International Journal of Scientific Engineering and Research (IJSER)
www.ijser.in
ISSN (Online): 2347-3878, Impact Factor (2015): 3.791

Promoting Privacy Protection, in Customized Web Search with Anchored User Profile

P. Hena Monica Anand[1], S. Radha Krishna[2]

[1]Jawaharlal Nehru Technological University, Department of CSE, M.tech Scholar, Dwarapudi, Vizianagaram, Andhra Pradesh

[2]Jawaharlal Nehru Technological University, Department of CSE, *Assistant Professor*, Dwarapudi, Vizianagaram, Andhra Pradesh

Abstract: ***Web search engines are widely used to find certain data among a huge amount of information in a nominal amount of time. We can implement the String Matching KMP Algorithm for improving the better search quality results. To come up with this privacy threat, present solutions introduce new mechanisms that precede a high cost in terms of computation and communication. Personalized search is a promising means to obtain better accuracy of web search, and has been attracting more attention lately. Nevertheless, effective, personalized search needs collecting and aggregating user data, which often increases serious concerns of privacy infringement for many users. Indeed, these businesses have become one of the main barriers for deploying personalized search applications, and how to do privacy-preserving personalization is a big challenge. In this, we introduce and attempt to resist adversaries with more extensive background knowledge, such as richer relationship amongst topics. Richer relationship means we generalize the user profile results by applying the background knowledge which is starting to save in history. Through this, we can hide the user search results. By applying this mechanism, we can achieve the privacy.***

Keywords: Privacy protection, PWS, risk, String Matching KMP

1. Introduction

Accurately measure the semantic similarity between word is an important problem in web mining, information retrieval, and natural speech processing. Web mining applications such as community extraction, relation detection, and entity disambiguation; require the ability to accurately assess the semantic similarity between concepts or entities. In information retrieval, one of the principal problems is to retrieve a set of documents that is semantically linked to a given user query. Efficient approximation of semantic similarity between words is critical for several natural language processing projects such as word sense disambiguation (WSD), textual entailment, and automatic text summarization. Semantically related words are listed in manually created general-purpose lexical Ontologies such as WordNet.1 In WordNet, a synonym contains a set of synonymous words for a particular sense of a word. However, semantic similarity between entity changes over time and across disciplines. For instance, an apple is often related to computers on the Network. Withal, this sense of apple is not listed in most general-purpose thesauri or dictionaries. A user who searches for apple on the web [1] [2], might be interested in this sense of apple and not evoke as a yield. New words are constantly being made as easily as new senses are assigned to existing language. Manually maintaining ontologies to capture these new words and senses is dear if not unacceptable. We provide an automatic method to guess the semantic similarity between words or entities using WSE. Because of the vastly numerous documents and the high development rate on the web, it is time- consuming to examine each document individually. WWW search engines provide an efficient interface to this vast information. Page counts and snippets are useful information sources offered by most web search engines. The Page count of an inquiry is an approximation of the number of pages that hold back the query [3] words. In general, page count may not necessarily be equal to the word because the queried word might appear many times on one page. Page count for the query P AND Q can be reckoned as a global measure of co- occurrence of words P and Q. For example, the page count of the query “apple” AND “phone” in Google is 288,000,000, whereas the same for “banana” AND “phone” is only 3,590,000. The more than 100 times more numerous page numbers for “apple” AND “phone” indicate that Apple is more semantically similar to phone than is a banana. Despite its simplicity, using page counts only as a measure of co-occurrence of two words presents several disadvantages. Firstly, the page count analysis ignores the position of a word in a page. So, even though the two words appear on a page, they might not be really reached. Second, the page count of a word with multiple senses might need a combination of all its sensations. For example, page count for apple contains page counts for apple as a fruit and apple as a fellowship. Moreover, given the scale and noise along the web, some words might co-occur on some pages without being actually affected. For those reasons, page counts are unreliable when measuring semantic similarity.

