Which is also rooted sub-tree of H.

I) Compute the Preferences for the topics

1. If t is a leaf node and $t \in T_H(q)$, its preference

$pref_H(t,q) = \sum_{t' \in C(t,H)} prefH(t', q) = rel_R(t', H)$

Where $rel_R(t', H)$ indicates the relevance value with the query.

2. Normalized preference for each $t \in H$ as

$Pr(t|q, H) = pref_H(t,q) / \sum_{t' \in TH(q)} prefH(t', q)$

3. The relevance values can be used to model a probability that indicates how frequently topic t is covered by q.

$$Pr(t|q) = Pr(t|q,R) = rel_R(t,q) / \sum_{t' \in TH(q)} relR(t', q) \quad (2)$$

The preferences which we calculated above of t can be estimated as with its long-term preference ($sup_H(t,q)$in case of t is leaf node) in the user profile.

Where $sup_H(t,q)$is support of the topic which is obtained by 'how often the respective topic is touched by human knowledge from'. Meaning that the $sup_H(t,q)$ can be calculated as count of leaves in subtr(t,R). Where subtr(t,R) is the sub-tree of t node.

$$sup_H(t,q) = \sum_{t' \in C(t,H)} supH(t')$$

Finally, by using the conditional-probability is used to evaluate the discriminating power of q, and decide whether to personalize a query or not.

1. Profile Generalization

Here, Seed-Profile G_o is generalized in a cost-based iterative manner relying on the privacy and utility metrics.

1. Metric of Utility

This metric is to predict the search quality (in revealing the user's intention)of the query q on a generalized profile G. For fulfilling this requirement we have to evaluate 5 things.

a) Information Content (IC):

It estimates how specific a given topic is. Formally the IC of a topic is given by

$$IC(t) = \log^{-1} Pr(t) \quad (3)$$

b) Profile Granularity (PG)

It is the **KL-Divergence** between the probability distributions of the topic domain with and without <q,G>exposed.

With respect to generalized profile G and query q we can intuitively (to know something without any evidence) expect more discriminating power when

i) More specific topics are observed in $T_G(q)$, or

ii) The distribution of Pr(t|q , G)is more concentrated on a few topics in $T_G(q)$, or

iii) The topic in $T_G(q)$ are more similar to each other.

So, Profile-Granularity helps in justifying first two points by using the below evaluation.

$$PG(q,G)=\sum_{t\in TG(q)} Pr(t|q,G)\log\frac{Pr(t|q,G)}{Pr(t)}$$
$$=\underbrace{\sum_{t\in TG(q)} Pr(t|q,G)\,IC(t)}_{\text{First point (i)}}-\underbrace{H(t|q,G)}_{\text{Second point (ii)}} \quad (4)$$

c) Topic-Similarity (TS)

This measures the semantic similarity among topics in $T_G(q)$ as the third point (iii) suggests
This can be computed as Information Content of the Least Common Ancestor of $T_g(q)$ as follows:

$$TS(q,g)=IC(lca(Tg(q))) \quad (5)$$

It relays on the more specific common ancestor.

d) Discriminating Power (DP)

The discriminating power is a refined state of profile, and it is explained as normalized composition of PG (q,G) and TS(q,G)as follows:

$$DP(q,G)=\frac{\mathbf{PG(q,G)+Ts(q,G)}}{\mathbf{2}\sum_{t\in TH(q)} Pr(t|q,H)IC(t)} \quad (6)$$

e) Personalization Utility

It is defined as the gain of Discriminating Power achieved by exposing profile g together with query q,

$$util(q,G) = DP(q,G)-DP(q,R)$$

Here DP(q,R)is discriminating power of the query q without exposing any profile and it is obtained by simply replacing all occurrences of Pr(t|q,G)in (6) with Pr(t|q)

2) Metric of Privacy

The privacy risk when exposing G is defined as the total sensitivity contained in it, give in normalized form. In the worst case i.e., in the case of no sensitive data available, complete profile will be exposed.

