

## EaZSearch

*Aayesha Mullani<sup>1</sup>, Deepali Patil<sup>2</sup>, Ishrat Tamboli<sup>3</sup>, Aishwarya Chougule<sup>4</sup>, Kiran Gavade<sup>5</sup>,*

*R. V. Chavan<sup>6</sup>*

*<sup>1-5</sup> Student, Department of Computer Sci. & Engg., DYPET, Maharashtra, India*

*<sup>6</sup>Lecturer, Department of Computer Sci. & Engg., DYPET, Maharashtra, India*

-----  
\*\*\*  
-----

**ABSTRACT** - *EaZSearch has useful for its effectiveness in increasing the quality of search on the Internet Service. However, in general history show that users' reluctance to disclose their secret data at the time of search has become a major problem. In EaZSearch application model where hierarchical user profiles are used. We develop one framework called UPSearch that can create profiles by queries while respecting user specified requirements. We presented two greedy algorithms, DP\_Algo and IL\_Algo. We also provide an prediction technique for deciding whether personalizing a query is beneficial. The results also proves that IL\_Algo significantly outperforms Dp\_Algo in terms of efficiency.*

**Key Words:** *Privacy protection, EaZSearch, utility, risk, Web search, User profile, information.*

### 1. INTRODUCTION

When amount of information on the internet increases rapidly, it creates many new challenges or problems for Web search. When the same query is fired by different users, a general search engine gives the same out, without knowing of who submitted the query. This output may not be suitable for users with different required information needs.

The web search engine has long become the most important door for searching information on the web. However, users might got failure when search engines return unexpected results that do not meet their real requirement. Such unexpected results are due to the excessive variety of users' contexts and backgrounds, as well as the ambiguity of texts. EaZSearch is a general category of web search techniques aiming at providing better search results, which are tailored for user needs. As the expense, user information has to be collected and analyzed to find out the user intention behind the entered query.

### 2. EXISTING SYSTEM

Now days to protect user privacy in profile-based search is becoming very difficult, the researchers are considering two effects during the search process. First, they try to improve the search quality with the personalization utility of the user profile. And secondly, they have to hide the private contents present in the user profile to consider the privacy risk. The problems with the existing systems are explained in the following observations:

**2.1. The present profile-based search engines do not support runtime profiling.** A user profile is typically made available for only once offline, and used to personalize all queries from a same user without any judgments. This strategy certainly has drawbacks which give rise to variety of queries. The only solution to these problem is online decision on:

- a. do we have to personalize the query (by considering the profile) and
- b. what to considering in the user profile at runtime.

**2.2. The existing search engines do not consider the customization of privacy requirements.**

Because of which some of the personal data of user is to be overprotected while others are insufficiently protected. For example if any user wants to perform an query for personal use will not have any privacy regarding it, and that all will be insufficiently protected which may lead to users privacy risk.

**2.3. Many existing search engines techniques require repeated user interactions when user wants to creating personalized search.**

Many of the searches require multiple user interactions. This will take too much risk of privacy, but also demand processing time for profiling.

**3. LITERATURE SURVEY****Privacy –Enhanced Web Personalization: [3]**

**Author** – Rupali Keshavrao Aher, Akshay Rajdhar Adik

For users web personalization is used to improve search quality by customizing search results, based on the personal data of user provided to the search engine. Thus, there is balance between the search quality and privacy protection. Two greedy algorithms, namely Greedy discriminating power (GreedyDP) and Greedy Information Loss algorithm (GreedyIL), are used for runtime generalization. GreedyIL algorithm achieves high efficiency than the GreedyDP algorithm.

**Supporting Privacy Protection in Personalized Web Search: [1]**

**Author** – Lidan Shou, He Bai, Ke Chen, and Gang Chen

Personalized web search (PWS) has demonstrated its effectiveness in improving the quality of various search services on the Internet. However, evidences show that user's reluctance to disclose their private information during search has become a major barrier for the wide proliferation of PWS. We study privacy protection in PWS applications that model user preferences as hierarchical user profiles. We propose a PWS framework called UPS that can adaptively generalize profiles by queries while respecting user specified privacy requirements. Our runtime generalization aims at striking a balance between two predictive metrics that evaluate the utility of personalization and the privacy risk of exposing the generalized profile.

**Enhance Search In Web Search Engine Using Greedy Algorithm: [2]**

**Author** – V. Ramya, S. Gowthami

UPS stands for User customizable Privacy-preserving Search framework. The framework assumes that the queries do not contain any sensitive information, and aims at protecting the privacy in individual user profiles while retaining their usefulness for PWS. UPS consists of a non-trusty search engine server and a number of clients. Each client (user) accessing the search service trusts no one but himself/ herself. The key component for privacy protection is an online profiler implemented as a search proxy running on the client machine itself. The proxy maintains both the complete user profile, in a hierarchy of nodes with semantics, and the user-specified (customized) privacy requirements represented as a set of sensitive-nodes.