1.1 Basics of Personalized Search

1.1.1. Creation of User Profile

To offer personalized search results to users, personalized web search maintains a user profile for each person. A user profile stores information about user interests and predilections. It is generated and updated by exploiting user related data. Such information may include:

- Information about the user like age, gender, education, language, country, place, interest, areas, and other info.
- Search history, including previous questions and clicked documents.
- Other user documents, such as bookmarks, favorite sites, visited pages, and emails.

1.1.2 Server-Side and Client-Side Implement:

PWS can be implemented on either server side (in the browser) or client side (on the user’s computer). For server-side personalization, user profiles are constructed, updated, and stored on the search engine side. User information is now incorporated into the ranking processor is used to help initial search results. The benefit of this architecture is that the search engine can utilize all of its resources, in its personalization algorithm. Besides, the personalization algorithm can be easily adjusted without any client efforts. This architecture is assumed by some universal search engines such as Google Personalized Search. The drawback of this architecture is that it brings high storage and computation costs when millions of users are using the search browser, and it also brings up privacy concerns when information about users is stored on the host. For node-side personalization [4] [5], user data are compiled and stored on the customer side (personalization agent), usually by installing a node software on a user's computer. In client side, not exclusively the user's search, but also his contextual activities (e.g., history) and personal information (e.g., emails, documents, and bookmarks) could be integrated into the user profile. This permits the building of a much richer user model for personalization. Privacy worries are also reduced since the user profile is strictly stored and utilized on the [2] guest side. Some other benefit is that the overhead in the calculation and storage for personalization can be broadcast among the nodes. A primary drawback of personalization on the customer side is that the personalization algorithm cannot use any knowledge that is exclusively available on the server side. Furthermore, due to the limits of network bandwidth, the client can usually only process limited top results.

1.2 Benefits of Privacy Protection

Privacy and the Internet have impacted people in many positive ways. The Internet itself has allowed us to deliver more serious communication and increase people's education. The Internet has had an astonishing effect on society as a whole and has allowed people to get instant news, encounter fresh people, apply for jobs, shop, study books, and has allowed people to do many other things.

The Internet has allowed for openness and for people to share their thoughts with other masses. Through the Internet, people can now share their ideas all over the world without having to be published, broadcasted, or manage anything else. Matters such as blogs, Facebook, and other social networks allow people to share information with others. Delivering the Internet affords people a chance to not only provide their ideas, but see other people's as well.

Another advantage of the internet is communication. People can now pass on using email, social nets, chat rooms, and many other affairs with their kinship groups and friends. These services provided by the Internet allow society to maintain better contact with people at a more consistent, quicker pace. It also allows people to communicate with others across the world.

1.3 Motivation

Researchers have to weigh two main contradicting effects during the search, for protecting user's privacy in profile-based PWS. On the single hand, they prove to prove to ameliorate the quality of hunt with the personalization utility of the user profile. On the other hand, to place the privacy risk under control, they need to cover the privacy contents existing in the user profile. A few previous studies indicate that people are trying to compromise privacy if the personalization is made out by providing user profile to the search engine [7] yields better search quality.

In universal, personalized search is supposed as one of the most promising techniques to start around the limitation of search engines and improve the quality of search results. Consequently, without compromising the personalized [6] search quality user privacy can be protected. In general, there is a tradeoff between the level of privacy protection and search quality which is reached by induction. Unluckily, the last works of privacy preserving PWS are far from optimal.

2. System Anlaysis

2.1 Problem Statement

Retrieving accurate Data for users in Search Engine faces a great deal of troubles. This is due to accurately evaluate the semantic similarity between words is an important issue. For instance, the word "apple" consists of two substances, one show the fruit apple and the other is the apple company. So retrieving accurate information to users to such kind of similar words is challenging. An architecture and method to measure semantic similarity between words. Which consists of snippets, page-count and SVM. We are working to implement and work out the semantic similarity between words in Search browser without using Snippets or SVM. Because using Snippets or Support Vector Machines makes the task of finding similarity easier. Then we are starting to carry out the same concept without using snippets or support Vector machines.

2.2 Existing System

A privacy-preserving PWS framework UPS, which can generalize profiles for each question, according to user-specified privacy essential. Relying on the definition of two conflicting metrics, namely personalization use and privacy risk [1], for hierarchical user profiles, we state the problem of privacy-preserving PWS as Risk Profile Generalization, with its NP-hardness proved.