Some sensitive nodes might be pruned (removed) after the generalization of profile, and then their ancestors will become as the leaf nodes. We have to calculate the risk of exposing these ancestors and it can done using the cost layer that which evaluated at (6). If the ancestor are leafs after pruning the risk is equals to the cost of t else the unnormalized risk of exposing the generalized profile is recursively given by.

$$Risk(t,G)=\begin{cases} cost(t) & \text{if tis leaf} \\ \sum_{t'\in C(t,H)} Risk(t',G) & \text{otherwise}\end{cases} \quad (7)$$

There are some cases where cost of non-leaf node is even greater than the total risk aggregated from its children. In this situations (7) might under estimate the risk. So, we amend the equation for non-leaf nodes as

$$Risk(t,G) = \max(cost(t),\sum_{t'\in C(t,H)} Risk(t',G)) \quad (8)$$

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

$$risk(q,G)=\frac{Risk(root,G)}{\sum_{s\in S} sen(s)} \quad (9)$$

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

Algorithm Of Proposed System

GreedyIL Algorithm

The GreedyIL algorithm improves the efficiency of the generalization using heurists (experience-based techniques which are good enough a given set of goals) based on several findings. GreedyIL involves a theorem and three heuristics. Let us explore them in detail.

1.1 Theorem:

Discriminating power of the profile displays monotonicity by prune-leaf.
"If 'G' is a profile obtained by appearing a prune leaf operation on g, then $DP(q,G) \geq DP(q,G')$"

Considering operation $G_i \xrightarrow{-t} G_{i+1}$ in the i^{th} iteration, maximizing DP(q, G_{i+1}) is equivalent to minimizing the incurred(become subject to) information loss, which is defined as DP (q,G_i) - DP (q,G_{i+1})

After finding information loss the all DPs, store the information loss values in a priority queue in descending order.The priority queue contains the information loss value which is caused by an operator.
Here, operator in the queue is a tuple like

$$op=<t,IL\ (t,G_i)>,$$

Where t is the leaf to be pruned by op and IL (t,G_i) indicates the IL incurred by pruning t from G_i. This queue, denoted by Q, allows fast retrieval of best so far candidate operator and as a result two heuristics which reduce the total computational cost.

1.1 Heuristic 1:
The iterative process can terminate whenever δ-risk is satisfied.
The second finding is the computation of IL can be simplified to the evaluation of $\Delta PG(q,G) = PG(q,G_i) - PG(q,G_{i+1})$, referring to equation(6), the second term (TS(q,G))remained unchanged for any pruning operation until a single leaf is left (In such case the only choice for pruning is the single leaf itself).

Consider two possible cases..
Case 1: t is a node with no siblings. And is easy to handle
Case 2: t is anode with siblings. It requires introducing a shadow sibling of t.

Each time if attempt to prune t, we actually merge t in to shadow to obtain a new shadow leaf shadow'
The shadow sibling is a dynamically generated to maintain the non over lapping property of $T_G(q)$.
The shadow sibling preferences are initialized to 0.In semantics, the shadow stands for ANY OTHER sub-topics of a topics s ∈ Gapart from those presented in C(s,G).

Thus, the probability of shadow can always be dynamically evaluated as $Pr(shadow)=Pr(s)-\sum_{t' \in C(s,G)} Pr(t)$

$$Pr(shadow'|q,g)=Pr(shadow|q,g)+Pr(t|q,g)$$

Finally, we have the given below heuristic, which significantly eases the computation of IL(t)it can be seen in equation(10) can be computed efficiently.

1.1 Heuristic 2:

$$IL(t)=\begin{cases} Pr(t|q,G)(IC(t))-IC(par(t,G))), & \text{case C1} \\ dp(t)+dp(shadow)-dp(shadow'), & \text{case C2} \\ \text{where} dp(t)=Pr(t|q,G)\log\frac{Pr(t|q,G)}{Pr(t)} \end{cases} \qquad (10)$$

As described in C1 case above, prune-leaf only operates on a single topic t. Thus, it does not affect the IL of other candidate operator in Q.
While C2, pruning t incurs recomputation of the preference values of its sibling nodes. Therefore, we have the given below Heuristic.

1.1 Heuristic 3:
Once a leaf topic t is pruned, only the candidate operators pruning t' sibling topics need to be updated in Q. In other words, we only need to figure out the IL values for operators attempting to prune t' sibling topics

VIII. Experimental Setup

The UPS framework is implemented on a PC with a Processor Intel core i3, Speed - 3.00 GHz; RAM - 2.00 GB running Microsoft Windows XP/7/8. All the algorithms are implemented using Java. The topic repository uses the DMOZ-ODP (open directory project) web Directory. To focus on the pure English categories, we filter out taxonomies "Root/World" and "Root/Adult/World." The click logs are downloaded from the online AOL query log, which is the most recently published data we could find.

