Volume 2, Issue 10 October 2016 ISSN NO: 2454 - 7514

North Asian International Research Journal Consortium

North Asian International Research Journal Of Science, Engineering and Information Technology

NAIRJC JOURNAL PUBLICATION

North Asian International Research Journal Consortium

Welcome to NAIRJC

ISSN NO: 2454 -7514

North Asian International Research Journal of Science, Engineering & Information Technology is a research journal, published monthly in English, Hindi. All research papers submitted to the journal will be double-blind peer reviewed referred by members of the editorial board. Readers will include investigator in Universities, Research Institutes Government and Industry with research interest in the general subjects

Editorial Board

Address: -North Asian International Research Journal Consortium (NAIRJC) 221 Gangoo, Pulwama, Jammu and Kashmir, India - 192301, Cell: 09086405302, 09906662570, Ph. No: 01933-212815,
Email: nairjc5@gmail.com, nairjc@nairjc.com, info@nairjc.com Website: www.nairjc.com

IRJIF IMPACT FACTOR: 3.821

SUPPORTING MULTIFEATURE PREFERRNCE AND PRIVACY PROTECTION IN PERSONALIZED WEB SEARCH (PWS)

M. RAJA*

***Reserarch Scholar, Department of Computer Science, Chikkanna Govt. Arts College, Tirupur – 2.**

MRS. B. HEMALATHA**

****MCA, B.Ed., M.Phil., Asst. Prof. & Head, Department of Computer Science, Chikkanna Govt. Arts College, Tirupur – 2.**

Abstract: ***Today Internet and web search engines have become an important part in ones day to day life. Present search engines generally handle search queries or keywords without considering user preferences or contexts in which users submit their queries. To provide more relevant and effective results to user, Personalization technique is used. Personalized web search refer to search information that is tailored specifically to a person's interests by incorporating information about query provided. Several personalized web search techniques are currently used which are based on web contents, web link structure, browsing history, user profiles and user queries. In the existing Work, the personalized search mainly leverages semantic features extracted from user history and ignores other nonsemantic latent features, let's alone adapt to preference distribution on non-semantic features. To solve this problem, we propose an adaptive model for multi-feature preferences, in which we adapt latent non-semantic features extracted from visited pages to reflect diverse aspects of user preferences. We also proposed 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.***

I. INTRODUCTION

The Internet has become a common source of information for billions of people, and for certain users, inevitably there is much more useless information than useful information, which poses more challenges for users to target what they want. The web search engine has long become the most important portal for ordinary people looking for useful information on the web. However, users might experience failure when search engines return irrelevant results that do not meet their real intentions. Such irrelevance is largely due to the enormous variety of users' contexts and backgrounds, as well as the ambiguity of texts. Personalized web search (PWS) is a general category of search techniques aiming at providing better search results, which are tailored for individual user needs. As the expense, user information has to be collected and analyzed to figure out the user intention behind the issued query.

The solutions to PWS can generally be categorized into two types, namely click-log-based methods and profile-based ones. The click-log based methods are straightforward they simply impose bias to clicked pages in the user's query history. Although this strategy has been demonstrated to perform consistently and considerably well, it can only work on repeated queries from the same user, which is a strong limitation confining its applicability. In contrast, profile-based methods improve the search experience with complicated user-interest models generated from user profiling techniques. Profile-based methods can be potentially effective for almost all sorts of queries, but are reported to be unstable under some circumstances. Although there are pros and cons for both types of PWS techniques, the profile-based PWS has demonstrated more effectiveness in improving the quality of web search recently, with increasing usage of personal and behavior information to profile its users, which is usually gathered implicitly from query history, browsing history, click-through data , bookmarks,, user documents, and so forth. Unfortunately, such implicitly collected personal data can easily reveal a gamut of user's private life.

Search engines are considered to be an effective solution for efficiently using large data and helping to find relevant results on the huge information on the Internet, but because of their diversity of preferences, it is still difficult to filter out different users' Expected results. There are many differences among users, who prefer documents with different characteristics (e.g., sports themes, long blogs), and the distribution of preferences for multiple features varies from user to user. We should build an adaptive model for multi-feature preferences to maintain consistency with a variety of user preferences.