**Automatic Identification of User Interest For Personalized Search: [5]**

**Author** – FengQiu, Junghoo Cho

The key idea behind Page Rank is that, multiple web pages are available on web but user wants only some pages then user always prefers famous or well-known pages. In this paper two techniques are used. Various formulas are provided in to calculate Page Rank.

**1. Simple Page Rank:**

However, in real life, a user does not follow links all the time in web browsing; she sometimes types URL's directly and visits a new page. The computed Page Rank values are used by search engines during query processing, such that the pages with high Page Rank values are generally placed at the top of search results.

**2. Topic-Sensitive Page Rank:**

The Topic-Sensitive Page Rank scheme (TSPR) proposed is an interesting extension of Page Rank that can potentially provide different rankings for different queries, while essentially retaining the efficiency advantage of the standard Page Rank.

**4. PROPOSED SYSTEM**

In the proposed system the users have the facility to create their own profile by considering their profession. These will help user easily to find out any query related to any data which will be generated by taking the users profile into consideration. The user will be getting their own personalizes EaZSearch.

These will also help to reduce the repeated work and repeated interaction of user with search engines as explained in existing system.

**Admin Login:**

This module provides the admin with login facility with their login id and password, in that admin can maintain the profiles with the consideration of user interest or maintain the data according to that profile and that data will be provide to the user with respect to profile defined by user and also Admin can be able to view the user information with their defined profile

**User Registration:**

This module provides registrations facility to user. The user must register to the website by entering their personal information and after successful registration, user personal account will be created so that user can able to login to their personal account with the own user id and password, in that user should select the profile which maintained by admin on which it wants data and also user can able to update that define profile

**User customizable Privacy-preserving Search (UPS):**

In this module when the user enters query based on profiles which should be defined in their personal account and the system will provide the all required information maintained by admin based on that user entered query.

**User Log Collection:**

Here admin will store logs which are maintained by ranking the websites that are more visited by users. Here if any other user enters the same query which is referred by previous user then by user customizable search user will get all required data with data which mostly preferred by other user.

#### 4.1 Block Diagram

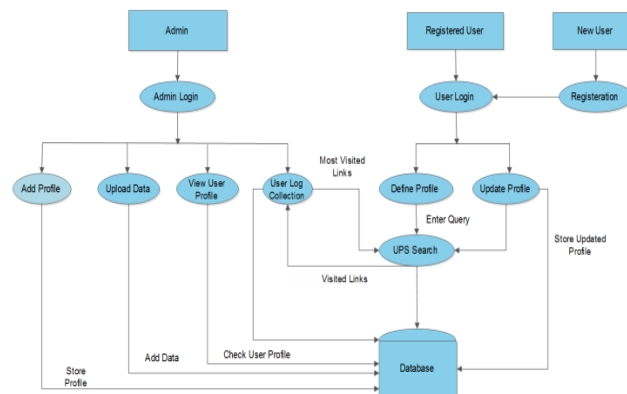


Figure 1-: Block Diagram

The block diagram describes overall structure of the system. It includes three users' of system namely admin, registered user and new user. The First user is the admin of system which administrate the overall performance of the system during administrating the performance of the system admin has given authorities for performing the tasks like viewing the user profile, uploading data, store any changes related to user data as well.

The another user of system is the new user which will register to the system for performing further tasks such as searching any query or defining own profile etc. The new user is became registered user after performing registration which we can call it as third user of the system. Now registered user can perform same tasks as UPS search and can also update the profile and all changes are stored in database.

#### 4.2 System Architecture

The below diagram shows the system component of the system which includes the client side and server side. There are components like the user of the system, profile, privacy factor and the multiple generalized profiles as G\* of user have been created on the basis of user profile.

The user of the system is performing UPS search after registration, login and creating the profile based on his/her own requirement.

The user of system first enter the query then the query is submitted to database and then admin of the system will fetches the appropriate result by referring the user generalized profile as G\* which was created at the time of registration and the result will given to user in the form of list of page link.

The user of system then go through the list and by referring the page link it will obtain the appropriate result which he/she will expect. The user can refer multiple page links at time and definitely the system generates the appropriate result.

If in any case the result will not stored within the system then admin of the system can perform the Google Search and collect the appropriate results from the Google and store in the database. Then it will be given to the user of the system.