Synthetic: We cluster AOL queries by their DP (discriminating power) into two groups using the 1-dimensional k-means algorithm. These two groups, namely Distinct Queries, Medium Queries, can be specified according to the following empirical rules obtained by splitting the boundaries between two neighboring clusters.

Distinct Queries for DP (q; R) ∈ 1
Medium Queries for DP (q; R) ∈ 0

Each synthetic profile is built from the click log of two queries, with one from each group. The forbidden node set S is selected randomly from the topics associated with the clicked documents.

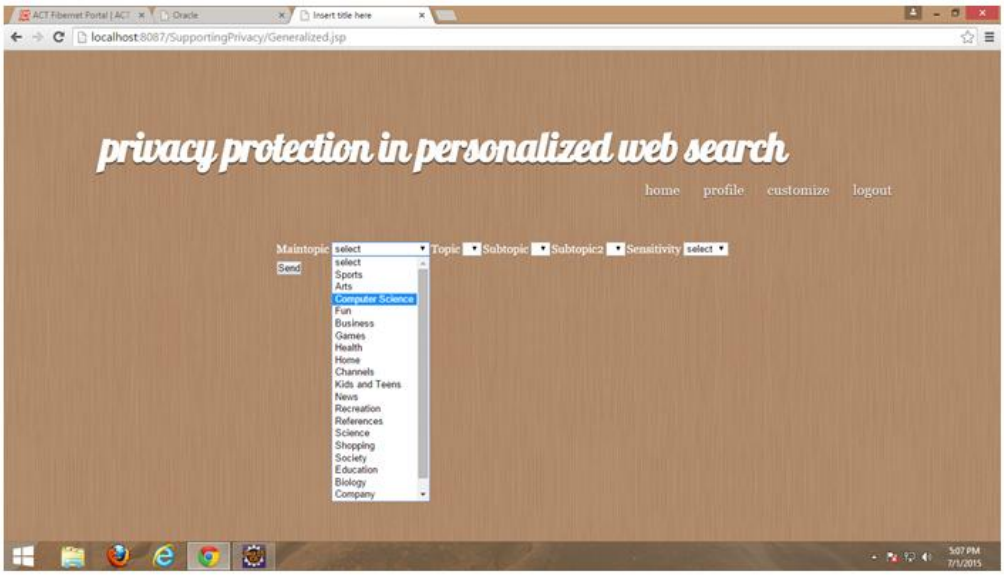


Fig.4 Screenshot of Privacy Protection in Personalized Web Search via Taxonomy Structure