In order to deal with the problem of user preference diversity, the current research mainly focuses on the topic relevance between query and document to reduce the ambiguity of query. Several studies have explored several potential features (e.g., amount of images and page length) to analyze their impact on user retention time and user preference for web pages, but little attention is paid to all of these potential features are used to improve the performance of personalized search[1].,. In addition, although Bennett, et al [2]. Learn to adapt to the user's time preference changes, this work still cannot adapt to different users of different preferences distribution. Ignoring the various distributions that augment the user's preferences, the results do not fully satisfy the needs of the user.

Considering the expanded feature quantities and the diverse preference distributions among these features, it is very challenging to propose an accurate model to dynamically adapt to different user preferences. In order to solve the above problems and corresponding challenges, we have formulated the above problem of user preference diversity as the task of multi-feature preference adaptation strategy of different users. In this paper, we propose an adaptive multi-feature model (AMM) to cover the diversity of user preferences by adaptively selecting the accessed documents to reflect user preferences.

In addition to that, privacy concerns have become the major barrier for wide proliferation of PWS services.

To protect user privacy in profile-based PWS, researchers have to consider two contradicting effects during the search process. On the one hand, they attempt to improve the search quality with the personalization utility of the user profile. On the other hand, they need to hide the privacy contents existing in the user profile to place the privacy risk under control. A few previous studies suggest that people are willing to compromise privacy if the personalization [2]., by supplying user profile to the search engine yields better search quality. In general, there is a tradeoff between the search quality and the level of privacy protection achieved from generalization. Unfortunately, the previous works of privacy preserving PWS are far from optimal. The existing methods do not take into account the customization of privacy requirements. This probably makes some user privacy to be overprotected while others insufficiently protected.

The above problems are addressed in the proposed system called UPS (literally 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 [3]., while retaining their usefulness for PWS. UPS is distinguished from conventional PWS in that it

1) Provides runtime profiling, which in effect optimizes the personalization utility while respecting user's privacy requirements

2) Allows for customization of privacy needs

3) It does not require iterative user interaction.

Our contributions are listed below:

- We construct an adaptive model with multiple potential features that re-order the search results more satisfactorily than the fixed selection of user history.
- We bring a number of non-semantic features and semantic features, and with all of these features, we can systematically and comprehensively represent user preferences.
- We develop a new algorithm to accommodate a variety of preference distributions, helping to more accurately deploy features that reflect user preferences.
- We evaluated our model based on a real-world data set, and the results show that our model is clearly better able to satisfy different users.
- Finally, we also propose a privacy-preserving personalized web search framework UPS, which can generalize profiles for each query according to user-specified privacy requirements.

    - We develop two simple but effective generalization algorithms, Greedy DP and Greedy IL, to support runtime profiling. While the former tries to maximize the discriminating power (DP), the latter attempts to minimize the information loss (IL).
    - We provide an inexpensive mechanism for the client to decide whether to personalize a query in UPS.
    - 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.

II. PROBLEM DEFINITION

Previous works on profile-based PWS mainly focus on improving the search utility. The basic idea of these works is to tailor the search results by referring to, often implicitly, a user profile that reveals an individual information goal. In the remainder of this section, we review the previous solutions to PWS on two aspects, namely the representation of profiles, and the measure of the effectiveness of personalization. Many profile representations are available in the literature to facilitate different personalization strategies. Earlier techniques utilize term lists/vectors or bag of words to represent their profile. However, most recent works build profiles in hierarchical structures due to their stronger descriptive ability, better scalability, and higher access efficiency. The majority of the hierarchical representations are constructed with existing weighted topic hierarchy/graph, such as ODP1, Wikipedia, and so on. Another work builds the hierarchical profile automatically via term-frequency analysis on the user data.

To adapt to user preference diversity, we propose an adaptive model, which extracts multiple latent features to reflect preferences systematically, evaluates user profile adaptively, and eventually obtains better search results effectively. We construct the new framework of the adaptive multi-feature model for personalized search, in which with user profiles implying both semantic features and non-semantic features of user clicked documents, we evaluate user preferences on original results separately from semantic and non-semantic perspectives, finally re-ranking [5]., the original search results are acquired with the help of these two preference evaluations.