We prepare two simple but effective generalization algorithms, GreedyDP and GreedyIL, to support runtime profiling. While the former attempts to maximize the discriminating power (DP), the latter seeks to minimize the information loss (IL).

We offer an inexpensive mechanism for the customer to settle whether to personalize a query in UPS. This finish can be cleared before each runtime profiling to enhance the stability of the search results while avoid the unnecessary exposure of the profile.

2.2.1 Disadvantages of Existing System:

- Users might experience failure when search engines return irrelevant results that do not match their actual aims.

- Such irrelevance is largely due to the tremendous variety of users' contexts and settings, as well as the ambiguity of the texts.
- The existing profile-based PWS does not support runtime Profiling.
- The existing methods do not bring into account the customization of privacy requirements.
- Personalization methods require iterative user inter react when creating personalized search results.

All the sensitive topics are found using an absolute measure called surprise based on the information theory.

2.3 Proposed System

A central characteristic feature of transaction data is the extreme scarcity, which hands over any single technique ineffective in anonym zing such data. Among recent works, some incur high information loss, some result in data hard to interpret, and some suffer from performance disadvantages. This story proposes to integrate generalization and compression to reduce data loss. Nevertheless the integration is non-superficial. We propose novel techniques to address the efficiency and scalability challenges.

2.3.1 Advantages of Proposed System

Our proposed system gives better quality results and gives more efficiency. Privacy is too good when compared with the Existing system. In the Existing System, an only generalization technique is used. Our String matching algorithm gives more accuracy when compared with the Greedy IL algorithm. Generalization and suppression technique achieves better privacy when compared with the existing arrangement. We can carry out the hierarchical divisive approach for retrieving the search results. It will afford better performance when compared to our proposed System.

3. Implementation

3.1 Module Description

- Profile-Based Personalization.
- Privacy Protection, in PWS System**.**
- Generalizing User Profile.
- Online Decision.

3.1.1 Profile-Based Personalization

This paper presents an approach to personalize digital multimedia content based on user profile data. For this, two main techniques were developed: a profile generator that naturally creates user profiles representing the user behavior, and a content-based recommendation algorithm that calculate the user's interest in unknown content by holding in her profile to metadata descriptions of the substance. Both appearances are integrated into a personalization system.

3.1.2 Privacy Protection, in PWS System

We prefer a PWS framework called UPS that can generalize profiles in for each question, according to user-specified privacy requirements. Two predictive metrics are suggested to assess the privacy breach risk and the query utility for a hierarchical user profile. We prepare two simple but effective generalization methods for user profiles allowing for query-level customization using our proposed metrics. We likewise offer an online prediction mechanism based on query utility for deciding whether to personalize a query in UPS. Extensive tests demonstrate the efficiency and effectiveness of our theoretical account.

3.1.3 Generalizing User Profile

The generalization process has to meet specific necessity to handle the user profile. This is achieved by pre-processing the user profile. At the beginning, the process initializes the user profile by reading the indicated parent user profile into account. The process adds the inherited equity to the attributes of the local user profile. Thenceforth the process loads the data in the foreground and the background of the map according to the described choice in the user profile.

3.1.4 Online Decision

The profile-based personalization gives little or even cuts back the search quality, while revealing the profile to a server would for sure risk the user’s privacy. To come up with this problem, we acquire an online mechanism to resolve whether to personalize a query. The basic idea is pretty straightforward. If a distinct question is identified during generalization, the entire runtime design will be aborted and the query will be transmitted to the server without a user profile.

3.2 String Matching Algorithm

A simple but inefficient means to find out where one string occurs in another is to check each place it could be, one by one, to get word if it's there. Then first we discover if there's a transcript of the needle in the initiatory part of the haystack; if not, we look to find out if there's a transcript of the needle popping out at the second part of the haystack; if not, we look starting at the tertiary part and thus. In the normal case, we just have to await at one or two parts for each wrong side to ensure that it is a wrong position, so in the average case, this takes O ($n + m$) steps, where n is the distance of the haystack and m is the length of the needle, but in the worst case, searching for a string like "aabbb" in a string like "aaaaaabbbb", it takes O(nm).