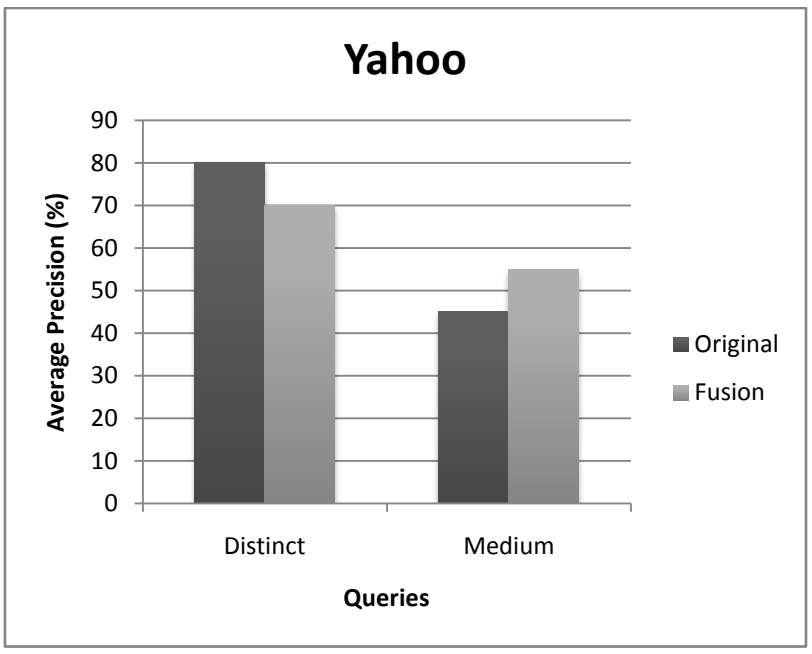
IX. Result And Analysis

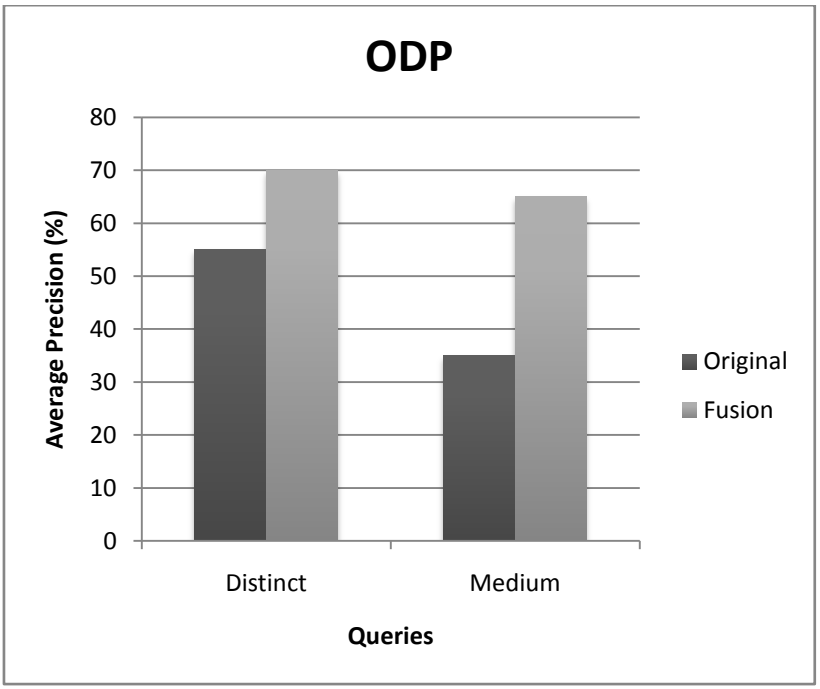
The following performance parameters are normally used in privacy protection technique evaluation. The existing system is compared with proposed system using these evaluation parameters. The system is estimated in terms of Precision and Recall.

Precision

It is defined as the number of relevant documents retrieved by a search divided by the total number of documents retrieved by that search. It is measure of correctly predicted documents by the system among all the predicted documents.

Precision = number of correct results/ number of all returned results.

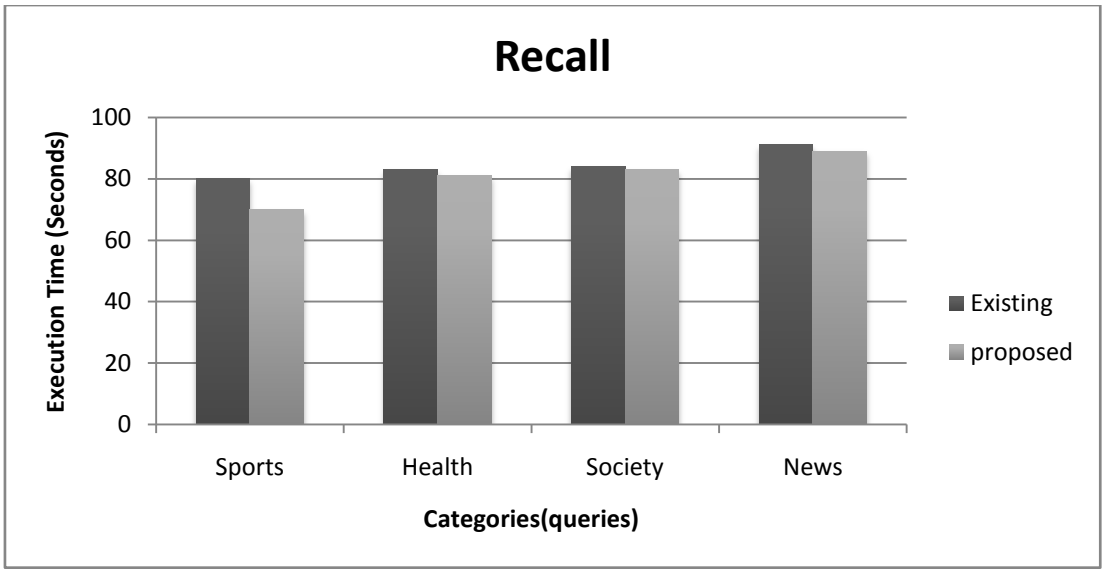




Evaluation of Precision using GreedyIL Algorithm
The proposed approach accuracy level is high when compared with the existing one.