First, we will concentrate on constructing adaptive user profile to comprehensively reflect user preferences.

In second stage, we mainly evaluate user preferences on non semantic features, because semantic features can be concluded into a high level features topic by analyzing words and word frequency, and there are plenty of algorithms to evaluate user's semantic preferences according to semantic similarity between these results and visited documents in user profile. Unlike the semantic features, non semantic features are scattered and related to each other sophisticatedly, we then bring forward a novel algorithm evaluating non-semantic preferences by filtering out visited representative documents in user profile and scoring the original search results in terms of non-semantic similarity between these results and aforementioned documents.

In third stage, we incorporate the scores of search results according to both semantic and non-semantic preferences, and re-rank search results using Lambda MART

Finally, to maintain the privacy in PWS, we propose a UPS framework, which can potentially adopt any hierarchical representation based on taxonomy of knowledge.

Meanwhile, this proposed work is distinguished from previous studies as it also proposes two predictive metrics, namely personalization utility and privacy risk, on a profile instance without requesting for user feedback.

The Proposed UPS framework is that can adaptively generalize profiles by queries while respecting user- specified privacy requirements. This 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. The Proposed system presents two greedy algorithms, namely Greedy DP and Greedy IL, for runtime generalization. It also provides an online prediction mechanism for deciding whether personalizing a query is beneficial. Extensive experiments demonstrate the effectiveness of this framework. The experimental results also reveal that GreedyIL significantly outperforms GreedyDP in terms of efficiency. UPS presents the procedures carried out for each user during two different execution phases, namely the offline and online phases. Generally, the offline phase constructs the original user profile and then performs privacy requirement customization according to user-specified topic sensitivity. The subsequent online phase finds the Optimal Risk Generalization solution in the search space determined by the customized user profile.

III. EXISTING SYSTEM

Profile based PWS

A user profile is typically generalized for only once offline, and used to personalize all queries from a same user indiscriminatingly. Such "one profile fits all" strategy certainly has drawbacks given the variety of queries. Profile-based personalization may not even help to improve the search quality for some ad hoc queries, though exposing user profile to a server has put the user's privacy at risk. A better approach is to make an online decision on whether to personalize the query and what to expose in the user profile at runtime.

Customization of privacy requirements

This considers, all the sensitive topics are detected using an absolute metric called surprisal based on the information theory, assuming that the interests with less user document support are more sensitive.

Iterative user interactions

This Method usually refine the search results with some metrics which require multiple user interactions, such as rank scoring, average rank, and so on. This paradigm is, however, infeasible for runtime profiling, as it will not only pose too much risk of privacy breach, but also demand prohibitive processing time for profiling. Thus, the system needs predictive metrics to measure the search quality and breach risk after personalization, without incurring iterative user interaction.

Disadvantages: The existing profile-based PWS do not support runtime profiling. The existing methods do not take into account the customization of privacy requirements. Many personalization techniques require iterative user interactions when creating personalized search results.

IV. PROPOSED METHOD:

In this paper, we propose an adaptive multi-feature model (AMM) to cover user preference diversity by adaptively selecting visited documents to reflect user preferences.

Our contributions are listed in the following:

- We build an adaptive model for multiple latent features to re-rank the search results more satisfactorily than the fixed selection of user history.
- We bring up multiple non-semantic features together with the semantic features, and with all these features, we can represent user preference systematically and comprehensively.
- We develop a novel algorithm to adapt to the diverse preference distributions which contribute to more precise deployment of features reflecting user preferences.
- We also propose a privacy-preserving personalized web search framework UPS, which can generalize profiles for each query according to user-specified privacy requirements.

Advantages: The proposed system provides faster search and accurate results. In addition to that it provides an inexpensive mechanism for the client to decide whether to personalize a query in UPS.