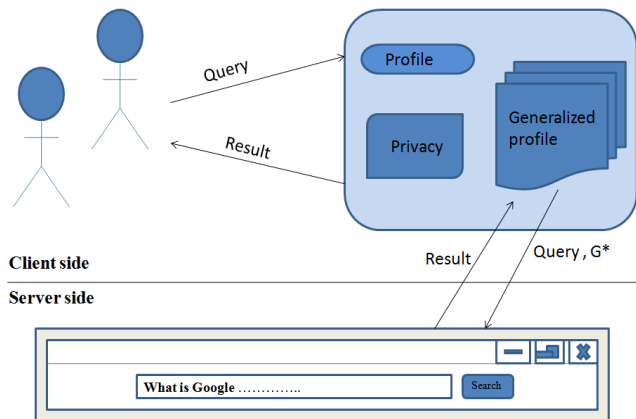


Figure 2: System Architecture

## 5. ALGORITHMS

### 5.1 IL\_Algo ( $\partial, k, \epsilon$ ) [4]

**Input:** Peak profile  $G_0$  ; Query  $k$ ; threshold value  $\epsilon$

**Output:** Generalized profile  $G^*$  satisfying  $\epsilon$ -Risk

**let**  $D$  be the IL-priority queue of node-leaf decision;

$i$  be the iteration variable, initialized to 0;

**if**  $DP(k, R) < \alpha$  **then**

Obtain the peak profile  $G_0$  from online 0;

Insert  $\{e, IL(e)\}$  into  $D$  for all  $e \in T\partial(q)$ ;

**While**  $risk(k, G_i) > \epsilon$  **do**

Pop a node leaf operation on  $e$  from  $D$ ;

Set  $n \leftarrow part(e, G_i)$ ;

Process node-leaf  $G_i \rightarrow G_{i+1}$ ;

**if**  $t$  has no siblings **then**

insert  $\langle n, IL(n) \rangle$  to  $D$ ;

**else if**  $n$  has siblings **then**

merge  $n$  into  $shadow\_siblings$ ;

if No operation on t's siblings in D then

insert  $\langle e, IL(e) \rangle$  to D;

else

update the IL-values for all operation on

n's sibling in D;

update  $i \leftarrow i+1$ ;

**return**  $G_i$  as  $G^*$ ;

**return** root (D) as  $G^*$ ;

### 5.2 DP\_Algo [4]

Step 1: Defined an operator  $\rightarrow$  called node leaf

Where  $\rightarrow =$  indicates the removal of a leaf topic, t  
from a profile.

Step 2: Mathematical Equation

$G(i) \rightarrow G(i+1) \dots \dots \dots G = \text{Seed Profile}$

which denotes, to obtain leaf t from profile  $G(i)$  to  $G(i+1)$ .

Step 3 :  $G^*$  can be generated with a finite length transitive closure of prune leaf.

Step 4: It works in bottom up manner initial point  $G(0)$

$(i=0,1,\dots,n)$ .

GreedyDP chooses leaf topic

$t \in T$

$G_i(q) \dots \dots \dots t = \text{Leaf Profile}$

$q = \text{node}$

maintains Best Profile  $(G_{i+1})$  which has highest discriminating power to satisfy  $(\delta)$  risk constraint.

Note: Iterative process terminates when the profile is generated to a root topic.

Step 5: Best Profile

G\*=Final result of algorithm.