Recall

Recall is defined as the number of relevant documents retrieved by a search divided by the total number of existing relevant documents. Recall is a measure of correctly predicted documents by the system among the positive documents.
Recall= no. of correct results/total no. of actual results



Evaluation of Recall using GreedyIL Algorithm
The proposed approach takes less time when compared with existing design.

X. Conclusions

This paper presented a client-side privacy protection framework called UPS (user customizable privacy preserving search) for personalized web search. UPS could be adopted by any PWS that captures user profiles in a hierarchical taxonomy structure. The framework allowed users to specify customized privacy requirements via the hierarchical user profiles. In addition, UPS also performed online generalization on user profiles to protect

the personal privacy without compromising the search quality. Our experimental results show that UPS could achieve quality search results while preserving user's customized privacy requirements and give the relevance result to the user by eliminating ambiguity of text. The results also confirmed the effectiveness of our solution.

For future work, we will try to stop adversaries with large background knowledge, such as richer relationship among topics (e.g., sequentiality, exclusiveness and so on), or capability to catch a series of queries from the victim. We will also use better metric to estimate the performance of UPS.

References

[1]. Lidan Shou, He Bai, Ke Chen, and Gang Chen,Feb 2014"Supporting Privacy Protection in Personalized Web Search," vol . 26, no. 2.

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

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

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

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

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

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

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

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

[10]. J. Pitkow, H. Schu¨tze, T. Cass, R. Cooley, D. Turnbull, A. Edmonds, E. Adar, and T. Breuel,2002 "Personalized Search," Comm. ACM, vol. 45, no. 9, pp. 50-55.

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

[12]. K. Hafner,Aug. 2006 Researchers Yearn to Use AOL Logs, but They Hesitate,

[13]. New York Times.

[14]. A. Krause and E. Horvitz,2010 "A Utility-Theoretic Approach to Privacy in Online Services," J. Artificial Intelligence Research, vol. 39, pp. 633-662.

[15]. J.S. Breese, D. Heckerman, and C.M. Kadie, 1998"Empirical Analysis of Predictive Algorithms for Collaborative Filtering," Proc. 14th Conf. Uncertainty in Artificial Intelligence (UAI), pp. 43-52.

[16]. P.A. Chirita, W. Nejdl, R. Paiu, and C. Kohlschu¨tter, 2005 "Using ODP Metadata to Personalize Search," Proc. 28th Ann. Int'l ACM SIGIR Conf. Research and Development Information Retrieval (SIGIR)

[17]. A. Pretschner and S. Gauch,1999 "Ontology-Based Personalized Search and Browsing," Proc. IEEE 11th Int'l Conf. Tools with Artificial Intelligence (ICTAI '99)

[18]. E. Gabrilovich and S. Markovich, 2006 "Overcoming the Brittleness Bottleneck Using Wikipedia: Enhancing Text Categorization with Encyclopedic Knowledge," Proc. 21st Nat'l Conf. Artificial Intelligence (AAAI).

[19]. K. Ramanathan, J. Giraudi, and A. Gupta, 2008"Creating Hierarchical User Profiles Using Wikipedia," HP Labs.

[20]. K. Ja¨rvelin and J. Keka¨la¨inen, 2007"IR Evaluation Methods for Retrieving Highly Relevant Documents," Proc. 23rd Ann. Int'l ACM SIGIR Conf. Research and Development Information Retrieval (SIGIR), pp. 41-48, 2000.

[21]. R. Baeza-Yates and B. Ribeiro-Neto, 1999, Modern Information Retrieval. Addison Wesley Longman.

[22]. X. Shen, B. Tan, and C. Zhai,2007 "Privacy Protection in Personalized Search," SIGIR Forum, vol. 41, no. 1, pp. 417.