A. Adaptive Multi feature Model (Web Search)

To adapt to user preference diversity, we propose an adaptive model, which extracts multiple latent features to reflect preferences systematically, evaluates user profile adaptively, and eventually obtains better search results effectively. Fig.1 illustrates the framework of the adaptive multi-feature model for personalized search, in which with user profiles[4]., implying both semantic features and non-semantic features of user clicked documents, we evaluate user preferences on original results separately from semantic and non-semantic perspectives, finally re-ranking the original search results are acquired with the help of these two preference evaluations. In Section A, we will concentrate on constructing adaptive user profile to comprehensively reflect user preferences. In Section B, we mainly evaluate user preferences on non semantic features, because semantic features can be concluded into a high level features topic by analyzing words and word frequency, and there are plenty of algorithms to evaluate user's semantic preferences according to semantic similarity between these results and visited documents in user profile. Unlike the semantic features, non semantic features are scattered and related to each other sophisticatedly, we then bring forward a novel algorithm evaluating non-semantic preferences by filtering out visited representative documents in user profile and scoring the original search results in terms of non-semantic similarity between these results and aforementioned documents. In Section C, we incorporate the scores of search results according to both semantic and non-semantic preferences, and finally re-rank search results using Lambda MART.

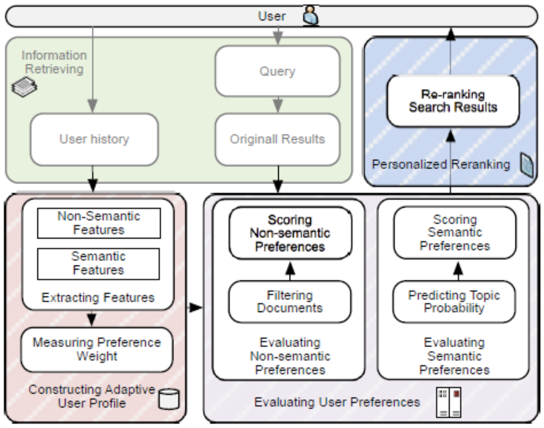


Figure 1. The Framework of the Adaptive Multi-feature Model

Constructing Adaptive User Profile

User profile is full of documents clicked before to reflect user preferences. We denote each document di as a multi dimension vector di = ffi1; fi2; :::; fimg where fik is kth feature of di, and m is the total number of features. Since the user may prefer some of the documents more than others, we attach a preference weight wi to document di based on user historic preferences. Finally, we obtain the user profile Du = f(d1;w1(t)); (d2;w2(t)); :::; (dn;wn(t))g where n is the total number of documents in Du. In order to construct adaptive user profile, we first need to systematically extract features and dynamically measure preference weight. Semantic Features are leveraged to analyze the topical relevance between queries, search results and clicked documents. Non-semantic Features determine the underlying layout of pages and directly affect user experience. Instead of only focusing on semantical features like most of the recent researches, we mine multiple latent non-semantic features directly from documents in the user history. Non-semantic features can imply users' preferences efficiently as well as semantic features. For example, users prefer entertaining documents with more multimedia resources which can be reflected by semantic features like topics and non-semantic features like the frequency of HTML tags <meta>. Thus we include all existing features and construct the following two sets of features.

Semantic Features

A document can have several topics which can be leveraged to reveal user preferences on semantic features. These topics are always extracted by topic models which attempt to probabilistically uncover the underlying semantic structure of a collection of documents.

Non-semantic Features

A document is built by words and format, we then classify non-semantic into two categories which are literal and constructive features. Literal features measure the content length and reading difficulty. Page Length is the total number of words in the page and reading difficulty is measured by Dale-Chall readability formula to predict the document readability which corresponds to American school grade levels. Constructive features determine the layout of pages, i.e., page size and the frequency of each of the 128 HTML tags which are not visible to users but they play important roles in determining the page formatting and layout that affect users' feeling, e.g., the amount of images can be reflected by the amount of <img>. We dynamically crawl each visited documents in which we first downloads the backbone page to count tags' frequency. In addition, we crawl the page content to extract topics, measure content length and reading difficulty. We then downloads all the secondary URLs (e.g., javascript, flash, image, etc.) to get the page size. After extracting aforementioned features from all visited documents, we attach each document with a preference weight. Since preferences always change with time, to adapt to dynamic preference change, we introduce decay function to reflect the current preferences. In the time window t,

we integrate the history preference weight wi(t - 1) to formulates the current preference weight wi(t) for di. The preference weight wi(t) is described as below.

$$w_i(t) = w_i(t-1) \times e^{-(t-last)\times\frac{\log 2}{hl}+\frac{click_i^t}{\sum click_i^t}}, \quad (1)$$

where t-last denotes the time since the last clicked, decay factor hl is set to 7 because empirical analysis shows user interests reduce to 1/2 in a week, and clickt i denotes the click numbers in the time window t which is a key parameter for estimating user satisfaction.