## 6. SET THEORY

Let **S** represent our proposed system

$$S = \{I, O, Su, Fa, \emptyset\}$$

Where,

**I** = Input, **O** = Output, **Fa**= Failure, **Su** = Success

**I** is Input **I** = {**U,Q,P**}

Where,

**U** = User information, **Q** = Query, **P** = Updated Profile

**O** is Output **O** = {**Ur, Vr**}

Where,

**Ur** = UPS search result, **Vr** = Visited link result

**Su** is Success **Su** = **SR**

Where,

**SR** = Successful result Received

**Fa** is Failure **Fa**= **RF**

Where,

**RF** = Result not found

### Actual inputs of the System :-

**I** is Input **I** = {**U,Q,P**}

Where,

**U** = User information

**Input** :- Filled information

**Output** :- Registration will be done

**Success** :- Succeeded to filled correct information

**Failure** :- Fails to fill correct information

#### Q = Query

**Input** :- Typed Query

**Output** :- Query matched

**Success** :- Successfully match query result

**Failure** :- Query result not matched

#### P = Updated Profile

**Input** :- Created user profil

**Output** :- Found Profile updated

**Success** :- Successful updated profile

**Failure** :- Profile unable to update due to filled up invalid information

#### Actual outputs of the System

O is Output  $O = \{Ur, Vr\}$

Where ,

Ur = UPS search result

**Input** :- Typed Query

**Output** :- Query matched and result found

**Success** :- Successful found result

**Failure** :- Fails to found result

Vr =Mostly Visited links Result



**Input** :- Ranking process

**Output** :- Mostly ranked links list

**Success** :- Successfully found ranked links list

**Failure** :- Fails to found ranked links list

$\emptyset$  is constraints of system  $\emptyset = \{PM, PE\}$

Where,

**PM**= Possibility of query will not match with stored database

**PE** = Possibility of user will not enter appropriate query according to their profile

### 6.1 Mathematical model using “Deterministic Finite Automata”

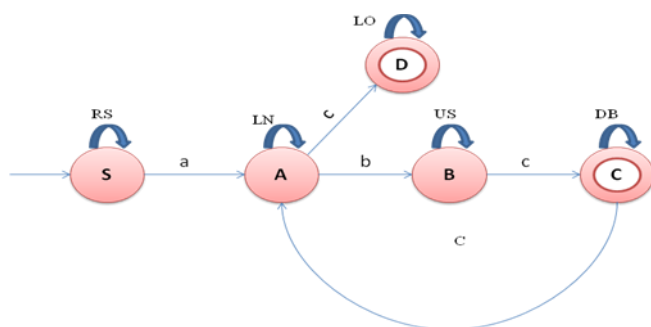


Figure 3: Deterministic Finite Automata (DFA)

Where,

**RS** = Registration

**LN** = Login

**US** = UPS search

**DB** = Database

**LO** = Logout

#### **Deterministic finite state automata**

A deterministic finite automata M is a 5-tuple,  $(Q, \partial, \Sigma, q_0, F)$  consisting of following fields :-

A finite set of states  $(Q) = \{ S, A, B, C, D \}$

A finite set of input symbols called the alphabet  $(\Sigma) = \{ a, b, c, d, e \}$

A transition function  $(\partial : Q \times \Sigma \rightarrow Q) = \{ S, LN, US, DB, LO \}$

A start state  $(q_0 \in Q) = S$

A set of accept states  $(f \subseteq Q) = D$

Where,

RS- Registration, LN- Login, US- UPS search,  
 DB- Database, LO- Logout St (A,B,C), Alp- alphabets.  
 Derivation() is defined by below transition table.

St/trans	S	LN	US	DB	LO
A	a	∅	∅	∅	∅
B	∅	b	∅	∅	∅
C	∅	∅	c	∅	∅
D	∅	∅	∅	∅	∅

### 7. Multiplexer Logic

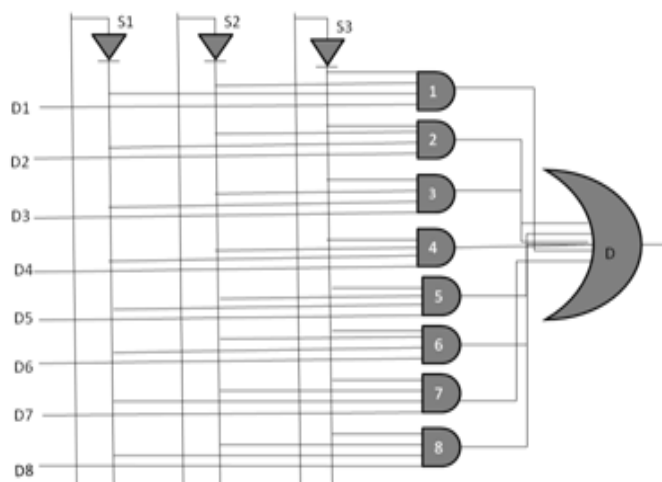


Figure 4: Multiplexer Logic

Where,

S1 ,S2, S3 = select lines

D = data lines

Here Input I=(S1 ,S2 S3)

Where,

S1 = Registration

S2 =Login

S3 = UPS search

Here O= Output

## 7. CONCLUSION

At client side privacy is maintained by preventing the users private account. Any EaZSearch can create user profile in hierarchical tree view. EaZSearch allows user to specify the privacy requirement of client and thus the personal data of user profile is kept private without compromising the quality of search query. EaZSearch implements two algorithms DP\_Algo and IL\_Algo.

## REFERENCES

- [1]LidanShou, He Bai, Ke Chen, and Gang Chen,“*Supporting Privacy Protection in Personalized Web Search*”, IEEE Transactions On Knowledge And Data Engineering Vol:26 No: 2 Year 2014.
- [2]RupaliKeshavraoAher, AkshayRajdharAdik,“*Privacy Enhanced Web Personalization*”, IEEE Transactions On Knowledge And Data Engineering Vol:50 No: 5 Year 2015.
- [3]V.Ramya,S.Gowthami,“*Enhance Search In Web Search Engine U sing Greedy Algorithm*”, IEEE Transactions On Knowledge And Data Engineering Vol:89 No: 3 Year 2014.
- [4] Based on a handout by Tim Roughgarden, Alexa Sharp, and Tom Wexler, “*Guide to Greedy Algorithms*”.
- [5] FengQiu, Junghoo Cho, “*Automatic Identification of User InterestFor Personalized Search*”, IEEE Transactions On Knowledge And Data Engineering Vol:98 No: 4 Year.