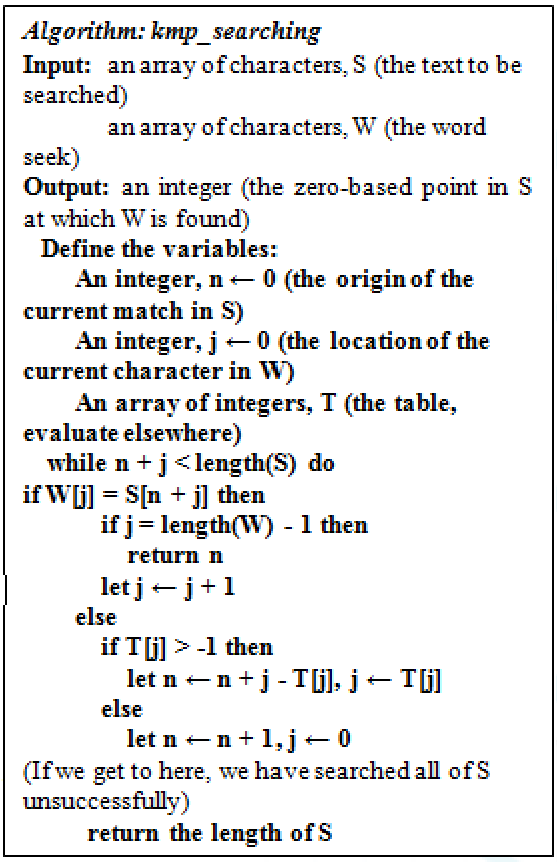
3.3 KMP Algorithm

KMP(Knuth, Morris and Pratt) invents first linear time string-matching algorithm by following a rigorous analysis of the naïve algorithm. KMP(Knuth-Morris-Pratt) algorithm keeps the data that naïve approach wasted gathered during the scan of the text. By keeping off this waste of information, it achieves a moving time of O (n + m), which is optimal in the worst case sense. That is, in the worst case KMP algorithm we have to analyze all the references in the text and pattern at least once.

3.4 Input Design

The input design is the tie-in between the information system and the user. It comprises the developing specification and processes for data provision and those steps are necessary to put transaction data into a usable form for processing can be accomplished by inspecting the data processor to record data from a written or printed document or it can occur by having people keying the information directly into the system. The design of input focuses on controlling the sum of input required, controlling the errors, avoiding delays, avoiding extra steps and holding open the operation simple. The input is planned in such a manner so that it offers security and simplicity of use with retaining the secrecy. Input Design considered the following things:

- What information should be passed as input?
- How the data should be arranged or coded?
- The dialog to run the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error happens.



```
Algorithm: kmp_searching
Input:  an array of characters, S (the text to be searched)
        an array of characters, W (the word seek)
Output: an integer (the zero-based point in S at which W is found)
  Define the variables:
    An integer, n ← 0 (the origin of the current match in S)
    An integer, j ← 0 (the location of the current character in W)
    An array of integers, T (the table, evaluate elsewhere)
   while n + j < length(S) do
if W[j] = S[n + j] then
      if j = length(W) - 1 then
         return n
      let j ← j + 1
    else
      if T[j] > -1 then
         let n ← n + j - T[j], j ← T[j]
      else
         let n ← n + 1, j ← 0
(If we get to here, we have searched all of S unsuccessfully)
     return the length of S
```

3.5 Output Design

A character output is one, which matches the demands of the end user and gives the information clearly. In any system outcomes of processing are conveyed to the users and to other system through outputs. Methodical and intelligent output design betters the system's relationship to help user decision-making.

1) Designing computer result should get on an organized, well thought out manner; the right end product must be developed while ensuring that each production element is planned so that people will find the system can use easily and adequately. When analysis, design, computer result, they should Identify the specific output that is demanded to satisfy the demands.
2) Produce documents, accounts, or other formats that contain data produced by the organization.

The output phase of an information system should achieve one or more of the following objectives.

