Privacy-enhancing personalized web search

By Yabo Xu Benyu Zhang, Zheng Chen , Ke Wang

Presentated by Yanhong Wang

Personalized research necessity

2. large amount of information on the web, it has become increasingly difficult for web search engines to find information that satisfies users' individual needs.
3. Personalized search is a promising way to improve search quality by customizing search results for people with different information goals.

General Personalization approaches:

A good personalization algorithm relies on rich user profiles and web corpus.

3. **By User side:**

Re-ranking query results returned by search engines locally using personal information;

- *Advantage*: better data privacy.
- *Disadvantage*: bandwidth intensive

It requires a large number of search results transmitted to the client before re-ranking.

Alternative solution is to transmit limited amount of information through filtering on the server side, however, it may result in the loss of our desired information.

General Personalization approaches:

1. By Search Engine Side:

Sending personal information and queries together to the search engine; the search engine analyze collected personal information, e.g. personal interests, and search histories and return search results.

Used by Most online personalized search services like Google Personalized Search and Yahoo! My Web.

- *Advantage*: reduce bandwidth intensity
- *Disadvantage*: privacy issues on exposing personal information to a public server.

Data Privacy vs Personalization Benefits

Recent survey conducted by *Choicestrea* indicate more people show concern about their privacy:

- 80% of respondents : interested in personalization remains at a remarkably high
- 32% of respondents : willing to share personal information in exchange for personalized experience, down from 41% in 2004.

However, in practice:

- people give up some privacy to gain economic benefit.
- detailed personal information might not be necessary if it is possible to catch a user's interests at more general level.

Data Privacy vs Personalization Benefits

Objective:

1. Distinguishing between useful information and noise

2. Striking balance between search quality and privacy protection by proper filtering of a user's private information.

Challenges

Characteristic of Personal data:

Unstructured: i.e. browsing history, emails, etc.,

- it is hard to measure privacy.
- it is also difficult to incorporate unstructured data with search engines without summarization.

We need to collect, summarize, and organize a user's personal information into a structured user profile.

The notion of privacy

- highly subjective and depends on the individuals involved.

the user should have control over which parts of the user profile is shared with the server.

Related Work

User profiles building:

- Require users explicitly specify their interests.

  Not practical: users are typically unwilling to spend the extra effort on specifying their intentions. Even if they are motivated, they are not always successful in doing so.

- Implicitly building user profiles to infer a user's intention:

  using a wide range of implicit user activities have been proposed as sources of enhanced search information. This includes a user's search history , browsing history , click-through data , web community , and rich client side information in the form of desktop indices. Our approach is open to all kinds of different data sources for building user profiles, provided the sources can be extracted into text. In our experiments data sources like IE histories, emails and recent personal documents were tested.

Related Work

User profiles representation

- User profiles can be represented by a weighted term vector , weighted concept hierarchical structures like ODP3, or other implicit user interest hierarchy.
- For the purposes of selectively exposing users' interests to search engines, the user profile is a term based hierarchical structure that is related to frequent term based clustering algorithms .
- In this study:

The difference here is that the hierarchical structure is implicitly constructed in a top-down fashion.

And the focus is the relationships among terms, not clustering the terms into groups.

Privacy measurement

3. Difference in prior and posterior knowledge of a specific value:
- This can be formalized as the conditional probability or Shannon's information theory.

2. notion of k-anonymity :
- personally identifying attributes be generalized such that each person is indistinguishable from at least *k*-1 other persons.

3. In this study
- we does not compare information from different users, but rather the information collected over time for a single user.
- addresses unstructured data.