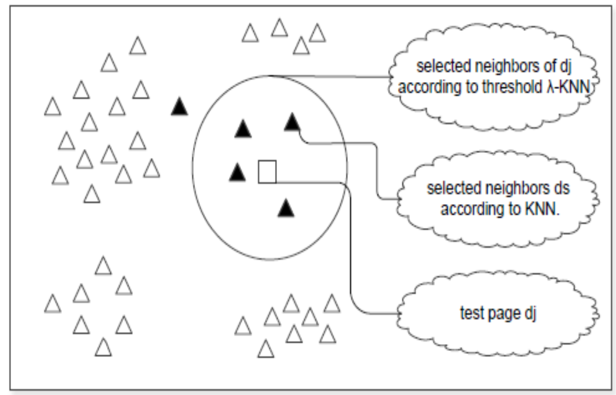


Figure 2. Illustration of KNN

In this way, we construct user profile Du which cannot only reflect the diverse aspects of user preferences but also adapt to dynamic change of user preferences.

B. Evaluating Non-semantic Preferences:

To re-rank original search results, we need to evaluate user preferences on the result documents separately from semantic and non-semantic sides. Since non-semantic features are scattered and related to each other sophisticatedly, we then try to filter out the documents that can reflect user preferences on specific non-semantic features accurately. For example, a user profile contains both short documents with low readability and long documents with high readability.

In this case, while we estimate whether a specific short search result can satisfy this user, it doesn't have to be similar to the two types of documents, and we just need to compare this search result with selected history documents representing preference of this user on short documents. If this search result is similar to the corresponding documents in user profile, it should be recommended to the user. How to effectively select the documents considering the diverse distribution of user preferences on multiple features? There are two ways, one is to cluster documents into groups and only calculate the similarity in the same group.

However, unsupervised learning of clustering algorithm like K-means spends too much time on interactions to get stable results, and fix number K may limit the accuracy of clustering. The other way is to employ K-Nearest Neighbor (KNN) algorithm for personalized search However, just as K- means, the fixed rigid number K may lead to involvement of the unrelated documents and reduce the accuracy, thus we need an algorithm to obtain adaptive number of similar documents accurately.

Considering all these problems, we propose a novel algorithm named KNN to filter out visited representative documents by both quantity limitation k and similarity limitation _. Fig.2 illustrates the workings of algorithm KNN to select neighbors for comparison, where the square denotes result document dj, triangles denote documents in the user profile, black triangles are documents selected by KNN, and the large circle denotes the neighborhood within the threshold.

Similar to the traditional KNN algorithm, we first select k nearest documents of document dj. Specially, we recheck whether the similarity between each selected neighbor ds and document dj is smaller than the threshold, and we filter documents if the similarity is larger than the threshold. We only need to compare the search results with the selected documents, which reflect the user preferences on non-semantic features specific to the features of search results, thus k should not be too large.

On the other hand, if we don't compare the search results with sufficient visited documents, we cannot ensure these results coincident to user preferences on theses non-semantic features, therefore, k should not be too small. Obviously, there is a tradeoff on k selection, and similarly, we should make a tradeoff on selection. With KNN, we obtain the documents in userprofile relevant to a specific search result and then we can accurately estimate whether this specific search result can satisfy this user.