- Communicate information about past actions, current status or projections of the Future.
- Signal important events, opportunities, troubles, or warnings.
- An interrupt occurs before the triggering action.
- Confirm an action.

4. Methodology

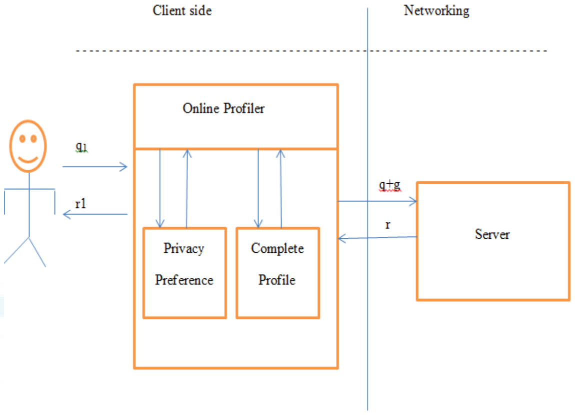


Figure 1: System Architecture of UPS.

As shown in UPS consists of an act of clients/users and a server for fulfilling the client's request. In client's machine, the online profile is implemented as a search proxy which preserve users profile in the hierarchy of nodes and also support the user specified privacy requirement as a set of sensing nodes. There are two phases, namely Offline and Online phase of the framework. During Offline, a hierarchical user profile is created and user specified privacy requirement is checked off on it. The query fired by the user is handled in the online phase as: When user fires a query on the client, proxy generates user profile at run time. The production is a generalized user profile considering the secrecy requirements. And so, the query along with generalized profiles of the user is transmitted to the PWS server for personalized web search. The search results are personalized and the response is transmitted back to query proxy. Lastly, the proxy shows the raw result or re-ranks them with user profile.

Fig 1 When a user issues a query q1 on the node, the proxy generates a user profile in runtime in the light of question terms. The output of this measure is a generalized user profile G1 satisfying the privacy demands. The generalization process is guided by considering two different metrics, namely the personalization utility and the privacy risk, both set in user profiles. Afterwards, the questions and the generalized user profile are directed together to the PWS server to personalized search. The search outputs are personalized with the profile and conveyed back to the query proxy. Lastly, the proxy either presents the raw outputs to the user, or ranks them with the complete user profile.

5. Expermental Results

In this part, we introduce the observational results of UPS. Here, we look at the effectiveness of the proposed query-topic mapping. Next, we study the scalability of the proposed algorithms in terms of response time. In the Third experiment, we study the efficiency of clarity prediction and the search quality of UPS.

5.1 Experimental Setup

The UPS framework is implemented on a PC with Intel CORE i3 with 1.1GHz and 2GB main memory, running Microsoft Windows XP. All the algorithms implemented in Java.

The profiles used in our tests can be either synthetic or generated from real question logs:
Synthetic- These groups, namely Distinct Queries, Medium Queries, and Ambiguous Queries, can be set according to the following empirical rules obtained by collapsing the bounds between two neighboring clusters.

- Distinct Queries for DP
- Medium Queries for DP
- Ambiguous Queries for DP

Every synthetic profile is built from the click log of three queries, with one from each group. The closed node set S is chosen arbitrarily from the topics associated with the clicked documents.

Real-The real user profiles are extracted from 50 distinct user click logs from AOL. For each user, the user profile is built with the evidences dumped from all urls in his/her log.

5.2 Efficiency of Generalization Algorithms

To examine the efficiency of the proposed generalization algorithms, we perform Optimal and KMP algorithms on real profiles. The questions are taken selected from their respective query log We give the outcomes in terms of mean number of iterations and the reaction time of the generalization.

Fig. 2 Shows the results of the experiment. For comparability, we also plot the logical number of iterations of the Optimal algorithm. It can be seen that KMP algorithm outperform Optimal. The greater the privacy threshold, the less iterations the algorithm calls off.

The advantage of KMP is more obvious in terms of response time, as Fig.3 shows. This is because KMP requires much more recomputation of DP, which gets lots of logarithmic operations. The problem worsens as the inquiry gets more ambiguous. For instance, the average time to process KMP for queries in the ambiguous group is more than 7 seconds.