System Overview

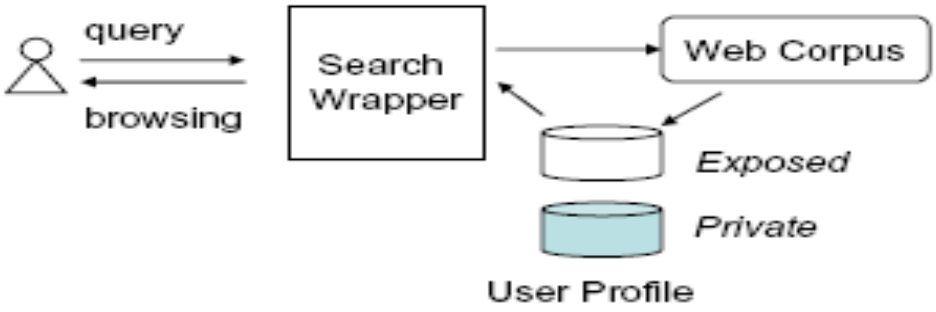

Figure 1. System Overview

1. First, a scalable algorithm automatically builds a hierarchical user profile from available source data.
2. Then, privacy parameters are offered to the user to determine the content and amount of personal information that will be revealed.
3. Third, a search engine wrapper personalizes the search results with the help of the partial user profile.

Constructing a Hierarchical User Profile

we based on frequent terms. In the hierarchy, general terms with higher frequency are placed at higher levels, and specific terms with lower frequency are placed at lower levels.

- ***D*** : the collection of all personal documents and each document is treated as a list of terms.
- **D(*t*)** : all documents *covered* by term *t*, i.e., all documents in which *t* appears, and |D(*t*)| represents the number of documents covered by *t*.
- ***minsup*** : a user-specified threshold, which represents the minimum number of documents in which a frequent term is required to occur.

  A term *t* is *frequent* if $|D(t)| \geq minsup$, where Each frequent term indicates a possible user interest.

Constructing a Hierarchical User Profile

In order to organize frequent terms into a hierarchical structure, relationships between the frequent terms are defined below.

1. ***Similar terms***: *Two terms that cover the document sets with heavy overlaps might indicate the same interest.* Here we use the *Jaccard* function to calculate the similarity between two terms: Sim(tA, tB) = | D(tA)∩D(tB) | / | D(tA)∪D(tB) |. If Sim(tA , tB) > δ, where δ is another user-specified threshold, we take tA and tB as similar terms representing the same interest.
2. ***Parent-Child terms***: *Specific terms often appear together with general terms, but the reverse is not true*. For example, "badminton" tends to occur together with "sports", but "sports" might occur with "basketball" or "soccer", not necessarily "badminton". Thus, tB is taken as a child term of tA if the condition probability P(tA | tB)> δ, where δ is the same threshold in Rule 1.

Constructing a Hierarchical User Profile

$\mathrm{Sim}(tA, tB) \leq \mathrm{P}(tA \mid tB)$, Rule 1 has to be enforced earlier than Rule 2 to prevent similar terms to be misclassified as parent-child relationship.

3. **S*upporting documents* S(tA):** the union of D(tA) and all D(tB), where either $\mathrm{Sim}(tA, tB) > \delta$ or $\mathrm{P}(tA|tB) > \delta$ is satisfied.

   For a term tA, any document covered by tA is viewed as a natural evidence of users' interests on tA. In addition, documents covered by term tB that either represents the same interest as tA or a child interest of tA can also be regarded as supporting documents of tA.

Constructing a Hierarchical User Profile

Algorithm:

Idea:

Using the above rules, our algorithm automatically builds a hierarchical profile in a top-down fashion. The profile is represented by a tree structure, where each node is labeled a term t, and associated with a set of supporting documents S(t), except that the root node is created without a label and attached with D, which represent all personal documents. Starting from the root, nodes are recursively split until no frequent terms exist on any leave nodes.

Below is an example of the process. Before running the algorithm on the documents, preprocessing steps like stop words removal and stemming needs to be performed first.

Constructing a Hierarchical User Profile

D1:sports, badminton
D2:ronaldo,soccer,sports
D3:sex, playboy, picture
D4:sports,soccer,english premier
D5:research, AI, algorithm
D6:research,adpative,personalized, search
D7:Fox, channel, sports, sex
D8:MSN,search
D9:research,AI,neuro network
D10:personalized,search,google, research

minsup = 2
δ = 0.6.
descending order of (document) frequency:
<research:4>
<sports:4>
<search:3>
<peronalized:2>
<soccer:2>
<AI:2>
<sex:2>

Constructing a Hierarchical User Profile

minsup = 2
δ = 0.6.
descending order of (document) frequency:
<research:4> : {D5, D6, D9, D10}.
<sports:4>: {D1, D2, D4, D7}.
<search:3>: {D6, D8, D10}
<peronalized:2>:{D6,D10}
<soccer:2>: {D2,D4}
<AI:2>: {D5,D9}
<sex:2> {D3,D7}