In this way, we compare the search results with the selected documents according to their non-semantic similarity for scoring the original search results in terms of user preferences on non-semantic features. We use the non-semantic preferences score n Score j to measure the non-semantic similarity between the search results and the selected documents, which is formulated as following.

$$nScore_j = \sum_{d_i \in D_{ug}} w_i(t) \cdot sim(d_i, d_j), \qquad (2)$$

Where wi(t) is formulated by Equation 1, Dug is the group of documents we selected to represent user preferences, and sim(di; dj) denotes the similarity between search result dj and document di in the Dug as below

$$sim(d_i, d_j) = 1 - \frac{\sqrt{\sum_{k=1}^{m} (f_{ik} - f_{jk})^2}}{m}, \qquad (3)$$

where we normalize each feature into [0; 1] since features in the vector are in different scope. Eventually, nScorej can adapt to diverse preference distribution on non-semantic features by selecting a group of relevant visited documents in user profile. In this way, KNN algorithm improves the adaption and thus guarantees the accuracy in re-ranking process.

C. Re-ranking Search Results

In this section, we incorporate both semantic and non semantic preference to re-rank the original search results. To evaluation user preferences on semantic features, we match search results with user non-semantic preferences with semantic preference score semScorejl which can be calculated by personalized LDA. In detail, it is estimated.

Table I

FEATURES FOR PERSONALIZED RE-RANKING.

Feature	Description
Document Feature	
nScore	The non-semantic preference score by compare between the document and the user profile.
Doc-Query Features	
Original Rank	Rank of the document on the initial returned list.
semScore	The semantic preference score by calculating probability of the document to be clicked under a query for a user.

as the probability of the document dj to be clicked under a query ql for a user u,

$$\begin{aligned} semScore_{jl} &= P(d_j|q_l, u) \\ &\propto P(d_j) \prod_{w \in q_l} \sum_{z} P(w|z) P(u|z)^{\delta} P(z|d), \end{aligned} \qquad (4)$$

where z means topic, w means word, weights the amount of user's preference influence on the overall ranking, and P(dj) denotes the frequency document dj is clicked among all documents in the selected group. Finally, we combine semantic preference score nScore and non-semantic preference score semScore as features for a learning to rank (LTR) algorithm LambdaMART to re-rank the search results, and all the features we used are presented in Table I. LambdaMART is a type of gradient boosted decision tree including multiple feature input, and has been selected as the base learning algorithm of approaches to search personalization. Lambda MART is not the determinant to our model, and any reasonable learning to rank algorithm would likely provide similar results based on our model.

V. UPS FRAMEWORK FOR PRIVACY:

The Proposed system propose a PWS framework called UPS that can adaptively generalize profiles by queries while respecting user- specified privacy requirements. This 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. The Proposed system presents two greedy algorithms, namely GreedyDP and GreedyIL, for runtime generalization. It also provides an online prediction mechanism for deciding whether personalizing a query is beneficial. Extensive experiments demonstrate the effectiveness of this framework. The experimental results also reveal that GreedyIL significantly outperforms GreedyDP in terms of efficiency. UPS presents the procedures carried out for each user during two different execution phases, namely the offline and online phases. Generally, the offline phase constructs the original user profile and then performs privacy requirement customization according to user-specified topic sensitivity. The subsequent online phase finds the Optimal Risk Generalization solution in the search space determined by the customized user profile. Each user has to undertake the following procedures in the Proposed System:

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

The greedy method is a general algorithm design paradigm, built on the following elements:

- Configurations: different choices, collections, or values to find
- Objective function: a score assigned to configurations, which we want to either maximize or minimize

It works best when applied to problems with the greedy-choice property:

A globally-optimal solution can always be found by a series of local improvements from a starting configuration.

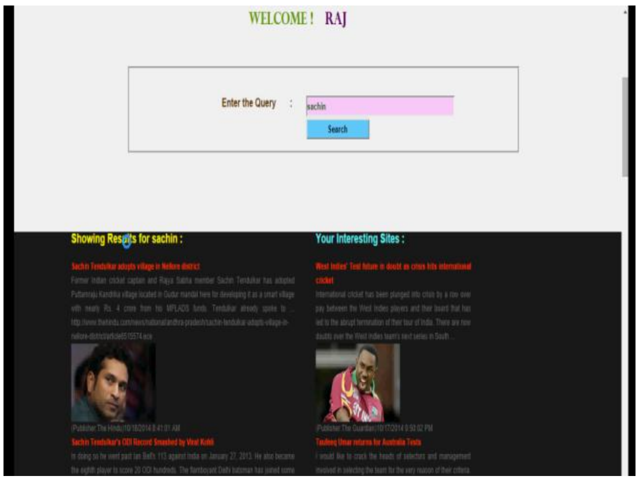
Given a string X, efficiently encode X into a smaller string Y Saves memory and/or bandwidth
A good approach: Huffman encoding