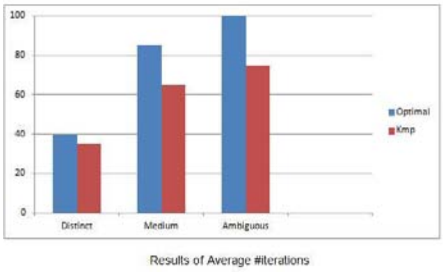


Figure 2: Results of Average Iterations

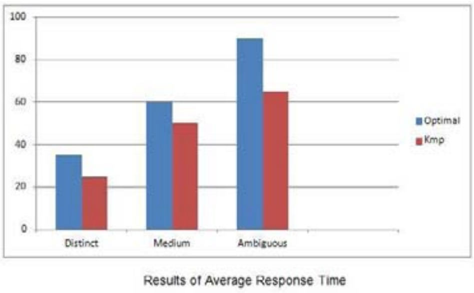


Figure 3: Results of Average Response Time

5.3 Scalability of Generalization Algorithms

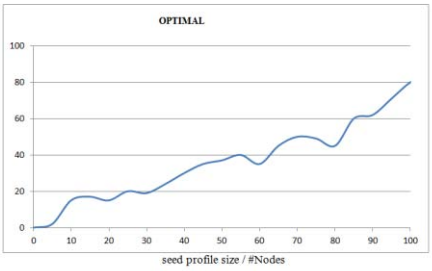


Figure 4: Results of seed profile Optimal

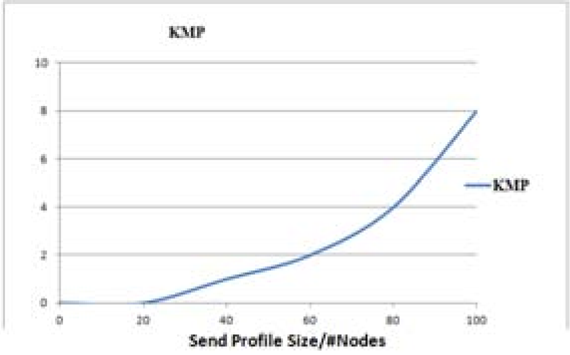


Figure 5: Results of seed profile KMP

We consider the scalability of the proposed algorithms by varying 1) the seed profile size (i.e., a number of nodes), and 2) the dataset size (i.e.,a number of queries). For each possible seed profile size (ranging from 1 to 108), we randomly choose 80 queries from the query log and convey their respective as their seed profiles. All leaf nodes in the same seed profile are given equal user preference. These questions are then processed using the KMP algorithm. Fig.4,5 Shows the average response time of the algorithms while varying the seed profile size. It can be conceded that the cost of KMP grows exponentially and exceeds 8 seconds when the profile contains more than 100 nodes.

Fig.6,7 Illustrates the effects of datasets containing different numbers of queries (from 1,000 to 100,000 queries). Apparently, algorithms have linear scalability by the data set size.

5.4 Effective Analysis of Personalization

In this experiment, we assess the absolute search quality on commercial search engines using our UPS framework. The search results are rated with the generalized profile output by KMP over 50 target users. The final search quality is assessed using the Average Precision of the click data of the users.