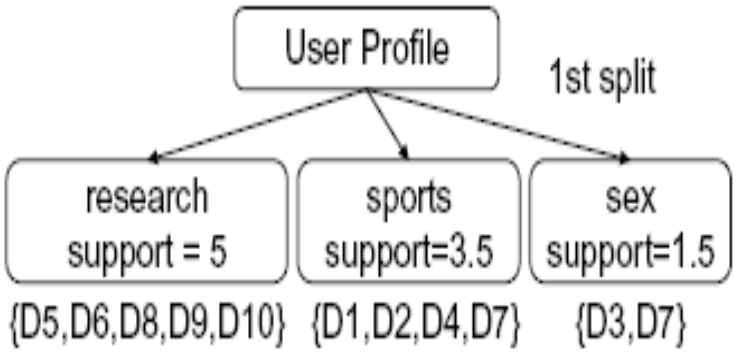

Figure 3. User profile after 1st split.

Sim("search" , "research") = 2/5 $\leq \delta$
P("research" | "search") = 2/3 > δ
Since Rule 2 is satisfied, "search" is taken as a specific term under "research", and D("search") is merged into S("research").
In this example, D7 appears in both S("sports") and S("sex"), so Sup("sports")=1+1+1+1/2=3.5,
and Sup("sex")=1.5.

Constructing a Hierarchical User Profile

minsup = 2, $\delta = 0.6$.
descending order of (document) frequency:
<research:4> : {D5, D6, D9, D10}.
<sports:4>: {D1, D2, D4, D7}.
<search:3>: {D6, D8, D10}
<peronalized:2>:{D6,D10}
<soccer:2>: {D2,D4}
<AI:2>: {D5,D9}
<sex:2> {D3,D7}

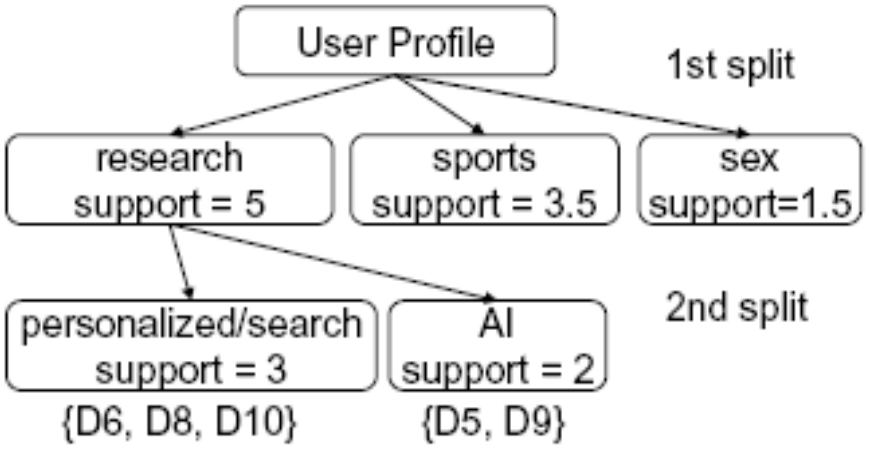

Figure 4.User profile after 2nd split

The node "research" is subsequently examined for further splitting. First S("research") is scanned, and the frequency for each term *t* is counted. Note that any term like "research" that appears in an ancestor node will not be counted again. Frequent terms and their frequency are listed as follows: <search:3>, <personalized:2>, <AI:2>. According to Rule 2, "search" and "personalized" is combined together and the node is labeled "personalized/search" since Sim("search", "personalized") = 2/3>δ. The child nodes after splitting are shown in Figure 4. The splitting can be recursively done until no term is frequent.

Measuring Privacy

Approaches of controlling private information:

3. Grant users full control over the terms in the hierarchy so that they can choose to hide any terms manually as they desire.

- Without any privacy risk
- Users always reluctant to provide any explicit input on their interests

8. Offer users a more convenient way of controlling private information they would agree to have exposed, two parameters derived from information theory are proposed below.

The user profile is established as an indicator of the users' possible individual interests. According to probability theories, the possibility of one interest (or a term) can be calculated as $P(t)=Sup(t)/|D|$.