- Compute frequency f(c) for each character c.
- Encode high-frequency characters with short code words
- No code word is a prefix for another code

- Use an optimal encoding tree to determine the code words

ALGORITHM USED FOR WEBSEARCH:

RESULT:



VI. CONCLUSION

In this paper, we represent adaptive multi-feature user profile which takes consider of features systematically reflecting user preferences, adapts to the dynamic change of user preferences. In addition, by a novel adaptation algorithm KNN, our model can adapt to the diverse distribution of user preferences in re-ranking process. This project also presented a client-side privacy protection framework called UPS for personalized web search. This framework allowed users to specify customized privacy requirements via the hierarchical profiles. In addition, UPS also performed online generalization on user profiles to protect the personal privacy without compromising the search quality. In general, there is a tradeoff between the search quality and the level of privacy protection achieved from generalization. The results also confirmed the effectiveness and efficiency of the proposed system. Future work will explore further aspects of the interaction between different features to improve user experience through personalization.

REFERENCES:

1. Dou, Z., Song, R., Wen, J. R.: A large-scale evaluation and analysis of personalized search strategies. In: WWW 2007, pp. 581-590. ACM (2007)
2. Bennett, P. N., White, R. W., Chu, W., Dumais, S. T., Bailey, P., Borisyuk, F., Cui, X.: Modeling the impact of short- and long-term behavior on search personalization. In: SIGIR 2012, pp. 185-194. ACM (2012)
3. Harvey, M., Crestani, F., Carman, M. J.: Building user profiles from topic models for personalised search. In: CIKM 2013, pp. 2309-2314. ACM (2013)
4. Vu, T., Willis, A., Tran, S. N., Song, D.: Temporal latent topic user profiles for search personalisation. In: ECIR 2015, pp. 605-616. Springer, Heidelberg (2015)
5. Xu, S., Jiang, H., Lau, F. C. M.: Mining user dwell time for personalized web search re-

ranking. In: IJCAI 2011, pp. 2367- 2372. Morgan Kaufmann Publishers Inc. (2011)

6. Liu, C., White, R. W., Dumais, S.: Understanding web browsing behaviors through weibull analysis of dwell time. In: SIGIR 2010, pp. 379-386. ACM (2010)
7. Butkiewicz, M., Madhyastha, H. V., Sekar, V.: Understanding website complexity: measurements, metrics, and implications. In: SIGCOMM, pp. 313-328. ACM (2011)
8. Jiang, J., Hassan Awadallah, A., Shi, X., White, R. W.: Understanding and Predicting Graded Search Satisfaction. In: WSDM 2015, pp. 57-66. ACM (2015)
9. Sugiyama, K., Hatano, K., Yoshikawa, M.: Adaptive web search based on user profile constructed without any effort from users. In: WWW 2004, pp. 675-684. ACM (2004)
10. Li, L., Yang, Z., Wang, B., Kitsuregawa, M.: Dynamic adaptation strategies for long-term and short-term user profile to personalize search. In Advances in Data and Web Management, pp. 228-240. Springer, Heidelberg (2007)
11. Eickhoff, C., Collins-Thompson, K., Bennett, P. N., Dumais, S.: Personalizing atypical web search sessions. In: WSDM 2013, pp. 285-294. ACM (2013)
12. Yan, J., Chu, W., White, R. W.: Cohort modeling for enhanced personalized search. In: SIGIR 2014, pp.505-514. ACM (2014)
13. Kim, J. Y., Collins-Thompson, K., Bennett, P. N., Dumais, S. T.: Characterizing web content, user interests, and search behavior by reading level and topic. In: WSDM 2012, pp. 213- 222. ACM (2012)
14. Wu, Q., Burges, C. J.C. Svore, K. M., and Gao, J.: Ranking, boosting, and model adaptation. Microsoft Research Technical Report MSR-TR-2008-10 (2008)