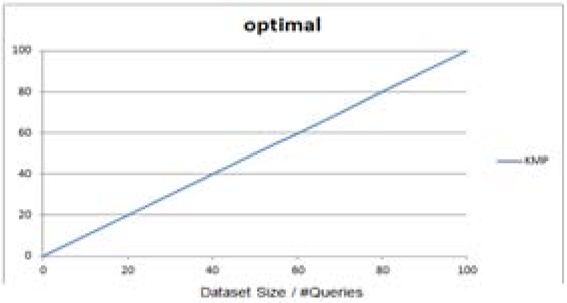


Figure 6: Results of Datasize optimal

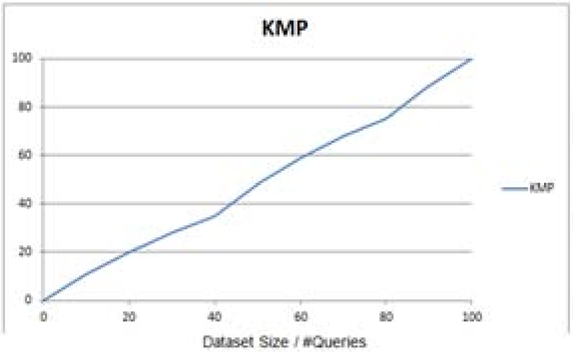


Figure 7: Results of Datasize KMP

Fig.8, 9 Shows the average AP of the ranks before (Original) and after (Fusion) personalizing the test queries on Yahoo and ODP, respectively. From the results of both search engines, we can observe that the improvements of the search quality for Medium Questions and Ambiguous Questions are much more significant than that of Distinct Questions. In particular, the personalization on Distinct Questions of Yahoo results reduces the average performance from 73.4 to 66.2 percent. This is because some irrelevant profile topics are added. The results demonstrate that profile-based personalization is more suitable for queries with small DP.

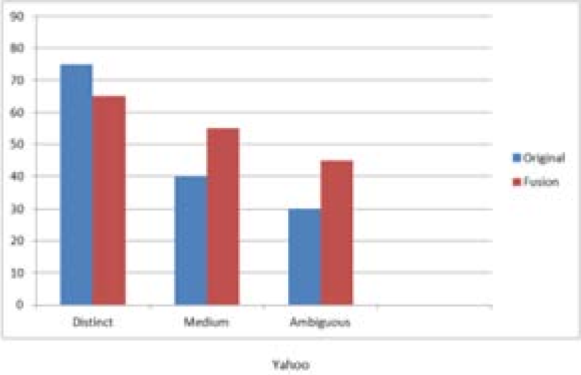


Figure 8: Yahoo

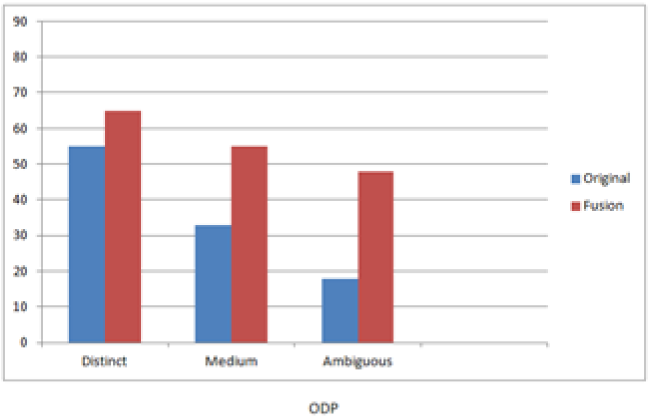


Figure 9: ODP

6. Conclusion and Future Work

A node-side privacy protection framework called User customizable Privacy-preserving Search. Any PWS can adapt the UPS for creating a user profile in a hierarchical taxonomy. UPS allows the user to specify the privacy requirement and thus the personal information of the user profile is kept private without compromising the search quality. Here we used String Matching KMP Algorithm for online generalization. Our experimental results revealed that UPS could achieve quality search results while preserving the user's customized privacy requirements. The results also confirmed the effectiveness and efficiency of our solution. We can implement the hierarchical divisive approach for retrieving the search results. It will give better performance.

References

[1] Lidan Shou, He Bai, Ke Chen, and Gang Chen " Supporting Privacy Preserving In Personalized Web Search".

[2] D. Chum, Untraceable electronic mail, return addresses, and digital pseudonyms, Commun. ACM 24 (2) (1981).

[3] A. Krause and E. Horvitz, "A Utility Theoretic Approach to Privacy in Online Services", Journal of Artificial Intelligence Research 39 (2010).

[4] Y. Xu, K. Wang, B. Zhang, and Z. Chen, "Privacy-Enhancing Personalized Web Search,"

[5] X. Xiao and Y. Tao, "Personalized Privacy Preservation," Proc.ACM SIGMOD Int'l Conf. Management of Data.

[6] Lidan Shou, He Bai, Ke Chen, and Gang Chen, "Supporting Privacy Protection In Personalized Web Search",