Measuring Privacy

minDetail: any term *t* in the user profile with P(t)=Sup(*t*)/|*D*|<*minDetail*, will be protected from the server.

If *minDetail* = 0.3, details under the node "sports" are hidden, as well as "sex" that are on the same level with "sports",
for P("sex") = Sup("sex")/|D|=1.5/10 < 0.3.

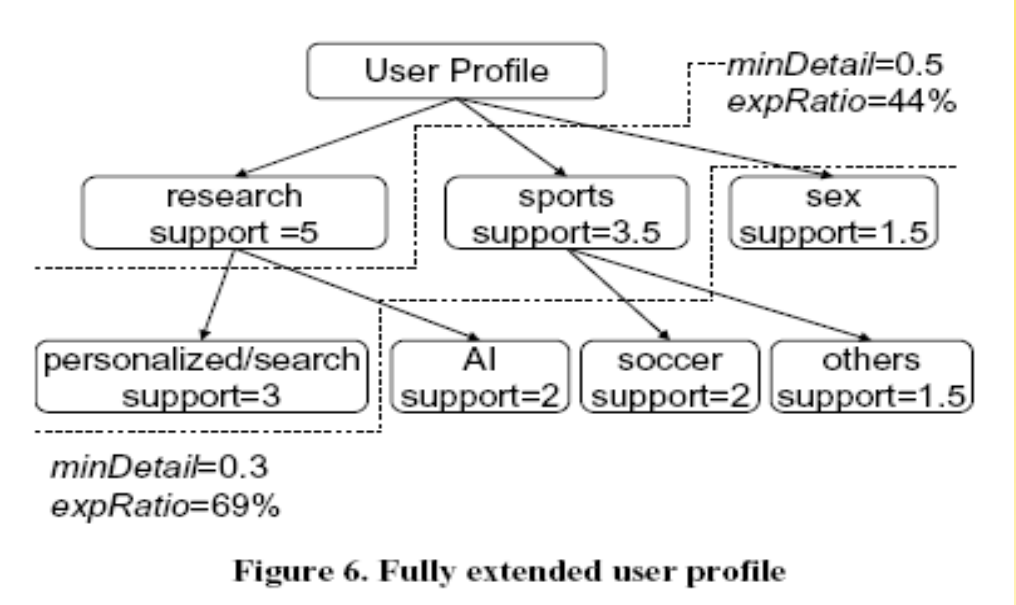

Figure 6. Fully extended user profile

Measuring Privacy

expRatio. The threshold *minDetail* filters specific or sensitive terms by their supports. Still, it is necessary to evaluate the "amount" of private information that is actually protected.
where t corresponds to any of a user's possible interests
$P(t) = \mathrm{Sup}(t)/|D|$.
where t is any term on the leaves of $U[exp]$. Only the leaves are considered as the presence of terms on non-leaf nodes have already been counted by their children. Thus for any threshold *minDetail*, the exposed privacy can be calculated as *expRatio* = $H(U[exp])/H(U)$.

$$H(U[\exp]) = -\sum_{t} P(t) \times \log(P(t))$$

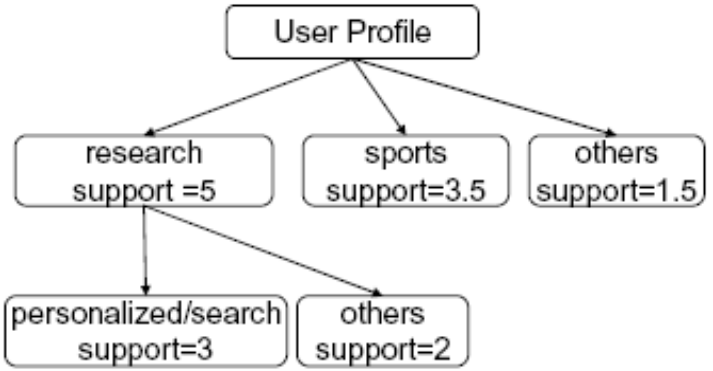

Figure 7. U[exp] when minDetail = 0.3 and expRatio = 69%

Measuring Privacy

Two parameters, *minDetail* and *expRatio*, offer users the ability to determine the content and the amount of private information exposed. As in the example, the lower the *minDetail* quotient, the more information that will be exposed, and *expRatio* will grow in relation to *minDetail*.

The assumption behind two parameters is that more general and frequent terms, which carry smaller self-information, represent information users are more willing to share. Nevertheless, we realize that it might not apply to some extreme cases. For example, a user may have a frequent and general interest in a sensitive topic (i.e. sexuality or politics) that he wants to keep private. Under this circumstance, a beneficial supplement to our solution is to allow users to hide certain branches of user profiles manually. However, more often than not, it is not necessary and a tedious work to most users. Our experiment results verified this.

Personalizing Search Results

Idea:

U[*exp*] *is transformed into a list of weighted terms* where a search wrapper calculates a score for each of the returned search results. The final ranking of the search results is decided by the search engine and U[*exp*].

The weight of each term in U[*exp*] *is estimated by applying the* concept of IDF(Inverse Document Frequency)

Given a term *t, the weight of t, denoted by wt,* is calculated as:

$$wt = log(|D| / Sup(t))$$

|D| : ***the total number of documents (or total* support),**

Sup(*t*) : ***the support of this term on the node in* U[*exp*].**

The partial user profile is expressed by a list <t, wt>,

t: ***a term in U[exp]***

wt : ***the weight.***

***Take U[exp] in* Figure 7 as an example. The list is <research, 0.301>, <sports, 0.456>, personalized/search, 0.523>. The anonymous node labeled "others" is ignored.**

Personalizing Search Results

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 $<t, wt>$ pairs.
2. 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 MSN search, called *MSNRank*. These links are passed to the partial user profile.
3. For each of the returned link l, a score called *UPScore* is calculated by the partial user profile as follows:

$$UPScore(l) = \Sigma \text{tWt} \times \text{tf}$$

where t is any term in the partial user profile, and tf is the frequency of the term t in the webpage of the link l. An *UPRank* is assigned to each link according to its *UPScore*, and the link with the highest *UPScore* will be ranked first.

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

$$PPRank = \alpha * UPRank + (1 - \alpha) * MSNRank,$$

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

Personalizing Search Results

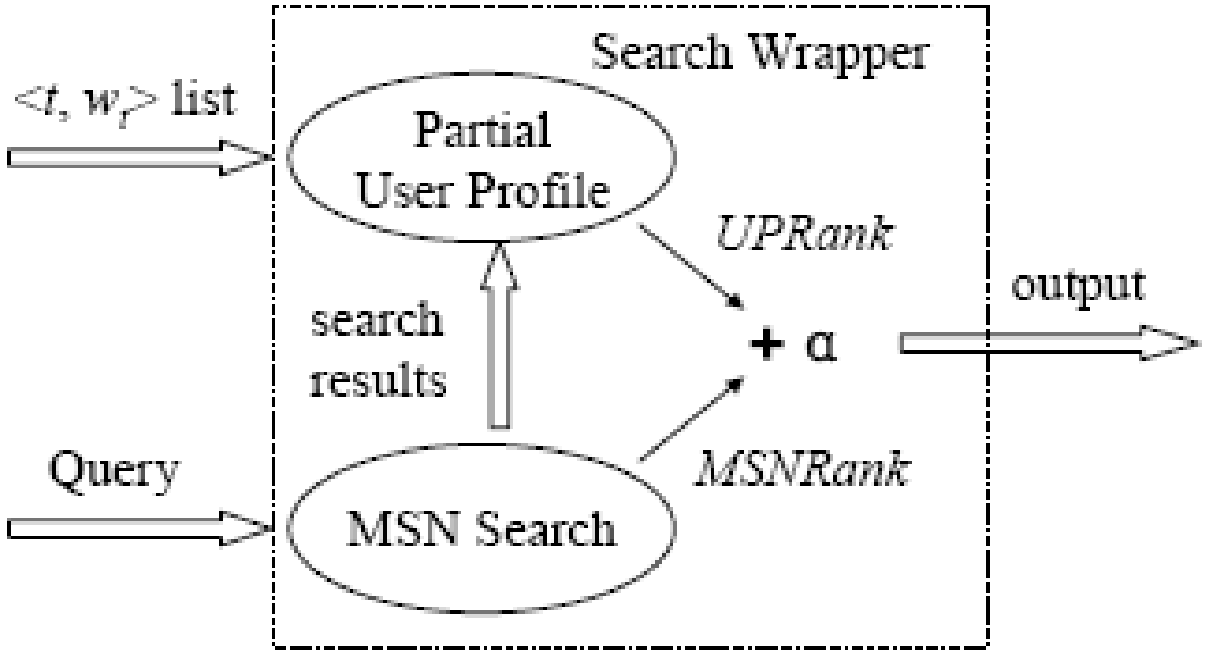

Figure 8. The workflow in the search wrapper

Experiment(Experiment Setup)

1. **Objectives:**

   to verify the effectiveness of the user profile to help improve search quality, and to explore the relationship between search quality and personal privacy.

4. **Setup:**

   In the user interface, three parameters could be adjusted:

(1) ***personal data available for building a user profile*** - the choices given to the user were internet browsing history,emails, personal documents or any combinations thereof;

(2)***minDetail*** – the threshold offered to a user for determining which part of user profile is exposed. For any given *minDetail*, *expRatio* is updated to indicate the amount of information currently exposed;

(3) ***α*** – the weight assigned to the user profile ranking.

Experiment (Effective of User Profile)

- **All parameters fixed:** ***minDetail*=0, *expRatio*=100%, α=0.5**

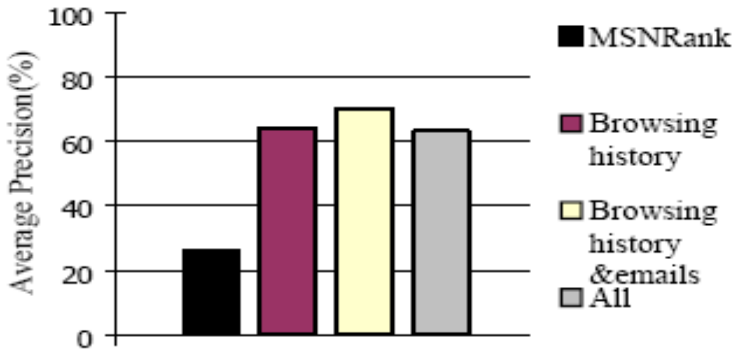

Figure 9. Effect of different personal data options.

The average precision that incorporates the user profile is much higher than the original *MSNRank*, and the search quality improves. additional personal information does not always yield better results. The best search quality is achieved when data sources are set as browsing history and emails. The user profiles built from "all" personal data, including browsing history, emails and recent documents, have a similar performance to using only browsing history. Recent documents seem to have the negative effect on search quality because some of extremely lengthy documents introduce morenoise than useful information.

Experiment (Effective of User Profile)

Within the same group of queries, the impact of the user profile for *PPRank* is studied by varying only parameter α.

- **The personal data options are set to browsing history & emails,**
- ***minDetail* = 0, and**
- ***expRatio* = 100%.**
- **Parameter α varies from 0 to 1, where α=1 indicates ranking search results by *UPScore* only, and α=0 showsthe results from the original MSN search ranking.**

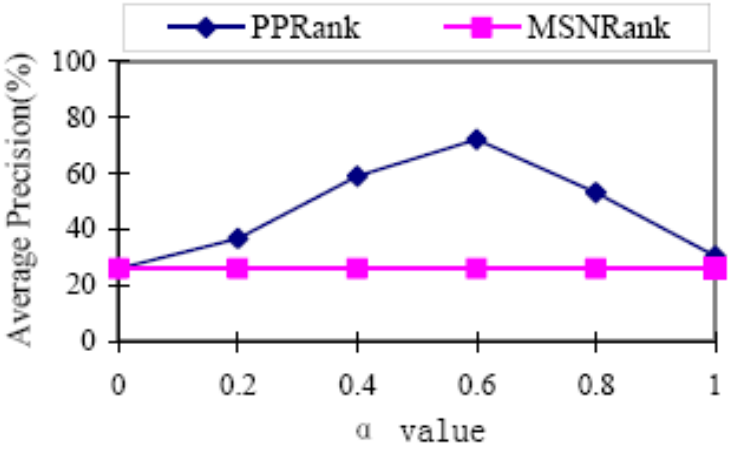

Figure 10. Impact of different α value

Experiment (Privacy vs Search Quality)

In this experiment, users are required to try different privacythresholds to explore the relationship between privacy preservation and search quality. For each query, all parameters are fixed (personal data options are set to browsing history & emails, $\alpha = 0.6$). *expRatio* will be updated in relation to a specified *minDetail*.

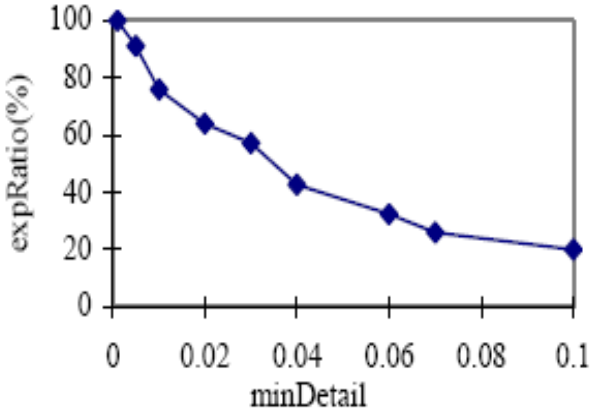

Figure 11. minDetail vs expRatio

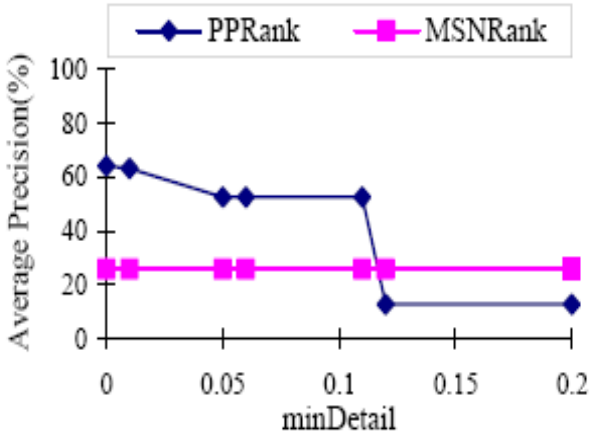

Figure 13. minDetail vs Search Quality

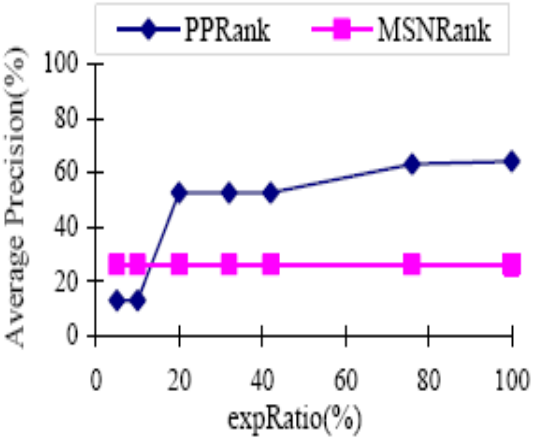

Figure 12. expRatio vs Search Quality

Experiment (Manual Privacy Option)

- Problem: The aforementioned privacy parameters *minDetail* and *expRatio*, incorporating the hierarchical term-based user profile, offer users a convenient way to determine the extent to which personal information is exposed. This relies on the assumption that more general and frequent terms, which carry smaller self-information, represent information users are more willing to share. However, as we discussed in section 4.2, in some extreme cases a user may have a frequent and general interest in a sensitive topic that he wants to keep private.
- Solution: the client program provides users the interface of hiding certain branches of user profiles manually. Consistently, any term labeled as private results in hiding all terms under this branch. This facilitates a user who has to perform manual privacy option as he only needs to examine only a few high-level terms.

CONCLUSIONS AND FUTURE WORK

- This paper targets at bridging the conflict needs of personalization and privacy protection, and provides a solution where users decide their own privacy settings based on a structured user profile.
- Benefits:

4. *Offers a scalable way to automatically build a hierarchical user profile on the client side.*
5. *Offers an easy way to protect and measure privacy.*

- Experiments showed that the user profile is helpful in improving search quality when combined with the original MSN ranking.
- there is an opportunity for users to expose a small portion of their private information while getting a relatively high quality search. Offering general information has a greater impact on improving search quality.

Thank you for your attention!